FANUC Robot series

R-50*i*A/R-50*i*A Mate CONTROLLER

Optional Function

OPERATOR'S MANUAL

B-84464EN-2/03

© FANUC CORPORATION, 2023

Translation of the original instruction
The original language of this manual is English. The other language manuals are a translation from the English one.

Thank you very much for purchasing FANUC Robot.
Before using the Robot, be sure to read the "FANUC Robot series SAFETY HANDBOOK (B-80687EN)" and understand the content.

- No part of this manual may be reproduced in any form.
- The appearance and specifications of this product are subject to change without notice.
- Any copyright and other intellectual property rights regarding this manual shall belong to FANUC.
- The safety precautions described in this manual may no longer be accurate and up-to-date due to revisions or changes in laws and regulations. Before using our products, each user must confirm the latest laws and regulations.

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

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

SAFETY PRECAUTIONS

This chapter describes the precautions which must be followed to enable the safe use of the robot. Before using the robot, be sure to read this chapter thoroughly.

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

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

1 PERSONNEL

Personnel can be classified as follows.

Operator:
- Turns the robot controller power ON/OFF
- Starts the robot program from operator panel

Programmer or Teaching operator:
- Operates the robot
- Teaches the robot inside the safeguarded space

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

- The operator is not allowed to work in the safeguarded space.
- The programmer or teaching operator and maintenance technician are allowed to work in the safeguarded space. Work carried out in the safeguarded space include transportation, installation, teaching, adjustment, and maintenance.
- To work inside the safeguarded space, the person must be trained on proper robot operation.

Table 1 (a) lists the work outside the safeguarded space. In this table, the symbol "○" means the work allowed to be carried out by the specified personnel.

Table 1 (a) List of work outside the Safeguarded Space

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

During robot operation, programming and maintenance, the operator, programmer, teaching operator and maintenance technician take care of their safety using at least the following safety protectors:

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

2 DEFINITION OF SAFETY NOTATIONS

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

Symbol	Definitions
⚠**WARNING**	Used if a danger resulting in the death or serious injury of users are expected to occur if they fail to observe the approved procedure.
⚠**CAUTION**	Used if a danger resulting in the minor or moderate injury of users or equipment damage is expected to occur if they fail to observe the approved procedure.
NOTE	Used if a supplementary explanation not related to any of WARNING and CAUTION is to be indicated.

PREFACE

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

ABOUT MANUAL

About this manual

FANUC Robot series OPERATOR'S MANUAL (Basic Operation) (B-84464EN) describes how to operate the FANUC Robot. It is controlled by the FANUC R-50*i*A and R-50*i*A Mate controller (called the robot controller hereinafter) containing the FANUC Robot software.

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

Related manuals

The following manuals are available:

Robot controller	Optional Function OPERATOR'S MANUAL **B-84464EN-2** (This manual)	Topics: Description of the software optional functions. Use: Application designing, introducing, adjusting the robot at the work site, and , guide to teaching
	OPERATOR'S MANUAL (Basic Operation) **B-84464EN**	Topics: Functions, operations and the procedure for operating the robot. Programming procedure, interface and alarm. Use: Application designing, introducing, adjusting the robot at the work site, and , guide to teaching
	OPERATOR'S MANUAL (Alarm Code List) **B-84464EN-1**	Topics: Error code listings, causes, and remedies. Use: Installing and activating the system, connecting the mechanical unit to the peripheral device and maintenance the robot system.
	Arc Welding Function OPERATOR'S MANUAL **B-84464EN-3**	Topics: Description of the setting and operation for arc welding application software. Use: Application designing, introducing, adjusting the robot at the work site, and , guide to teaching
	Spot Welding Function OPERATOR'S MANUAL **B-84464EN-4**	Topics: Description of the setting and operation for spot welding application software. Use: Application designing, introducing, adjusting the robot at the work site, and , guide to teaching
	Dispense Function OPERATOR'S MANUAL **B-84464EN-5**	Topics: Description of the setting and operation for dispense application software. Use: Application designing, introducing, adjusting the robot at the work site, and , guide to teaching
	MAINTENANCE MANUAL **B-84375EN (for R-50*i*A), B-84675EN (for R-50*i*A Mate)**	Topics: Installing and activating the system, connecting the mechanical unit to the peripheral device and maintenance the robot system.

Mechanical unit	OPERATOR'S MANUAL	Topics: Installing and activating the robot, connecting the mechanical unit to the controller, maintaining the robot. Use: Guide to installation, activation, connection, and maintenance.

TABLE OF CONTENTS

SAFETY PRECAUTIONS s-1

PREFACE p-1

1 OVERVIEW 1

2 *i*RCalibration Signature 3

2.1 Overview 3
2.2 Benefits of Signature Calibrated Robot 4
2.3 Custom Signature Calibration 5
2.3.1 Calibration System 5
2.3.2 Calibration Type 6
2.3.3 Custom Signature Calibration Preparation 7
2.3.4 Position Program 8
2.3.5 Coordinate Systems 10
2.4 Custom Signature Calibration Procedure 12
2.4.1 Custom Signature Calibration Preparation 12
2.4.2 Display Signature Screen 14
2.4.3 Select Calibration Type 15
2.4.4 Setup Calibration settings 16
2.4.5 Create Calibration Program Screen 17
2.4.6 Run Calibration Program Screen 18
2.4.7 Calculate Parameters Screen 21
2.4.8 View/Apply Calibration Result 22
2.4.9 Create Signature Calibration Parameters Setup File 23
2.5 Advanced Function 24
2.5.1 Calibration Recovery 24
2.5.2 TCP/Uframe Calibration 26
2.5.3 Reference Frame Function 28
2.5.4 Enable/Disable iRCalibration Signature Compensation 29
2.6 iRCalibration Signature Status Screen 29
2.7 Mastering 31
2.7.1 Mastering for Signature Calibrated Robot 31
2.7.2 Quick Mastering 31
2.7.3 Monitoring the Change of Mastering Data 32
2.8 Cautions 32
2.8.1 Accuracy Around Singularity 32
2.8.2 Restrictions 33
2.9 Reference 33
2.9.1 Measuring the robot installation position 33
2.9.2 Design of jig for mounting reflector 37

3 SOFTFLOAT FUNCTION 40

3.1 JOINT / CARTESIAN SOFTFLOAT 40
3.2 PUSHOUT SOFTFLOAT 45

3.3 SOFTFLOAT GUIDELINES 48
3.4 CARTESIAN SOFTFLOAT GUIDELINES 50
3.5 CAUTIONS AND LIMITATIONS 52

4 CONTINUOUS TURN FUNCTION 54
4.1 OVERVIEW 54
4.2 SETUP 55
4.3 USING CONTINUOUS TURN 57
4.3.1 Starting Continuous Turn 57
4.3.2 Stopping Continuous Turn 57
4.3.3 Execution Notes 57
4.3.4 Continuous Turn Speed instruction example 58
4.4 NOTES AND LIMITATIONS 58
4.4.1 Functions that cannot be used together 58
4.4.2 Useing Continuous Turn with Constant Path 59
4.4.3 Other notes 59
4.5 RESETTING THE ROTATION OF POSITIONER AND INDEPENDENT AXES USING A KAREL PROGRAM 60
4.5.1 CALIBNBT.PC Conditions 60
4.5.2 Using CALIBNBT.PC 61
4.5.3 Reset to a specified range 61
4.5.4 Execution Notes 62
4.5.5 Error messages 62

5 OPERATION GROUP DO OUTPUT FUNCTION 64

6 AUTOMATIC ERROR RECOVERY FUNCTION 66
6.1 AUTOMATIC ERROR RECOVERY FUNCTION OVERVIEW 66
6.2 RESUME PROGRAM FUNCTION 66
6.3 FAST EXIT/ENTRY FEATURE 67
6.4 RESUME_PROG INSTRUCTION 67
6.4.1 RESUME_PROG [1: Comment] 68
6.4.2 CLEAR_RESUME_PROG 69
6.5 RETURN_PATH_DSBL INSTRUCTION 70
6.5.1 RETURN_PATH_DSBL 70
6.6 MAINT_PROG INSTRUCTION 71
6.6.1 MAINT_PROG [1:Comment] 72
6.7 SETTING UP THE AUTOMATIC ERROR RECOVERY FUNCTION 74
6.7.1 Automatic Error recovery function common setup 75
6.7.2 RESUME PROGRAM type recovery (Setup items related to the Resume program function) 81
6.7.3 MAINTENANCE PROGRAM type recovery (Setting items related to the Fast exit/entry function) 83
6.8 MANUAL FUNCTIONS SCREEN FOR THE RESUME PROGRAM FUNCTION 85
6.8.1 Displaying detail conditions of the Error recovery status DO 86
6.9 EXECUTION OF THE RESUME PROGRAM FROM THE TEACH PENDANT AND TEST MODE 87
6.10 OTHER SPECIFICATIONS AND RESTRICTIONS 88
6.10.1 Automatic Error Recovery function 88
6.10.2 Resume program function 88

6.10.3 Fast exit/entry feature....88
6.11 WARNINGS (Be sure to read this section for safety.)....89
7 REMOTE TCP FUNCTION....90
7.1 SETUP....92
8 HIGH-SENSITIVITY COLLISION DETECTION....95
8.1 SPECIFICATIONS....95
8.2 SETTINGS....95
8.3 COLLISION DETECT SETUP SCREEN....96
8.4 PROGRAM INSTRUCTIONS....97
8.4.1 COL GUARD ADJUST....97
8.5 CAUTIONS....100
9 LOAD ESTIMATION....101
9.1 OPERATING PROCEDURE....101
9.2 LOAD ESTIMATION PROCEDURE (for 6-Axis Robots)....101
9.2.1 Preparation....102
9.2.2 Estimation Motion Setup....104
9.2.3 Performing Estimation and Applying the Result....107
9.2.4 Estimation Result....109
9.3 CALIBRATION PROCEDURE (for 6-Axis Robots)....109
9.4 OTHER RELATED MATTERS....112
9.4.1 Motion range....112
9.4.2 Calibration data....112
9.4.3 Payload setting for estimating motion....112
9.4.4 Collision Detection for estimating motion....113
10 PAYLOAD OVER AND PAYLOAD SETTING CONFIRM FUNCTION114
10.1 LIMITATIONS....115
10.2 PROCEDURE....115
10.2.1 Procedure to Show Payload Over and Payload Setting Confirm Screen....115
10.2.2 Procedure to Use Payload Setting Confirm (When Payload is Changed)....115
10.2.3 Procedure to Use Payload Over Confirm (When The Program or Payload is Changed)....116
11 GRAVITY COMPENSATION....118
11.1 SETUP....118
11.2 MASTERING....118
11.2.1 Normal Mastering and GC Mastering....118
11.2.2 Checking and Choosing the Mastering Method....119
11.2.3 Mastering Procedure....120
11.2.4 GC Mastering Guidance....121
12 OPERATION LOG BOOK....123
12.1 RECORDED EVENTS....125
12.2 SETTING UP BOOK....132
12.3 OPERATIONS....133
12.4 EXTENDED ALARM LOG....137
12.4.1 Setup....137
12.4.2 How to Display Alarm Log....138

13 PROGRAM TOOLBOX 140
13.1 SOFT LIMIT SETTING 140

14 ADVANCED CONSTANT PATH 143
14.1 LINEAR DISTANCE 143
14.1.1 How Linear Distance Works 144
14.1.2 Linear Distance Limitations 146
14.1.3 Using Linear Distance 146
14.2 CORNER REGION AND CORNER DISTANCE 147
14.2.1 Limitations of Corner Size by Distance between Taught Points 148
14.2.2 Corner Region and Corner Distance Limitations 149
14.3 PROCESS SPEED 150
14.3.1 Process Speed Limitations 151
14.4 MAX SPEED 151
14.4.1 Max Speed Limitations 152

15 SINGULARITY AVOIDANCE FUNCTION 153
15.1 HOW TO USE SINGULARITY AVOIDANCE 153
15.1.1 How to Use Singularity Avoidance in Jogging 153
15.1.2 Using Singularity Avoidance in a TPE program 154
15.2 SYSTEM CONFIG MENU 154
15.3 LIMITATIONS 155
15.4 CAUTIONS 155

16 PATH SWITCHING FUNCTION 156
16.1 PS INSTRUCTION 156
16.2 TRIGGER POINT OF PS INSTRUCTION 158
16.3 ABOUT MULTI-GROUP CONFIGURATIONS 161
16.4 LIMITATIONS 161
16.5 CAUTIONS 161
16.6 SYSTEM CONFIG MENU 162

17 ALARM RECOVERY 163
17.1 Alarm Recovery Faults [CHOICE] Screens 164
17.2 User Alarm Recovery 166

18 DUAL MOTOR DRIVE 172
18.1 SETUP OF DUAL DRIVE PAIR FOR EXTENDED AXIS (J518), INDEPENDENT AXIS (H895) AND POSITIONER (H874, H896) 173
18.1.1 Condition 173
18.1.2 Add Dual Drive axes 174
18.1.3 Display / Modify / Clear Dual Drive settings 176
18.1.4 Mastering of Dual Drive axes 177
18.2 SETUP OF DUAL DRIVE PAIR FOR 1-AXIS POSITIONER 178

19 MULTI UOP INTERFACE FUNCTION 181
19.1 PERIPHERAL I/O 182
19.1.1 Setting of the Number of UOP Set 183
19.1.2 Modification and Addition of Signal 184
19.1.3 HOLD#1 to #N 185
19.1.4 CSTOPI#1 to #N 186

19.1.5 START#1 to #N 186
19.1.6 RSR 187
19.1.7 PNSTROBE#1 to #N,PROD_START#1 to #N 187
19.1.8 CMDENBL#1 to #N 187
19.1.9 SYSRDY#1 to #N 187
19.1.10 PROGRUN#1 to #N,PAUSED#1 to #N 188
19.1.11 HELD#1 to #N 188
19.1.12 FAULT#1 to #N 188
19.1.13 ATPERCH 189
19.2 SELECT PROGRAM 189
19.2.1 Setting of Motion Groups Intended by Each UOP Set 190
19.2.2 Select Program at Running or Paused 191
19.2.3 Program Edit Screen Display When Operating Motion Group is Changed 191
19.3 RUN PROGRAM 192
19.3.1 Robot Service Request (RSR) 192
19.3.2 Program Number Select (PNS) 192
19.4 STOP PROGRAM 193
19.4.1 Pause Program by UOP 193
19.4.2 Abort Program by UOP 193
19.4.3 Other 193
19.5 SYSYTEM CONFIG SCREEN ITEMS 193

20 ERROR CODE OUTPUT FUNCTION 196
20.1 SPECIFICATIONS 196
20.1.1 Types of Alarms 196
20.1.2 Input and Output Signals 196
20.2 ALARM CODE DEFINITIONS 197
20.2.1 Alarm Severity 198
20.2.2 Alarm ID 199
20.2.3 Alarm Number 200

21 DATA MONITOR FUNCTION 201
21.1 DATA MONITOR SETUP 203
21.2 DATA MONITOR SCHEDULE 213
21.3 PROGRAMMING 216
21.4 DATA MONITOR CHART 217
21.5 DATA MONITOR OPERATION 218

22 BRAKE CHECK 220
22.1 INITIAL SETTING BEFORE USE 220
22.2 START BRAKE CHECK 222
22.3 RESULTS OF BRAKE CHECK 223
22.4 LIMITATIONS 224
22.5 CAUTION 224

23 NC PROGRAM CONVERSION 225
23.1 OVERVIEW 225
23.1.1 NC program syntax 225
23.1.2 Specifications 225
23.2 OPERATION 226
23.2.1 Conversion Operation 226
23.2.1.1 Load NC program in FILE menu 226

23.2.1.2 Copying an NC program to the MD drive 227
23.3 NC PROGRAM SYNTAX 227
23.3.1 O code – Program Name/Number 227
23.3.2 N code – Line numbers 228
23.3.3 G code – Motion modifiers 228
23.3.4 F code – Feed Rate / Speed 229
23.3.5 Position data 229
23.3.6 K code - Config string 230
23.3.7 M code – Miscellaneous codes 230
23.3.7.1 Predefined M-codes 231
23.3.7.2 Special M-codes 231
23.3.7.3 Predefined M-codes 231
23.3.7.4 Other (user-defined) M-codes 233
23.4 NC PROGRAM CONVERSION EXAMPLE 233

24 ENHANCED MIRROR IMAGE 236
24.1 PARALLEL MIRROR IMAGE 237
24.2 ROTATIONAL MIRROR IMAGE 239
24.3 MIRROR IMAGE USING EXISTING FRAMES AND MIRROR PLANES, WITH CONTROLLED ORIENTATION 240
24.4 MIRROR IMAGE OF EXTENDED AXES 241

25 VIA INSTRUCTION 251
25.1 HOW TO USE 251
25.1.1 DEFAULT 251
25.1.2 FINE 251
25.1.3 CNT 251
25.2 LIMITATIONS 252

26 LEARNING PATH CONTROL 253
26.1 OVERVIEW 253
26.1.1 Procedure 253
26.2 HARDWARE AND SOFTWARE 253
26.2.1 Hardware 253
26.2.2 Software 253
26.3 LPC SETUP MENU 254
26.3.1 “Main Program” ITEM 254
26.3.2 “Start Line” ITEM 254
26.3.3 “End Line” ITEM 255
26.3.4 “Geometric Type” ITEM 255
26.3.5 “Geometric Data” ITEM 255
26.3.6 “Process Speed” ITEM 256
26.3.7 “Change Blend-In Speed” ITEM 256
26.3.8 “Path Tolerance” ITEM 256
26.3.9 “Orientation Control” ITEM 256
26.3.10 “Auto Change CIR to L” ITEM 256
26.3.11 “Max. Iterations” ITEM 256
26.3.12 “Process Disabled” ITEM 257
26.3.13 “LEARN” FUNCTION KEY 257
26.3.14 “RESULT” FUNCTION KEY 257
26.3.15 “APPLY” FUNCTION KEY 257
26.3.16 RELATED VIEW 257

26.4 LPC LEARNING AND PLAYBACK 259
26.4.1 Learning 259
26.4.2 Limitations 259
26.4.3 Production with LPC Playback 259
26.5 LPC BACKUP 260

27 TORQUE CONTROL 261
27.1 DYNAMIC TORQUE LIMIT 261
27.1.1 SETTING UP THE DYNAMIC TORQUE LIMIT 262
27.1.2 PROGRAM INSTRUCTION 265
27.1.3 LIMITATIONS 267
27.1.4 CAUTIONS 269
27.2 PERMANENT TORQUE LIMIT FOR ADDITIONAL MOTOR 270
27.2.1 SETTING UP PERMANENT TORQUE LIMIT 270
27.2.2 DISPLAY / MODIFY PERMANENT TORQUE LIMIT 271
27.2.3 COMBINED USE OF PERMANENT TORQUE LIMIT AND DYNAMIC TORQUE LIMIT 273

28 TCP SPEED OUTPUT 275
28.1 LIMITATIONS 275
28.2 SETTING UP TCP SPEED OUTPUT 276
28.3 TCP SPEED OUTPUT INSTRUCTION 278

29 TP DRAM/FILE STORAGE FUNCTION 280
29.1 STORAGES 280
29.1.1 CMOS Programs 280
29.1.2 SHADOW Programs 280
29.1.3 SHADOW ONDEMAND Programs 281
29.1.4 FILE Programs 281
29.2 STORAGE CONFIGURATION 281
29.3 SAVE / LOAD PROGRAMS 286
29.3.1 Save / Load TP Files 286
29.3.2 Copy Programs 287
29.3.3 Save / Load LS Files 287
29.3.4 Make Backup of Programs 287
29.4 LOADING PROCESS IN PROGRAM EXECUTION 287
29.5 PROGRAM EXCHANGE FUNCTION WITHOUT ENOUGH MEMORY SPACE 288
29.6 PRECAUTION 289
29.6.1 Cause and Remedy for Alarm Occurrence 289

30 CYCLE TIME LOGGING 291
30.1 DISPLAY MODES OF CYCLE TIME 292
30.1.1 Cycle Mode 292
30.1.2 Hourly Mode 293
30.1.3 Line-by-Line Mode 293
30.2 LOGGING CYCLE TIME 294
30.2.1 Change Display Format of Cycle Time Data 294
30.2.2 Save Cycle Time Data 295
30.2.3 Update Cycle Time Data 295
30.2.4 Display Target Cycle Time Line 295
30.2.5 Display Reference Cycle Time Line 296

30.3 EXECUTION EXAMPLE OF SPOT PROGRAM 297

31 MATH FUNCTION INSTRUCTION 299

31.1 TYPE OF MATH FUNCTIONS 299
31.2 INSTRUCTION FORMAT OF MATH FUNCTION 299
31.2.1 Instruction Format of Assignment Statements 300
31.2.2 Instruction Format of Relational Statements 300
31.2.3 Instruction Format of Wait Command Statements 301
31.3 FUNCTION SPECIFICATION OF MATH FUNCTIONS 301
31.3.1 Square Root（SQRT） 301
31.3.2 Trigonometric Function（SIN） 301
31.3.3 Trigonometric Function（COS） 302
31.3.4 Trigonometric Function（TAN） 302
31.3.5 Inverse Trigonometric Function（ASIN） 302
31.3.6 Inverse Trigonometric Function（ACOS） 303
31.3.7 Inverse Trigonometric Function（ATAN2） 303
31.3.8 Inverse Trigonometric Function（ATAN） 304
31.3.9 Exponent 304
31.3.10 Natural Logarithm 304
31.3.11 Absolute（ABS） 305
31.3.12 Truncate（TRUNC） 305
31.3.13 Round Off（ROUND） 305
31.4 BACKGROUND OPERATION OF MATH FUNCTION 306
31.5 TEACH MATH FUNCTION INSTRUCTION 306
31.6 RESTRICTION OF TEACHING MATH FUNCTION 308
31.7 EXCEPTIONS AND RESTRICTION 309

32 SERVO TOOL CHANGE FUNCTION 310

32.1 OUTLINE 310
32.1.1 Feature of Function 310
32.1.2 Basic Specification 310
32.1.3 Restrictions, prohibitions, and precautions 310
32.1.4 System Configuration 312
32.1.5 Outline of Servo Tool Change Installation 313
32.2 INITIAL SETUP 314
32.3 PRELIMINARY TOOL ATTACHING OPERATION 314
32.4 TOOL CHANGE SETUP 316
32.5 SETTING THE REFERENCE POSITION 322
32.5.1 Battery-less Tools 322
32.5.2 Battery-Containing Tools 322
32.6 TOOL CHANGE INSTRUCTION 323
32.6.1 TOOL DETACH Instruction 323
32.6.2 TOOL ATTACH Instruction 323
32.6.3 Example Program 323
32.6.4 Forward Execution (FWD) 324
32.6.5 Backward Execution (BWD) 324
32.7 TOOL CHANGE STATUS 324
32.8 TEACHING 325
32.8.1 Points to keep in mind when teaching 325
32.8.2 Example Program 326

32.9 SERVO TOOL CHANGE SCREENS....326
32.10 TOOL CHANGE INITIAL SETUP....327
32.10.1 Setting Motion Parameters for Servo Tool Axes....327
32.10.2 Assigning Tool Numbers to Servo Tool Axes....327
32.11 TOOL CHANGE REFERENCE POSITION SETUP METHOD (BATTERY-LESS SERVO TOOL TYPE)....329
32.11.1 Reference Position Setup for Calibration Types 3 and 4....329
32.11.2 Reference Position Setup for Calibration Types 5 and 6....331
32.11.3 Quick Mastering Reference Position Setup....333
32.12 TROUBLESHOOTING....334
32.12.1 Attach Instruction is Executed when the Tool is not Attached.....334
32.12.2 Robot Stopped during Calibration.....334
32.12.3 Calibration Motion Failed.....335
32.12.4 Different Tool from that Specified by the Attach Instruction is Attached.....335
32.12.5 Attached Tool has been Detached by Mistake (without Using the Detach Instruction).....336
32.12.6 Tool Axis of a Detached Tool has Moved.....336
32.12.7 Battery Voltage has Dropped....336
32.12.8 Battery Ran Low while the Tool was Detached.....336

33 OPERATION WITHOUT SHIFT FUNCTION....338
33.1 JOG OPERATION WITHOUT SHIFT....338
33.2 TEST EXECUTION OPERATION WITHOUT SHIFT....339

34 HIGH SPEED SHIFT KEY FUNCTION....342
34.1 SETUP FOR HIGH SPEED SHIFT KEY FUNCTION....342
34.2 HIGH SPEED SHIFT JOG OPERATION....342
34.3 HIGH SPEED SHIFT TEST EXECUTION OPERATION....343

35 *i*RDIAGNOSTICS....345
35.1 ROBOT CONDITION DETECTION....345
35.1.1 Setup....345
35.1.2 Program Execution....347
35.1.3 Robot Condition Detection Status....347
35.1.4 After Replacement....348
35.2 MOTION PROFILER....348
35.2.1 Setup....349
35.2.2 Results....349
35.2.2.1 Details....351
35.2.2.2 Detailed Data Comparison....355

36 MENU UTILITY....357
36.1 PROMPT BOX MESSAGE MENU....358
36.2 PROMPT BOX YES/NO MENU....363
36.3 LIST MENU....368
36.4 STATUS MENU....374
36.5 OPERATOR ENTRY MENU....383
36.6 USING YOUR TEACH PENDANT MENUS....390

37 4D GRAPHICS FUNCTION....392
37.1 OVERVIEW....393
37.1.1 Graphic Models....393

37.1.2 Operation 393
37.2 4D GRAPHICS SCENE 396
37.2.1 4D GRAPHICS Display Scene 396
37.2.1.1 Visual jog 396
37.2.1.2 Jog preview 397
37.2.1.3 Set visibility: 4D GRAPHICS display 398
37.2.2 4D GRAPHICS Node Map 399
37.2.2.1 4D GRAPHICS edit node map 400
37.2.2.2 4D GRAPHICS select node map 401
37.2.2.3 Set Visibility: node map position numbers 401
37.2.2.4 Node Map supported data 401
37.2.3 4D GRAPHICS Frame Display 402
37.2.4 4D GRAPHICS TCP Trace 402
37.2.5 4D GRAPHICS Position Register 403
37.2.5.1 Set Visibility: position register numbers 404
37.2.6 4D GRAPHICS DCS 404
37.3 4D EDITOR FUNCTION 405
37.3.1 Overview 405
37.4 FULL SCREEN 4D DISPLAY 406
37.4.1 Overview 406
37.4.2 Setup 406
37.4.2.1 Requirements 406
37.4.3 Operation 406
37.4.3.1 Accessing the Full Screen 4D Display 406
37.4.3.2 View adjustment mode 408
37.4.3.3 Scene visibility 409

38 DATA TRANSFER BETWEEN ROBOTS FUNCTION 411
38.1 TERMINOLOGY 411
38.2 SETUP 412
38.3 TCP/IP SETUP FOR ROBOGUIDE 413
38.4 STANDARD DATA TRANSFER PROGRAMS 414
38.4.1 Program to Get a Numeric Register 414
38.4.2 Program to Set a Numeric Register 415
38.4.3 Program to Get a Position Register 415
38.4.4 Program to Set a Position Register 416
38.5 RECOVERY FROM ERROR 417
38.6 KAREL BUILT-INS 418
38.6.1 RGET_PORTCMT Built-in ROUTINE 419
38.6.2 RGET_PORTSIM Built-in ROUTINE 420
38.6.3 RGET_PORTVAL Built-in ROUTINE 420
38.6.4 RGET_PREGCMT Built-in ROUTINE 421
38.6.5 RGET_REG Built-in ROUTINE 422
38.6.6 RGET_REG_CMT Built-in ROUTINE 422
38.6.7 RGET_SREGCMT Built-in ROUTINE 423
38.6.8 RGET_STR_REG Built-in ROUTINE 424
38.6.9 RNUMREG_RECV Built-in ROUTINE 424
38.6.10 RNUMREG_SEND Built-in ROUTINE 425
38.6.11 RPOSREG_RECV Built-in ROUTINE 426
38.6.12 RPOSREG_SEND Built-in ROUTINE 427
38.6.13 RSET_INT_REG Built-in ROUTINE 428
38.6.14 RSET_PORTCMT Built-in ROUTINE 428
38.6.15 RSET_PORTSIM Built-in ROUTINE 429

38.6.16 RSET_PORTVAL Built-in ROUTINE....430
38.6.17 RSET_PREGCMT Built-in ROUTINE....430
38.6.18 RSET_REALREG Built-in ROUTINE....431
38.6.19 RSET_REG_CMT Built-in ROUTINE....432
38.6.20 RSET_SREGCMT Built-in ROUTINE....432
38.6.21 RSET_STR_REG Built-in ROUTINE....433
38.7 TIMEOUT AND RETRY....434
38.8 LIMITATIONS....434
38.9 CAUTIONS....434
38.10 CONFLICT BETWEEN DATA WRITES....435
38.11 TROUBLESHOOTING....435

39 TOUCH SENSING....438
39.1 TOUCH SENSING I/O ASSIGNMENT....440
39.2 TOUCH FRAME SETUP....442
39.3 SEARCH PATTERN....445
39.4 TOUCH SCHEDULES....449
39.5 TOUCH SENSING PROGRAMMING....453
39.6 EXECUTION OF TOUCH SENSING PROGRAM....460
39.7 TOUCHING UP OF TOUCH SENSING PROGRAM....462
39.8 MULTIPLE SEARCHES....465
39.9 COORDINATED TOUCH SENSING....466

40 TOUCH SKIP FUNCTION....469
40.1 TOUCH SKIP SCREEN....469
40.2 TOUCH SKIP PROGRAM....469
40.3 NOTES / LIMITATIONS....470

41 MROT INSTRUCTION....471
41.1 HOW TO USE MROT....471
41.2 LIMITATIONS....471

42 SMALL CIRCLE ACCURACY IMPROVEMENT....473
42.1 SPECIFICATIONS....473
42.2 SETTINGS....473
42.2.1 PARAMETER SETTINGS....473
42.2.2 MACRO PROGRAM : SV_ADJST.TP....474
42.2.3 COMPENSATION VALUE SETTING SCREEN....474
42.3 PROGRAMMING....475
42.4 TROUBLE SHOOTING (TUNING GUIDELINES)....475
42.4.1 LEMON SHAPE CIRCLE....476
42.4.2 LEMON SHAPE CIRCLE WITH RIPPLES....476
42.4.3 CIRCLE WITH FLAT SPOT....476
42.4.4 CIRCLE WITH INWARD BUMP....477

43 SHAPE GENERATION FUNCTION....478
43.1 OVERVIEW....478
43.1.1 Shape Generation Programs....479
43.2 SHAPE GENERATION SETUP....479
43.2.1 Basic Setup....480
43.2.2 Common Settings for All Shapes....483

43.2.3 Circle Shape Setup 484
43.2.4 Slot Shape Setup 485
43.2.5 Hexagon Shape Setup....486
43.2.6 Rectangle Shape Setup....487
43.2.7 Keyhole Shape Setup 487
43.2.8 Pommel Shape Setup....488
43.2.9 Custom 489
43.3 SHAPE GENERATION UTILITY 491
43.3.1 UTILITIES Shape Settings 491
43.3.2 Creating and Editing a Shape program 491

44 MATERIAL HANDLING GRIPPER OPTION 494
44.1 SETTING UP A MATERIAL HANDLING TOOL 494
44.1.1 Selecting the Handling Config (SpotTool+ only) 495
44.1.2 Setting Up Tooling Valves....497
44.1.3 Setting Up Tooling Valve Signals....505
44.1.4 Configuring Tooling Valve I/O....507
44.1.5 Controlling Valve I/O....509
44.1.6 Importing and Exporting Material Handling Gripper xml 511
44.1.7 Example mhgripcfg.xml file....513
44.2 CONTROLLING A TOOLING VALVE IN MANUAL MODE 515
44.3 MATERIAL HANDLING VALVE MACRO PROGRAMS 516
44.3.1 Examples 516
44.3.2 Grip Part 517
44.3.3 Release Part....519
44.3.4 Part Present....521
44.3.5 Check No Part 522
44.3.6 Prepare to Pick 522
44.3.7 Clear to Proceed 523
44.3.8 Turn ON Vacuum....524
44.3.9 Turn OFF Vacuum 525
44.3.10 Turn OFF Blowoff 526
44.3.11 Set CurrentValve....527
44.3.12 Set MH Tool....528
44.4 EXAMPLE PROGRAM 528
44.5 PRODUCTION 529
44.5.1 MH Tryout Mode 529
44.5.2 Disable for 20 Cycles 529
44.5.3 Alarms 529

45 SPECIAL JOG SEQUENCE 530
45.1 USAGE....530
45.2 SETUP....530
45.3 APPLICATION....530

46 MOTION INSTRUCTION ENHANCED EDITING 532
46.1 MOTION INSTRUCTION INSERT AND INSTRUCTION DELETE FUNCTION 532
46.1.1 Usage....532
46.1.2 Precautions 533
46.2 DATA OFFSET FUNCTION 534
46.2.1 Feed Rate Conversion 534

46.2.2 Position Data Conversion (direct method) 537
46.2.3 Position Data Conversion (2 point teach method) 540
46.2.4 Precautions 542

47 FINISHING FUNCTION PACKAGE 543

48 AUTO BACKWARD EXIT 544
48.1 Specification Overview 545
48.2 Setup 545
48.3 Programming 548

49 EXPANDED REGISTERS FUNCTION 554
49.1 SETTING THE NUMBER OF REGISTERS 554
49.2 SAVING AND LOADING FILES 555
49.2.1 LOADING A NOT EXPANDED xxxREG.VR FILE 555
49.2.2 LOADING AN EXPANDED xxxREG.VR FILE 556

50 STITCH FUNCTION 558
50.1 SPECIFICATIONS 558
50.1.1 Instructions 558
50.1.2 Stitch Schedules 558
50.1.3 Stitch Process Flow 560
50.1.4 Other Specifications 561
50.2 ADJUSTMENTS 562
50.3 SYSTEM CONFIG MENU 563
50.4 LIMITATIONS 564
50.5 STITCH FUNCTION SCREEN STRUCTURE 565

51 VISUAL DIAGNOSTICS 566
51.1 CREATING AND EDITING VISUAL DIAGNOSTIC SCREENS 567
51.2 MANAGING SCREENS 574
51.3 VIEWING VISUAL DIAGNOSTIC SCREENS 575

52 PDF VIEWER FUNCTION 578
52.1 OPENING A PDF DOCUMENT 578
52.2 NAVIGATING WITH THE PDF VIEWER 579

53 HELP AND DIAGNOSTICS DISPLAY 580
53.1 ONLINE HELP FUNCTION 580
53.2 ALARM CAUSE/REMEDY DISPLAY FUNCTION 581
53.3 HELP/DIAGNOSTICS SCREEN 583
53.4 HELP/DIAGNOSTICS MENU 584

54 MAINTENANCE REMINDER 586
54.1 INTELLIGENT CHANGE REMINDER 586
54.1.1 Intelligent Grease Change Reminder 586
54.1.2 Intelligent Cable Change Reminder (option: S511) 586
54.2 MAIN MENU 586
54.3 MAINTENANCE REMINDER SETUP 587
54.3.1 Common Settings 587
54.3.2 Maintenance Item Settings 589

54.3.3 Ambient Temperature Setting 591
54.4 CHECK MAINTENANCE TIME AND COMPLETE MAINTENANCE 591
54.4.1 Checking the Maintenance Time 592
54.4.2 Maintenance Reminder Notification 592
54.4.3 Completing Maintenance 593
54.5 ZDT 594
54.5.1 Maintenance Reminder setup 594
54.6 MAINTENANCE RECORDS 595
54.6.1 Displaying Maintenance Record 595
54.6.2 Maintenance Record File 596
54.7 ROBOT SETTINGS 597
54.7.1 i-grease Change 597
54.7.2 i-cable Change 598

55 HMI DEVICE COMMUNICATION 599
55.1 CONNECTION AN HMI DEVICE 600
55.1.1 Modbus RS-232-C Connection 600
55.1.2 Modbus/TCP Slave Ethernet Connection 602
55.1.3 OPC UA Server Ethernet Connection 603
55.2 MODBUS DATA MODEL 604
55.2.1 Modbus data model 604
55.2.2 OPC UA Organization of the Modbus data model 604
55.2.3 Correspondence of Modbus Address to Robot Data 605
55.2.4 Modbus Function Codes 606
55.3 ASSIGNMENT OF HOLDING REGISTERS 607
55.3.1 Data type of Holding Registers 608
55.3.2 Assigning Robot Registers 609
55.3.3 Assigning Position Registers 611
55.3.4 Assigning String Registers 615
55.3.5 Assign the Current Position 615
55.3.6 Assigning Alarm History 617
55.3.7 Assigning Program Execution Status 619
55.3.8 Assigning Data ID 620
55.3.9 Assigning comments 624
55.3.10 Assigning I/O data and simulation status 625
55.3.11 Assigning Integrated PMC address data 626
55.3.12 Assigning Symbols and Comments for Integrated PMC addresses 627
55.3.13 Hints 628
55.4 DYNAMIC ASSIGNMENT TO HOLDING REGISTERS 629
55.5 OPC UA DATA MODEL 630
55.5.1 RobotInformation 631
55.5.2 MotionDeviceSystem 632
55.5.2.1 MotionDevices 633
55.5.2.2 Controllers 637
55.5.2.3 SafetyStates 640

56 FAULT & INCIDENT REPORT 641
56.1 ALARMS REPORTING SCREEN 641
56.1.1 Incident screen 641
56.1.2 Total Time (T_TIME) screen 641
56.1.3 Longest Time (L_TIME) screen 642

57 PROCESS AXES 643

57.1 BASIC PROCESS AXES (R689) 643
57.1.1 Setup 644
57.1.2 TP Instructions 646
57.1.3 PROCESS AXES STATUS 648
57.1.4 Torque Monitor Function 649

58 SPLINE MOTION INSTRUCTION 651
58.1 NORMAL MOTION 651
58.2 TERMINATION TYPE OF SPLINE MOTION 654
58.3 SINGLE STEP EXECUTION 654
58.4 BACKWARD EXECUTION 654
58.5 EFFECT OF A SINGLE TAUGHT POINT ON THE PATH 655
58.6 WHEN SPLINE MOTION IS NOT POSSIBLE 655
58.6.1 Insufficient Number of Spline Motion Instructions 655
58.6.2 Teaching the Same Position 656
58.6.3 Path has a High Curvature 656
58.7 RESUMING AFTER PAUSING 657
58.7.1 Resuming after Jogging 657
58.7.2 Pausing and Resuming from Another Spline Motion Instruction 657
58.7.3 Resuming After a Pause During Backward Execution 658
58.7.4 Starting a Program from a Spline Motion Instruction 658
58.8 LOGIC INSTRUCTIONS BETWEEN SPLINE MOTION INSTRUCTIONS 658
58.9 AVAILABLE MOTION MODIFIERS 660
58.10 CHANGING POSITION DATA DURING S MOTION EXECUTION 661
58.11 RESUMING AFTER MODIFYING THE PROGRAM 662
58.12 RESTRICTIONS 663

59 HIGH SPEED POSITION OUTPUT 664
59.1 DATA FORMATS 664
59.1.1 Binary 664
59.1.1.1 TCP Cartesian Position with Binary 665
59.1.1.2 Joint Angles with Binary 665
59.1.1.3 Variables with Binary 665
59.1.2 XML 666
59.1.2.1 TCP Cartesian Position with XML 666
59.1.2.2 Joint Angles with XML 666
59.1.2.3 Variables with XML 666
59.2 SETUP 666
59.2.1 Communication Setup 667
59.2.2 Position Data Packet Settings 667
59.2.3 Variable Data Packet Settings 668
59.2.4 Start Output 668

60 SOFTWARE PLC 669
60.1 Overview of Software PLC 669
60.2 Preparation 674
60.3 I/O assignment for The Software PLC 675
60.3.1 Software PLC Input/Output area 675
60.3.2 Software PLC External I/O Assignment 676
60.3.2.1 Relationship with I/O Assignment (Robot) 678
60.3.2.2 Example of Setting the Software PLC External I/O Assignment and Mapping 680
60.3.3 Software PLC internal I/O Assignment 682

60.3.3.1 Example of Software PLC Internal I/O Assignment Configuration 683
60.4 How to use the CODESYS Development System 686
60.4.1 Creating a Project for the Software PLC 686
60.4.2 PLC programming and Task assignment 687
60.4.3 Addition of I/O devices for the FANUC robot and the I/O mapping..... 691
60.4.4 Login/Logout to the robot control device, download and start/stop the PLC program. 694
60.4.5 Monitoring the PLC Program..... 697
60.4.6 Verification of the PLC program 697
60.5 Saving and Loading Software PLC related data..... 699
61 DIGITAL I/O JOG FUNCTION 700
61.1 OVERVIEW 700
61.2 SPECIFICATION 700
61.2.1 Setup..... 700
61.2.2 Limitations 701
62 PALLETIZING MODE 702
62.1 Overview..... 702
62.2 TP program execution 702
62.2.1 Enable/Disable setting..... 702
62.2.2 Cartesian motion (Linear, Circle, Arc, and Spline) 704
62.2.3 Joint motion..... 704
62.2.4 Restrictions for TP program execution 704
62.3 Jogging Operation 705
62.3.1 Enable/Disable setting..... 705
62.3.2 Cartesian jogging (WORLD, TOOL, USER, and JGFRM)..... 706
62.3.3 Joint jogging..... 706
62.4 Manual Guided Teach 707
62.4.1 Enable / disable setting..... 707
62.4.2 Translation..... 707
62.4.3 Rotation 707
62.4.4 Free..... 707
62.5 Remarks 708
62.5.1 Align function 708
63 CUSTOMIZE SUPPORT FUNCTION..... 709
63.1 KAREL CONFIG..... 709
63.1.1 KAREL Config Screen..... 709
63.1.2 Use KAREL Config Screen 709
63.1.3 Run KAREL Program 710
63.1.4 Abort KAREL Program..... 711
63.1.5 Start Mode Config of KAREL program..... 712
63.1.6 Detail Screen of KAREL Config 713
63.1.7 Limitation and Caution of KAREL Config..... 714
63.1.8 Cycle Power 714
63.2 CUSTOM MENU..... 715
63.2.1 Overview 715
63.2.2 Starting Custom Menu 715
63.2.3 Set Custom Menu 715
63.2.4 Delete Set 717
64 PALLETIZING FUNCTION..... 719

64.1 PALLETIZING FUNCTION 719
64.2 PALLETIZING INSTRUCTIONS 721
64.3 TEACHING THE PALLETIZING FUNCTION 723
64.3.1 Selecting a Palletizing Instruction 723
64.3.2 Inputting Initial Data 725
64.3.3 Teaching a Stacking Pattern 732
64.3.4 Setting Path Pattern Conditions 738
64.3.5 Teaching a Path Pattern 741
64.3.6 Notes on Teaching the Palletizing Function 744
64.4 EXECUTING THE PALLETIZING FUNCTION 745
64.4.1 Palletizing Register 746
64.4.2 Controlling the Palletizing Function by a Palletizing Register 748
64.5 MODIFYING THE PALLETIZING FUNCTION 749
64.6 PALLETIZING FUNCTION WITH EXTENDED AXES 751
64.7 PALLETIZING ALL-POINT TEACHING 751

1 OVERVIEW

This manual describes the following software options.

Chapter 2 *i*RCALIBRATION SIGNATURE
Chapter 3 SOFTFLOAT FUNCTION
Chapter 4 CONTINUOUS ROTATION FUNCTION
Chapter 5 OPERATION GROUP DO OUTPUT FUNCTION
Chapter 6 AUTOMATIC ERROR RECOVERY FUNCTION
Chapter 7 REMOTE TCP FUNCTION
Chapter 8 HIGH SENSITIVITY COLLISION DETECTION
Chapter 9 LOAD ESTIMATION
Chapter 10 PAYLOAD OVER AND PAYLOAD SETTING CONFIRM FUNCTION
Chapter 11 GRAVITY COMPENSATION
Chapter 12 OPERATION LOG BOOK
Chapter 13 PROGRAM TOOLBOX
Chapter 14 ADVANCED CONSTANT PATH
Chapter 15 SINGULARITY AVOIDANCE FUNCTION
Chapter 16 PATH SWITCHING FUNCTION
Chapter 17 AUXILIATY AXIS SERVO OFF (LOCAL STOP) FUNCTION
Chapter 18 DUAL MOTOR DRIVE
Chapter 19 MULTI UOP INTERFACE FUNCTION
Chapter 20 ERROR CODE OUTPUT FUNCTION
Chapter 21 DATA MONITOR FUNCTION
Chapter 22 BRAKE CHECK FUNCTION
Chapter 23 NC PROGRAM CONVERSION
Chapter 24 ENHANCED MIRROR IMAGE
Chapter 25 VIA INSTRUCTION
Chapter 26 LEARNING PATH CONTROL
Chapter 27 SERVO TOOL FUNCTION
Chapter 28 TCP SPEED OUTPUT
Chapter 29 TP DRAM/FILE STORAGE FUNCTION
Chapter 30 CYCLE TIME LOGGING
Chapter 31 MATH FUNCTION INSTRUCTION
Chapter 32 SERVO TOOL CHANGE FUNCTION
Chapter 33 OPERATION WITHOUT SHIFT FUNCTION
Chapter 34 HIGH SPEED SHIFT KEY FUNCTION
Chapter 35 *i*RDIAGNOSTICS
Chapter 36 MENU UTILITY
Chapter 37 4D GRAPHICS FUNCTION
Chapter 38 DATA TRANSFER BETWEEN ROBOTS FUNCTION
Chapter 39 TOUCH SENSING
Chapter 40 TOUCH SKIP FUNCTION
Chapter 41 MROT INSTRUCTION
Chapter 42 SMALL CIRCLE ACCURACY IMPROVEMENT
Chapter 43 SHAPE GENERATION FUNCTION
Chapter 44 MATERIAL HANDLING GRIPPER OPTION
Chapter 45 SPECIAL JOG SEQUENCE
Chapter 46 MOTION INSTRUCTION ENHANCED EDITING
Chapter 47 FINISHING FUNCTION PACKAGE
Chapter 48 AUTO BACKWARD EXIT
Chapter 49 EXPANDED REGISTERS FUNCTION

Chapter 50 STITCH FUNCTION
Chapter 51 VISUAL DIAGNOSTICS
Chapter 52 PDF VIEWER FUNCTION
Chapter 53 HELP AND DIAGNOSTICS DISPLAY
Chapter 54 MAINTENANCE REMINDER
Chapter 55 HMI DEVICE COMMUNICATION
Chapter 56 FAULT & INCIDENT REPORT
Chapter 57 PROCESS AXES
Chapter 58 SPLINE MOTION INSTRUCTION
Chapter 59 HIGH SPEED POSITION OUTPUT
Chapter 60 SOFTWARE PLC
Chapter 61 DIGITAL I/O JOG FUNCTION
Chapter 62 PALLETIZING MODE
Chapter 63 CUSTOMIZE SUPPORT FUNCTION
Chapter 64 PALLETIZING FUNCTION

2 *i*RCalibration Signature

2.1 Overview

Product Concept

With the latest machining technologies and the advancements in product design and analysis tools, it is still not possible to eliminate deflection of a robot arm nor to assemble mechanical parts without error. Therefore, to achieve absolute robot accuracy, software is required to compute and compensate individual robot's inherited errors, and the ***i***RCalibration Signature (A05B-2600-R726) is the software that can improve absolute accuracy of a robot.

High Accuracy Factory Calibration

For all robots ordered with the ***i***RCalibration Signature software, FANUC performs High Accuracy Factory Calibration as part of our manufacturing process. The signature calibration measures hundreds of robot postures in whole robot working envelope and computes the coefficients that compensate for errors of the robot. The result is a robot that is capable of achieving a high level of accuracy for the whole working space that cannot be obtained by robots without calibration. For the robot that went thought the calibration process, a label like the one shown in Fig2.1(a) is affixed, and accuracy evaluation data is delivered as an attachment. Note that the High Accuracy Factory Calibration is model specific, please contact FANUC for a list of supported robot models.

Once a robot is signature calibrated, the robot runs in signature compensated mode for all types of motion with very minor restrictions (see section 2.8.2). In signature compensated mode, all teach pendant program motion instructions benefit from the high accuracy compensation built in FANUC's motion system and there is no special treatment required for these TP programs to perform high-accuracy motions. User simply use the robot like a standard robot and ***i***RCalibration signature software ensures the robot reaches its taught position as accurately as possible.



Fig 2.1 (a) Label for high accuracy calibration

Custom Signature Calibration

To maintain the best accuracy of a robot, it is recommended to check its accuracy periodically during the life of the robot and to perform calibration of the robot if required. In addition, if a robot went through reducer/motion replacement, the users would have to recalibrate the robot to get the robot back to its known accuracy. In either case, FANUC user can use the custom signature calibration function provided by the ***i***RCalibration signature software to calibrate the robot at their own facility to get the robot back into production with reduced down time.

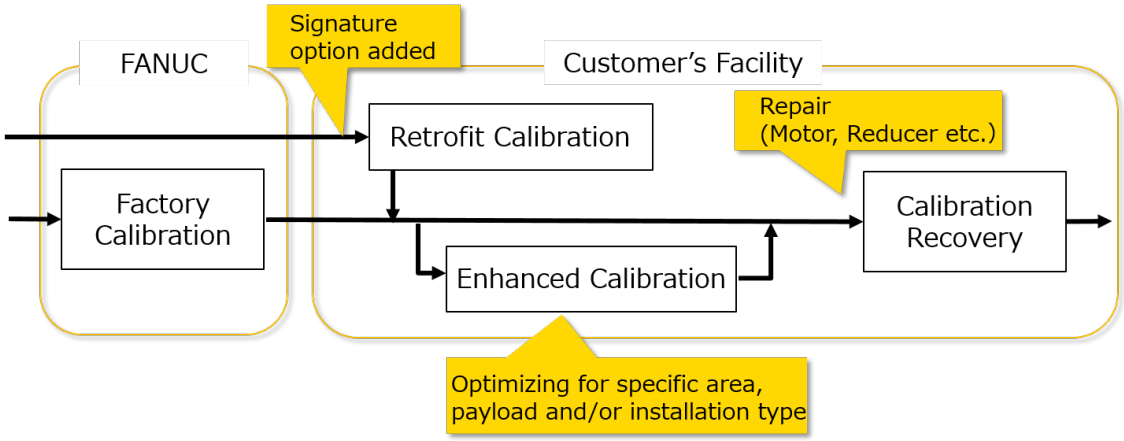


Fig 2.1 (b) Overview of custom calibration function

A customer signature calibration will require a laser tracker and a laser beam reflector as the target for the laser tracker. The reflector is mounted on the tool of the robot, and it should be fixed during the calibration process. Like factory calibration, this calibration requires measuring the reflector at various robot postures. Different calibration types have different requirement for number of measurements needed. Once the measurement is completed and the measured data is loaded to the controller. The *i*RCalibration signature software computes the new compensation parameters and notifies you the calibration result. If the result is acceptable, the new compensation parameters, will be loaded into the controller and the robot is back in production.

NOTE
1 A laser tracker and measurement software is required to perform customer calibration. In addition, the operator should have training in metrology and know how to setup measurement frames. Please contact FANUC if help is needed.
2 Please refer to the manual for the proper procedures when performing Custom Calibration. Please contact FANUC if this is your first time performing Custom Calibration.

Signature Compensation

Once signature calibration parameters are applied to the robot controller, the software compensates motor position in real time to achieve better accuracy regardless of motion type, position representation or motion modifier. Therefore, the signature calibrated robot can be used in a wide range of applications.

NOTE
Like gravity compensation, *i*RCalibration signature uses the payload information to compute the bending of the robot arm due to the gravity loading. Please set the accurate payload to take full benefits of the signature compensation function. Please refer to the OPERATOR'S MANUAL (Basic Function) or the MECHANICAL UNIT OPERATOR'S MANUAL for payload setting.

2.2 Benefits of Signature Calibrated Robot

Simplifying offline programming

Signature calibrated robot makes Teach Pendant (TP) programs generated offline much easier to implement in the real work cell. The following example shows the required steps to utilize an offline generated TP program on the factory floor. (A) Measure robot installation position and jig location using

a laser tracker and (B) Shift the offline created TP program based on the measured value in the simulation. (C) Download the shifted TP program to the controller. (D) Verify the TP program and make touch up when necessary.

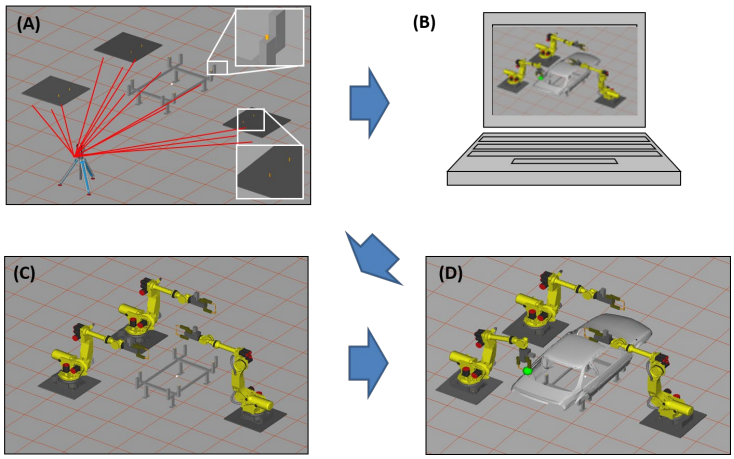


Fig 2.2(a) Example of robot system start-up

Easy Robot Replacement

Most FANUC's robot models have locating pin holes in its base as shown in figure below. By using these locating pin holes, a new robot can be installed at the same location as an old robot. These pin holes are also used in our Factory calibration as reference. Therefore, a newly signature calibrated robot can be installed in the same location as the previously calibrated robot, resulting in minimum touchup. This "drop-in replacement" can reduce down time of a production line significantly in case a robot is damage during production.

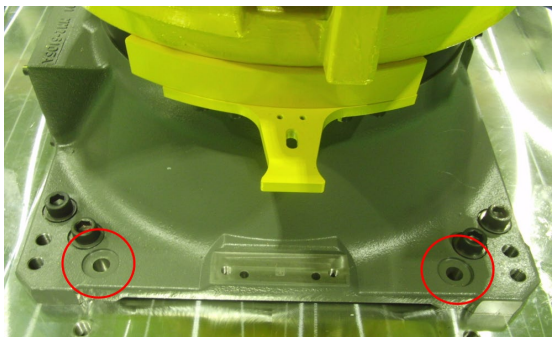


Fig 2.2(b) Locating pin holes on J1 base

2.3 Custom Signature Calibration

2.3.1 Calibration System

To perform a signature calibration, the following hardware/software are required:

- A Laser Tracker
- A Measurement reflector
- A PC for measurement software
- A Robot
- A Robot controller loaded with *i*RCalibration Signature software
- A calibration program contains postures that you want to measure

After running through the calibration program and measuring a variety of robot postures, a measurement data file from the PC is loaded to robot controller via memory device (USB etc.). The robot controller

then computes signature calibration parameters using loaded 3D measurement data to minimize robot errors. Note that the laser tracker, measurement reflector, PC and measurement PC software are 3rd party products that you should obtain before conducting the calibration.

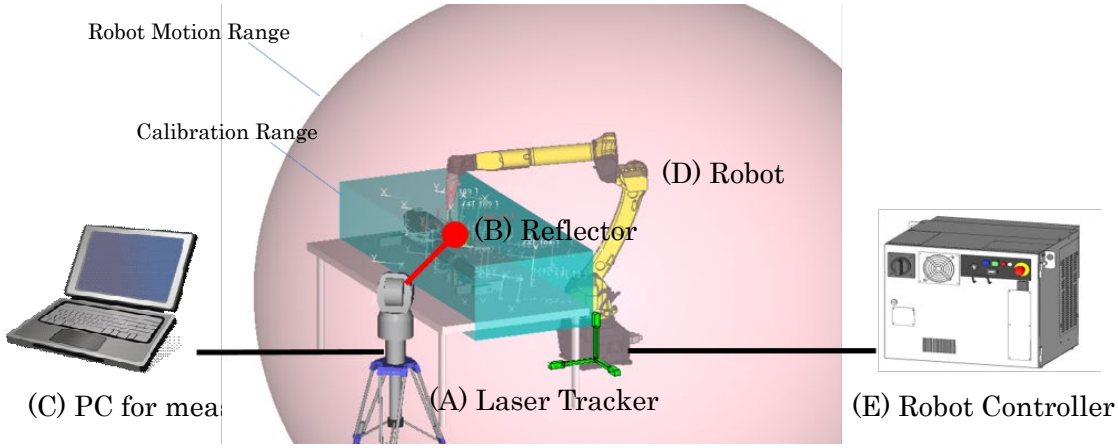


Fig 2.3.1(a) Calibration System

2.3.2 Calibration Type

In addition to the High Accuracy Factory Calibration, the Custom Signature Calibration provides 4 calibration types listed below to maintain absolute accuracy during the life cycle of the robot.

Retrofit Calibration

Retrofit calibration is a function to perform the signature calibration for robots that have been previously installed and were not Factory calibrated before installed on customer's facility. This function computes optimized signature calibration data using limited space after installation.

Enhanced Factory Calibration (Enhanced Calibration)

Enhanced Factory Calibration is a function to perform signature calibration in addition to the existing factory calibration. This function computes optimized signature data in actual installation condition (upside-down, angle mount etc.) by measuring additional robot positions to improve accuracy for specific robot configuration.

Calibration Recovery

When a signature calibrated robot needs a repair, such as replacing a reducer or a motor, the Calibration Recovery function can get the robot back to similar accuracy as before.

TCP/Uframe Calibration

TCP/Uframe Calibration computes robot installation position and measurement reflector position mounted on the robot's tool.

NOTE
1 Custom Calibration should be performed refer to the manual for the proper procedures to ensure an acceptable result.
2 The "High Accuracy Calibrated" label shown in Fig2.1(a) can be ordered with A370-8037-0256. Please use this label for robots calibrated by Retrofit Calibration or Enhanced Calibration to distinguish them from other robots.

2.3.3 Custom Signature Calibration Preparation

Figure in below shows preparation work before custom signature calibration.

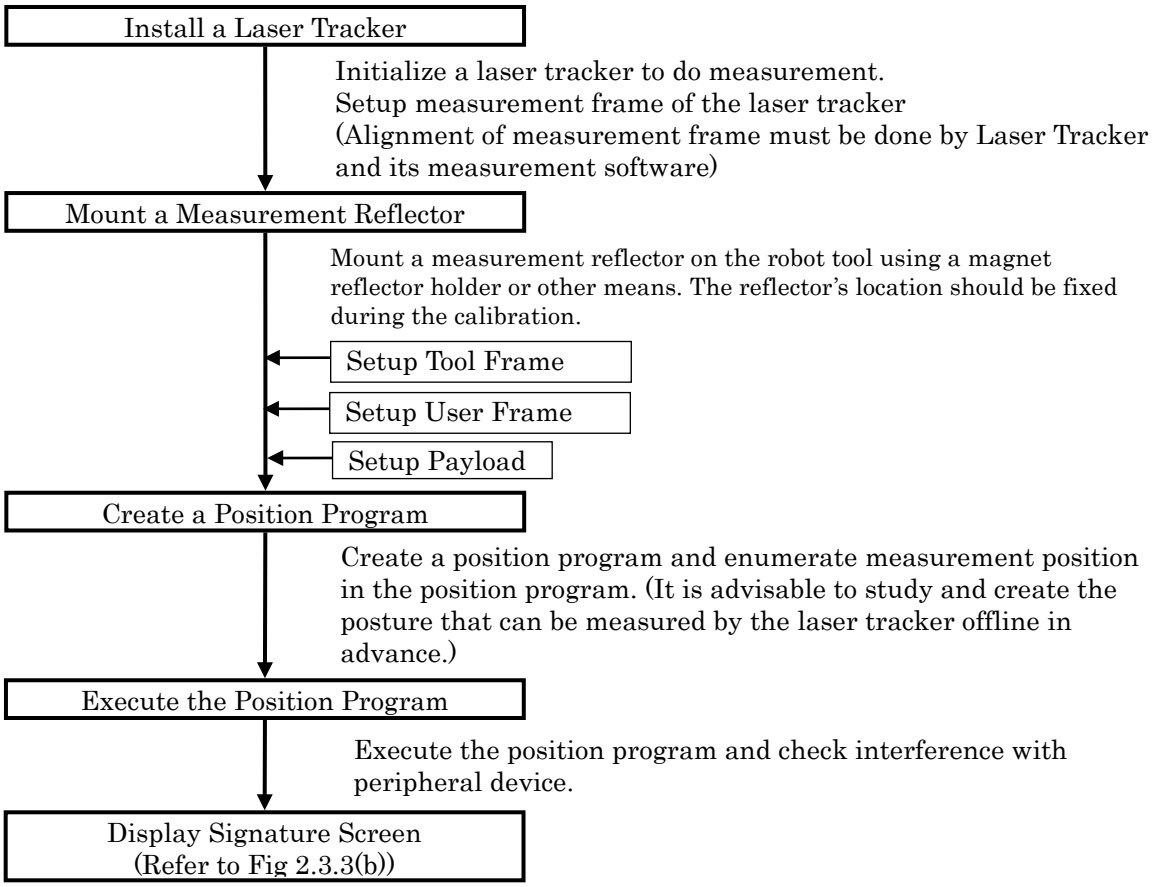


Fig. 2.3.3(a) Outline of custom signature calibration preparation

NOTE
1 The robot needs to be mounted on a solid surface that will not bend/deflect during calibration.
2 The tooling mounted on the robot must be rigid and the measurement reflector need to be fixed on the tooling.

Please make sure the robot will not collide with its environment during the calibration and the laser tracker can see the measurement reflector in all robot measurement positions. If everything checked out, please proceed to the following procedure:

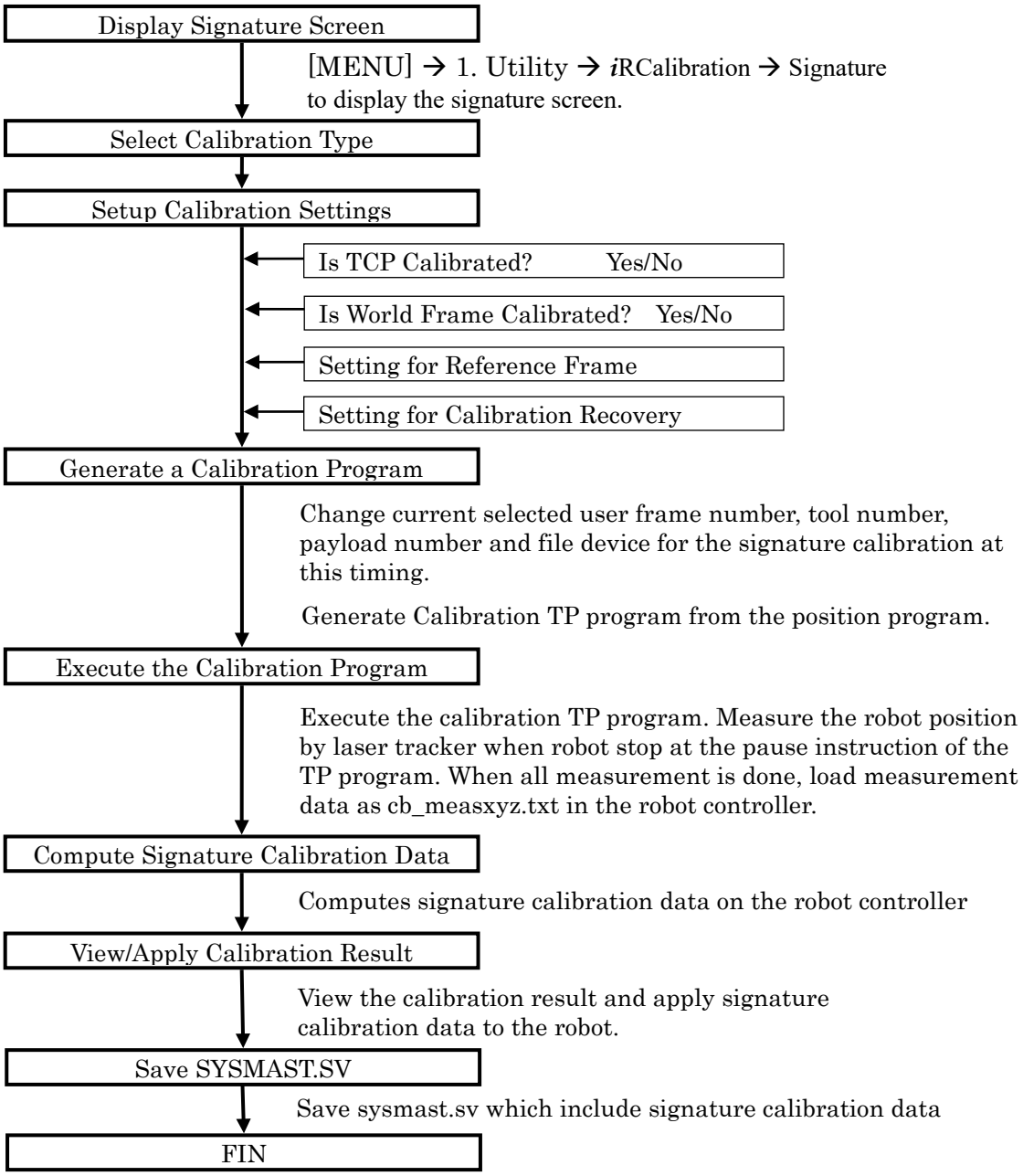
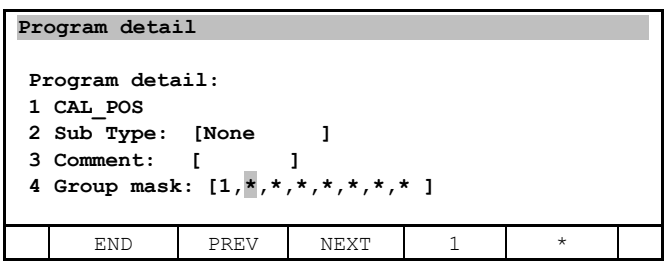


Fig 2.3.3(b) Operation flow of Signature screen

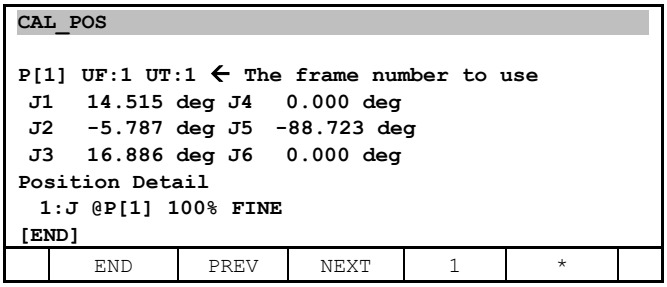
2.3.4 Position Program

Please create a position program with enumerating measurement posture for the calibration as follows.

a. Set 1 for group mask for the robot group which is going to be signature calibrated.



b. Choose a set of tool and user frame numbers which are used for signature calibration. Set current user frame and tool frame number to them and make sure the program's position data are set to the same user and tool frame number.



c. All program's positioning termination type should be FINE.
d. The First position of position program is not measured. Set home position for the first position.
e. Increase number of measurement points using position which is used in actual application or position during its path.

1: J P[100] 20% FINE ← home position
2: J P [1] 20% FINE
3: J P [2] 20% FINE
…
40: J P [39] 20% FINE
41: J P [40] 20% FINE

f. Tool orientation shouldn't be constant. The TP program should include various postures.
g. When the TP program has limited range of motion, accuracy could not be improved.
h. Please refer to the table below for the required number of measurement points for each calibration type.

The required number of measurement point

Calibration Type	Description
Retrofit Calibration	Minimum: 20, Recommended: over 100 Better to include various posture in the area robot can move without interference
Enhanced Factory Calibration	Minimum: 20, Recommended: 40-100 Better to include actual taught point where accuracy improvement is required.
Calibration Recovery	Minimum: 20, Recommended: 40-100 Better to include various postures from area robot can move without interference. Better to move axis related to repaired/replaced parts in its full range of motion if possible.
TCP/Uframe Calibration	Minimum: 12, Recommended: about 20 Better to include actual taught point where accuracy improvement is expected.

As shown in figure below, please create position program combining various posture (left) and actual taught points (right).

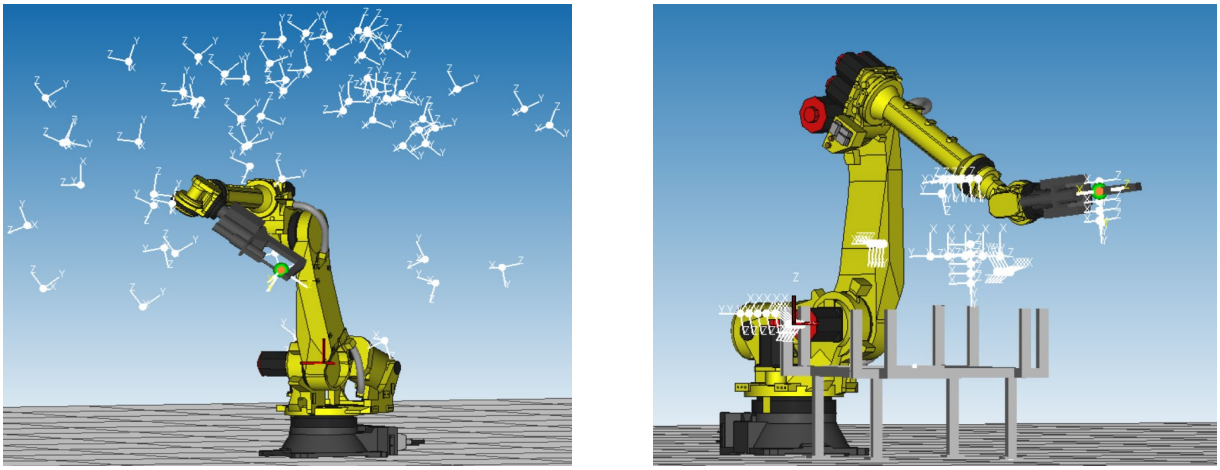


Fig 2.3.4(a) Example of position program

2.3.5 Coordinate Systems

Coordinate Systems in Signature Calibration

AS shown in below, this function uses (L) Laser tracker frame, (W) World frame, (M) Measurement frame, (F) Mechanical Interface Frame, (T) Tool frame.

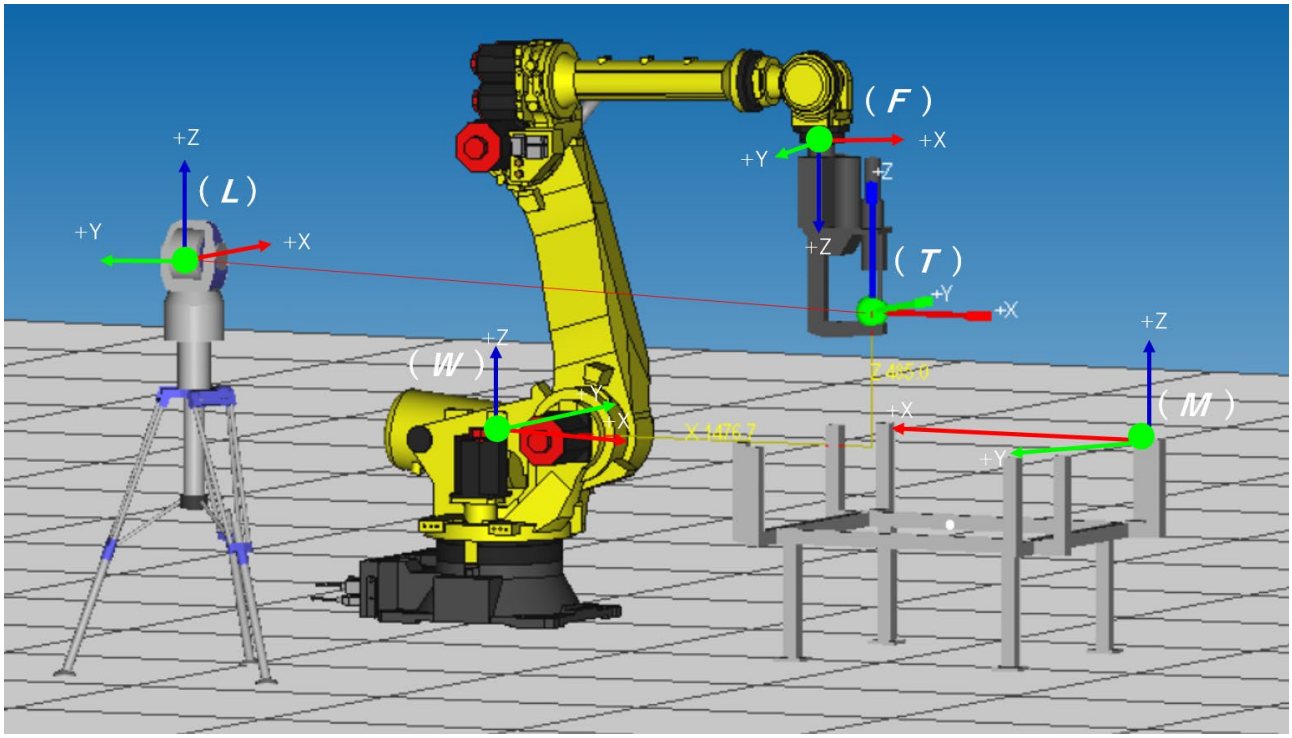


Fig 2.3.5(a) Coordinate systems in the signature calibration

Laser Tracker Frame (*L*)

- Laster tracker frame is a coordinate system integrated in the laser tracker itself. Laser tracker frame therefore can be used as measurement frame.

World Frame (*W*)

- World frame is a standard Cartesian coordinate system fixed in a work space. The coordinate system is fixed at a position determined by the robot.

Measurement Frame (*M*)

- Measurement PC software in general is capable of setting up the measurement frame using its alignment function.
- It's possible to set the measurement frame consistently by using multiple fixed measurement points with known design values in the production line. Measurement frame is thus reproducible by measuring those points when setting up the laser tracker.
- Set the position and orientation (x,y,z,w,p,r) of the measurement frame (with respect to the robot's world frame) into the user frame for signature calibration.

Mechanical Interface frame (*F*)

- Mechanical Interface Frame is a standard Cartesian coordinate system defined for the mechanical interface of the robot (the surface of wrist flange). When setting up a tool frame, the tool frame is referencing to the mechanical interface frame.

Tool Frame (*T*)

- Tool Frame defines the position of the tool center point (TCP) seen from the mechanical interface frame.
- Set the position (x,y,z) of measurement reflector into the tool frame used for signature calibration.

Setup of Measurement Frame and its Features

Features in below can be obtained by setting measurement frame to fit world frame or user frame.

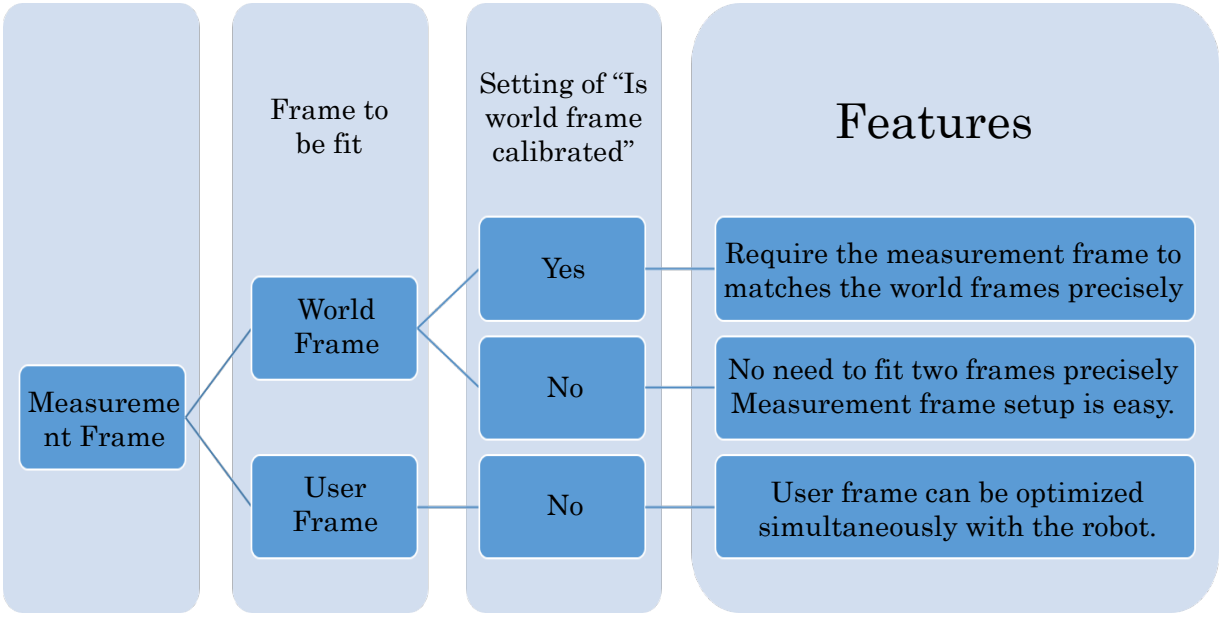


Fig. 2.3.5(b) Setup of measurement frame and its features

World Frame Calibration

- When the measurement frame is aligned with the world frame precisely, such status is defined as the world frame calibration done.
- If the world frame calibration is done, set (x,y,z,w,p,r) = (0,0,0,0,0,0) into the user frame used for signature calibration. In case of Not done, set the position and orientation (x,y,z,w,p,r) of the measurement frame as seen from the robot's world frame into the user frame used for signature calibration.
- In the setup of signature calibration, when "Is world frame calibrated" is set to Yes, signature calibration parameters are computed assuming the measurement frame matches the robot's world

frame precisely. When “Is world frame calibrated” is set to No, the measurement frame seen from the robot’s world frame is computed simultaneously with the robot’s signature calibration parameters and is applied to the user frame.

- In the retrofit calibration and enhanced factory calibration, it is recommended to set the world frame calibration to Not done. In calibration recovery, it is recommended to set the world frame calibration to Done because the world frame calibration must be done when recovering parameter related to J1.

Calibration of User Frame

- When a user frame is defined on the fixture/jig etc. of the production line and used in TP program, the software can calibrate the user frame as part of signature calibration.
- In this case, the answer to the “Is world frame calibrated?” question in the signature setup menu is No.
- When “Is world frame calibrated?” is set to No, the actual user frame position (with respect to the robot’s world frame) is computed.
- If the user frame is calibrated as part of signature calibration, there is no need to adjust the offline TP program’s user frame to correct actual jig positions.

Tool Frame Setup and Calibration

- The Tool frame for signature calibration is defined as the measurement reflector location with respect to the robot’s mechanical interface frame.
- If the measurement reflector location is known exactly, select “YES” to the question “Is TCP Calibrated?” in the signature setup screen. Please go to Menu SETUP/Frame screen to enter the known (x, y, z) value for the designated utool. If the measurement location is unknown, answer “NO” for the question “Is TCP Calibrated?”. Please go to the Menu Setup/Frame to enter an estimated (x, y, z) value so the actual location is within a 10 mm radius of the estimated location for the designated utool.
- When “Is TCP Calibrated?” is set to YES, signature calibration assumes the measurement reflector’s position is known and will not compute an offset to the tool frame. If the “Is TCP Calibrated” is set to NO, signature calibration will compute the offset to the tool frame as part of the signature computation.
- To achieve a good accuracy of the robot, it is recommended to change tool posture as much as possible when calibrating. However, when “Is TCP Calibrated?” is set to YES, it is possible to achieve good accuracy even if the posture change is small. For example, if it is difficult to change the tool posture due to limitations in the operating range, it is recommended to use a tool that the location of the reflector can be accurately set in the tool frame. So that the calibration can be performed even if the posture change is small.

2.4 Custom Signature Calibration Procedure

Following the outline introduced in 2.3.3, this section introduces the custom calibration procedure. A similar procedure can be used for all calibration types. Steps 2-1 to 2-8 show an example of performing retrofit calibration using the configuration in Section 2.4.1.

2.4.1 Custom Signature Calibration Preparation

Install a Laser Tracker

As shown in Figure 2.4.1, place a laser tracker near the robot and make it ready for measurement.

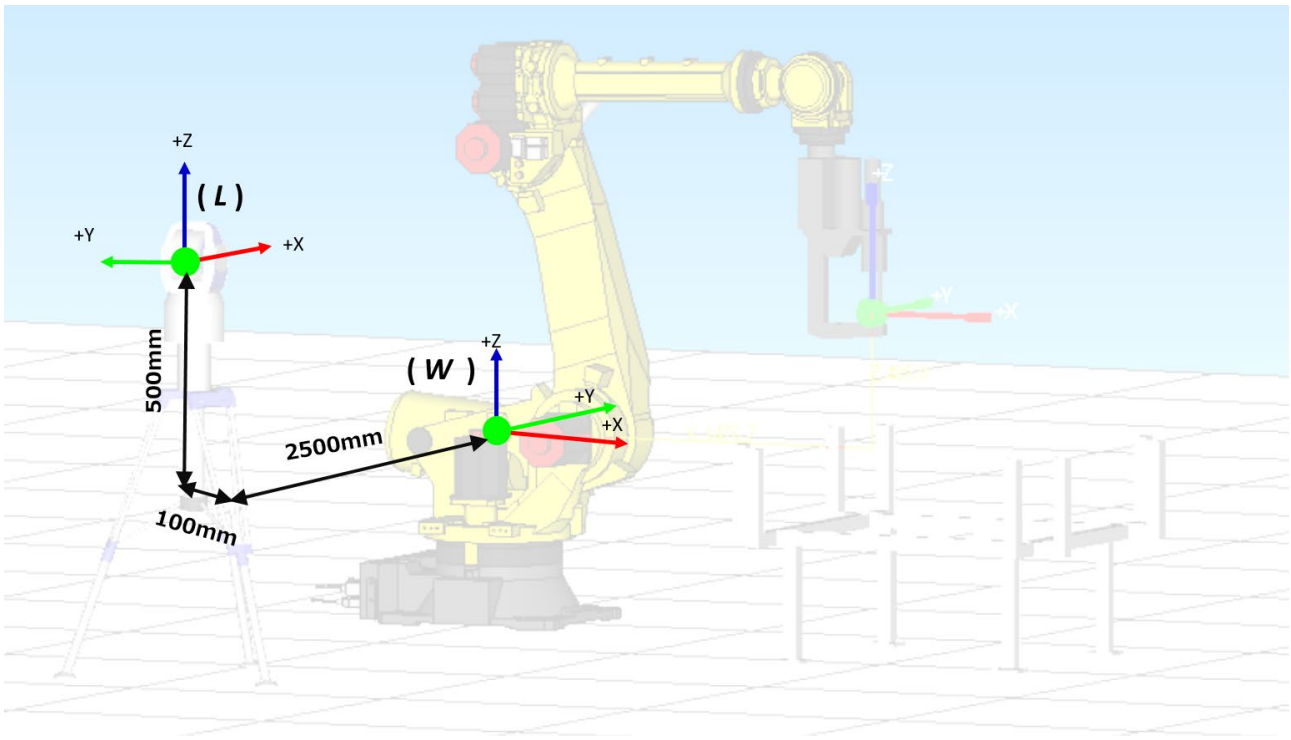


Fig 2.4.1 Install a Laser Tracker

Setup User Frame

Set the position and orientation (x, y, z, w, p, r) to the user frame used in Custom Calibration, which is measurement frame seen from the robot's world frame. The example in the figure above uses the laser tracker frame as the measurement frame. As shown in Figure 2.4.1, place a laser tracker near the robot and make it ready for measurement. First, use a tape measure to measure the distance from the robot's world frame to the origin of the laser tracker frame. In this example, the approximate distances are x = -100mm, y = -2500mm, z = 500mm. Next, determine the orientation of the measurement frame from the orientation of the laser tracker. In this example, the measurement frame is a posture rotated 90 degrees around the Z axis of the world frame, so w = 0deg, p = 0deg, r = 90deg. Therefore, set (-100, -2500, 500, 0, 0, 90) as the user frame to use.

Note

The calibration may not be performed correctly if the setup in user frame differs greatly from the actual measurement frame. Please compare position displayed on the current position screen with the position measured by laser tracker, and correct the user frame so that the difference is within about 100mm.

Mount a Measurement Reflector

Mount a measurement reflector on the robot tool.

Mount the reflector to any location using a magnet reflector holder, etc. as shown in Figure 2.4.2. Please make sure that the reflector holder does not move during the calibration.

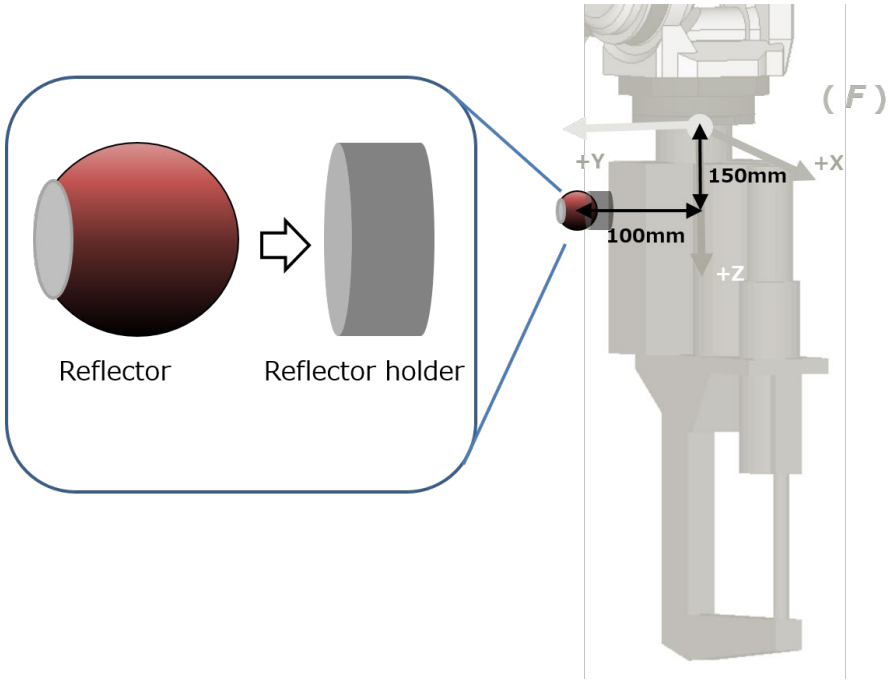


Fig 2.4.2 Mount a measurement reflector

Please arrange the reflector and reflector holder from the laser tracker manufacturer.

Setup Tool Frame

Set the mounted position (x, y, z) of the measurement reflector to the tool frame used in Custom Calibration.
In this example, we will use a tape measure to the mounted position of the measurement reflector from the mechanical interface frame. Set the tool frame to be used as a rough position (0, 100, 150) in units of about 10 mm.

Setup Payload

Please set the payload information of the tool correctly.

> **Note**
> It is recommended to actually measure the weight and calculate the center of gravity position from the design information. Since the payload information needs to be set correctly, please be sure to enter the mass when using the load estimation function.

Create a Position Program

Create a position program referring to section 2.3.4.
Execute the position program and verify that measurement can be taken without interference.

2.4.2 Display Signature Screen

To run signature calibration, you should use the ***i***RCalibration Signature menu. This menu is common for all calibration types and each item in the menu is shown in the table below. To complete the calibration process, just complete each item in succession till you get the calibration result in the final screen.

***i*RCalibration Signature screen**

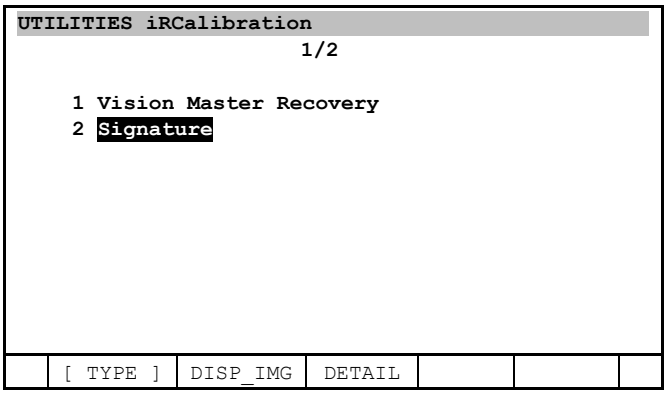
Items	Description
Group	Set the group number.

Items	Description
Calibration Type	Select the calibration type.
Setup	Setup calibration settings.
Create Calibration Prog	Generate a calibration TP program from an existing position program.
Run Calibration Program	Execute the calibration TP program to collect measurement data for computation.
Calculate Parameters	Compute the signature calibration data using the measured data.
View Calibration Result	Display calibration result and apply the computed signature calibration parameters.

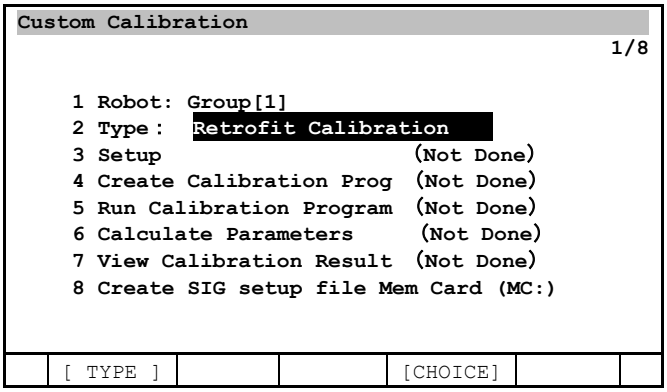
Procedure 2-1 Display signature screen

Step

1 Press the [MENU] key to display the screen menu.
2 Select “1 UTILITIES”
3 Press F1, [TYPE] to display the selection menu.
4 Select *i*RCalibration. Then the *i*RCalibration screen will be displayed.



5 Move the cursor to the Signature item and press the “enter” key. The signature screen will be displayed.



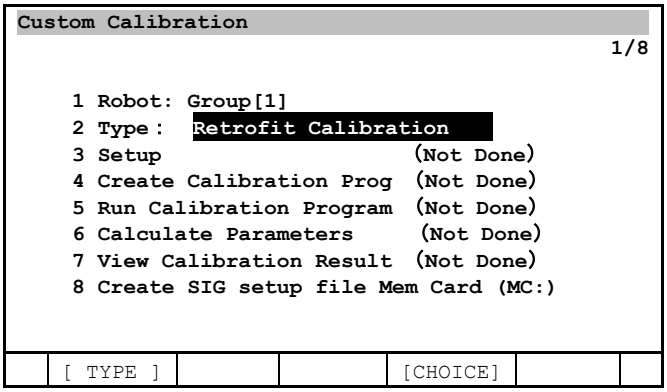
2.4.3 Select Calibration Type

On the item of Type on the signature screen, select a calibration type from 4 calibration types shown in the subsection 2.3.2.

Procedure 2-2 Change Calibration Type

Step

1 Display the signature screen. (Procedure 2-1)
2 Move the cursor to the Type item and press F4 “[CHICE]” to display a popup menu of selectable calibration type.
3 Move the cursor to the Calibration Type you want and press the “enter” key, and the following screen is displayed and calibration type is changed to the one you selected. In the example below, the “Retrofit Calibration” type is selected.



NOTE
The Enhanced Factory Calibration and the Calibration Recovery are selectable after completion of High Accuracy Factory Calibration or Retrofit Calibration.

2.4.4 Setup Calibration settings

Items on the Setup Screen

For Retrofit Calibration, Enhanced Factory Calibration or Calibration Recovery, their setup menu has the following items. The Calibration recovery has additional setup items, please refer to Subsection 2.5.1 Calibration Recovery. For the TCP/Uframe Calibration setup menu, please refer to Subsection 2.5.2.

Items	Description
Is TCP Calibrated	Select Yes if the TCP is known, otherwise, select No.
Is World Frame Calibrated	Select Yes if the laser tracker’s measurement frame is aligned to the robot’s world frame. Otherwise, select No.
Reference Frame Enabled/Disabled	Select Yes if a known reference is used for laser tracker’s measurement frame.
Reference Frame number	Set a reference user frame number.

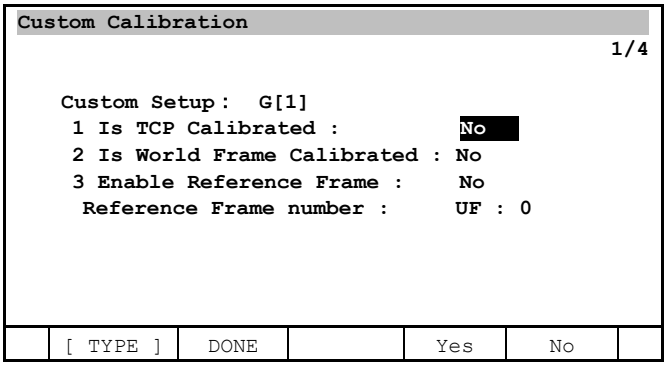
Please refer to Subsection 2.3.5 Coordinate Systems for the detail of the world frame calibration and the tool frame calibration. For the Reference Frame function, please refer to Subsection 2.5.3.

Procedure 2-3 Display setup screen

Step

1 Display the signature screen. (Procedure 2-1)
2 Switch the calibration type to Retrofit Calibration. (Procedure 2-2)
3 Move the cursor to the Setup item and press F4 “SELECT” to display a setup screen shown in below.
4 In the example introduced in Section 2.4.1, both TCP and world frame are not calibrated, so select "No" for both.

5 After setting up all items, press F2 “DONE” to complete the setup, return to the signature screen and the status of the Setup item is changed to “DONE”.
6 If a message shows up ask you to recycle power, please cycle power.



2.4.5 Create Calibration Program Screen

The “Create Calibration Program” screen allows you to generate calibration TP program for the signature calibration from the pre-prepared position program.

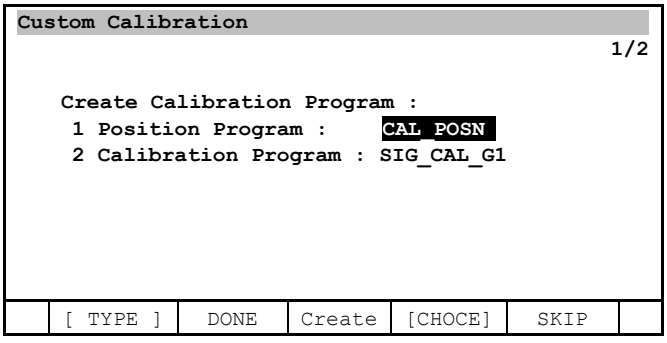
Procedure 2-4 Create Calibration TP Program

Condition

The status of the “Setup” item is “Done”.

Step

1 Display the signature screen. (Procedure 2-1)
2 If the controller is in STEP mode, press STEP key in the teach pendant to release the STEP mode.
3 Move the cursor to the Create Calibration Prog item and press the F4 “SELECT” key to display a screen shown in below. (If a program for calibration already exists, you can skip from this screen by pressing F5 “SKIP”)



4 Move the cursor to the item of position program and press F4 “[CHOICE]” to select a position program as the base for creating the signature calibration program. In the example shown above, CAL_POSN is selected.
5 Move the cursor to the item of the calibration program and press the “enter” key, and you can enter the name for the new calibration program.
6 Press F3 “Create” to create the calibration TP program. In the example shown above, SIG_CAL_G1 is created.

When setting the “Is World Frame Calibrated?” item is “No” on setup screen and user frame number used in position data of position program is 0 (world frame), the following message will be displayed

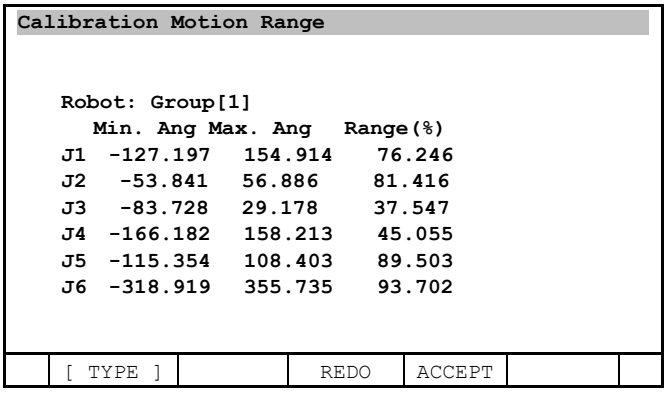
to set Uframe number for the calibration program. Select the Uframe number for recording the measurement frame seen from the robot's world frame.

1 Target Uframe Number : 1

If the number of positions is less than the minimum number of measurement point described in Subsection 2.3.4, the following message will be displayed. Select "Yes" to continue as it is, select "No" to add some more positions to the position program. If the number of positions is less than the minimum number, computing signature data will fail before starting the calculation. If you select "Yes", please make sure to add some more measurement points before running signature calibration.

Not enough motion. Continue?
Press F2 for Yes.
Press F3 for No.

7 When the calibration program is created, the minimum angle, maximum angle, and the ratio to the motion range of each axis will be displayed by pressing F2 "DONE". Select F4 "ACCEPT" to execute calibration in this motion range, select F3 "REDO" to recreate the calibration program.



8 Return to the sinecure screen by pressing F4 "ACCEPT" in above screen and the status of the Create Calibration Prog item is changed to "DONE" .

2.4.6 Run Calibration Program Screen

Run Calibration Program Screen

Please follow the descriptions shown on the screen to select calibration TP program from the program selection screen, run the program and transfer measurement data file to the robot controller.

NOTE

Before running the calibration program, please make sure the following items are set correctly:

- The measurement reflector is securely attached and its location is properly set in the tool frame.
- Correct payload number is selected and its payload information is accurate and based on the current physical payload.
- The laser tracker must be securely tied down and should not move during measurement.
- The medium for recording measurement data is inserted and the current device is correctly selected.
- If the robot has secondary encoders, make sure the secondary encoder control is enabled.

Outline of the Calibration TP Program

The example program shown below is created by Procedure 2-4 “Create Calibration TP Program”. In this example, line 1-4 and line 86-87 are special instructions added for the signature calibration and they should not be edited. Line 5-85 are added for measurement point based on the position program and a call instruction to a measurement TP program is added. Please touch up or add teaching points as appropriate.

	SIG_CAL_G1.TP	Description
1:	UTOOL_NUM = 1	Set tool frame number used for the calibration
2:	CALL CALB_START	An internal setup for the calibration
3:	CALL ACCU_END or CALL ACCU_START	Turn on or Turn Off the High Accuracy Mode
4:	CALL CB_PCSTART(‘SIG_CAL_G1’, 1, 1, 1)	Create a directory to save measurement data
5:	J P[100 : Home] 20% FINE	Move to 1st position copied from position program
6:	J P[1] 20% FINE	Move to Measurement point #1 of position program
7:	CALL RECD_CAL_POS	Call measurement sub TP program
8:	J P[2] 20% FINE	Move to Measurement point #2 of position program
9:	CALL RECD_CAL_POS	Call measurement sub TP program
…		
84:	J P[40] 20% FINE	Move to Measurement point #40 of position program
85:	CALL RECD_CAL_POS	Call measurement sub TP program
86:	CALL ACCU_START	Enable the High Accuracy Mode
87:	CALL CALB_END	Finish the internal setup for the calibration

Listed below is the measurement TP program which is automatically loaded when entering the run calibration program screen. The program will be paused by the PAUSE instruction at line 3. At the line 3 pause measure the reflector position using the laser tracker. Take only one measurement, then press the START button to move the robot to the next position.

	RECD_CAL_POS.TP	説明
1:	PAUSE	Make robot complete stop

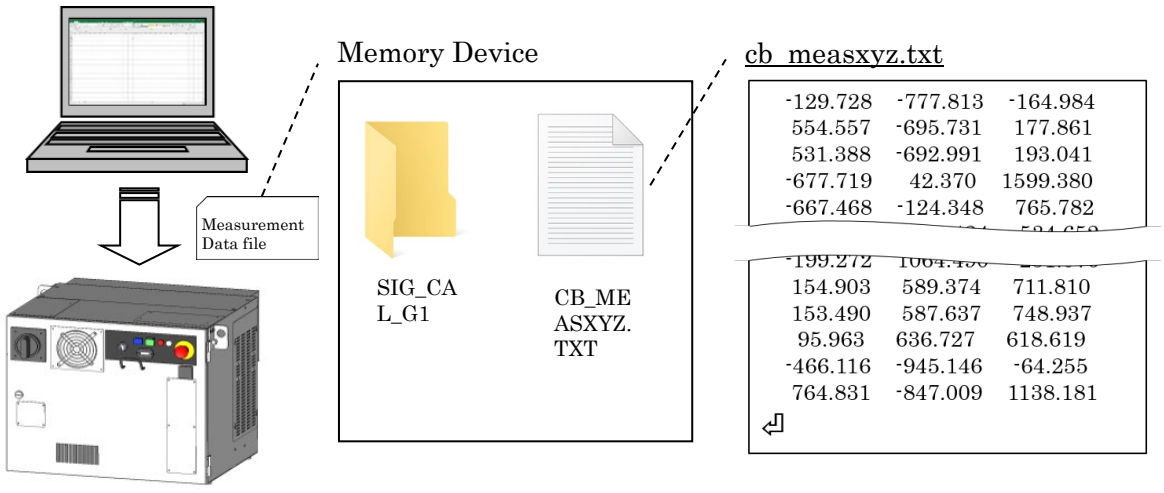
2: CALL CB_PCDATA	Record internal data
3: PAUSE	Measure by laser tracker

NOTE
Recording internal data at line 2 and measurement by laser tracker at line 3 must be executed at same robot posture. ACAL-150 “Near Singularity, Skip Meas.” could occur during the execution of line 2 when the measurement posture is around singularity position. In this case, recording internal data of line 2 is Not performed. Therefore, you must skip the laser tracker measurement in line 3 as well.

Loading Measurement Data to Robot Controller

After taking measurement at all measurement point and the calibration program has run to its completion, please eject the memory device from the robot controller and plug the memory device to the PC... Note that the calibration software created a directory in the memory device and for our example, the directory is named “SIG_CAL_G1”, please do not change the contents of this directory. Copy the measurement data file to the root (top) directory of the memory device. After copying the file, insert the memory device back to the robot controller. The measurement data file should have the following characteristics:

- The file name should be cb_measxyz.txt.
- Each row indicates the X, Y, Z position(unit: mm) of the measurement data. Please use space to separate them.
- Please insert a line break at the last measurement point in cb_measxyz.txt.



NOTE
Make sure the number of measurement data and the number of measurement points match. Calibration parameters can’t be calculated when the number of measured data does not match the number of points measured in the TP program. The cb_cmndxyz.txt contains ideal Cartesian position of measurement points, you can compare the cb_measxyz.txt to the cb_cmndxyz.txt to find the mismatch.

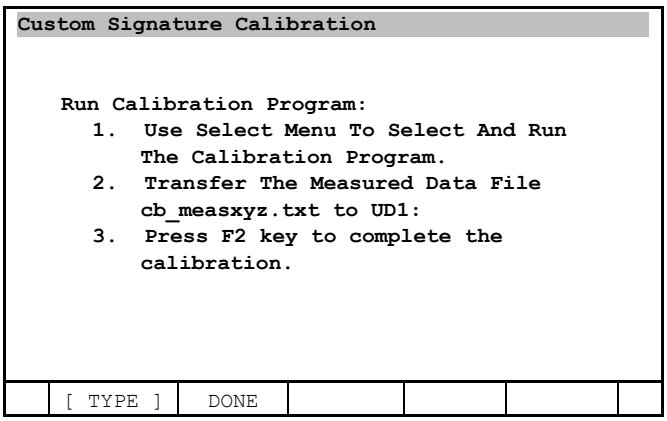
Procedure 2-5 Run Calibration TP Program

Condition

The status of “Create Calibration Prog” item is “DONE”.

Step

1 Display the signature screen. (Procedure 2-1)
2 Move the cursor to the Run Calibration Program item and press the F4 “SELECT” key to display the screen shown below.



3 Select the calibration TP program from the program selection screen and run it. At the pause instruction of measurement TP program, measure the reflector position by laser tracker. After the measurement, press START to move to the next point.
4 After collecting all measurement data, rename measurement data file to cb_measxyz.txt and copy it to the root (top) directory of the selected memory device.
5 Display the Run Calibration Program screen again and press F2 “DONE”.

The cb_measxyz.txt in selected medium will be checked when F2 “DONE” is pressed. If cb_measxyz.txt does not exist or the number of measurement data does not match the number of measurement points, the following message will be displayed. Select “Yes” to continue, or “No” to start over.

In valid CB_MEASXYZ.TXT, skip copy?
Press F2 for YES
Press F3 for NO

6 Return to the signature screen by pressing F2 “DONE” in above screen and the status of the Run Calibration Program item is changed to “DONE”.

2.4.7 Calculate Parameters Screen

The calculate parameters screen, computes the signature calibration data.

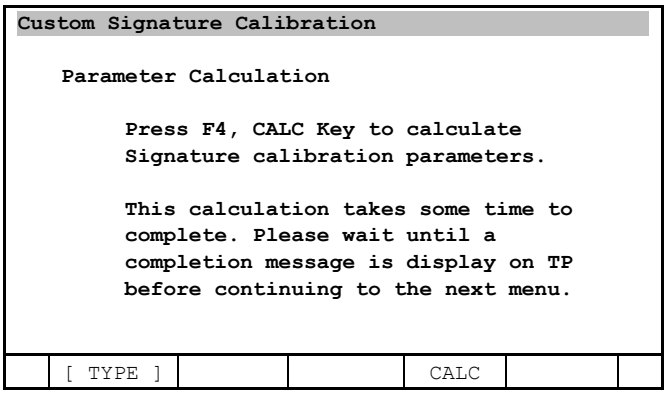
Procedure 2-6 Calculate Parameters

Condition

The status of “Run Calibration Program” item is “DONE”.
Memory device including measurement data is inserted to the robot controller.

Step

1 Display the signature screen. (Procedure 2-1)
2 Move the cursor to the Calculate Parameters item and press the F4 “SELECT” key to display a screen shown below.



3 Press F4 to start the calculation of signature calibration parameters. This calculation may take several minutes. When the calculation is completed, a message "Calibration calculation completed." is displayed at the top of the teach pendant.

If the parameters calculation fails, "Calibration calculation failed." will be displayed. In that case, please make sure the cb_measxyz.txt is recorded correctly.

4 Press the "PREV" key to return to the signature screen and the status of the Calculation Parameters item is changed to "DONE".

2.4.8 View/Apply Calibration Result

On the calibration result screen, following items are displayed.

Contents of calibration result screen

Item	Description
Number of Points	The number of measurement points used for the calculation.
Mean Error	Mean error expected after applying signature calibration parameters.
Max. Error	Maximum error expected after applying signature calibration parameters
Max. Error Point	Measurement point number where maximum error is.
Number of Iteration	Number of iterations to convergence. 10 or less is normal.
Variety of Meas Posture	Indicator of variety of measurement posture. About 50-75% is a target.
Convergence Properties	Indicator of convergence properties. 100% is a target.

For TCP/Uframe Calibration, addition display is added to the items above. Please refer to Subsection 2.5.2 for the details.

After viewing the calibration result and the result is acceptable, apply the signature calibration parameters. If the maximum error is large due to a large measurement error at specific point, it's possible to remove the abnormal point and re-compute the calibration parameters.

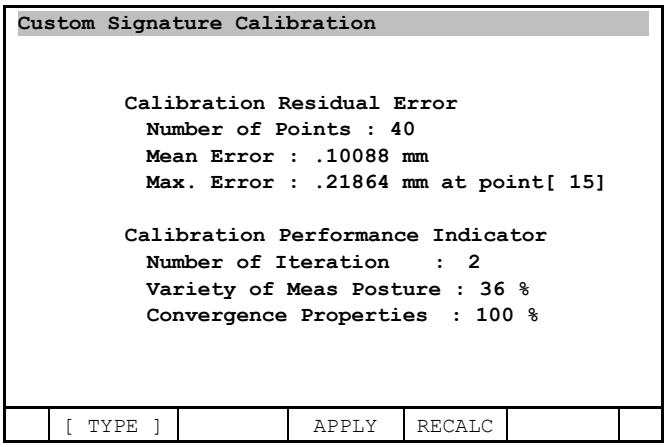
Procedure 2-7 View/Apply Calibration Result

Condition

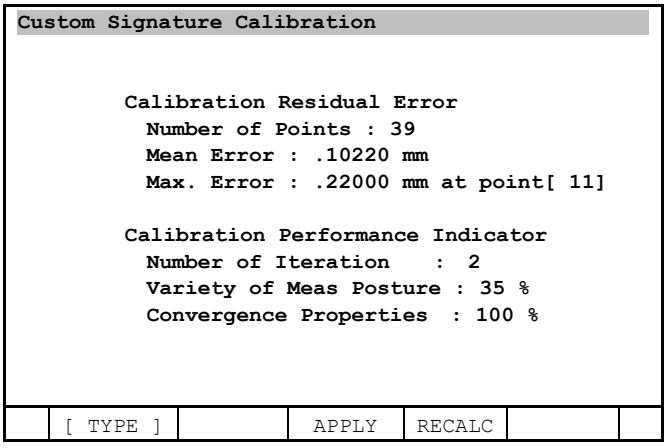
The status of "Calculate Parameters" item is "DONE".
Memory device including measurement data is inserted to the robot controller.

Step

1 Display the signature screen. (Procedure 2-1)
2 Move the cursor to the View Calibration Result item and press the F4 "SELECT" key to display a screen shown in below.



3 Press F4 “RECALC” to start calculation by removing the measurement point with the maximum error. When the calculation is completed, a message “Calibration calculation completed.” is displayed at the top of the teach pendant and return to the signature screen.
4 Display Calibration Result Screen again by step 2. If you want to re-calculate with all measurement points, re-do the procedure 2-6 Calculate Parameters again.



5 Press F3 “APPLY” to apply, the following message will be displayed.
Robot path will change. Continue?
Press F2 for YES
Press F3 for NO
6 Press F2 “Yes”, the signature calibration parameter will be applied.

2.4.9 Create Signature Calibration Parameters Setup File

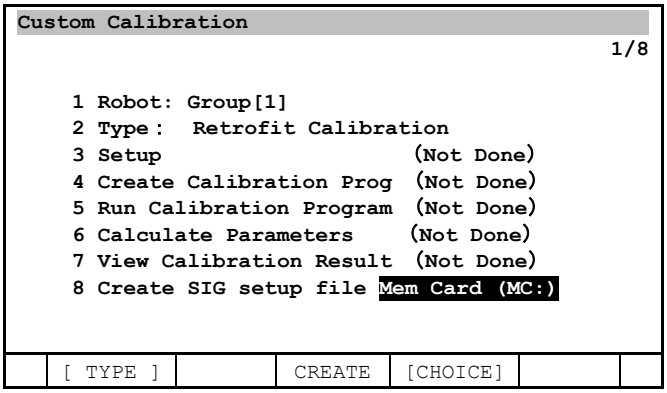
The signature calibration parameters are saved in a system file called SYSMAST.SV. In addition, it is possible to restore only the signature parameters by executing a command file. To create the command file, use the “Create SIG setup file” item in the signature menu of the retrofit calibration, enhanced factory calibration and calibration recovery. The created file is named SET_SIGNATURE.CM. You can apply the signature parameter by executing this command file from the file screen and then cycle power for these parameters to take effect. Note: the same command file is provided for every factory calibrated robot in an attached CD and the file name is PSIRSIG.CM.

NOTE
Please recycle power after executing the command file.

Procedure 2-7 Create Signature Calibration Parameter File Setup File

Step

1 Display the signature screen. (Procedure 2-1)
2 Switch the calibration type to either the retrofit calibration, the enhanced factory calibration or calibration recovery. (Procedure 2-2)
3 Move the cursor to the character string on the right side of the “Create SIG setup file” and press F4 “[CHOCE]” to select the file device to output.



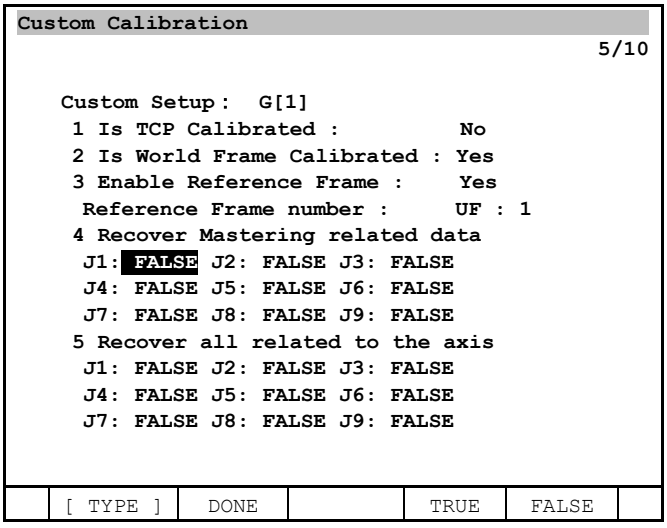
4 Press F3 “CREATE”. A message “MC:set_signature.cm is created” will be displayed when the file is successfully created.

2.5 Advanced Function

2.5.1 Calibration Recovery

Calibration Recovery Setup

Calibration recovery is a function to perform signature calibration after parts replacement. This function computes optimized signature calibration parameters related to the replaced parts. In addition to the setup items described in Subsection 2.4.3, “Recover Mastering related Data” and “Recover all related axis” is selectable in each axis.



Please refer to the table below to select the recovery method corresponding to the parts replaced. You can select multiple recovery methods.

Selection of recovery method corresponding to the replaced parts

Item	Mastering	Calibration Recovery
Pulsecoder battery replacement Pulsecoder cable replacement	Quick Mastering	No need
When the accuracy and stiffness enhancement option is specified, pulsecoder or motor replacement		
Pulsecoder replacement	Mastering methods other than Quick Mastering	Recover mastering related data
Secondary encoder's sensor head detachment/replacement Secondary encoder's scale detachment/replacement		
Motor replacement		
Reducer replacement		Recover all related to the axis
Robot arm replacement		
Wrist replacement		

For the replacement of pulsecoder and motor, when the accuracy and stiffness enhancement option is specified, the quick mastering can be applied to regain the correct pulse count. Therefore, the robot will not need to be recalibrated to maintain its accuracy. For the other robots, the accuracy of the robot may be recovered if mastering can be accurately restored, and the Calibration Recovery may be unnecessary. Mastering Recovery and Axis Vision Mastering functions are available to accurately restore mastering data.

For other maintenance items, even if the mastering data is accurately restored, there is a high possibility that the accuracy will be lost, so Calibration Recovery is recommended.

Position Program

As described in Subsection 2.3.4, position program should include various postures that utilizes most of the motion range for all axes.

To accurately recover the mastering related parameters, the position program should include a wide motion range of the axis to be recovered. For example, assuming J2 motor has been replaced, we need to move J2 as much as possible within the constraint of the production environment.

For world frame calibration the measurement points should not be on the same plane.

For tool frame calibration, the measurement positions should include various orientation change to compute an accurate tool frame.

Recovery of J1 signature calibration parameters

A world frame calibration is required when recovering the J1 calibration parameters after a robot repair like replacing a motor, reducer, casting, or wrist, changes the physical characteristics of the robot. However, a measurement frame needs to be created in advance using the TCP/Uframe calibration and saved to a user frame before the robot is damaged. By setting the user frame number as reference frame number, the world frame calibration can be set to done by using reference frame function. Please refer to 2.5.2 TCP/Uframe Calibration and 2.5.3 Reference Frame Function for the detail.

Recovery of J6 signature calibration parameters

Tool frame calibration is required when recovering the J6 calibration parameters after a robot repair like replacing a motor, reducer, casting, or wrist, changes the physical characteristics of the robot. Regarding the jig required for Calibration Recovery, please refer to 2.9.2 Design of jig for mounting reflector and create it.

2.5.2 TCP/Uframe Calibration

TCP/Uframe Calibration computes robot installation position and measurement reflector position mounted on the robot's tool. Fig 2.5.2(a) shows a schematic of TCP/Uframe Calibration.

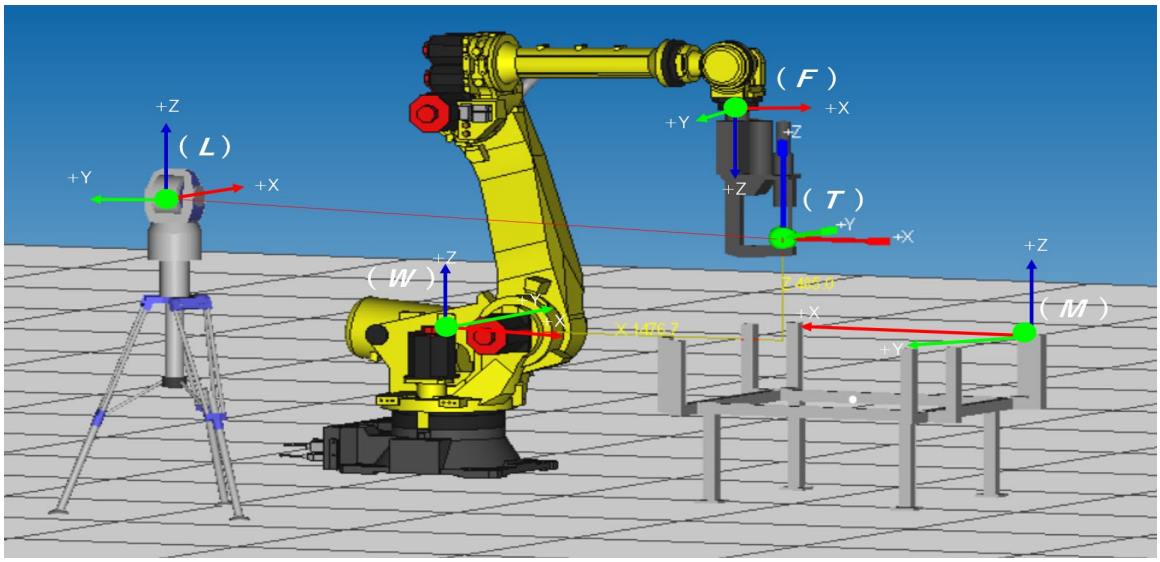


Fig 2.5.2(a) Schematic of TCP/Uframe Calibration

Measurement reflector location is calculated as (*T*) tool frame seen from (*F*) Mechanical Interface Frame. Calculated user frame is position and rotation of (*M*) measurement frame seen from (*W*) robot's world frame.
To calculate tool frame, require 3 measurement points or more that include tool orientation change. To calculate user frame, 4 measurement points or more are required and all the points are not on the same plane.

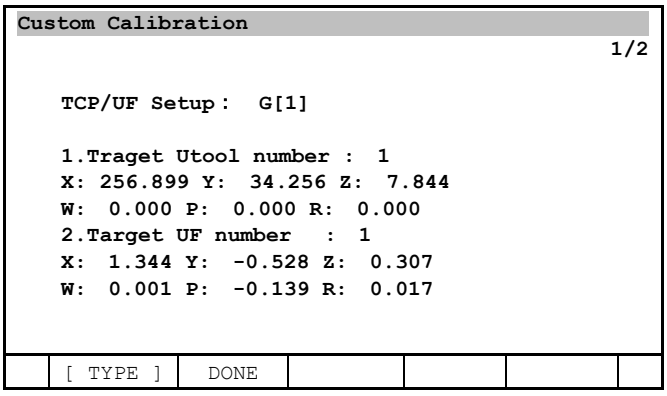
Setup screen and view calibration result screen for TCP/Uframe Calibration are different from another calibration type described in section 2.3 Basic Function.

NOTE
1 To calibrate the tool frame, a rough estimation of the measurement reflector's position (x, y, z) with respect to the Mechanical Interface Frame is required. The estimated position should be within 10 mm radius of the actual TCP position.
2 To compute the user frame, it is required to have an estimated uframe value (x, y, z, w, p, r) before the calibration. A rough position and orientation (x, y, z, w, p, r) within 100 mm and 10 deg is sufficient.

Procedure 2-8 Setup screen for TCP/UF Calibration

Procedure

1 Display the signature screen. (Procedure 2-1)
2 Switch the calibration type to the TCP/Uframe calibration. (Procedure 2-2)
3 Move the cursor to the Setup and press the F4 "SELECT" key to display a screen shown in below.



4 Move the cursor to the number in the right side of “Target Utool Number” and enter the tool frame number to use.
5 Move the cursor to the number in the right side of “Target UF Number” and enter the user frame number to use.
6 After setting up all items, press F2 “DONE” to complete the setup, return to the signature screen and the status of the Setup item is changed to “DONE”.

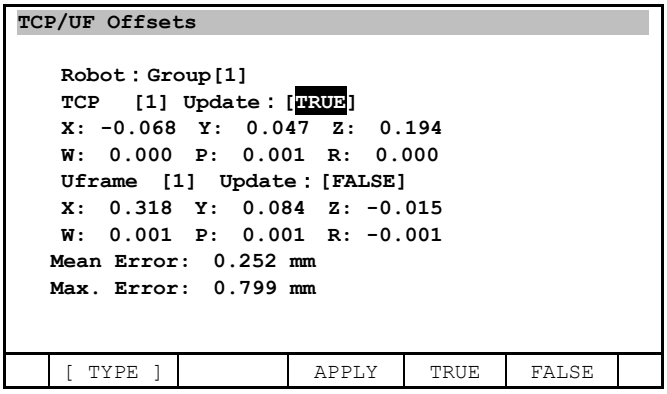
Procedure 2-9 Calibration Result Screen for TCP/Uframe Calibration

Condition

The status of “Calculate Parameters” item is “DONE”.
Memory device that contains measurement data is plugged into the robot controller.

Step

1 Display the signature screen. (Procedure 2-1)
2 Switch the calibration type to the TCP/Uframe calibration. (Procedure 2-2)
3 Move the cursor to the View Calibration Result and press the F4 “SELECT” key to display a screen shown in below.



4 The first X, Y, Z value in the menu is the TCP offset. When the “Update” item is TRUE, the offset can be applied to the TCP (Tool Frame) number. By default, the update item for TCP is TRUE. Move the cursor to the “Update” item and press F5 “FALSE” if you don’t want to apply the displayed result.
5 The displayed result of Uframe is correction amount of the world frame. When the “Update” item is TRUE, the displayed result can be applied to the Uframe (user frame) number. By default, the update item for Uframe is FALSE. Move the cursor to the “Update” item and press F4 “TRUE” if you want to apply the displayed result.
6 Press F3 “APPLY”, the following message will be displayed.
Robot path will change. Continue?
Press F2 for YES

Press F3 for NO

7 Press F2 “Yes”. Tool frame and/or user frame will be updated if the update items are set to TRUE.

2.5.3 Reference Frame Function

The reference frame function is a function to convert measurement data of (*M*) the measurement frame to (*W*) the world frame when position and rotation of (*M*) measurement frame seen from (*W*) world frame is known. As shown in the schematic below, four reference points (measurement reflector holder) are placed on a jig whose positional relationship with the robot is fixed. (*M*) The Measurement frame can be set consistently by measuring four reference points. In this case, we can use the TCP/Uframe calibration to calculate the relationship between *M* and *W* and save the data into a user frame. Once the user frame is known, by using reference frame function, measurement data on (*M*) the measurement frame can be handled internally as data on (*W*) the world frame and can be used for signature calibration. By using this feature, accuracy of calibration recovery can be improved since the reference frame is calibrated.

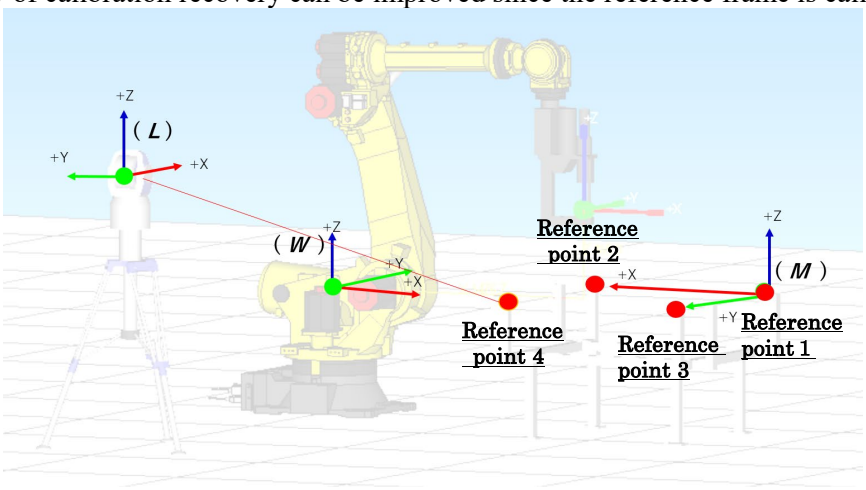


Fig 2.5.3(a) Schematic of the reference frame function

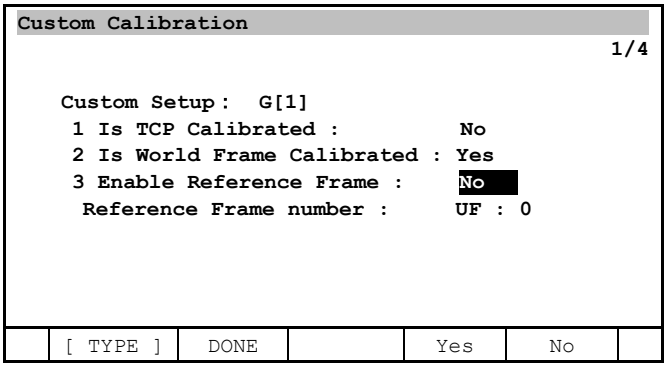
Procedure 2-10 Enabling Reference Frame Function

Condition

The measurement frame seen from (*W*) the world frame is calculated in advance by using TCP/Uframe calibration and the positional relationship is stored in the user frame.

Step

1 Display the signature screen. (Procedure 2-1)
2 Switch the calibration type to the retrofit calibration, the enhanced factory calibration or calibration recovery. (Procedure 2-2)
3 Move the cursor to the Setup item and press F4 “SELECT” to display a setup screen shown in below.



4 Move the cursor to the Enable Reference Frame item and press F4 “Yes” to enable the reference frame function.
5 Move the cursor to the Reference Frame number item and input user frame number that positional relation of the measurement frame seen from world frame is stored.
6 In case “Is World Frame Calibrated” item is set to No, move the cursor to its item and press “Yes” to indicate the world frame calibrated.

2.5.4 Enable/Disable iRCalibration Signature Compensation

For a given TP program position, the robot may reach a different physical position when signature compensated mode is enabled than when the signature compensated mode is disabled, due to the applied compensation. If for some reason you need to enable or disable signature compensated mode, you can use the following KAREL program in your Teach Pendant program and using following technique.

NOTE
Robot position will change slightly when the Signature compensation is enabled or disabled.

Enabling/Disabling the Signature function can execute the following KAREL programs.

CALL ACCU_START(1)	: Enable the Signature function
CALL ACCU_END(1)	: Disable the Signature function

You can create a sub-TP programs like below and calling them from your main TP program. A DO[] = ON is used to indicate the robot is in High Accuracy Mode.

SIG_DISENB.TP	TP program to disable the Signature function
1: IF DO[...]=ON,CALL ACCU_END(1) ;	Call disabling process if Signature enabled signal is ON
2: DO[...]=OFF ;	Set Signature enabled signal to OFF
SIG_ENB.TP	TP program to enable the Signature function
1: IF DO[...]=OFF,CALL ACCU_START(1) ;	Call enabling process if Signature enabled signal is OFF
2: DO[...]=ON ;	Set Signature enabled signal to ON

NOTE
The robot should be at complete stop when toggling the Signature function enabled/disabled.

The Signature enabled/disabled can be toggled in the status screen. Please refer to Procedure 2-12.

2.6 iRCalibration Signature Status Screen

Signature Status Screen

This screen shows the status of a signature calibrated robot and allows you to enable or disable high accuracy mode.

Robot Group

Robot group item indicates the robot group that has its signature status on the screen. You can change the robot group number using function key F3 “GROUP”

Signature

Signature item is the status of *i*RCalibration Signature function itself. You can Enable/Disable *i*RCalibration Signature compensated mode using function key F4/F5.

Calibration

Calibration item displays whether signature calibration is done or not.

Mastering Data

Mastering Data item displays "Changed" if current mastering data have been changed from the mastering data before signature calibration.

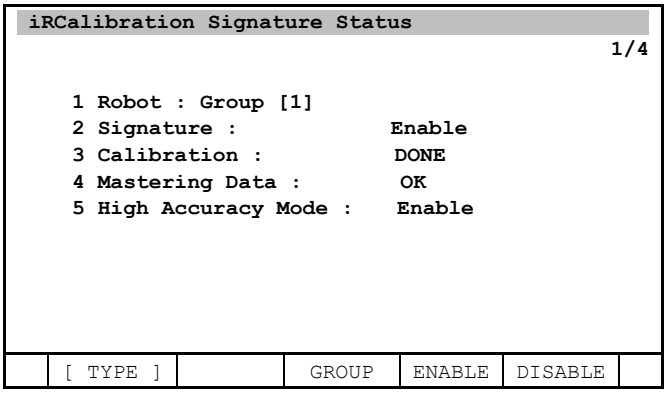
High Accuracy Mode

When the robot applies its signature calibration parameters during motion, the robot is in High Accuracy Mode since the robot compensates for all its mechanical/gravity loading errors during motion using these calibrated parameters. If a signature calibrated robot does not apply these parameters during motion, then robot is not in high accuracy mode. When enabling signature compensated mode, high accuracy mode will be enabled as well. When disabling signature compensated mode, robot will exit the high accuracy mode, too. This item shows the status of high accuracy mode. At the signature calibration, calibration TP program automatically generated toggles the high accuracy mode enabled/disabled properly according to the selected calibration type.
(Data ID: $IRCALIBRATION_SIGNATURE.HIGH_ACCURACY_MODE[a] / BOOLEAN / RO / a: group number)

Procedure 2-11 Display *i*RCalibration Signature Status Screen

Step

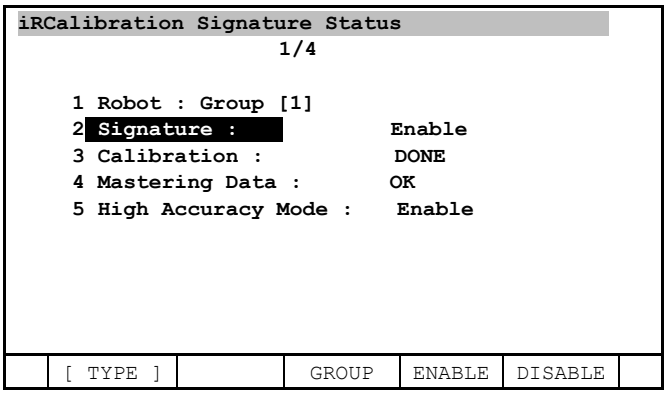
1 Press the [MENU] key to display the screen menu.
2 Select "0 NEXT → 4 STATUS"
3 Press F1, [TYPE] to display the screen selection menu.
4 Select Signature Status. Then the signature status screen will be displayed.



Procedure 2-12 Enable/Disable iRCalibration Signature

Step

1 Display *i*RCalibration Signature Status Screen. (Procedure 2-11)



2 Set the cursor to the Signature item and press F4 "ENABLE" to enable *i*RCalibration Signature. Press F5 "DISABLE" to disable *i*RCalibration Signature. A message will pop up.
To disable ***i***RCalibration Signature: "Disabling Signature. Continue?"
To enable ***i***RCalibration signature: "Enabling Signature. Continue?"
3 Select Yes or No to continue/quit the toggling. Select Yes to continue. Select No to quit.
4 If a message showed up ask to recycle power, please cycle power.

2.7 Mastering

2.7.1 Mastering for Signature Calibrated Robot

Our factory mastering is performed with Signature compensation disabled (that is, the robot is not in high accuracy mode when it is mastered). When a signature calibrated robot needs to be re-mastered due to parts replacement or other reasons, it's necessary to disable the signature compensated mode as well. Please refer to Procedure 2-12 to disable the Signature compensation.

NOTE
Attempting to master a robot while the robot is in signature compensated mode, will result in a popup message prompting you to disable the signature compensated mode. Please follow the displayed message to disable the Signature compensation.

NOTE
The zero position marks indicate the zero-degree position when signature is disabled. The zero position marks will not match when signature is enabled and the robot moves all its joints to zero degree.

After the mastering is correctly performed, please enable the signature compensated mode by Procedure 2-12 and validate accuracy of the robot. If accuracy is not acceptable, please refer to 2.5.2 to perform calibration recovery.

2.7.2 Quick Mastering

Robot can lose its pulse count due to removing of battery, disconnection of Pulsecoder cable or other reasons. The pulse count is comprised of two numbers: the number of revolutions and Pulsecoder value within the revolution. When a robot loses its pulse counts, it loses the number of revolutions value but keeps the pulse value within the revolution data. Quick mastering using this fact to quickly re-establish the pulse counts correctly. Therefore, quick mastering does not affect the accuracy of a signature calibrated robot.

NOTE
When a Pulsecoder is replaced, the quick mastering cannot be applied to regain the correct pulse count. Therefore, the robot will need to be recalibrated to maintain its accuracy.

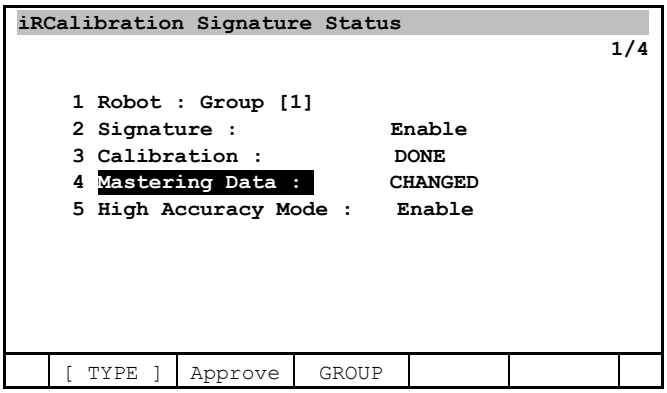
2.7.3 Monitoring the Change of Mastering Data

When a signature calibrated robot needs to be mastered again, it is necessary to validate the robot's accuracy after performing the mastering procedure. If the robot has more error than expected, the calibration recovery is required. When *i*RCalibration Signature software is installed, the software monitors changes to the mastering data and will post a notification icon that the mastering data has changed from the data recorded at the time when signature calibration was performed. If the change of mastering data is intentional, the change can be approved by the procedure below.

Procedure 2-13 Approving Mastering Data

Step

1 Display *i*RCalibration Signature Status Screen. (Procedure 2-11)



2 Set the cursor to the Mastering Data item and press F2 "Approve" to approve the change of mastering data. The following popup message will be displayed.
"Mastering data have been changed and it could affect the absolute position accuracy of the robot."
3 Press OK, then the following message is displayed.
"Validate robot accuracy with new master counts. If accuracy has deteriorated beyond acceptable limit, please recalibrate."
4 Press OK, the then following message is displayed.
"Press YES button if you approved the new mastering data."
Press YES to approve
Press No to disapprove
5 When YES is pressed, the status of mastering data item is set to "OK" and notification icon related to this will disappear.

2.8 Cautions

2.8.1 Accuracy Around Singularity

In general, a robot in its singularity position means either the robot is fully extended or two or more of the robot axes are aligned. When a robot is close to its singularity position, the robot's accuracy will degenerated when compared to a position that is far away from singularity. This is true even for signature

calibrated robots. To maintain the accuracy of the robot, it is recommended to design robot's tool and production cell layout to avoid positions that are close to robot's singularity. The following pictures show the four singularity areas that should be avoided.

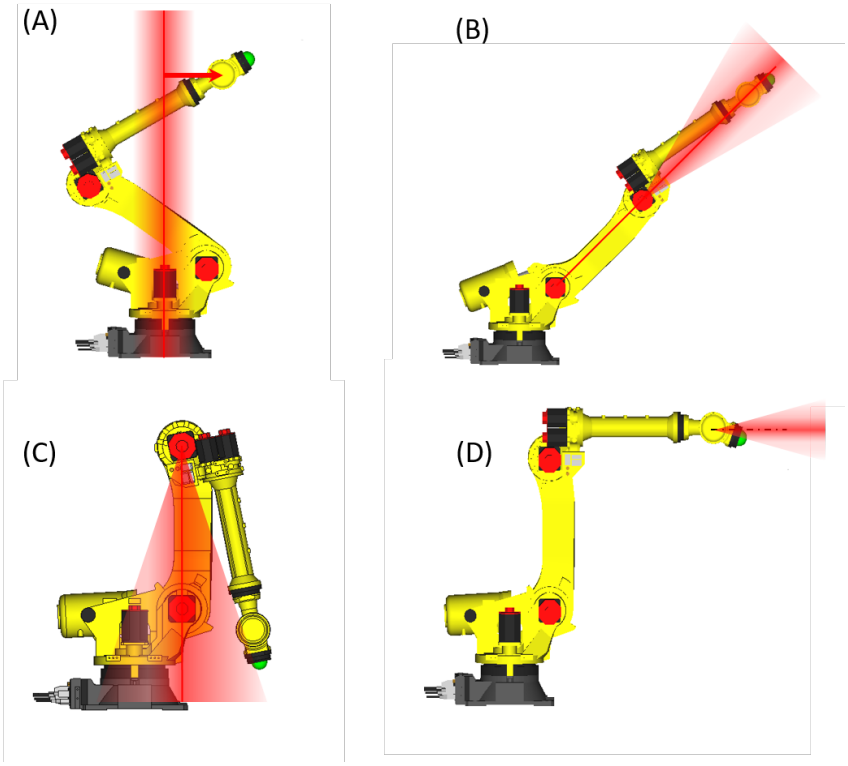


Fig 2.8.1(a) Example of posture around singularity

(A) Where the robot's wrist center is close to the J1 axis.
(B) Where the robot's J2 an dJ3 axis are fully extended.
(C) Where robot's J2 and J3 axis are folded.
(D) Where robot's J5 axis is at 0 degree.

2.8.2 Restrictions

The *i*RCalibration Signature can be used in combination with other software options. However, there are the following restrictions:

- Multi arm configuration could requires special operation in our manufacturing process. Please contact your local representative for the required configuration.
- Auxiliary axis is supported. Integrated Rail is not supported.

2.9 Reference

2.9.1 Measuring the robot installation position

The robot signature option can facilitate offline simulation and reduce the time for program touch ups. However, the error of robot installation position also introduces larger error to the positions in the offline TP programs. As a result, beside the absolute position accuracy of the robot itself, it is also important to accurately reflect the actual installation position of the robot in the offline simulation. There are two ways to accurately measure the robot installation position. The first is to use the TCP/Uframe Calibration described in section 2.3.5. Robot's TCP (tool center point, where the reflector mounted) can be measured in multiple orientations and the coordinate system

of the installation position can be calculated. This method is effective when it is an existing robot or when it is difficult to measure the installation surface. The second method is to measure the installation position using a laser tracker before installing the robot. Since the measurement is performed before the robot is installed, it has the advantage of allowing offline consideration to be carried out in advance. This section describes how to implement the second method with high accuracy.

Robot installation

Install the robot using the locating pin holes of the J1 base shown in Figure 2.2(b). Refer to the MECHANICAL UNIT OPERATOR'S MANUAL of each robot and prepare a base plate or stand on which to install the robot using the locating pin holes.

NOTE

The measurement accuracy of the robot installation position is affected by the flatness of the robot installation surface and the play between the pins and locating pin holes. Design the base plate and stand with the following in mind.

- The J1 base locating pins and the locating pin holes should follow the robot base dimensions in the MECHANICAL UNIT OPERATOR'S MANUAL of each robot.
- The flatness of the robot installation surface should be approximately 0.05 mm.
- A tolerance of H7 is recommended for the locating pin holes on the J1 base.
- A tolerance of h7 is recommended for the locating pins.

Measuring the robot installation position

Use the measurement software of the laser tracker for the measurements and the coordinate system calculations. An example of how to obtain the data required for the measurement software is explained below.

Required items

- Reflector holder with shank for 1.5 inch reflector
  Example: 1.5SM-1.500"-20.0000-Stainless Steel (Shank tolerance +.0000'/-.0005')
- Shankless reflector holder for 0.5 inch reflector
  Example: 0.5SSM-12mm-Stainless Steel (Height tolerance +.0000'/-.0005')
- Laser tracker
- 1.5 inch reflector
- 0.5 inch reflector

Measurement method

1 Prepare the laser tracker.
2 Insert the reflector holder with shank into the locating pin hole.
3 Place the 1.5 inch reflector on the 1.5 inch reflector holder as shown in Figure 2.9.1, and measure the positions of the two locating pin holes.

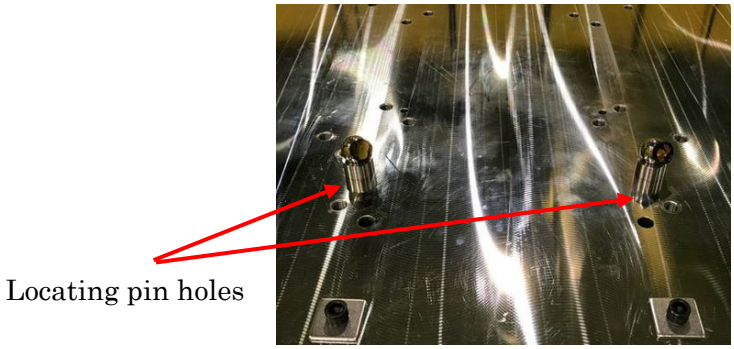


Fig 2.9.1 Measuring the locating pin holes position

4 Place a 0.5 inch reflector on the 0.5 inch reflector holder and measure the installation surface at several points. If the surface is not flat, measure evenly at 10 cm intervals.

5 Enter the nominal values in the X and Y directions of each measurement point as the design values in the robot's world frame. Enter the nominal values of the locating pin holes by referring to the robot base dimensions in the MECHANICAL UNIT OPERATOR'S MANUAL of each robot.

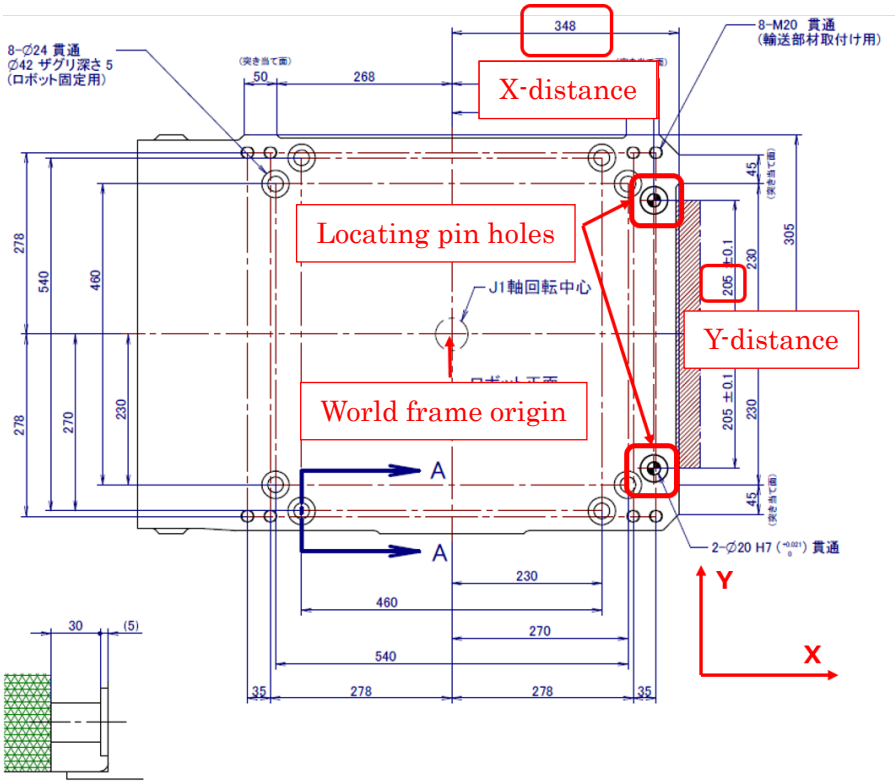


Fig 2.9.2 How to calculate the nominal value of locating pin holes position

6 For the nominal value in the Z direction of each measurement point, refer to the distance from the origin of the world frame to the installation plane in the operating space graph, calculate and enter the height to the reflector.

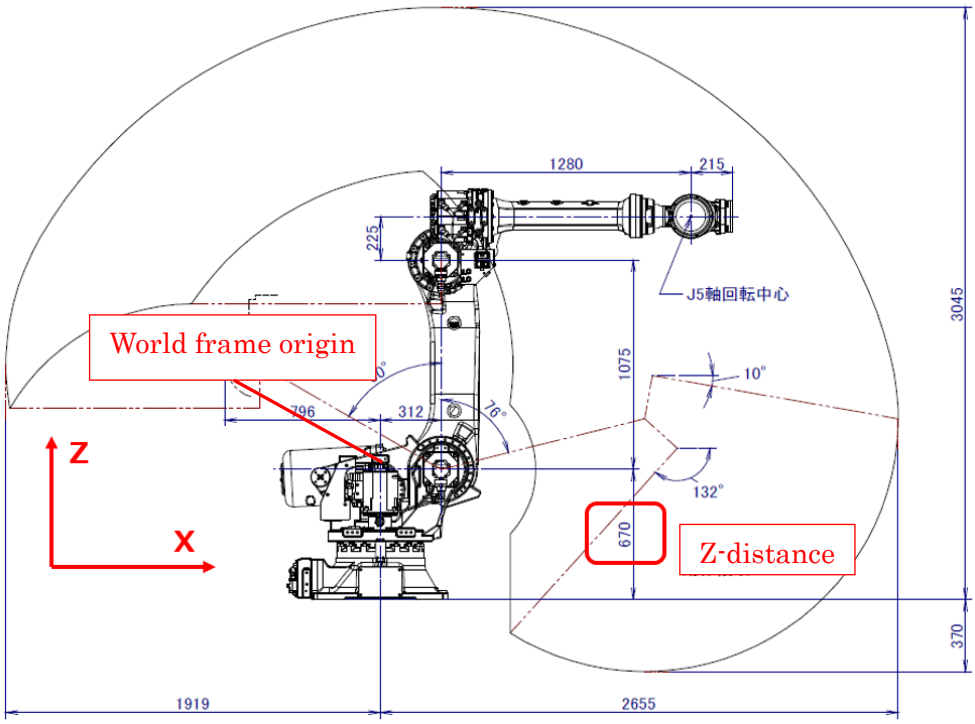


Fig 2.9.3 How to calculate the nominal value of the measurement position on the robot installation surface

7 When calculating the coordinate system with the laser tracker measurement software, enter the nominal values and weights as shown in the table below. In the table, positions 1 and 2 are the positions of the locating pin holes. When entering the nominal positions, refer to step 5 to enter the design values for the X and Y directions, refer to step 6 for the Z direction, and also enter 20 mm as an offset due to the reflector holder and reflector, which is subtracted from 670 mm. Positions 3 and onwards are the measurement positions on the installation plane. When entering the nominal positions, refer to the rough position of the installation plane for the X and Y directions, and enter 12 mm as an offset for the Z direction, which is subtracted from 670 mm. The weights represent the accuracy of the measurement position in each direction. There is no need to enter an accurate nominal value where the weight is 0.

Nominal value and weight of each measurement position

	Nominal value			Weight		
	X	Y	Z	X	Y	Z
Position1	310	-205	-650	1	1	0
Position2	310	205	-650	1	1	0
Position3	550	-550	-658	0	0	1
Position4	450	-550	-658	0	0	1
Position5	350	-550	-658	0	0	1
…	…	…	-658	0	0	1

8 Calculate the world frame of the robot by using the coordinate system creation function of the laser tracker.

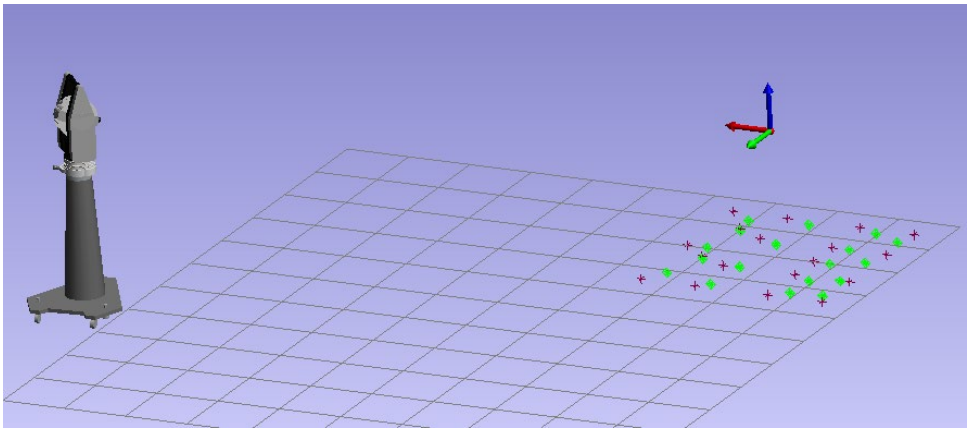


Fig 2.9.4 Calculate the world frame of the robot

2.9.2 Design of jig for mounting reflector

This section describes an example of a jig design where the reflector mounting position can be used as known.

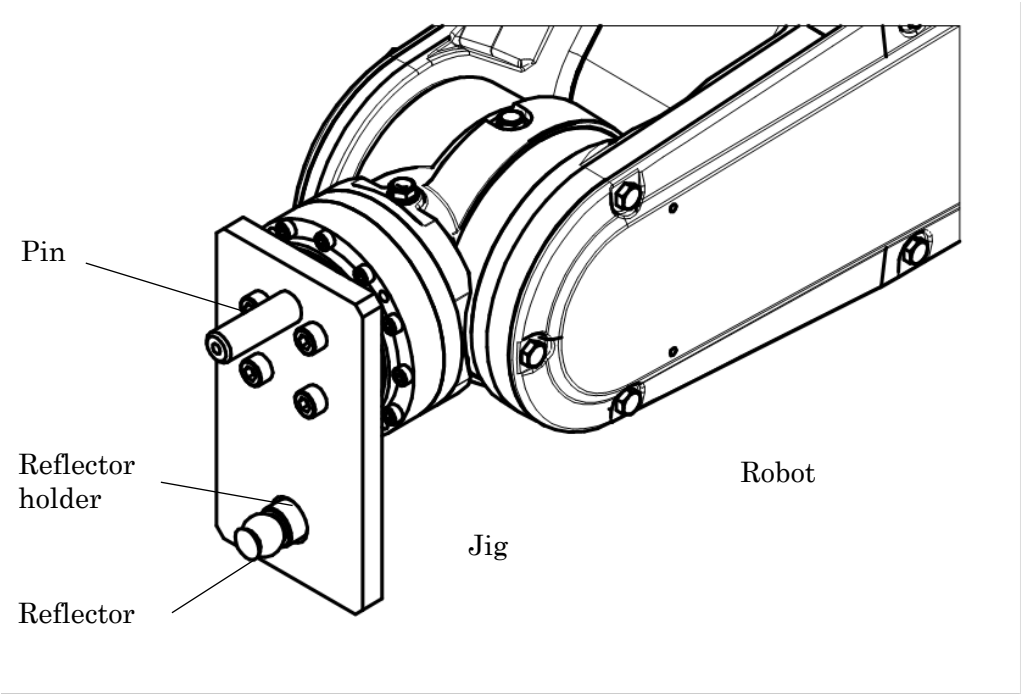


Fig 2.9.5 Jig design example

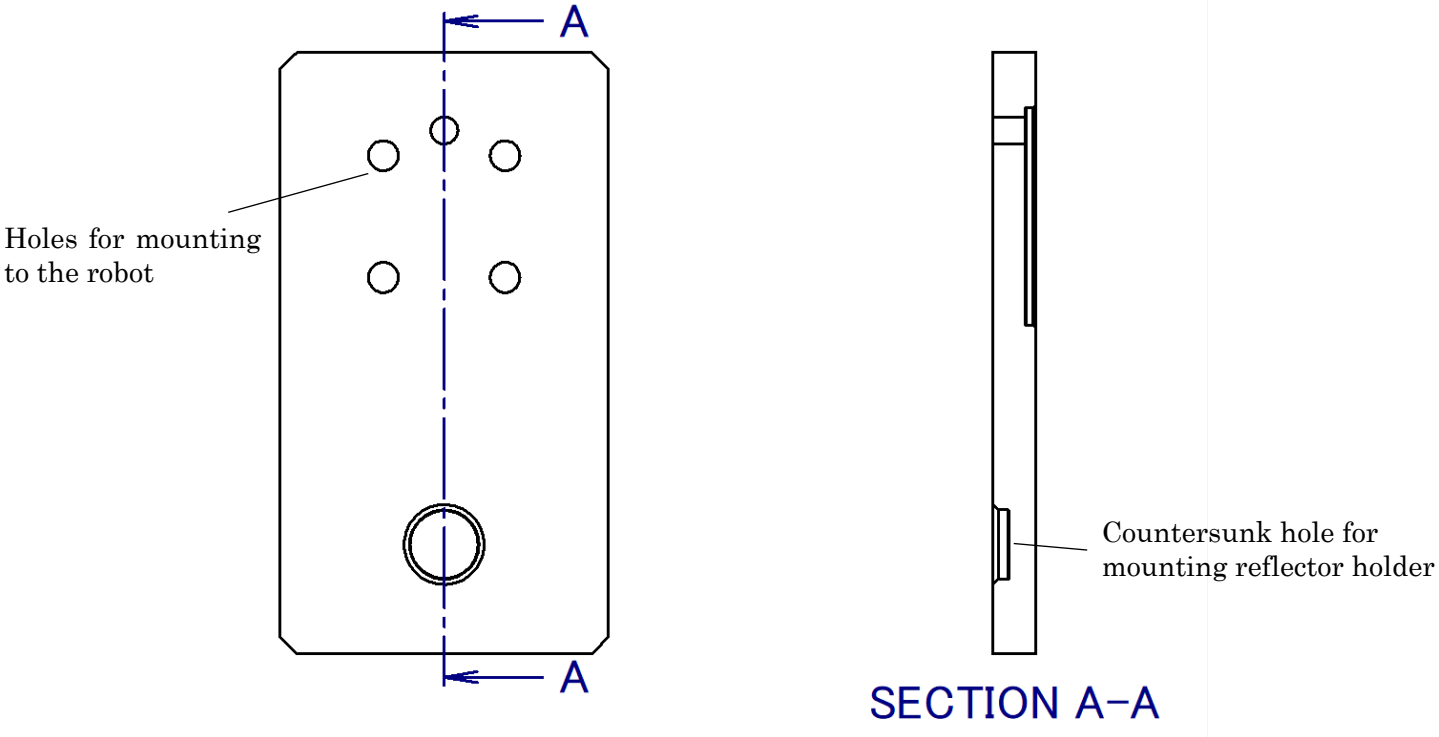


Fig 2.9.6 Cross-section of the jig

Required items

- 0.5 inch reflector
- Shankless reflector holder for 0.5 inch reflector
- Pin

NOTE

The accuracy of the tool frame is affected by the accuracy of the jig and reflector holder. Please note the following.

- Use a pin to attach the jig. Built-in parallel pin is also recommended.
- Holes for mounting to the robot
  - ➢ To attach to the robot, create the holes position and dimensions holes by referring to the robot end effector mounting face in the MECHANICAL UNIT OPERATOR'S MANUAL of each robot.
  - ➢ Machine the hole for inserting the pin with high precision.
- Countersunk hole for mounting reflector holder
  - ➢ Machine the position from the center of the robot end effector with high precision.
  - ➢ Machine the tolerance of the hole's inner diameter with high precision to match the tolerance of the reflector holder.
- Use a reflector holder with guaranteed height and outer diameter precision.
  Example: 0.5SSM-12mm-Stainless Steel (height tolerance +.0000'/-.0005', outer diameter tolerance +.0000'/-.0005')

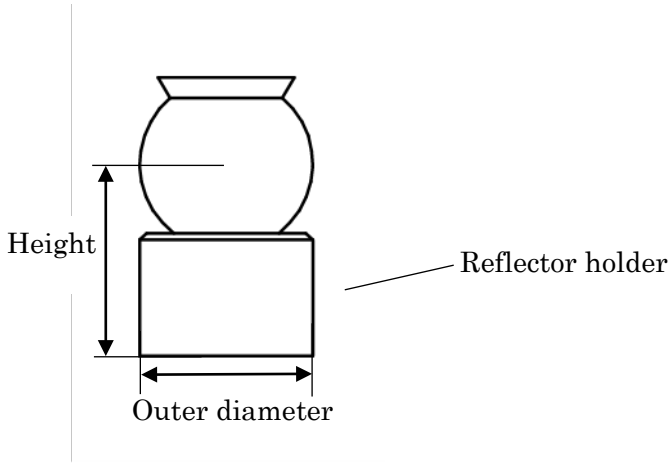


Fig 2.9.6 Reflector holder specification example

3 SOFTFLOAT FUNCTION

The Softfloat function is used to compensate for variances in workpiece precision in applications where the robot is used to mount workpieces on a machine tool.
In most applications, the robot moves accurately to the taught positions. When the robot is used to mount workpieces on a machine tool, variances in workpiece precision may cause the workpiece position to shift relative to the tool. This shifting could result in interference between the workpiece and the tool. The Softfloat function can compensate for these variances.

Softfloat is also very effective if the synchronization speed is unstable, such as when workpiece removal is synchronized with hydraulic extrusion, and when handling workpieces that the robot cannot grip accurately, such as rough-machined workpieces.

3.1 JOINT / CARTESIAN SOFTFLOAT

Function

Joint/Cartesian Softfloat works as follows:

- In Joint Softfloat, you specify softness around each robot joint. In Cartesian Softfloat, you specify softness along/around Cartesian axes.
- You enable and disable Softfloat and specify Softfloat schedules using program instructions.
- You can specify *servo flexibility* for each axis. The term servo flexibility indicates how strongly the axis resists external forces. You can specify a value between 0% and 100%, with a servo flexibility of 100% being the most flexible. You specify the servo flexibility using a schedule table that contains data for one group (for nine axes).
- If an external force exceeding a certain level (sufficient to overcome a static frictional force) is applied to the robot, the robot axes are pushed and moved.
- An external force applied to the robot may prevent it from reaching a taught point. The distance between a taught point and the point that the robot can reach is almost proportional to the magnitude of the external force.
- If a static load such as gravity is applied to a robot, the robot controls the force to maintain its orientation, even if Softfloat is enabled.

Details

This section provides details of the Softfloat function.

- Program instructions

Softfloat uses the following program instructions:

- SOFTFLOAT[n]
  Enables Softfloat using the data in schedule n.
  * The setting of the Softfloat schedule is explained in ”Schedule setting menu.”

- SOFTFLOAT END
  Disables Softfloat.

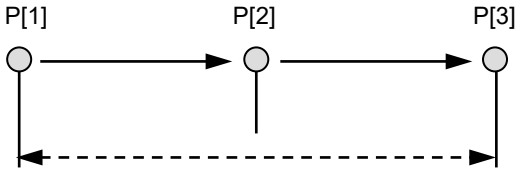
- FOLLOW UP
  When an external force is removed from the robot, it usually tries to go back to the taught point. However, the FOLLOW UP instruction causes the robot to assume that the current position is the taught point, and prevents it from going back to the taught point. Use the FOLLOW UP instruction to override the robot’s usual behavior when an external force is removed.

- Softfloat effective interval

You can use the SOFTFLOAT[n] instruction either as an independent instruction or as motion modifier in a motion instruction. The type of instruction you choose affects the interval during which Softfloat is enabled.

- Standalone instruction
  With a standalone instruction, Softfloat is enabled after the end of the motion specified on the line preceding the standalone SOFTFLOAT[n] instruction.
  In the following example, Softfloat is enabled after the motion specified in line 1 ends, and disabled with a SOFTFLOAT END in line 5.

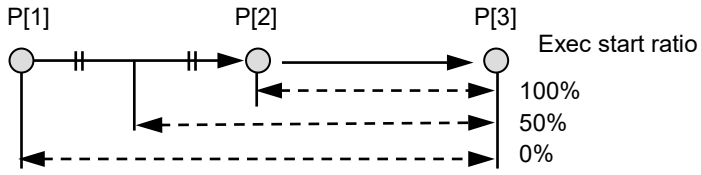
```
1: J P[1] 100% FINE
2:  SOFTFLOAT[1]
3: L P[2] 100mm/sec FINE
4: L P[3] 100mm/sec FINE
5:  SOFTFLOAT END
```



The Softfloat function is enabled.

- Motion modifier
  When specified as a motion modifier, Softfloat becomes enabled during execution of the motion instruction that includes the SOFTFLOAT [n] instruction.
  The point at which Softfloat becomes enabled is determined by the Softfloat schedule item, Exec Start Ratio.
  The Exec Start Ratio is specified as a percentage (from 0% to 100% in 1% steps) of the distance to be traveled before the robot reaches the taught point that corresponds to the motion instruction that includes the SOFTFLOAT[n] motion modifier.
  In the following example, Softfloat is effective between P[1], taught using a motion instruction on line 1, and P[2], taught using a motion instruction on line 2 that contains the SOFTFLOAT[n] instruction.

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



The softfloat function is enabled.

NOTE
For Cartesian Softfloat, you cannot use a Softfloat motion modifier.

- Schedule setting menu

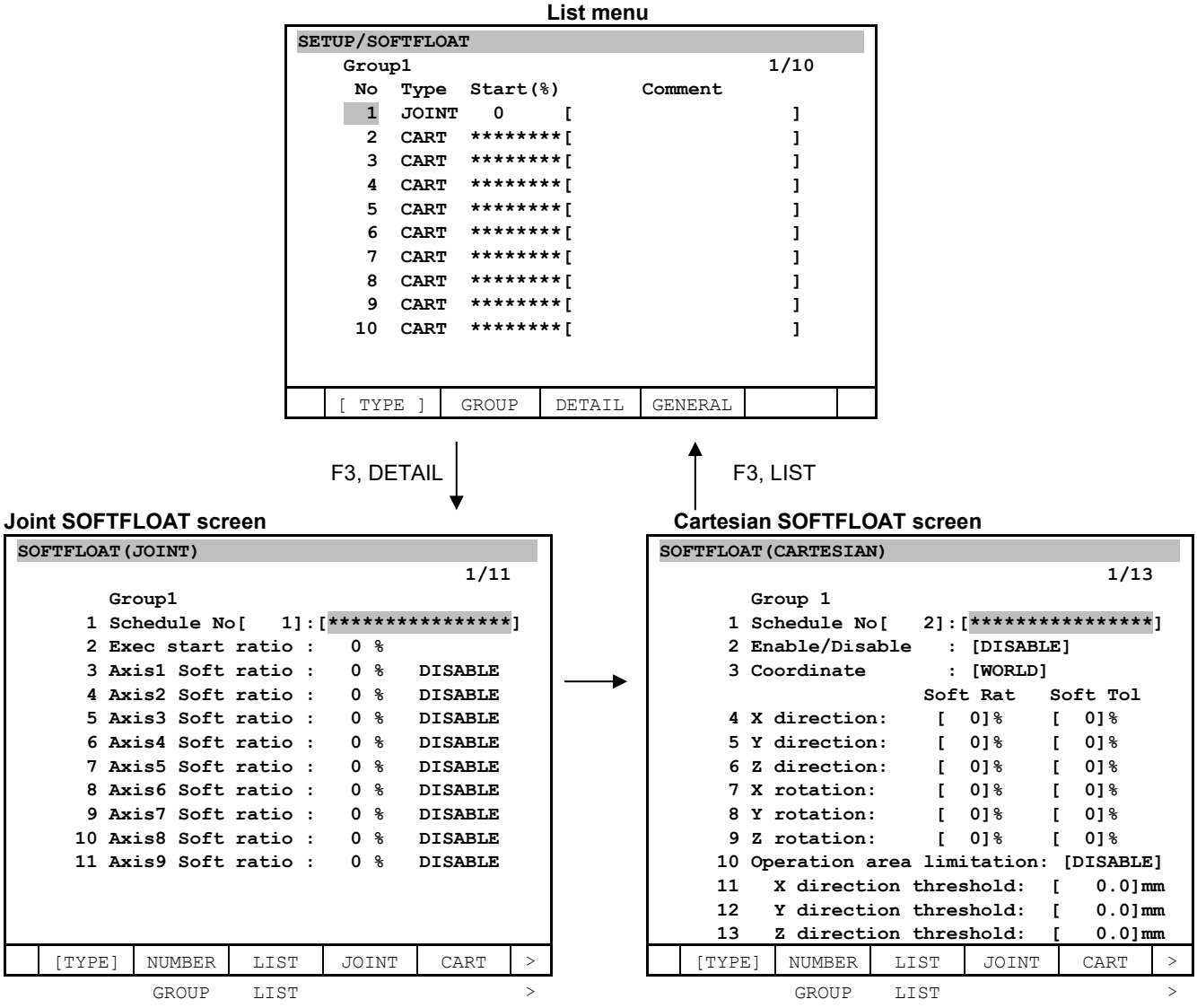
You specify Softfloat schedules on the SETUP /SOFTFLOAT menu, which consists of the following two menus:

- List menu
- Detail menu

The List and Detail screens are toggled using function keys:

- Press F3, DETAIL on the list menu to display the detail menu.
- Press F3, LIST on the detail menu to display the list menu.

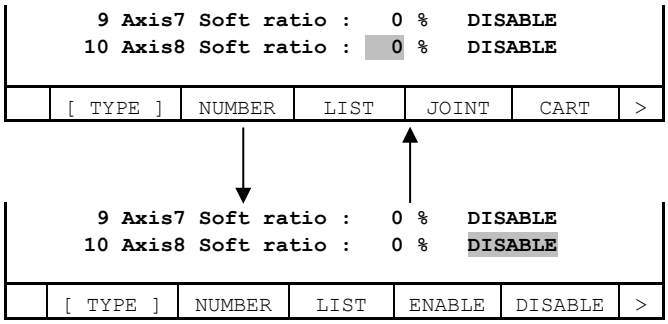
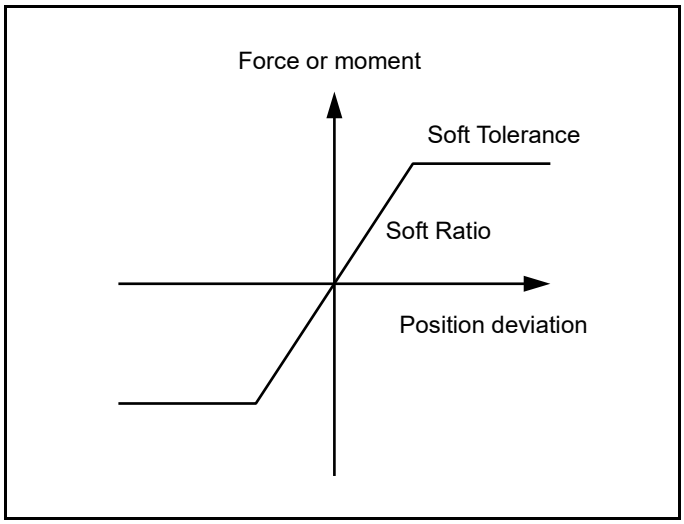
You can specify up to 10 Softfloat schedules.



You can specify the information in Table 3.1 on the SOTFLOAT detail menu.

Table 3.1 SOFTFLOAT Detail setup items

Item	Description
Schedule No. / Comment	This item is the Softfloat schedule number. By default, you can set up to 10 schedules. To add a comment, move the cursor to Schedule No., press ENTER, and enter the comment text,
Exec start ratio	This item specifies the point where Softfloat is enabled when SOFTFLOAT [n] is used as a motion modifier in a motion instruction. See ”Softfloat effective interval” for details.

Item	Description
Soft ratio	This item, also known as servo flexibility or softness ratio, indicates how strongly the axis resists external forces. Specify a soft ratio between 0% and 100% for each axis on line 3 and subsequent lines. A soft ratio of 0% is the hardest and a soft ratio of 100% is the most flexible. Specify whether to enable or disable Softfloat for each axis on line 3 and subsequent lines. Move the cursor to the rightmost end of each line (enabled/disabled setting position) to display F4, ENABLE and F5, DISABLE, and press the appropriate key. **NOTE** Pressing F2, NUMBER selects another page of the detail menu for other schedules. 
Enable/Disable	When this item is set to DISABLE, Softfloat cannot be executed.
Coordinate	This item specifies the frame used: WORLD, USER, or TOOL. **NOTE** If Remote TCP is used, USER indicates the coordinate system for the Remote TCP.
X direction **Y direction** **Z direction** **X rotation** **Y rotation** **Z rotation**	These items specify the softness on or around the X, Y, and Z axes. When the Soft Ratio increases, the spring constant in that direction decreases, allowing the robot to move with less force. When the Soft Tolerance increases, the maximum force and moment applied by the robot in that direction decreases, allowing the robot to move with less force. The difference between Soft Ratio and Soft Tolerance is illustrated below. 

Item	Description
Operation area limitation **X direction threshold** **Y direction threshold** **Z direction threshold**	This item specifies the operation area during Cartesian Softfloat. If the difference between command TCP position and actual TCP position in the frame specified in the Softfloat schedule exceeds the threshold, an alarm occurs and the robot stops. For safety, before executing Cartesian Softfloat, enable “Operation area limitation”, and set values to “X direction threshold”, “Y direction threshold”, “Z direction threshold” so as not to interfere with the surroundings. When “Operation area limitation” is enabled and one of the thresholds is 0.0, an alarm always occurs.

- Multi-arm case

In a multi-arm system, you can enable or disable Softfloat for each robot. You can select the motion group to set up by pressing F2, GROUP and entering the number you want.

For example, if you set as follows:

Schedule No. 1 - G1: Softfloat enabled G2: Softfloat disabled,
Schedule No. 2 - G1: Softfloat disabled G2: Softfloat enabled,

and execute the following program,

SAMPLE1.TP
Group Mask: [1,1,*,*,*,*,*,*]

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

then Softfloat will be enabled only for G1 at line 3, and enabled only for G2 at line 6.

Note that multiple SOFTFLOAT[n] schedules cannot be activated at the same time. For example, the SOFTFLOAT END instruction in line 4 in the above program cannot be omitted. If SOFTFLOAT[2] is executed after SOFTFLOAT[1], a SOFTFLOAT END instruction must be executed before the SOFTFLOAT[2].

Softfloat is not enabled for a motion group that is not a control target of the program. For example, if you set as follows:

Schedule No. 3 - G1: Softfloat enabled G2: Softfloat enabled,

and execute the following program,

SAMPLE2.TP
Group Mask: [1,*,*,*,*,*,*,*]

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

then Softfloat will be enabled only for G1 at line 3.

3.2 PUSHOUT SOFTFLOAT

Pushout Softfloat is suitable for applications where workpiece removal is synchronized with hydraulic extrusion.
Each Softfloat schedule can select a different Softfloat type.

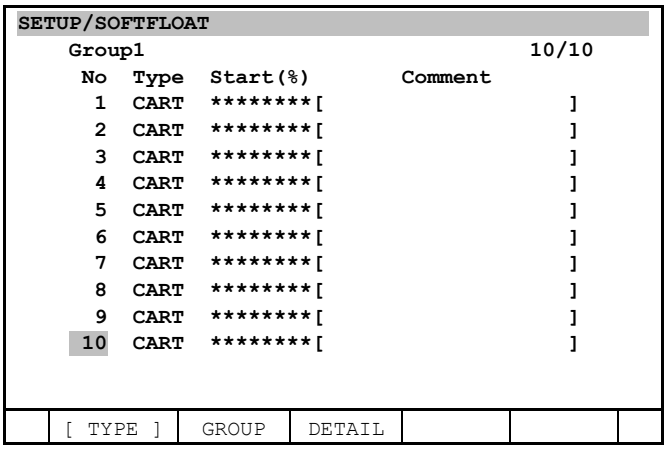
- Joint Softfloat
  You specify softness around a robot joint.
- Cartesian Softfloat
  You specify softness along and around Cartesian coordinates.
  Cartesian Softfloat is suitable for contouring and face matching with robot motion.
- Pushout Softfloat
  In pushout Softfloat, less force and moment than Cartesian Softfloat are necessary.
  You cannot specify softness around Cartesian coordinates.
  You cannot use pushout Softfloat where the J5 axis is within +- 3.5 [deg].
  Pushout Softfloat is suitable for applications where the robot follows the extrusion.

Procedure 3-1 Displaying the SOFTFLOAT (PUSHOUT) screen

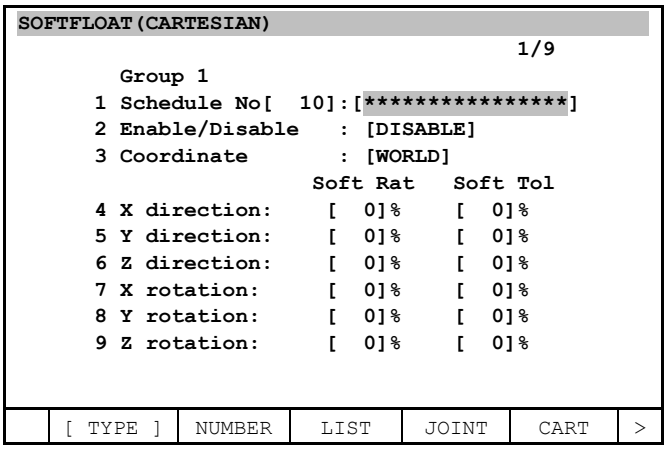
Select PUSH in the SOFTFLOAT detail screen.

Step

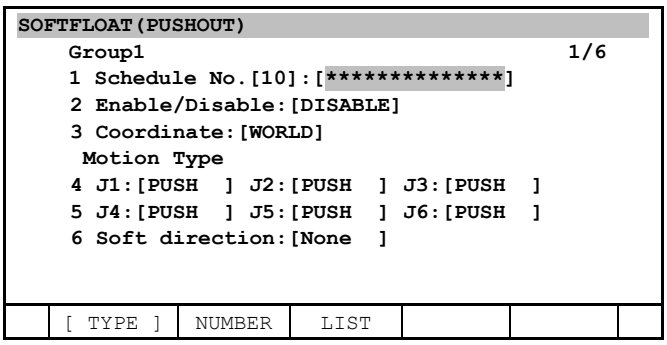
1 Press MENU.
2 Select SETUP.
3 Press F1, [TYPE].
4 Select Softfloat. The SETUP SOFTFLOAT list screen will be displayed.



5 Move the cursor to the schedule number for which you want to set pushout Softfloat and press F3, DETAIL.



6 Press NEXT, > until PUSH is displayed on the function key line.
7 Press F4, PUSH. The SOFTFLOAT (PUSHOUT) detail screen will be displayed.

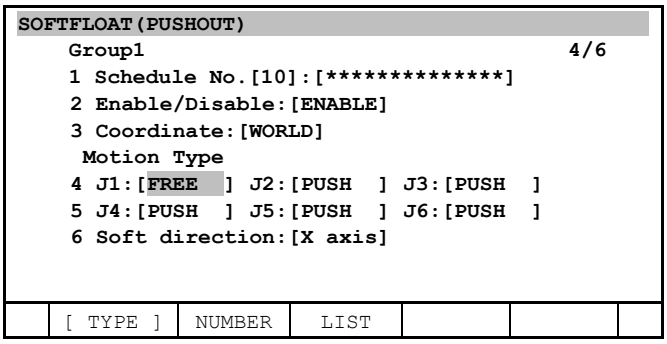


Procedure 3-2 Setting Up Pushout Softfloat

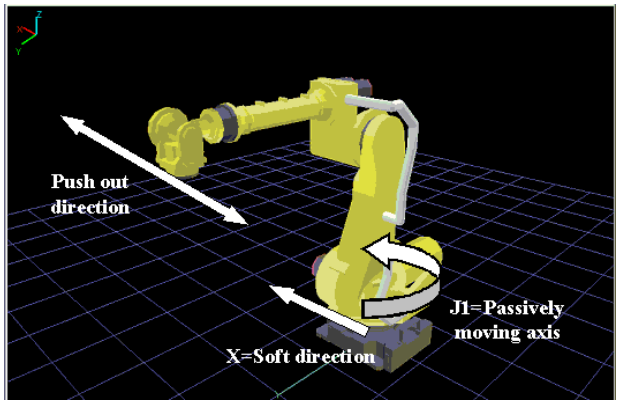
To set up pushout Softfloat, you must set up Motion Type and Soft Direction in addition to Cartesian Softfloat. You do not need to set the soft ratio or soft tolerance.

Step

1 Set the Enable/Disable and Coordinate items in the same way as Cartesian Softfloat.
2 Set the Soft direction item to the pushout direction for the schedule you selected.
3 Set the Motion Type of the axis you want to passively follow the external force to FREE. Only one axis can be FREE in one motion group. You can select only J1 or J2 as a FREE axis.



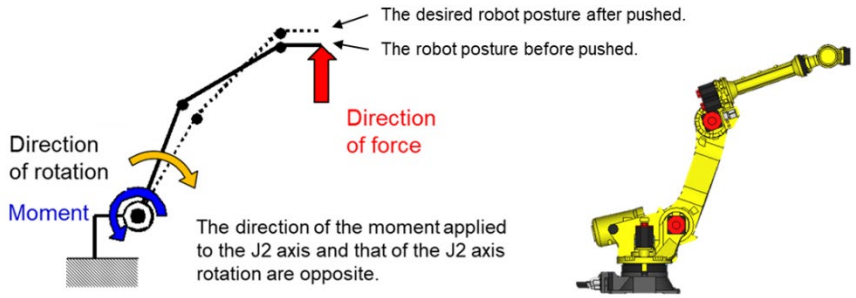
Example
When J1 is at 90 degree, to make the X-direction of the World coordinate soft, the Motion Type of J1 should be set to FREE.



⚠ CAUTION

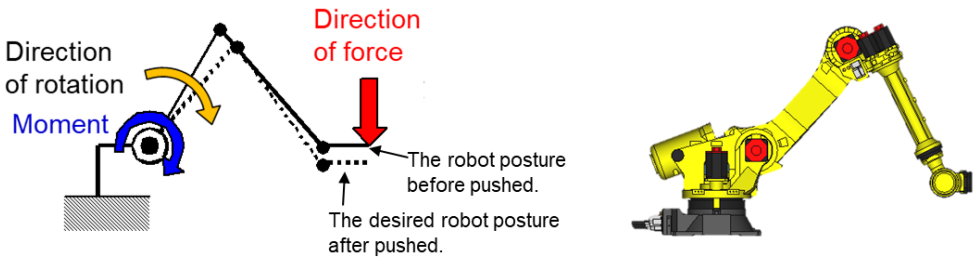
In pushout Softfloat, some axes cannot be used as FREE axes depending on the robot posture.
The following examples show when the J2 axis is set to FREE and the soft direction is set to Z axis.

Bad case



In the case shown in this figure, a force is applied in the positive direction of the Z axis, and a counterclockwise moment is applied to the J2 axis. However, the direction of this moment is opposite to the direction in which the J2 axis should rotate to move in the positive direction on the Z axis. In this case, the J2 axis will not be able to passively follow the external force properly. For this reason, do not set the J2 axis to FREE.

Good case



In the case shown in this figure, a force is applied in the negative direction of the Z axis, and a clockwise moment is applied to the J2 axis. The moment is in the same direction that the J2 axis must rotate to travel in the negative direction on the Z axis. In this case, set the J2 axis to FREE.

Procedure 3-3 Using pushout Softfloat

This procedure is the same as the procedure for Joint and Cartesian Softfloat.

Step

1 On the Program edit screen, open the program that will use pushout Softfloat.
2 Press NEXT, > until [INST] is displayed on a function key label.
3 Teach the SOFTFLOAT[…] instruction on the line where you want to start pushout Softfloat.
4 Enter the Softfloat schedule number of the pushout Softfloat in the index of the SOFTFLOAT[…] instruction.
5 Teach a SOFTFLOAT END instruction on the line where you want to stop pushout Softfloat.

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

3.3 SOFTFLOAT GUIDELINES

Timing to start softfloat

When Softfloat is enabled, the robot must be in a state where no external force is being applied. If a SOFTFLOAT[n] instruction is executed with an external force other than gravity on the robot, the robot may drop or rise slightly, resulting in a “Stop/Move error excess” or “Softfloat time out” alarm.
For example, in a pushout handling application, you should create a program in which Softfloat starts before the hand (gripper) is closed.
This guideline applies to all kinds of Softfloat, including Joint, Cartesian, and pushout.

Bad example

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

Force from outside will be added if taught point has an error and softfloat starts after hand closed.

Good example

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

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

Improving softness

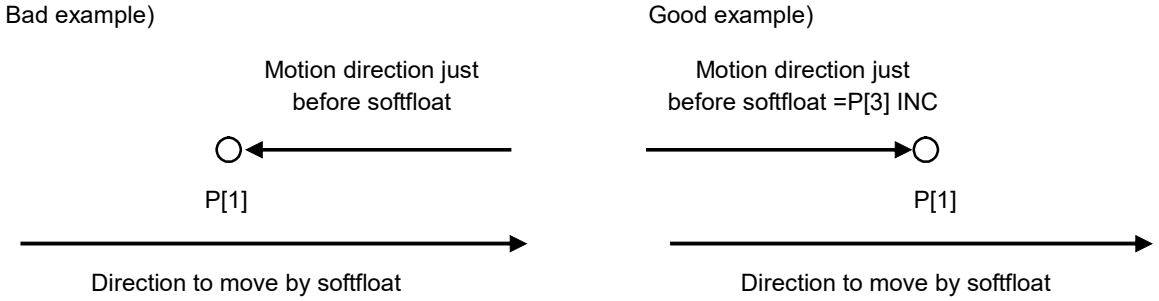
Even with the same Softfloat schedule settings, the softness of Softfloat depends on the direction of the motion just before Softfloat starts.
This is because Softfloat tends to move more soft in the direction of the previous motion and less soft in the opposite direction of the previous motion.
To compensate for this behavior, in the program before the SOFTFLOAT instruction, move the robot slightly in the opposite direction of the desired motion and then return to the position where you want to start Softfloat. The distance necessary is 1-2mm or longer.
For example, if you want to start Softfloat to move in the +X direction in P[1], you should add the following motion instruction between the motion instruction to P[1] and SOFTFLOAT instruction.

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

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

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

With this method, the robot will become more soft to move in the +X direction and less soft to move in the –X direction.
This guideline applies to all types of Softfloat: Joint, Cartesian, and pushout.

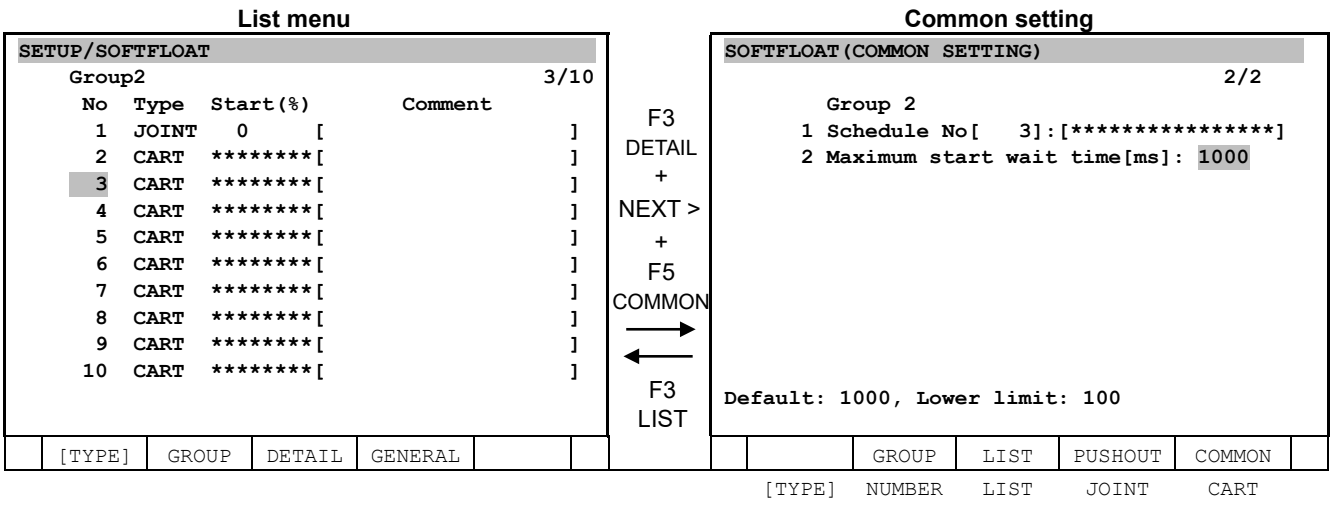


Reducing start wait time of Softfloat

Softfloat function starts controlling of softness at SOFTFLOAT[n] instruction. It waits up to 1 second when SOFTFLOAT[n] instruction is executed in order to start controlling after the robot has completely stopped (If the robot is not moving, Softfloat starts without waiting fully 1 second). This is because if Softfloat starts before the robot's vibration has completely diminished, the movement during Softfloat may become unstable, for example, the robot may start to move slowly. However, depending on the execution conditions, the robot may have almost stopped when SOFTFLOAT[n] instruction is executed, and its movement during Softfloat may be stable enough even without waiting for 1 second. In such cases, you can reduce the maximum start wait time of Softfloat.

> **⚠ CAUTION**
> The reason the maximum start wait time is 1 second by default is to ensure stable robot's movements during Softfloat under various conditions. By reducing the maximum start wait time, the movement may become unstable during Softfloat.
> If you reduce the maximum start wait time, be sure to tune it confirming robot's movement during Softfloat is stable. Specifically, you shorten the time by 0.1 seconds (100 milliseconds) at a time, and each time you change the time, you run program through about 5 cycles to confirm that the robot arm does not rise or fall on its own.

You can change the maximum start wait time in milliseconds on the common setting screen in the detail menu. When you place the cursor over the value of maximum start wait time, the bottom line on the screen displays the default value and the lower limit value that can be input. To open the common setting screen, first press F3, DETAIL on the list menu to the details menu. Next, press NEXT, > until COMMON is displayed on the function key line. Finally, when COMMON is displayed, press F5, COMMON.
The common setting screen is available from software version 7DH1/16 (V10.10P/16). You can change the maximum start wait time on this software version and later.



> **⚠ CAUTION**
> In software version 7DH1/16 (V10.10P/16) to which common setting screen is newly added, when you move from common setting screen to each Softfloat detail screen, the items you previously set on the Softfloat detail screen will be initialized. Therefore, if you switched the screen as mentioned above, check again that the parameters you set on the Softfloat detail screen are registered correctly. In software version 7DH1/17 (V10.10P/17) and later, the parameters of the items you have set will be kept even if you move from common setting screen to each Softfloat detail screen.

You can change the maximum start wait time for each motion group and each schedule No. However, the maximum start wait time is effective only when the schedule No. is enabled. In a multi-arm system, the longest wait time among the effective schedule No. is applied to ensure stable movements of all robots. For example, if you set up as follows and execute SOFTFLOAT[n] instruction;

Schedule No. 1 - G1: Maximum start wait time 750ms (Softfloat enabled)
G2: Maximum start wait time 900ms (Softfloat enabled)
Schedule No. 2 - G1: Maximum start wait time 750ms (Softfloat enabled)
G2: Maximum start wait time 900ms (Softfloat disabled)

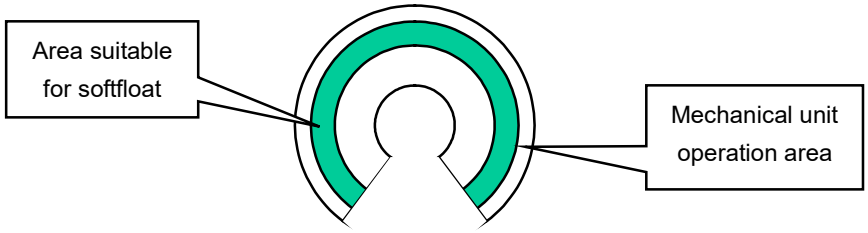
SOFTFLOAT[1] will apply the longest wait time 900 milliseconds of G2, and SOFTFLOAT[2] will apply the 750 milliseconds of G1 because G2 Softfloat is disabled.

3.4 CARTESIAN SOFTFLOAT GUIDELINES

You can change the softness along and around Cartesian coordinates.
Cartesian Softfloat is ideal for applications such as aligning surfaces while the robot is moving.
This section includes guidelines for effectively using Cartesian Softfloat.

General Cartesian Softfloat Guidelines

- Layout and tool selection
  It is recommended that Softfloat be performed in an area of about 10-20[%] inside the outside border of the work envelope.
  Also, use tools that are as short as possible.



- Soft ratio setup
  Set the Soft Ratio for the direction or rotation you want to make soft to 100%.
  Adjust the Soft Tolerance in the pushing direction to limit the force and avoid pushing with too large a force.

- Taught motion speed
  In Softfloat with robot motion, if the taught speed is too slow, the robot might not be able to move.
  Teach a speed of greater than approximately 100mm/s.

Contouring application guidelines

- Softfloat schedule setup
  Example: Refer to the following table for an example of pushing in the Z direction.

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

- TCP setup
  In applications where you want to maintain a constant tool direction, set the TCP X, Y and Z =0 only when Softfloat starts. This makes the difference clear between softness along soft direction and hardness along hard direction.

Example

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

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

TCP set as original tool

Surface matching application guidelines

- Softfloat schedule setup
  Example: Refer to the following table for an example of pushing in the Z direction.

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

Z rotation	100	%	0	%	

- TCP setup
  Use the tool frame set as the actual tool. You do not have to change the TCP when Softfloat starts.
  Select the shortest possible tool. If the tool is too long, the X, Y, and Z directions will not be able to become hard enough.
  Apply the moment in the direction the surfaces meet. Select a tool that can apply a moment in the direction the surfaces meet by pushing, even with the largest orientation mismatch.

3.5 CAUTIONS AND LIMITATIONS

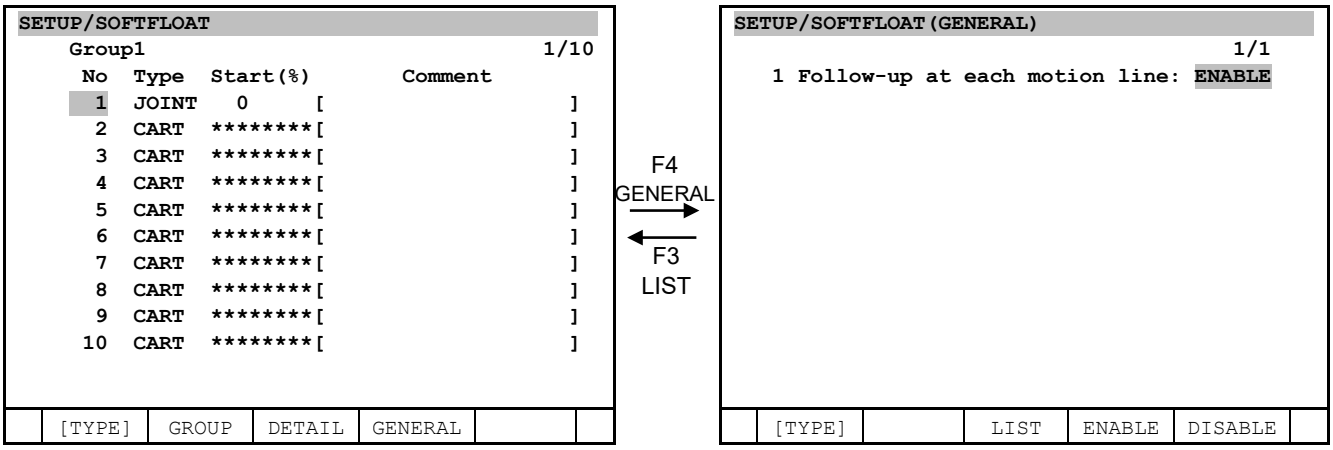
When using Softfloat, note the following cautions and limitations.

- The following limitations apply when Softfloat is enabled:
  - During teaching, the robot will not always follow the taught path.
  - The taught path changes according to the override.
  - The operation time may increase compared to normal operation.
- Softfloat is automatically disabled when:
  - Program execution starts.
  - Program execution ends.
  - The program is stopped due to an alarm that turns off the servo.
  - The robot is jogged while the program is paused.
  - The program is resumed after the cursor has been moved manually while the program was paused.
  - Backward execution is performed.
  - Power is turned on.
- When a paused program is resumed, Softfloat will use the schedule settings (such as enable/disable and start ratio) that were in effect when the program was paused. However, Softfloat will be automatically disabled if any of the operations in the previous bullet items are performed.
- Softfloat can be enabled only with a SOFTFLOAT instruction.
- When Softfloat is enabled, the robot moves with CNT 0 termination type (no position check is made) even if FINE has been specified as the motion instruction termination type.
- When Softfloat is enabled, if an external force causes the robot to move beyond a certain distance, one of the following servo alarms occurs:
  - If the robot is at rest:

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

  SRVO-024 Move error excess(G:i A:j)
- If you try to enable Softfloat while a brake is applied, the brake is released automatically before Softfloat is enabled.
- When Softfloat is enabled, brake control is disabled.
- If the motion group mask in a program is [*,*,*,*,*,*,*,*] (there is no motion group), when the program tries to execute Softfloat instructions, the following alarm occurs:
  INTP-216 (program name, line number) Invalid value for group number
- Minimize the motion when Softfloat is enabled. Depending on the Softfloat ratio and distanced traveled, the weight balance may vary thus shifting the axis on which gravity is applied upward or downward.
  For the same reason, when Softfloat is used as a motion modifier, minimize the motion range and keep the motion speed as slow as possible.
- During Softfloat, follow-up processing is executed at each motion instruction by default. To disable follow-up processing, on the SETUP SOFTFLOAT list screen, press F4, GENERAL, and then disable the Follow-up at each motion line item.

List menu **General setup menu**



- No external force can be applied when Softfloat starts, ends, and during follow up.
  If Softfloat starts when an external force other than gravity is applied, the robot arm may rise or fall slightly, resulting in a “Stop/Move error excess” or “Softfloat time out” alarm.
  In these cases, modify the TP program by changing the position data or by adding a wait instruction so that no external force is applied at the start or end of Softfloat.

NOTE
Follow-up definition
With Softfloat, external forces applied to the robot cause it to operate at positions slightly different from taught positions. When the external force is removed after completion of the operation, the robot usually tries to return to a taught point abruptly. Follow-up prevents this abrupt motion.

- Softfloat cannot be installed with ArcTool.
- Softfloat cannot be used with Line Tracking.
- Softfloat cannot be used on an Independent Axes or Positioner. Also, joint Softfloat cannot be used on an extended axis added to a robot group.
- When using Softfloat together with the Gravity Compensation function, pay attention to the following.
  The Gravity Compensation function calculates the compensation amount based on the command position. Even if the command position and the actual position deviate due to Softfloat, the Gravity Compensation function calculates the compensation amount based on the command position. Therefore, for example, in an application where Softfloat is enabled and the robot is held stationary by applying it to a pressing surface, the Gravity Compensation amount might change and the robot might move slightly after pressing.

4 CONTINUOUS TURN FUNCTION

The Continuous Turn function allows unlimited continuous rotation of robot axes in one direction for the final axis or an additional rotation axis. A *final axis* is the last axis of a robot. For example, if your robot has six axes excluding extended axes, J6 is the final axis.

Continuous Turn can be used to move a device that requires continuous rotation, such as a conveyor, pump, or grinder, about a robot axis or additional rotational axis.

NOTE
Prior to the R-50iA controller, this feature was named the Continuous Rotation function in the operator's manual.

4.1 OVERVIEW

You set up Continuous Turn on the SETUP Continuous Turn screen. You start and stop Continuous Turn in a teach pendant program.
You must set up Continuous Turn before you can use it.
You can designate only one Continuous Turn axis for each motion group. The axis must satisfy one of the following requirements:

- Final axis of the robot
- Final axis of the positioner
- Final axis of integrated additional rotation axes
- Any of auxiliary (non-integrated) additional rotation axes
- Final axis of the independent axes

The Continuous Turn axis must satisfy the following mechanical requirements:

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

NOTE
Gear Ratio (motor) and Gear Ratio (axis) mean a ratio in an integer form that represents the amount of change of the axis position (angle displayed on the current position screen) by motor rotation.
For example, if the axis position changes by 1 degree when the motor rotates 100 degrees, Gear Ratio (motor) and Gear Ratio (axis) are as follows:
Gear Ratio (motor) = 100, Gear Ratio (axis) = 1.

The Continuous Turn option is required to use this function.

Angular representation of a Continuous Turn axis

When Continuous Turn is enabled, the axis assigned as a Continuous Turn axis allows unlimited rotation. As a result of this, the angle of that axis is indicated by its relative position between -180 deg and +180 deg, not by its absolute position. For example, Fig. 4.1 (a) shows rotation from 0 deg to 200 deg in the positive direction. However, the angle of the axis after the rotation is -160 deg, not 200 deg.

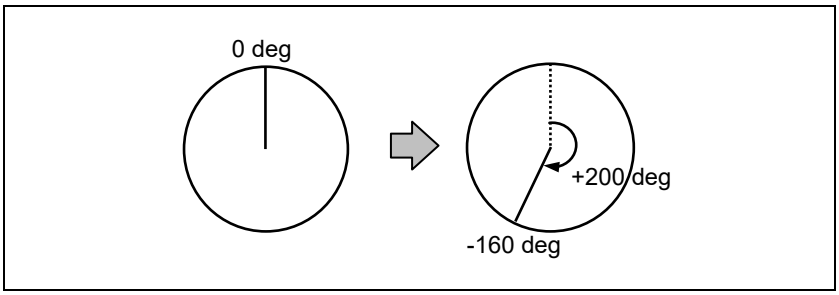


Fig. 4.1 (a) Angle on the Continuous Turn Axis

When Continuous Turn is enabled but not executed (see the next section for information on how to use Continuous Turn), the Continuous Turn axis moves from the current position to the target position by selecting the direction of rotation with the smallest amount of rotation. Normally, the rotation direction about the axis is determined by the relationship between the current and target angles. This *shorter-way motion* is effective in reducing cycle time. However, after the end of a continuous motion, it may not performed *shorter-way motion*. Refer to Section 4.3.2 for details on this.

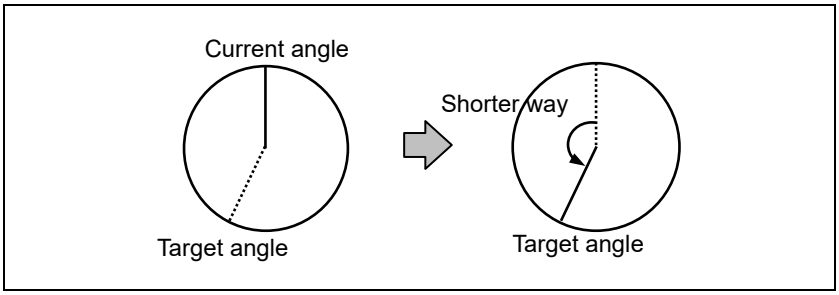


Fig. 4.1 (b) Shorter-Way Operation

4.2 SETUP

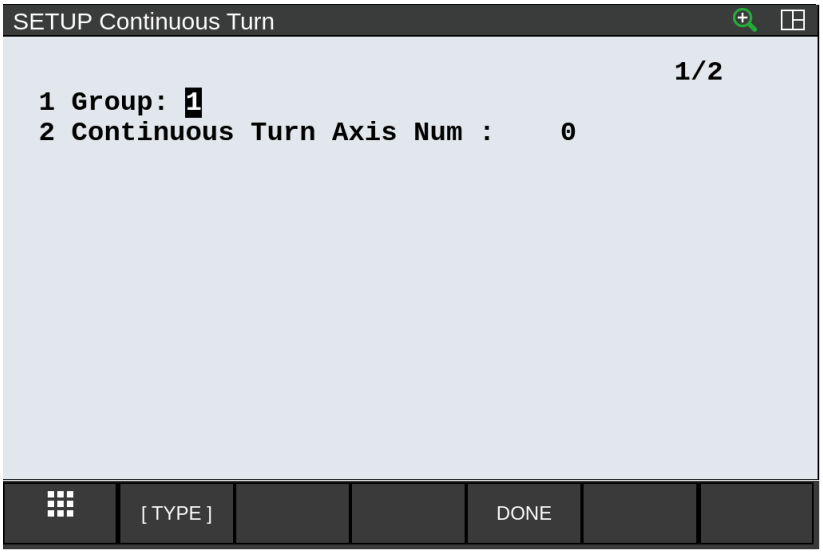
To use Continuous True:

- Perform setup on the SETUP Continuous Turn screen, and
- Specify the start/stop of Continuous Turn with the Continuous Turn Speed motion modifier.

Procedure 4-1 Setting up Continuous Turn

Step

1 Press MENU.
2 Select SETUP.
3 Press F1, [TYPE].
4 Select Cont Turn. The SETUP Continuous Turn screen will be displayed.



5 Specify the necessary items using the numeric and other keys.
- To disable Continuous Turn, set the Continuous Turn Axis Num item to 0.
- To set the motion group number, set the Group item to the motion group you want to use. When you change the value of the Group item to a different motion group, the screen will display the settings for that motion group.

6 When you are finished specifying the items, press F4, DONE.

7 Set up the Constant Path function. For more information about using Continuous Turn with the Constant Path function, refer to Section 4.4.2 Using Continuous Turn with Constant Path.
- If the Constant Path function needs to be disabled because of a limitation (such as when the Continuous Turn axis is set in the final axis of the robot), the Constant Path function will be disabled and the message “ConstPath is disabled.(Press ENTER)” will be displayed.
- If you try to disable Constant Path (such as when the Continuous Turn axis is set in an independent axis), “Disable Constant Path Function?” will be displayed.
  - If you press F2, YES, Constant Path will be disabled and “ConstPath is disabled. (Press ENTER)” will be displayed.
  - If you press F3, NO, the Constant Path setup will NOT be changed. If Constant Path is enabled, the message “ConstPath is enabled.(Press the [ENTER] key)” will be displayed. Otherwise, if Constant Path is disabled, the messeage “ConstPath is disabled. (Press [ENTER] key)” will be displayed.

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

9 Turn off the power, then turn it back on with a cold start. The first power-on after setting up the Continuous Turn axis is automatically a cold start. Subsequent power-ons will be performed using the selected start mode.

Table 4.2 (a) describes the items on the SETUP Continuous Turn screen.

Table 4.2 (a) SETUP Continuous Turn screen items

Item	Description
Group	This item allows you to set the motion group number.
Continuous Turn Axis Num	This item allows you to set the number of the Continuous Rurn axis. If you set this item to 0, Continuous Turn will be disabled for the motion group.

4.3 USING CONTINUOUS TURN

After setting up the Continuous Turn axis, specify the Continuous Turn start point using the Continuous Turn Speed motion modifier.

- Continuous turn speed instruction CTV

* where i = –100 to 100, which is the ratio of the rotation axis speed to the maximum axis speed (%)
* You can specify the Continuous Turn axis speed indirectly using a register. To use a register, press F1, REGISTER and enter a register number.
* Include the Continuous Turn Speed instruction in the same way as for other motion modifiers.

(→ Subsection 4.3.4, ”Teaching a Motion Modifier” in the *FANUC Robot series OPERATOR’S MANUAL (Basic Operation)* (B-84464EN))

4.3.1 Starting Continuous Turn

Continuous Turn is started as soon as a motion instruction with a Continuous Turn Speed instruction added is started.

4.3.2 Stopping Continuous Turn

Continuous Turn is stopped when the first motion instruction with no Continuous Turn Speed instruction added is started after a Continuous Turn Speed instruction was started.
When Continuous Turn is stopped, the motion of the other axes in the same motion group also stops. The robot, therefore, decelerates even if the termination type of the previous motion instruction is CNT.
In addition, the Continuous Turn axis starts decelerating and stops after the other axes come to a complete stop. At this time, the robot is not necessarily at the specified position on the Continuous Turn axis. Therefore, synchronization between the motion of the Continuous Turn axis with the other axes (including those for other motion groups) will be lost.
If a motion instruction is specified next, the axis rotates in the same direction as the previous continuous turn direction to move to the taught position.

4.3.3 Execution Notes

- Continuous Turn continues even if logic instructions (instructions other than those in motion instructions) are executed.
- During program playback, the turn number for the Continuous Turn axis is ignored, and is treated as 0.
- The turn number for the Continuous Turn axis at the position taught when Continuous Turn is enabled is always stored as 0.
- If 0 is specified as the rotation axis speed for a Continuous Turn Speed instruction, Continuous Turn motion is not performed. If a motion instruction is taught next, shorter-way motion is performed on the Continuous Turn axis. This feature is useful if you want to stop the Continuous Turn about the Continuous Turn axis once, but not stop the robot after it completes the Continuous Turn move. See the next section, ”Continuous Turn Speed instruction example”.
- In single-step execution (both forward and backward), Continuous Turn is not performed even if a Continuous Turn Speed instruction is added; shorter-way motion will be performed.
- Continuous Turn is stopped due to a hold. If program execution is subsequently resi,ed, if the target position has already been reached on axes other than the Continuous Turn axis, Continuous Turn is not performed. If the target position has not been reached on axes other than the Continuous Turn axis, Continuous Turn is resumed.
- Continuous Turn about a Continuous Turn axis can be performed while jogging.

4.3.4 Continuous Turn Speed instruction example

The following program shows an example of using the Continuous Turn Speed instruction.

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

- Lines 1 to 3:
  During motion from P[1] to P[2], Continuous Turn motion is executed. Although the termination type specified in line 2 is CNT, the robot decelerates (stops temporarily on all axes at the start of the move in line 3) because a Continuous Turn Speed instruction is not added to the next line, line 3.
- Lines 4 to 6:
  Continuous Turn starts as soon as the execution of line 4 starts. Because the rotation axis speed specified with the Continuous Turn Speed instruction in line 5 is 0, Continuous Turn stops temporarily at the start of line 5 execution. Because Continuous Turn continues, the termination type CNT100 in line 4 is valid and the robot does not decelerate.
  When line 6 is executed, a shorter-way move is executed on the Continuous Turn axis.
- Lines 7 to 9:
  The Continuous Turn move begins at the start of the move in line 7. Continuous Turn continues during the execution of the wait instruction (logic instruction) in line 8.
  The robot stops temporarily on all axes at the start of the move in line 9, and Continuous Turn stops.

4.4 NOTES AND LIMITATIONS

4.4.1 Functions that cannot be used together

- Continuous Turn cannot be used together with the following functions:
  - TAST
  - Weaving
  - TCP Speed Prediction function (sealing flow rate control)
  - TCP Speed Output
  - Coordinated Motion (You can use Continuous Turn with Coordinated Motion if you set up the Continuous Turn axis in a motion group without coordinated motion and use programs with a separate group mask as described in the section "Using of Continuous Turn with Constant Path" that follows.
  - Interference Prevention Area function
- When Continuous Turn is to be performed on a robot axis or integrated additional axis, the following functions of the Dual Check Safety function are not available:
  - Joint Position Check function for Continuous Turn axes.
  - Cartesian Position Check function.
  - Cartesian Speed Check function.
- The following instructions or functions cannot be used for Continuous Turn axes.
  - Asynchronous additional axis speed instruction. Synchronous additional axis speed instructions can be used.
  - Incremental instruction.
  - Dual Drive function.

4.4.2 Useing Continuous Turn with Constant Path

- The Continuous Turn axis group cannot include the Constant Path function.
- If a Continuous Turn axis is set for an independent axis and group mask is separated from a non-Continuous Turn motion group, Constant Path can be used only the non-Continuous Turn group.
- For example, the following programs show how Continuous Turn can be used with Constant Path.

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

- In a multi-group program, if a Continuous Turn axis exists in any of the groups, the alarm "CPMO-004 Feature not Supported (G:i)" will be posted at execution and the program will be aborted. To use Continuous Turn in a program, separate the groups that have a Continuous Turn axis from groups that do not have a Continuous Turn axis, or disable the Constant Path function.
- If a Continuous Turn axis is set in a robot final axis, the Constant Path function will be disabled.
- If the Constant Path function settings are changed by the Continuous Turn settings, the original Constant Path function state will be restored after the Continuous Turn setup is cleared.

4.4.3 Other notes

- When using Continuous Turn on a robot axis or integrated additional axis, the X and Y axis components of the tool coordinate system must both be 0. Only the Z axis component can have a non-zero value.
  If axes other than the Z axis have non-zero values, the linear, circular or circle arc path cannot be guaranteed in normal motion other than Continuous Turn motion.
- This function automatically updates mastering data (for the Continuous Turn axis only) according to the amount of rotation about the Continuous Turn axis. Therefore, previously recorded mastering data may not match the current mastering data.
  After disabling Continuous Turn, you do not need to re-master the system.
- When Continuous Turn is disabled, the current position of the Continuous Turn axis may fall outside the stroke limits. If this occurs, move the position of the Continuous Turn axis to within the stroke limits by jogging or within a program.
- In a multi-group system, even if there are multiple Continuous Turn axes, you cannot specify separate Continuous Turn Speeds for each of the axes.
- At the end of a Continuous Turn, the Continuous Turn axis may rotate one or more revolutions to ensure that the robot decelerates smoothly to a stop. The amount of rotation depends on the acceleration/deceleration constant.
- Even during backward execution (single-step execution), the Continuous Turn axis will move using the shorter-way motion method. If a forward step and backward step are executed consecutively for a motion instruction movement that is very close to 180° , the Continuous Turn axis may rotate in the same direction during the forward and backward execution.
- The Original Path Resume feature is disabled when any group has Continuous Turn axes. Even if the Original Path Resume feature is enabled, it will be disabled when you cycle power.
- Groups that have Continuous Turn axes will disable the Cycle Time Priority feature for Cartesian motion. This might result in longer cycle time and changes to the path. When the Continuous Turn axis is removed from the motion group, the Cycle Time Priority feature for Cartesian motion will be

restored to its original state. Execute the program carefully after changing the Continuous Turn axis setup.

- If the motion speed of the Continuous Turn axis is very fast, the warning "CNTR-009 Warn-Cont Vel too high(G:group)" will be posted. If such motion is executed in single-step mode, the alarm "CNTR-011 Axis speed exceeds lim(G:group)" will be posted.
- If a Continuous Rotation axis is set for a SR series axis, the position of the interaction axis may be incorrect after restoring a backup. In this case, you need to re-master the robot.

4.5 RESETTING THE ROTATION OF POSITIONER AND INDEPENDENT AXES USING A KAREL PROGRAM

You can use the CALIBNBT.PC KAREL program to reset the number of rotations of positioners and independent axes.
Using the CALIBNBT.PC KAREL program, you can reset the position of a rotational axis to between -180 deg and +180 deg. You can also rotate the axis in one direction continually without having to set up a Continuous Turn axis.
You can add 3rd argument to reset to the range of "[specified rotation] +/- 180 deg" instead.
You can also avoid the limitations imposed by the Continuous Turn axis listed in Section 4.4.
For example, if you have a system of coordinated motion with a robot and a positioner and you rotate the positioner in one direction continually, use CALIBNBT.PC instead, because the Continuous Turn function is not available in this case.

4.5.1 CALIBNBT.PC Conditions

You can use CALIBNBT.PC if all of the following conditions are met,

- The target axis does not belong to a robot group, such as a positioner or an independent axis.
- Mastering has already been done before using CALIBNBT.PC.
- Please execute CALIBNBT.PC **only while robot is not moving**. If you need to execute CALIBNBT.PC just after a motion line in a TP program, the term-type of that motion should be FINE. (Do not execute during CNT motion)
- The absolute values of the upper and lower stroke limits of the axis that will be reset is smaller than the value expressed in the following formula:

$$\lceil 4085 \times \frac{Gear\ Ratio\ (axis)}{Gear\ Ratio\ (motor)} \times 360 \rceil$$

motor rotation limit: angle of one rotation [deg]

Example: Suppose Gear Ratio (axis) / Gear Ratio (motor) = 0.01
4085*0.01*360 = 14706 [deg]
→ axis limit must be bigger than -14706[deg] and less than 14706[deg].

NOTE

Resetting position by CALIBNBT.PC is not recommended when the current position is very near 180 + 360*N deg (N: arbitrary integer).
The position after reset can be either near +180 deg or near -180 deg depending on minute deviations from exact 180 + 360*N deg.

Even if you teach the position to be 180.000 deg while making programs, there are the minute deviations from exact 180.000 deg.

4.5.2 Using CALIBNBT.PC

To use this program, call CALIBNBT.PC in a TP program as follows:

- Press the MENU → System → Config, then select Program settings → KAREL Program settings, then set KAREL Prog in the select menu to TRUE, to display KAREL in the program list menu.
- Include a CALL instruction in the TP program to execute CALIBNBT.PC.
- Add arguments 1 and 2, or all three arguments as below to call CALIBNBT.PC
  CALIBNBT (group number, axis number)
  CALIBNBT (group number, axis number, rotation number)
  Argument 1 : group number, 1 to 8
  Argument 2 : axis number, 1 to 9
  Argument 3 : the number of rotation to reset to (please refer to 4.5.3)
  ex) Reset the number of rotation of group 2 / axis 1.
  CALL CALIBNBT (2, 1)
  (CALL CALIBNBT (2, 1, 0) is the same meaning.)
- If the target axis is a dual drive axis, specify the axis number of the master group.

4.5.3 Reset to a specified range

You can add argument 3 to CALIBNBT.PC to reset the axis position to the range of "[argument 3]*360 +/- 180 deg".
For example, assume that you want to continuously rotate an axis in the positive direction. You will have a longer stroke in each cycle by resetting the axis position with a negative value in the argument 3 than resetting the position to 0 rotation (range between +/- 180deg) without argument 3.
Specifying 0 for argument 3 means same as specifying only arguments 1 and 2.

e.g.
When you call CALIBNBT.PC with the current axis position = 361 deg,
the position is reset to "1" deg with argument 3 := 0,
the position remains "361" deg with argument 3 := 1,
the position is reset to "721" deg with argument 3 := 2.

Table 4.5.3 The example of the argument 3 and Axis range after reset.

Argument 3	Axis range after reset [deg]		
...		...	
-1	-540	-	-180
0	-180	-	180
1	180	-	540
2	540	-	900
...		...	

NOTE
As described in the NOTE of 4.5.1, reset from the position very near 180 + 360*N deg (N: arbitrary integer) is not recommended, because the result can be indefinite.

You have to set the argument 3 to make the absolute values of Axis range after reset satisfy the condition described in the expression in 4.5.1.

e.g.
When gear ratio (axis) / gear ratio (motor) = 0.01, axis range after reset has to be in the range of +/- 14706 deg as calculated in 4.5.1.
When the argument 3 is "40", Axis range after reset is 14220-14580 deg. This is IN the range of +/- 14706 deg, and CALIBNBT.PC can reset axis position.
When the argument 3 is "41", Axis range after reset is 14580-14940 deg. This is OUT of the range of +/- 14706 deg, and CALIBNBT.PC fails to reset axis position with an error message 'Rotation angle is too large'.

This condition of valid value of argument 3 is expressed as follows.

$$|\text{Argument } 3| + 0.5 < 4085 \times \frac{gear\ ratio\ (axis)}{gear\ ratio\ (motor)}$$

4.5.4 Execution Notes

- Execute CALIBNBT.PC only while the robot is not moving. If you need to execute CALIBNBT.PC just after a motion instruction in a TP program, the termination type of that motion should be FINE. Do not execute during CNT motion.
- When calling the CALIBNBT.PC program, the number of rotations is reset so that the updated position of the target axis will be between -180 deg and +180 deg.
- If CALIBNBT.PC is executed when the position of the target axis is between -180 deg and +180 deg, it does nothing.
- When the DCS Position/Speed Check function is enabled, you cannot use CALIBNBT.PC. If you call CALIBNBT.PC when the DCS Position/Speed Check function is enabled, the alarm "SYST-212 Need to apply to DCS param" will occur. If you need to use CALIBNBT.PC, please disable DCS Position/Speed Check for that motion group.
- Resetting position by CALIBNBT.PC is not recommended when the current position is very near 180 + 360*N deg (N: arbitrary integer).

4.5.5 Error messages

When following messages are displayed, please check the remedy for each case.

- 'Argument missing (1: Group number)'
  Argument 1 is not assigned. Assign group number of from 1 to 8.
- 'Illegal parameter (1: Group number)'
  Argument 1 is not integer. Check Argument 1.
- 'Argument missing (2: Axis number)'
  Argument 2 is not assigned. Assign axis number of from 1 to 9.
- 'Illegal parameter (2: Axis number)'
  Argument 2 is not an integer. Check Argument 2.
- 'GETTING VARIABLES ERROR -1-'
- 'GETTING VARIABLES ERROR -2-'
- 'GETTING VARIABLES ERROR -3-'
- 'GETTING VARIABLES ERROR -4-'
- 'SETTING VARIABLES FAIL'
- 'Illegal Argument'
  Argument 1 or Argument 2 is an illegal number. Check the arguments.
- 'Axis type is not supported'
  Axis type is not a rotary axis. Assign a rotary axis to use CALIBNBT.PC.
- 'Robot axis is not supported'
  A robot axis or extended axis in a robot group is assigned.
  Assign an axis that is independent of the robot group.
- 'Find slave axis fail'

Dual drive axes are assigned but slave axis was not found. Check the dual drive settings.

- 'Current angle is too large'
  The current angle of the axis is too large. Confirm the upper and lower limit values of the axis range.
- 'Rotation angle is too large'
  The argument 3 is too large. Modify the value of argument 3 so that it satisfies the condition described in 4.5.3.
- 'Do not execute during moving '
  KAREL was executed while the robot was moving.
  Execute KAREL after the completion of a FINE termination type motion.
- ‘CNTR-28 Cannot use ContTurn axis(CALIBNBT)’
  A Continuous Turn axis cannot be specified as an the argument in CALIBNBT.PC.

5 OPERATION GROUP DO OUTPUT FUNCTION

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

Function

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

- Jog signals

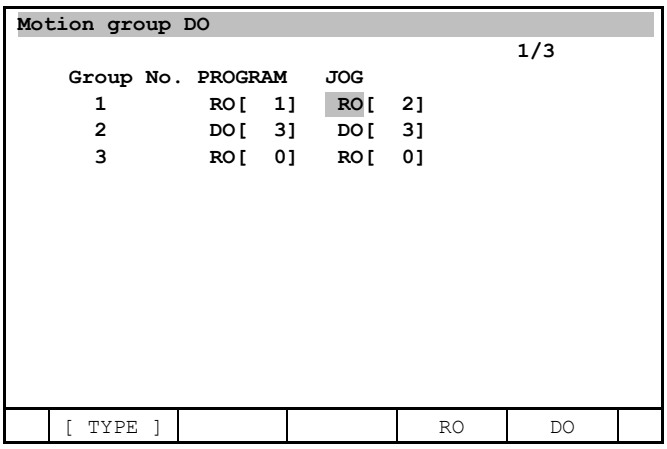
When the teach pendant is disabled, all signals turn off.
When the teach pendant is enabled, the signal for the currently selected operation group on the teach pendant turns on, while the other signals turn off.

- Program signals

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

Setup

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



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

Example of using this function with the multitask option

This section explains the operation of this function when a subprogram call or the multitask option is used.
The output program signal is the OR of the signals for all the operation groups of the program currently being executed or temporarily stopped.
If a program without an operation group calls a program having an operation group by using a subprogram call, the signal for the operation group of the subprogram turns on only while the subprogram is being executed. (The signal does not turn on when the main program without an operation group is merely selected/executed.)
If the execution instruction of the multitask function is to start another program that operates the robot (the main program that has the execution instruction does not have an operation group), the signal for the operation group of the program started by the execution instruction does not turn on when the main program is merely selected/executed. The program signal turns on when the program that operates the robot is actually started.
Consider the following three example programs:

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

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

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

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

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

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

6 AUTOMATIC ERROR RECOVERY FUNCTION

To use this function, the Automatic Error Recovery function option (A05B-2700-J924) is required.

6.1 AUTOMATIC ERROR RECOVERY FUNCTION OVERVIEW

During production, the robot may stop due to various alarms. When a robot is stopped, it is necessary to perform a recovery operation and then resume the originally running program.
For example, if the robot is arc welding, an alarm such as an arc start failure may occur, and the robot may stop. In this case, you must jog the robot to a safe position to cut the end of the wire or clean the nozzle, and then resume the original program.
The Automatic Error Recovery function performs this kind of recovery sequence automatically.

The Automatic Error Recovery function consists of the following functions:
- Resume program function
- Fast exit/entry feature

6.2 RESUME PROGRAM FUNCTION

If the robot is stopped due to an alarm during production, the Resume program function removes the cause of the alarm using a resume program, and resumes the original program.

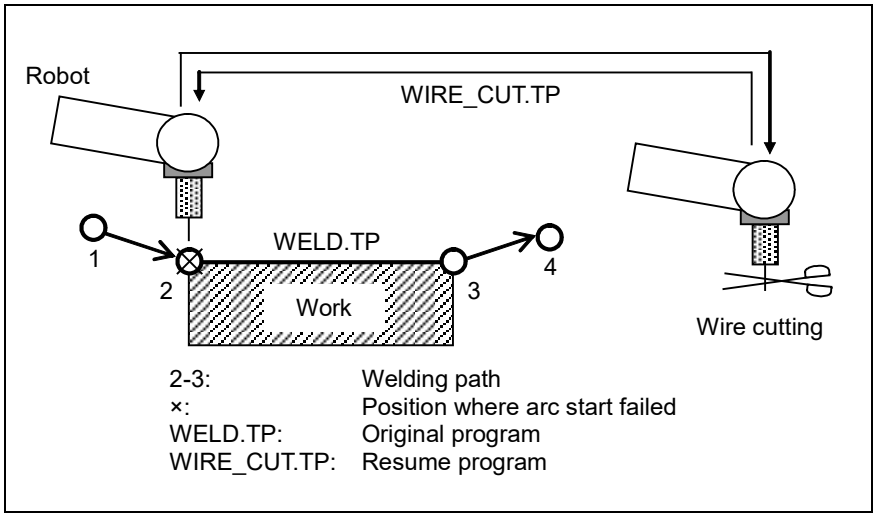


Fig.6.2 Resume program function

In the example shown in Fig 6.2, the robot executes WELD.TP to perform welding along a path from 2 to 3. If an arc start failure occurs at arc start position 2, if the resume program function is used, another program called the resume program, which is WIRE_CUT.TP in this case, can be started at the next start signal input. After this program finishes, another start signal input resumes the original program. If the resume operation function is then enabled (on the welding system setting screen), the robot automatically returns to the original pause position and then resumes the original program. If the resume motion return distance is set, the robot returns from the pause position by the set distance, and then resumes the original program. If no arc occurs, a scratch start is performed.

6.3 FAST EXIT/ENTRY FEATURE

If an alarm is issued during motion and the robot tries to execute the resume program in a complex environment, there is a risk of interference with the workpiece or surrounding equipment when the robot moves from the paused position to the resume program taught position. The Fast exit/entry feature helps prevent this kind of interference.

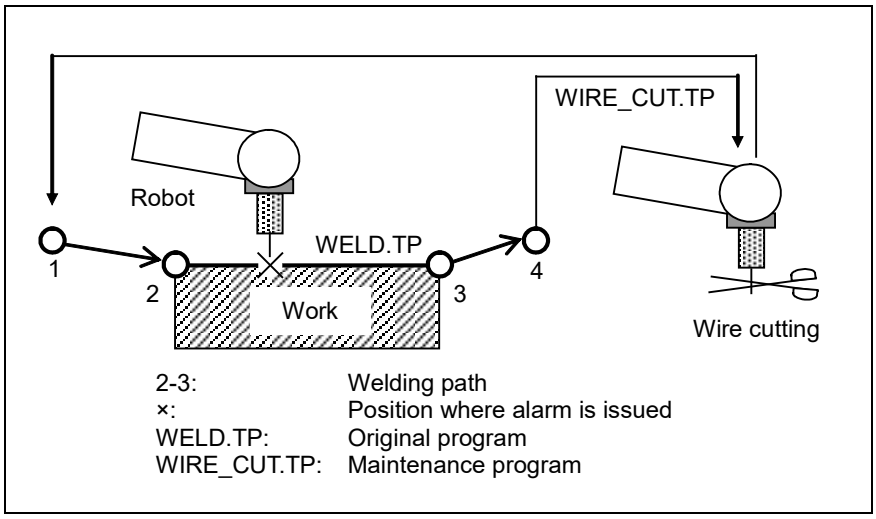


Fig.6.3 Fast exit/entry feature

In the example shown in Fig 6.3, the robot executes WELD.TP to weld along the path from 2 to 3. At this time, if any alarm occurs between positions 2-3 and the robot pauses, the Fast exit/entry feature automatically performs the *Fast Fault Recovery (FFR) sequence*, which is as follows:

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

6.4 RESUME_PROG INSTRUCTION

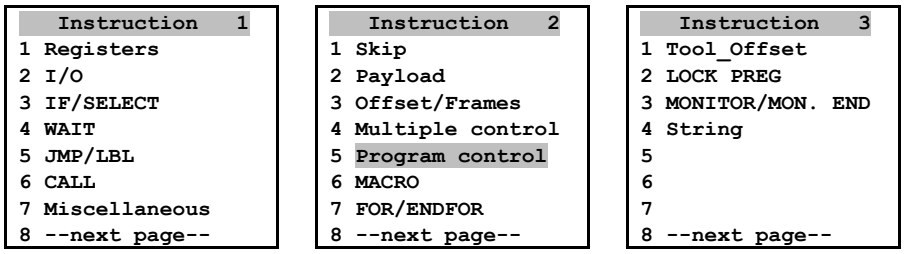
The Resume program function executes a resume program defined in the original program, instead of the original program. Use the following resume program instructions to define the resume program:

- RESUME_PROG [1: Comment] : For single-tasking or multi-tasking
- RESUME_PROG [2: Comment] : For multi-tasking only
- RESUME_PROG [3: Comment] : For multi-tasking only
- RESUME_PROG [4: Comment] : For multi-tasking only
- RESUME_PROG [5: Comment] : For multi-tasking only

To clear a defined resume program, use the following instruction:

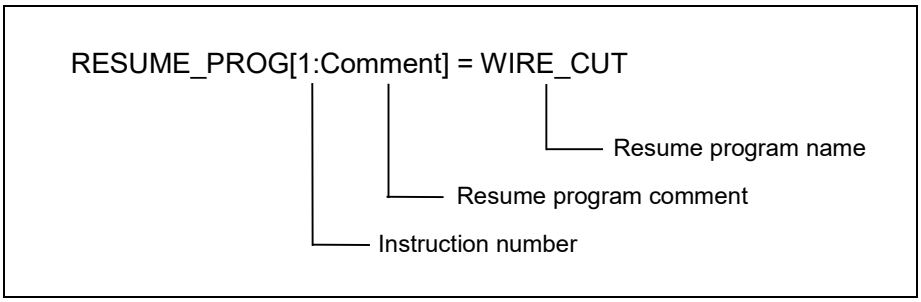
- CLEAR_RESUME_PROG

To teach the resume program instructions, press F1, [INST] and select Program control. The resume program name defined for each resume program instruction can be confirmed with the following Data ID.
(Data ID: $PROGRAM_EXECUTION.RESUME_PROG[a] / STRING / RO / a: Instruction number)



6.4.1 RESUME_PROG [1: Comment]

When the RESUME_PROG [1:Comment] instruction is executed in the original program, the program name defined by the RESUME_PROG instruction will be executed. If an alarm occurs while you are defining the resume program, the resume program will be executed.



The following options are required to use the resume program instructions:

- To use RESUME_PROG [1-5], the Automatic Error Recovery function (A05B-2700-J924) is required.
- To use RESUME_PROG [2-5], the following options are required in addition to the above option:

RESUME_PROG [2]

Dual Arm Control	A05B-2700-J605

RESUME_PROG [2-5]

Multi UOP interface function	A05B-2700-J964

When teaching the index number of a RESUME_PROG [ ] instruction in multi-tasking, the following conditions must be met:

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

For example, in multi-tasking, the RESUME_PROG [3: Comment] instruction can be used only in programs that can be started with the START#3/ PROD_STAR#3 UOP signals.
The RESUME_PROG [2-5] instructions cannot be used in single-tasking because they are for multi-tasking. Therefore, use the RESUME_PROG [1: Comment] instruction in single-tasking systems.

Keep in mind the following conditions when defining resume program instructions:

- Multiple RESUME_PROG [ ] instructions cannot be defined in the original program.
- RESUME_PROG [ ] instructions cannot be defined in the resume or maintenance program.

- Multiple RESUME_PROG [] instructions with the same instruction number cannot be executed simultaneously. That is, while the resume program is defined by the RESUME_PROG [2] instruction, the RESUME_PROG [2] instruction cannot be executed by another program. However, a RESUME_PROG [] instruction other than the RESUME_PROG [2] instruction can be executed.
- Resume program instructions cannot be defined in programs executed by RUN instructions.

6.4.2 CLEAR_RESUME_PROG

When the CLEAR_RESUME_PROG instruction is executed, the resume program defined in the RESUME_PROG [] instruction is deleted. Therefore, even if an alarm occurs after executing a CLEAR_RESUME_PROG instruction, the resume program will not be executed. In addition, the CLEAR_RESUME_PROG instruction is valid only within the program in which the CLEAR_RESUME_PROG instruction is used. For example, in a multi-tasking system, when a CLEAR_RESUME_PROG instruction is executed in a program in which a RESUME_PROG [1: comment] instruction is defined, only the resume program defined in the RESUME_PROG [1: comment] instruction will be cleared. The resume programs defined by other RESUME_PROG [2] - [4] instructions will not be cleared.

In addition to the CLEAR_RESUME_PROG instruction, a defined resume program will also be cleared in the following cases.

- When backward execution is performed in the original program.
- When you move the cursor to a different line in the original program.
- When the original program is terminated.

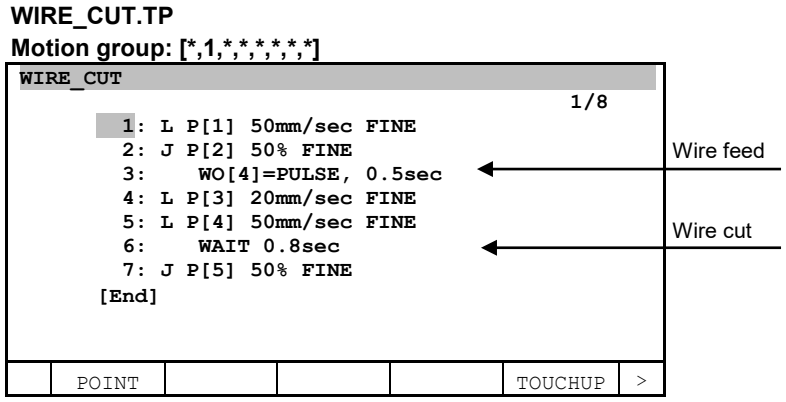
> **⚠ CAUTION**
> If the resume program is cleared while the resume program is paused, the resume program cannot be resumed. The original program will resume from the paused position of the resume program.

The following program is a program example of using the RESUME_PROG [2: comment] and the CLEAR_RESUME_PROG instructions.

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

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

   | POINT |  |  |  | TOUCHUP | >
```

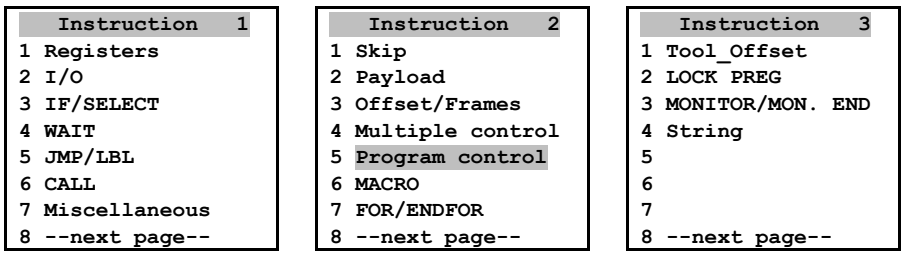


In the above program example, WIRE_CUT.TP is taught in line 2 of WELD.TP and is cleared in line 5. Since WIRE_CUT.TP is defined as the resume program between line 3 and line 4, it is executed as the resume program. In line 6 and subsequent lines, the resume program has been cleared, so the resume program will not be executed.

6.5 RETURN_PATH_DSBL INSTRUCTION

ArcTool systems normally operate with the return path function enabled. When the return path function is enabled, if the original program is resumed after the recovery program has finished executing, the robot always returns to the original pause position. However, some systems do not allow the robot to return to the original pause position. For example, when the nozzle touch state is input through a DI, a recovery program is used to move the torch slightly in the torch direction. During return path operation, the robot will return to the original pause position even if a recovery operation has been performed, and the nozzle can make contact again. In this case, the return path function should be enabled, but only after the resume program has been executed.
You can enable and disable whether the robot uses the return path function using the RETURN_PATH_DSBL instruction. When this instruction is executed in the resume program, the return path function will be disabled only when the next time the original program is resumed. The RETURN_PATH_DSBL instruction is valid only when it is executed within a resume program; it is invalid when executed in any other kind of program.

RETURN_PATH_DSBL instruction is on the same menu as the RESUME_PROG [ ] instruction.



6.5.1 RETURN_PATH_DSBL

The RETURN_PATH_DSBL instruction is valid only within the resume program where the RETURN_PATH_DSBL instruction is defined. For example, in a multi-tasking system, if the RETURN_PATH_DSBL instruction is executed in the resume program defined in the RESUME_PROG [1: comment] instruction, only the return path function of the original program for which the RESUME_PROG [1: comment] instruction is defined will be invalid. The return path function of the original program for which other RESUME_PROG [2] - [5] instructions are defined remains valid.

The following example program uses the RETURN_PATH_DSBL instruction. When defining the RETURN_PATH_DSBL instruction, define it in the resume program as shown in this example. If taught like this, even when the original program is resumed after the resume program is terminated, the return path function will not work even if the return path function itself is enabled in a welding system setting.

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

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

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

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

Wire feed

Wire cut

6.6 MAINT_PROG INSTRUCTION

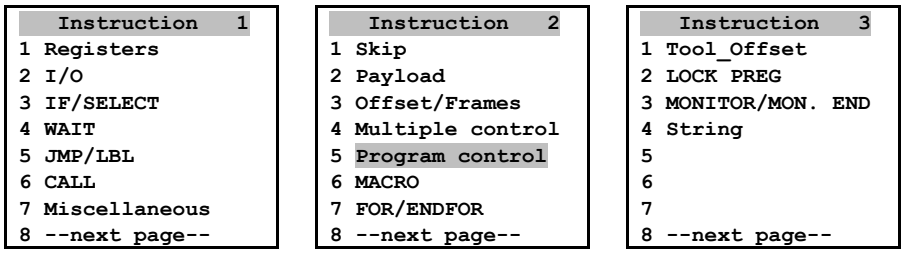
The fast exit/entry feature automatically performs the following operations, called the Fast Fault Recovery (FFR) sequence:

1 From the paused position, disables arc welding, and executes only the motion instructions of the original program to the end.
2 Executes a maintenance program.
3 Disables arc welding, and executes the motion instructions of the original program from the beginning to move the robot to the paused position.
4 Enables arc welding and resumes the original program execution.

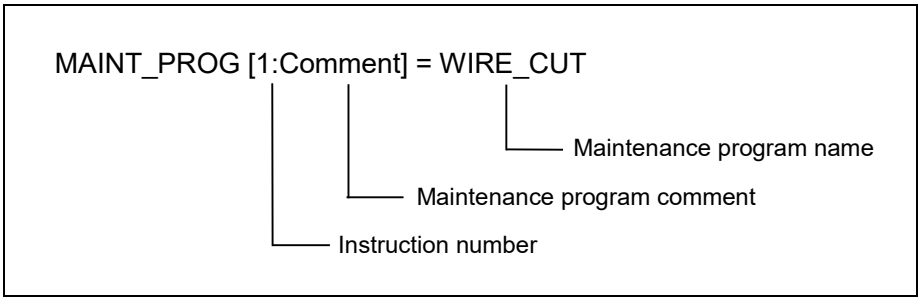
There are five maintenance program instructions for defining maintenance programs. Use MAINT_PROG [1] for single-tasking or multi-tasking, and MAINT_PROG [2] - [5]for multi-tasking

- MAINT_PROG [1: Comment] : For single-tasking or multi-tasking
- MAINT_PROG [2: Comment] : For multi-tasking only
- MAINT_PROG [3: Comment] : For multi-tasking only
- MAINT_PROG [4: Comment] : For multi-tasking only
- MAINT_PROG [5: Comment] : For multi-tasking only

The MAINT_PROG [ ] instruction appears in the same menu as the RESUME_PROG [ ] instruction.



6.6.1 MAINT_PROG [1:Comment]



When the MAINT_PROG [1: Comment] instruction is executed, the program name defined in the instruction is defined as a maintenance program. If an alarm occurs while defining the maintenance program, the FFR sequence is executed.

The following options are required to use maintenance program instructions:

- To use the MAINT_PROG [1] instruction, the Automatic Error Recovery function (A05B-2700-J924) is required.
- To use the MAINT_PROG [2]-[5] instructions, the following options are required:

MAINT_PROG [2]

Dual Arm Control	A05B-2700-J605

MAINT_PROG [2]-[5]

Multi UOP interface function	A05B-2700-J964

In multi-tasking systems, teach the instruction index number in [ ] in a MAINT_PROG [ ] instruction as follows:

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

For example, in multi-tasking, the MAINT_PROG [3: Comment] instruction can be used only in a program that can be started by the START#3/ PROD_STAR#3 UOP signal.
The MAINT_PROG [2]-[5] instructions cannot be used in single-tasking because they are used exclusively for multi-tasking. Instead, use the MAINT_PROG [1: Comment] instruction in single-tasking programs.

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

MOTN-431 Fast fault recovery 1	The FFR sequence executed by the MAINT_PROG [1]

	instruction is completed.
MOTN -432 Fast fault recovery 2	The FFR sequence executed by the MAINT_PROG [2] instruction is completed.
MOTN -433 Fast fault recovery 3	The FFR sequence executed by the MAINT_PROG [3] instruction is completed.
MOTN -434 Fast fault recovery 4	The FFR sequence executed by the MAINT_PROG [4] instruction is completed.
MOTN -435 Fast fault recovery 5	The FFR sequence executed by the MAINT_PROG [5] instruction is completed.

When you teach maintenance program instructions, keep in mind the following restrictions:

- Multiple maintenance program instructions cannot be defined in the original program.
- Maintenance program instructions cannot be defined in the resume or maintenance program.
- A maintenance program instruction with the same instruction number cannot be executed by another program. For example, the MAINT_PROG [5] instruction cannot be executed by other programs while the maintenance program has already been defined by the MAINT_PROG [5] instruction. However, MAINT_PROG [1]-[4] instructions can be executed by other programs.
- Maintenance program instructions cannot be defined for programs executed by RUN instructions.

Maintenance program definitions are deleted when:

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

> ⚠ **CAUTION**
> If the maintenance program definition is deleted while executing the FFR sequence, the maintenance program or original program cannot be restarted by the FFR sequence.

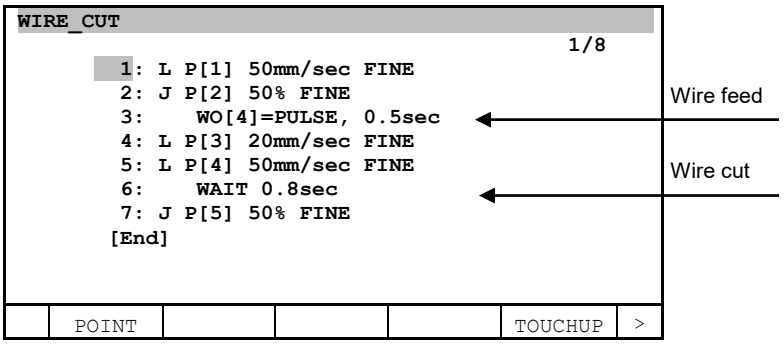
The following programs are examples that use the MAINT_PROG [2] instruction.

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

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

    POINT                                  TOUCHUP   >
```

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



In the above program example, the WIRE_CUT.TP program is taught in line 2 of the WELD.TP program. Since WIRE_CUT.TP is defined as the maintenance program from line 3 to line 7, it is executed as the maintenance program. When the execution of WELD.TP ends, the definition of the maintenance program is automatically deleted.

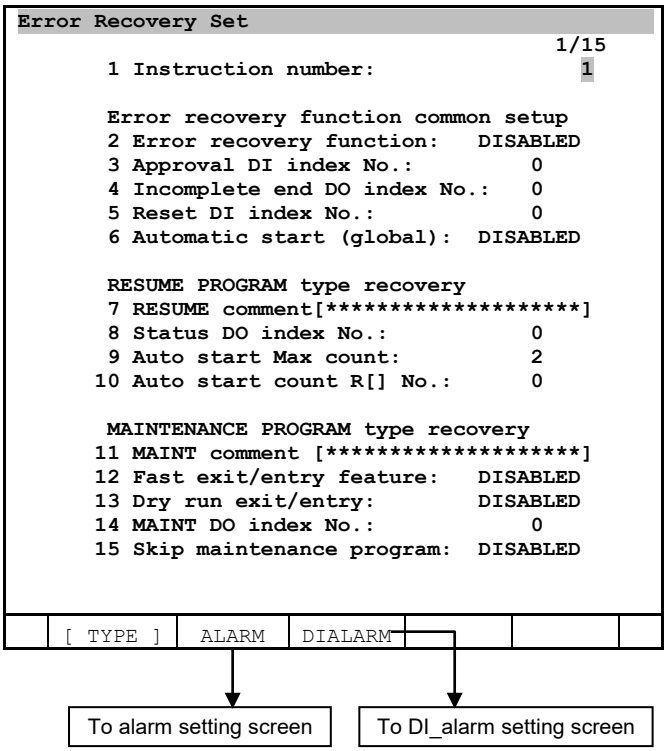
6.7 SETTING UP THE AUTOMATIC ERROR RECOVERY FUNCTION

The Automatic Error Recovery function setting screen can be displayed by following the steps below.

Procedure 6-1 Displaying the Automatic Error Recovery function setting screen

Steps

1 Press the [MENU] key. The menu will be displayed.
2 Select SETUP.
3 Press the F1, [TYPE].
4 Select Err recovery. The Automatic Error Recovery function setting screen will be displayed.



Selecting an instruction number

You can display the setup screen for each instruction number of the resume program instruction and maintenance program instruction by specifying the instruction number in the Instruction number item.

For multi-tasking

For example, in a system that uses RESUME_PROG [1-3] and MAINT_PROG[1-3], you can select instruction number 1 through 3 in the Instruction number item on this screen.

6.7.1 Automatic Error recovery function common setup

Error recovery function

This item allows you to enable and disable the Automatic Error Recovery function. When the Automatic Error Recovery function is enabled, and neither monitored alarm codes nor the Error recovery approval DI have been defined, the resume program will always resume execution from the paused position, except for when the Status DO is off. In addition, when the Automatic Error Recovery function and Fast exit/entry feature are enabled, and neither monitored alarm codes nor the Approval DI have been defined, the FFR sequence will always be executed when resuming from the paused position, except for when the FFR sequence information DO is off.

When the Error Recovery function item is disabled, the resume program and the FFR sequence will not be executed.

For multi-tasking

This item can be set for each instruction number.

You can also change or confirm the setting via Data ID.

(Data ID: $ERROR_RECOVERY.ERROR_RECOVERY_ENABLE[a] / BOOLEAN / RW / a: Instruction number 1-2 / TRUE: Enable, FALSE: Disable)

Approval DI index number

To use the Approval DI function, define a DI number. After you define the number, cycle controller power for the setting to take effect.

With this function, you can choose whether to execute the resume program or the FFR sequence when resuming from a paused state using a peripheral device. When the Approval DI is OFF, neither the resume program nor the FFR sequence will be executed. If the Approval DI signal number is not defined, this function is disabled.

The specifications of the Approval DI function are listed in Table 6.7.

Table 6.7 Specifications of the Approval DI function

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

⚠ CAUTION

To continue to execute the resume program even when the program is restarted after the resume program has been paused, set the Approval DI to ON. If it is OFF, the original program will be executed.

For multi-tasking

This item can be set for each instruction number.

Incomplete end DO index number

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

Before inputting the start signal, you must check the Incomplete end DO signal status. If the Incomplete end DO signal is ON, the resume or maintenance program will terminate in the middle, so the robot will not have reached the specified position.

⚠ WARNING

If a resume or maintenance program terminates in the middle of the program, and the start signal is input, the robot might behave unpredictably. Therefore, before inputting the start signal, check the program name and the program start line that will be executed with the next start signal. In addition, if an obstacle exists, jog the robot to a position that is clear of obstacles, and then input the start signal.

The Incomplete end DO signal can be added to the PLC start signal acceptance conditions.
If this signal is set to 0, the function is disabled.

For multi-tasking

This item can be set for each instruction number.

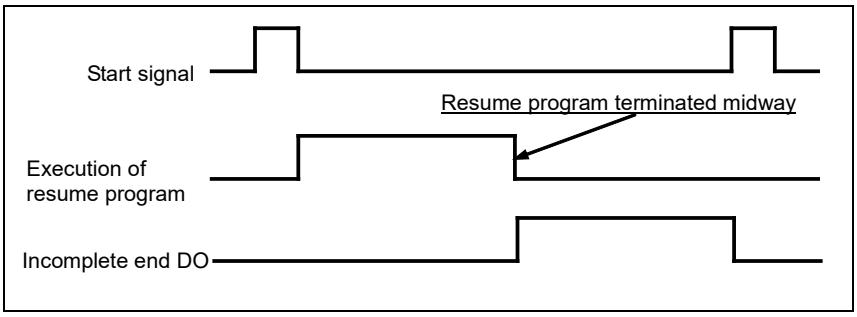


Fig. 6.7 (a) Incomplete end DO output timing chart

Reset DI index number
When the incomplete end DO is included in the PLC start signal acceptance conditions, you need a way to turn off the Incomplete end DO externally. Inputting the Reset DI signal turns off the Incomplete end DO. When the Incomplete end DO is output, you must first check the program name and the program start line that will be executed with the next start signal. In addition, if an obstacle exists, jog the robot to a position that is clear of obstacles, input the Reset DI, and then input the start signal.
If this signal is set to 0, this function is disabled.

For multi-tasking
This item can be set for each instruction number.

Automatic start feature
When this item is enabled, and an alarm defined in the monitored alarm codes is issued, the resume program or FFR sequence are automatically executed without outputting the alarm signal and stopping the robot. When the resume program has terminated, the original program is resumed automatically by this function. For the Fast exit/entry feature, when the maintenance program has terminated, the original program is resumed automatically by the FFR sequence. Therefore, you do not need to input the start signal during this time.
Because the alarm signal is not output, other robots will not be stopped, even if multiple robots are operating. The robot for which the alarm was issued moves by itself to the recovery station, and after recovery work, the original program is resumed.
When this item is disabled, and an alarm defined in the monitored alarm codes is issued, the program outputs an alarm signal and stops running. Input of the start signal executes the defined resume program or FFR sequence. In the resume program function, after the resume program terminates, another start signal input restarts the paused original program. For the Fast exit/entry feature, it is not necessary to re-enter another start signal to resume the original program. The FFR sequence automatically resumes the original program.

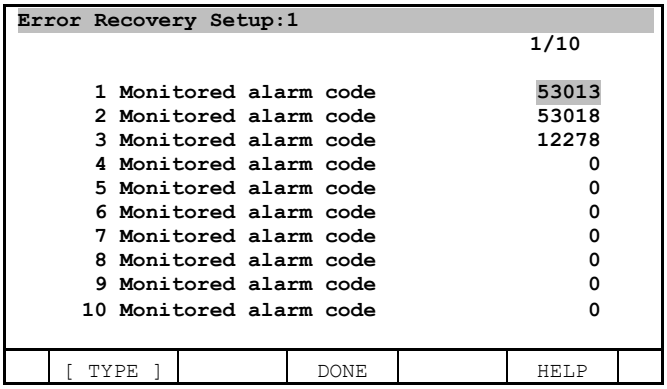
For multi-tasking
- This item cannot be set for each instruction number. When this item is enabled for any instruction number, the Automatic start function will be automatically enabled for all of the instruction numbers.
- Even if this function is enabled, this function will not be executed when an alarm defined in the monitored alarm codes is issued while the alarm is being issued in other robots. In this case, the program outputs an alarm signal and pauses. Then, when a start signal is input, the defined resume program or FFR sequence is executed.

You can also change or confirm the setting via Data ID
(Data ID: $ERROR_RECOVERY.AUTOMATIC_START_GLOBAL[a] / BOOLEAN / RW / a: Instruction number 1-2 / TRUE: Enable, FALSE: Disable)

Alarm code monitoring function

When this function is enabled, the resume program or FFR sequence is executed only when an alarm defined in the monitored alarm codes is issued. The alarm code monitoring function is automatically enabled when at least one monitored alarm code is defined and a resume or maintenance program is defined. If no alarm code has been defined, this function will be disabled. In this case, before a paused original program is resumed, the resume program or the FFR sequence will always be executed, except when the Approval DI function is enabled.

To define the alarm codes to be monitored, press F2, ALARM to display a screen for defining alarm codes.



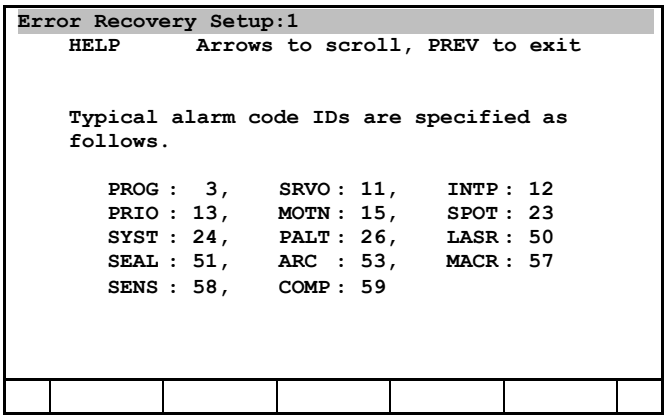
When a defined alarm code is issued, and a program is paused, the resume program or FFR sequence is executed at restart.

Each alarm code consists of an alarm code ID and alarm number. The alarm code ID indicates the type of alarm. For an arc start failure alarm, for example, the following alarm code is used:

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

For alarm numbers, refer to FANUC Robot series OPERATOR'S MANUAL (Alarm Code List).

Press F5, HELP to display the following screen:



The alarm code monitoring function is disabled when any of the following conditions are true:

- The recovery switch DI is off.
- No alarm code have been defined.
- A resume or maintenance program has not been defined in the currently running program.
- A resume program or FFR sequence is running.

The specifications of the alarm code monitoring function are as follows.

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

⚠ CAUTION
Defined alarms must be pause alarms. Do not define warning alarms as alarm codes.

For multi-tasking
This item can be set for each instruction number.

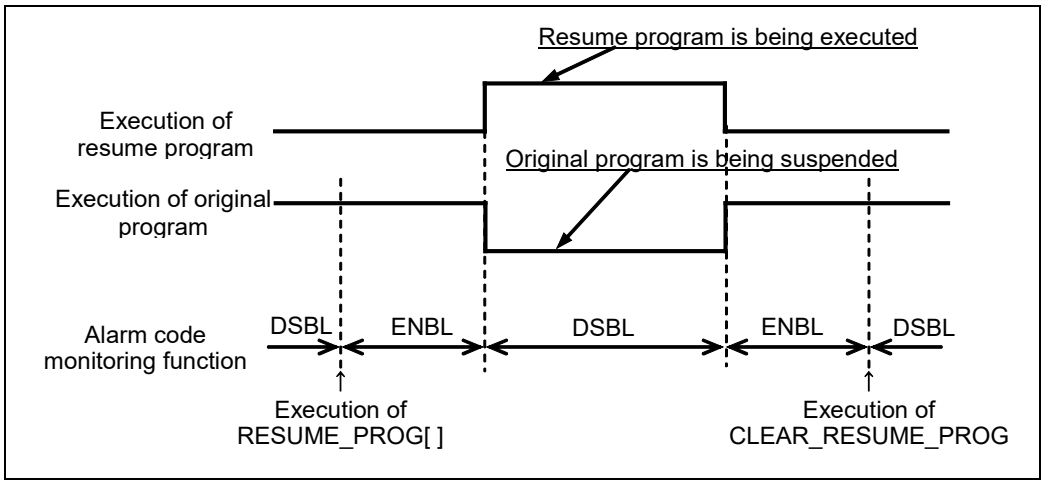


Fig. 6.7 (b) Enabling/disabling the alarm code monitoring function in the resume program function

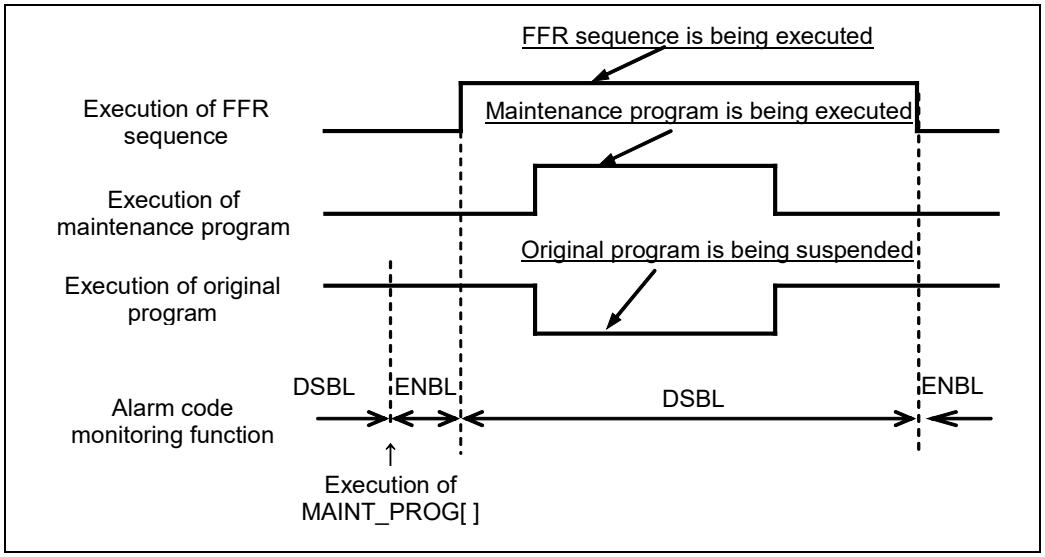


Fig. 6.7 (c) Enabling/disabling the alarm code monitoring function in the fast exit/entry feature

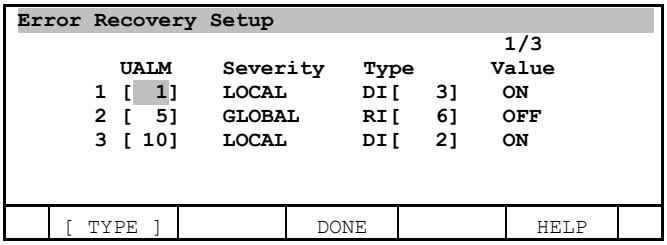
You can also change or confirm the setting via Data ID.
(Data ID: $ERROR_RECOVERY.ALARM_CODE[a,b] / INTEGER / RW / a: Instruction number 1-2, b: Line number)

Automatic error recovery alarm conditions

You can set up an automatic error recovery alarm that will be issued by inputting a defined digital input signal. After you set up the automatic error recovery alarm, when a defined alarm occurs, the resume program or FFR sequence will be executed automatically by inputting a defined digital input signal.

You set up automatic error recovery alarms as user alarms, for which you set up alarm severity and information about the monitored digital input signal. Set the alarm severity to either LOCAL or GLOBAL. LOCAL means that the alarm will be issued only for a program that defines a resume or maintenance program. For the monitored digital input signal, select a DI, RI, or WI, signal type, specify the signal number, and select the trigger status, ON or OFF.

Define the conditions for issuing an automatic error recovery alarm on the Error Recovery Setup screen, displayed by pressing F3, DI_ALARM on the Error Recovery Setup screen.



You can set the following items on the Error Recovery Setup screen. The alarm code for the automatic error recovery alarm is 12278.

- User alarm (UALM) number
  When the automatic error recovery alarm occurs, the user alarm message with the specified number is displayed as the alarm message. Changes to this setting take effect immediately after they are made.
  You can also change or confirm the setting via Data ID.
  (Data ID: $ERROR_RECOVERY.DIN_ALARM_USER_ALARM_NUMBER[a] / INTEGER / RW / a: Line number)

- Alarm severity
  You can set whether the automatic error recovery alarm is a local or global alarm. For LOCAL, the automatic error recovery alarm will work only for programs that define a resume or maintenance program. If there is no program that defines a resume or maintenance program, the alarm will be set to a GLOBAL alarm. Changes to this setting take effect immediately after they are made.

- Signal type
  Set the type of digital signal that generates an automatic error recovery alarm from among DI, RI, and WI. If you change this setting, you must cycle power for the new setting to take effect.

- Signal number
  Set the number of the digital signal that will generate an automatic error recovery alarm. If you change this setting, you must cycle power for the new setting to take effect.
  You can also change or confirm the setting via Data ID.
  (Data ID: $ERROR_RECOVERY.DIN_ALARM_DIN_INDEX[a] / INTEGER / RW / a: Line number)

- Detection signal status
  Set the status of the digital signal that generates an automatic error recovery alarm to ON (high) or OFF (low). If you change this setting, you must cycle power for the new setting to take effect.

For multi-tasking
- This item cannot be set for each instruction number. When this item is enabled for any instruction number, the setting will be automatically applied to all instruction numbers.
- If there are multiple programs that have a defined resume or maintenance program and the alarm severity is set to issue a LOCAL automatic error recovery alarm, an automatic error recovery alarm will be generated for all of the defined programs.
- Define the automatic error recovery alarm code in all of the instruction numbers that can be used when the automatic start feature is enabled, and the automatic error recovery alarm code is defined in the monitored alarm codes. If a LOCAL alarm is defined for the automatic error recovery alarm only by limited instruction numbers, even if a LOCAL alarm is issued while multiple programs are being executed at the same time, the automatic start feature might not be able to be executed.

6.7.2 RESUME PROGRAM type recovery (Setup items related to the Resume program function)

RESUME program comment
You can add a comment to the resume program instruction. To add a comment, move the cursor to the RESUME comment, press ENTER, and press the appropriate keys to enter the comment. Use a comment to display additional information that describes the resume program on the program edit screen.

For multi-tasking
This item can be set for each instruction number.

Status DO index number
When both the alarm code monitoring function and Approval DI function are disabled, the resume program will always be executed when the paused original program is resumed. However, when both functions are enabled, it is difficult to determine whether the original program or resume program will be executed at restart.
The Status DO function is provided to help you know which program will be executed upon restart. The status DO will turn on only when the resume program is executed at restart. When the Status DO is off, the original program will be executed at restart.

The status DO signal will turn on when the following conditions are true:
- The Automatic Error Recovery function is enabled. The Error recovery function item is set to ENABLED on the Error Recovery Set screen.
- Program execution is not in single step mode.
- The resume program is defined in the original program.
- The original program has a motion group.
- The original program is paused, and the resume program has not yet finished being executed.
- There is no optional function that disables the Automatic Error Recovery function. See Section 6.10, OTHER SPECIFICATIONS AND RESTRICTIONS.
- When the teach pendant is enabled:
  - The operation mode item on the Error Recovery MNFC (Manual Functions) screen is set to TP_TEST.
- When the teach pendant is disabled:
  - The operation mode item on the Error Recovery MNFC (Manual Functions) screen is set to AUTO.

- The remote conditions are met if the remote device is UOP.
- No alarm code has been defined. If any alarm code has been defined, the alarm will be issued.
- The Approval DI function is disabled. If this function is enabled, the Approval DI signal is on.

⚠ CAUTION
1. While the resume program is being executed, single step operation cannot be performed.
2. Even if the Status DO is on, the resume program will not be executed when backward execution of the original program is performed.
3. Backward execution is allowed during resume program execution.
4. The update cycle period for the Status DO is 300ms. When the conditions listed above have been changed, wait at least 300ms before executing the program.

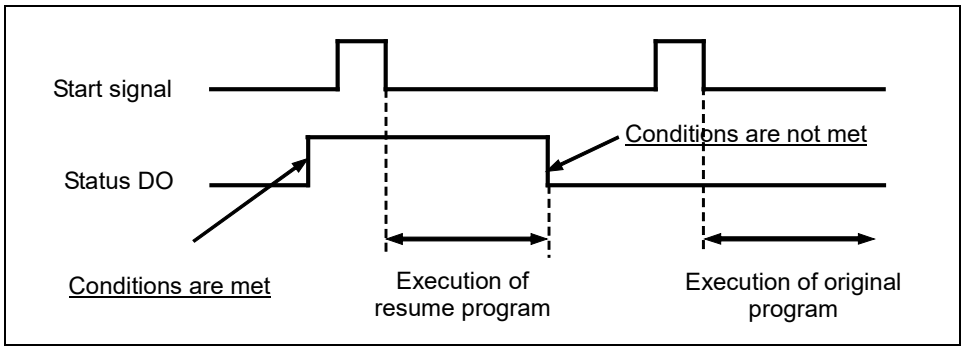


Fig. 6.7 (d) Status DO output timing chart

For multi-tasking
This item can be set for each instruction number.

You can also change or confirm the setting via Data ID.
(Data ID: $ERROR_RECOVERY.RESUME_PROGRAM_STATUS_DOUT_INDEX[a] / INTEGER / RW / a: Instruction number 1-2)

Automatic start maximum count

When a defined alarm occurs, the Automatic Error Recovery function automatically executes the resume program, then resumes the original program. If the defined alarm occurs again when the original program is resumed, the Automatic Error Recovery function will be executed again. For example, if the Automatic Error Recovery function is activated by an alarm indicating an arc start failure, then the same alarm will be issued again when the original program has resumed.
To prevent this kind of endlessly repeated condition, set the maximum number of automatic start repetitions in the Automatic start maximum count item.
The number of times the resume program is started repeatedly is counted internally. If the count exceeds the set value, the "INTP-134 Over automatic start Max counter" alarm occurs, and the Status DO is turned off at the same time. If this occurs, first eliminate the cause of the alarm in the original program and then input the start signal.

⚠ CAUTION
The number of repetitions counted internally is cleared when the execution of a motion instruction has been completed, or when the CLEAR_RESUME_PROG instruction has been executed.

For multi-tasking
This item can be set for each instruction number.

You can also change or confirm the setting via Data ID.
(Data ID: $ERROR_RECOVERY.RESUME_PROGRAM_AUTO_START_MAX_COUNT[a] / INTEGER / RW / a: Instruction number 1-2)

Automatic start count register
As mentioned above, the resume program might be executed repeatedly by the Automatic Error Recovery function. To avoid this situation, you can define the Automatic start count register, which will specify a different program to be executed as the resume program each time the resume program is executed. For example, if the resume program is executed for the first time by the Automatic Error Recovery function, the register value will be 1. If an alarm is issued again during execution of the original program, and the resume program is then executed again by the Automatic Error Recovery function, the register value will be 2. By executing a different subprogram in the resume program according to the register value, a different resume program operation can be performed each time the repetition count is incremented.

> ⚠ **CAUTION**
> When the resume program is executed by other than the Automatic Error Recovery function, the value of the register will be 0. Therefore, create a resume program that calls the same subprogram when the register value is 0 or 1.

> **For multi-tasking**
> This item can be set for each instruction number.

6.7.3 MAINTENANCE PROGRAM type recovery (Setting items related to the Fast exit/entry function)

MAINT program comment
You can add a comment to the maintenance program instruction. To add a comment, move the cursor to the MAINT comment item, press ENTER, and press the appropriate keys to enter the comment. Use a maintenance program comment to describe additional information on the program edit screen with the maintenance program instruction.

> **For multi-tasking**
> This item can be set for each instruction number.

Fast exit/entry feature
This item enables or disables the Fast exit/entry feature.
When the both the Automatic Error Recovery function and the Fast exit/entry feature are enabled, and neither monitored alarm codes nor the Approval DI are defined, the FFR sequence is always executed at restart from the paused state, except when the Maintenance DO is off.
When this item is disabled, the FFR sequence is not executed.

> **For multi-tasking**
> This item can be set for each instruction number.

You can also change or confirm the setting via Data ID.
(Data ID: $ERROR_RECOVERY.MAINTENANCE_PROGRAM_ENABLE[a] / BOOLEAN / RW / a: Instruction number 1-2 / TRUE: Enable, FALSE: Disable)

Dry run exit/entry operation

In the Fast exit/entry feature, the Dry run exit/entry item specifies whether exiting from the paused position and returning to the paused position after maintenance program execution will be performed at dry run speed. This item can be set only for instruction number 1 in single-tasking.

For multi-tasking
This item can be set for each instruction number.

MAINT DO index number
When both the alarm code monitoring function and Approval DI function are disabled, the FFR sequence will always be executed when the paused original program is resumed. However, when both of the functions are enabled, it is difficult to determine whether the original program or the FFR sequence will be executed at restart. The Maintenance DO function is provided to help you know whether the FFR sequence will be executed upon restart.
The Maintenance DO will turn on only when the FFR sequence is executed at restart. In addition, because the FFR sequence will not be executed even though the original program is executed at restart when the signal is off, the maintenance program will not be executed after the original program is completed.

The Maintenance DO signal will turn on when the following conditions are true:
- The Automatic Error Recovery function is enabled. The Error recovery function item is set to ENABLED on the Error Recovery Set screen.
- Program execution is not in single step mode.
- The maintenance program is defined in the original program.
- The Status DO is not on.
- The FFR sequence is being executed.
- The fast exit/entry feature is enabled.
- When the teach pendant is disabled:
  - The remote conditions are met if the remote device is UOP.
  - No alarm code has been defined. If any alarm code has been defined, the alarm will occur.
  - The Approval DI function is disabled. If this function is enabled, the Approval DI signal is on.

⚠ CAUTION
1 While the FFR sequence is being executed, do not perform backward execution or single step operation.
2 The update cycle period for the Maintenance DO is 300ms. If the conditions listed above have been changed, wait at least 300ms before executing a program.
3 When the teach pendant is enabled, the FFR sequence cannot be executed.

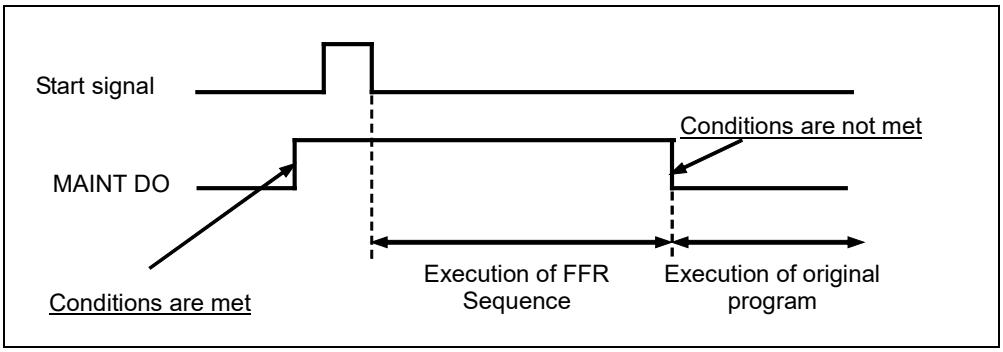


Fig. 6.7 (e) MAINT DO output timing chart

For multi-tasking
This item can be set for each instruction number.

You can also change or confirm the setting via Data ID.
(Data ID: $ERROR_RECOVERY.MAINTENANCE_PROGRAM_DOUT_INDEX[a] / INTEGER / RW / a: Instruction number 1-2)

Skip maintenance program
With this item, you can choose whether to execute the FFR sequence when resuming from a paused state. When this item is ENABLED, neither the FFR sequence will be executed.
You can also change or confirm the setting via Data ID.
(Data ID: $ERROR_RECOVERY.SKIP_MAINTENANCE_PROGRAM[a] / BOOLEAN / RW / a: Instruction number 1-2 / TRUE: Enable, FALSE: Disable)

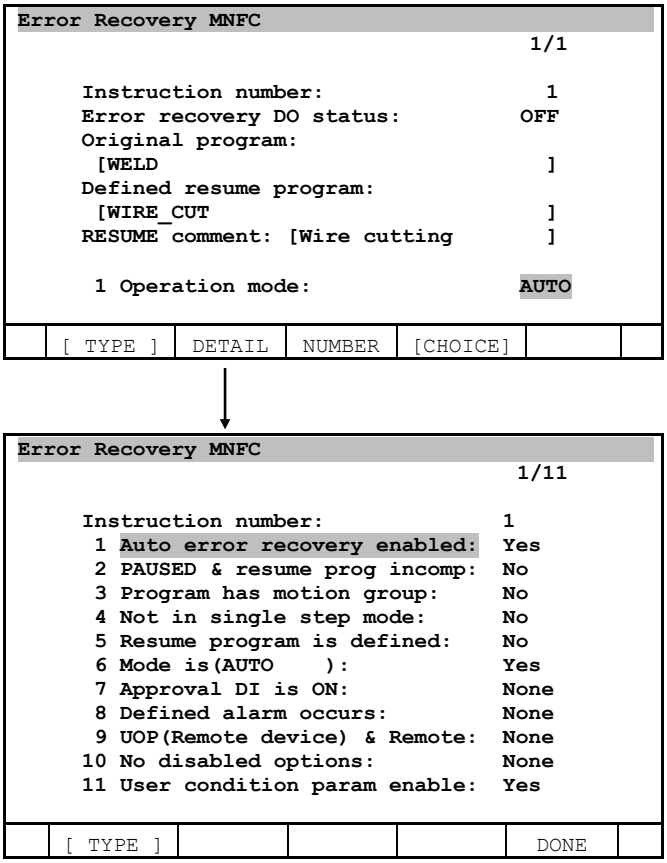
6.8 MANUAL FUNCTIONS SCREEN FOR THE RESUME PROGRAM FUNCTION

A manual functions screen is provided for the resume program function The Fast exit/entry feature does not have a manual function screen.
The manual functions screen can be displayed by following the steps below.

Procedure 6-2 Displaying manual functions screen for resume program function

Steps

1 Press the [MENU] key. The menu will be displayed.
2 Select MANUAL FCTNS.
3 Press the F1, [TYPE].
4 Select Err recovery. The manual functions screen will be displayed.



Instruction number
This item displays the instruction number of the resume program instruction. By changing this item, you can switch to a manual functions screen for other instruction numbers.

Error recovery DO status
This item displays the status of the Error recovery status DO, regardless of whether the Error Recovery Status DO has been defined on the Error Recovery Setup screen. From this information, you can tell whether the resume program or original program is to be executed.

Original program
This item displays the name of the original program in which the resume program is defined.

Defined resume program
This item displays the name of the resume program defined in the currently selected original program. From this information, you can verify whether an incorrect resume program has been defined.

> ⚠ **CAUTION**
> If an incorrect program has been defined as the resume program, robot operation will be unpredictable. Therefore, make sure that the resume program is correct.

Resume program comment
This item displays the comment for the resume instruction.

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

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

6.8.1 Displaying detail conditions of the Error recovery status DO

When you press F2, DETAIL on the Error Recovery MNFC screen of the Automatic Error Recovery function, detailed conditions related to the Error recovery status DO status are displayed. When all of the items on the detail screen are set to “Yes” or “None”, the Error recovery status DO is turned on. When the Error recovery status DO is off, and you cannot find the cause of the DO being off, check this screen for more information.

Auto error recovery enabled
This item indicates whether this function is enabled or disabled on the Error Recovery Set screen.
You can also check the results via Data ID.
(Data ID: $ERROR_RECOVERY.AUTO_ERROR_RECOVERY_IS_ENABLED[a] / INTEGER / RO / a: Instruction number 1-2 / 1: None, 2: No, 3: Yes)

PAUSED & resume prog incomp
When this item is set to Yes, it indicates that the following conditions are true:
- The original program is paused.
- A resume program has been defined in the original program, and the execution of the resume program has not been completed.

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

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

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

Mode is (xxxx)
This item indicates that the operation mode is suitable for the current status.
For example, if the teach pendant is disabled, "AUTO" is displayed in the field "xxxx." If the teach pendant is enabled, "TP_TEST" is displayed.

Approval DI is ON
This item indicates the status of the Approval DI signal. If the DI number has not been defined, or if the teach pendant is enabled, "None" is displayed.

Defined alarm occurs
This item indicates that an alarm code has been defined, and that alarm is has occurred.
If no alarm code has been defined, or if the teach pendant is enabled, "None" is displayed.

UOP (Remote device) & Remote
This item indicates whether remote conditions have been satisfied. This function is enabled only when the teach pendant is disabled and the remote device is UOP.

No disabled options
The Automatic Error Recovery function is incompatible with certain options. This item indicates whether an incompatible option is present.

User condition param enable
This item indicates the status of the user condition parameter.

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

Normally, the Automatic Error Recovery function is used when production is started with the teach pendant disabled. When checking the resume program during teaching, set the operation mode to TP_TEST on the Error Recovery MNFC screen. In TP_TEST mode, the resume program can be executed regardless of the Approval DI status and whether a defined alarm has occurred. A manual functions screen is not available for the Fast exit/entry feature.

6.10 OTHER SPECIFICATIONS AND RESTRICTIONS

6.10.1 Automatic Error Recovery function

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

6.10.2 Resume program function

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

6.10.3 Fast exit/entry feature

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

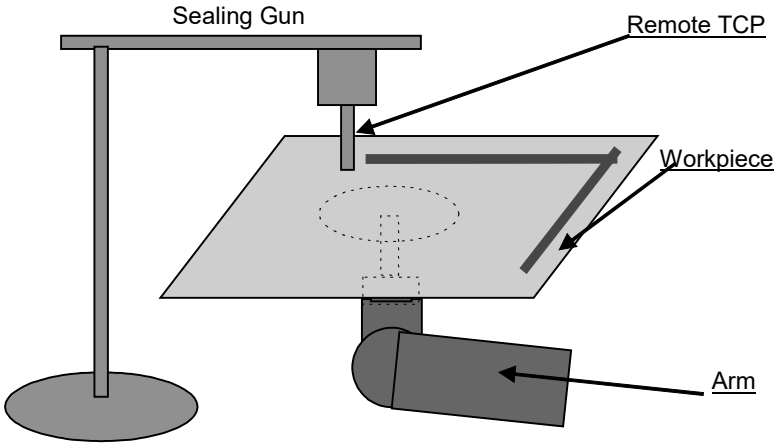
6.11 WARNINGS (Be sure to read this section for safety.)

When using the Automatic Error Recovery function, observe the following safety precautions:

- If an incorrect program or a program causing incorrect operation is defined as a resume or maintenance program, the robot will move in a direction that you cannot predict. Define a correct program.
- Before inputting the start signal and before pressing the FWD key on the teach pendant, for safety, check the resume program information DO status to confirm whether the original program or resume program is to be started.
- Before inputting the start signal and before pressing the FWD key on the teach pendant, for safety, check the Error Recovery status DO to confirm whether the original program or maintenance program is to be started.
- If the operation mode is set to TP-TEST on the Error Recovery MCFN screen of the automatic error recovery function, the resume program will be started even if a defined alarm has not occurred, or if the Approval DI is off.
- If an operation mode other than AUTO is set on the Error Recovery MCFN screen of the Automatic Error Recovery function, and then the display is changed to another screen, the operation mode is set to AUTO again automatically. To use an operation mode other than AUTO, keep displaying the Error Recovery MCFN screen.
- If a resume or maintenance program terminates in the middle, and the start signal is input, the robot might act unpredictably. Therefore, before inputting the start signal, check the program name and the program start line that will be executed with the next start signal. In addition, if an obstacle exists, jog the robot to a position free of obstacles, and then input the start signal.
- While executing the FFR sequence, do not perform backward execution or single step operation.

7 REMOTE TCP FUNCTION

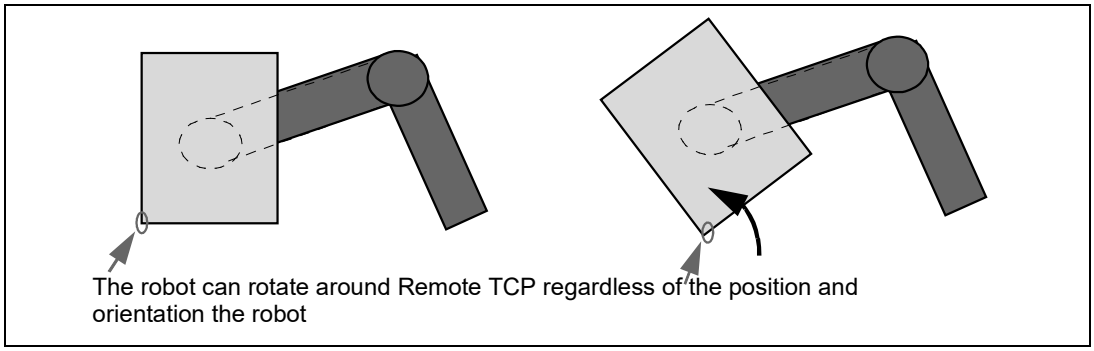
The Remote TCP function (RTCP) allows you to control the orientation of the robot in applications where the tool (TCP) is attached to the ground in the workcell and the robot manipulates the workpiece around the tool.
The following figure is an example of using the Remote TCP function in a sealing application.



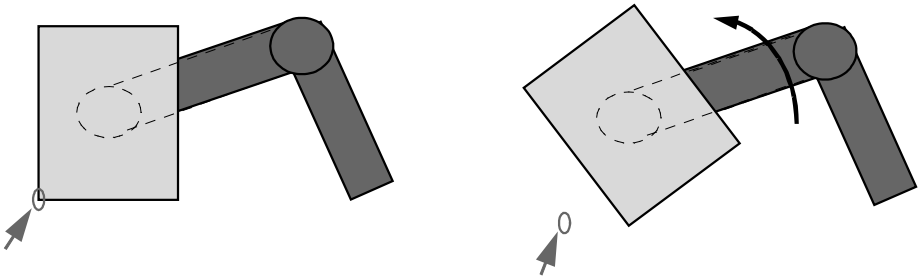
Advantage of Remote TCP function

- Since the gun is attached to the ground, the tool cabling is simplified.
- The robot no longer needs to carry a heavy gun.
- Even without the Remote TCP function, machining can be performed by moving a workpiece held by the robot against a TCP attached to the ground. However, the Remote TCP function makes teaching easier and more accurate.

With the Remote TCP function

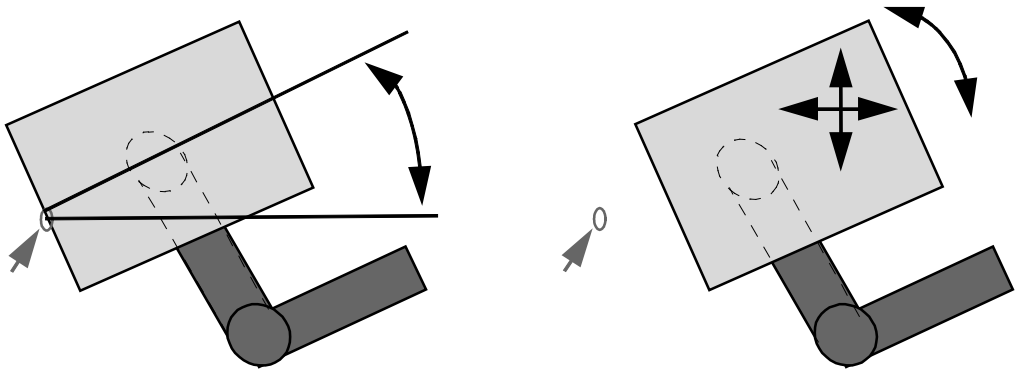


Without the Remote TCP function.



The robot CAN NOT rotate around Remote TCP since, it depends on the position and orientation the robot

- The Remote TCP jog function simplifies teaching.



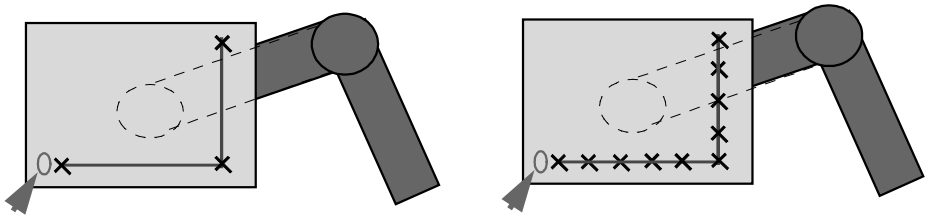
Remote TCP

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

No Remote TCP

Not easy to jog around the Remote TCP.

- The Remote TCP function drastically reduces the number of teaching points.



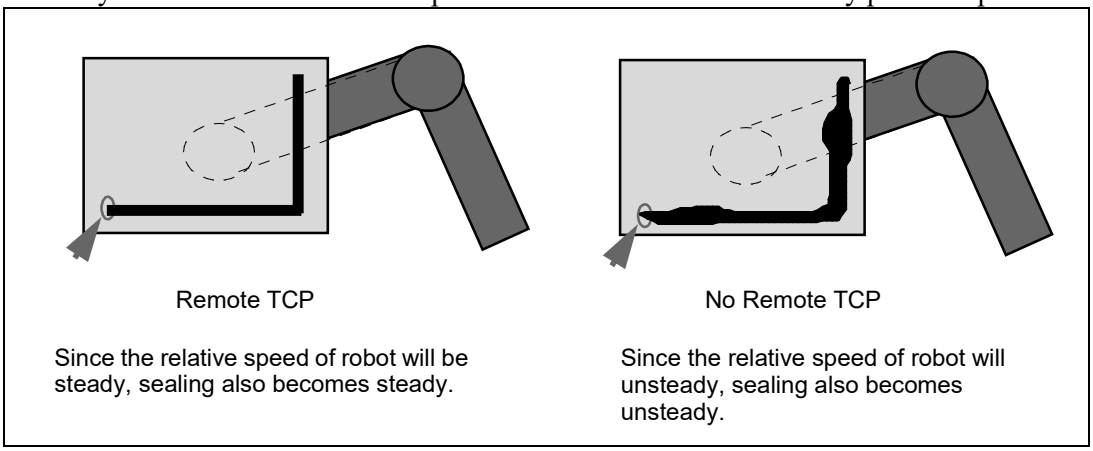
Remote TCP

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

No Remote TCP

You need many teaching points according to the orientation and location of workpiece.

- The ability to achieve stable relative speeds allows for uniform and steady process operations.



Required option

- Remote TCP (A05B-2700-J624)

Limitations

- Remote TCP cannot be used in joint motion and Wrist joint motion.
- Remote TCP cannot be used on robots with the Coordinated Motion function installed.
- Remote TCP cannot be used with the following functions.
  - Line tracking
  - Weaving
  - Touch sensor
  - Arc sensor
  - AVC (TIG arc length control)
  - MP/RPM
  - Singularity avoidance
- When using the TCP Speed Output function, the accuracy of the TCP speed prediction function may decrease when the robot motion changes from Remote TCP to normal operation.

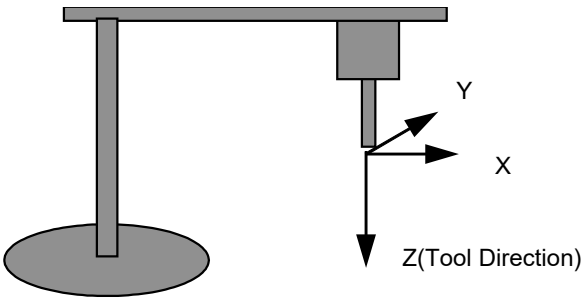
7.1 SETUP

Setup the motion group

To use the Remote TCP for group 1, this setting is not necessary.
Otherwise, please set motion groups for Remote TCP in System Config Menu described in 2.16 of B-84464EN/03 (FANUC Robot series R-50iA/R-50iA Mate OPERATOR'S MANUAL (Basic Operation)).

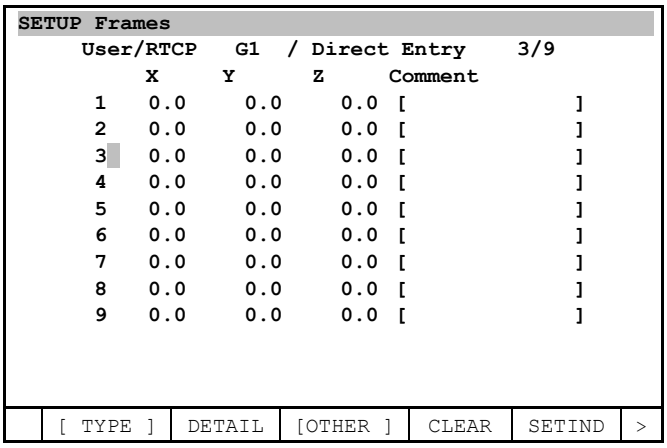
Setup the Remote TCP

To use the Remote TCP function, you need to teach the robot the tool center point that is attached to the ground.



You set the Remote TCP position using the same method as for a User frame.

1 Press MENU and select SETUP
2 Press F1, [TYPE] and select Frames.
3 If user frame is not displayed, press F3, [OTHER], and select User/RTCP. If F3, [OTHER], is not displayed, press PREV.
4 Move the cursor to the REMOTE TCP frame to use.



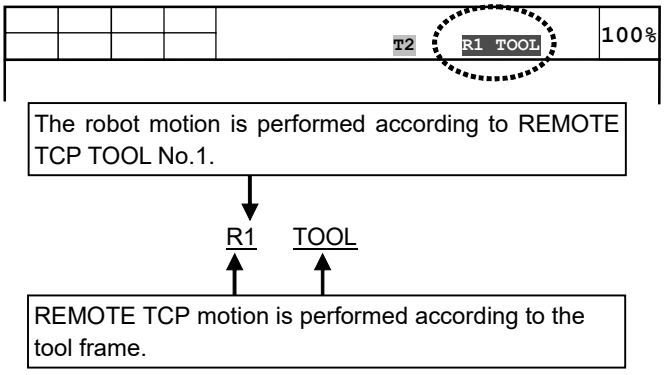
5 Press F2, DETAIL.
6 Select the method from Direct Entry, Three Point or Four Point, and set the frame using the same method as for User frame.

Remote TCP jogging function

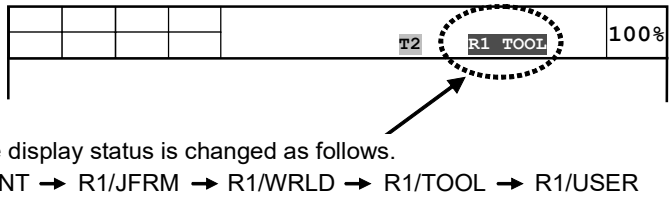
Use the Remote TCP jogging function to jog the robot around a remote TCP.

Use the following procedure to perform Remote TCP jogging.

1 Enter to Remote TCP jogging mode.
   a. Press FCTN.
   b. Select TOGGLE REMOTE TCP and then the screen will be changed as follows.



2 Press COORD to select the frame from among REMOTE TCP USER, REMOTE TCP TOOL, REMOTE TCP JOG. DO NOT select JOINT.



3 Jog the robot.

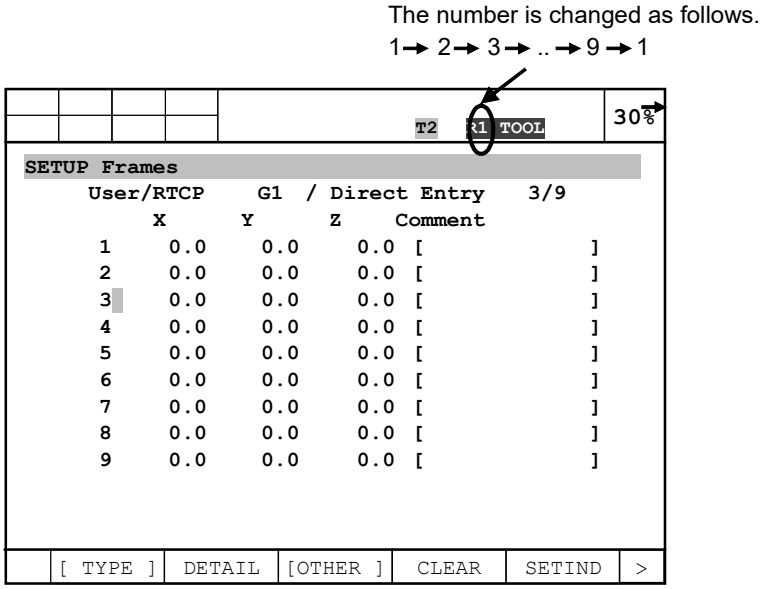
4 To exit Remote TCP jogging mode.
   a. Press FCTN
   b. Select TOGGLE REMOTE TCP.

You can confirm whether the Remote TCP jogging is enabled by the following Data ID.
(Data ID: $JOG_MODE.IS_REMOTE_TCP_JOG[a] / BOOLEAN / RO / a: group number)

Select REMOTE TCP frame

While in Remote TCP mode, you can use the following procedure to change the Remote TCP frame.

1 Press FCTN.
2 Select CHANGE RTCP FRAME.



Programming and running by using REMOTE TCP

To operate robots using a remote TCP instead of robot TCP in a program, you need to specify the Remote TCP motion modifier as follows.

1 Move the cursor to end of a motion instruction line.
2 Press F4, [CHOICE].
3 Select RTCP.

You cannot use RTCP with JOINT motion.

<Programming example>

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

The robot moves to P[1] using the relative speed 100 mm/sec against the remote TCP.

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

The robot moves to P[2] via P[1] using the relative speed 100 mm/sec against the remote TCP.

8 HIGH-SENSITIVITY COLLISION DETECTION

High-Sensitivity Collision Detection quickly detects a collision and stops the robot when the robot or end-of-arm tooling collides with a workpiece or another object.
This function is much more sensitive than the normal collision detection function and can detect collisions more quickly, greatly reducing the potential for damage to the end-of-arm tooling and robot.

8.1 SPECIFICATIONS

1) When collision is detected, this function issues an alarm and stops the robot quickly by decelerating it in a way that minimizes shocks to the robot.
2) Since the detection sensitivity has been adjusted for each robot type in advance, you can use this function without adjusting the detection sensitivity.
3) You can change the detection sensitivity using a program instruction or on a setup screen.
4) You can output a signal to a DO when a collision is detected.
5) You can output the collision detection status (enabled / disabled) to a DO.
6) During teaching, the detection sensitivity is automatically increased to minimize damage caused by mistakes, especially during teaching operations where the robot is prone to collisions.
7) This function is enabled when the robot controller is powered on.

8.2 SETTINGS

Load information / Information about Installed Devices (Required)

Specify load information and the information about the devices installed on the robot.
Since the function uses the load information and device information to detect a collision, you must supply this information. Be sure to specify the weight of the load, the center of gravity, and the weight of each device on the robot accurately.
If the load has a large inertia (shape), it might be necessary to specify the inertia around the load's center of gravity. If the tool is large, and simply specifying its weight and center of gravity does not provide accurate detection, specify its inertia.
See Section 2.18, "LOAD SETTING" in FANUC Robot series OPERATOR'S MANUAL (Basic Operation) (B-84464EN) or Section 9, " LOAD ESTIMATION" in this manual, for information on how to specify load information.

Collision Detect Setup screen (Optional)

The detection sensitivity has previously been adjusted for each robot type. If you want to change the sensitivity, set in this screen.
Also, If you want to output the status of the collision detection (enabled / disabled), set in this screen.
See Section 8.3, "Collision Detect Setup Screen".

Teaching program instructions (Optional)

If you want to temporarily change the detection sensitivity, insert program instructions.
See Section 8.4, "Program Instructions".

8.3 COLLISION DETECT SETUP SCREEN

Set as below.

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

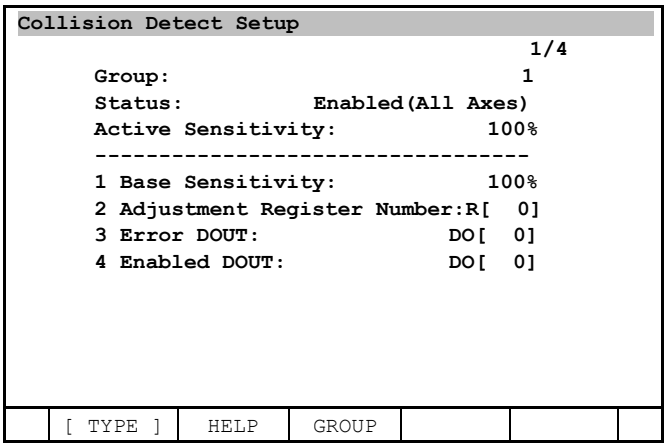


Fig. 8.3 Collision Detect Setup screen

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

Table 8.3 Setup items of Collision Detect Setup screen

Item	default	Description
Group	1	This item indicates the currently selected group. To change selected group, press F3, GROUP and entering the group number. You will only be able to select a group that supports High-Sensitivity Collision Detection.
Status	Enabled (All Axes)	This item shows status of collision detection for the currently selected group. One of the following three will be displayed. 1. Enabled(All Axes) 2. Disabled(All Axes) 3. Disabled(Partial) “Disabled(Partial)” indicates that collision detection is disabled for some axes and enabled for other axes. This item cannot be changed in this setup screen. (Data ID: $COLLISION_DETECT.STATUS[a] / INTEGER / RO / a:group number / The above index number will be retrieved.)
Active Sensitivity	100%	This item indicates current sensitivity for Collision Detection for the currently selected group. The sensitivity is specified by Base Sensitivity setting on this screen and COL DETECT ADJUST instruction. Minimum is 100%. Maximum is 200%. The lower the value, the lower the sensitivity. The higher the value, the higher the sensitivity. (Data ID: $COLLISION_DETECT.ACTIVE_SENSITIVITY[a] / INTEGER / RO / a:group number)

Item	default	Description
Base Sensitivity	100%	This item allows you to set the default level of sensitivity for Collision Detection for the currently selected group. Minimum is 100%. Maximum is 200%. The lower the value, the lower the sensitivity. The higher the value, the higher the sensitivity. Note that the detection sensitivity specified by COL DETECT ADJUST instruction has priority over the value specified in this item. (Data ID: $COLLISION_DETECT.BASE_SENSITIVITY[a] / INTEGER / RO / a:group number)
Adjustment Register Number	0	This item allows you to specify the register that can be used with the COL DETECT ADJUST instruction (without arguments) to adjust the sensitivity of Collision Detection within a program. A register number of 0 indicates that COL DETECT ADJUST instruction without arguments is not used. (Data ID: $COLLISION_DETECT.ADJUSTMENT_REGISTER_NUMBER[a] / INTEGER / RW / a:group number)
Error DOUT	0	This item allows you to specify a Digital Output that will be turned ON when a Collision Detect Alarm occurs. A value of 0 indicates that no output will be turned on. (Data ID: $COLLISION_DETECT.ERROR_DOUT[a] / INTEGER / RW / a:group number)
Enabled DOUT	0	This item allows you to specify a Digital Output that will be turned on when Collision Detect Status is Enabled(All Axes), and turned off when Collision Detect Status is Disabled(All Axes) or Disabled(Partial). A value of 0 indicates that no output will be used. (Data ID: $COLLISION_DETECT.ENABLED_DOUT[a] / INTEGER / RW / a:group number)

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

8.4 PROGRAM INSTRUCTIONS

The program instructions below are reserved.

- COL DETECT ADJUST

8.4.1 COL GUARD ADJUST

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

This instruction can be used in three ways depending on how to specify arguments. (See Table 8.4.1.)

Table 8.4.1 COL DETECT ADJUST instruction

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

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

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

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

This message is just a warning to indicate that no action was taken on that group.
To Use the COL DETECT ADJUST instruction without arguments in a TP program, the following procedure should be used:

- Set the register(s) specified in the Collision Detect Setup menu for the appropriate group(s) to the new sensitivity value(s).
- Insert the COL DETECT ADJUST instruction into the TP program at the desired location.

NOTE
If the TP program is a multiple group program, the appropriate register for each group needs to be set.

NOTE
in case of directly set, if specified sensitivity is not an integer between 100 and 200, then an error will be posted as follows.

```
MOTN-405 Sensitivity range is 100 - 200
```

NOTE
If the register number in the Collision Detect Setup menu is zero for one of the groups being adjusted, then an error will be posted as follows.

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

NOTE
If the data in specified register is not an integer between 100 and 200, then the following error will be posted.

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

Example 1
It assumes following conditions.

- Program contains a single motion group only.
- Register number has been set to 11 for group 1 in the Collision Detect Setup menu.

```
10: J P[1] 100% FINE
11: R[11]=120
12:  COL DETECT ADJUST
13: L P[2] 2000mm/sec CNT100
14: L P[3] 2000mm/sec CNT100
15: L P[4] 2000mm/sec CNT100
16: R[11]=100
17:  COL DETECT ADJUST
18: J P[5] 50% FINE
```

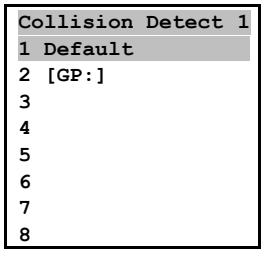
In this program, the collision detection sensitivity becomes 120% in lines 13 to 15, and is 100% in line 18.

Example 2

```
10: J P[1] 100% FINE
11:  COL DETECT ADJUST 120
12: L P[2] 2000mm/sec CNT100
13: L P[3] 2000mm/sec CNT100
14: L P[4] 2000mm/sec CNT100
15: R[1]=100
16:  COL DETECT ADJUST R[1]
17: J P[5] 50% FINE
```

In this program, the collision detection sensitivity becomes 120% in lines 12 to 14, and is 100% in line 17. To use the COL DETECT ADJUST instruction with arguments, move the cursor to the right of the instruction. Then, either type in the new sensitivity value directly, or select F3"INDIRECT" and type in the desired register number.

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



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

COL DETECT ADJUST GP1,2: ##

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

By default, when a TP program that calls COL DETECT ADJUST is aborted, the sensitivity will be reset to the value specified in the Collision Detect Setup menu for each group in the program's group mask.

8.5 CAUTIONS

1 Collision Detection might detect a false collision when a collision has not occurred in the following cases:
- Load information or information about installed devices has not been set correctly.
- The Load is larger than the maximum load for the robot, or the inertia of the load is too large.
- Not enough voltage has been supplied to the controller.
- Low temperature.
- Rapid motion such as reverse motion which Cnt. is used.
- Linear motion occurs near singularity point where axes revolve in high speed.

Action: If collision mis-detection occurred for the reasons above, try removing these reasons first.

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

3 In order to decrease the pushing force of robot after collision, vertical robot axes might fall slightly after detecting a collision.

4 If the robot is repeatedly subjected to the force that causes collision detection, the mechanical part will be overloaded and fragile. If collision detection occurs repeatedly during program execution, it is recommended to change the program so that the force applied to the robot is reduced.

5 Neither COL_DET_OFF nor COL_DET_ON instruction is supported. If those instruction is executed, then the following error will be posted.

```
MOTN-406 COL_DET_OFF/ON is not allowed
```

6 Collision detection cannot be disabled and sensitivity cannot be set below 100%, as this may cause damage to the robot or premature failure. However, for temporary use in emergencies, collision detection can be temporarily disabled in the System Config Menu.
For details, refer to "PPERATOR'S Manual (Basic Function)" (B-84464EN).

9 LOAD ESTIMATION

Load Estimation is a function for estimating the weight of the payload, such as a tool or workpiece, mounted on the robot's faceplate.
Using the Load Estimation function, payload information can be automatically estimated by moving the robot.

To use this function, the High-Sensitivity Collision Detection Package option or package option that includes it (such as the Motion Package, Basic Interference Check, Intelligent Interference Check, ARC Easy Smart Quick Recovery, Lincoln Asia-Pacific Arc package for Standard EQ, Lincoln Asia-Pacific Arc package) is required. Using this function also requires that your robot model supports the Load Estimation function. If your robot model does not support this function, you cannot use it.

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

9.1 OPERATING PROCEDURE

Load is estimated in the following steps:
1 Set the range of motion to be subjected to Load Estimation.
2 Execute Load Estimation.

When mechanical parts such as motors are replaced, calibration is required.
If these parts are replaced without performing calibration, the precision of the Load Estimation will decrease.

> ⚠ **CAUTION**
> When Load Estimation is executed, the J5 and J6 axes will move with maximum speed (100% joint motion). Do not enter the operating space of the robot. Also, make sure that all devices attached to the robot can withstand such high-speed motion.
> Some robot models support an additional type of estimation motion that moves the J3, J5 and J6 axes simultaneously.

9.2 LOAD ESTIMATION PROCEDURE (for 6-Axis Robots)

There are three types of Load Estimation, and the type supported depends on your robot model:

- **2 axes shuttle** (default type) moves J5 and J6 between two points at low speed, then repeats the same motion at high speed.
- **Estimation by 3 motions** adds a J5-only motion to the motions in the 2 axes shuttle method. You can use this type of estimation to estimate payload when the center of gravity is near the J6 axis of rotation, a configuration that cannot be estimated with the 2 axes shuttle type. Estimation by 3 motions moves the robot to four different points. First, it performs J5 and J6 motions at low speed as in the 2 axes shuttle type, then J5-only motion at high speed, and finally J5 and J6 motions at high speed as in the 2 axes shuttle type.
- **3 axes waves** moves J3, J5 and J6 simultaneously. Even if the mass of the load is unknown, it may be estimated with higher accuracy with this type than with the other types of estimation..

Although the estimation motion setup is different, Load Estimation for all three estimation types is executed with the same process.

9.2.1 Preparation

All three types of Load Estimation are performed on the PAYLOAD_ID screen. This screen is accessed from the PAYLOAD SETTING screen.

1 Press the [MENU] key.
2 Select “SYSTEM” on the next page.
3 Press F1, [TYPE].
4 Select "Payload". The PAYLOAD SETTING/LIST screen will be displayed. If any other screen is displayed, press the [PREV] key several times until the list screen is displayed. In a multi-group system, to display the list screen for another group, press F2, GROUP.

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

 [ TYPE ] | GROUP | DETAIL | ARMLOAD | SETIND | >
```

5 Press the [NEXT], > and press F2, IDENT. The PAYLOAD ID screen will be displayed.

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

    2 MASS IS KNOW    [NO ]    165.000[kg]

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

 [ TYPE ] | GROUP | NUMBER | EXEC | APPLY | >
```

6 Move the robot to the position where load estimation is to be performed.
For the 2 axes shuttle (default type) and Estimation by 3 motion estimation types, J1 to J4 will stay in this position during the estimation. For the 3 axes waves motion pattern, J1 and J2 will stay in this position.

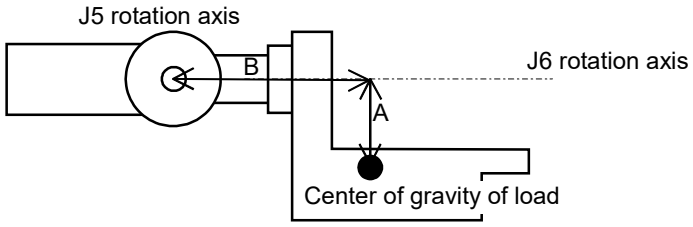
NOTE
1 Position the J3 arm so that it is horizontal. The closer the J3 arm is to vertical, the lower the precision of the estimation.
2 Position the J5 arm so that its axis of rotation is horizontal.
The closer the J5 rotational axis is to vertical, the lower the precision of estimation.

7 Press F3, NUMBER, and select the load setting schedule number for which a load estimate is to be set.
8 If the mass of the load for which Load Estimation is to be performed is known, move the cursor to MASS IS KNOWN, select YES, and specify (enter) the mass.

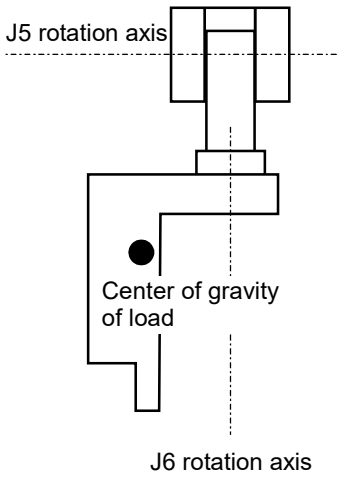
NOTE
A load with very small mass cannot be estimated. As a rough guide, the mass of the load should be at least 5% of the maximum payload.

The estimation precision increases when a mass is specified. Specify the mass as often as possible.
Even if no mass is specified, estimation with lower precision is still possible, provided that the following conditions are satisfied. However, the precision is reduced.

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

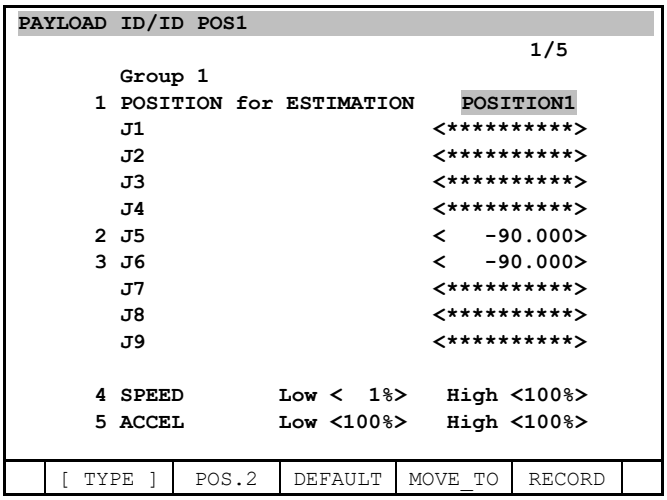


- For positions set up on the PAYLOAD ID/ID POS1 and POS2 screens, the center of gravity of the load must be in or near the plane that contains the J5 and J6 rotation axes.



- For the J6 axis, the interval between points specified on the PAYLOAD ID/ID POS1 and POS2 screens must be 180° in terms of angle.

9 Press the [NEXT] key, then press F4, DETAIL. PAYLOAD ID/ID POS1 screen will be displayed.



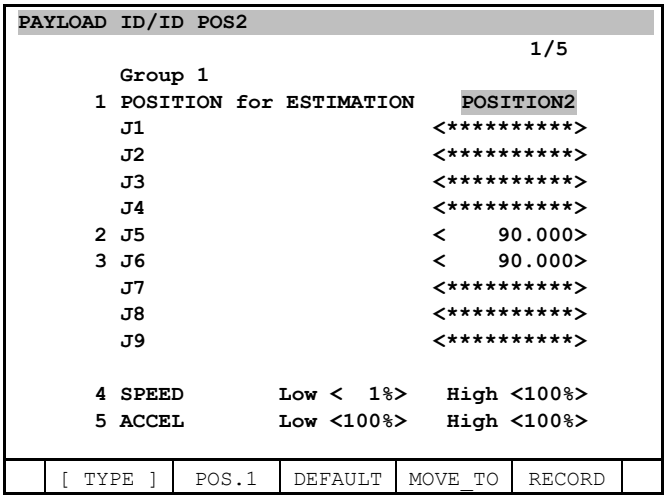
9.2.2 Estimation Motion Setup

This section describes estimation motion setup. This setup varies depending on the estimation type. Refer to the corresponding section to set up motion for the estimation type you are using.

Estimation Motion Setup for the 2 axes shuttle estimation type

Use this procedure to set up motion for the 2 axes shuttle estimation type.

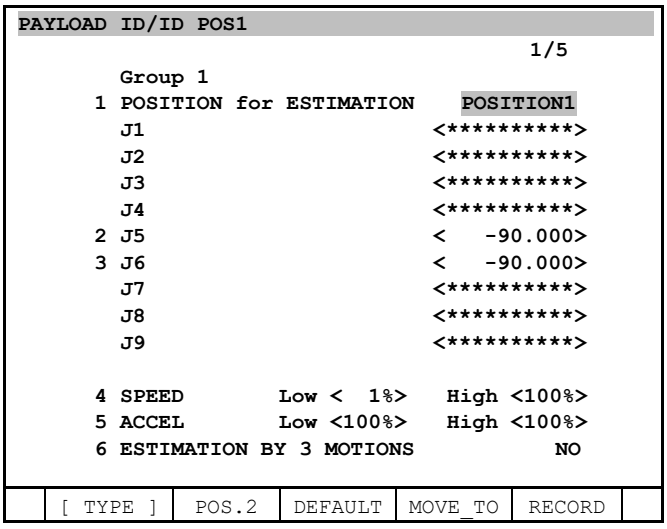
1 Set up estimation position 1. Alternatively, you can leave the default values as they are.
Specify the positions of the J5 and J6 axes either by entering their values directly or by jogging the robot to the desired position and pressing the [SHIFT] key + F5, RECORD to record the position.
Now, when you press the [SHIFT] key + F4, MOVE_TO, the robot will move to estimation position 1.
Use this procedure to check the set position.
2 Press F2, POS.2 to display the PAYLOAD ID/ID POS2 screen.



3 Specify estimation position 2. Alternatively, you can leave the default values as they are.
Specify the positions of the J5 and J6 axes by entering their values directly, or by jogging the robot to the desired position and pressing the [SHIFT] key + F5, RECORD to record the position.
Now when you press the [SHIFT] key + F4, MOVE_TO, the robot will move to estimation position 2.
Use this procedure to check the set position.
4 Press the [PREV] key to return to the PAYLOAD ID screen.

Estimation Motion Setup for the Estimation by 3 motions estimation type

Use this procedure to set up motion for the Estimation by 3 motions estimation type.
For robot models that support this type of estimation, a line with ESTIMATION BY 3 MOTIONS will be displayed at the bottom of the PAYLOAD ID/ID POS1 screen.



To use the Estimation by 3 motions, specify position 1 and position 2 for J5 and J6 motion, and position 3 and position 4 for J5-only motion.

1 Specify estimation position 1. Alternatively, you can leave the default values as they are.
Specify the positions of the J5 and J6 axes either by entering their values directly, or by jogging the robot to the desired position and pressing the [SHIFT] key + F5, RECORD to record the position.
Press the [SHIFT] key + F4, MOVE_TO to move the robot to estimation position 1.
2 Press F2, POS.2 to display the PAYLOAD ID/ID POS2 screen.
3 Specify estimation position 2 in the same way as estimation position 1.
Press the [SHIFT] key + F4, MOVE_TO to move the robot to estimation position 2. Use this procedure to check that the motion between position 1 and 2 has no interference.
4 Press F2, POS.1 to display the PAYLOAD ID/ID POS1 screen, and press the [SHIFT] key + F4, MOVE_TO to move the robot to position 1.
5 Move the cursor to ESTIMATION BY 3 MOTIONS, press F4 [YES], and note that the line display changes to YES.
6 Press the [F2] key twice to display the PAYLOAD ID/ID POS3 screen. F2 normally switches between [POS.1] and [POS.2], but when the ESTIMATION BY 3 MOTIONS is set to YES, pressing F2 will toggle between the screens for POS.1, POS.2, POS.3, and POS.4. Position 3 is the start and end point of J5-only motion.
7 Specify the estimation position 3 in the same way as positions 1 and 2.
8 Press the [SHIFT] key and F4 [MOVE_TO] to move the robot to position 3, and make sure that it will not interfere with its surroundings.
9 Press the F2 [POS. 4] to display the PAYLOAD ID/ID POS4 screen.
10 Specify estimation position 4. Position 4 is the turnaround position for J5-only motion. The J6 position in position 4 is automatically the same as that of position 3 and cannot be changed.
11 Press the [SHIFT] key and F4 [MOVE_TO] to move the robot to position 4, and confirm that it will not interfere with its surroundings.
12 Press the [PREV] key to return to the PAYLOAD ID screen.

NOTE

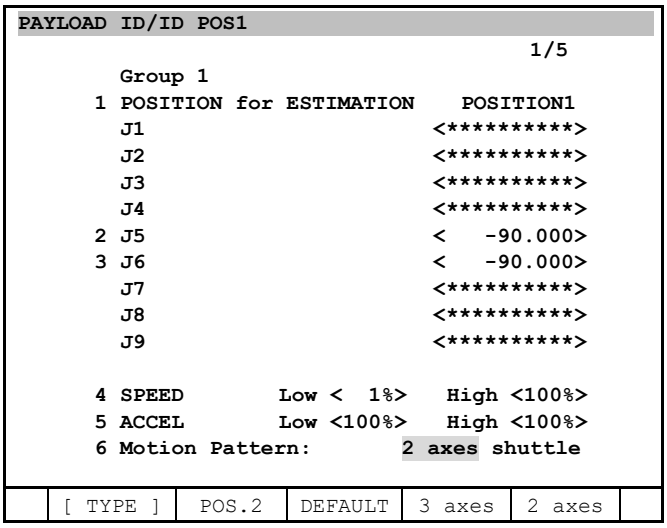
In Estimation by 3 motions, the robot will move in the following way.
Start position -> position 1 (override at the time) -> position 2 (low-speed) -> position 1 (low-speed) -> position 3 (high-speed) -> position 4 (high-speed) -> position 3 (high-speed) -> position 1 (high-speed) -> position 2 (high-speed) -> position 1 (high-speed).
Be sure to use SHIFT key and F4 [MOVE_TO] to confirm that the Estimation by 3 motions robot motion is safe.

Calibration, to be described later, will run in the same way as for 2 axes shuttle estimation, regardless of the value of the ESTIMATION BY 3 MOTIONS item.

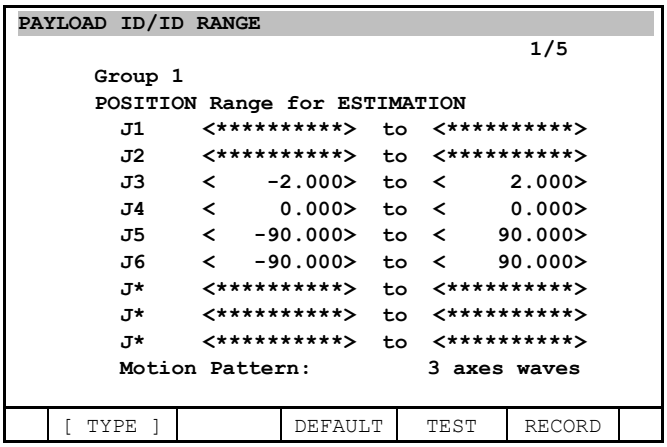
Estimation Motion Setup for the 3 axes waves estimation type

Use this procedure to set up motion for the 3 axes waves estimation type. The 3 axes waves estimation type requires that ARMLOAD settings, including center of gravity, are set correctly.

1 On the PAYLOAD ID/ID POS1 screen, there is an item for Motion Pattern, which should display 2 axes shuttle. Move the cursor to 2 axes and press F4 [3 axes] to select 3 axes. The PAYLOAD ID/ID RANGE screen will be displayed.
If the item is not selectable, the robot model does not support tye 3 axes waves type of estimation.



2 Specify the motion range for estimation. Axes marked with *s will remain at the position where the estimation is executed. Axes that cannot be selected cannot be set to a position other than the one indicated. There are limits to the width of the motion range. Some axes have a fixed width, while others require certain widths.
To return the values to their default settings, press F3 [DEFAULT]. To set the current position of the selected axis to the selected side of the range, press F5 [RECORD].

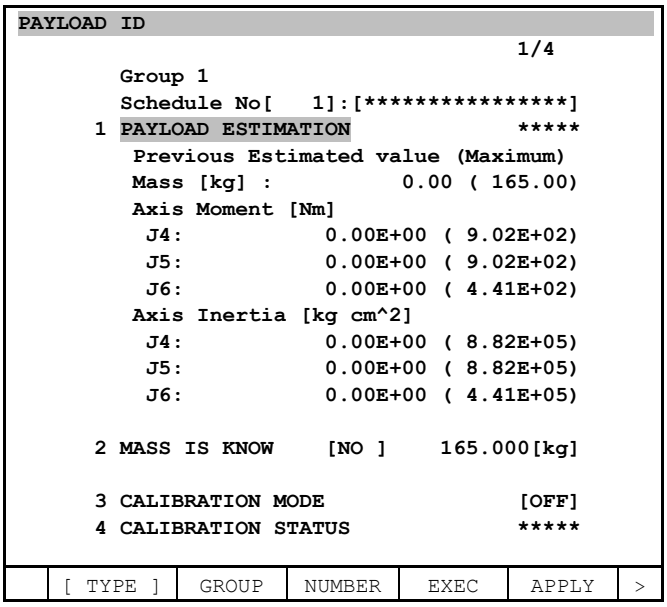


3 Press SHIFT key and F4 [TEST] to test the estimation motion with the range you set. The robot will move using the current override. Not that changing override after starting the TEST will not change the motion speed.
4 Press the [PREV] key to return to the PAYLOAD ID screen.

NOTE
1 The J3, J5, and J6 axes move during the estimation. J4 moves at the beginning of the estimation.
2 This type of estimation requires that the ARMLOAD settings, including the center of gravity, are correct. Make sure that the ARMLOAD is set correctly before executing the estimation. If you change an ARMLOAD setting, be sure to cycle power for the setting to take effect.
3 Set the motion range so that the center of the range is 0 degrees. The farther away the center, the lower the estimation accuracy.
4 There are minimum widths of the motion ranges; setting one side of the range may automatically change the other to maintain the minimum width.
5 If the motion range is too wide, it will fail to create an estimation motion. Run a TEST to confirm.

9.2.3 Performing Estimation and Applying the Result

After setting up and testing estimation motion, run the estimation from the PAYLOAD ID screen using the following procedure.



1 Set the teach pendant ON/OFF switch to OFF, and press F4, EXEC. The message "Robot moves and estimates. Ready?" will be displayed.

> ⚠ **WARNING**
> In the next step, the robot will move. Make sure all personnel and unnecessary equipment are out of the workcell and that all safeguards are in place; otherwise, you could injure personnel or damage equipment.

2 Specify whether to execute load estimation.
- To perform load estimation by running the robot, press F4, YES.
- To quit execution without performing estimation, press F5, NO.

3 After low-speed and high-speed operations are finished, load information is estimated. When the estimation is complete, the "*****" value of the PAYLOAD ESTIMATION ***** item changes to DONE. The system automatically changes from low speed to high speed motion.

> ⚠ **WARNING**
> Even when the robot is running at low speed, do not get close to it, because the robot will start running at high speed; otherwise, you could injure personnel or damage equipment.

4 Press F5, APPLY to set the estimated value to a load setting schedule No. The message "Path and Cycle time will change. Set it?" will be displayed .

5 Specify whether to set the estimated result.
- To set the estimate, press F4, YES.
- To stop setting the estimate, press F5, NO.

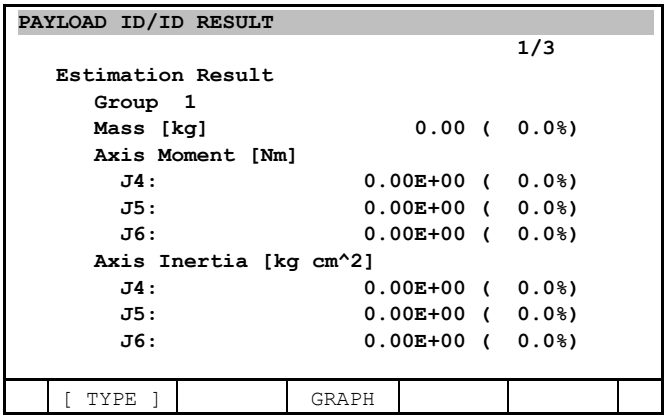
6 If the value to be set is greater than the maximum allowable load (indicated in parentheses), the message "Load is OVER spec! Accept?" will be displayed. Specify whether to set this value, just as in the step above.

NOTE
If the estimation is interrupted, such as paused by external input such as IMSTP or HOLD, or by the Emergency Stop button or HOLD key, or terminated by CSTOPI or ABORT from the function menu, it changes to “ABORT”. The payload is not estimated, so do not “APPLY” and execute the estimation again.
If the estimation fails, any or all of the applied results (payload, payload center, payload inertia) would be 0. Please check the PAYLOAD SET screen of the applied schedule number after APPLY.
If mass is very small, mass is not specified, or moment around J5 or J6 is small, estimation might fail. Please refer to the NOTE at the subsection “Preparation”, to see if the load fulfills the conditions necessary to estimate.

9.2.4 Estimation Result

Estimation results are displayed as Previous Estimates value on the PAYLOAD ID screen. On the PAYLOAD ID/ID Result screen, the estimation results also display percentages of the maximum allowed values.

1 Press the [NEXT] key, and press F5, RESULT.
The estimation results will be displayed along with percentages of the maximum allowed values.



2 Press F3, GRAPH to display the percentages as a bar graph.

9.3 CALIBRATION PROCEDURE (for 6-Axis Robots)

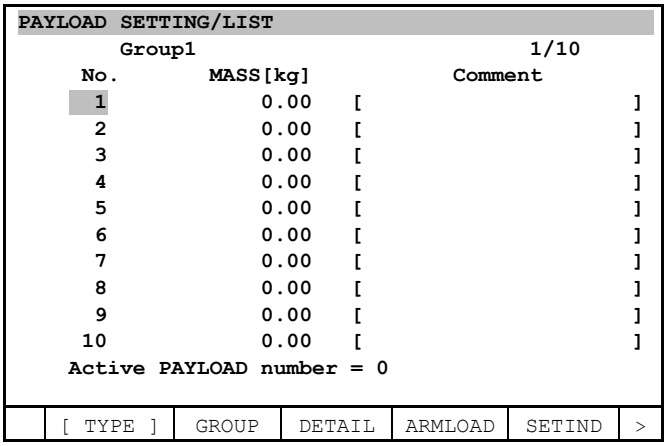
Calibration is required when mechanical parts such as motors are replaced.
If calibration is not performed even though those parts have been replaced, the accuracy of the estimation will deteriorate. You perform calibration on the PAYLOAD ID screen. You perform calibration by setting the calibration switch to ON and executing load estimation, as described in the following procedure.

1 Make sure that nothing is attached to the faceplate of the robot.
Calibration must be performed with nothing attached to the robot.

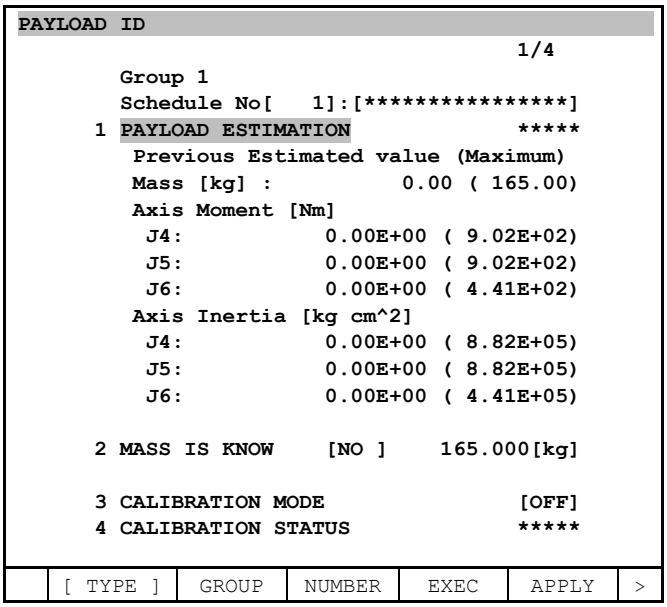
NOTE
If calibration is performed with anything attached to the faceplate of the robot, incorrect calibration data will be set up, hindering normal estimation.
In this case, remove any attachments to the robot and perform calibration again.

2 Press the [MENU] key.

3 Select SYSTEM on the next page.
4 Press the F1, [TYPE].
5 Select Motion. The PAYLOAD SETTING/LIST screen will be displayed. If any other screen is displayed, press the [PREV] key several times until the list screen is displayed. In a multi-group system, press the F2, GROUP to display the list screen for another group.



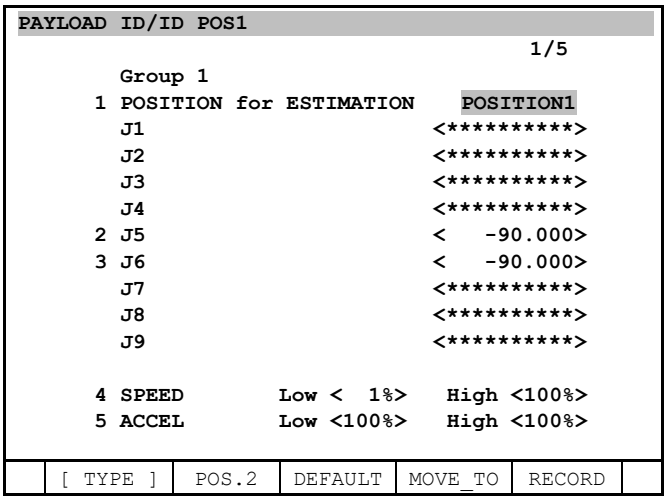
6 Press the [NEXT] key, and then F2, IDENT. The PAYLOAD ID screen will be displayed.



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

NOTE
1 During load estimation, only the J5 and J6 axes move. The other axes maintain the positions they were in when estimation started and do not move.
The motion range is between two points set in the PAYLOAD ID/ID POS1 and PAYLOAD ID/ID POS2 screens. See Steps 9, 10, and 12.)
2 Position the J5 axis so that the axis of rotation is horizontal.
The closer the J5 rotational axis is to vertical, the less precise the estimation will be.

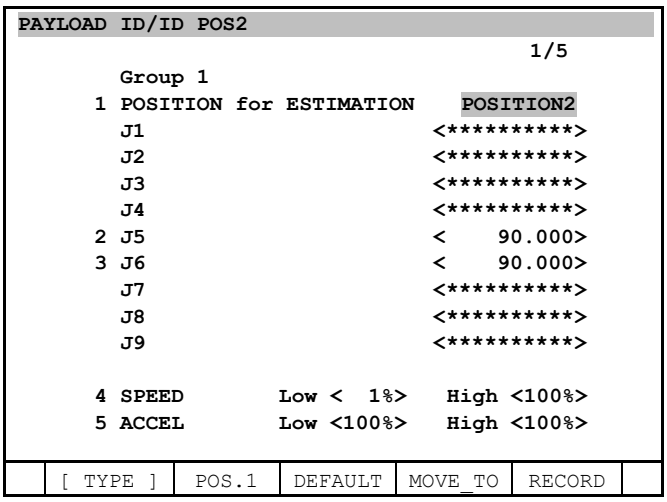
8 Press the [NEXT] key, then F4, DETAIL. The PAYLOAD ID/ID POS1 screen will be displayed.



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

10 Before you move the robot to estimation position 1, make sure that it is safe to move the robot to estimation position 1. If it is dangerous to move the robot to estimation position 1, jog the J1 to J4 axes to move the robot to a position where the robot can move safely. To move the robot to estimation position 1, press the [SHIFT] key + F4, MOVE_TO.

11 To display the PAYLOAD ID/ID POS2 screen, press the F2, POS.2.



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

13 Press the [PREV] key to return to the PAYLOAD ID screen.

14 Move the cursor to CALIBRATION MODE to turn it on.
A message will confirm whether to “Change to calibration mode?” will be displayed. Select YES.
Then, the message “Please execute without any load on the wrist.” Will be displayed. Select OK.
If you select NO or CANCEL CALIBRATION MODE will be reset to off.

NOTE
When calibration is finished, the CALIBRATION MODE item will automatically be changed to off.
Do not change the CALIBRATION MODE item during calibration or load estimation. Otherwise, calibration will be incorrect or incomplete.

15 Move the cursor to CALIBRATION STATUS, set the teach pendant ON/OFF switch to OFF, and press F4, EXEC. The message "Robot moves and estimates. Ready?" will be displayed.

⚠ WARNING
In the next step, the robot will move. Make sure all personnel and unnecessary equipment are out of the workcell and that all safeguards are in place; otherwise, you could injure personnel or damage equipment.

16 Specify whether to perform load estimation.
- To perform load estimation by running the robot, press the F4, YES.
- To quit execution without performing estimation, press the F5, NO.

17 Calibration is completed when the low-speed and high-speed motions are finished. The robot automatically changes from low speed to high speed motion. Even when the robot is running at low speed, do not get close to it, because otherwise you may get in a dangerous situation when the robot suddenly starts running at high speed.

The calibration procedure for the 3 axes waves estimation type is basically the same as in the previous procedure. Set the Motion Pattern item to 3 axes waves and set the motion range instead of positions. Make sure that no payload is on the faceplate or the arm of the robot, armload is correctly set to 0, and that you cycle controller power after changing armload settings.

9.4 OTHER RELATED MATTERS

9.4.1 Motion range

A narrower motion range between positions 1 and 2 may result in poor estimation precision.
Set the motion range as wide as the default motion range, if possible.

9.4.2 Calibration data

You can confirm and set calibration data on the System Config screen.
If improper calibration data is set up, for example, by performing calibration with an attached load by mistake, resetting the previous data on the System Config screen can restore the previous calibration data. Keep a note of the previous calibration data so that it can be recovered if necessary.

9.4.3 Payload setting for estimating motion

If the payload setting data while estimating motion is far different from the actual payload, the estimating motion may cause oscillation. To stabilize the estimating motion, set up a rough estimate of the payload data before executing estimation.
Setting rough payload data in advance may help improve the estimation precision, especially for payloads that do not have sufficient moments around the J5 and J6 axes.

9.4.4 Collision Detection for estimating motion

If payload settings during estimation are different from the actual payload, it may cause false collision detection because the motion may vibrate and the difference from the internal motion model may be too large. If a false collision detection occurs, use a rough payload estimate in advance, as described in Section 9.4.4.

10 PAYLOAD OVER AND PAYLOAD SETTING CONFIRM FUNCTION

This chapter describes the Payload over and payload setting confirm function. This function is a part of "Payload identification function" (J669) option.
However, "packages that include Payload identification function" do not include this function.

Payload over and payload setting function adds a screen, "PayLoadCheck" in pull-up menu of system screen.
In "PayLoadCheck" screen, payload over confirm function and payload setting confirm function can be used.

Payload Over Confirm Function

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

NOTE
The overload warning will not be shown even if the maximum torque of the other program is higher than the rated torque.

Payload Setting Confirm Function

This function evaluates payload setting accuracy with a little robot motion.
With this function, the payload setting can be confirmed after a robot is installed on production line.
It may be difficult to move robot widely on production lines. This function roughly estimates a payload by moving J5 and J6 axes slightly, and compares with the value of payload setting and shows an error.
When this function is used, robot has to be in the position where a large moment acts on J5 and J6 axes. J5 and J6 axes have to have the almost maximum moment for the current payload acting on them, but other axes have no limitation.
This function moves J5 and J6 axes between current position+1[degree] and current position–1[degree].

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

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

10.1 LIMITATIONS

Group

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

Robot Type

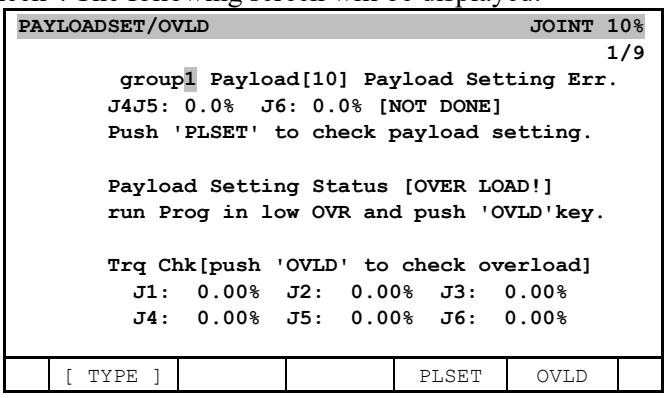
This function supports robot with 6 rotary axes robots.

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

10.2 PROCEDURE

10.2.1 Procedure to Show Payload Over and Payload Setting Confirm Screen

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

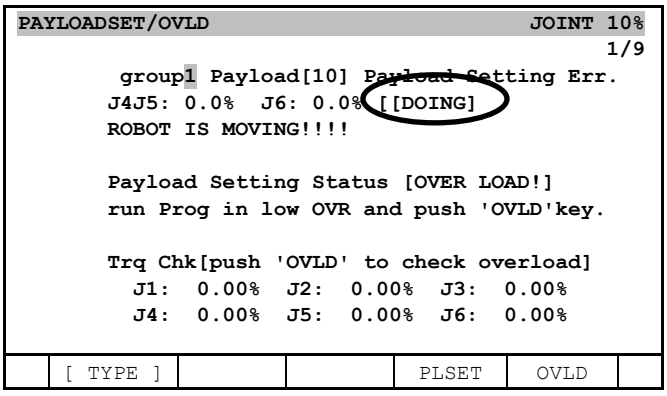


10.2.2 Procedure to Use Payload Setting Confirm (When Payload is Changed)

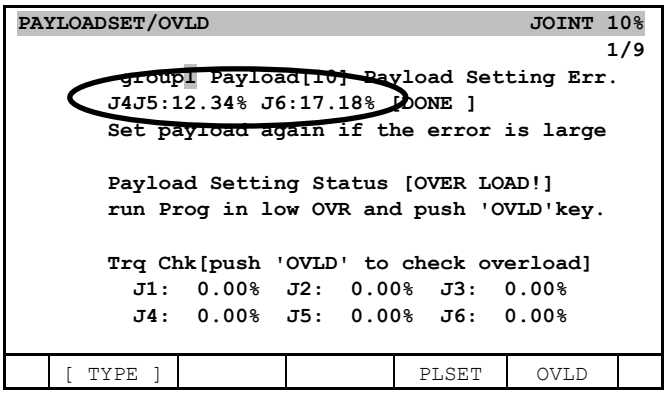
1 Abort all programs which are running or paused.
2 Move robot to the position where J5 and J6 have almost the maximum moment for the current payload acting on them.

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

3 Turn off the teach pendant ON/OFF switch.
4 Display payload over and payload setting confirm screen. (Refer to the subsection 10.2.1.)
5 Press the F4(PLSET) key.
6 Press the F3(yes) key.
7 Wait until the right column of the line below “Payload Setting Err.” becomes ”DONE” from “DOING”.(about 30 seconds)



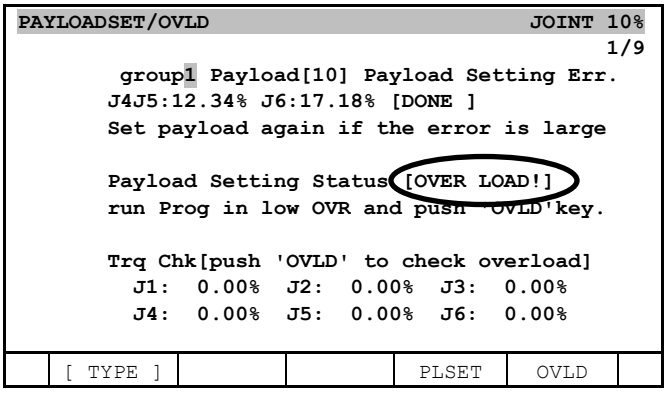
8 From “Payload Setting Err.”, you can confirm whether a payload setting is done correctly. Threshold to decide that a payload setting is done correctly is 10%.



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

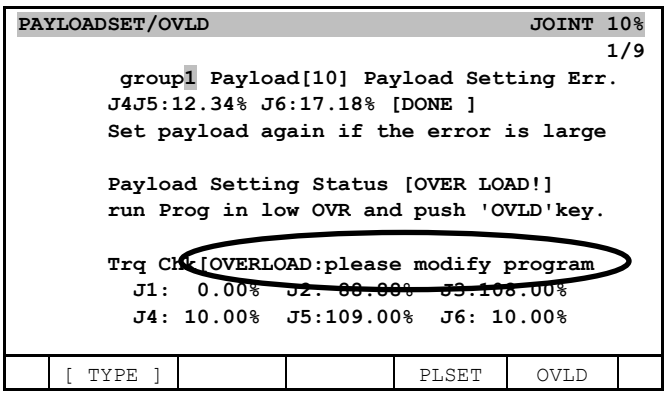
10.2.3 Procedure to Use Payload Over Confirm (When The Program or Payload is Changed)

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

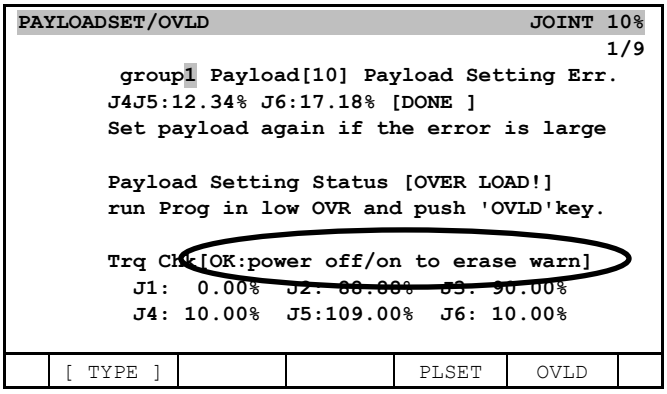


2 The program to be payload over confirmed has to be executed beforehand. Please select the program to be payload over confirmed. (Push [select] key, select program and press the [ENTER] key.)
3 Set override 10% or lower. And execute the program.
4 Then display payload over and payload setting confirm screen. (Refer to the subsection 10.2.1.)

5 Push the F5(OVLD) key.
6 Message to ask you to modify program is shown when “Trq Chk” became “OVERLOAD”.



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



NOTE

Overload warning is disabled when the maximum torque of checked program is lower than the rated torque. Note that overload warning is not shown even if maximum torques of the other programs are higher than the rated torque.
It must be confirmed that the maximum torque is lower than the rated torque in all of the programs used.
Please change setting of MOTN-171 in setup/accuracy table screen from “NODISP” to “DEFAULT” to enable overload warning again.
Result shown in the screen that is displayed while the program is still running is not correct. Please display the screen again after the program to evaluate has ended.

11 GRAVITY COMPENSATION

Gravity compensation improves absolute positioning accuracy by calculating the amount of robot arm bending caused by the weight of a tool or workpiece on the flange, equipment on the arm, or the arm's own weight. Gravity Compensation then corrects the motor position by the calculated amount of bending, and it improves the absolute position accuracy.

The Gravity Compensation option (J649) is required to use the Gravity Compensation function.

11.1 SETUP

Enabling Gravity Compensation

To enable Gravity Compensation, display the SYSTEM / CONFIG screen and select the following:
Gravity Compensation function -> Group n -> Enable this function

Gravity Compensation is disabled when the robot is shipped, except for some specific system configurations. To enable or disable Gravity Compensation, set this item and then cycle controller power.

Gravity Compensation can also be enabled or disabled with the following Data ID.
(Data ID: $GRAVITY_COMPENSATION.GRAVITY_COMPENSATION_ENABLE[a]
/ BOOLEAN / RW / a: group number)

Payload / Armload settings

As mentioned at the beginning of this chapter, Gravity compensation corrects for the amount of bending of the robot arm caused by the load attached to the robot. Therefore, you must set payload and armload information correctly for Gravity Compensation to work properly.
For details on payload setting, refer to the "Payload Setting" chapter in the *OPERATOR'S MANUAL (Basic Function)* (B-84464EN) or the appropriate *MECHANICAL UNIT OPERATOR'S MANUAL*.

11.2 MASTERING

11.2.1 Normal Mastering and GC Mastering

The following mastering methods are available:

(1) Normal Mastering : Mastering with Gravity Compensation DISABLED
(2) GC Mastering : Mastering with Gravity Compensation ENABLED

- Since the mastering result obtained by each method is different, you should choose the same method as the one you choose when previously mastering.
- Although Gravity Compensation works with either type of mastering, GC Mastering provides better absolute position accuracy than Normal Mastering.
- If you change the mastering method from Normal Mastering to GC Mastering, the TCP will shift slightly. Therefore, confirm in advance that the TCP shift will cause no problem. When switching to GC mastering, use Vision Mastering or Jig Mastering, not Single Axis Mastering. For details on Vision Mastering, refer to *iRCalibration OPERATOR'S MANUAL*.

Mastering method used at the time of robot shipment

For some specific system configurations, GC Mastering has been done at the time of robot shipment. You can verify this information on the Inspection Data Sheet attached to your robot controller. Refer to the following table.

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

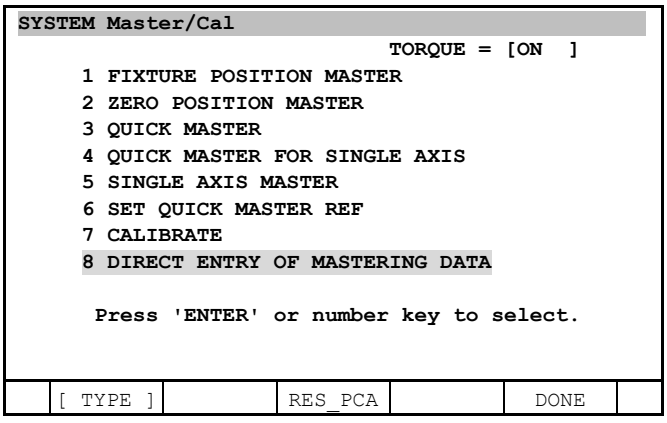
11.2.2 Checking and Choosing the Mastering Method

Use Procedure 11-1 to check which mastering method the current mastering data is derived from and choose the appropriate mastering method of Gravity Compensation.

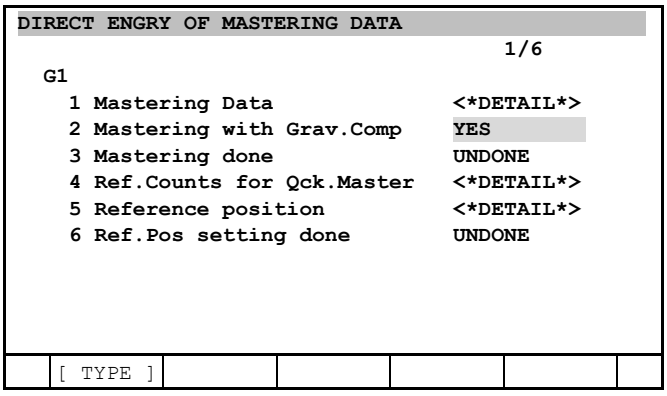
Procedure 11-1 Checking and Choosing the Mastering Method

Step

1 Press MENU.
2 Select SYSTEM.
3 Press F1, [TYPE].
4 Select Master/Cal.
If Master/Cal is not listed on the [TYPE] menu, do the following; otherwise, continue to Step 5.
   a Select Config from the [TYPE] menu.
   b Select Mastering settings.
   c Set Master/Cal screen to DISPLAY.
   d Press F1, [TYPE].
   e Select Master/Cal. The following screen will be displayed.



5 Select DIRECT ENTRY OF MASTERING DATA. The Direct Entry of Mastering Data screen will be displayed.



6 Check the setting of the Mastering with Grav.Comp item and choose the appropriate mastering method. Refer to Table 11.2.2 (a).

Table 11.2.2 (a) Mastering with Grav.Comp settings

Value	Description	Mastering method you should choose
YES	Current mastering data is derived from GC Mastering.	• Choose GC Mastering.
NO	Current mastering data is derived from Normal Mastering.	• Basically, choose the Normal Mastering. • If you need to improve absolute position accuracy, choose GC Mastering.
UNKNOWN	Unknown (not set).	• Choose the same one as “Mastering method at the shipment of the robot” which is described above.

This value of the Mastering with Grav.Comp item updated each time you execute mastering.

11.2.3 Mastering Procedure

Mastering procedure for Normal Mastering

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

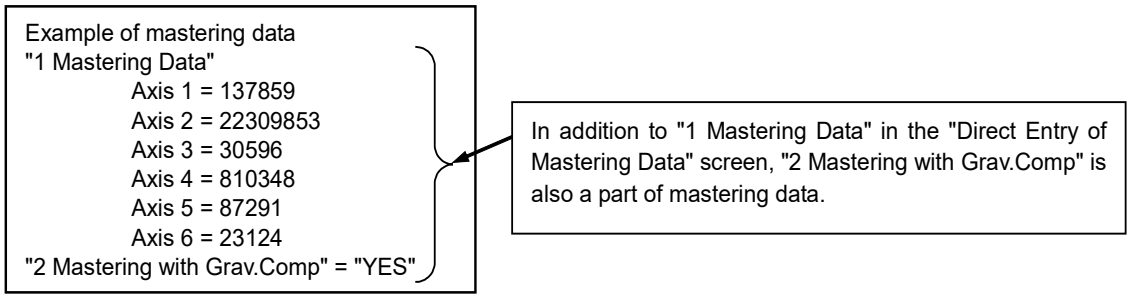
Mastering procedure for GC Mastering

1 Perform mastering with Gravity Compensation ENABLED.
  * Refer to the next section for guidance.

After mastering has been executed, record (write down or back up) the values of the Mastering with Grav.Comp and Mastering Data items on the DIRECT ENTRY OF MASTERING screen [SYSTEM / Master/Cal / [TYPE] / DIRECT ENTRY OF MASTERING DATA].
If the mastering data has been lost, you may be able to restore the mastering data by directly entering the mastering data you wrote down. For more information on direct entry of mastering data, refer to the Mastering chapter in Appendix B of the *OPERATOR'S MANUAL (Basic Function)* (B-84464EN).

NOTE

When you use the Gravity Compensation function, you must treat the value of the Mastering with Grav.Comp item as a part of the mastering data. You need this value to manually enter the mastering data.



11.2.4 GC Mastering Guidance

Use the following GC Mastering rules to obtain correct mastering data..

1 Before you perform GC Mastering, set correct payload parameters on the PAYLOAD SETTING screen.
2 When you write down the joint angles for a reference position that will be used for mastering, do it under the following conditions:
- Gravity Compensation is ENABLED, and
- Calibration is completed, and
- Correct payload parameters are set.
3 When you perform Single Axis Mastering, observe the following rules:
- Just to be safe, go to the DIRECT ENTRY OF MASTERING screen [SYSTEM / Master/Cal / DIRECT ENTRY OF MASTERING DATA] and then write down the Mastering Data of all axes before you perform mastering.
- If there is more than one axis that needs to be mastered, you must select and master them all at once.
  For example) Single Axis Mastering after J2, J4, and J5 motor replacement
  You must master the J2, J4, and J5 axes all at once.
  You cannot master only the J2 axis when J4 and J5 axes also need to be mastered.
  * If you have no way to master multiple axes all at once, perform the following steps:
    NOTE: Master in ascending order. In other words, from robot base to flange.
  (1) Master J2, J4, and J5 all at once. You need to master J2 correctly. At this time, you do not need to master J4 and J5 correctly, but master them moderately correctly. In other words, do temporary mastering for J4 and J5.
  (2) Master J4 correctly.
  (3) Master J5 correctly.
- If you need to move an axis that does not need mastering, but have already documented its reference position, move it after performing temporary mastering of the axis that requires mastering.
  For example) Single Axis Mastering using reference angles of J1, J2, J4, J5, J6 after J3 motor replacement:
  (1) Master J3 moderately correctly, then CALIBRATE.
  (2) Move the axes that do not require mastering (J1, J2, J4, J5, J6) to the joint angles you wrote down for reference positions.
  (3) Master J3 correctly.
- If you perform “mastering after temporary mastering” as described above, the mastering data of the axes that have an interaction with the recently mastered axis will change. This causes

unexpected position change. Therefore, after performing mastering, verify that the mastering data of axes that do not require mastering keep their original values. If the values have been changed, enter the original values that you wrote down in advance.

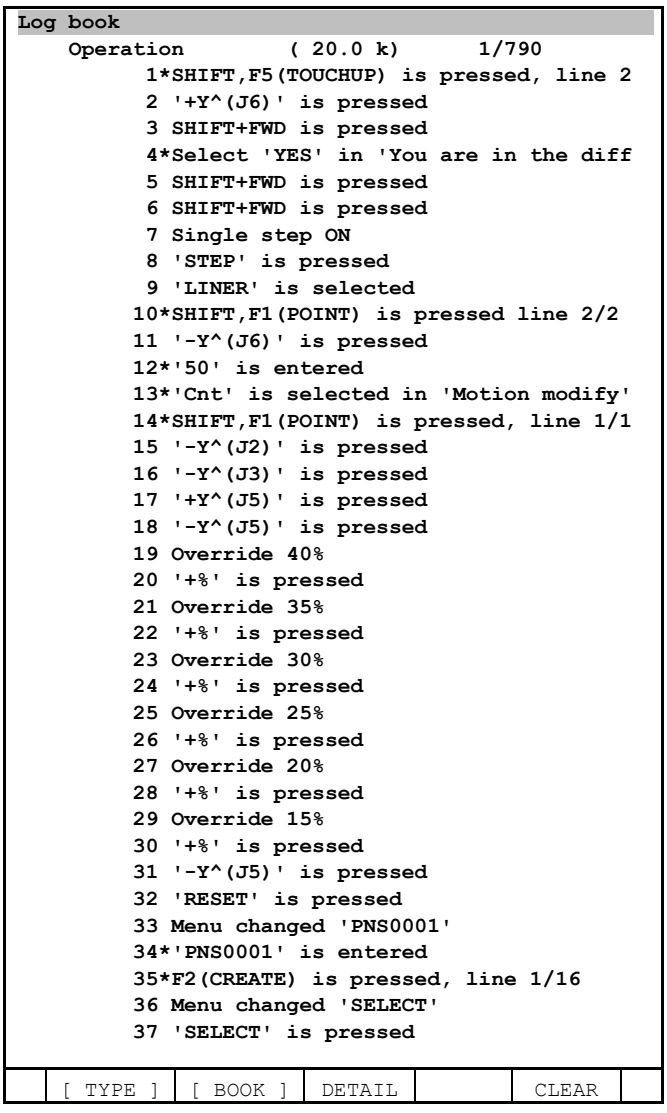
12 OPERATION LOG BOOK

Operation Log Book automatically records teach pendant operations and alarms in a buffer that can be displayed in the Log Book menu of the teach pendant or saved as a text file. Some operations can include a screen image to help in analyzing the event.
A system can have multiple Log Books and you can specify which events are recorded in each Log Book. In this way, frequent and infrequent events can be recorded to separate Log Books.
When a Log Book buffer is full and a new record is added, the oldest record is deleted. The number of events that can be stored in a Log Book buffer varies based on the type of events saved because not all events are the same size. You can change the size of the buffer by setting the “Size” configuration item of “Book Settings” on the Log Book Settings screen.

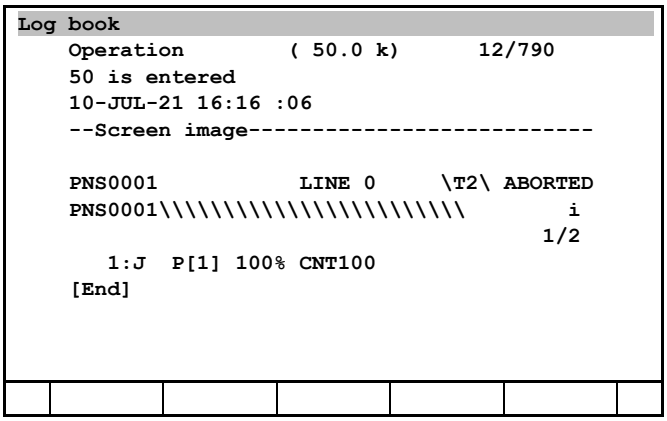
To use function, Operation Log Book option (J695) is required.

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

See the following example of a Log Book.



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



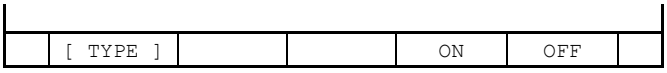
General Limitations

Note the following general limitations for Log Book:

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

  Example

  If the function key line is as follows:



  F3 has no label. If you press the F3 and a book is selected in the "Function key" configuration item, "F3( )" is pressed" is recorded.

- If you change the remote TCP number in remote TCP jog mode, this is recorded as a change of user frame.
- If you changed current JOG coordination by parameter instruction, the change is recorded when the group is selected.
- Screen image doesn't support double height font, which is used in the "on the fly" screen of ArcTool. It is recorded as two lines which have the same characters.

  Example

  0.0 Amps
  in "On the fly" screen is recorded as

```
0.0 Amps
0.0 Amps
```

- For KAREL read instructions, screen image is recorded just after input.
- There is possibility that the function won't work in the screen users made with KAREL.
- KCL is not supported.
- Screen changes caused by the automatic backup function are not recorded.
- "'x' is saved" and "'x' is loaded" may be recorded by internal process.

12.1 RECORDED EVENTS

The following is the list of all events that can be recorded by the Log Book function.
Recording of individual events can be disabled by setting the "Logged Events" configuration item on Log Book Settings screen. If the "Not record" is selected, the event is not recorded. Some of them are disabled by default. If the "Book x" is selected, the value x indicates the Log Book No. to which the event is recorded.
Some events have a screen image. This information helps to analyze the record. However, screen images take a lot of space in the Log Book; you can disable the recording of screen images by setting the "Screen image" configuration item of "Logged Events".

Refer to "12.3 OPERATIONS" later in this section.

Table 12.1 Events recorded by log book

Message	Event	Configuration item / Default	Save Screen / Default
Alarm history	When an alarm occurs, the alarm message is recorded. This record is the same as the record of the alarm history menu. To choose the alarms to be recorded, the 'filtering' function is provided.	Alarm / Book 1 (Data ID: $OPERATION_LOG_BOOK.ALARM_BOOK_NUMBER / INTEGER / RW)	None
'x' is entered	When a value or a word is entered, this is recorded. The 'x' in the message is the entered value or word. If screen image is enabled for this event, it will contain the previous value. Values will be recorded even if they are invalid. If you cancel an input, it is not recorded.	Value or word / Book 1 (Data ID: $OPERATION_LOG_BOOK.ENTRY_BOOK_NUMBER / INTEGER / RW)	Screen image / TRUE (Data ID: $OPERATION_LOG_BOOK.ENTRY_IMAGE_BOOK_NUMBER / BOOLEAN / RW)

Message	Event	Configuration item / Default	Save Screen / Default
'x' is selected 'x' is selected in 'y' menu	When a menu item is selected, this is recorded. The 'x' in the message is the selected item. If the menu has a title, the 'y' in the message shows the title. If screen image is enabled for this input, it will show the screen just before opening the menu.	Menu item / Book 1 (Data ID: $OPERATION_LOG_BOOK.MENU_ITEM_BOOK_NUMBER / INTEGER / RW)	Screen image / TRUE (Data ID: $OPERATION_LOG_BOOK.MENU_ITEM_IMAGE_BOOK_NUMBER / BOOLEAN / RW)
'x' is selected in 'y' window	When an item is selected in warning window, this is recorded. The 'x' in the message is the selected item. The 'y' in the message is the beginning of the message in the warning window. If screen image is enabled for this event it will show the warning window.	Warning window / Book 1 (Data ID: $OPERATION_LOG_BOOK.WARNING_BOOK_NUMBER / INTEGER / RW)	Screen image / TRUE (Data ID: $OPERATION_LOG_BOOK.WARNING_IMAGE_BOOK_NUMBER / BOOLEAN / RW)
'x' is selected in MENU 'x' is selected in FCTN	When an item is selected by [MENU] key or FCTN key, this is recorded. The 'x' in the message is the selected item.	MENU or FCTN / Book 1 (Data ID: $OPERATION_LOG_BOOK.MENU_OR_FCTN_BOOK_NUMBER / INTEGER / RW)	None
JOG menu TOOL 'x' JOG menu USER 'x' JOG menu JOG 'x' JOG menu GROUP 'x' JOG menu ROBOT JOG menu EXT	Operations of the JOG menu are recorded with these messages.	JOG menu / Book 1 (Data ID: $OPERATION_LOG_BOOK.JOG_MENU_BOOK_NUMBER / INTEGER / RW)	None
Menu changed 'x'	When a menu is changed, this is recorded. The 'x' in the message is the title of the new menu. Changes to sub menus such as Config or Detail are not recorded.	Menu changes / Book 1 (Data ID: $OPERATION_LOG_BOOK.MENU_CHANGES_BOOK_NUMBER / INTEGER / RW)	None

Message	Event	Configuration item / Default	Save Screen / Default
'x' is pressed	When a key is pressed, this message is recorded with the key name. It is not recorded when the key is released. If SHIFT is held when the key is pressed, the word 'SHIFT,' is added to the key name (except SELECT, EDIT, DATA). The keys are grouped as follows, and you can disable recording of each group.		
	F1, F2, F3, F4, F5	Function key / Book 1 (Data ID: $OPERATION_LOG_BOOK.FUNCTION_KEY_BOOK_NUMBER / INTEGER / RW)	Screen image / TRUE (Data ID: $OPERATION_LOG_BOOK.FUNCTION_KEY_IMAGE_BOOK_NUMBER / BOOLEAN / RW)
	+X(J1), -X(J1), +Y(J2), -Y(J2), +Z(J3), -Z(J3), +X^(J4), -X^(J4), +Y^(J5), -Y^(J5), +Z^(J6), -Z^(J6) +(J7), -(J7), +(J8), -(J8)	JOG key / Book 1 (Data ID: $OPERATION_LOG_BOOK.JOG_KEY_BOOK_NUMBER / INTEGER / RW)	None
	SELECT, EDIT, DATA,	Program key / Book 1 (Data ID: $OPERATION_LOG_BOOK.PROGRAM_KEY_BOOK_NUMBER / INTEGER / RW)	None
	UF1, UF2, UF3, UF4, UF5, UF6, UF7	User key / Book 1 (Data ID: $OPERATION_LOG_BOOK.USER_KEY_BOOK_NUMBER / INTEGER / RW)	None
	+%, -%, COORD	Override key / Book 1 (Data ID: $OPERATION_LOG_BOOK.OVERRIDE_KEY_BOOK_NUMBER / INTEGER / RW)	None
	FWD, BWD	FWD key / Book 1 (Data ID: $OPERATION_LOG_BOOK.FWD_KEY_BOOK_NUMBER / INTEGER / RW)	None

Message	Event	Configuration item / Default	Save Screen / Default
	HOLD	HOLD key / Book 1 (Data ID: $OPERATION_LOG_BOOK.HOLD_KEY_BOOK_NUMBER / INTEGER / RW)	None
	STEP	STEP key / Book 1 (Data ID: $OPERATION_LOG_BOOK.STEP_KEY_BOOK_NUMBER / INTEGER / RW)	None
	PREV	PREV key / Book 1 (Data ID: $OPERATION_LOG_BOOK.PREV_KEY_BOOK_NUMBER / INTEGER / RW)	None
'x' is pressed	ENTER	ENTER key / Book 1 (Data ID: $OPERATION_LOG_BOOK.ENTER_KEY_BOOK_NUMBER / INTEGER / RW)	None
	ITEM	ITEM key / Book 1 (Data ID: $OPERATION_LOG_BOOK.ITEM_KEY_BOOK_NUMBER / INTEGER / RW)	None
	RESET	RESET key / Book 1 (Data ID: $OPERATION_LOG_BOOK.RESET_KEY_BOOK_NUMBER / INTEGER / RW)	None
	DIAG/HELP	DIAG/HELP key / Book 1 (Data ID: $OPERATION_LOG_BOOK.HELP_KEY_BOOK_NUMBER / INTEGER / RW)	None

Message	Event	Configuration item / Default	Save Screen / Default
Override x%	When override is changed, this is recorded. The 'x' in the message is the new override value. This records all change of override by any method. For example, '+%' key, TP is enabled, override instruction of program.	Override / Book 1 (Data ID: $OPERATION_LOG_BOOK.OVERRIDE_BOOK_NUMBER / INTEGER / RW)	None
JOINT coordinate USER coordinate TOOL coordinate JOG coordinate PATH coordinate	When coordinate is changed, this is recorded. This records all change of coordinate by any method.	Coordinate / Book 1 (Data ID: $OPERATION_LOG_BOOK.COORDINATE_BOOK_NUMBER / INTEGER / RW)	None
Single step ON Single step OFF	When single step is changed, this is recorded. This records all change of single step by any method.	Single step / Book 1 (Data ID: $OPERATION_LOG_BOOK.STEP_BOOK_NUMBER / INTEGER / RW)	None
Motion group x	When motion group is changed, this is recorded. The 'x' in the message is the new motion group. This records all change of motion group by any method.	Motion group / Book 1 (Data ID: $OPERATION_LOG_BOOK.MOTION_GROUP_BOOK_NUMBER / INTEGER / RW)	None
Sub group ROBOT Sub group EXT	When sub group is changed, this is recorded. This records all change of sub group by any method.	Sub group / Book 1 (Data ID: $OPERATION_LOG_BOOK.MOTION_SUB_GROUP_BOOK_NUMBER / INTEGER / RW)	None
User frame x	When user frame number is changed, this is recorded. The 'x' in the message is the new user frame number. This records all change of user frame number by any method.	User frame number / Book 1 (Data ID: $OPERATION_LOG_BOOK.USER_FRAME_BOOK_NUMBER / INTEGER / RW)	None
Tool frame x	When tool frame number is changed, this is recorded. The 'x' in the message is the new tool frame number. This records all change of tool frame number by any method.	Tool frame number / Book 1 (Data ID: $OPERATION_LOG_BOOK.TOOL_FRAME_BOOK_NUMBER / INTEGER / RW)	None

Message	Event	Configuration item / Default	Save Screen / Default
Save file x Load file x	File x is saved. File x is loaded. These events are ONLY for file save/load in following screens. • File screen • Program list screen	File / Book 1 (Data ID: $OPERATION_LOG_BOOK.FILE_BOOK_NUMBER / INTEGER / RW)	None
WAIT is released (x, y)	The waiting WAIT instruction is canceled by WAIT RELEASE in program x line y.	Wait release / Not record (Data ID: $OPERATION_LOG_BOOK.WAIT_RELEASE_BOOK_NUMBER / INTEGER / RW)	None
Create program x Delete program x Write line x of y Delete line x of y	TP program x is created. TP program x is deleted. Line x of TP program y is written. Line x of TP program y is deleted.	TP Program / Not record (Data ID: $OPERATION_LOG_BOOK.TP_PROGRAM _BOOK_NUMBER / INTEGER / RW)	None
Write P[x] of y	Position data P[x] of TP program y is written. These are also recorded when program is changed internally.	Position data / Not record (Data ID: $OPERATION_LOG_BOOK.POSITION_BOOK_NUMBER / INTEGER / RW)	None
TP 'x' ON TP 'x' OFF	This records the low level key operation. All key operations of Teach Pendant are recorded. When a key is pressed, 'TP x ON' is recorded. When a key is released, 'TP x OFF' is recorded. ENABLE switch and E-STOP are also recorded. [SHIFT] key is treated as normal key. CRT operations are not recorded by this event. 'x' in the message is key name, the following keys are recorded. ENABLE, ESTOP PREV, F1, F2, F3, F4, F5, NEXT SHIFT, MENU, SELECT, EDIT, DATA,FCTN, i UP, DOWN, LEFT, RIGHT, DISP RESET, STEP, BACKSPACE, ITEM, HOLD, FWD, BWD, COORD, +%, -%, GROUP 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, -/,, ., ENTER, HELP/DIAG, UF1, UF2, UF3, UF4, UF5, UF6, UF7, +X(J1), -X(J1), +Y(J2), -Y(J2), +Z(J3), -Z(J3), +X^(J4), -X^(J4), +Y^(J5), -Y^(J5), +Z^(J6), -Z^(J6), +(J7), -(J7), +(J8), -(J8)	All key / Not record (Data ID: $OPERATION_LOG_BOOK.ALL_TP_KEYS_BOOK_NUMBER / INTEGER / RW)	None

Message	Event	Configuration item / Default	Save Screen / Default
Cold start Power failure recovery Controlled start	Start mode is recorded.	Start mode / Book 1 (Data ID: $OPERATION_LOG_BOOK.START_MODE_BOOK_NUMBER / INTEGER / RW)	None
Run FWD w,x (TASK:y,TB:z) Run BWD w,x (TASK:y,TB:z) Dry Run w,x (TASK:y,TB:z) Ignored w,x (TASK:y,TB:z) Paused w,x (TASK:y,TB:z) Aborted w,x (TASK:y,TB:z)	The TP program execution is recorded. In the message, the 'w' is program name, the “x" is the line number, the "y" is the task number, and the "z" is the time.	TP program execution / Book 2 (Data ID: $OPERATION_LOG_BOOK.PROGRAM_EXECUTION_BOOK_NUMBER / INTEGER / RW)	None
Focus changed to x	Changing focus screen is recorded. The 'x' in the message is the title of the new menu.	Focus screen / Book 1 (Data ID: $OPERATION_LOG_BOOK.FOCUS_WINDOW_BOOK_NUMBER / INTEGER / RW)	None
Refer to OPERATOR7S MANUAL (Collaborative Robot Function) (B-83744) for detail.	Refer to OPERATOR7S MANUAL (Collaborative Robot Function) (B-83744) for detail.	Collaborative Robot / Book 4 (Data ID: $OPERATION_LOG_BOOK.COLLABORATIVE_STATUS_BOOK_NUMBER / INTEGER / RW)	None
None	Certain function keys, such as [TYPE] and [DETAIL], will no longer be recorded.	Function key filtering / FALSE (Data ID: $OPERATION_LOG_BOOK.IGNORE_FUNCTION_KEY_MASK / BOOLEAN / RW)	None
None	“File x is saved” will no longer be recorded.	Save log filtering / FALSE (Data ID: $OPERATION_LOG_BOOK.IGNORE_SAVE / BOOLEAN / RW)	None

12.2 SETTING UP BOOK

A system can have 16 books at a maximum. User can setup each book by the “Book Settings” configuration item on Log Book Settings screen. Followings are items the user can setup.

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

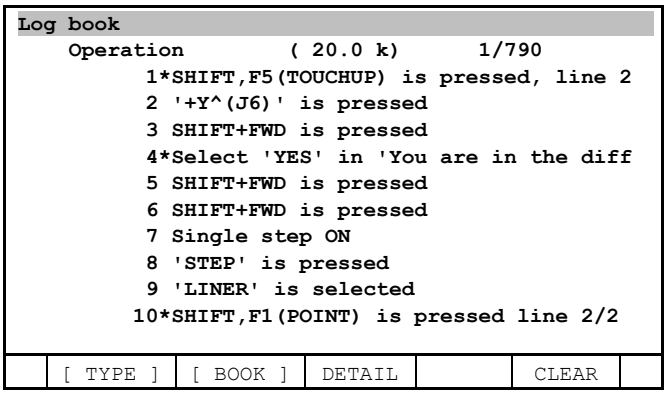
Table 12.2 “Book Settings” configuration item

Configuration item	Default	Range	Description
TITLE (Data ID: $OPERATION_LOG_BOOK.BOOK_TITLE[1-16] / STRING / RW / Array corresponds to book number)	[1-16] ‘’	Max 13 chars	Title of every buffer. This string is displayed on the top line of LogBook menu. It also be displayed in pop up menu of F2([BOOK]) in LogBook menu. If the value is ‘’, the default title is displayed. Default title of BOOK1 is ‘Operation’. Default title of ‘BOOK2 is ‘I/O’. Default title of BOOK3 is ‘BOOK 3’.
SIZE (Data ID: $OPERATION_LOG_BOOK.BOOK_SIZE[1-16] / INTEGER / RW / Array corresponds to book number)	[1] 50 [2] 100 [3-16] 0	0, 100000	Size of the buffer to save log. Unit is K Bytes. One record takes about 300 bytes. If this value is too big, maximum available size of Log Book buffer is allocated automatically. If there is not enough memory to make Log book buffer, SYST – 188 WARN “book(%d) was not created” is posted and the book is not created. Cycle power is needed to enable the change of this variable.
Memory type (Data ID: $OPERATION_LOG_BOOK.BOOK_MEMORY_TYPE[1-16] / INTEGER / RW / Array corresponds to book number)	[1] NVRAM [2] DRAM [3-16] NVRAM	0/1	Memory type of every buffer. 0: In NVRAM, Log is kept at power down. 1: In DRAM, Log is cleared at power down.
Display (Data ID: $OPERATION_LOG_BOOK.DISPLAY_BOOK [1-16] / BOOLEAN / RW / Array corresponds to book number)	[1] TRUE [2] FALSE [3-16] TRUE	FALSE/ TRUE	If FALSE, the buffer is not displayed in pop up menu of F2([BOOK]) in LogBook menu.

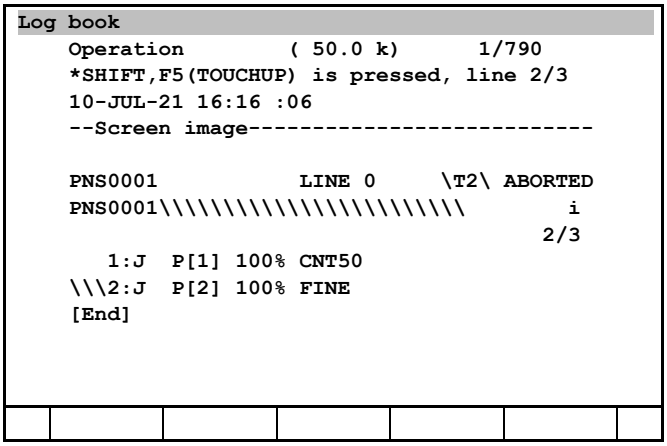
12.3 OPERATIONS

Displaying the Log Book Screen

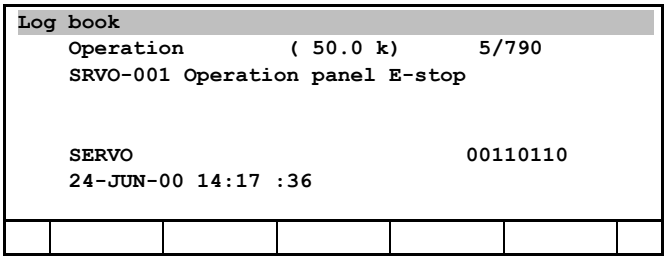
1 Press the [MENU] key.
2 Select 4, ALARM.
3 Press F1, [TYPE].
4 Select Log Book. The following screen will be displayed.



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



7 To view details of an entry, select the entry and press the F3, DETAIL. The following screen will be displayed:



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

Saving Log Book data as a Text File

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

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

See the following example of a LOGBOOK.LS file.

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

Filtering Alarms and Screens

You can set the “Alarm filtering” and “Screen filtering” configuration items on Log Book Settings screen to filter specific alarms and screens.

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

Table 12.3 (a) "Alarm filtering" configuration item

Configuration item	Default	Range	Description
Severity of alarm (Data ID: $OPERATION_LOG_BOOK.ALARM_FILTER_SEVERITY / INTEGER / RW)	PAUSE + ABORT	None/ WARN/ PAUSE/ WARN + PAUSE/ ABORT/ WARN + ABORT/ PAUSE + ABORT/ Full	Filter by severity of error. When the item is selected, the corresponding errors are logged. • WARN: Log warning errors. • PAUSE: Log pause errors. • ABORT: Log abort errors. Ex. PAUSE + ABORT = pause and abort errors are logged. Priority of this is lower than "Type of alarm" and "Alarm codes".
Type of alarm (Data ID: $OPERATION_LOG_BOOK.ALARM_FILTER_TYPE[1-20]/ INTEGER / RW / Array corresponds to the number of filters)	0	-2147483648 2147483647	Filter by type of error. If a positive value is specified, alarms of the specified type are logged. Ex. 11 = SRVO alarms are logged. If a negative value is specified, alarms of the specified type are not logged. Ex. -11 = SRVO alarms are not logged. Priority of this is higher than "Severity of alarm" but lower than "Alarm codes".
Alarm codes (Data ID: $OPERATION_LOG_BOOK.ALARM_FILTER_ITEM[1-100] / INTEGER / RW / Array corresponds to the number of filters)	0	-2147483648 2147483647	Filter by individual error. If a positive value is specified, the specified alarm is logged. Ex. 11001 = SRVO-001 is logged. If a negative value is specified, the specified alarm is not logged. Ex. -11001 = SRVO-001 is not logged. Priority of this is higher than "Severity of alarm" and "Type of alarm".

Control error filtering by setting the following Configuration item.
- Alarm code 1 – Alarm code n
- Severity of alarm
- Type 1 – Type n

The priority of the settings is: Severity of alarm < Type of alarm < Alarm codes

See the following for an example of error filtering:

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

Table 12.3 (b) Alarm filtering example

Severity of alarm	Type of alarm	Alarm codes	SRVO-001	SRVO-038	TPIF-104
PAUSE + ABORT	0, 0, ..	0 ,0, ..	Logged	Logged	Not logged
PAUSE + ABORT	0, 0, ..	9104, 0, ..	Logged	Logged	Logged
PAUSE + ABORT	0, 0, ..	9104, -11001, ..	Not logged	Logged	Logged
Full	-11, 0, ..	11001, 0, ..	Logged	Not logged	Logged
None	11, 9, ..	-11001	Not logged	Logged	Logged

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

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

NOTE
Supplementary explanation TP'x' ON/OFF is not supported.

By default, screen filtering is disabled. Screen filtering is enabled/disabled by "Enable" configuration item in "Screen filtering".

- If "Switch filter" configuration item is TRUE, events on registered screens are NOT recorded. If FALSE, events on registered screens are recorded.
- To register screens, software ID and screen ID must be set to "Screen registration" configuration item in "Screen filtering". By default, no screen is registered for filter.

Table 12.3 (c) "Screen filtering" configuration item

Configuration item	Default	Range	Description
Enable (Data ID: $OPERATION_LOG_BOOK.ENABLE_SCREEN_FILTER / BOOLEAN / RW)	FALSE	FALSE/TRUE	Screen filter is disabled/enabled.
Switch filter (Data ID: $OPERATION_LOG_BOOK.SCREEN_FILTER_BLOCK / BOOLEAN / RW)	TRUE	FALSE/TRUE	TRUE: Events on registered screens are not recorded. Events on other screens are recorded. FALSE: Events on registered screens are recorded Events on other screens are not recorded.
Number of filter (Data ID: $OPERATION_LOG_BOOK.SCREEN_FILTER_NUMBER / INTEGER / RW)	20	1,200	Number of screen filtering
Screen registration > Software ID (Data ID: $OPERATION_LOG_BOOK.SCREEN_FILTER_SOFTWARE_ID [1-200] / INTEGER / RW / Array corresponds to the number of filters)	0	0, 2147483647	Software ID of screen for filter

Configuration item	Default	Range	Description
Screen registration > Screen ID (Data ID: $OPERATION_LOG_BOOK.SCREEN_FILTER_SCREEN_ID[1-200] / INTEGER / RW / Array corresponds to the number of filters)	0	0, 2147483647	Screen ID of screen for filter

To filter screens so that operations on the password setting screen are not recorded, set the following:

```
Enable screen filtering = TRUE
Software ID 1 = 1309
Screen ID 1 = 3
```

12.4 EXTENDED ALARM LOG

12.4.1 Setup

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

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

Table 12.4.1 Configuration item

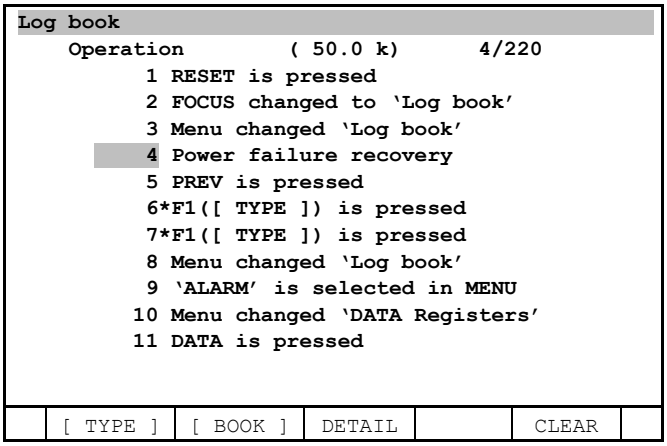
Configuration item	Default value	Value	Comment
Log Book Settings > Book 3 > Title	‘’	ALARM	Title of book
Log Book Settings > Book 3 > Size	0	500	Size of book is 500kbytes. The number of alarms that this book can store depends on contents of alarms.
Log Book Settings > Book 3 > Memory type	NVRAM	NVRAM	Buffer of this book is made on permanent pool. Cycle power doesn’t clear records.
Log Book Settings > Book 3 > Display	TRUE	TRUE	The book of “ALARM” is displayed in F2 ([BOOK]) pull up menu. Please note that this book is not always displayed as the 3rd item on the pull-up menu.
Log Book Settings > Logged Event > Alarm	Book 1	Book 3	Alarms are recorded in book3.

Configuration item	Default value	Value	Comment
Log Book Settings > Alarm filtering > Severity of alarm	PAUSE + ABORT	Full	Filter by severity of error. When the item is selected, the corresponding errors are logged. • WARN: Log warning errors. • PAUSE: Log pause errors. • ABORT: Log abort errors. Ex. Full = warn, pause and abort errors are logged (All alarms are recorded).

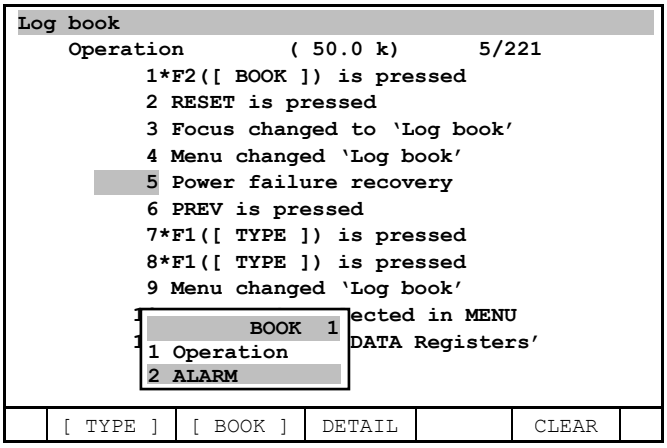
Cycle power is needed to enable these changes.

12.4.2 How to Display Alarm Log

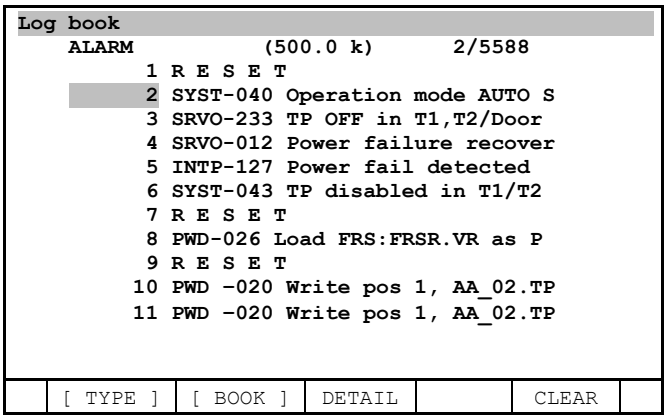
1) Press the [MENU] key.
Select ALARM.
Press the F1 and select “Log book”. Logbook screen will be displayed.



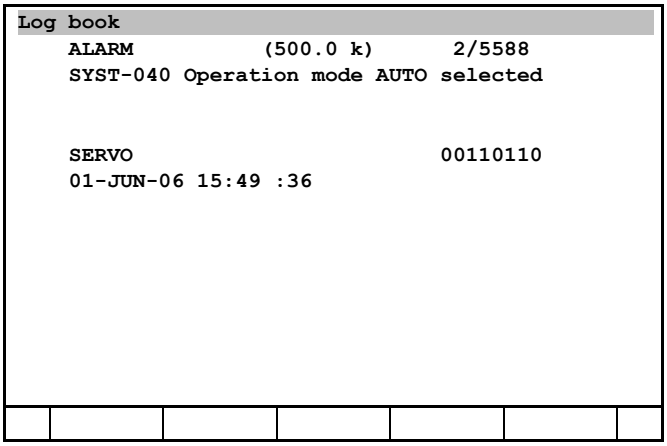
2) Press the F2 and select “ALARM”.



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



4) F3 shows detail screen.



13 PROGRAM TOOLBOX

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

- Soft limit setting

13.1 SOFT LIMIT SETTING

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

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

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

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

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

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

Using the limit setting option

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

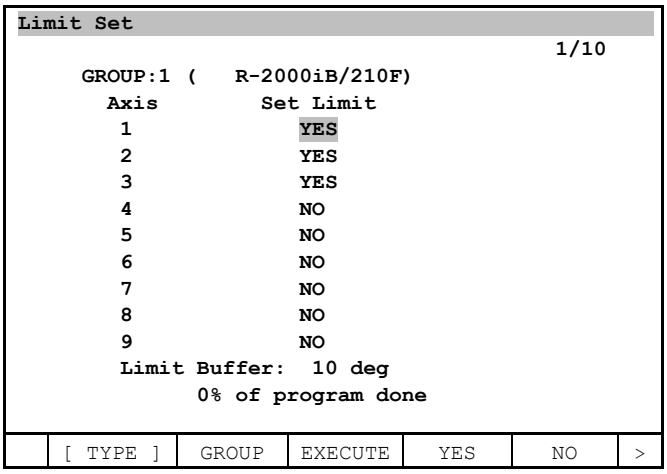
Table 13.1 Toolbox soft limit setup menu items

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

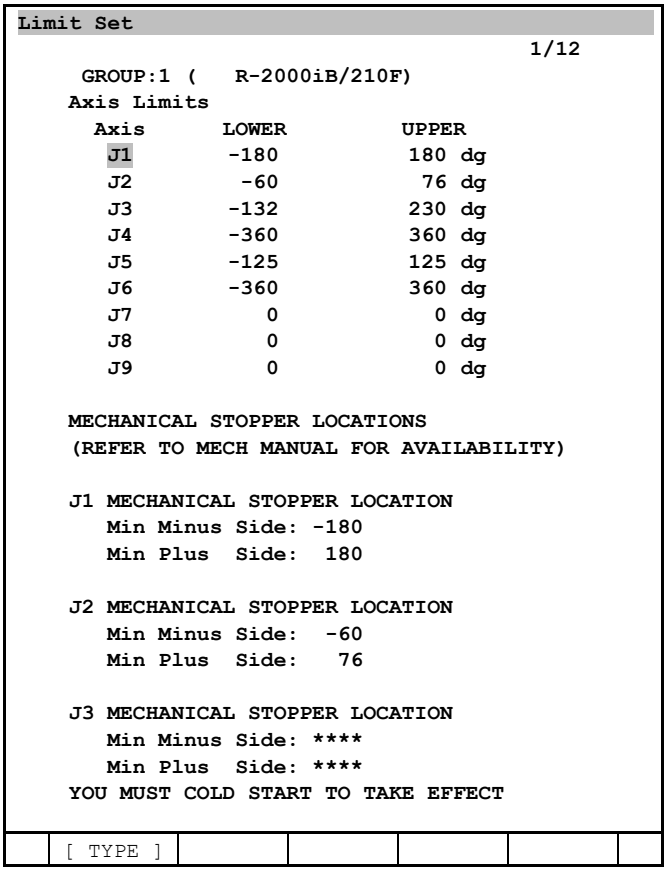
Procedure 13-1 Using the soft limit setting option to set up software axis limits

Step

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



5 To select an axis for limit setting, follow the steps below:
   a. Move the cursor to the number assigned to a desired axis.
   b. Press the F4, Yes to select the axis. If F5, No is pressed, no limit is set up for the axis,
6 To specify a limit buffer, move the cursor to Limits buffer, enter a degree, and press the [ENTER] key.
7 To set up the axis limits, press the F3, EXECUTE.
   When limit setting is completed, the following information will be displayed on the screen.



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

Additional information

The displayed mechanical stop numbers indicate the positions of the center bolt hole on each mechanical stop.

Note: The current version does not support the mounting positions of mechanical stops.

Additional information

The following operation is not supported on some robot models.

8 To reset the axis limits to their factory settings, press the F1, DEFAULT.

Additional information

Enabling new software axis limit settings always requires cold-starting the controller.

9 After axis limit setting is completed, enable the new software axis limit settings by cold-starting the controller according to the following steps.

a. If the controller is already on, turn it off.
b. On the teach pendant, press and hold down the [SHIFT] and [RESET] keys.
c. While still holding down the teach pendant keys, turn the power on.
d. After the teach pendant has displayed its menu, release the teach pendant keys.

Additional information

If many joint limit errors occur during program execution, increase the limit buffer and re-run the program.

14 ADVANCED CONSTANT PATH

Corner Path LD CR allows you to use Linear Distance and Corner Region for adjusting the path. Corner Path LD CD allows you to use Linear Distance and Corner Distance for adjusting the path. The method of specifying the corner size differs between Corner Region and Corner Distance. These options cannot be installed at the same time. You can install either option, but not both
ADV-Max Speed CTRL allows you to use Process Speed and Max Speed for adjusting the speed of a robot.

Corner Path LD CR (R806) uses the following to adjust a path:
- Linear Distance
- Corner Region

Corner Path LD CD (R748) uses the following to adjust a path:
- Linear Distance
- Corner Distance

ADV-Max Speed CTRL (R805) uses the following to adjust the robot speed:
- Process Speed
- Max Speed

The Advanced Constant Path functions requires that the Constant Path function to be installed be enabled.

14.1 LINEAR DISTANCE

Linear Distance is a robot motion feature that is useful for material handling applications such as palletizing.
Fig.14.1(a) shows a typical pick and place application. If a robot starts at P1, goes through P2, picks up a part at P3, goes through P2 to P4, and then to P5 where it places the part and all termination types are FINE or CNT0, then the path would be as shown in Fig. 14.1 (a).

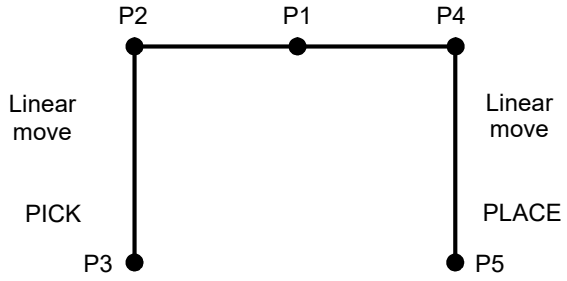
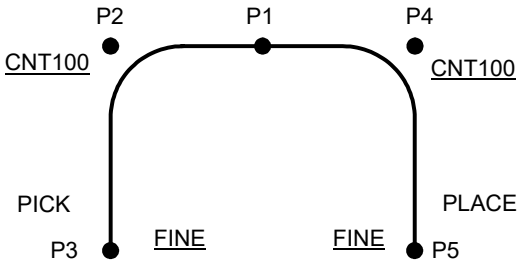


Fig. 14.1 (a) Path with FINE or CNT0

Typically, however, FINE and CNT0 are used only to reach P3 for PICK and to PLACE at P5. For all other motions, high CNT values are used. For example, if CNT100 is used, the actual path might look like the one shown in Fig.14.1(b).

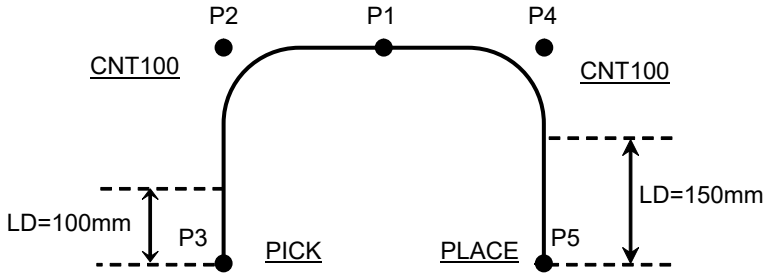



```
Robot is at P1
J  P[2] 100% CNT100
L  P[3] 2000mm/sec FINE
L  P[2] 2000mm/sec CNT100
L  P[4] 2000mm/sec CNT100
L  P[5] 2000mm/sec FINE
```

Fig. 14.1 (b) Path with CNT100

This kind of a path will give you better cycle time. However, you cannot specify the linear part of the path from P2 to P3 or from P4 to P5 directly. If you want to adjust the linear part of the path, you need to adjust the values of CNT.

With the Linear Distance feature, you do not have to guess and experiment with different termination type. If you want the last 100mm before pick and the last 150mm above place to always be straight above P3 and P5 respectively, you can use Linear Distance to specify these amounts. Refer to Fig. 14.1 (c).




```
The program would be (from P1)
J  P[2] 100% CNT100
L  P[3] 2000mm/sec FINE AP_LD100
L  P[2] 2000mm/sec CNT100 RT_LD100
L  P[4] 2000mm/sec CNT100
L  P[5] 2000mm/sec FINE AP_LD150
```

Fig. 14.1 (c) Path with Linear Distance

14.1.1 How Linear Distance Works

Linear Distance allows you to use two motion modifiers:

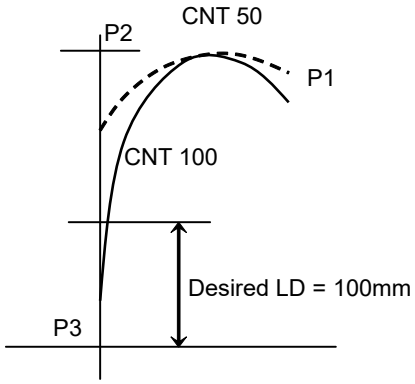
RT_LD: Retract Linear Distance is used for specifying the desired linear distance from the start position to the start of the corner.

AP_LD: Approach Linear Distance is used for specifying the desired linear distance from the end of the corner to the destination position.

WITHOUT USING LINEAR DISTANCE
Robot is at P3

•Dashed line example :
L P[2] 2000mm/sec CNT50
J P[1] 100% CNT100

•Solid line example :
L P[2] 2000mm/sec CNT100
J P[1] 100% CNT100



USE OF LINEAR DISTANCE
Robot is at P3

•Dashed line example :
L P[2] 2000mm/sec CNT50 RT_LD100
J P[1] 100% CNT100

•Solid line example :
L P[2] 2000mm/sec CNT100 RT_LD100
J P[1] 100% CNT100

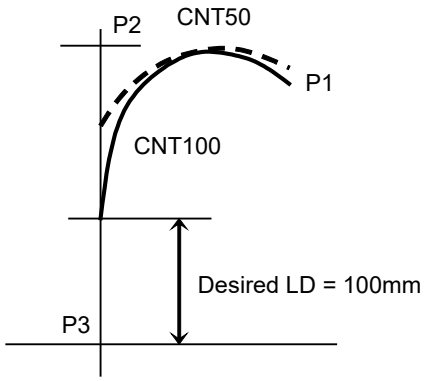


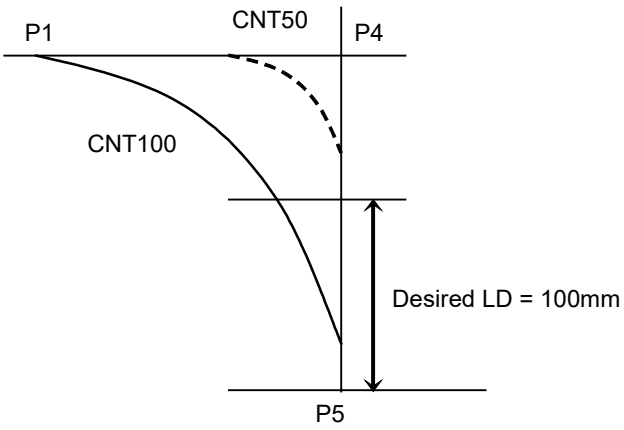
Fig. 14.1.1 (a) Linear Distance (RT_LD)

The RT_LD value affects the corner of P3-P2-P1 in Fig. 14.1.1(a). The higher the value of RT_LD, the shorter the distance between the path and P2 will be. When the RT_LD value is greater than or equal to the distance between P3 to P2, the motion will automatically become like FINE regardless of the CNT value you specify.

WITHOUT USING LINEAR DISTANCE
Robot is at P1

•Dashed line example :
J P[4] 100% CNT50
L P[5] 2000mm/sec FINE

•Solid line example :
J P[4] 100% CNT100
L P[5] 2000mm/sec FINE



USE OF LINEAR DISTANCE
Robot is at P1

•Dashed line example :
J P[4] 100% CNT50
L P[5] 2000mm/sec FINE AP_LD100

•Solid line example :
J P[4] 100% CNT100
L P[5] 2000mm/sec FINE AP_LD100

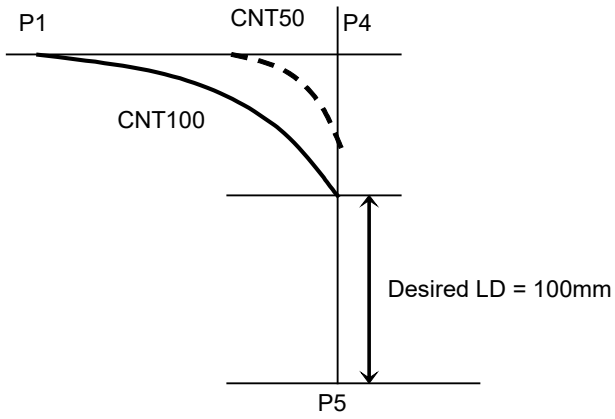


Fig. 14.1.1 (b) Linear Distance (AP_LD)

The AP_LD value affects the corner of P1-P4-P5 in Fig. 14.1.1(b). The higher the value of AP_LD, the shorter the distance between the path and P4 will be. When the AP_LD value is greater than or equal to the distance between P4 to P5, the preceding motion (motion from P1 to P4) will become like FINE regardless of the CNT value for that move.

14.1.2 Linear Distance Limitations

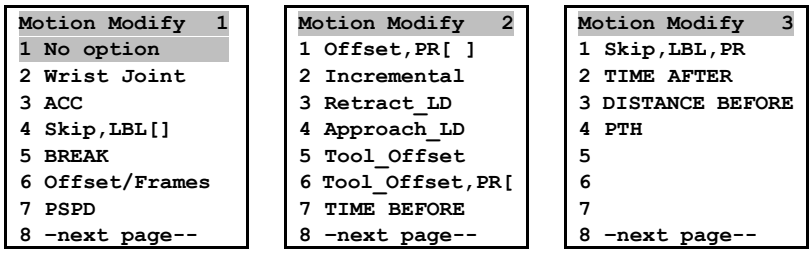
- Linear Distance cannot match the specified value exactly. However, it will never provide linear distance less than what you specified.
- Linear Distance can be used with only the linear motion type.
- Linear Distance cannot be used with an Independent Axis or Positioner. If you use linear distance instructions, the instructions will be disabled.
- When multiple group motion is used, the motion will be synchronized. However, if more than one group has linear distance enabled, all of the groups will have linear distance satisfied.
- The local condition trigger time might be different than without Linear Distance. However the timing is repeatable.
- Linear Distance cannot be used with the following options.
  - Max Speed
  - Weaving
  - Continuous turn
  - Robot Link
  - TCP speed output function
- Linear Distance can be used with Coordinated Motion.

14.1.3 Using Linear Distance

Procedure 14-1 Using Linear Distance

Step

1 Press the [SELECT] key.
2 Move the cursor to the name of the program you want to modify and press the [ENTER] key.
3 Move the cursor to the empty space at the end of the linear motion instruction that you want to modify and press the F4, [CHOICE]. The following menu will be displayed.



4 If you want to use Linear Distance from the start position to the start of the corner, select Retract_LD. If you want to use Linear Distance from the end of the corner to the destination position, select Approach_LD.
5 Type the number of millimeters that you want the tool center point (TCP) to approach or retract using Linear Distance.

NOTE
The default value is “direct” which means that the value is a specific number in millimeters. To use a value stored in a register, press the F3, INDIRECT, and type the register number.

14.2 CORNER REGION AND CORNER DISTANCE

Corner Region (CR) and Corner Distance (CD) are optional termination type that can be used to adjust the corner rounding for Cartesian motions. When you use these termination types, the speed at the corner remains constant as far as possible.

- Corner Path LD CR (R806) is required when using Corner Region.
- Corner Path LD CD (R748) is required when using Corner Distance.
- You can install either option.

Corner Region

Example: L P[1] 100mm/sec CR10

Corner Region (CR) specifies the distance from the taught position to the start of a corner path. The actual distance is less than or equal to the specified value.

Corner Distance

Example: L P[1] 100mm/sec CD10

Corner Distance (CD) specifies the distance from the taught position to the corner path. The actual distance is less than or equal to the specified value.

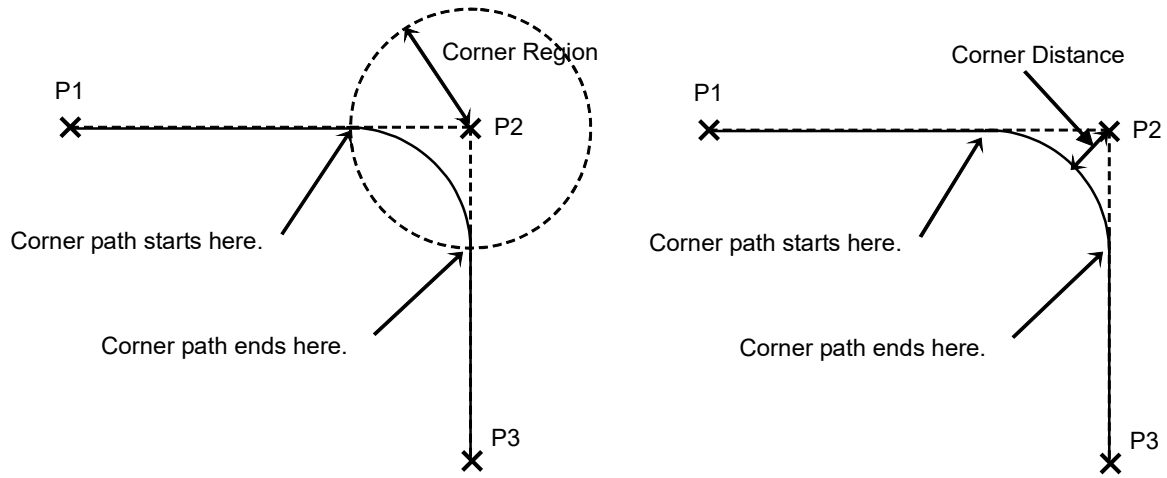


Fig. 14.2 Corner Region and Corner Distance

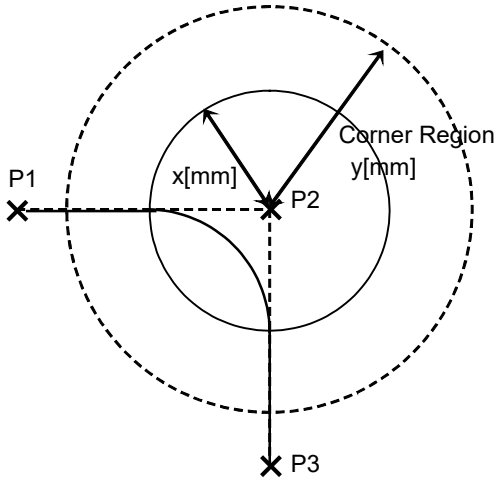
When you use Corner Region or Corner Distance, use the following guidelines:
- Specify the value in millimeters.
- The value can range from 0 mm to 1000 mm.
- The smaller the value, the closer the robot will get to the taught position, and the less the corner rounding.
- With a larger value specified, the robot will not get as close to the position and the more corner rounding.

14.2.1 Limitations of Corner Size by Distance between Taught Points

Corner Size is limited by the distance between taught points. Even if the specified value is greater, the actual corner size is limited to half distance between P1 and P2 (P2 and P3).

If Corner Region (CR) is used, the corner size is limited as shown below. Similarly for Corner Distance, the corner size is limited.

Corner Region y[mm] > half distance between P1-P2 (P2-P3)
Actual Corner Region x=half distance between P1-P2 (P2-P3)



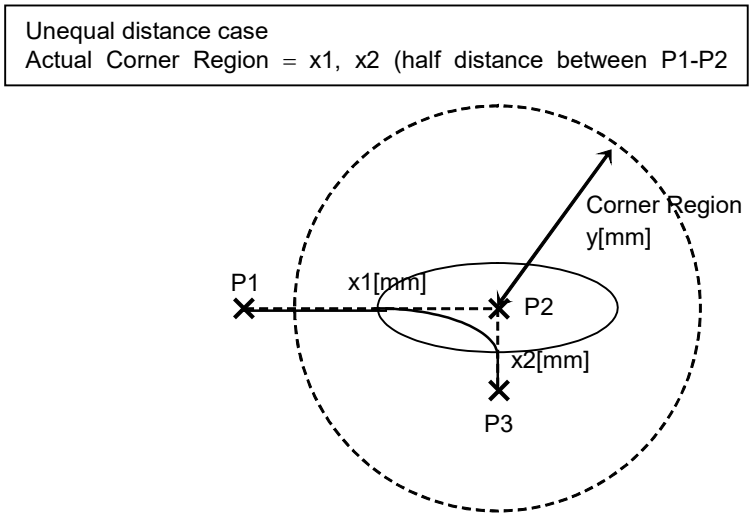


Fig. 14.2.1 Limitation of the specifiable CR value

Keep in mind that because of the above rule, even with a large specified value, the corner path may still be close to the taught position when the distance between the continuous taught points is short. When you teach a path, minimize the number of taught positions, increase the interval between taught positions, and reteach positions using CR and CD to fit the path instead of adding positions.

14.2.2 Corner Region and Corner Distance Limitations

- CR and CD cannot be used with the JOINT motion type.
- The shape of the path of the corner part cannot be a circular arc.
- CR and CD cannot be used with an Independent Axis or Positioner. When an Independent Axis or Positioner moves with the robot, it is possible to use these termination types. However, in this case, CR and CD take effect only for robot motion. An Independent Axis or Positioner synchronizes with the robot.
- When multiple robots move in a multi-group system, the motion of the robots is synchronized. In this case, all robots move so that the corner region is within the specified value.
- If the angle between a motion and the next motion is almost 180 degrees, CR and CD will not work and the path becomes the same as when CNT100 is specified.
- The speed at the corner remains constant as far as possible. However, in the following cases, the speed decreases because it is impossible to anticipate the motion.
  - Motion instructions with the target position specified by a position register.
  - Motion instructions with an offset instruction in which the offset is stored in a position register.
  - Motion instructions with the indirect addressing with the register.
  - Branch instructions (IF or SELECT JMP LBL[*]) used before the next motion instruction.
  - WAIT instruction with TIMEOUT LBL[*] is used before the next motion statement.
  - Frame instructions used before the next motion instruction.
- CR and CD can be used with the following options.
  - Remote TCP
  - Line Tracking
    During the transition (between Remote TCP and non-Remote TCP or between Tracking and non-Tracking), the corner cannot be defined because the frame has been changed. Therefore, the corner region cannot be specified by CR and CD but the corner path is connected continuously.
  - Linear Distance
    If you specify both Linear Distance and CR and CD, then Linear Distance has preference over CR and CD, as shown in Fig.14.2.2.

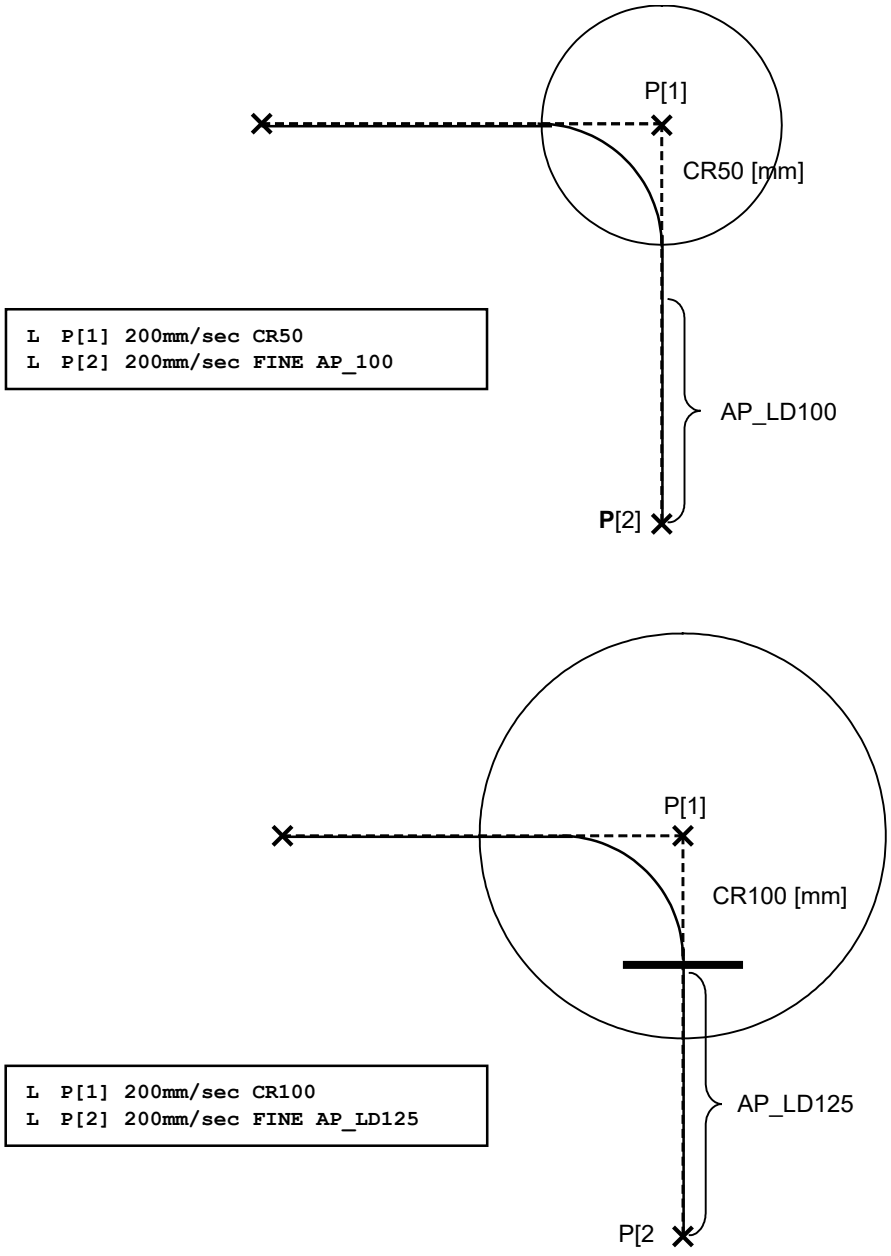


Fig. 14.2.2 Corner Path determined if Linear Distance is satisfied

14.3 PROCESS SPEED

PSPD xxx

Process speed is a motion option control feature that allows you to adjust robot speed to be slower along a given path (if applicable, the path would be maintained the same regardless of xxx), where xxx is an integer you specify.

Process speed is useful for applications with continuous path motion; for example, sealing and waterjet cutting. For these applications, teach the desired path using normal methods, tweaking the taught position, speed, termination type and ACC. After the path has been taught, if you want to adjust its process speed from the nominal taught value, but do not want to change the path, you can use the Process Speed feature.

Add this motion option to the range of motion lines where adjustment is required.

- PSPD 100 is equivalent to the default cases without PSPD option.

- PSPD less than 100 means slower process speed, while maintaining the same path.
- You can still change other fields in the motion for further tweaking, but the same original rules apply; that is, the path will change. This allows you to adjust the path easily, even though PSPD is used.

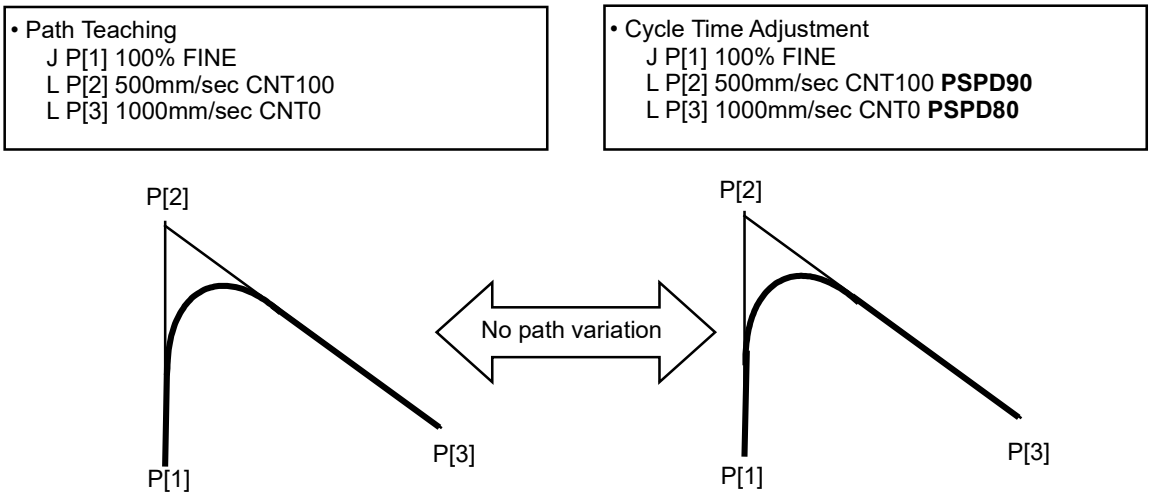


Fig. 14.3 Process speed tweaking

14.3.1 Process Speed Limitations

- The PSPD option does not support the TCP Speed Prediction function (TCPP). That is, for the motions with the PSPD option, TCPP might not result in the correct results.
- With very short segments, the actual corner path might deviate from the one without the PSPD option.

14.4 MAX SPEED

L P[1] max_speed CNT100

In some applications, the desired speed is the maximum speed that the robot can deliver. For joint motion moves, the system delivers the maximum capability of the robot; that is, one of the axes reaches its maximum speed. For linear moves, the system delivers the speed that is specified in the teach pendant instruction. However, the maximum linear speed of 2000mm/sec imposes a limit on the capability of the motor to reach higher speeds. The robot can move faster than the speed specified in the motion instruction.

The max speed option allows you to specify a linear motion that will use the maximum speed capability of the robot. It improves cycle times in loading/unloading applications by speeding up long linear motions. When this option is loaded, the choice of max_speed will be displayed in the speed field of the teach pendant motion instruction for a linear motion. The max_speed option affects only the motions for which the speed is specified as max_speed.

- If you change the motion type from Linear to Joint, the speed field will change to 100%.
- When the speed field changes from max_speed to another choice, the speed value will return to the original speed value.

> ⚠ **WARNING**
> When you specify max_speed, the robot will run at high speed. Be sure any loose parts are firmly attached and that the workpiece is secured. Otherwise, you could injure personnel or damage equipment.

14.4.1 Max Speed Limitations

- If unsupported options are used, max_speed will be disabled automatically. No warning or error message will be displayed. This option does not support the following:
  - Any tracking option, such as line tracking, TAST, Coordinated motion, and so forth.
  - Multiple group motion
  - RTCP function
- If you run a program with an override speed different than 100%, the system will drive the robot so that one of its axes will reach the override value of its maximum joint speed.
- The local condition trigger time might have some variation.
- If the path becomes too aggressive, you might need to use ACC to smooth it.
- If you are using Dry Run, max speed will be disabled and the speed specified in dry run will be used.
- If you are using Org path resume, max speed will be disabled for the motion line that is resumed.
- If T1 is selected, the T1 speed will be used.
- In single step mode (FWD/BWD) max speed will be disabled and the maximum speed value will be used.
- Max speed will be disabled automatically for a circular motion.
- The max speed option will still apply when the Miscellaneous teach pendant instruction LINEAR_MAX_SPEED is used.
- The robot will try to attain the maximum speed capability of at least one of its axes. It determines the maximum speed for the current move by comparing the teach pendant instruction LINEAR_MAX_SPEED with the maximum linear speed of 2000 mm/sec. The ratio of these two speeds is the percentage of the maximum axis speed that the axis will reach.

  For example: the maximum linear speed is 2000 mm/sec.

```
1. LINEAR_MAX_SPEED = 1200
2. L P[1] max_speed CNT100
```

  The ratio of 1200 to 2000 is 60%. The system will drive the robot so that one of its axes will reach 60% of its maximum joint speed for line 2 of the program above.
- Max speed does not work with the TCP Speed Prediction function (TCPP). That is, with max_speed TCPP results may not be accurate.

15 SINGULARITY AVOIDANCE FUNCTION

For most FANUC 6-axis robots, the singularity that occurs at the elbow (at the boundary between "up" and "down" configurations) or at the boundary between "front" and "back" configurations causes regional structure degeneration. This kind of singularity can easily be avoided by restricting the workspace of a robot. However, the wrist singularity may happen virtually any place inside the workspace. When a FANUC robot travels through or near a wrist singular position, motion performance becomes undesirable because

- Joints 4 and 6 will change significantly within a short time period;
- The TCP speed of the robot will slow down;
- The path might deviate from the taught path one if the motor speed limit is exceeded.

The Singularity Avoidance function provides a real-time solution to avoid wrist singularity. With this function, the following results are achieved:

- The rotation of joints 4 and 6 is minimized, the robot can travel through or near a wrist singular position smoothly, and TCP speed can be maintained
- This function works for both LINEAR TP program motion and LINEAR jogging

To use the Singularity Avoidance function, the Singularity Avoidance function option (R792) or Motion package (A05B-2600-R809) is required. Some robot models do not support this function.

15.1 HOW TO USE SINGULARITY AVOIDANCE

This section explains how to use Singularity Avoidance in jogging and TP programs.

15.1.1 How to Use Singularity Avoidance in Jogging

You can confirm whether the Singularity Avoidance function is enabled for jogging by checking whether “S/” is added before the jog coordinate system shown on the teach pendant status line.
For example: S/WORLD
If “S/” is added, Singularity Avoidance in enabled for jogging. If “S/” is not added, Singularity Avoidance in disabled for jogging. The enabled status of Singularity Avoidance for jogging is automatically changed when the program is selected. If you select a program for which Singularity Avoidance is enabled, it will be enabled, and if you select a program for which it is disabled, it will be disabled.

You can also confirm whether the Singularity Avoidance jogging is enabled by the following Data ID.
(Data ID: $JOG_MODE.IS_SINGULARITY_AVOIDANCE_JOG[a] / BOOLEAN / RO / a: group number)

Use Procedure 15-1 to enable and disable Singularity Avoidance for jogging, without changing the selected program.

Procedure 15-1 Select enabled/disabled of singularity avoidance in jogging

Step

1 Press FCTN.
2 Select T/Singularity JOG to toggle enabling and disabling Singularity Avoidance for jogging.

15.1.2 Using Singularity Avoidance in a TPE program

You can enable and disable Singularity Avoidance in each TPE program. When a new TP program is created, the Singularity Avoidance enabled/disabled status of the program is automatically determined depending on the Singularity Avoidance enabled/disabled status for jogging. Note that if you load a TP that was saved before the Singularity Avoidance option was added, or was saved in a system that did not have the Singularity Avoidance option, Singularity Avoidance will be disabled in that TP program.

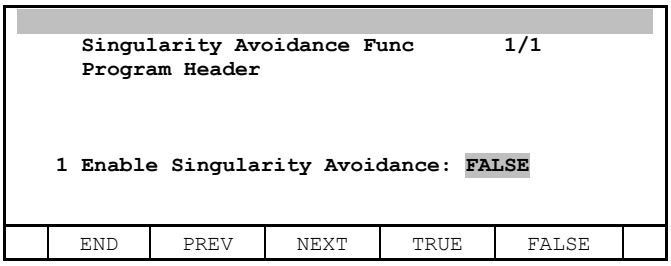
Use Procedure 15-2 to enable and disable Singularity Avoidance in a program.

Procedure 15-2 Enabling and disabling Singularity Avoidance in a program

Step

1 On the program Select screen, move the cursor to the program you want and press F2, DETAIL to display the detail screen for that program.

2 Press the F3, NEXT and the following screen will be displayed. This screen shows that Singularity Avoidance is disabled.



3 To enable the Singularity Avoidance, press F4, TRUE. To disable Singularity Avoidance, press F5, FALSE.

15.2 SYSTEM CONFIG MENU

Singularity avoidance displays on Setting category items of System config menu when “Singularity Avoidance” (R792) is installed. This item allows you to set up Auto Modify Target Points and Display Warning Message.

Setting hierarchy 2	Description
Auto Modify Target Points	Auto Modify Taught Points in the relevant group is ENABLE, the Auto Singularity Avoidance function will internally modify the destination position when it is inside a pre-defined singularity zone in the following way: ・The joint angle of Axis 4 is kept the same as the value of the previous position. ・The TCP location (x, y and z) of the position is the same as it was taught (no location error). ・The orientation (w, p and r) of the position is kept as close to the taught position as possible (minimal orientation error). This setting can be configured for each group.
Display Warning Message	If this setting is disabled, warnings (MOTN-205, 207, 209, 330) regarding the Singularity Avoidance will not be displayed. This setting can be configured for each group.

15.3 LIMITATIONS

The Singularity Avoidance function has the following limitations:

- This function is only available in Handling Tool. This function is also available when using LR Tool or LR Handling Tool, or changing to Handling in Spot Tool +.
- Some robot models does not support this function.
- This function cannot be ordered with the Coordinated Motion option.
- This function cannot be used with the Line Tracking option.
- This function cannot be used with Continuous Turn option.
- This function cannot be ordered with the Shape Generation II option.
- This function can be used only with linear motion. This function cannot be used with Circular/Arc motion.

15.4 CAUTIONS

Keep the following important information in mind when you set up and use Singularity Avoidance:

- With the Singularity Avoidance function, the actual wrist configuration (flip/non_flip) might be different from the taught destination positions. The Singularity Avoidance function might change the configuration internally not only for the destination position in the motion line where a singularity is detected, but also for the subsequent destination positions in the following motions in the TP program.
- When a taught point is near a singularity, the actual joint position that is reached may be different from the taught joint position.
- Since the Singularity Avoidance function will change the configuration internally, single step forward and backward execution might produce different behavior. To prevent this from happening, during single step forward motion, the system will post a warning message "MOTN-208 Config Not Reached" at the point where the actual configuration is different from the taught one. Use the information in this message to touch up the taught point at the specific line shown in the warning message. This will update the taught configuration to the actual configuration. As a result, when stepping backward, the motion will behave the same way as stepping forward.
- Jogging and program motion might be different when the robot moves near a singularity.
- The function might not help for the singularity at the a corner path.
- If the position reached in single-step mode is near a singularity, “MOTN-023 In singularity” alarm will occur at the next operation. This alarm does not occur during continuous operation mode (single-step mode OFF).

16 PATH SWITCHING FUNCTION

The Path Switching function (J693) allows you to execute assignment instructions at several points in a motion path.

16.1 PS INSTRUCTION

When a Path Switching (hereinafter called PS) instruction is included in a DB calling program as follows, PS instructions define the trigger points and assignment instructions to be executed.

Main Program (DB CALL instruction)

```
1: L P[1] 200mm/sec FINE DB 100mm, CALL A
```

Contents of subprogram A

```
1:    PS -100mm +0.2sec,DO[1]=(ON)
2:    PS -150mm -0.2sec,IF(DI[1]),DO[2]=(DI[2] AND !DI[3])
```

The distance specified just after the DB instruction does not affect the trigger timing of each of the PS instructions in the subprogram. The PS instructions in the subprogram start their execution when the motion starts. You can mix another normal logical instruction with a PS instruction in a subprogram.

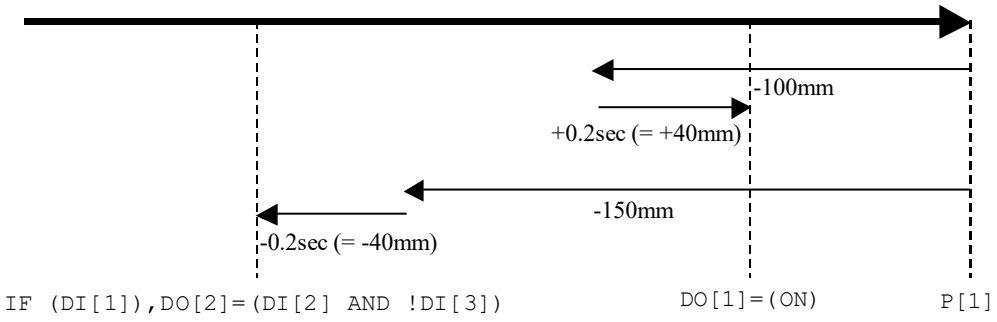


Fig. 16 (a) Path switching application

A PS instruction is composed of two parts, a trigger point and an assignment instruction.

```
PS -100mm +0.2sec,DO[1]=(ON)
     Trigger point   Assignment instruction
```

The trigger point part specifies the trigger distance (mm) from the destination position and the offset time (sec).
If you enter a negative value, you can specify a trigger point before the motion is completed (distance-before type).
If you enter a positive value, you can specify a trigger point after the motion is completed. (distance-after type).
The offset time is converted to a distance according to the specified speed. For example, in the above case, 0.2sec is converted to 40mm (= 200mm/sec * 0.2sec). This conversion is not accurate in the acceleration and deceleration periods.
The adjustment of the trigger point by an offset time is affected by the override value.

The assignment instruction part of the PS instruction specifies the instruction to be executed at the trigger point. The assignment instruction can have an IF condition.

```
PS -150mm -0.2sec,IF (DI[1]),DO[1]=(ON)
```

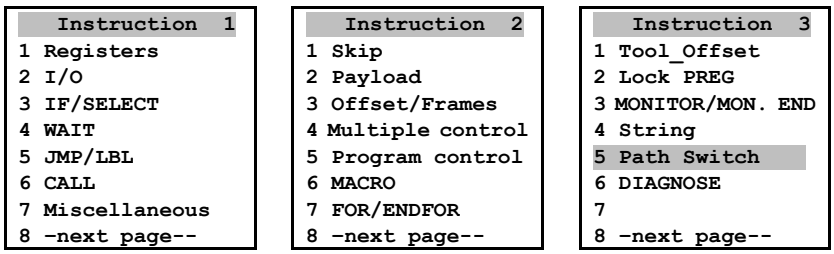
In this case, when DI[1] is ON, DO[1] is turned ON; when DI[1] is OFF, DO[1] is not changed.
The assignment instruction can use the following items and operators.

Items for left side	Operator for right side or IF condition	Items for right side or IF condition
DO[]	( )	ON OFF
RO[]	+	Constant value
WO[]	-	DI[] DO[]
AO[]	*	RI[] RO[]
GO[]	/	WI[] WO[]
SO[]	MOD	AI[] AO[]
UO[]	DIV	GI[] GO[]
F[]	AND	SI[] SO[]
R[]	OR	UI[] UO[]
PR[i,j]	!	F[]
M[]	= (comparison)	M[]
TC_ONLINE	<	R[]
	>	PR[i,j]
	<=	TIMER[]
	>=	TCP_SPD[]
	<>	

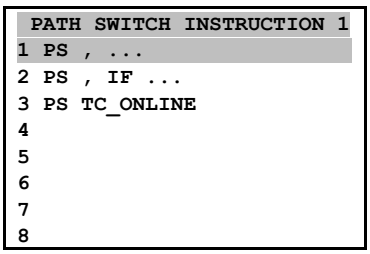
A marker assignment and TC_ONLINE can be specified in a PS instruction.
An AR[] (calling parameter) cannot be specified in a PS instruction.
Up to 20 operations can be specified in one motion line in total on the right side formula and IF condition.
One motion instruction can have up to 20 PS instructions.

TP editor operation

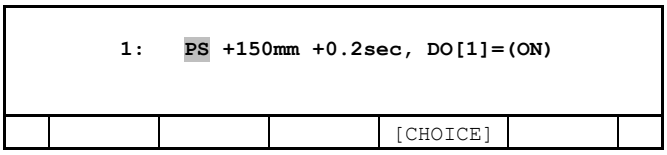
1 The Path Switching item is displayed on the [INST] menu.



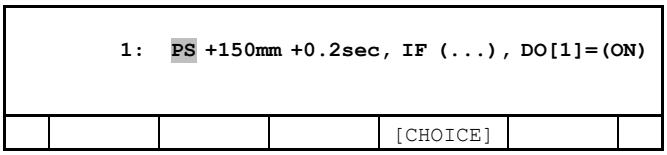
2 Select the Path Switching item. The following menu will be displayed.



3 Set the distance and offset, and edit the assignment instruction in the same way as for other mixed logic instructions.



4 To change between different PS instructions, move the cursor to PS, press F4, [CHOICE], and select the PS instruction you want.



16.2 TRIGGER POINT OF PS INSTRUCTION

When the TCP path moves away without entering the trigger area using CNT motion

When using CNT motion, the taught point is not on the actual robot path. If the robot path is too far from the trigger area, the PS instruction will remember internally that it has not been triggered and will be triggered when the TCP path starts to move away from the taught position.

```
1: L P[1] 200mm/sec FINE
2: L P[2] 200mm/sec CNT100 DB 100mm, CALL A
3: L P[3] 200mm/sec FINE
```

Contents of sub-program A

```
1:   PS -10mm +0sec,DO[1]=(ON) (distance-before type)
2:   PS +10mm +0sec,DO[2]=(ON) (distance-after type)
```

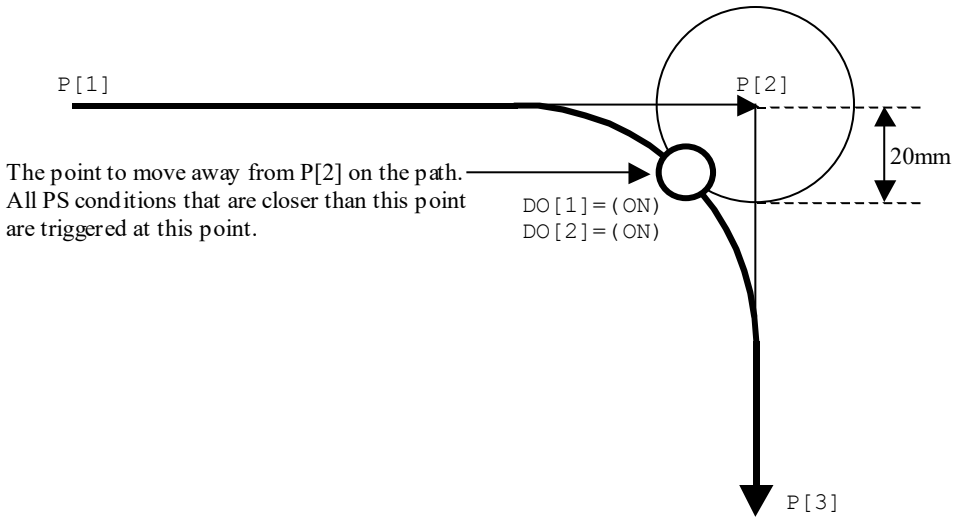


Fig. 16 (b) Path Switching application

You can choose to issue error in this situation by setting the Pass outside the area item on the Path Switching Config screen which is on the Program Instructions of the SYSTEM / CONFIG screen, as follows.

- (DB)Pass outside the area - distance-before type

Value of Pass outside the area item	Description
0	Trigger PS condition at the closest point, but post neither alarm nor warning.
1 (default)	Trigger PS condition at the closest point, and post “INTP-541 PS(program name, line number, distance[mm]) Forced trigger” (warning).
other than (0, 1)	Not triggered, and post “INTP-542 PS(program name, line number, distance[mm]) No trigger” (PAUSE).

- (DA) Immediately trigger - distance- after type

Value of Immediately trigger item	Description
0	Trigger PS condition immediately, but post neither alarm nor warning.
1 (default)	Trigger PS condition immediately, and post “INTP-539 PS(program name, line number, distance[mm]) Already in area” (warning).
other than (0, 1)	Not triggered, and post “INTP-540 PS(program name, line number, distance[mm]) Already in area” (PAUSE).

When trigger point is negative and is farther than the start position (distance-before type)

If the trigger point is negative and is farther than the start position, this is triggered immediately.

```
1: L P[1] 200mm/sec FINE
2: L P[2] 200mm/sec FINE DB 100mm, CALL A
```

Contents of sub-program A

```
1:   PS -1000mm +0sec,DO[1]=(ON) (distance-before type)
```

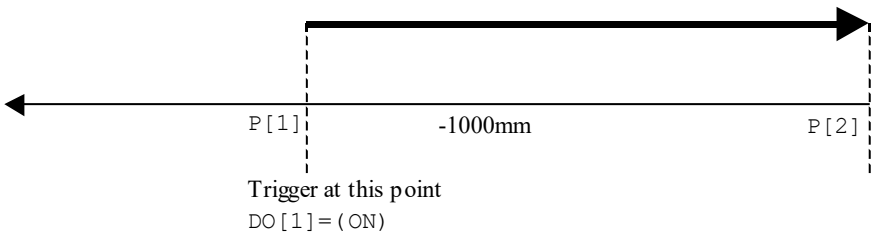


Fig. 16 (c) Path switch application

You can choose to issue an error in this situation by setting (DB)Immediately trigger item on the Path Switching Config screen which is on the Program Instructions of the SYSTEM / CONFIG screen, as follows.

- (DB)Immediately trigger - distance-before type

Value of Immediately trigger item	Description
0	Trigger PS condition immediately, but post neither alarm nor warning.
1 (default)	Trigger PS condition immediately, and post “INTP-539 PS(program name, line number, distance[mm]) Already in area” (warning).
other than (0, 1)	Not triggered, and post “INTP-540 PS(program name, line number, distance[mm]) Already in area” (PAUSE).

When the trigger point is passed by jogging while the program is paused

If the Original Path Resume function is disabled, if a program is paused during PS motion, the robot is jogged, and the trigger point is passed while jogging, then the PS condition is triggered immediately when the program is resumed.

If the Original Path Resume function is enabled, the program returns to the paused position and resumes from that position. In this case, the PS condition is triggered when the trigger point is passed after resuming and restarting.

You can choose to issue error in this situation by setting of (DB)Immediately trigger and (DA)Immediately trigger.
As for the status of original path resume, you can check on the program execution setting screen in the system config menu.

```
1: L P[1] 200mm/sec FINE
2: L P[2] 200mm/sec FINE DB 100mm, CALL A
```

Contents of sub-program A

```
1:    PS -500mm +0sec,DO[1]=(ON) (distance-before type)
```

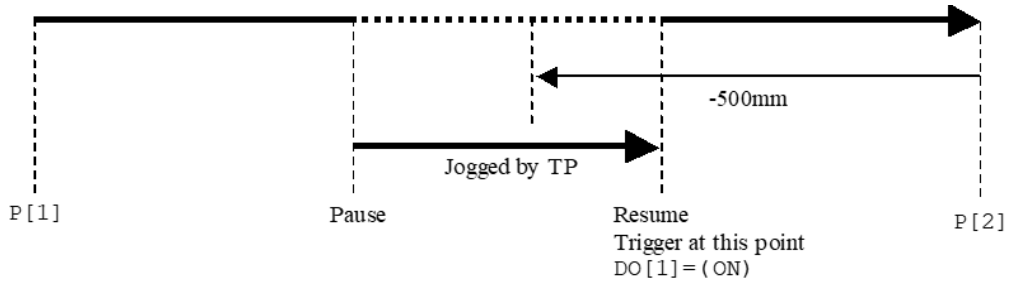


Fig. 16 (d) Path Switching application (Original Path Resume is disabled, distance-before type)

```
1: L P[1] 200mm/sec FINE DB 100mm, CALL A
2: L P[2] 200mm/sec FINE
```

Contents of sub-program A

```
1:    PS +500mm +0sec,DO[1]=(ON) (distance-after type)
```

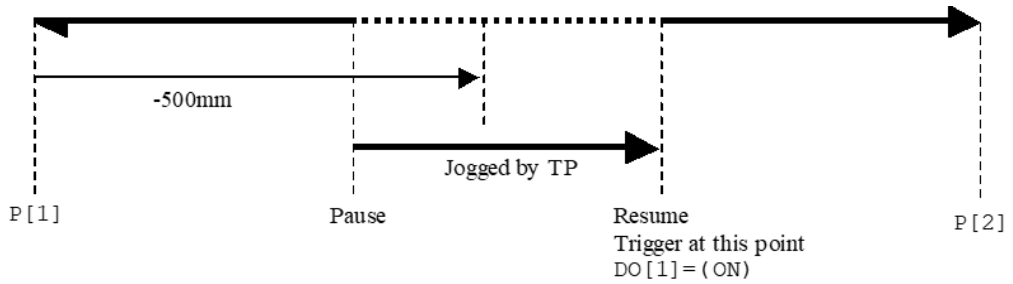


Fig. 16 (e) Path Switching application (Original Path Resume is disabled, distance-after type)

- (DB)Immediately trigger - distance-before type
- (DA)Immediately trigger - distance-after type

Value of Immediately trigger item	Description
0	Trigger PS condition immediately, but post neither alarm nor warning.
1 (default)	Trigger PS condition immediately, and post “INTP-539 PS(program name, line number, distance[mm]) Already in area” (warning).
other than (0, 1)	Not triggered, and post “INTP-540 PS(program name, line number, distance[mm]) Already in area” (PAUSE).

16.3 ABOUT MULTI-GROUP CONFIGURATIONS

In multi-group configurations, the PS instruction is implicitly executed for the first motion group in the group mask of a TP program. For example, if the group mask is [*,1,1,*,*,*,*,*], then the PS instruction will be executed for group 2.

16.4 LIMITATIONS

The PS instruction is not available for the following motion types:

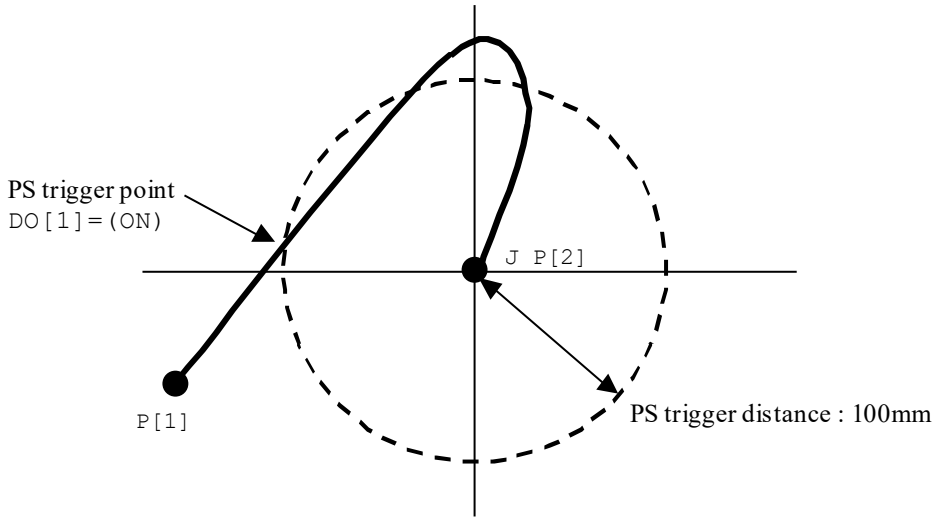
- Weaving motion
- Tracking motion that cannot define the actual destination point beforehand, such as Line Tracking, Dynamic Path Modification, laser sensor tracking, and Robot-Link.

16.5 CAUTIONS

- A PS instruction is interpreted at the moment a new motion instruction is started. When a motion instruction that includes a PS instruction is paused, if you edit and change the PS instruction in the subprogram during this pause, the modified PS instructions will not take effect until the motion instruction is newly executed. After pausing a program, if you want a modified PS instruction to take effect immediately, execute the motion instruction with BWD execution and then restart the program. Then, the modified PS instructions will take effect.

- A PS instruction for joint motion is not prohibited. However, it must be used carefully. Regardless of the motion type: Linear, Circular, or Joint, the PS instruction uses the linear distance between the TCP and destination point as the trigger threshold.

> ⚠ **CAUTION**
> For Distance-Before PS instructions, once the TCP enters the trigger area sphere around the destination point, the PS instruction will be triggered regardless of the distance to the target position on the path. Note that the trigger distance checked by the PS instruction is the absolute distance to the destination position, not the total remaining path length to the destination position. See the following figure for an example.




```
1: J P[1] 100% FINE
2: J P[2] 100% FINE DB 100mm, CALL PROG1
```

Contents of sub-program PROG1

```
1: PS -100mm +0.0sec, DO[1] = ON
```

16.6 SYSTEM CONFIG MENU

Path Switching displays on Program Instructions of System config menu when “Path Switch” (J693) is installed. This item allows you to set up (DB)Pass outside the area, (DB)Immediately trigger, and (DA)Immediately trigger.

Setting hierarchy 2	Setting hierarchy 3	Description
Path Switching	(DB)Pass outside the area	In distance-before type (DB), you can set whether or not an alarm is posted when triggered without entering the trigger area.
	(DB) Immediately trigger	In distance-before type (DB), you can set whether or not an alarm is posted when the robot is already within the trigger area at the time of TP program execution.
	(DA) Immediately trigger	In distance-after type (DA), you can set whether or not an alarm is posted when the robot has already left the trigger area at the time of TP program execution.

17 ALARM RECOVERY

The Alarm Recovery menu is used to quickly and easily recover from faults that occur while a production program is running in automatic mode. Many different recovery methods for different applications are available.
Alarm recovery is only available if feature Alarm Recovery Operation (R913) is loaded on the robot. The Alarm Recovery screen can also be used for user alarms. Refer to User Alarm Recovery section of the manual.
The alarm recovery screen shows the first fault that caused program execution to pause. The cause and suggested remedy for this fault are shown on screen. Figure 17(a) contains an example of the Alarm Recovery screen.

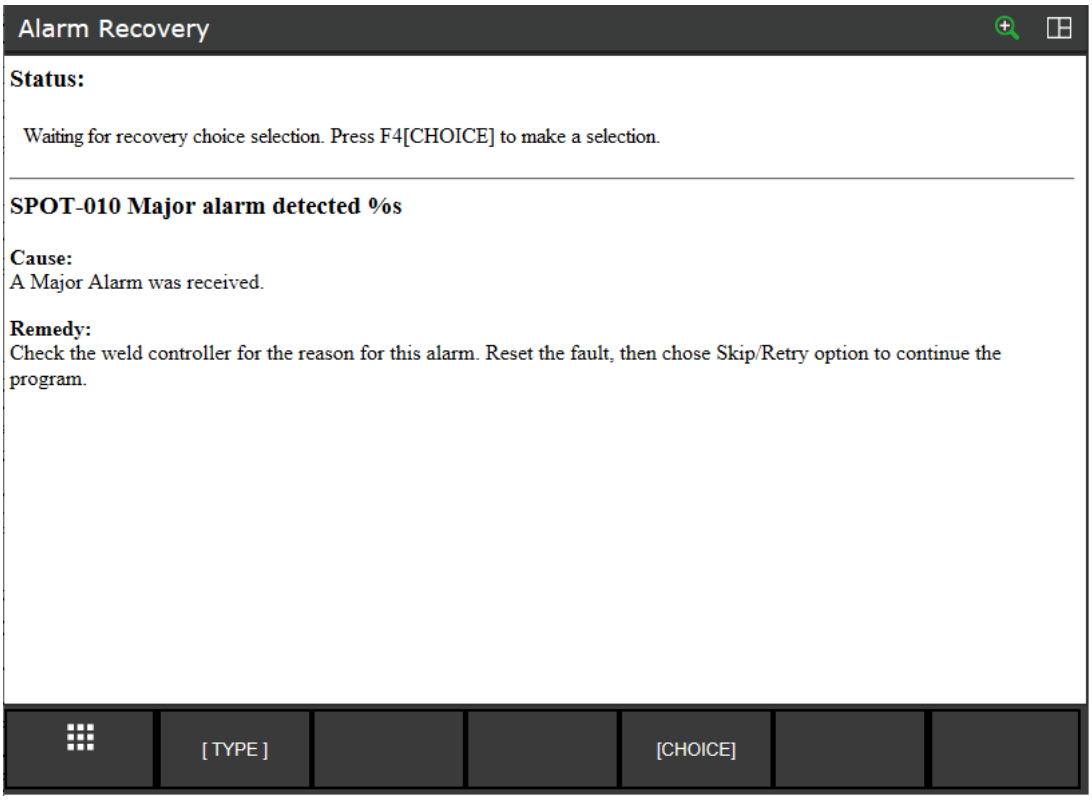


Fig. 17(a) Alarm Recovery Screen Example

You can display other teach pendant screens when the Alarm Recovery screen is displayed. However, if you display another screen, you must do the following Procedure 17-1 to display the Alarm Recovery screen again:

Procedure 17-1 Alarm Recovery Setup

Step

1. Press MENU.
2. Select ALARM.
3. Press F1, [TYPE].
4. Select [Recovery]

If a fault occurs for which no specific cause/remedy exists, or if R665 Alarm Cause/Remedy is not loaded on the robot you will see the screen shown in Figure 17(b).

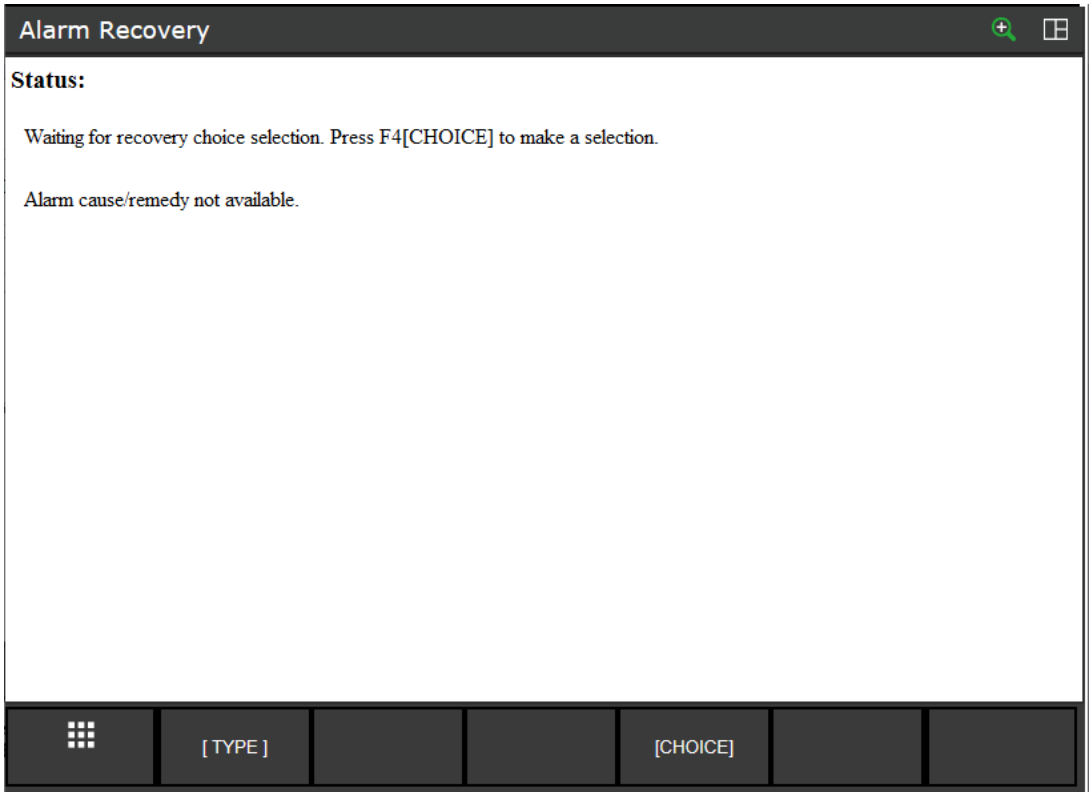


Fig. 17(b) Screen Example without Specific Cause/Remedy Text

17.1 Alarm Recovery Faults [CHOICE] Screens

Recovery choices are available from the fault recovery screen when a fault occurs that pauses a production TP program. The recovery choices remain available on the menu until one is selected or a TP program is run/resumed. To display the current choices Press F4[CHOICE] from the alarm recovery screen.

Recovery choices shown on the menu vary depending on what fault occurred and what application is in use when the fault occurs. If the fault or facility code of the fault that occurred does not have application specific recovery choices, the software will try to determine what application is in use. The software looks at the application process selected in the TP program header. In a multi-tasking scenario, the parent TP program header is used.

Refer to Table 17.1(a) for a description of each kind of recovery [CHOICE].

> ⚠ **WARNING**
> When you select some choices, such as Fast Fault Recovery, the program will be executed immediately. Be ready for the robot to move before you select a choice; otherwise, you could injure personnel or damage equipment.

> **NOTE**
> Fast fault recovery items listed in Table 17.1(a) will be displayed only if the fast fault recovery option has been installed.

Table 17.1(a) Recovery [CHOICE] menu items

CHOICE	DESCRIPTION
ALL Application	
Resume	Resume TP program
SPOT Application	
Resume in weld	Puts the robot in stroke and weld mode.
Resume in noweld	Puts the robot in noweld mode. Stroke mode is unchanged.
Resume in nostroke	Puts the robot in no stroke and no weld mode.
Disable Alarm	Disables the fault that occurred for the max number of cycles.
Retry weld	This item automatically resets the robot fault and resumes the program. The weld that caused the fault will be retried. If the weld fails again, the program will pause and the Fault Recovery screen will be displayed.
Skip weld	This item automatically resets the fault and resumes the program. The weld that caused the fault will be skipped.
Reset water saver; resume in WELD	Resets the water saver. If successful, the program is resumed with welding enabled.
Bypass water saver; resume in NOSTROKE	Bypasses the water saver and resumes the program in NOSTROKE
Skip dress	This item automatically resets the fault and resumes the program. The dress that caused the fault will not be skipped.
Retry dress	This item automatically resets the fault and resume the program. The dress that caused the fault is retried. If the dress fails again, the program will pause and the Fault Recovery screen will be displayed.
Initiate fast fault recovery	This item initiates the fast fault recovery sequence. This item will only be displayed if the fast fault recovery option has been installed.
Retry	Resume TP program and retry operation. This choice is presented for SVGN alarms like SVGN-020 (pressure shortage), for example.
Skip	Skip to next line in TP program. This choice is presented for SVGN alarms like SVGN-020 (pressure shortage), for example.
Dispense Application	
Resume in wet mode	Places the robot into WET mode
Resume in dry mode	Places the robot into DRY mode
Initiate fast fault recovery	This item initiates the fast fault recovery sequence. This item will only be displayed if the fast fault recovery option has been installed.
Arc Welding Application	
Resume	Resume TP program
Skip weld and resume	Skip paused weld and resume TP program
Initiate fast fault recovery	This item initiates the fast fault recovery sequence. This item will only be displayed if the fast fault recovery option has been installed.
Material Handing Application	
Continue From Current Position	Continue the TP program.
Disable Alarm	Disable posting the alarm for the maximum number of cycles and continue the TP program. The default is 20 cycles.
Disable Alarm 1 Cycle	Disable posting the alarm for 1 cycle and continue the TP program. This will let the current cycle complete without reposting the alarm again, but will re-

	enable posting the alarm on the next cycle.
Recheck (Check I/O Only)	Recovery choice for a part presence sensor fault. This will typically be used if the part presence sensor had a momentary fault state. It will recheck the input to see if it is in the correct state, and if so it will continue.
Disable Alarm 20 Cycles	Disable posting the alarm for the maximum number of cycles. The default is 20 cycles.
Fast Fault Recovery	This item initiates the fast fault recovery sequence. This item will only be displayed if the fast fault recovery option has been installed.
Toggle I/O and recheck	This recovery choice will re-open or re-close the gripper in the reverse order in which it was actuated, then re-check the input sensors. If they are in the correct state the program will continue, otherwise there will be another error posted.
Cancel and Recover	Cancel and recover.
Vision Application	
Retry the vision process	Retry the run_find execution of the vision process
Retry find targets	Retry finding all targets of the vision process
Move to error pos 1	Move to an error position as programmed by the user
Move to error pos 2	Move to an error position as programmed by the user
Go to runtime display	Go to the vision runtime display screen.
Press Application	
Ignore and Continue	Disable the part presence checking fault for one cycle and continue the program.
Resume	Resume the TP program.
Cancel and auto exit	If auto backward exit is installed and set up properly, this recovery choice will execute the auto backward exit path which should reverse the robot motion back to the start position of the recorded auto backward exit path.
Cancel and drop part here	Abort the program and turn off the gripper outputs so the part is dropped.
Cancel	Abort the program.

Recovery Failure

In some cases a recovery operation might fail when a choice is selected. For example, if a recovery operation tries to reset a weld controller and the weld controller remains faulted. In this case the alarm recovery menu will show a fault and allow the user to pick another recovery choice. The fault and applicable recovery choices might be from a different fault that occurred during the recovery operation, not necessarily the original fault that halted the TP program.

17.2 User Alarm Recovery

User Alarm Recovery allows the user to create a custom set of choices the operator can select from and a corresponding programmable action when a user alarm occurs. This is handled through the Alarm Recovery screen. The screen can also display custom content pertaining to the user alarm.

NOTE
User alarm recovery is only available if feature R913 is loaded on the robot.

NOTE
User alarm recovery is only available for user alarms that pause TP program execution

Example

This example illustrates how user alarm recovery can be used. In this example:

- UALM[1] is defined to be “Process Error”
- UALM[1] choices are defined as:
  - Move home
  - Perform repair procedure
  - Reject part
- R[5] is used for choice selection

Figure 17.2(a) shows an example of the Alarm Recovery menu when UALM[1] occurs.



Fig. 17.2(a) Alarm Recovery Menu with User Alarm

Figure 17.2(b) shows the custom choices available after pressing the F4,[CHOICE] key.

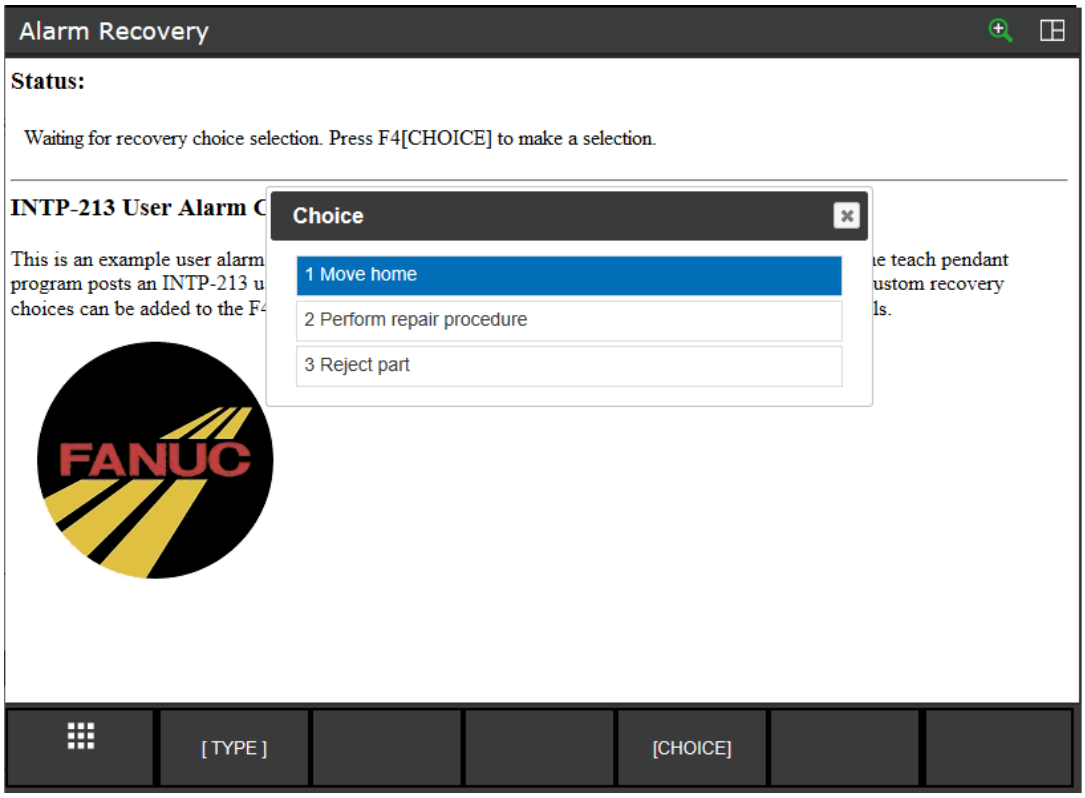


Fig. 17.2(b) Alarm Custom [CHOICE] Menu

When a choice selection is made, the choice selection index number is written into the register number (R[]) defined on the setup screen. The value written to the register is the choice selection index number in the choice definitions menu, not the pop-up window index number. When the TP program resumes, it can condition the selection to run the appropriate logic. Figure 17.2(c) shows an example of the conditional logic.

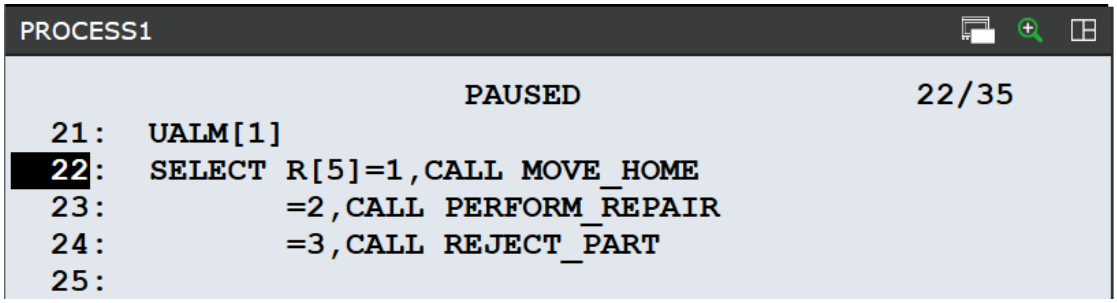


Fig. 17.2(c) Example [CHOICE] Selection Logic

Refer to Procedure 17-2 to configure user alarm recovery

Procedure 17-2 User Alarm Recovery Setup

Step

1 Press MENU
2 Press SETUP
3 Press F1, [TYPE].
4 Select User Alarm.

5 Press F2, [RECOV]. A menu similar to Figure 17.2(d) will be shown.

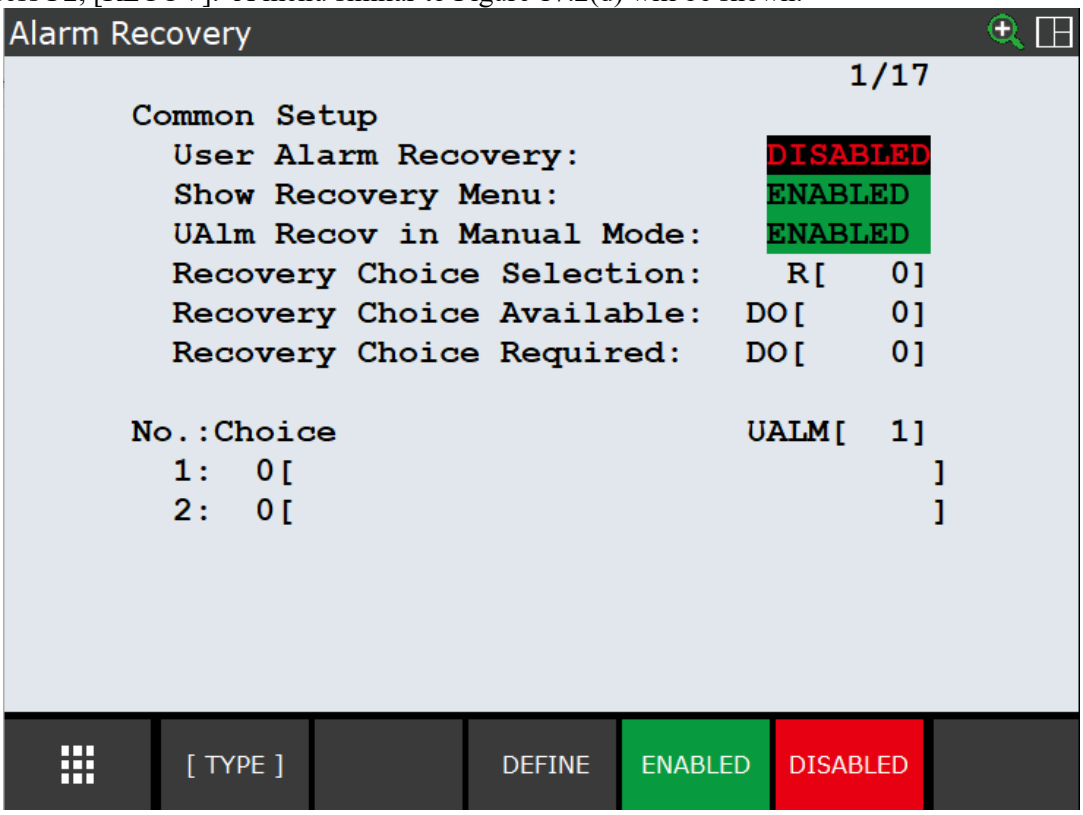


Fig. 17.2(d) Alarm Recovery Setup Menu

6 Press F3, [DEFINE] to navigate to the Recovery Choice Definitions screen. A menu similar to Figure 17.2(e) will be shown.

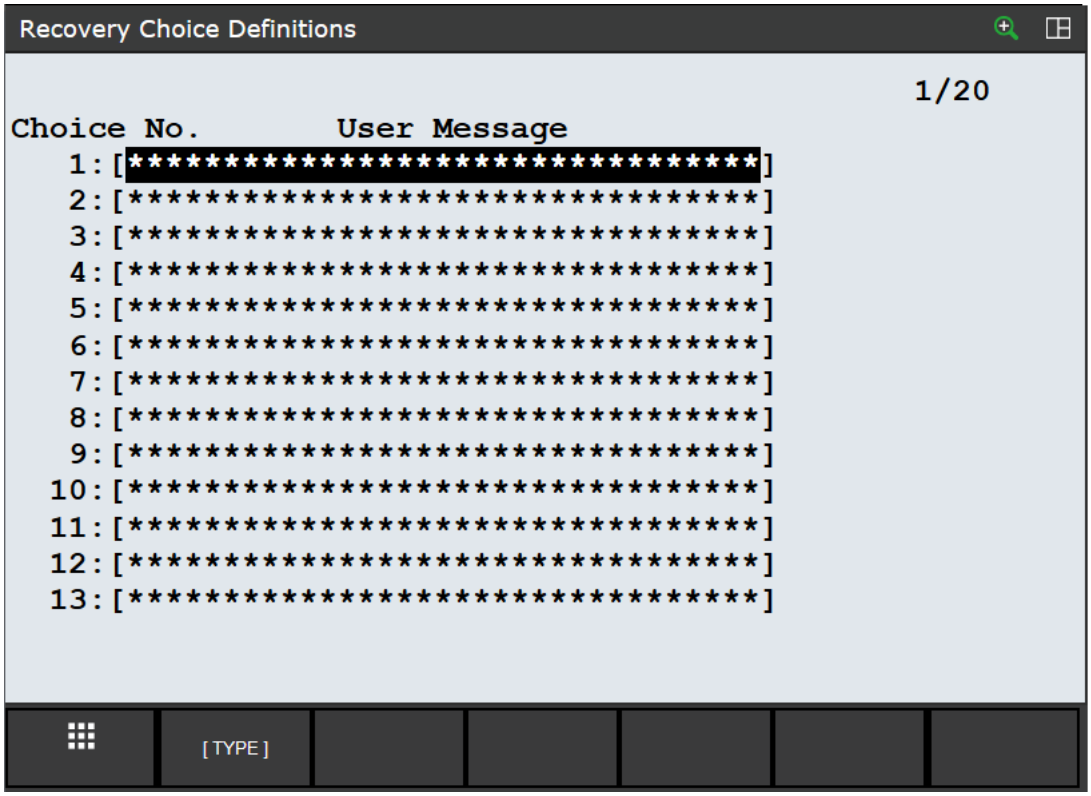


Fig. 17.2(e) Alarm Recovery Choice Definitions Menu

7 Use the UP or DOWN arrow keys to select any Choice.
8 Press ENTER.
9 Enter in the custom message for the Choice.
10 Press the PREV key to return to the previous menu.
11 Use the UP or DOWN array keys to navigate to the UALM[ ] field.
12 Enter in the User Alarm number.
13 Use the UP or DOWN arrow keys to select any choice and press F4, [CHOICE].
14 Select the custom Choice message.
15 Selections can be cleared by pressing F5, [CLEAR].
16 ENABLE User Alarm Recovery.
17 Select a register to use for the Choice selection.
18 To return to the main User Alarm menu, press PREV

Alarm Recovery Menu

The alarm recovery menu contains both common settings and settings specific to individual user alarms. Common settings apply to alarm recovery (except user alarm) and/or user alarm recovery features. Those that apply to user alarm recovery affect all user alarms. For example, disabling user alarm recovery will disable this feature for all user alarms. Individual user alarm settings can be unique for each user alarm.

Table 17.2(a) Alarm Recovery Common Settings

ITEM	DESCRIPTION
User Alarm Recovery	ENABLED: Show custom recovery choices and menu content when a user alarm occurs. DISABLED: Do not show custom recovery choices and menu content when a user alarm occurs.
Show Recovery Menu	ENABLED: Automatically display the alarm recovery menu when fault occurs. DISABLED: Do not automatically display alarm recovery menu when fault occurs. This setting is also applied on alarm recovery operation.
UAlm Recov in Manual Mode	ENABLED: Allow user alarm recovery when running TP programs in manual mode. DISABLED: Do not allow user alarm recovery when running TP programs in manual mode.
Recovery Choice Selection	Register that stores the choice selection index number. This register is reset to zero each time a user alarm occurs. It is populated with the choice selection index number when selected from the recovery screen. This setting is also applied on alarm recovery operation.
Recovery Choice Available	This digital output will turn on when there is a recovery choice available at the alarm recovery menu. The alarm recovery menu does not need to be displayed on the teach pendant for the output to be on. This setting is also applied on alarm recovery operation.
Recovery Choice Required	This digital output will turn on when a recovery choice is required at the alarm recovery menu in order to resume the TP program. The alarm recovery menu does not need to be displayed on the teach pendant for the output to be on.

Table 17.2(b) Individual User Alarm Settings

ITEM	DESCRIPTION
No.	The index number shown on the choice pop-up menu. Ten choices are available for each user alarm.

ITEM	DESCRIPTION
Choice	Choice selection index number as indicated on the Recovery Choice Definitions screen. This number is written to the choice selection register when a choice is selected on the recovery screen after a user alarm occurs.
UALM[ ]	User alarm number being configured

Recovery Menu Setup

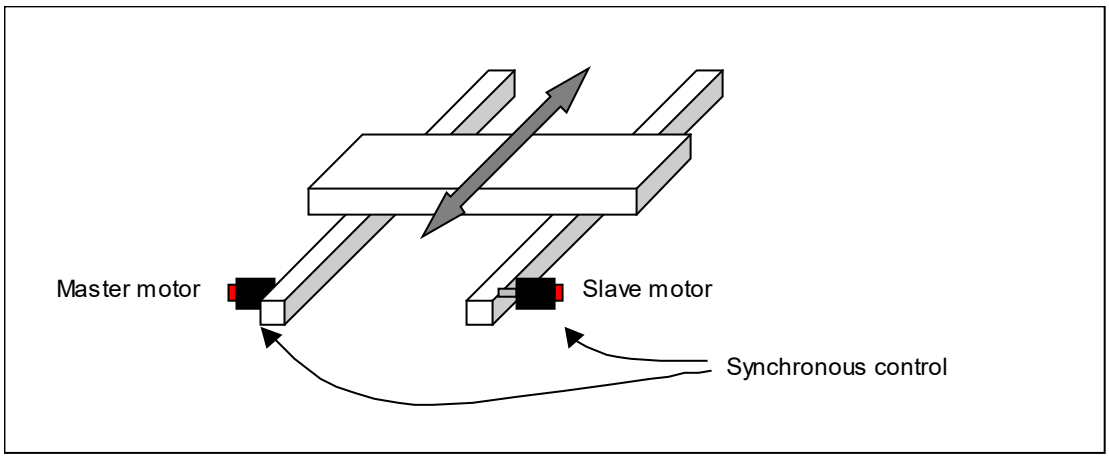
When a user alarm occurs, the recovery menu displays a webpage. The webpage can be modified to display custom content. The webpage shown can be the same for all user alarms, different for all user alarms, or a combination of the two.

A template webpage, FR:\ualmdiag.stm, is loaded on the robot controller when this feature is installed.

The software looks for webpages by name. The software will search for FR:\ualmX.stm where X is equal to the user alarm number. For example, if user alarm 9 occurs, the software will try to display the webpage FR:\ualm9.stm. If not found, the software will try to display the default user alarm webpage FR:\ualmdiag.stm.

18 DUAL MOTOR DRIVE

The Dual Motor Drive (Dual Drive) feature allows single axis control of two motors. One motor (axis) is called the master motor (master axis) and the other motor (axis) is called the slave motor (slave axis).



- The main function of the Dual Drive feature is to compensate for synchronous error due to servo delay between master and slave axes.
- Dual Drive feature can realize a large size and high-load system which cannot be realized by a single motor.
- Dual Drive feature suppresses the slave motor status on the current position screen, teaching screen, and so on. By jogging or teaching only of the master motor, both motors move in synchronization.

To use this function, Dual Drive option (A05B-2700-J836) is required. Extended axis (J518), Independent axis (H895) and Positioner (H874, H896) can be used as Dual Drive.
You can also configure a Dual Drive motor pair of 1-axis positioners. Please refer to “18.2 SETUP OF DUAL DRIVE PAIR FOR 1-AXIS POSITIONER” for detail.

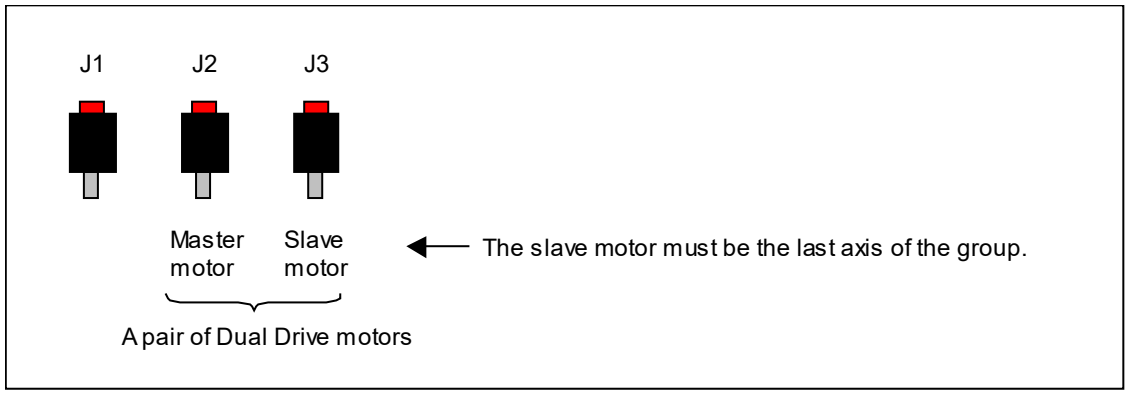
⚠ CAUTION
This function synchronizes based on positions. This function doesn’t synchronize based on torques. So This function doesn’t support several synchronous motion similar to transmitting power with two motors to one shared gear.

18.1 SETUP OF DUAL DRIVE PAIR FOR EXTENDED AXIS (J518), INDEPENDENT AXIS (H895) AND POSITIONER (H874, H896)

18.1.1 Condition

There are following conditions on axis configuration.

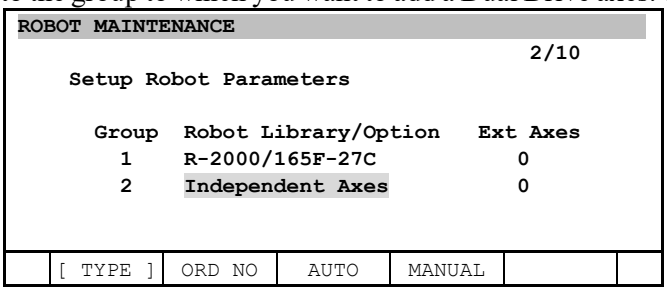
- In the case of Extended axes, you can add 1 pair of Dual Drive motor in one motion group.
- In the case of Independent axes or Positioner axes, you can add up to 2 pairs of Dual Drive motor in one motion group.
- The maximum number of axes (motors) per motion group is 9 including slave axes of Dual Drives.
  - If the robot has more than 6 motors, such as a robot model which itself contains Dual Drive axes, the number of auxiliary axes that can be added to the same motion group will be decreased.
  - For example, if the robot itself has 8 motors, only 1 auxiliary axis can be added to the same motion group. If you want to add a Dual Drive pair, please add it into a different motion group.
- If the system has multiple motion groups, Dual Drive can be set up for each group.
- The pair of Dual Drive motors must belong to the same motion group.
- The slave motor must be the last axis in the group. If there exist multiple Dual Drive motor pairs in one group, the slave axes must be the last axes in the group though they can be set up in an arbitrary order.



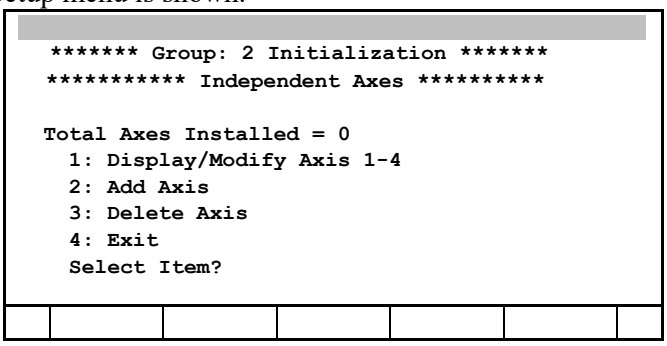
18.1.2 Add Dual Drive axes

The following is an example of Independent axes.

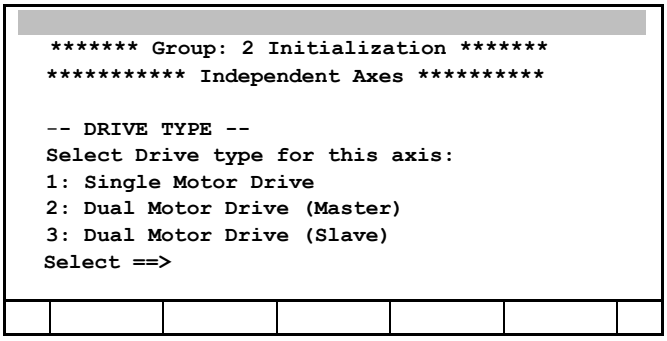
1. Open the Auxiliary Axis Setup menu.
   (a) In the Controlled Start, press [MENU] key and select "9. MAINTENANCE".
   (b) Move the cursor to the group to which you want to add a Dual Drive axes. Then press F4, MANUAL.



   (c) Auxiliary Axis Setup menu is shown.



2. When you select "2: Add Axis" on the Auxiliary Axis Setup menu, the following drive type selection screen is shown.



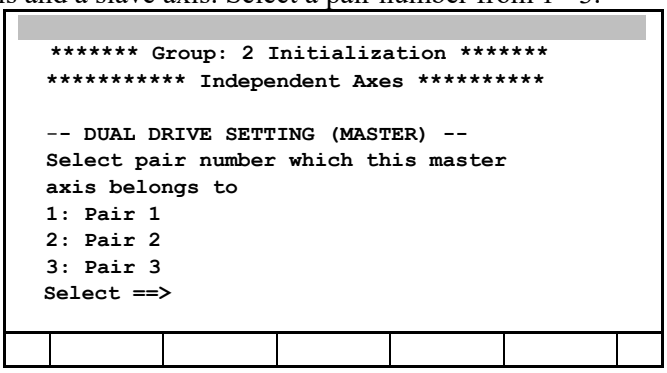
NOTE
- This screen will not be displayed if the Dual Drive option (A05B-2700-J836) is not installed.
- When adding a Dual Drive axis, you must add a master axis first. Slave axis cannot be added when the master axis does not exist. If you set 2 pairs of Dual Drive, add 2 master axes first, and then add 2 slave axes.

Select the drive type according to the following:

[If add a Single Motor]
Select "1: Single Drive"

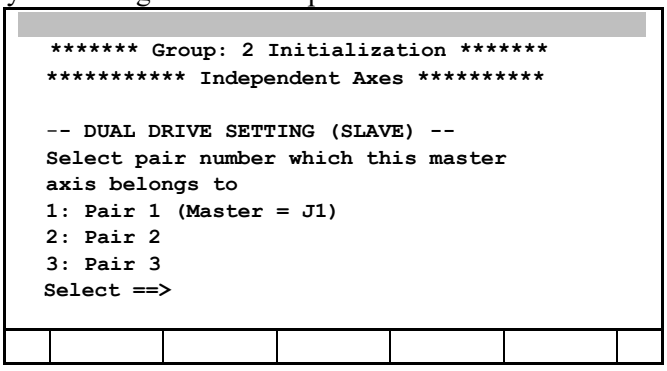
[If add a Master Motor]

Select “2: Dual Motor Drive (Master)”
After selection, the pair number setting screen for master axis is displayed. A pair number is an id number that links a master axis and a slave axis. Select a pair number from 1 - 3.



NOTE
- You need to select a pair number in order starting from Pair 1.
- You cannot select a pair number that is already in use on another axis.

[If add a Slave Motor]
Select “2: Dual Motor Drive (Slave)”
After selection, the pair number setting screen for the slave axis is displayed.
For a slave axis, you can only select from a pair number in which a master axis has already been assigned. If a label such as “(Master = J*)” is shown besides the pair number (* indicates an axis number), the master axis has already been assigned. Select a pair number from the ones with such labels.



3. The rest of the procedure is the same as for normal axis addition. Please refer to the following section in chapter B. SPECIAL OPERATION of the "OPERATOR’S MANUAL (Basic Function)" (B-84464EN) to set up the axis.
   - B.9 POSITIONER SETUP
   - B.10 EXTENDED AXIS SETUP
   - B.11 INDEPENDENT ADDITIONAL AXIS SETUP

NOTE
- For a slave axis, the settings related to mechanical unit (Gear ratio, Stroke limit, Maximum speed, Load ratio, etc.) are automatically set to the same values as those of the master axis. The motor direction of rotation, Amplifier Number, Amplifier Type, and Brake Number must be set manually.

4. When you complete axis addition, you will return to the screen in step 1. If you add next axis, start the procedure from step 2.

Example

[Example 1]

System configuration:
- Group 2 has three motors as Independent Additional Axes.
- J2-J3 of Group 2 is a pair of Dual Drive motors.
- J2 is a master motor and J3 is a slave motor.

Setting:
- J1: Add an axis as a single drive axis.
- J2: Add an axis as a Dual Drive master axis. Select pair number 1.
- J3: Add an axis as a Dual Drive slave axis. Select pair number 1.

When jogging or teaching, the Dual Drive pair is treated as "J2".

[Example 2]
System configuration:
- Group 2 has four motors as Independent Additional Axes.
- J1-J3 and J2-J4 of Group 2 are pairs of Dual Drive motors.
- J1 and J2 are master motors and J3 and J4 are slave motors.

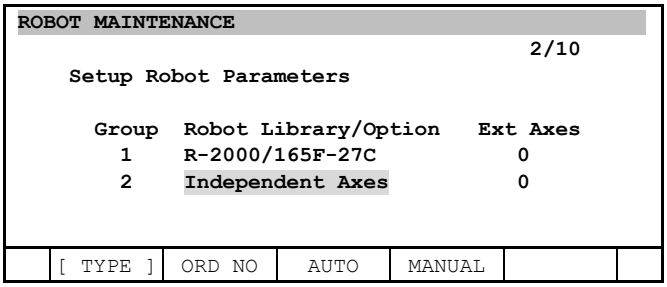
Setting:
- J1: Add an axis as a Dual Drive master axis. Select pair number 1
- J2: Add an axis as a Dual Drive master axis. Select pair number 2.
- J3: Add an axis as a Dual Drive slave axis. Select pair number 1.
- J4: Add an axis as a Dual Drive slave axis. Select pair number 2.

When jogging or teaching, the Dual Drive pairs are treated as "J1" and "J2".

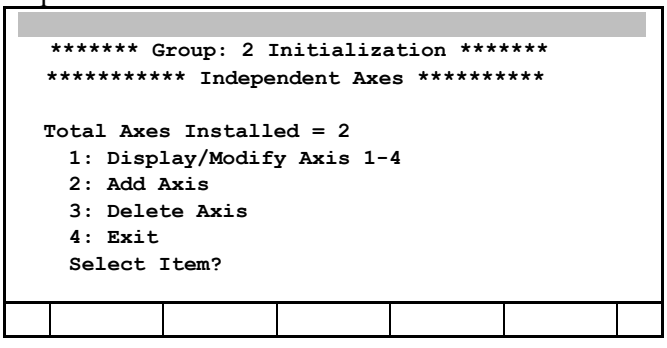
18.1.3 Display / Modify / Clear Dual Drive settings

The following is an example of Independent axes.

1. Open the Auxiliary Axis Setup menu.
   (a) In the Controlled Start, press [MENU] key and select "9. MAINTENANCE".
   (b) Move the cursor to the group where you want to check the Dual Drive settings. Then press F4, MANUAL.



   (c) Auxiliary Axis Setup menu is shown.



2. Select "1: Display/Modify Axis" on the Auxiliary Axis Setup menu, and then enter the axis number you want to confirm.

3. When the Display/Modify screen shows, enter “0: Next Item” several times, and then confirm the "Drive Type". One of the following will be shown in "Drive Type" depending on the current drive type setting.
   - Case-1 Single Motor: Drive Type = Single Motor
   - Case-2 Master Motor: Drive Type = Dual Motor#[Pair No.] (Master)
   - Case-3 Slave Motor: Drive Type = Dual Motor#[Pair No.] (Slave)

   The [Pair No.] will display the numbers 1~3.

```
*** Group: 2 Axis: 1 Initialization ***
*********** Independent Axes **********

10. Minimum Accel Time = 384
11. Load Ratio = 2.000
12. Amplifier number = 2
13. Amplifier type = 2
14. Brake number = 2
15. Drive Type = Dual Motor#1 (Master)
Select (10~15: Modify, 0: Exit)?
```

[If modify / clear Dual Drive setting]

When you select "Drive Type" item number, the following screen will be displayed. You can change the drive type setting from this screen. If you want to clear the dual drive setting, select “Single Motor Drive” on this screen. See step 2 in “18.1.2 Add Dual Drive motor axes” for details.

```
******* Group: 2 Initialization *******
*********** Independent Axes **********

-- DRIVE TYPE --
Select Drive type for this axis:
1: Single Motor Drive
2: Dual Motor Drive (Master)
3: Dual Motor Drive (Slave)
Select ==>
```

NOTE
- If both master and slave axes are assigned to a dual drive pair, the drive type or the pair number of the master axis cannot be changed.
- When clearing a dual drive setting, you must clear the slave axis first, then the master axis. The master axis cannot be cleared first.

⚠ CAUTION

If you change the settings (Gear ratio, Stroke limit, Maximum speed, Load ratio, etc.) of a master axis when it is used in a dual drive pair, make the same changes for the slave axis. You can make the same changes on the slave axis as on the master axis, by moving the cursor to the target group and pressing F3 [AUTO] in the Robot Maintenance screen.

18.1.4 Mastering of Dual Drive axes

After a motor or a mechanical unit is replaced, do single axis mastering with keeping the variables of the Dual Drive motor axes setup. Though you can perform single axis mastering respectively on each axis of a pair of Dual Drive motor axes, it is recommended that you do mastering of both axes at the same time to ensure that the mastering positions of the pair have no difference.

When the mechanical unit gets a permanent stress due to a mechanical slippage etc. of one axis of the pair, reset the variables of Dual Drive motor axes, then perform single axis mastering after adjust the position of each axis individually. After mastering, set the variables of Dual Drive motor axes again.

18.2 SETUP OF DUAL DRIVE PAIR FOR 1-AXIS POSITIONER

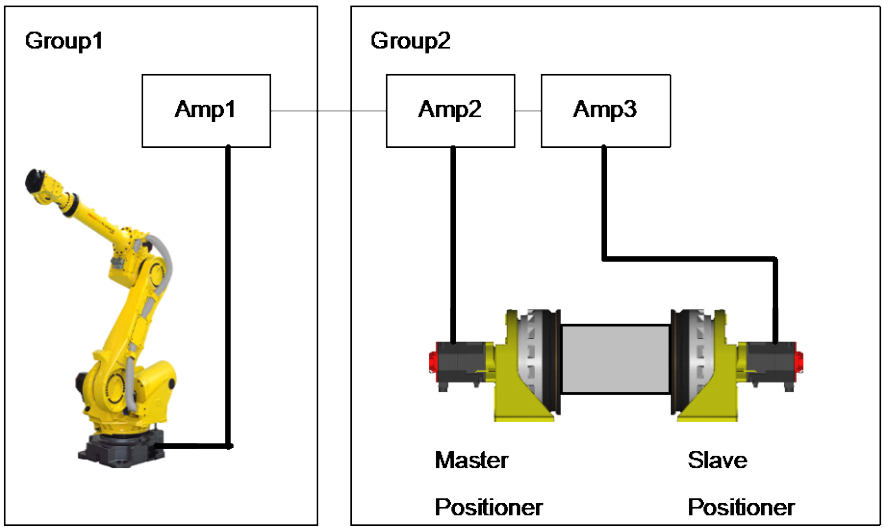
If the Dual Drive option (A06B-2700-J836) is installed, you can configure a Dual Drive motor pair of the following 1-axis positioners. This feature supports a configuration that two same positioner mechanical units are installed opposite each other.

Dual Drive supported 1-axis positioner	Software specification No. (A05B-2700-****)	Mechanical unit specification No.
500kg payload 1-axis positioner (Hollow type)	H887	A05B-1229-J101
1500kg payload 1-axis positioner (Hollow type)	H898	A05B-1229-J103
2000kg payload 1-axis positioner (Solid type)	H899	A05B-1229-J104

Preparation

Connect two same positioners and install them opposite each other.
One positioner software is required, not two.

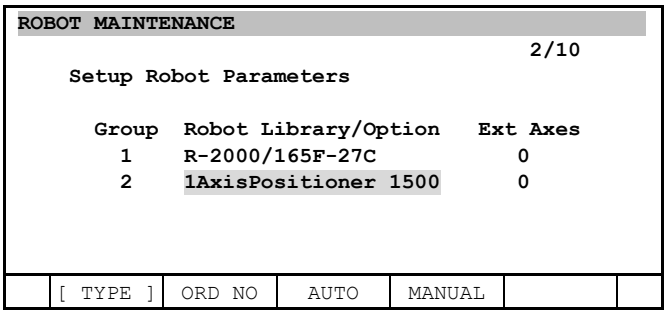
- Example of system configuration



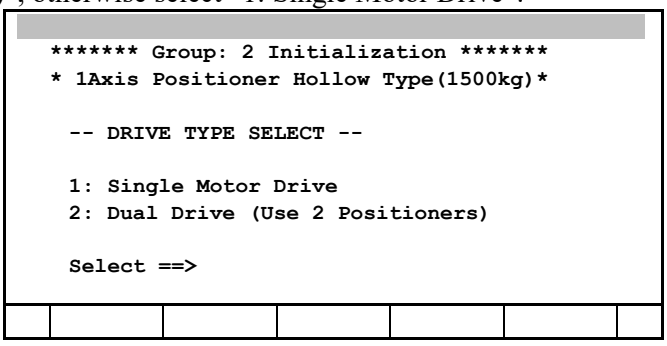
Setup of Dual Drive pair for 1-axis positioner

You can set up Dual Drive to 1-axis positioner in the Robot Maintenance menu.
Procedure:

1. In the Controlled Start, press [MENU] key and select “9. MAINTENANCE”.
2. Move the cursor to the 1-axis positioner where you want to set up a Dual Drive. Then press F4, MANUAL.



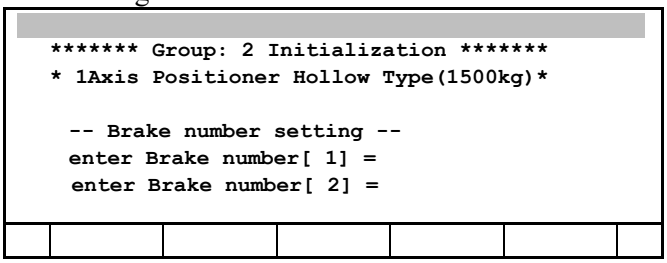
3. Some positioner software will display a payload setting screen. Enter a payload.
4. Drive type selection screen will be displayed only when the Dual Drive option (A05B-2700-J836) is installed. If you want to make a Dual Drive pair, select "2: Dual Drive (Use 2 Positioners)", otherwise select "1: Single Motor Drive".



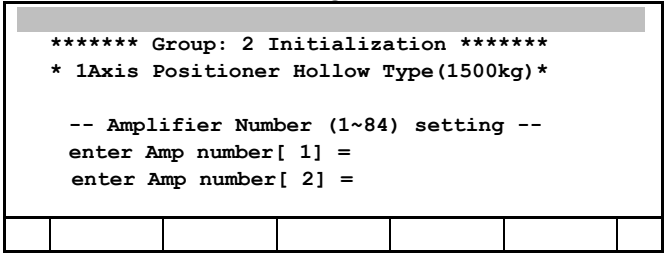
- If "2: Dual Drive (Use 2 Positioners)" is selected, the second axis will be added automatically and will be set up as Dual Drive slave axis.
- If "1: Single Motor Drive" is selected, the positioner will be set up as a normal 1-axis positioner.
  - If Dual Drive has already been set to this positioner, the Dual Drive setting will be cleared and the second axis will be deleted.

NOTE
- Do not select "2: Dual Drive (Use 2 Positioners)" when the system has only one positioner mechanical unit. If you select it by mistake, execute the Robot Maintenance again and select "1: Single Motor Drive".
- The rotational direction (motion sign) of the master positioner will be the normal one. That of the slave positioner will be the reverse one.

5. Enter the brake number. Only when "2: Dual Drive (Use 2 Positioners)" is selected in procedure 4, the setting of the second axis will be displayed. Enter the brake number of the slave positioner for the second axis setting.



6. "FSSB configuration setting" screen will be displayed. Please set up FSSB line and hardware start axis.
7. Enter the amplifier number. Only when "2: Dual Drive (Use 2 Positioners)" is selected in procedure 4, the setting of the second axis will be displayed. Enter the amplifier number of the slave positioner for the second axis setting.



8. Select the amplifier type. The "*" in "enter Amp Type[*]" is the amplifier number set in procedure 7.

```
******* Group: 2 Initialization *******
* 1Axis Positioner Hollow Type(1500kg)*

  -- Amplifier Type setting --
  1. A06B-6660 series 6 axes amplifier
  2. A06B-6540 series Alpha i SV amp. or
     A06B-6600 series Alpha i PSV amp.

  enter Amp Type[ 2] =
  enter Amp Type[ 3] =
```

Reset Dual Drive for 1-axis positioner

- If you want to reset the Dual Drive setting, make a Robot Maintenance and select "1: Single Motor Drive" in the drive type selection. The dual drive setting will be cleared and the slave positioner will be deleted automatically.

Constraints

- If an extended axis has been added to the motion group of the 1-axis positioner, you cannot set Dual Drive to the positioner. If you try to set it, a message will appear stating that it cannot be set.
- If the 1-axis positioner is set to Dual Drive, you cannot add an extended axis to the motion group of the positioner. If you try to set it, a message will appear stating that it cannot be added.

Mastering

If you replace a motor or a mechanical unit, do single axis mastering keeping the variables of the Dual Drive motor axes as they are. Though you can perform single axis mastering respectively on each axis of a pair of Dual Drive motor axes, it is recommended that you do mastering of both axes at the same time to ensure that the joint positions of the pair have no difference.

When there is a mechanical phase error between the master axis and the slave axis, or when the mechanical unit gets a constant stress due to a mechanical slippage of one axis etc., do mastering in the following steps.

1. Select "1: Single Motor Drive" in the drive type selection to reset the Dual Drive setting.
2. Move the master positioner to a position where the mechanical phase is the same as the slave positioner.
3. Select "2: Dual Drive (Use 2 Positioners)" in the drive type selection to configure the Dual Drive again.
4. Do mastering of both positioners at the same time.

19 MULTI UOP INTERFACE FUNCTION

Multi UOP Interface function enable us to use up to 5 UOP sets that have a connection with start of program. Therefore, It is possible to control (select, execute, pause and so on) multiple program independently with a single robot controller.

Specification

The number of usable UOP set
Up to 5 UOP sets are usable.

UOP Interface
The number of UOP signals depend on how many UOP sets is used.

Number of set	1	2	3	4	5
Input	18	23	32	41	50
Output	20	26	36	46	56

Software Options

The following software option is needed for Multi UOP Interface function.

Multi UOP Interface	A05B-2600-J964

If 2 or more motion groups (multiple robot, independent axis and so on) is used, the following software option is needed.

Multi-Group Motion	A05B-2600-J601

Multi UOP Interface for Dual Arm Control (J605)

Multi UOP interface for Dual Arm Control (J605) is the function that is Multi UOP Interface function but it is limited to use UOP sets up to 2. If more than 2 UOP sets are not needed, we can choose not Multi UOP Interface (J964) but Multi UOP interface for Dual Arm Control (J605).

Multi UOP interface for Dual Arm Control is different from Multi Robot Control Function that had been used since R-30*i*A in the following point:

- One CSTOPI signal was prepared for 2 UOP sets by Multi Robot Control Function, but two CSTOPI signals are prepared for 2 UOP sets by Multi UOP interface for Dual Arm Control. (For this difference, Multi UOP interface for Dual Arm Control uses more one signal than Multi Robot Control Function by CSTOPI signal.)
- In case of Multi Robot Control function, the motion groups intended by each UOP depend on coupling structure of Isolation switch. But in case of Multi UOP Interface function, the user can configure the motion groups intended by each UOP freely.

And it is possible to use UOP as the specification of Multi Robot Control function. Please refer to “19.5 System Config screen items”.

19.1 PERIPHERAL I/O

The following signals are provided in response to the number of UOP set.

INPUT SIGNAL	OUTPUT SIGNAL
<Number of set : 1>	
*IMSTP	CMDENBL#1
*HOLD#1	SYSRDY#1
*SFSPD	PROGRUN#1
CSTOPI#1	PAUSED#1
FAULT_RESET	HELD#1
START#1	FAULT#1
HOME	ATPERCH
ENBL	TPENBL
RSR1/PNS1/STYLE1	BATALM
RSR2/PNS2/STYLE2	BUSY
RSR3/PNS3/STYLE3	ACK1/SNO1
RSR4/PNS4/STYLE4	ACK2/SNO2
RSR5/PNS5/STYLE5	ACK1/SNO1
RSR6/PNS6/STYLE6	ACK1/SNO1
RSR7/PNS7/STYLE7	ACK1/SNO1
RSR8/PNS8/STYLE8	ACK1/SNO1
PNSTROBE#1	ACK1/SNO1
PROD_START#1	ACK1/SNO1
	SNACK
	RESERVE
Total 18 points	Total 20 points
<Number of set : 2>	
*HOLD#2	CMDENBL#2
START#2	SYSRDY#2
PNSTROBE#2	HELD#2
PROD_START#2	FAULT#2
CSTOPI#2	PROGRUN#2
	PAUSED#2
Total 23 points	Total 26 points
<Number of set : 3>	
RSR9	ACK9
RSR10	ACK10
RSR11	ACK11
RSR12	ACK12
*HOLD#3	CMDENBL#3
START#3	SYSRDY#3
PNSTROBE#3	HELD#3
PROD_START#3	FAULT#3
CSTOPI#3	PROGRUN#3
	PAUSED#3
Total 32 points	Total 36 points

<Number of set : 4>	
RSR13	ACK13
RSR14	ACK14
RSR15	ACK15
RSR16	ACK16
*HOLD#4	CMDENBL#4
START#4	SYSRDY#4
PNSTROBE#4	HELD#4
PROD_START#4	FAULT#4
CSTOPI#4	PROGRUN#4
	PAUSED#4
Total 41 points	Total 46 points
<Number of set : 5>	
RSR17	ACK17
RSR18	ACK18
RSR19	ACK19
RSR20	ACK20
*HOLD#5	CMDENBL#5
START#5	SYSRDY#5
PNSTROBE#5	HELD#5
PROD_START#5	FAULT#5
CSTOPI#5	PROGRUN#5
	PAUSED#5
Total 50 points	Total 56 points

I/O Configuration:

In case that DI/DO are 40 points

Number of set	1	2	3	4	5
Specialized input（UI）	18	23	32	-	-
General-purpose input（DI）	22	17	8	-	-
Specialized output（UO）	20	26	36	-	-
General-purpose output（DO）	20	14	4	-	-

In case that DI/DO are 96 points

Number of set	1	2	3	4	5
Specialized input（UI）	18	23	32	41	50
General-purpose input（DI）	78	73	64	55	46
Specialized output（UO）	20	26	36	46	56
General-purpose output（DO）	76	70	60	50	40

19.1.1 Setting of the Number of UOP Set

The number of UOP set is 1 by default. Please change [System Config screen > Multi UOP Interface settings > The Number of UOP Set] from 1 to the number you want to use when you want to change the number of UOP set. You can change the value up to 5. And the number of UOP set is changed after you cycle power.

Then the number of UI/UO signal is changed, but signal assignment isn't changed. If you clear assignment, new assignment in response to new number of UI/UO signal is given automatically (You can clear

assignment by selecting F5, CLR_ASG in I/O Link Device screen. Power cycling is needed after that. And all assignments are cleared. If this is inconvenient for you, please change assignment manually).

Here from, 'N' means the number of UOP set, in other words, the value of [System Config screen > Multi UOP Interface settings > The Number of UOP Set] ('N' is used for explanation after "19.1.3 HOLD#1 to #N").

19.1.2 Modification and Addition of Signal

In Multi UOP Interface Function, the meaning of the following signals are changed.

Input:	HOLD#1:	UI[2] HOLD is changed to HOLD#1.
	CSTOPI#1:	UI[4] CSTOPI is changed to CSTOPI#1.
	START#1:	UI[6] START is changed to START#1.
	PNSTROBE#1:	UI[17] PNSTROBE is changed to PNSTROBE#1.
	PROD_START#1:	UI[18] PROD_START is changed to PROD_START#1.
Output:	CMDENBL#1:	UO[1] CMDENBL is changed to CMDENBL#1.
	SYSRDY#1:	UO[2] SYSRDY is changed to SYSRDY#1.
	PROGRUN#1:	UO[3] PROGRUN is changed to PROGRUN#1.
	PAUSED#1:	UO[4] PAUSED is changed to PAUSED#1.
	HELD#1:	UO[5] HELD is changed to HELD#1.
	FAULT#1:	UO[6] FAULT is changed to FAULT#1.

In Multi UOP Interface Function, the following signals are added.

<Number of set : 2>

	Specialized I/O signal	UI/UO[*]
Input	HOLD#2	UI[19]
	START#2	UI[20]
	PNSTROBE#2	UI[21]
	PROD_START#2	UI[22]
	CSTOPI#2	UI[23]
Output	CMDENBL#2	UO[21]
	SYSRDY#2	UO[22]
	HELD#2	UO[23]
	FAULT#2	UO[24]
	PROGRUN#2	UO[25]
	PAUSED#2	UO[26]

<Number of set : 3>

	Specialized I/O signal	UI/UO[*]
Input	RSR9	UI[24]
	RSR10	UI[25]
	RSR11	UI[26]
	RSR12	UI[27]
	HOLD#3	UI[28]
	START#3	UI[29]
	PNSTROBE#3	UI[30]
	PROD_START#3	UI[31]
	CSTOPI#3	UI[32]
Output	ACK9	UO[27]
	ACK10	UO[28]
	ACK11	UO[29]
	ACK12	UO[30]
	CMDENBL#3	UO[31]
	SYSRDY#3	UO[32]
	HELD#3	UO[33]
	FAULT#3	UO[34]
	PROGRUN#3	UO[35]
	PAUSED#3	UO[36]

<Number of set : 4>

	Specialized I/O signal	UI/UO[*]
Input	RSR13	UI[33]
	RSR14	UI[34]
	RSR15	UI[35]
	RSR16	UI[36]
	HOLD#4	UI[37]
	START#4	UI[38]
	PNSTROBE#4	UI[39]
	PROD_START#4	UI[40]
	CSTOPI#4	UI[41]
Output	ACK13	UO[37]
	ACK14	UO[38]
	ACK15	UO[39]
	ACK16	UO[40]
	CMDENBL#4	UO[41]
	SYSRDY#4	UO[42]
	HELD#4	UO[43]
	FAULT#4	UO[44]
	PROGRUN#4	UO[45]
	PAUSED#4	UO[46]

<Number of set : 5>

	Specialized I/O signal	UI/UO[*]
Input	RSR17	UI[42]
	RSR18	UI[43]
	RSR19	UI[44]
	RSR20	UI[45]
	HOLD#5	UI[46]
	START#5	UI[47]
	PNSTROBE#5	UI[48]
	PROD_START#5	UI[49]
	CSTOPI#5	UI[50]
Output	ACK17	UO[47]
	ACK18	UO[48]
	ACK19	UO[49]
	ACK20	UO[50]
	CMDENBL#5	UO[51]
	SYSRDY#5	UO[52]
	HELD#5	UO[53]
	FAULT#5	UO[54]
	PROGRUN#5	UO[55]
	PAUSED#5	UO[56]

19.1.3 HOLD#1 to #N

Here from, the word “Program#1” appears. “Program#1” means the program selected by UOP set 1. “Program#2” to “Program#N” are same as “Program#1”. Please refer to “19.2 SELECT PROGRAM”.

Multi UOP Interface has one HOLD signal per UOP set (#1 to #N). The function of HOLD signal is changed by setting [System Config screen > Multi UOP Interface settings > The function of HOLD signal].

- In the case when [System Config screen > Multi UOP Interface settings > The function of HOLD signal] is [All] (Default)
  If whichever of HOLD signals (HOLD#1 to #N) is turned OFF, all the executing programs are paused.

- In the case when [System Config screen > Multi UOP Interface settings > The function of HOLD signal] is [Each]
  If HOLD#X is turned OFF, “Program#X' is paused (X = 1 to N).
  The running program which is not selected as “Program#X”, can NOT be held. Please turn off ENBL to pause the program.
  (However, if “Program#X” is not selected, when HOLD#X is OFF, all the programs in execution are paused.)

But, when a program is tried to execute, regardless of [System Config screen > Multi UOP Interface settings > The function of HOLD signal] setting, if any HOLD signal is OFF, it is impossible to execute program. Then if program execution is needed, please turn ON all HOLD signals.

19.1.4 CSTOPI#1 to #N

Multi UOP Interface has one CSTOPI signal per UOP set (#1 to #N). The function of CSTOPI signal is changed by setting "CSTOPI for ABORT" or "Abort all programs by CSTOPI" in System Config screen.

- When "Abort all programs by CSTOPI" is FALSE (Default)
  If CSOTPI#X is turned ON, "Program#X" is aborted (X = 1 to N).
  It also release (clear) programs in UOP set X queue from the wait state by RSR.
  Please refer to "19.1.6 RSR".

- When "Abort all programs by CSTOPI" is TRUE
  If whichever of CSTOPI signals (HOLD#1 to #N) is turned ON, all the executing programs are aborted. It also release (clear) programs in all UOP set queue from the wait state by RSR.

TRUE for "Abort all programs by CSTOPI" is recommended for Multi UOP Interface function.

The way of terminating program is changed by setting "CSTOPI for ABORT" in System Config screen.

- When "CSTOPI for ABORT" is FALSE (Default)
  If CSOTPI signal is turned ON, this signal aborts the program in execution as soon as execution of the program completes.

- When "CSTOPI for ABORT" is TRUE
  If CSOTPI signal is turned ON, this signal immediately aborts the program in execution.

19.1.5 START#1 to #N

Note)

In the system with Multi UOP Interface, multiple programs are controlled at the same time, and it is more complex to control program than control single program. So it is recommended that the Multi UOP Interface is in condition that "START for CONTINUE only:" in system config screen is "TRUE". Only paused program can be started by START signal under this setting. In case of starting the ended program, use RSR or PROD_START signal, please.

Multi UOP Interface has one START signal per UOP set (#1 to #N).

"Program#X" is started when START#X is turned off (X = 1 to N).

The program is continued to execute from paused line in case that START signal is input with paused program.

When "START for CONTINUE only:" in system config screen is "TRUE", the program, which is aborted (neither running nor paused), can NOT be started and the message "PROG-023 Task is not paused" is displayed in TP screen in case that START signal is inputted with the ended program (neither running nor paused).

19.1.6 RSR

Multi UOP Interface has four RSR signal per UOP set (#1 to #N).
(However, if the number of UOP set is 1, eight RSR is usable.)

Number of set	RSR1 to 4	RSR5 to 8	RSR9 to 12	RSR13 to 16	RSR17 to 20
1	#1		-	-	-
2	#1	#2	-	-	-
3	#1	#2	#3	-	-
4	#1	#2	#3	#4	-
5	#1	#2	#3	#4	#5

And each UOP set has RSR queue, so the number of RSR queue is N.
Please refer to “19.3.1 Robot Service Request (RSR)”.

19.1.7 PNSTROBE#1 to #N,PROD_START#1 to #N

Multi UOP Interface has one PNSTROBE and one PROD_START signal per UOP set (#1 to #N).
Please refer to “19.3.2 Program Number Select (PNS)”.

19.1.8 CMDENBL#1 to #N

Multi UOP Interface has one CMDENBL signal per UOP set (#1 to #N).

CMDENBL#1 is ON if all the following conditions are satisfied.

- Remote condition is satisfied.
- No alarm. (Every FAULT#X are OFF. (X = 1 to N))
- No single step.
- If Isolation switch is used, motion groups that is intended by UOP set 1 are connected.
  (“Remote condition” and “motion groups that is intended by UOP set 1”are mentioned after.)

In case of CMDENBLE#2 to #N, only last condition is different.

- If Isolation switch is used, motion groups that is intended by UOP set 2 (to N) are connected.

1. Remote Condition:
   - The Teach Pendant enable switch is set OFF.
   - Setting of Remote/Local is set to Remote.
   - The *SFSPD input is ON.
   - The ENBL input is ON.
   - [Remote start device] in System Config screen is [PeripheralEquip].

2. Motion groups that is intended by UOP set
   Motion groups that is intended by UOP set is changed by setting [System Config screen > Multi UOP Interface settings > Intended motion groups].
   Please refer to “19.2.1 Setting of Motion Group Intended by Each UOP Set”.

19.1.9 SYSRDY#1 to #N

Multi UOP Interface has one SYSRDY signal per UOP set (#1 to #N).

SYSRDY#1 is ON if all the following conditions are satisfied.

- All the axes are ready. (If system has Isolation switch, all the axes except for isolated are ready.)
- If Isolation switch is used, motion groups that is intended by UOP set 1 are connected.

In case of SYSRDY#2 to #N, only last condition is different.
- If Isolation switch is used, motion groups that is intended by UOP set 2 (to N) are connected.

19.1.10 PROGRUN#1 to #N,PAUSED#1 to #N

Multi UOP Interface has one PROGRUN and one PAUSED signal per UOP set (#1 to #N).
PROGRUN and PAUSED signals show the program execution status as the following.

- When program is running:
  PROGRUN#X (X = 1 to N)
  ・ In case that "Program#X" is in execution.

  In case that the program, which is not "Program#X", is in execution, every PROGRUN#X is output. (Default)

  If the setting of [System Config screen > Multi UOP Interface settings > Output for not targeted program] is [Not output], the behavior of signals is changed as follows.
  In case that the program, which is not "Program#X", is in execution, PROGRUN#X is not output. But BUSY is output.

- When program is paused:
  PAUSED#X (X = 1 to N)
  ・ In case that "Program#X" is paused.

  In case that the program, which is not "Program#X", is paused, every PAUSED#X is output. (Default)

  If the setting of [System Config screen > Multi UOP Interface settings > Output for not targeted program] is [Not output], the behavior of signals is changed as follows.
  In case that the program, which is not "Program#X", is paused, PAUSED#X is not output.

19.1.11 HELD#1 to #N

Multi UOP Interface has one HELD signal per UOP set (#1 to #N).

HELD#X is ON if one of the following conditions is satisfied. (X = 1 to N)
- HOLD#X is OFF.
- The HOLD switch on the Teach Pendant is pushed.
- The HOLD switch on the Operator Panel is pushed.

19.1.12 FAULT#1 to #N

Multi UOP Interface has one FAULT signal per UOP set (#1 to #N).

FAULT#X turns ON when the alarm occurs by "Program#X" execution or other than program execution. (X = 1 to N)

When system is in alarm status, at least one of FAULT#X must be ON.

19.1.13 ATPERCH

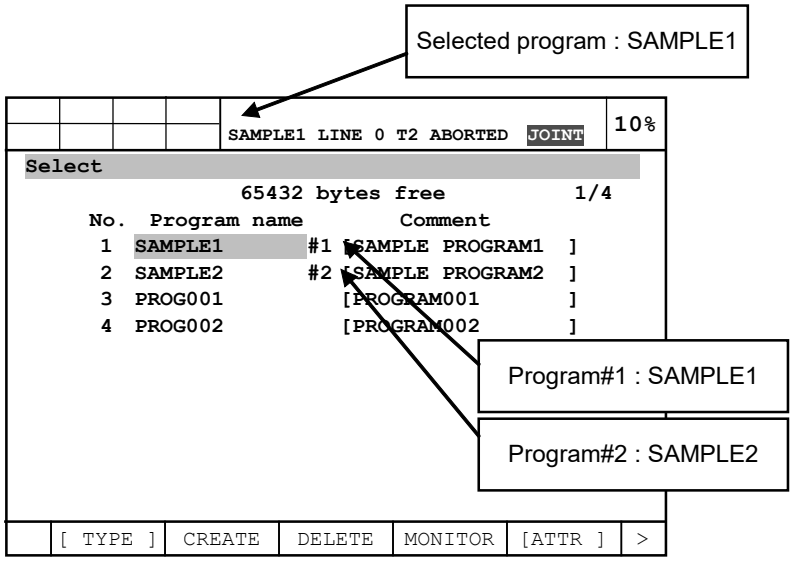
The specified signal ATPERCH is set to ON when the robot of motion group 1 is at the reference position 1. Please assign reference position output except motion group 1.

19.2 SELECT PROGRAM

In Multi UOP Interface function, it is possible for each UOP set to select program individually.
One UOP set is given one program # for selected program.

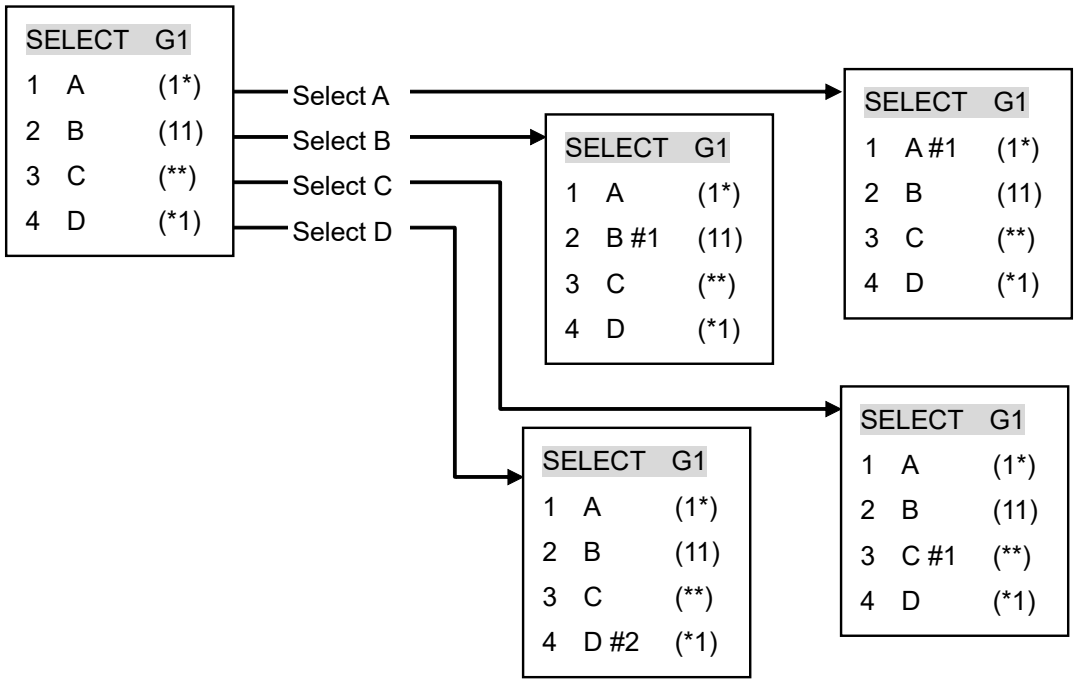
Here from, “Program#1” means the program selected by UOP set 1.
“Program#1” is run by UOP START#1, PROD_STAR#1.
On program selection screen, “#1” is displayed to the right of program name that is used as “Program #1”.
“Program#2” to “Program#N” are same as “Program#1”.
(Data ID: $MULTI_UOP_INTERFACE.SELECTED_PROGRAM_NAME[a] / STRING / RO / a:UOP set number)

On the contrary, a program that is displayed on program edit screen is referred as “selected program”.
[FWD]/[BWD] keys on teach pendant or start button on Operator Panel run “selected program”.
“Selected program” is displayed on left side of second line of TP screen.



If you set cursor to a program and presses the [ENTER] key, “selected program” is changed to the program. Then, the program changes to “Program#X” automatically (What # is set depend on motion group of the program and what motion group is intended by each UOP sets).

For example, when the number of UOP sets is 2, and UOP set 1 intends Motion group 1 and UOP set 2 intends Motion group 2, the behavior is as the following figure.



(1*) The program has only group 1.
(*1) The program has only group 2.
(11) The program has group 1 and 2.

19.2.1 Setting of Motion Groups Intended by Each UOP Set

What UOP set intends what motion group, in other words, what # is set to a program is changed by setting [System Config screen > Multi UOP Interface settings > Intended motion groups].

[Intended motion groups] has 5 settings, and each one corresponds one UOP set. If the number of UOP set is smaller than 5, [Intended motion groups] settings are available as many as the number of UOP set, and the other alignments is ineffective.

The each bit of [Intended motion groups] value is equivalent to each motion group.

Bit31	・・・	Bit1	Bit0
Motion Group32	・・・	Motion Group2	Motion Group1

For example, the default values of [Intended motion groups] are set as the following, it is assumed that one UOP set is equivalent to one motion group.

Example1 (Default)

UOP set 1: 1
UOP set 2: 2
UOP set 3: 4
UOP set 4: 8
UOP set 5: 16

In case that PROGRAM1 is equivalent to motion group 1,4,
PROGRAM2 is equivalent to motion group 2,5,
and PROGRAM3 is equivalent to motion group 3,

[Intended motion groups] value is the following.

Example 2

UOP set 1: 9
UOP set 2: 18
UOP set 3: 4
UOP set 4: 0
UOP set 5: 0

Every UOP set can select a program having no motion group. However, when a program having no motion group is selected by TP, if [Intended motion groups > UOP Set #x] is 0, the program become “Program#X” by priority.

Detailed explanation is mentioned in “19.5 System Config screen items”. Please refer it.

19.2.2 Select Program at Running or Paused

When “Program#1” is paused, if you try to select another program as “Program#1”:

- When program is selected by UOP, you can not select the program, and the message “TPIF-013 Other program is running” is displayed.
- When program is selected by Teach Pendant and Teach Pendant Enable switch is OFF, you can not select the program, and the message “TPIF-013 Other program is running” is displayed.
- When program is selected by Teach Pendant and Teach Pendant Enable switch is ON, the original “Program#1” is aborted, and the selected program becomes “Program#1”.(This is the case that “Multi Program Selection” is disabled. If “Multi Program Selection” is enabled, “Program#1” doesn’t change.)

When “Program#1” is running, if you select another program as “Program#1”, you can not select the program regardless of selection way, and the message “TPIF-013 Other program is running” is displayed.

“Program#2” to “Program#N” are same as “Program#1”.

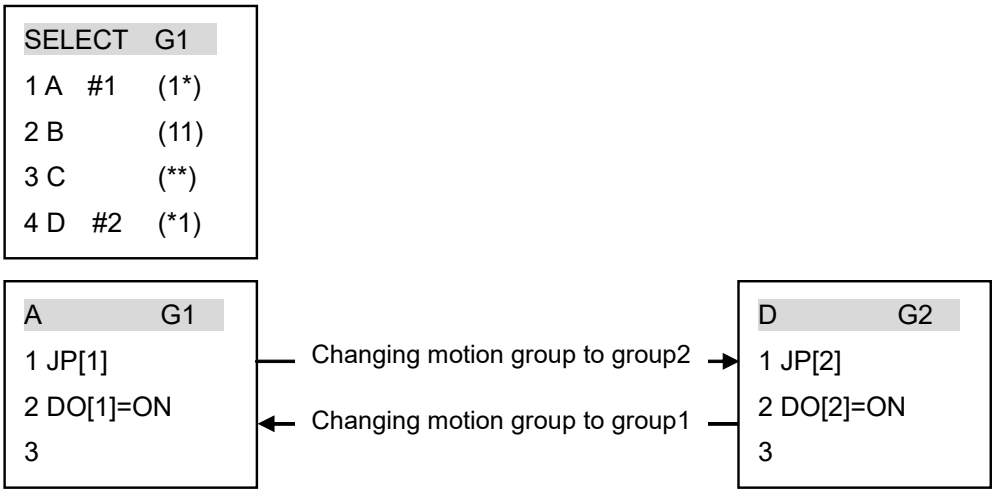
19.2.3 Program Edit Screen Display When Operating Motion Group is Changed

When the user teach motions to each motion group, the user needs to operate to select motion group that can be operated by teach pendant.
It is possible to select motion group to select “CHANGE GROUP” in function menu.

Currently selected motion group is displayed on teach pendant like “G1”,”G2”.

By changing motion group, “selected program” is also changed according to selected motion group. Changed ”selected program” is displayed on program edit screen.

For example, UOP set 1 intends Motion group 1, and UOP set 2 intends Motion group 2.
Suppose “Program #1 “ has only group1 and “Program #2 has” only group2.
By Selecting group1, “Program #1”
By Selecting group2, “Program #2”
automatically becomes “selected program” and the program is displayed on program edit screen.



If "Selected program" has both group1 and group2 or "selected program" has no motion group, change of motion group doesn't change "selected program".
To change "selected program" in these cases, display program selection screen, set cursor to a program that you want to display and press the [ENTER] key.

19.3 RUN PROGRAM

19.3.1 Robot Service Request (RSR)

If the number of UOP set is 1, when RSR1 to 8 are turned on, the program corresponded to the RSR number is started if the program is aborted.
If another program is running or paused, the request is joined to the queue. And the program is started when the running program is completed.

Multi UOP Interface has four RSR signals per UOP set (#1 to #N) if the number of UOP set is 2 and over. RSR1 to 4 are for UOP set 1. RSR5 to 8, RSR9 to 12, … are same as RSR1 to 4 (Please refer to "19.2.6 RSR".). And each UOP set has RSR queue, so the number of RSR queue is N.
By this function, N programs can run at the same time.

The program started by RSR1 to 4 becomes "Program#1".

In the following situation, program is not started.

- The program started by RSR1 to 4 does not have motion group intended by UOP set 1 and it has motion group intended by other UOP set.

In this case, message "TPIF-44 Program is unsuitable for robot" is displayed.

RSR5 to 8, RSR9 to 12, … are same as RSR1 to 4.

19.3.2 Program Number Select (PNS)

Multi UOP Interface has one PNSTROBE and one PROD_START signal per UOP set (#1 to #N).

The program selected by PNSTROBE#X becomes “Program#X”. This program can be started by PROD_START#X and it is resumed from pause by START#X (X = 1 to N).

In the following situation, program is not selected.

- The program selected by PNSTROBE#X does not have motion group intended by UOP set X and it has motion group intended by other UOP set.

In this case, message “TPIF-44 Program is unsuitable for robot” is displayed.

19.4 STOP PROGRAM

19.4.1 Pause Program by UOP

When whichever of HOLD signals (HOLD#1 to #N) is turned OFF, running program is paused.
Please read “19.1.3 HOLD#1 to #N” for detail.

19.4.2 Abort Program by UOP

If “CSTOPI for ABORT” in System Config is TRUE, programs are aborted when CSTOPI turns ON.
The default setting of ‘CSTOPI for ABORT’ is FALSE.
Please read “19.1.4 CSTOPI#1 to #N” for detail.

19.4.3 Other

When hold button on Teach Pendant or Operator Panel, all running programs are paused.

19.5 SYSYTEM CONFIG SCREEN ITEMS

Name Default Range	Enabled/Disabled Disabled Disabled/Enabled
Meaning	Show whether Multi UOP Interface function is enabled or not. (Data ID: $MULTI_UOP_INTERFACE.IS_ENABLED / BOOLEAN / RO)

Name Default Range	The Number of UOP Set 1 1 to 5
Meaning	The number of UOP set to use. Set this value to 1 if you want to disable Multi UOP Interface function. (Data ID: $MULTI_UOP_INTERFACE.NUMBER_OF_UOP_SETS / INTEGER / RW)

Name Default Range	The function of HOLD signal [All] [All]/[Each]
Meaning	Select the function of HOLD#1 to #N. All: All programs are paused when one of the HOLD#1 to #N signals turns OFF. Each: “Program#X” is paused when HOLD#X turns OFF.

Name Default Range	Intended motion groups [UOP Set 1] 1 [UOP Set 2] 2 [UOP Set 3] 4 [UOP Set 4] 8 [UOP Set 5] 16 0 to 0xFFFF
Meaning	Bit mask for intended motion group of each UOP set. In the default, UOP set 1 intends Motion group 1, UOP set 2 intends Motion group 2, and so on.

[Supplement] Basis of selection for program #

We can select program by UOP or TP, and program number is set on the basis of [Intended motion groups]. Program number setting rule is the following:

< Common rule for UOP and TP >

Rule 1) If motion group of [Intended motion groups > UOP Set #x] and motion group of program match at least one group, the program can be set #n.

Rule 2) No motion group program can be set ALL program number.

Rule 3) If motion group of every [Intended motion groups > UOP Set #x] and motion group of program match none, the program can be set ALL program number.

Example [Intended motion groups > UOP Set #1-5] = 1, 2, 3, 4, 0

	Motion group of program A to G G1 , 2 , 3 , 4 , 5 , 6 , 7 , 8	Program number #1	#2	#3	#4	#5
PROGRAM A	(1 , * , * , * , * , * , * , *)	#1		#3		
PROGRAM B	(* , 1 , * , * , * , * , * , *)		#2	#3		
PROGRAM C	(1 , 1 , * , * , * , * , * , *)	#1	#2	#3		
PROGRAM D	(1 , * , 1 , * , * , * , * , *)	#1		#3	#4	
PROGRAM E	(* , * , * , * , * , * , * , *)	#1	#2	#3	#4	#5
PROGRAM F	(* , 1 , * , 1 , * , * , * , *)		#2	#3		
PROGRAM G	(* , * , * , 1 , * , * , * , *)	#1	#2	#3	#4	#5

The above figure illustrates PROGRAM C can be set #1, #2, or #3, and conversely, #2 can select PROGRAM B, C, E, F or G, for instance.

[Intended motion groups > UOP Set #1] = 1 >> match at least one group to PROGRAM A, C, D
[Intended motion groups > UOP Set #2] = 2 >> match at least one group to PROGRAM B, C, F
[Intended motion groups > UOP Set #3] = 3 >> match at least one group to PROGRAM A, B, C, D, F
[Intended motion groups > UOP Set #4] = 4 >> match at least one group to PROGRAM D
[Intended motion groups > UOP Set #5] = 0 >> match at least one group to none
(by Rule 1)

[Intended motion groups > UOP Set #1-5] >> can select no motion group PROGRAM E, G
(by Rule 2 or 3)

< Rule for TP only >

Rule 4) If selected program is already set program number, program number don't change.

Rule 5) If selected program is not set program number yet, the program is set program number along the following priority. And if some program number have same priority, small number comes first. (In the following Priorities from 1 to 4, #X is the number that "Program #X" is aborted or is not selected.)

[Priority 1] Motion group of [Intended motion groups > UOP Set #x] is equal to motion group of program. (If PROGRAM C is selected by TP, set #3 not #1 or #2. If PROGRAM E is selected by TP, set #5 not #1, #2, #3 or #4.)

[Priority 2] Motion group of [Intended motion groups > UOP Set #x] inhere motion group of program. (If PROGRAM B is selected by TP, set #2 not #3.)

[Priority 3] Motion group of [Intended motion groups > UOP Set #x] and motion group of program match at least one group.

[Priority 4] If program isn't along the above priority from 1 to 3, the program is set to smallest number X that "Program #X" is aborted or is not selected. (A program along Rule 2 or 3 is treated here unless along priority 1.)

[Priority 5] #X is the number that "Program#X" is paused. On that basis, the smallest number that complies with priority 3. (Priority 5 is applied when TP is enabled and "Multi Program Selection" is disabled. Then paused program is aborted.)

When none of the above Priorities is corresponded, it means that all "Program#X" that #X complies with Priorities from 1 to 4 are in execution. Then warning message is displayed and # isn't set to selected program.

Name Default Range	Output for not targeted program [Output] [Output]/[Not output]
Meaning	Output form of PROGRUN#1 to #N signals at the execution of an unselected program and PAUSE#1 to #N signals at the pause of an unselected program. [Output] : When an unselected program is executed, all PROGRUN#1 to #N signals are output. When an unselected program is paused, all PAUSE #1 to #N signals are output. [Not output] : When an unselected program is executed, PROGRUN#1 to #N signals are not output, but BUSY is output. When an unselected program is paused, PAUSE #1 to #N signals are not output.

Name Default Range	Traditional Mode FALSE FALSE/TRUE
Meaning	Compatible flag for Multi Robot Control Function used in the old robot controller. TRUE: Compatible mode - There is single CSTOPI signal. - [The Number of UOP Set] is 2. FALSE: Normal Multi UOP Interface mode

Name Default Range	Screen switching by group TRUE FALSE/TRUE
Meaning	Enable/disable flag for EDIT screen switching by group change.

20 ERROR CODE OUTPUT FUNCTION

The Error Code Output function outputs the contents of a robot alarm to another device as a digital output signal code with a strobe. To use this function, the Error Code Output Function option (J527) and expanded I/O board are required.

20.1 SPECIFICATIONS

20.1.1 Types of Alarms

All alarms except those with WARN severity are eligible for external output.

20.1.2 Input and Output Signals

An expanded I/O board is necessary to use this function.
33 points of DO are used as the external code output signal for errors, and 1 point of DI is used as the input signal.

Set the signals used in this function using the following Alarm Settings screen items:

[Alarm Settings screen > Error Code Output > Output signal number] : **Output signal number**
(Data ID: $ERROR_CODE_OUTPUT.OUTPUT_SIGNAL_NUMBER / INTEGER / RW)
[Alarm Settings screen > Error Code Output > Input signal number]: **Input signal number**
(Data ID: $ERROR_CODE_OUTPUT.INPUT_SIGNAL_NUMBER / INTEGER / RW)

- Output signals
  The Output signal number defines which DO is used.
  The 33 signals from the DO with the setup number are used.
  If the setup number is 0, this function is disabled.

- Types of output signal
  Both the code and strobe signals are held on until a reset signal is input.

Table 20.1.2 DO signal values (when [Output signal number]=1)

DO	Types of alarms
1 to 32	Output alarms in binary (32 bit).
33	Used for the strobe.

- Input signal (1bit)
  If there are multiple errors, the input signal is used as a search signal.
  The Output signal number item defines which DI is used.

- Signal timing diagrams
  When an alarm occurs, the timing is as shown in the diagram in Figure 20.1.2 (a).

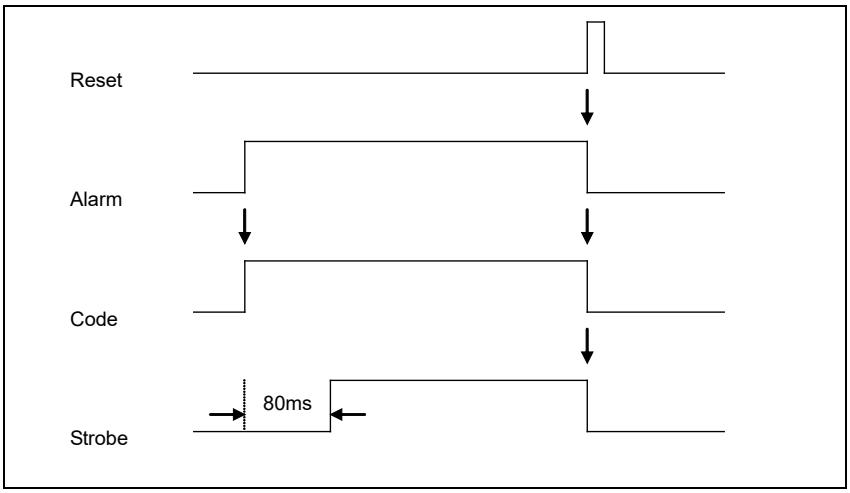


Fig. 20.1.2 (a) Timing diagram of an alarm

If multiple alarms occur simultaneously, the first alarm is output first. Then, each time a search signal is input, the next alarm is output in the order of occurrence. These signals are output cyclically, and after one cycle, a strobed reset signal is output. Up to 100 alarms can be stored.

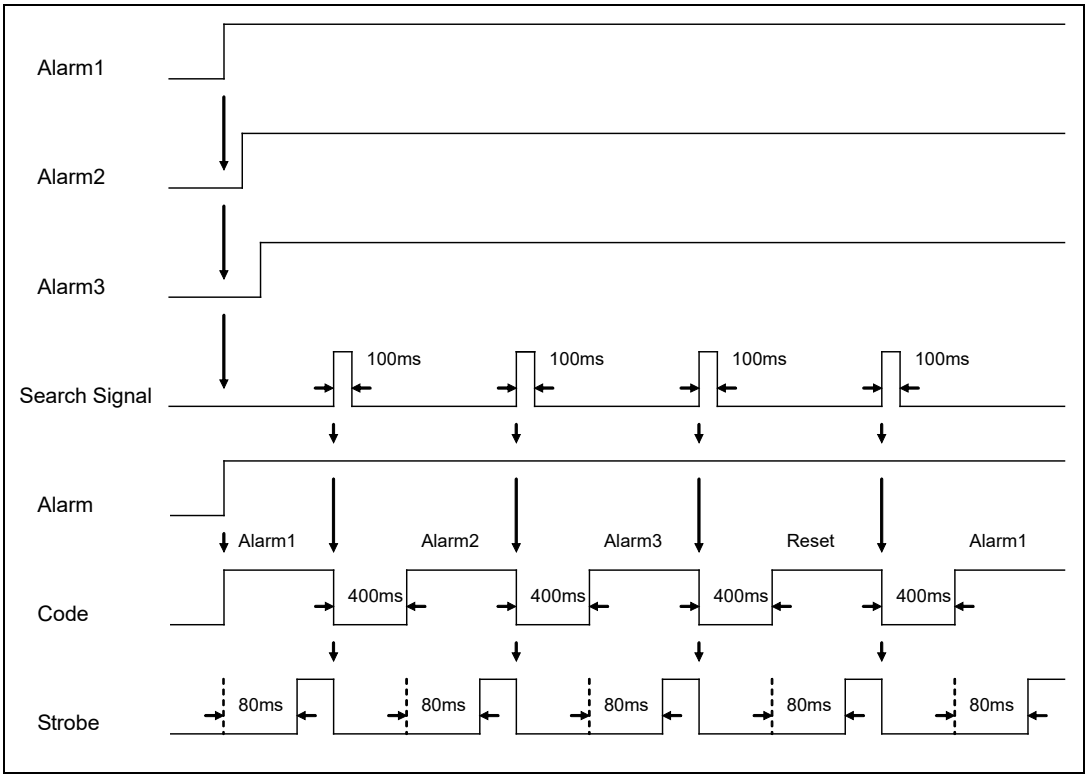


Fig. 20.1.2 (b) Timing diagram of multiple alarms

20.2 ALARM CODE DEFINITIONS

When Output signal number = 1

DO	Types of alarms
1 to 16	Output the alarm number in binary (16bit).
17 to 24	Output the alarm ID in binary (8bit).
25 to 32	Output the alarm Severity in binary (8bit).
33	Used for the strobe.

20.2.1 Alarm Severity

DO / Severity	25	26	27	28	29	30	31	32
NONE	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON
WARN	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF

Errors with NONE and WARN severity will not be output.

DO / Severity	25	26	27	28	29	30	31	32
PAUSE.L	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF
PAUSE.G	OFF	ON	OFF	OFF	OFF	ON	OFF	OFF
STOP.L	OFF	ON	ON	OFF	OFF	OFF	OFF	OFF
STOP.G	OFF	ON	ON	OFF	OFF	ON	OFF	OFF
SERVO	OFF	ON	ON	OFF	ON	ON	OFF	OFF
ABORT.L	ON	ON	OFF	ON	OFF	OFF	OFF	OFF
ABORT.G	ON	ON	OFF	ON	OFF	ON	OFF	OFF
SERVO2	ON	ON	OFF	ON	ON	ON	OFF	OFF
SYSTEM	ON	ON	OFF	ON	ON	ON	ON	OFF

DO[25] and DO[26] are used for the severity of a running program.

DO [25]	DO [26]	
OFF	OFF	The program continues running.
OFF	ON	The program will be paused.
ON	ON	The program will be aborted.

DO[27] and DO[28] are used for the motion severity.

DO [27]	DO [28]	
OFF	OFF	The motion continues running.
ON	OFF	The motion will be stopped.
OFF	ON	The motion will be canceled.

DO[29] is used for the severity of a servo motor.

OFF	The servo motor continues running.
ON	The servo motor will be powered down.

DO[30] is the severity of the target tasks.

OFF	Effective only for the target task.
ON	Effective for all tasks.

DO[31] is the request for recovery.

OFF	No need to power down.
ON	Cold start is necessary.

DO[32] means is request for the displaying on the TP.

OFF	No need to display.
ON	Need to display.

20.2.2 Alarm ID

The Alarm ID will be output as a number as shown in the following.

Typical alarm IDs:

Number	Alarm ID
0	OS
3	PROG
7	MEMO
9	TPIF
11	SRVO
12	INTP
15	MOTN
19	JOG
20	APPL
23	SPOT
24	SYST
26	PALT
53	ARC
57	MACR
58	SENS
59	COMP

Other alarm IDs:

Number	Alarm ID
1	SRIO
2	FILE
4	COND
5	ELOG
6	MCTL
8	GUID
10	FLPY
13	PRIO
14	TPAX
16	VARS
17	ROUT
18	WNDW
21	LANG
25	SCIO
27	UAPL
33	DICT
34	KCLI
35	TRAN

Number	Alarm ID
36	TKSP
37	COPT
38	APSH
42	CMND
43	RPM
44	LNTK
45	WEAV
46	TCPP
47	TAST
48	MUPS
49	MIGE
50	LSR
51	SEAL
52	PANE
54	TRAK
55	CALB
56	SP
60	THSR

20.2.3 Alarm Number

Alarm numbers will be directly output as binary data.

Example

For SRVO 002 (severity SERVO)
Alarm number is 2.
Alarm ID SRVO is 11.
The severity is SERVO.

DO [1]	OFF	Alarm number
DO [2]	ON	2
DO [3]	OFF	
DO [4]	OFF	
DO [5]	OFF	
DO [6]	OFF	
DO [7]	OFF	
DO [8]	OFF	
DO [9]	OFF	
DO [10]	OFF	
DO [11]	OFF	
DO [12]	OFF	
DO [13]	OFF	
DO [14]	OFF	
DO [15]	OFF	
DO [16]	OFF	

DO [17]	ON	Alarm ID
DO [18]	ON	11
DO [19]	OFF	
DO [20]	ON	
DO [21]	OFF	
DO [22]	OFF	
DO [23]	OFF	
DO [24]	OFF	
DO [25]	OFF	Alarm Severity
DO [26]	ON	SERVO
DO [27]	ON	
DO [28]	OFF	
DO [29]	ON	
DO [30]	ON	
DO [31]	OFF	
DO [32]	OFF	

21 DATA MONITOR FUNCTION

Data Monitor is a tool for improving process quality. You can use it to monitor and record important process parameters. It can alert you to a parameter going out of limit and it can record data for use in a quality record. To use this function, data monitor option (J675) is required.

Data Monitor operates much like a strip chart recorder or a data acquisition system.
To use the Data Monitor feature you make selections in two teach pendant screens and add two teach pendant program instructions. Specifically,

- Enable or disable any specific features in the Data Monitor Utility screen.
- Select items to monitor (such as arc current feedback) with one of the Data Monitor schedules.
- Add Sample Start[schedule number] and Sample End instructions to your TP program to control when monitoring occurs.

You can monitor up to eighteen items at once with a Data Monitor schedule. The maximum sampling frequency is 250 Hz. You can specify separate frequencies for limit checking and for recording. As the items specified in the schedule are recorded, the following data is also collected: time, date, distance, program name and line number. You can choose the items you want to monitor from the Data Monitor Schedule screen.

As data is recorded, it can be formatted as a report and sent to a file. A short example report is shown in Fig.21 (a). The data is tab delimited for importing into a spreadsheet application.

DATA MONITOR REPORT

Number	Tick	Time	Program	Line	Voltage [Volts]	Wire feed [Amps]
1	48	.192	TEST	2	0.000	0.000
2	98	.392	TEST	3	20.000	200.000
3	148	.592	TEST	3	20.000	200.000
4	198	.792	TEST	3	20.000	200.000
5	248	.992	TEST	3	20.000	200.000
6	298	1.192	TEST	3	20.000	200.000
7	348	1.392	TEST	4	0.000	0.000

Fig.21 (a) Report example

Definitions

This section contains definitions of terms you should know to use Data Monitor.

Item - A specific data element to be monitored. There are the following table items.

Table 21(a) Data monitor items

ITEM	DESCRIPTION
INTEGER	Data IDs or KAREL program variables, whose data type is integer, will be monitored or recorded.
REAL	Data IDs or KAREL program variables, whose data type is real, will be monitored or recorded.
I/O	Assigned I/O port will be monitored or recorded.
REGISTER	The value of Register will be monitored or recorded.
JOINT POS	Feedback value about assigned Axis will be monitored or recorded.
TORQUE COMM	Torque command value about assigned Axis will be monitored or recorded. Unit is [Ap]. Data will be updated by about 100 ms cycle.
Q-CURRENT	A value of Q phase current about assigned Axis will be monitored or recorded.
TC ONLINE	Be monitored or recorded about TC_ONLINE. Please refer to OPERATOR'S MANUAL (Basic function)(~~B-83284EN~~) "9.12.4 Other Instructions and Functions" about TC_ONLINE.
Bundle I/O	Continuous physical port corresponding to assigned logical port will be monitored or recorded.
SquareCurrent	Square current value about assigned Axis will be monitored or recorded. Unit is [Ap^2].
CART CAL POS	Cartesian position calculated by feedback. Assigned position element(X,Y,..,R) will be monitored or recorded.
JOINT COMM POS	Joint command position. Assigned Axis will be monitored or recorded. Data will be updated by about 100 ms cycle.
CART COMM POS	Cartesian command position. Assigned position element(X,Y,..,R) will be monitored or recorded. Data will be updated by about 100 ms cycle.

Schedule -A set of parameters that define how to monitor specific items and where to save recorded data.
Trigger - A condition that must be met to begin or end monitoring.
Limit - A defined high or low value for a monitored item.

Monitoring Limits

Data monitoring can check each sampled item against upper and lower limits. If the average of the item samples is out of limits for a specified time period (Tmin in Figure 21 (b)), a warning or a pause alarm will occur, if enabled. You must specify a nominal value, a warning limit, a pause limit, and a time duration for each monitored data item. A warning limit and a pause limit are indicated by the difference from nominal value. If the monitored data exceeds the range of (nominal value + limit value), an alarm is posted.

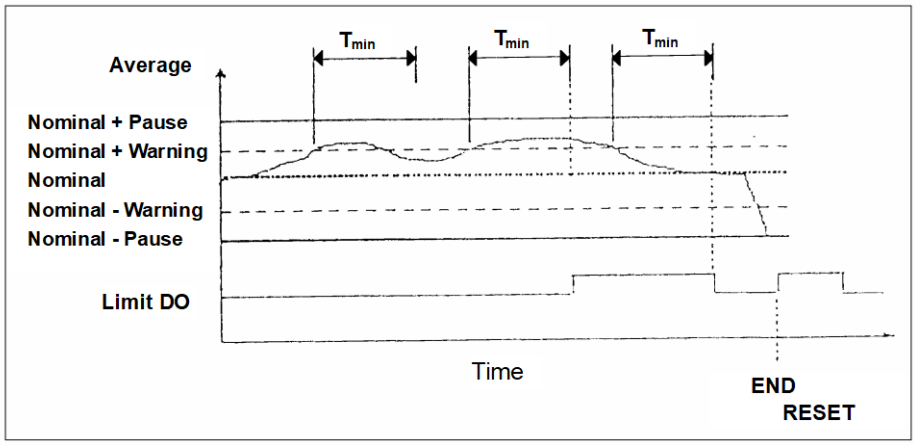


Fig.21 (b) Process Limit DO

If a warning limit is crossed for the specified time, a WARN severity error is posted and the limit digital output is turned ON. If the average of the item samples returns to within the WARN limit for the specified time the digital output is turned OFF.

If a pause limit is crossed for the specified time a PAUSE severity error is posted, the limit digital output is set to ON, and the program is paused.

When the program ends, if a WARN or PAUSE limit error occurred during execution, the limit digital output is turned ON. It is turned OFF by a system RESET.

21.1 DATA MONITOR SETUP

Data monitor setup screen items and their description are written as follows.

Table 21.1 Data monitor setup screen menu items

ITEM	DESCRIPTION
Data Monitor Operation Default: Enabled	This item enables and disables operation of the Data Monitor function. NOTE Data monitoring is also disabled during step mode.
Recording Default: Enabled	This item turns data recording ON and OFF.
Filing Default: Enabled	This item turns report filing ON and OFF.
Pause on File Errors Default: Disabled	This item controls the severity of certain file errors. If this item is disabled, WARNING severity errors are posted and execution of the program being monitored will continue. If it is enabled, PAUSE severity errors are posted and execution of the program being monitored stops.
Recording Priority Default: Enabled	This item switches between prioritizing Recoding/Monitoring or prioritizing Cycle time when enabling Filing. -Enabled: prioritizing Recoding -Disabled: prioritizing Cycle time. However, there are cases such as failing file output, failing monitoring, or data recording/monitoring missed by situation of output device.
Warning Limits Default: Disabled	This item turns Warning Limits ON and OFF.
Pause Limits Default: Disabled	This item turns Pause Limits ON and OFF.
Limit Error Output Default: DO[0]	This item defines the port type and port number for the digital output(Limit DO) when monitored data is out of setup range. This digital output is turned ON when a limit error is detected.
Sample Buffer Size Default:10 Min:1 Max:99	This item specifies the size of the sample buffer. Usually you do not have to change the default value.
Record Buffer Size Default:10 Min:1 Max:99	This item specifies the size of the record buffer. If “ONE BUFFER” is specified as record mode (refer table 21.2 Data monitor menu items), this item should be number of data you want to record. (Recording will be finished when record buffer become full.) if “CONTINUOUS” is specified as record mode, you don’t have to change the default value.
Setup Default: Disabled	This item enables or disables printing of data monitor setup information in the report header.
Items Default: Disabled	This item enables or disables printing of information in the report header about each of the items you want to monitor.
Schedule Default: Disabled	This item enables or disables printing of Schedule information in the report header.
Triggers Default: Disabled	This item enables or disables printing of Trigger information in the report header.
Program Name	This item enables or disables printing the program name column in the Data

ITEM	DESCRIPTION
Default: Enabled	Monitor report. See Fig. 21.1.
Line Number Default: Enabled	This item enables or disables printing the line number column in the Data Monitor report. See Fig. 21.1.
Date Default: Disabled	This item enables or disables printing the date and time of day column in the Data Monitor report. See Fig. 21.1.
Tick + time Default: Disabled	This item enables or disables printing the Tick and Time column in the Data Monitor report. See Fig. 21.1.
Event Default: Disabled	This item enables or disables printing the event column in the Data Monitor report. See Fig. 21.1. NOTE At this time, there is only one event defined. An event value of 1 indicates the data was recorded as a result of the recording frequency.
Distance Default: Enabled	This item enables or disables printing the distance column in the Data Monitor report. See Fig. 21.1.

Number	Date	Tick	Time	Event	Program	Line	Distance	Voltage(Command)[Volts]
1	18-AUG-98 10:41:46	0	0	1	TEST	2	0	0
2	18-AUG-98 10:41:46	62	0.248	1	TEST	2	0	0
3	18-AUG-98 10:41:46	124	0.496	1	TEST	2	0	0
4	18-AUG-98 10:41:46	186	0.744	1	TEST	2	0	0
5	18-AUG-98 10:41:46	248	0.992	1	TEST	3	0	0
6	18-AUG-98 10:41:46	310	1.24	1	TEST	3	0	0
7	18-AUG-98 10:41:46	372	1.488	1	TEST	3	0	0
8	18-AUG-98 10:41:48	434	1.736	1	TEST	3	0	0
9	18-AUG-98 10:41:48	496	1.984	1	TEST	3	0	0
10	18-AUG-98 10:41:48	558	2.232	1	TEST	3	0	0
11	18-AUG-98 10:41:50	620	2.48	1	TEST	3	0	0

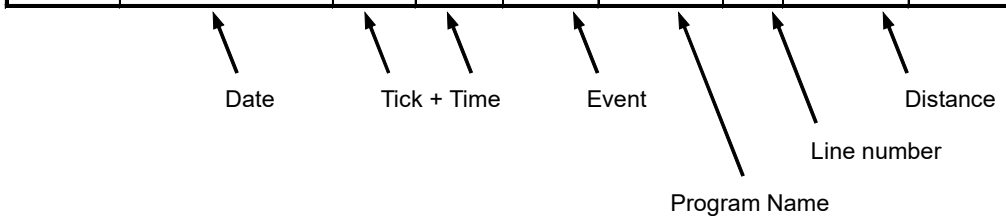


Fig. 21.1 Sample report(when each setup item is enabled)

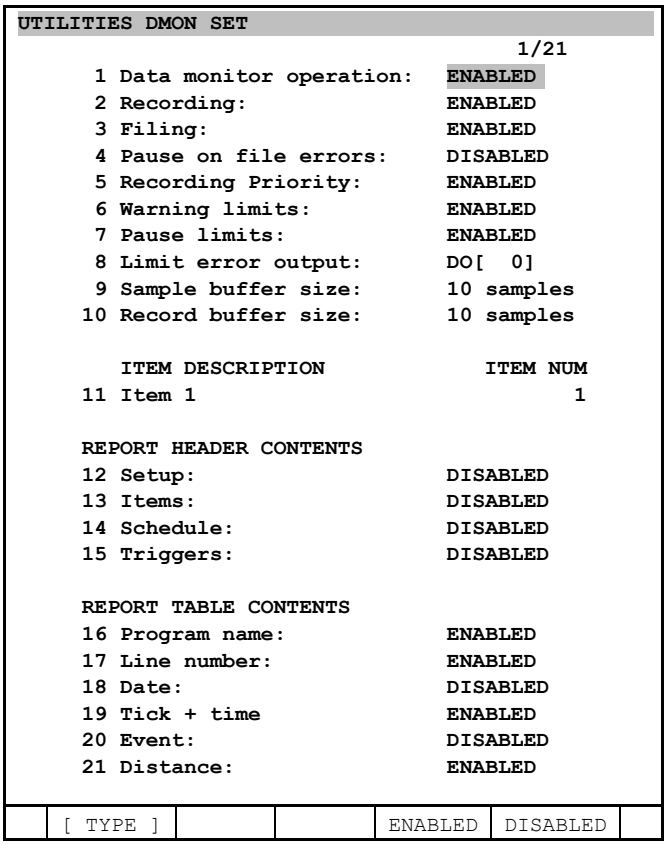
Procedure 21-1 Setting up data monitor

Condition

Data Monitor is installed on your controller.

Step

1 Press the [MENU] key.
2 Select UTILITIES.
3 Press the F1, [TYPE].
4 Select Data Monitor. The following screen will be displayed.

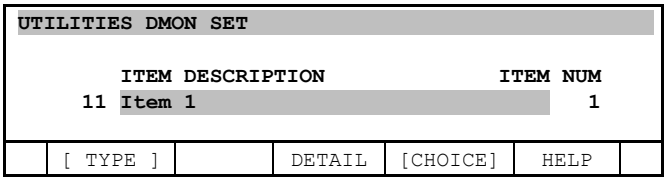


5 Select each item on the menu and set it as desired.

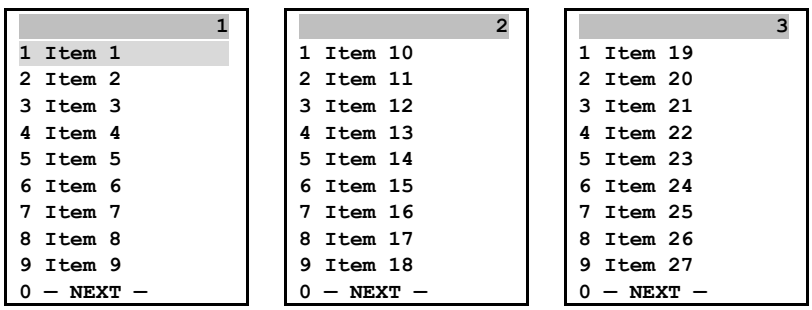
Edit Data Monitor Items

6 Data Monitor allows you to define 60 items to monitor. These items are initialized for you, but you can edit them to suit your needs. If you want to edit Data Monitor items, move the cursor to item 11. When the cursor is on item 11, the DETAIL, [CHOICE], and HELP function keys will be available. [CHOICE] allows you to choose an item from a list. DETAIL allows you to edit that item.

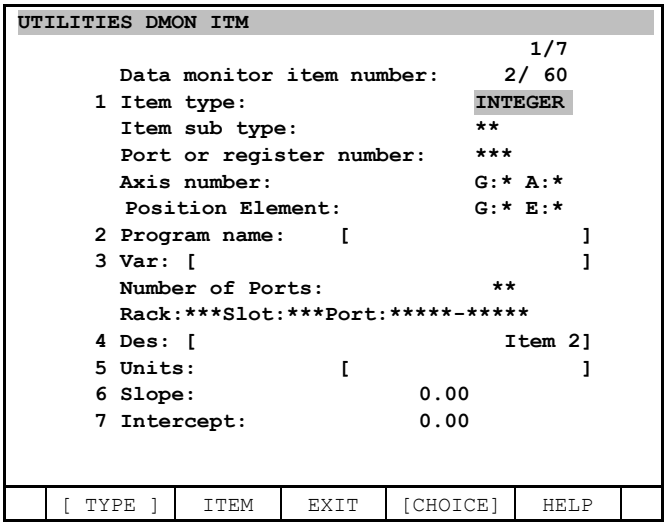
Item 11 has two columns. The right column contains an item number from 1 to 60. The left column contains the corresponding item description.



7 Press F4, [CHOICE]. The following screen will be displayed.



8 Select an item from the list by number or by moving the cursor and pressing the [ENTER] key.
9 Press the F3, DETAIL function key to edit the selected item. In this example, [2 Item 2] has been selected. The following screen will be displayed.



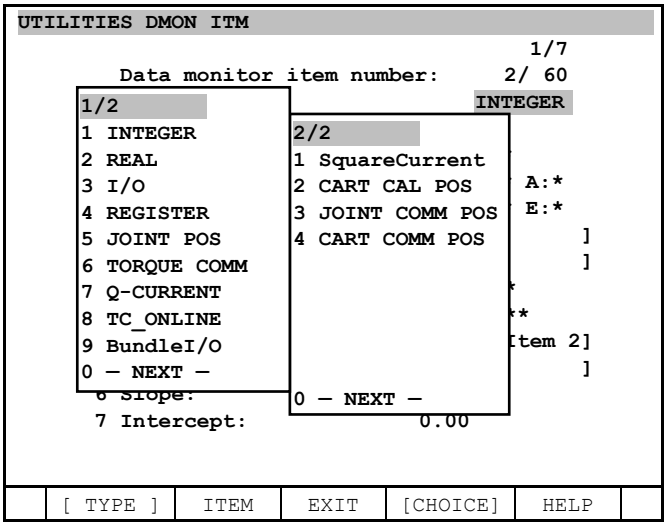
You can specify comment about the item in “4 Des” and “5 Unit”.
“6 Slope”, “7 Intercept” is for linear transformation. When Non-zero value is specified as slope, Data is transformed as follows.

Data = [Slope] * [raw data] + [intercept]

When 0 is specified as slope, raw data is used without transformation.

NOTE
Transformed data is used for comparison between PAUSE/WARN limit or condition of Start/End trigger.

10 This is the screen you use to edit an item. Press F2, ITEM, to select a different item by number. Not all of the menu items are available for all item types. If they are not available the item is not numbered, you cannot move the cursor to it, and it displays as ***.
To set the Item type, move the cursor to line 1 and press the [CHOICE]. The following screen will be displayed.

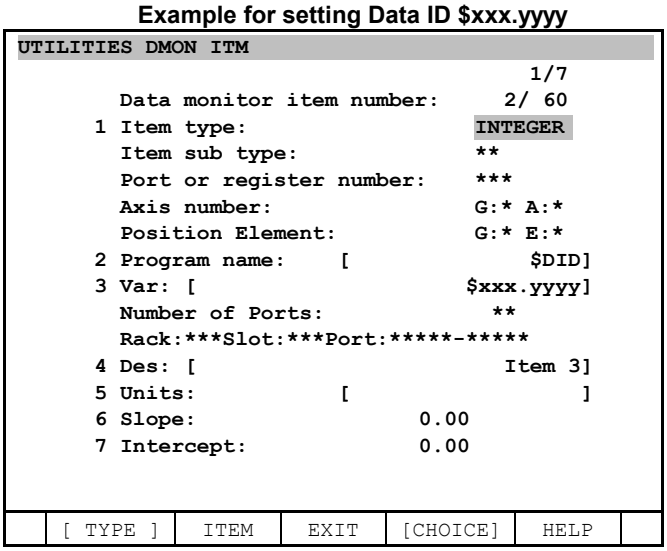


11 To change the Item type, move the cursor to Item type, and press the [ENTER] key.
Case that INTEGER or REAL is selected:
Select Integer or Real depending on data type of Data ID or KAREL variable.

Specify “2 Program name”.
Specify [$DID] for Data ID or [name of KAREL program] for KAREL variable.

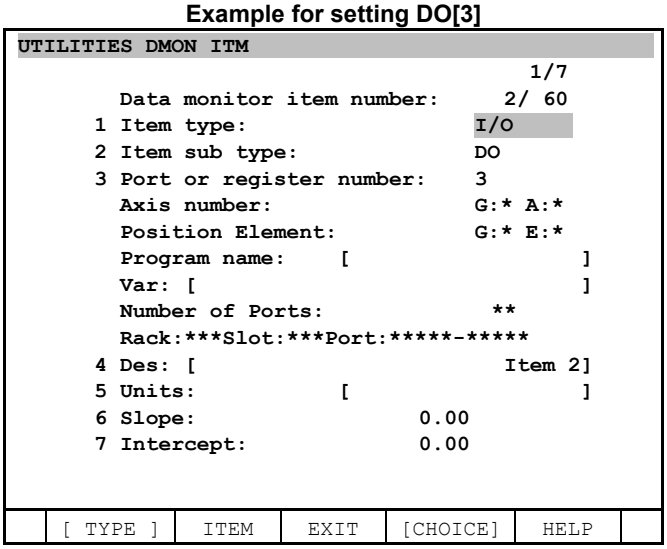
Specify “3 Var”.
Specify variable name in “3 Var”.

Example for setting Data ID $xxx.yyyy



Case that I/O is selected:
Specify “2 Item sub type” and “3 Port or register number”.

Example for setting DO[3]



Case that REGISTER is selected:
Specify “2 Port or register number”.

Example for setting register[5]

```
UTILITIES DMON ITM
                                                  1/6
        Data monitor item number:      2/ 60
      1 Item type:                     REGISTER
        Item sub type:                 **
      2 Port or register number:       5
        Axis number:                   G:* A:*
        Position Element:              G:* E:*
        Program name:     [                      ]
        Var: [                                   ]
        Number of Ports:                 **
        Rack:***Slot:***Port:*****-*****
      3 Des: [                           Item 2]
      4 Units:            [                    ]
      5 Slope:                    0.00
      6 Intercept:                0.00

 |  | [ TYPE ] | ITEM | EXIT | [CHOICE] | HELP |  |
```

Case that JOINT POS is selected:

Specify “2 Axis number”.

Specify group number of robot in “G:” and axis number in “A: “.

Example for setting J1 axis of group 1 robot

```
UTILITIES DMON ITM
                                                  1/6
        Data monitor item number:      2/ 60
      1 Item type:                     JOINT POS
        Item sub type:                 **
        Port or register number:       ***
      2 Axis number:                   G:1 A:1
        Position Element:              G:* E:*
        Program name:     [                      ]
        Var: [                                   ]
        Number of Ports:                 **
        Rack:***Slot:***Port:*****-*****
      3 Des: [                           Item 2]
      4 Units:            [                    ]
      5 Slope:                    0.00
      6 Intercept:                0.00

 |  | [ TYPE ] | ITEM | EXIT | [CHOICE] | HELP |  |
```

Case that TORQUE COMM is selected:

Specify “2 Axis number”.

Specify group number of robot in “G:” and axis number in “A: “.

Example for setting J1 axis of group 1 robot

```
UTILITIES DMON ITM
                                                1/6
        Data monitor item number:      2/ 60
      1 Item type:                  TORQUE COMM
        Item sub type:                 **
        Port or register number:       ***
      2 Axis number:                   G:1 A:1
        Position Element:              G:* E:*
        Program name:     [                    ]
        Var: [                                 ]
        Number of Ports:                 **
        Rack:***Slot:***Port:*****-*****
      3 Des: [                           Item 2]
      4 Units:            [                    ]
      5 Slope:                    0.00
      6 Intercept:                0.00

 |  [ TYPE ] |  ITEM |  EXIT |  [CHOICE] |  HELP |  |
```

Case that Q-CURRENT is selected:

Specify “2 Axis number”.

Specify group number of robot in “G:” and axis number in “A: “.

Example for setting J1 axis of group 1 robot

```
UTILITIES DMON ITM
                                                1/6
        Data monitor item number:      2/ 60
      1 Item type:                  Q-CURRENT
        Item sub type:                 **
        Port or register number:       ***
      2 Axis number:                   G:1 A:1
        Position Element:              G:* E:*
        Program name:     [                    ]
        Var: [                                 ]
        Number of Ports:                 **
        Rack:***Slot:***Port:*****-*****
      3 Des: [                           Item 2]
      4 Units:            [                    ]
      5 Slope:                    0.00
      6 Intercept:                0.00

 |  [ TYPE ] |  ITEM |  EXIT |  [CHOICE] |  HELP |  |
```

Case that TC_ONLINE is selected:

No need to specify anything for TC_ONLINE.

Example for TC_ONLINE

```
UTILITIES DMON ITM
                                          1/6
      Data monitor item number:      2/ 60
    1 Item type:                   TC_ONLINE
      Item sub type:                **
      Port or register number:      ***
    2 Axis number:                  G:* A:*
      Position Element:             G:* E:*
      Program name:     [                    ]
      Var: [                                 ]
      Number of Ports:                **
      Rack:***Slot:***Port:*****-*****
    3 Des: [                          Item 2]
    4 Units:            [                    ]
    5 Slope:                    0.00
    6 Intercept:                0.00

[ TYPE ] | ITEM | EXIT | [CHOICE] | HELP
```

Case that BundleI/O is selected:

Specify “2 Item sub type” and “3 Port or register number” and “4 Number of Ports”.
Please set a logical port, which corresponds to start port for sequence of physical port you want to monitor, to “2 Item sub type” and “3 Port or register number”. However only DI/DO, RI/RO, F, M can be set as the logical port. Please set 1-16 as port number for sequence of physical port you want to monitor to “4 Number of Ports”.

Example for sequence of 8 physical ports(Rack0,Slot 1,Ports21-28) whose start port is DI[1]

```
UTILITIES DMON ITM
                                          1/8
      Data monitor item number:      2/ 60
    1 Item type:                   BundleI/O
    2 Item sub type:                 DI
    3 Port or register number:       1
      Axis number:                   G:* A:*
      Position Element:              G:* E:*
      Program name:     [                    ]
      Var: [                                 ]
      Number of Ports:                   8
      Rack:  0Slot:  1Port:   21-  28
    4 Des: [                          Item 2]
    5 Units:            [                    ]
    6 Slope:                    0.00
    7 Intercept:                0.00

[ TYPE ] | ITEM | EXIT | [CHOICE] | HELP
```

Case that SquareCurrent is selected:

Specify “2 Axis number”.
Specify group number of robot in “G:” and axis number in “A: “.

Example for setting J1 axis of group 1 robot

```
UTILITIES DMON ITM
                                                1/6
        Data monitor item number:      2/ 60
      1 Item type:                SquareCurrent
        Item sub type:               **
        Port or register number:     ***
      2 Axis number:                 G:1 A:1
        Position Element:            G:* E:*
        Program name:     [                    ]
        Var: [                                 ]
        Number of Ports:              **
        Rack:***Slot:***Port:*****-*****
      3 Des: [                          Item 2]
      4 Units:            [                    ]
      5 Slope:                  0.00
      6 Intercept:              0.00

|  | [ TYPE ] | ITEM | EXIT | [CHOICE] | HELP |  |
```

Case that CART CAL POS is selected:

Specify “2 Axis number”.

Specify group number of robot in “G:” and position element in “E: “.

Example for X coordinate of group 1 robot

```
UTILITIES DMON ITM
                                                1/6
        Data monitor item number:      2/ 60
      1 Item type:                  CART CAL POS
        Item sub type:               **
        Port or register number:     ***
        Axis number:                 G:* A:*
      2 Position Element:            G:1 E:X
        Program name:     [                    ]
        Var: [                                 ]
        Number of Ports:              **
        Rack:***Slot:***Port:*****-*****
      3 Des: [                          Item 2]
      4 Units:            [                    ]
      5 Slope:                  0.00
      6 Intercept:              0.00

|  | [ TYPE ] | ITEM | EXIT | [CHOICE] | HELP |  |
```

Case that JOINT COMM POS is selected:

Specify “2 Axis number”.

Specify group number of robot in “G:” and axis number in “A: “.

Example for setting J1 axis of group 1 robot

```
UTILITIES DMON ITM
                                                1/6
        Data monitor item number:      2/ 60
      1 Item type:                    JOINT COMM POS
        Item sub type:                **
        Port or register number:      ***
      2 Axis number:                  G:1 A:1
        Position Element:             G:* E:*
        Program name:     [                    ]
        Var: [                                 ]
        Number of Ports:                **
        Rack:***Slot:***Port:*****-*****
      3 Des: [                          Item 2]
      4 Units:            [                    ]
      5 Slope:                    0.00
      6 Intercept:                0.00

 | [ TYPE ] | ITEM | EXIT | [CHOICE] | HELP |
```

Case that CART COMM POS is selected:

Specify “2 Axis number”.

Specify group number of robot in “G:” and position element in “E: “.

Example for X coordinate of group 1 robot

```
UTILITIES DMON ITM
                                                1/6
        Data monitor item number:      2/ 60
      1 Item type:                 CART COMM POS
        Item sub type:                **
        Port or register number:      ***
        Axis number:                  G:* A:*
      2 Position Element:             G:1 E:X
        Program name:     [                    ]
        Var: [                                 ]
        Number of Ports:                **
        Rack:***Slot:***Port:*****-*****
      3 Des: [                          Item 2]
      4 Units:            [                    ]
      5 Slope:                    0.00
      6 Intercept:                0.00

 | [ TYPE ] | ITEM | EXIT | [CHOICE] | HELP |
```

12 When you are done editing this item you can press the F2, ITEM to select a different item by number, or press the F3, EXIT to return to the setup screen.

21.2 DATA MONITOR SCHEDULE

You choose the items you want to monitor in a Data Monitor schedule.
You can also specify:
- Report file naming details
- Sampling frequencies
- WARN and STOP limits
- Start and Stop triggers

Reports

Reports are created automatically when you set the Reporting item on the Data Monitor schedule screen to ENABLED. You can specify a file name and a file device on the Data Monitor Schedule screen for the Data Monitor report.

Table 21.2 Data monitor schedule menu items

ITEM	DESCRIPTION
Schedule [ ]	You can add a comment to each Data Monitor schedule.
File device	This item allows you to specify the name of the device to be used when writing a report. You can choose from FLPY:, PRN:, FR:, MC:, CONS:, RD:, UD1: or UT1:.
File name Default: Blank	This item allows you to specify the name of the file to be used for a report. A .DT file extension is always used. If you leave this item blank, and Reporting is enabled, the saved data file will be named "SAMPL".
File name index Default: 0 Min: 0 Max: 999	This item allows you to specify an index number to be appended to the file name when a report is generated. If this item is non zero, each time a report is generated this index will be incremented. For example if the file name is SAMPL then successive reports will be named SAMPL001.DT, SAMPL002.DT, and so forth.
File size Default: 0 Min:0 Max:99999	This item specifies the amount of memory in KB you expect to use on the file device. During execution of Sample Start[n], the device is checked for this amount of free memory. If it is not available, an error is posted. If you specify 0 as the file size, the system only checks that there is at least one available block on the media.
Sampling	This item specifies the sampling frequency. -Request - This is the sampling frequency you specify. -Actual - This is the actual sampling frequency that Data Monitor will use. Since there are only certain frequencies available, it may be greater than or less than the sampling frequency you specify.
Monitoring	This item specifies the monitoring frequency. -Request - This is the monitoring frequency you specify. -Actual - This is the actual monitoring frequency that Data Monitor will use. Since there are only certain frequencies available, it may be greater than or less than the monitoring frequency you specify.
Recording	This item specifies the recording frequency. -Request - This is the recording frequency you specify. -Actual - This is the actual recording frequency that Data Monitor will use. Since there are only certain frequencies available, it may be greater than or less than the recording frequency you specify.
Record mode Default: CONTINUOUS	This item specifies the recording mode. -ONE BUFFER - Data will be recorded until the record data buffer is full. -CONTINUOUS - The record buffer is re-used when full.

ITEM	DESCRIPTION
Number of items Default: 1 Min: 1 Max: 18	This item specifies how many items are monitored by this particular schedule.
Start item	This item specifies the condition for starting data recording and enable/disable of this condition after Sample Start instruction. If this condition is set to disable, data recording is started just after Sample Start instruction. The condition is set as the comparison between the value of the monitoring item and setting item. For example, if condition is [recording is started after current command (item No.=2) becomes more than 150.], please set to [2 > 150]. NOTE: The user can check the item No. for monitoring by data monitor schedule screen.
Stop item	This item specifies the condition for finishing data recording and enable/disable of this condition after Sample Start instruction. If this condition is set to disable, data recording is not finished till performing Sample End instruction. The method of setting condition is the same as Start item.

Procedure 21-2 Setting Up and Editing a Data Monitor Schedule

Condition

You have installed the Data Monitor schedule on your controller.

Step

1 Press the [MENU] key.
2 Select UTILITIES.
3 Press the F1, [TYPE].
4 Select Data Monitor Schedules. The following screen will be displayed.

```
UTILITIES DMON SCH
    Sched: 1/5 [Weld cmd + fbk ]      12/17

      1 Schedule:  1      [Weld cmd + fbk  ]

      2 File device:      [MC:  ]
      3 File name:        [                ]
      4 File name index:            1
      5 File size:                  0     KB

        FREQUENCY        REQUEST    ACTUAL
      6 Sampling:         250.00    250.00 Hz
      7 Monitoring:       125.00    125.00 Hz
      8 Recording:         10.00     10.00 Hz

      9 Record mode:             CONTINUOUS
     10 Number of items to monitor:  5

        ITEM DESCRIPTION              ITEM NUM
     11 Voltage (Command)                    1
     12 Wire Feed (Command)                  1
     13 Current (Feedback)                   1
     14 Voltage (Feedback)                   1
     15 Fast Clock                           1

     16 Start item: 2   >= 22.5       DISABLED
     17 Stop item:  3   >= 200.0      DISABLED

 [ TYPE ] | SCHEDULE | LIMITS | [CHOICE] | HELP
```

5 There are 5 Data Monitor schedules. The top line of the Schedule screen displays the current schedule number and its comment. To select a different schedule, press the F2, SCHEDULE, and enter the number of the schedule you want to modify after the prompt.

6 To enter or modify the schedule comment, move the cursor to menu item 1 and press the [ENTER] key.
7 To specify the File device, move the cursor to line 2 and press the F4, [CHOICE].
8 To specify the data monitor report file name, move the cursor to menu item 3 and press the [ENTER] key.
9 If you want to generate multiple report files with an index number in the file name, specify the starting index number on line 4. If you don't want to create a new indexed file each time this schedule is used, enter 0 on line 4.
10 Move the cursor to line 5, and specify the maximum report size, if desired, or leave it set to 0.
11 You specify the Sampling, Monitoring, and Recording frequencies on lines 6, 7 and 8 of the Data Monitor Schedule screen. There are two values shown for these three items, the Requested Frequency that you specify, and the Actual Frequency that will be used as the Sampling, Monitoring, or Recording Frequency. When you enter the desired frequency in the Requested column, the Actual column will update with the closest available frequency.

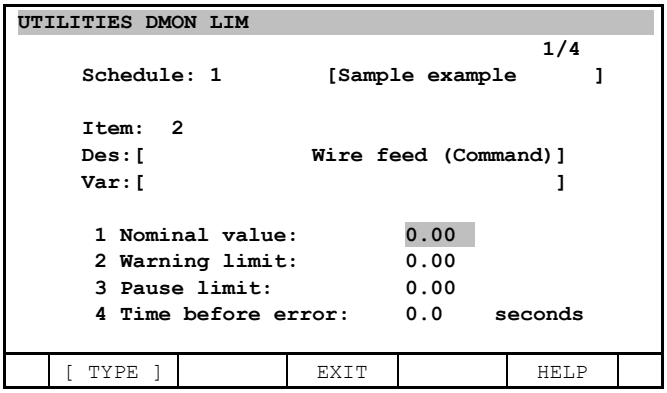
NOTE
The Monitoring and Recording Frequencies must be fractions of the Actual Sampling frequency. If you have an actual sampling frequency of 125 Hz, the maximum Monitoring and Recording frequency can only be 125 Hz. If you modify the Sampling frequency, the Actual frequency may change for all three frequencies.

12 To modify the Record mode, move the cursor to line 9 on the Data Monitor Schedule screen and press the F4, [CHOICE].
13 Each Data Monitor schedule can monitor up to 18 items simultaneously. You can specify the number of items to monitor on line 10 of the Data Monitor Schedule screen.
14 Specify the items you want to monitor on lines 11 through 15 of the Data Monitor Schedule screen. Press the F4, [CHOICE], and select the item you want to monitor from the list of items displayed. You can also specify an item by number in the Item Num column.

NOTE
When you select Current (Feedback) or Voltage (Feedback), the values may not become 0 except during welding by the specification of weld EQ models.

Specify WARN and STOP Limits for Process Signals

15 To specify WARN and STOP limits, move the cursor to each item you have specified on lines 11 to 15 and press the F3, LIMITS function key.
16 Move the cursor to one of monitored items.
17 Press the F3, LIMITS. You will see a that contains details on how specifically to monitor the item you selected.



18 Select each item and set it as desired.
19 When you finished setting items, press the F3, EXIT, to display the previous screen.

Start and Stop Trigger Items

20 You can specify the Start and Stop trigger items and conditions using the fields in menu items 16 and 17 on the Data Monitor Schedule screen.

Setting Up Number of Data Monitor Schedules

Change the value of "Number of schedule" in MENU → CONFIG → Data Monitor function at controlled start to set up the number of the Data Monitor schedules.

21.3 PROGRAMMING

You can use the following teach pendant instructions to start and end data monitoring:
- Sample Start[]
- Sample End

The Start instruction has a schedule number as an input parameter.

NOTE
You cannot start multiple data monitoring sessions at one time. You must end a monitoring session with a Sample End before executing another Sample Start.

Sample Start[1]
- To start data monitoring, include the Sample Start[] instruction in a teach pendant program.

Sample End
- To stop data monitoring, include the Sample End instruction in a teach pendant program.

When "Recording" and "Filing" are enabled,
recorded data file will be output when executing the Sample End instruction.
However, if program is paused before executing the Sample End instruction, a file with data recorded up till that timing will be output. Even if the paused program is resumed, no further data will be recorded.

See Fig. 21.3 for an example of how to use these instructions in a teach pendant program.

```
     1:   Sample Start[1]
     2: J P[1] 40% FINE
      :   Arc Start[1]
     3: L P[2] 20.0cm/min FINE
         Arc End[1]
     4:   Sample End
  [End]
```

Fig. 21.3 Example of using sample start[] and sample end in a teach pendant program

The Sample Start and Sample End instructions are located in the Data Monitor category of the Teach Pendant Editor INST menu.

21.4 DATA MONITOR CHART

Data monitor chart screen enables you to display the graphs of the monitor reports on the teach pendant. The graphs of the monitor reports are displayed after reading from other memory (MC etc.). So it is needed to insert other memory to a robot controller before displaying data monitor chart screen.

NOTE
This screen does not support on Tablet Teach Pendant.

Procedure 21-3 Data monitor chart

Condition

Data monitor function is ordered.
Robot controller have already detect other memory which have monitor reports.

Step

1 Press the [MENU] key.
2 Select “1 Utility”.
3 Press the F1, [TYPE].
4 Select “Data Mon Chart”. The following screen will be displayed.

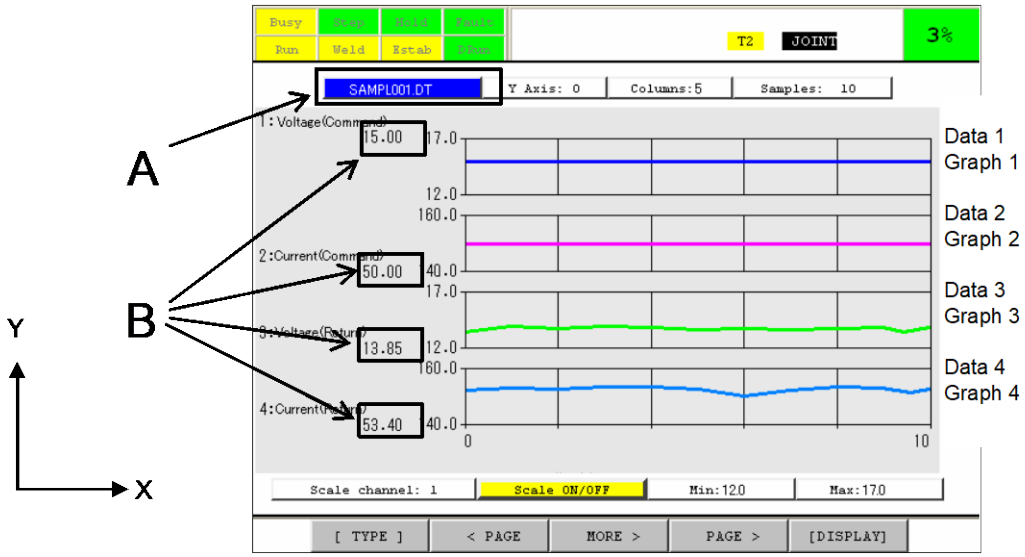


Fig. 21.4 Data monitor chart screen

5 Data monitor chart screen can have up to 12 graphs (8 graphs in the case of software version 7DC3 or older). Y direction of a graph is reported data and X direction is the reported count.
6 In FILE screen, change selected device to the device which is saved monitor report files, and set the current directory to the directory which is saved monitor report files.
7 The file name of the monitor report is displayed on Fig 21.4 A. When the cursor is moved to A and the [ENTER] key is pressed, file select screen will be displayed and the user can select the “.DT” file in other memory.
8 In Fig 21.4 B, newer value of the Y direction of each graph is displayed.
9 “Y Axis:” can compare displayed graphs. If 1 is set to “Y Axis:”, Data 2-4 are displayed on Graph 1. If 2 is set to “Y Axis:”, Data 1,3,4 are displayed on Graph 2.
10 “Columns:” can specify the number of the scale of X direction. “Columns:” can be set from 1 to 20.

11 “Samples:” can specify the number of displayed data. “Samples:” can be set from 1 to 300 (from 1 to 500 in the case of software version 7DC3 or older). If the user want to the part of the data, please set small number to this item.
12 “Scale channel:”, “Scale ON/OFF”, “Min:” and “Max:” can change the scale of Y direction in each graph. First, enter the graph number which you want to change scale to “Scale channel:”. If 1 is set to “Scale channel:”, the user can change the scale of graph 1. Next, set the minimum and maximum value of Y axis to “Min:” and “Max:”. After that, move the cursor to “Scale ON/OFF” and press the [ENTER] key. Y axis scale of the graph which is specified by “Scale channel:” is changed.
13 F2, < PAGE and F4, PAGE > key can change the scale of X direction of each graph.
14 F4, PAGE > key can move the column of X axis by +1.
15 F5, [DISPLAY] key can switch display enable/disable of each graph.

21.5 DATA MONITOR OPERATION

Data monitor operation screen enables you to start / stop the monitoring from the screen. When starting monitoring from this screen, monitoring will continue even if program is paused. Press the STOP button on this screen to output the recorded data file. When starting and stopping the monitoring from this screen, the “DMER-108 Data monitor started” and “DMER-109 Data monitor stopped” warnings will be displayed respectively.

NOTE
This screen does not support on Tablet Teach Pendant.

Procedure 21-4 Data monitor operation

Condition

Data monitor function is ordered.

Step

1 Press the [MENU] key.
2 Select “1 Utility”.
3 Press the F1, [TYPE].
4 Select “Data Mon Operate”. The following screen will be displayed.

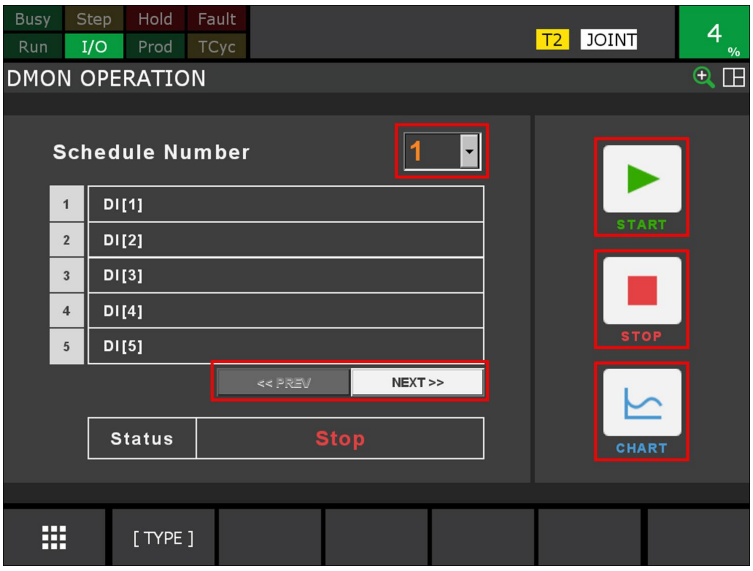


Fig. 21.5 Data monitor operation screen

5 The red boxes show operable areas.

6 “Schedule Number” can specify the schedule number for monitoring. The schedule number for monitoring can be set from 1 to 5.
7 The items that will be recorded are displayed under the “Schedule Number”. ”NEXT >>” button or “<< PREV” button can display the items that will be recorded. Data monitor operation screen can record up to 10 items.
8 “START” button can start the monitoring.
9 “STOP” button can stop the monitoring. Please note that monitoring will continue until the “STOP” button is pressed.
10 “CHART” button can take you to data monitor chart screen.
11 “Status” shows current monitoring status (“Monitoring...” or “Stop”).

22 BRAKE CHECK

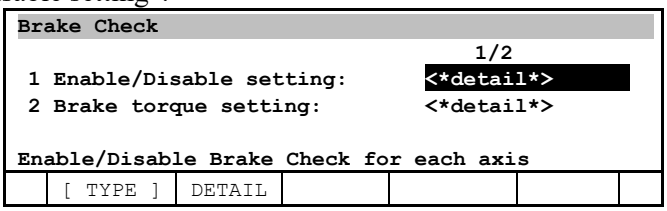
Brake Check (A05B-2700-J951) is a preventive maintenance function. This function commands the motor to generate a certain torque while the brake keeps engaged to check if the motor does not move. This allows you to perform early detection of potential brake failure. This function informs you of the checking result by warning message or Brake Check status screen.

22.1 INITIAL SETTING BEFORE USE

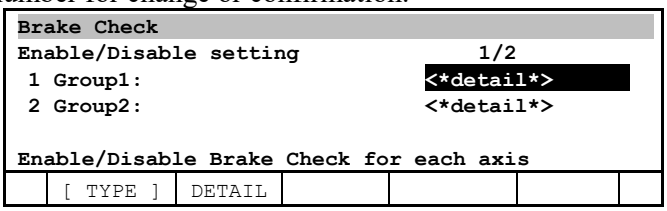
Enable/Disable switch for each axis

On this function, you can customize enable/disable brake check for each axis. On robot axes, the default setting is “Enable”. On an axis other than robot axes (aux axis, servo gun, or etc.), the default setting is “Disable”. You can change or confirm the setting with the following procedure.

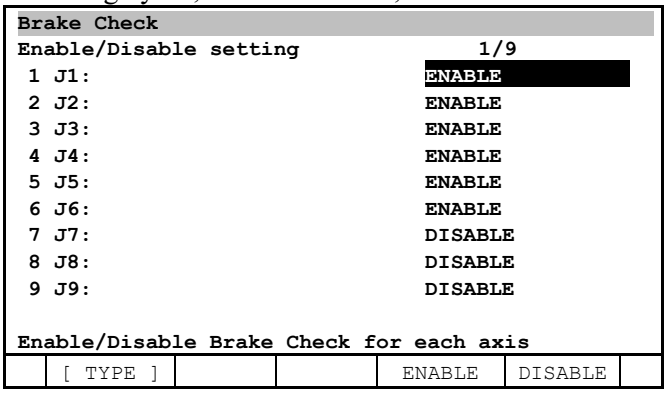
1. Open the Brake Check setup screen.
   (a) Press the [MENU] key.
   (b) Select “SETUP”
   (c) Press the F1, [TYPE]
   (d) Select “Brake Check”
2. Select “Enable/Disable setting”.



3. Select the group number for change or confirmation.



4. The enable/disable settings for each axis is shown.
   You can change the setting by F4, ENABLE or F5, DISABLE.



You can also change or confirm the setting via Data ID.

(Data ID: $BRAKE_CHECK.CHECK_ENABLE[a,b] / BOOLEAN / RW / a:group number b:axis number / TRUE: Enable, FALSE: Disable)

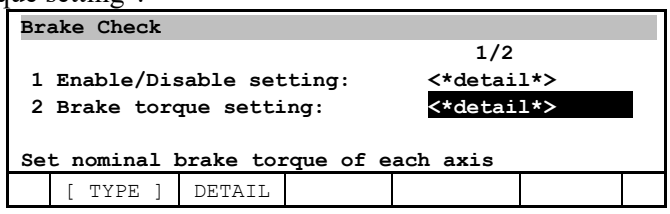
If you execute brake check on dual drive axis, please set the “Enable/Disable setting” to ENABLE for both master and slave axis.

Setting brake torque

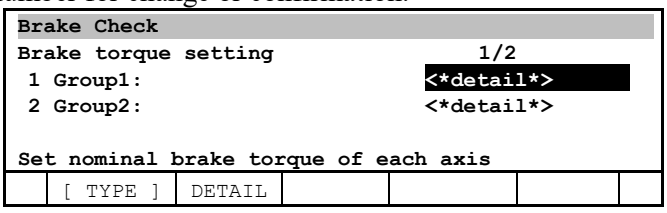
This function needs the specification value of brake torque for each axis in order to determine the torque that is to be commanded during brake check.
On robot axes, the brake torques have been set automatically, however, on an axis other than robot axes (aux axis, servo gun, or etc.), brake torque is not set automatically, so please manually set the brake torque if you need brake check on such an axis. For the specification value of brake torque, please refer to “BRAKE SPECIFICATIONS” on the corresponding AC SERVO MOTOR DESCRIPTIONS. Please note that the unit is [kgf*cm].

You can change or confirm the brake torque [kgf*cm] for each axis according to the following procedure.

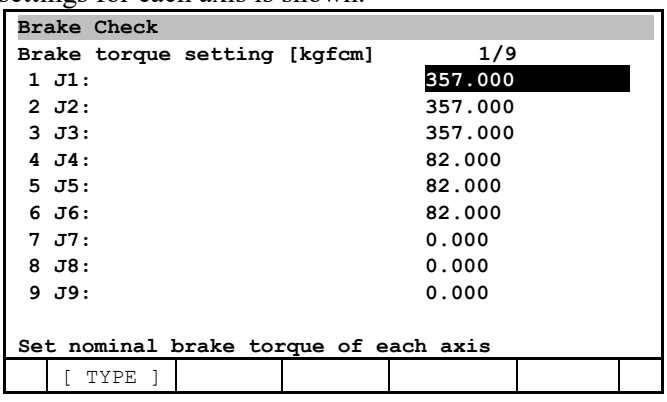
1. Open the Brake Check setup screen.
   (a) Press the [MENU] key.
   (b) Select “SETUP”
   (c) Press the F1, [TYPE]
   (d) Select “Brake Check”
2. Select “Brake torque setting”.



3. Select the group number for change or confirmation.



4. The brake torque settings for each axis is shown.



You can also change or confirm the setting via Data ID.
(Data ID: $BRAKE_CHECK.NOMINAL_BRAKE_TORQUE[a,b] / REAL / RW / a:group number b:axis number)

If you execute brake check on dual drive axes (except robot axes), please set the brake torques as follows. (For robot axes, you don't need to set them even if they are dual drive axes.)

Case-1: Either master or slave motor has one brake.
Please set the brake torque [kgf*cm] only for the axis that has a brake. For the axis that has no brake, please set 0.0 (default).

Case-2: Both master and slave motors have brake respectively.
Please set the brake torque [kgf*cm] for both master and slave axes. And then the brake torque value of master and slave axis must be same. This function does not support a configuration where these values are different.

22.2 START BRAKE CHECK

To start brake check, please call the KAREL program BC_START from TP program.
If BC_START is not displayed in program list menu, press the [MENU] key → System → Config, then select "Program settings" → "KAREL Program settings", then set "KAREL Prog in select menu" to TRUE.
BC_START automatically finishes when brake check is completed. Please not to move robot during the brake check execution.

Below is a sample of TP program that executes brake check.

```
1: J P[1] 100% FINE
2: L P[2] 2000mm/sec FINE
3:  WAIT 1.00 (sec)
4:  CALL BC_START
5: L P[3] 2000mm/sec FINE
```

You cannot start brake check while the robot is moving. Therefore, as you can see in Line 3 of the sample program above, add a WAIT instruction so the program waits until robot stops completely. This program calls BC_START at Line 4 to start brake check. BC_START ends when brake check completes, then the next motion starts at Line 5. As just described in this sample program, brake check must be executed under the condition the robot stops completely.

> ⚠ **CAUTION**
> If the brake has a failure, the motor can move during brake check. Please be sure to execute brake check in a sufficiently large space. Total motion amount per single check varies depending on robot models, however, it is 200000[pulse] at a maximum. This is about 0.4[rev] of motor revolution and about 1.4[deg] of the angle assuming the gear ratio as 1/100.

Additionally, if the power is shut OFF during brake check for some reason, the following alarm will be posted at the next power ON. This alarm indicates an abnormal termination at the last check.

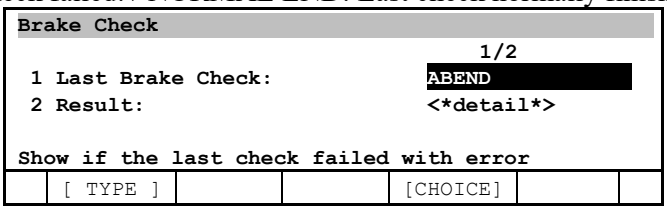
```
BRCH-003  Last check was failed
```

You can also check whether the last check was terminated abnormally or not in the Brake Check status screen with the following procedure.

1. Open the Brake Check status screen.
   (a) Press the [MENU] key.
   (b) Select "STATUS"

(c) Press the F1, [TYPE]
(d) Select “Brake Check”

2. “Last Brake Check” indicates whether the last check was terminated abnormally or not.
ABEND: Last check failed. / NORMAL END: Last check normally finished.



If the brake check fails, the results will not be generated, therefore please execute brake check again at next power ON. If the next brake check normally finishes, “Last Brake Check” will become “NORMAL END”.

22.3 RESULTS OF BRAKE CHECK

When the brake check finishes, the results are posted as warning messages.
If all brakes passes the check, the following message will be posted.

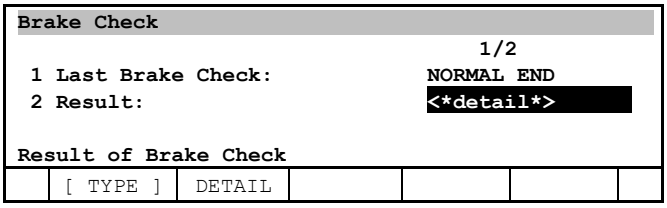
```
BRCH-001  All brake are normal
```

If a brake failure is detected, the following message will be posted. (This example shows that a brake failure was detected on J2 axis of Group 1.)

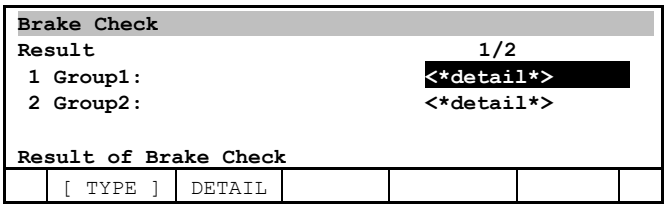
```
BRCH-002  Brake abnormal (G:1,A:2)
```

You can also check the results in the Brake Check status screen with the following procedure. The results are kept until next brake check execution.

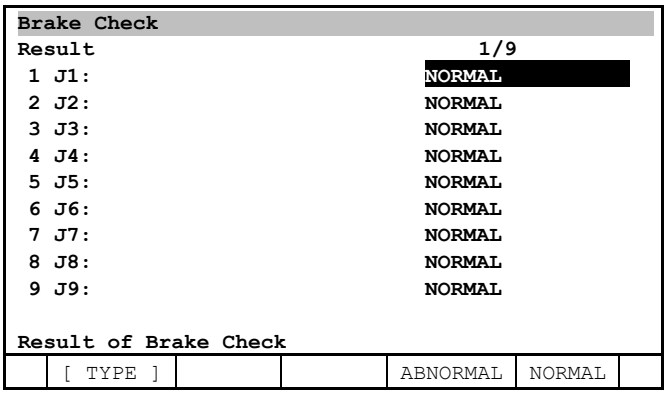
1. Open the Brake Check status screen.
   (a) Press the [MENU] key.
   (b) Select “STATUS”
   (c) Press the F1, [TYPE]
   (d) Select “Brake Check”
2. Select “Result”.



3. Select the group number to be checked.



4. The result of brake check for each axis is shown.
   If a brake failure is detected, “ABNORMAL” is displayed. If a brake is normal, “NORMAL” is displayed. “NORMAL” is also displayed for the axes for which brake check is not executed.



You can also check the results via Data ID.
(Data ID: $BRAKE_CHECK.RESULT[a,b] / BOOLEAN / RO / a:group number b:axis number / TRUE: Abnormal, FALSE: Normal)

If a Dual Drive axis or an additional brake is used, the axis can have two or more brakes. In such a case, you cannot identify which brake has a failure since the total brake torque is tested.

22.4 LIMITATIONS

Brake check cannot be executed under the following conditions. An alarm will be posted.

- Calibration is not done.
- Robot is moving.
- Servo is OFF.
- Servo parameter is under updating.
- Even though the axis has a brake and brake check is enabled, brake torque is not set.
- Teach Pendant is enabled.
- Any axis is near its stroke limit.

22.5 CAUTION

- It takes about 10[sec] for single brake check execution.
- When the motor is heating up, the motor torque can decrease, so it may not be able to detect brake failure. To detect it with high accuracy, execute brake check when the motor temperature is low, e.g. in the morning before production start.

23 NC PROGRAM CONVERSION

23.1 OVERVIEW

The NC Program Conversion function allows you to load a robot program written in the NC programming language (also known as G-code programs) into a robot controller, and convert it to a TP program, thus enabling the robot to execute a user program that is written in the NC programming language.

23.1.1 NC program syntax

As is always the case, an NC program is written specifically for a target machine tool. For example, an NC program for a lathe is different than a program for a machining center, and is therefore incompatible. Likewise, an NC program for a robot has to be written specifically for a robot application. This feature is not a general-purpose program convertor, but can only convert an NC program specifically written for a robot to a robot TP program.
One example is the position representation. The positions must be in the robot mechanical configuration. Typically, the positional representation XYZWPR is used for robot programs, where each character represents the location and orientation of the robot tool center point (TCP).
M-codes (miscellaneous codes) are provided for each application as definitions. Although the scope of the configuration is limited, the NC Program Conversion function is able to convert a set of predefined M-codes into TP program instructions, or CALL a subprogram that does the processing specific to that M-code. A list of supported M-codes is shown later.

23.1.2 Specifications

The specifications and limitations of The NC Program Conversion function are as follows:

1. This function converts an NC program written in ASCII format to a TP program. The file extension for the source program is .NC.
2. The output TP program has the same program name as the original NC program (for example, a TP program will have the name: “O0081”; for an NC program named “%O0081”). The source file name has to be the same as the program name; namely, O0081.NC, and so forth.
3. There will be a line-by-line correspondence between the source NC program and the output TP program, with the following exceptions:
   - Inline comments are not supported in a TP program.
   - The maximum length of a comment is limited by the TP program specifications, and will be truncated if it is too long (up to 32 characters for a comment line, up to 16 characters for a comment on a position, and so forth).
4. The source NC program must comply with a certain NC program syntax rule, or a conversion error will be posted. For example,
   - Commands are case sensitive. Upper case letters need to be used.
   - All commands (except for the K command) start with a letter followed by one or more numerals. Leading zeros can be omitted.
5. Each line of an NC program needs to begin with an N-code, indicating the program line number.
   - The N-code number must increase as the program proceeds. Duplicate number are not allowed
   - The N-code number does not have to be consecutive. Blank lines will be inserted in the output TP program if the number is not consecutive.
   - An N-code indicates a new line. In other words, the instructions of one NC program can spread over several lines (in the NC program) until the next N-code appears.
   - A comment line also needs to start with an N-code, except for one comment right after an O-code, which becomes the program comment.

6. The position data in the NC program must be in XYZWPR (or XYZWPRE for a 7-axis robot) representation, with config data, as used in robot programming. The format of the config string is (N L U T,0,0,0). Note that space between letters and commas is required within the config data. Refer to Section 3.3.2 Position Data in the *OPERATOR'S MANUAL (Basic Function)* (B-84464EN) for an explanation of position representation.
7. The conversion software does not check the validity of the data. Limit or singularity checking should be performed by the off-line programming system when the NC program is created, or a runtime error will be posted. If an invalid numeral is specified in the positional value, an ASCII-to-Binary conversion error will be posted.
8. Part of the positional data can be omitted. The omitted portion of the position data will be copied from the previous motion instruction. The omission of position data is not allowed in the first motion instruction.
9. The motion types of the robot motions can be specified in the source program. Because all G-codes are already defined, special M-codes are used for this purpose (M210 for Linear motion, and M211 for Joint motion). These M-codes are both "Modal."
10. Converting a TP program back to an NC program is not supported.
11. Circular motion is not supported.
12. M-code instructions are implemented as a subprogram "Mnn," with optional numerical parameters specified by the T-code. The subprogram contents are to be provided and to be tailored by the manufacturer of the target robot system, as they are application-dependent. M-codes with multiple parameters support up to three parameters.
13. An M-code can be specified as a standalone instruction, or as part of a motion instruction, in which case the instruction is executed at the timing "TB (time-before) 0.0sec" as implemented in the robot motion sub-system.
14. Only one M-code can be specified within one line of a program.
15. Internally, the NC program is first converted to an ASCII format TP program, and it is then converted to a binary format TP program. The temporary ASCII TP program is created in the internal RAM drive (RD). The maximum size of a program that can be converted is limited to the size of the RAM drive.
16. For the convenience of the operator as a debugging aid, the intermediate TP program (in ASCII format) may be output by adding the following line as the first line of the program.

    (OUTDEV XX)

    where XX stands for the drive letters to represent the output device such as MC or FR.

23.2 OPERATION

23.2.1 Conversion Operation

There are two ways to use the NC Program Conversion function:

- Load the NC program from the FILE menu. You can load an NC program from the FILE menu and convert it to a TP program. If you perform a load operation on an NC program file, it will convert the file to a TP program.

- Copy the NC program to an MD device. Another way to start conversion is to copy the NC program file onto an MD device. The MD device is a virtual device, and it will automatically convert the NC program file to a TP program file.

23.2.1.1 Load NC program in FILE menu

Performing a LOAD operation within the FILE menu will start conversion of the NC program to a TP program. Use the following procedure to load an NC program from the FILE menu.

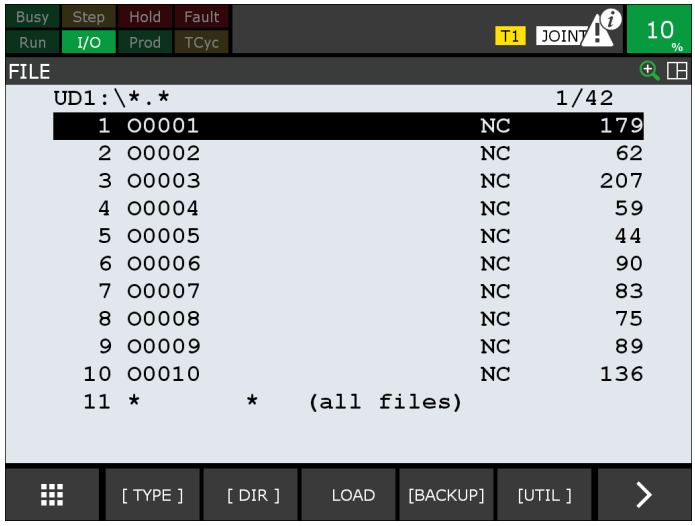
Procedure: Loading an NC Program from the FILE menu

Conditions

An NC program file is listed on the FILE screen.

Step

1 On the FILE screen, display the contents of the device that contains the NC programs.



2 Move the cursor to the file you want to load, using the up or down arrow keys.
3 Press F3, LOAD and then press F4, YES.
4 The system will start loading the selected NC program. After the loading operation is over, you can verify that the converted TP program exists on the program SELECT screen.

23.2.1.2 Copying an NC program to the MD drive

Performing the file COPY operation of an NC program to the MD drive will cause the system to convert the NC program to a TP program. The COPY operation can be performed either on the FILE menu, KCL command, or via the network using an FTP (File Transfer Protocol) operation.
Note that the MD drive is a special drive that automatically performs the conversion operation. The source NC program itself will not be found on the MD drive after the copy operation has completed.

23.3 NC PROGRAM SYNTAX

Since the target machine is a robot, there are several modifications and additions to the standard NC programming language.
Note that although the following sections describe the default syntax, the syntax can be modified.

23.3.1 O code – Program Name/Number

All programs must start with an O-code to specify the program number.

Syntax:

Onnn

Explanation:

1. It must start with the letter O, followed by a number. (up to 9999)
2. This program name (Onnnn) must match the file name.
3. The comment line (other than the N-code) immediately following the O-code becomes the “program comment”.
   The maximum length is 16 characters.

Example:

%O6781
The comment on the first line is a program comment.

23.3.2 N code – Line numbers

Syntax:

Nnnn

Explanation:

An 8-digit sequence number can be specified at the beginning of a block in the program.

1. The line number is specified after the letter “N”.
2. The N-code must be the first instruction in each program line.
3. The numbers become the line numbers in the output TP program.
4. A number can be skipped. However, a blank line will be inserted in the robot TP program for each skipped line. Therefore, it may be a good idea to use consecutive line numbers to avoid several blank lines.

Example:

N001
N7975

23.3.3 G code – Motion modifiers

Syntax:

Gnnn

Explanation:

The NC Program Conversion function supports only the following G-codes. G-codes are usually followed by X, Y, Z, and so forth, because they are is usually attached to motion instructions. G-codes are used to modify motion instructions.
The modal of the M code takes priority, so if joint motion has been specified previously with M code, moving with joint motion even if a linear motion is specified with a G code.

Code	Description	Modal	Explanation
G00	Rapid traverse mode	Yes	The machine moves at the fastest possible speed. The machine might not move in a straight line. It is converted to J (Joint) motion type in the output TP program.
G01	Linear Interpolation	No	The machine moves in a straight line. This is the default setting. It is converted to L (Linear) motion type in the output TP program.
G09	Exact Stop Check	No	Same as the FINE termination type on the robot for the current motion only. See also G61.
G20	Positions in inches	Yes	Positional values are in inches in the source program (the output TP program still contains positions in millimeters)

Code	Description	Modal	Explanation
			Note: When the units are changed, all positional values have to be specified in the next motion, or an uninitialized position error will be posted.
G21	Positions in millimeters	Yes	Positional values are in millimeters. Note: When the units are changed, all positional values have to be specified in the next motion, or an uninitialized position error will be posted.
G61	Exact Stop Check (Modal)	Yes	Same as the FINE termination type on the robot. Unlike G09, the effect remains until it is cancelled by G64.
G64	Cancels G61	Yes	This cancels the effect of G61, and sets the following motions to regular "Continuous" termination type. It is converted to CNT100 termination type in the output TP program.
G90	Absolute programming	Yes	This is the default setting. The positional value is absolute. When this setting is changed, all of the positional values have to be specified in the next motion, or an uninitialized position error will be posted.
G91	Incremental programming	Yes	The positional value is in reference to the previous position, for translational moves only. When this setting is changed, all of the positional values have to be specified in the next motion, or an uninitialized position error will be posted.

23.3.4 F code – Feed Rate / Speed

The F-code specifies the speed of the robot.

Syntax:

Fnnn

Explanation:

For Linear motions (M210):

1. If G21 (positions in millimeters) is enabled: The units are in mm/min. In the TP program, the units are converted to cm/min (values are rounded to nearest integer).
2. If G20 (positions in inches) is enabled: The units are in inch/min. In the TP program units are in inch/min (values are not converted).

For Joint motions (M211): The units are in % (percent). In the TP program, the same speed value is used with no conversion.

If the F-code is not specified in the line, the last specified value will be used (modal setting). If it is the first line of a program, the default value is used.

23.3.5 Position data

Syntax:

Xnnn - value of the first component of a position (X value)
Ynnn - value of the second component of a position (Y value)
Znnn - value of the third component of a position (Z value)
Wnnn - value of the fourth component of a position (W value)
Pnnn- value of the fifth component of a position (P value)
Rnnn - value of the sixth component of a position (R value)
Hnnn - value of the J7 axis (if applicable)
Innn - value of the J8 axis (if applicable)
Jnnn - value of the J9 axis (if applicable)

Explanation:

1. Refer to the Positional Information section in the *OPERATOR'S MANUAL (Basic Operation)* B-84464EN for the explanation of an position representation.

2. All positional data must be specified in the first motion of a program, or an "Uninitialized position" error will be posted.
3. From the second and subsequent motions, some position components can be omitted. The previous values will be used for the omitted components.

23.3.6 K code - Config string

This Config information is part of the position data, and is to be specified with the XYZWPR data.

Syntax:

K(string)

Explanation:

1. The config string value is specified as a string enclosed in parentheses. Refer to the *OPERATOR'S MANUAL (Basic Operation)* B-84464EN for the meaning of the config string.
2. If the K code is omitted, the last value that appeared in the program, or the system default value, will be used.
3. The format of config string is, for example, (N L U T,0,0,0). Please note that a space between letters and commas is required.

Example:

K(N L U T,0,0,0)

23.3.7 M code – Miscellaneous codes

M-codes are generally executed by calling a TP sub-program. Because the M-codes can be defined on the machine tool by each user, the TP subprogram must be provided and customized by the machine tool builder or user. Some predefined M-codes have special meaning, and are converted to equivalent TP instructions.

Syntax:

Mnn [Tmm]

Explanation:

The letter M is always followed by numbers, and by ab optional T-code with numbers.

Run-time processing:

1. Some pre-defined M-codes generate equivalent TP instructions.
2. Other M-codes are converted to specific TP subprogram CALL instructions.
   For example, an M-code (for example, "MppTqq", where pp and qq represent numbers) will generate an instruction to call a TP program, "Mpp(qq)" at runtime.
   In this example, the TP program "Mpp" has the M-code number as part of its program name. The T-code number (qq) is passed as the argument for the program.
3. Since M-codes provide special functionality for a specific system, it is the machine tool builder's responsibility to provide the TP program "Mpp" for the M-codes used for a specific system, except for the predefined M-codes.
4. The numbered portions of the T-code are passed to the TP subprogram as arguments. If no T-code is specified, no argument will be passed.

Notes:

You can only specify up to one M-code and up to three T-codes within one line (block) of a program instruction.

23.3.7.1 Predefined M-codes

The following M-codes are predefined, and you cannot change their meaning.

Code	Description	Modal	Explanation
M00	Compulsory stop	No	It is converted to a PAUSE TP instruction.
M01	Optional stop	No	Machine will stop at M01 only if the operator has pushed the optional stop button. (generates “CALL M01” instruction)
M02	End of program	No	It is converted to an ABORT TP instruction.
M30	End of program with return to program top	No	It is converted to an END TP instruction.

23.3.7.2 Special M-codes

There are some M-codes that are specific to the robot application, or require special processing during the program conversion. Note that the M-code number may be different depending on the system configuration, and therefore, these M-code numbers are configurable.

Code	Description	Modal	Explanation
M31	Vision Resync	Yes	Vision Resync command. Besides calling a sub-program, it causes subsequent motions to have a vision “offset” clause attached. (see note below)
M35	Cancel Resync offsets	Yes	Clears vision resync offset. Besides calling a sub-program, it causes the vision “offset” clauses to be removed from the subsequent motions.
M39, M60, M61	Sensor resync	Yes	Sensor Resync command. Besides calling a sub-program, it causes subsequent motions to have a sensor “offset” clause attached. (see note below)
M26	Cancel Sensor offset	Yes	Clears sensor resync offset. Besides calling a sub-program, it causes the sensor “offset” clauses to be removed from the subsequent motions.
M221	User Frame Selection	Yes	Set the user frame number to the one specified by the T-code
M222	User Tool selection	Yes	Set the user tool number to the one specified by the T-code
M210	Linear Robot Moves	Yes	Subsequent motions move with Linear motion type. This instruction can be specified as a standalone instruction or as part of a motion instruction. If standalone, it generates a NOP (comment line).
M211	Joint Robot Moves	Yes	Subsequent motions move with Joint motion type. This instruction can be specified as a standalone instruction or as part of a motion instruction. If standalone, it generates a NOP (comment line).

Note: The “Offset” clause could be either OFFSET, VOFFSET, TOOL_OFFSET, depending on the configuration specified.

Other M-codes can be defined by the Machine Tool Builder or the user, by creating the corresponding TP sub-programs.

23.3.7.3 Predefined M-codes

The default conversions for predefined M-codes are as follows.

M00 (compulsory stop)

The M-code “M00” will be converted to a PAUSE instruction in a TP program.

Example: TP instruction for “M00” M-code

PAUSE

The functionality of a PAUSE instruction is to pause the program at this line. Resuming the program will resume from the next line.

M01 (optional stop)

The M-code “M01” will be converted to a CALL instruction to the predefined sub-program “M01”

Example: TP instruction for “M01” M-code

CALL M01

The equivalent functionality of M01 does not exist on the robot, and therefore, the logic has to be created as the sub-program “M01”. The TP program M01 should be provided for each application by the machine tool builder.

M02 (End of Program)

The M-code “M02” will be converted to an ABORT instruction in a TP program.

Example: TP instruction for “M02” M-code

ABORT

The functionality of ABORT instruction is to abort the program at this line. Once aborted, the program may not be resumed. (You may “start” the program from the current line, however).

M30 (End of program)

The M-code “M30” will be converted to an END instruction in a TP program.

Example: TP instruction for “M30” M-code

END

The functionality of END instruction is to finish the program at this line, and set the cursor back to the beginning (if it is the main program). If the program is executed as a sub-program, the END instruction works as the RETURN instruction to go back to the caller program and continue. Note that the functionality of the END instruction may not be exactly equivalent to the M30 code in the typical NC machines.

M221 (User Frame Selection)

Syntax:

M221Tnn - Selection of User Frame number

Explanation:

It generates a “UFRAME_NUM=nn” instruction.

This is not a standard M-code definition. This is a new M-code specifically for robots, due to the requirement for how positions are specified on robots. The functionality of M221 is as follows:

- For program convertors – This will set the user frame number for all subsequent positions therafter (user frame number is part of the position data)
- For program execution – This will generate a code to set the user frame number at runtime.

Example: TP instruction for “M221” M-code

uframe_num = nn

The frame number should be specified by a T-code that follows.

M222 (User Tool Selection)

Syntax:

M222Tnn - Selection of the User Tool Frame number

Explanation:
It generates a "UTOOL_NUM=nn" instruction.

This is not a standard M-code definition. This is a new M-code specifically for robot, due how positions are specified on robots. The number can be configurable. The functionality of M222 is as follows.
- For program convertor – This will set the user tool number for all subsequent positions (the user tool number is part of the position data)
- For program execution – This will generate a code to set the user tool number at runtime

Example: TP instruction for "M222" M-code
```
utool_num = nn
```

The TOOL number should be specified by a T-code that follows.

23.3.7.4 Other (user-defined) M-codes

M-codes with no associated T-code
Since no T-code accompanies the M-code, the TP subprogram will be called without arguments.

Example: TP instruction for "M25" M-code
CALL M25

Since the functionality of M25 is machine-dependent, the TP program corresponding to M25 must be provided by the machine tool builder. In many cases, the program will communicate with the peripheral equipment via General I/O unit / General I/O board.

Note that the name of the sub-program will be composed of the command letter "M" followed by the command number letters exactly as specified.

M-codes with associated T-codes
The TP sub-program for such an M-code will be called with the arguments specified in the T-code. Up to three arguments can be specified.

Example: TP instruction for "M56T4321" M-code
CALL M56(4321)

The meaning of M-code "M56" is machine dependent, and therefore, the TP program associated with M56 will have to be defined by the machine tool builder. In most cases, the program will communicate with the peripheral equipment via a General I/O unit / General I/O board.

23.4 NC PROGRAM CONVERSION EXAMPLE

Input G-code Program:
```
%O6781
(first line comment becomes the program comment)
N1 (,140A4112-678,2GNCD,BANNON,REV003,1-1,FWC,1G2K,1G2K,CAT)
N2 (MCD,REV003...SYSTEM.034)
N3 (UNIX.POST.GEM5F2,OPTION.5...,REV.16..)
N4M52T3
N5M00(LOAD.RIVET.OFFSET)
(6/32.OFFSET.RIVET.ANVIL.3)
(6/32.RIVET)
```

```
N6M26
N7M211X130.Y123.4Z81.W81.P0.R180.F30.
N8M210X130.Y123.4Z0.W0.P0.R180.F300.
N9M40T0
N10M39
N11M31T2
N12M60
N13X130.Y-33.Z-47.12W-47.12P-70.R350.
N14X130.Y8.Z-35.12W-35.12P0.R270.F300.
N15X95.228Y7.152Z-55.677W-2.538P1.475R270.M80T0355F230.
(ANVIL:265)
N16X95.228Y7.152Z-57.677W-4.538P1.475R270.
N17M35
N18M26
N19M211X130.Y123.4Z81.W81.P0.R180.F30.
N20M56T60
```

Output TP program (in ASCII format):

File Name: O6781.LS

```
/PROG  O6781
/ATTR
OWNER                = NCCONVER;
COMMENT              = "first line comme";
PROG_SIZE            = 1209;
CREATE               = DATE 00-03-13  TIME 17:58:46;
MODIFIED             = DATE 00-03-13  TIME 17:58:46;
FILE_NAME            = ¥O6781.N;
VERSION              = 0;
LINE_COUNT           = 20;
MEMORY_SIZE          = 1573;
PROTECT              = READ;
TCD:   STACK_SIZE            = 0,
       TASK_PRIORITY         = 50,
       TIME_SLICE            = 0,
       BUSY_LAMP_OFF         = 0,
       ABORT_REQUEST         = 0,
       PAUSE_REQUEST         = 0;
DEFAULT_GROUP     = 1,*,*,*,*;
CONTROL_CODE      = 00000000 00000000;
/APPL
/MN
    1:  !,140A4112-678,2GNCD,BANNON,REV00 ;
    2:  !MCD,REV003...SYSTEM.034 ;
    3:  !UNIX.POST.GEM5F2,OPTION.5...,REV ;
    4:  CALL M52(3) ;
    5:  PAUSE ;
    6:  CALL M26      ;
    7:J P[7:M211] 30% CNT100      ;
    8:L P[8:M210] 30cm/min CNT100      ;
    9:  CALL M40(0) ;
   10:  CALL M39      ;
   11:  CALL M31(2) ;
   12:  CALL M60      ;
   13:L P[13] 30cm/min CNT100 Offset Tool_Offset      ;
```

```
   14:L P[14] 30cm/min CNT100 Offset Tool_Offset      ;
   15:L P[15:ANVIL:265] 23cm/min CNT100 TB     0.00sec,CALL M80(355) Offset Tool_Offset      ;
   16:L P[16] 23cm/min CNT100 Offset Tool_Offset      ;
   17:   CALL M35      ;
   18:   CALL M26      ;
   19:J P[19:M211] 30% CNT100      ;
   20:   CALL M56(60) ;
/POS
P[7:"M211"]{
    GP1:
          UF : 0, UT : 1,          CONFIG : 'N U T, 0, 0, 0',
          X =     130.000   mm,      Y =     123.400   mm,      Z =      81.000   mm,
          W =      81.000 deg,       P =       0.000 deg,       R =     180.000 deg
};
P[8:"M210"]{
    GP1:
          UF : 0, UT : 1,          CONFIG : 'N U T, 0, 0, 0',
          X =     130.000   mm,      Y =     123.400   mm,      Z =       0.000   mm,
          W =       0.000 deg,       P =       0.000 deg,       R =     180.000 deg
};
P[13]{
    GP1:
          UF : 0, UT : 1,          CONFIG : 'N U T, 0, 0, 0',
          X =     130.000   mm,      Y =     -33.000   mm,      Z =     -47.120   mm,
          W =     -47.120 deg,       P =     -70.000 deg,       R =     350.000 deg
};
P[14]{
    GP1:
          UF : 0, UT : 1,          CONFIG : 'N U T, 0, 0, 0',
          X =     130.000   mm,      Y =       8.000   mm,      Z =     -35.120   mm,
          W =     -35.120 deg,       P =       0.000 deg,       R =     270.000 deg
};
P[15:"ANVIL:265"]{
    GP1:
          UF : 0, UT : 1,          CONFIG : 'N U T, 0, 0, 0',
          X =      95.228   mm,      Y =       7.152   mm,      Z =     -55.677   mm,
          W =      -2.538 deg,       P =       1.475 deg,       R =     270.000 deg
};
P[16]{
    GP1:
          UF : 0, UT : 1,          CONFIG : 'N U T, 0, 0, 0',
          X =      95.228   mm,      Y =       7.152   mm,      Z =     -57.677   mm,
          W =      -4.538 deg,       P =       1.475 deg,       R =     270.000 deg
};
P[19:"M211"]{
    GP1:
          UF : 0, UT : 1,          CONFIG : 'N U T, 0, 0, 0',
          X =     130.000   mm,      Y =     123.400   mm,      Z =      81.000   mm,
          W =      81.000 deg,       P =       0.000 deg,       R =     180.000 deg
};
/END
```

24 ENHANCED MIRROR IMAGE

The enhanced mirror image option (A05B-2600-R698) allows you to translate an entire teach pendant program or portion of a teach pendant program to mirror-image the original programmed points. This option can be used to teach symmetrical parts easily. The enhanced mirror image option provides additional functionality so that existing frames such as user frames can be used directly with any specified XY, YZ, or XZ plane. Also, additional orientation shifting is provided for different application requirements.

> ⚠ **CAUTION**
> For Paint Tool and paint robots, this is almost always accomplished using left-hand and right-hand robots. Use of this option is not recommended.

Mirror imaging of a program can be accomplished either as:

- mirror image using existing frames such as World frame or user frames. No teaching required.
- mirror image using flexible mirror planes such as X-Y, Y-Z, or X-Z from existing frames for better selection
- parallel and rotational mirror image with mirrored orientation
- mirror image orientation shifting can be selected based on application requirements.

Refer to Table 24 for descriptions of the Mirror Image Shift screen items.

Table 24 ENHANCED MIRROR IMAGE screen items

ITEM	DESCRIPTION
Original Program	This item is the name of the program that you want to shift using mirror image. To select a program, press the F4, [CHOICE], select the name of program, and press the [ENTER] key again.
Range	This item indicates whether the whole program (F5, WHOLE) or a portion (F4, PART) will be shifted using mirror image. ●F5, WHOLE shifts program as a whole. ●F4, PART shifts part of the program.
Start line (used only for “PART” range)	This item is the first line in the program that will be shifted using mirror image. When range is WHOLE, you can't select this item.
End line (used only for “PART” range)	This item is the last line in the program that will be shifted using mirror image. When range is WHOLE, you can't select this item.
New Program	This item is the name of the program that results when you offset or shift the Original Program. If you want the resulting offset or shifted program to insert into the Original Program, make the New Program name the same as the Original Program name.
Insert line	This item indicates the location in an existing program where new shifted position information will be inserted. When creating a new program , you can't select this item.
Position data (used only for “TEACH” method)	This item indicates the teaching position where the cursor is currently located. The position uses the world frames.
Image Shift Method	● F5, TEACH is a method that requires the user to enter position data to define the mirror plane. ● F4, FRAME is a method that allows the user to choose existing frames.
Orientation Shift	● F5, MIRROR is a mirror image shifting method that mirrors the orientation as well as the positions. ● F4, FIXTURE is a mirror image shifting method that only mirrors the approach vector of the orientation but leaves the other two vectors intact with flexible tool orientation control.

ITEM	DESCRIPTION
Mirror Frame (used only for “FRAME” method)	World, User Frame, or Jog Frame can be selected as the mirror frame.
Frame No. (used only for “FRAME” method)	When User frame or Jog frame is selected, this indicates which frame number will be used. When World frame is selected for Mirror Frame, you can't select this item.
Mirror Plane (used only for “FRAME” method)	The XY, YZ, or XZ plane of the selected mirror frame can be used as mirror plane.
Rotation (used only for “TEACH” method)	This item, if set to ON, indicates that you want to rotate three position about an axis. If set to OFF, you can shift a single source position to a single destination position.
Source position (used only for “TEACH” method)	This item is the original position that you will be shifting using mirror image. If Rotation is set to ON, there will be three source positions.
Destination position (used only for “TEACH” method)	This item is the destination position to which you will be shifting using mirror image. If Rotation is set to ON, there will be three destination positions.

24.1 PARALLEL MIRROR IMAGE

A parallel mirror image mirrors the program about a mirror plane without an offset or a rotation. See Fig. 24.1 (a) and Fig. 24.1 (b).

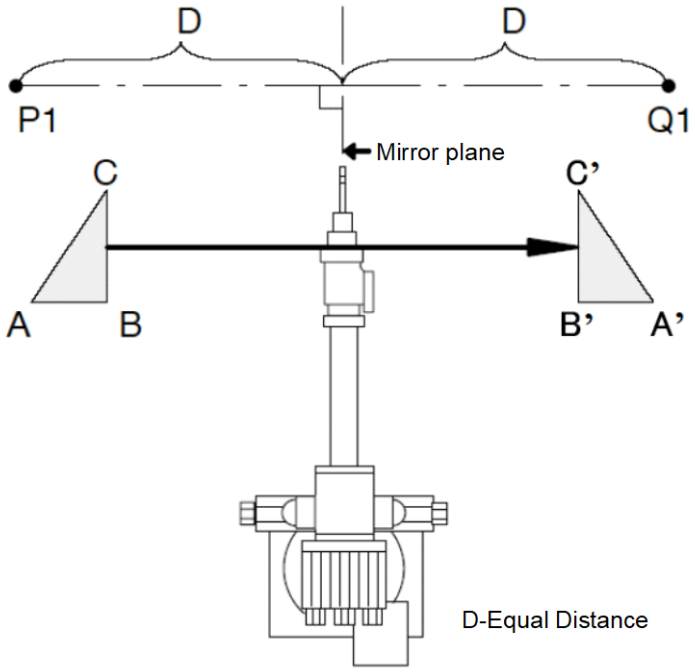


Fig. 24.1 (a) Parallel mirror image with mirror plane in center of robot

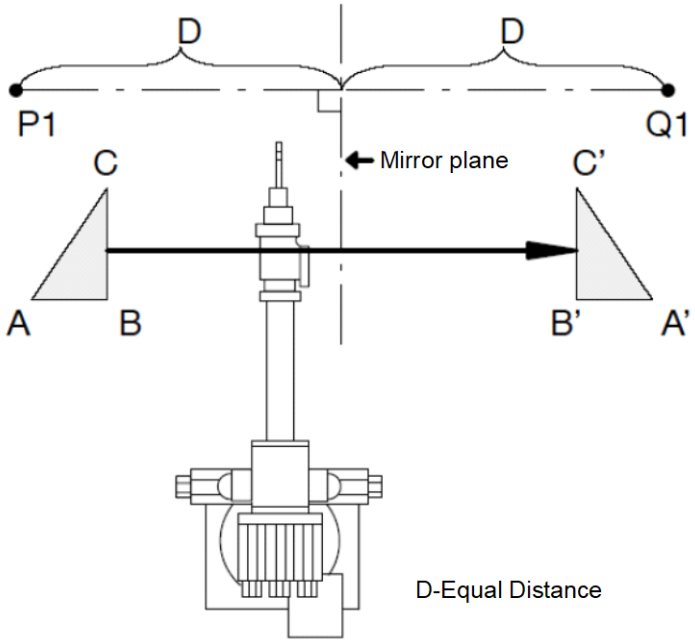


Fig. 24.1 (b) Parallel mirror image with mirror plane offset from center of robot

> ⚠ **CAUTION**
> To be sure the parallel mirror image works correctly, you must have an exact TCP. If you do not, the resulting mirror image program will contain an offset value. See Fig. 24.1 (c).

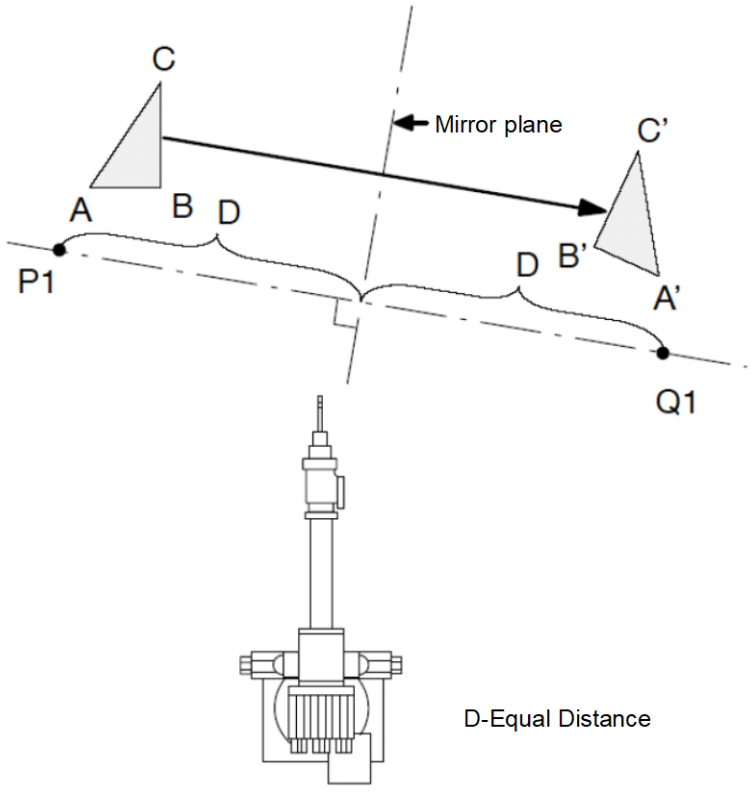


Fig. 24.1 (c) Parallel mirror image with offset

Mirror Image Example

Fig. 24.1 (d) displays a mirror image of A, B, and C to A', B', and C' when P1 and Q1 are taught positions. Fig. 24.1 (d) also displays a mirror image with an offset when P1 and Q1 are taught but Q1 is taught at an offset of 200mm. In this case, the result is A'', B'', and C''.

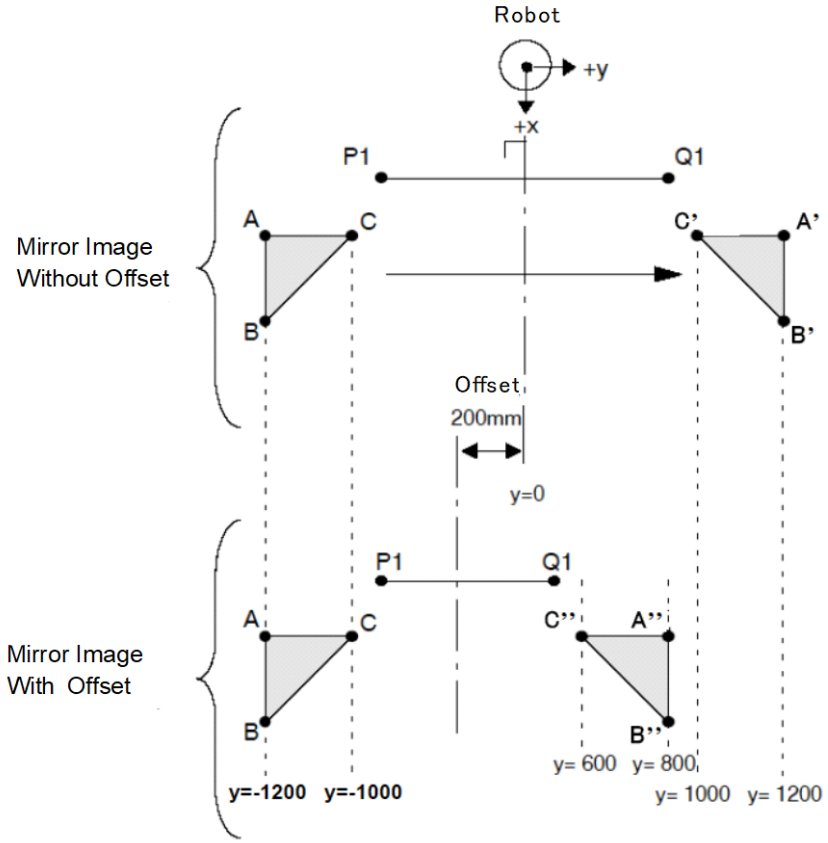


Fig. 24.1 (d) Positional mirror image

24.2 ROTATIONAL MIRROR IMAGE

A parallel mirror image mirrors the program about a mirror plane first, and then the mirrored program is rotated about a center of rotation. Orientation of the part to be imaged is rotated about one or more of its axes relative to the mirror plane. See Fig. 24.2 (a).

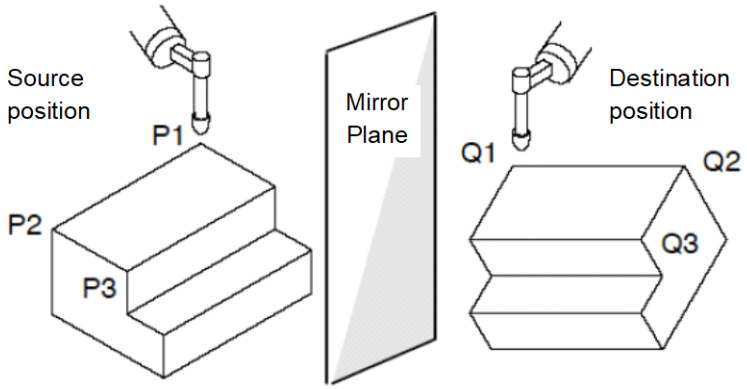


Fig. 24.2 (a) Rational mirror image

In Fig. 24.2 (b) the positions, P1, P2 and P3 are mirrored about the mirror plane as Q1, Q2 and Q3. These positions are then rotated 45° about Q1 and stored as Q1', Q2' and Q3'.

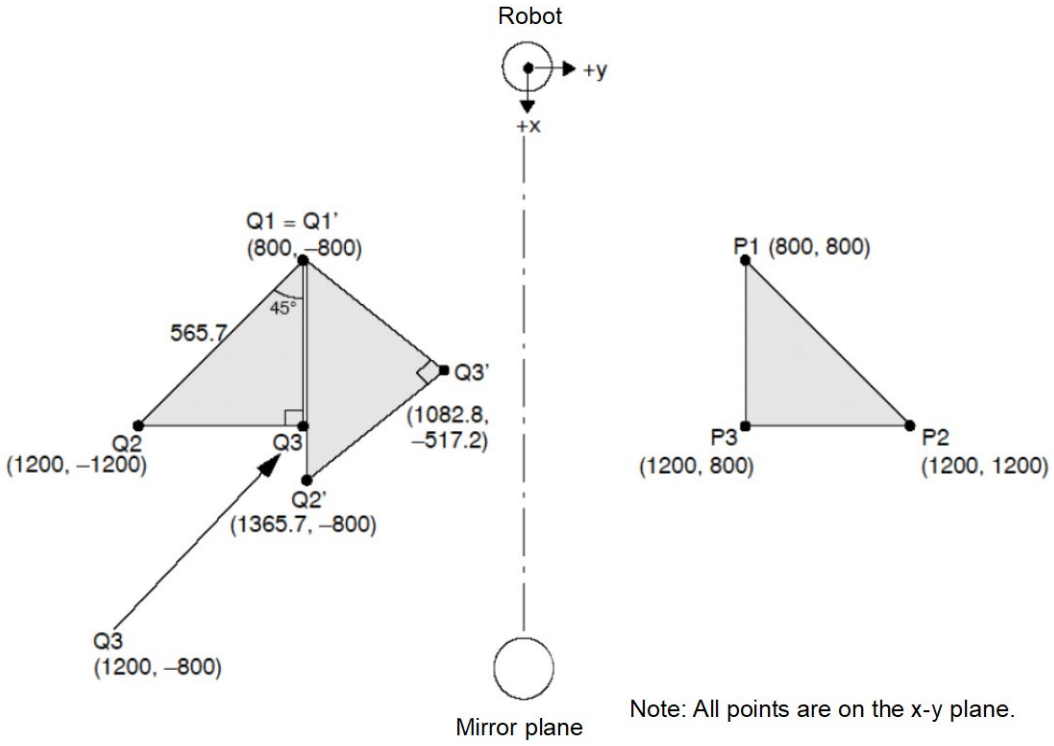


Fig. 24.2 (b) Rational mirror image

24.3 MIRROR IMAGE USING EXISTING FRAMES AND MIRROR PLANES, WITH CONTROLLED ORIENTATION

The standard mirror image utility mirrors orientation as well as positions, as shown in Fig. 24.3 (a).

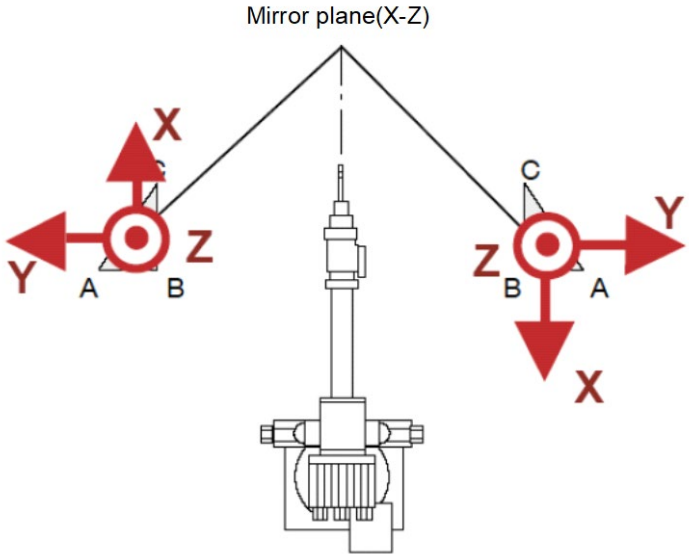


Fig. 24.3 (a) Mirror image shift with orientation mirrored (mirror method)

The enhanced mirror image option also provides a simple way to use existing frames, such as World, User, or Jog Frame, with defined mirror planes, such as X-Y, Y-Z, or X-Z. Also, to control the orientation as well as the positions in the mirrored program, the Orientation Mirror method can be selected as "Mirror" as shown in Fig. 24.3 (a), or "Fixture", as shown in Fig. 24.3 (b).

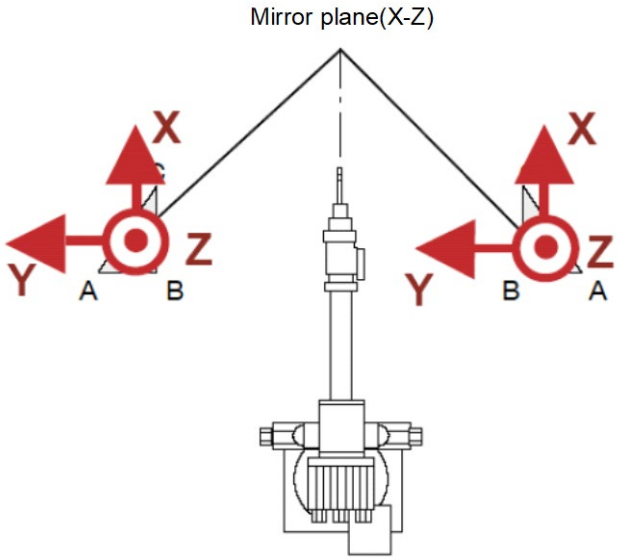


Fig. 24.3 (b) Enhanced mirror image with orientation controlled (fixture method)

24.4 MIRROR IMAGE OF EXTENDED AXES

Extended (Ext) axes determine how the mirror image function will translate the program when you are using extended axes. Refer to Fig. 24.4 (a) through Fig. 24.4 (d).

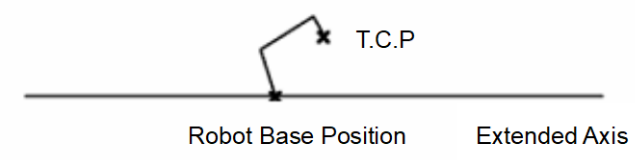


Fig. 24.4 (a) Mirror image of extended axis

Possible extended axes configurations are:

- **Robot axes only -** allows you to mirror the axes of the robot without mirroring any non-integrated extended axes such as a positioning table. The shift is calculated using the change in the robot tool center point (TCP). See Fig. 24.4 (b).

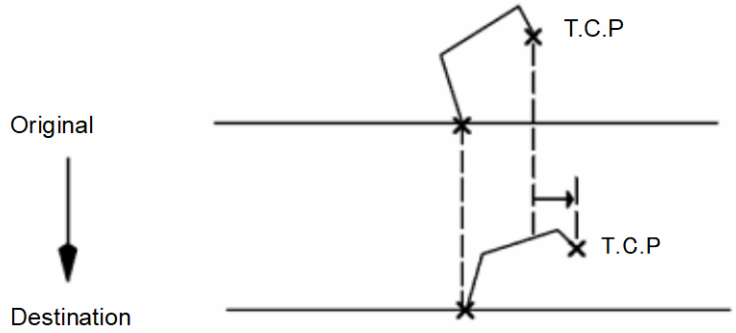


Fig. 24.4 (b) Example of robot axes only mirror image

- **Ext integrated -** allows you to mirror the axes of the robot and any integrated axes. The amount of mirror image for the robot and the extended axes is calculated using the change in the TCP. See Fig. 24.4 (c).

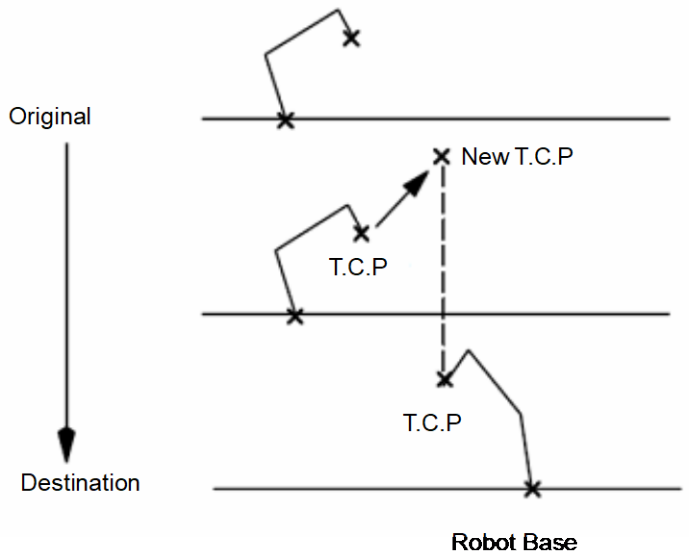


Fig. 24.4 (c) Example of extended axes integrated mirror image

- **With ext axes -** allows you to mirror positions for robot axes and any extended axes in your system. The amount of mirror image for the robot is calculated by using the change in the TCP. The amount of shift for the extended axes is calculated using the center of the difference between an original position (P1) and a new position (Q1) as the point where the mirror image occurs. See Fig. 24.4 (d).

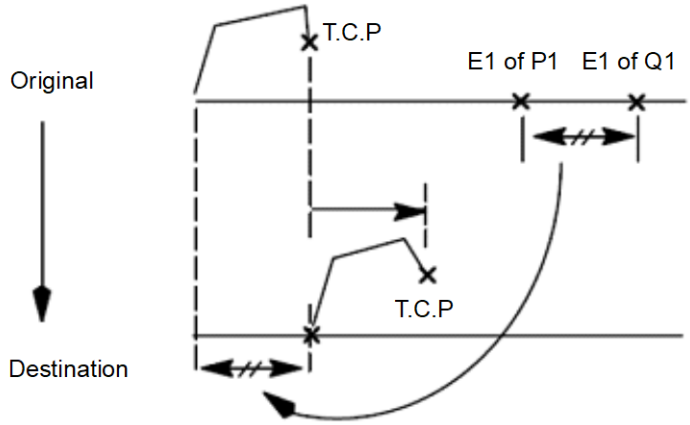


Fig. 24.4 (d) Example of with extended axes mirror image

Procedure 24-1 Using Enhanced Mirror Image Shift

Conditions

- The program you want to mirror has been created and contains recorded positions.
- All robot joint axes are at zero degrees.

Steps

1 Press the [MENU] key. The menu will be displayed.
2 Select UTILITIES.
3 Press the F1, [TYPE].
4 Select Mirror Image Shift. The MIRROR IMAGE SHIFT screen will be displayed.

```
MIRROR IMAGE SHIFT
    Program                                1/9
        Original Program :
      1
      2   Range:                          WHOLE
      3   Start  line: (not used)         *****
      4   End    line: (not used)         *****
      5 New Program : [                        ]
      6   Insert line:                    *****
      7 Image Shift Method:             TEACH
      8 Orientation Shift:              MIRROR
      9 EXT axes:          Robot axes only

[ TYPE ]  |  |  [CHOICE]  |  >
```

5 Move the cursor to 1 under Original Program. If the program you want to mirror is not selected, press F4, [CHOICE]. Select the program from the list and press the [ENTER] key.

NOTE
The last program selected using the SELECT menu will automatically be named as the original program.

6 Move the cursor to Range. Use the function keys to select whether you want to mirror all or part of the original program.

- **To mirror the whole program,** press F5, WHOLE. The following screen will be displayed.

```
MIRROR IMAGE SHIFT
    Program                                1/9
        Original Program :
      1
      2   Range:                          WHOLE
      3   Start  line: (not used)         *****
      4   End    line: (not used)         *****
      5 New Program : [                        ]
      6   Insert line:                    *****
      7 Image Shift Method:             TEACH
      8 Orientation Shift:              MIRROR
      9 EXT axes:          Robot axes only

[ TYPE ]  |  |  [CHOICE]  |  >
```

NOTE
EXT axes will be displayed only if you are using mirror image for extended axes.

- **To shift part of the program,** press F4, PART. You will see a screen similar to the following.

```
MIRROR IMAGE SHIFT
    Program                                1/9
        Original Program :
      1
      2   Range:                          PART
      3   Start  line: (not used)            0
      4   End    line: (not used)            0
      5 New Program : [                        ]
      6   Insert line:                    *****
      7 Image Shift Method:             TEACH
      8 Orientation Shift:              MIRROR
      9 EXT axes:          Robot axes only

[ TYPE ]  |  |  [CHOICE]  |  >
```

7 **If you selected to mirror PART of a program,** select the lines of the program that you want to mirror:

- Move the cursor to Start line and type the starting line number. Press the [ENTER] key.
- Move the cursor to End line and type the ending line number. Press the [ENTER] key.

8 Move the cursor to New Program and press the [ENTER] key. Use the appropriate function keys to type the name of the new program and press the [ENTER] key. This is the program to which you will be mirroring the positions.

NOTE
You can mirror the positions of an entire program or portion of a program from within a program or from one program to another. See Fig. 24.4 (e) and Fig. 24.4 (f).

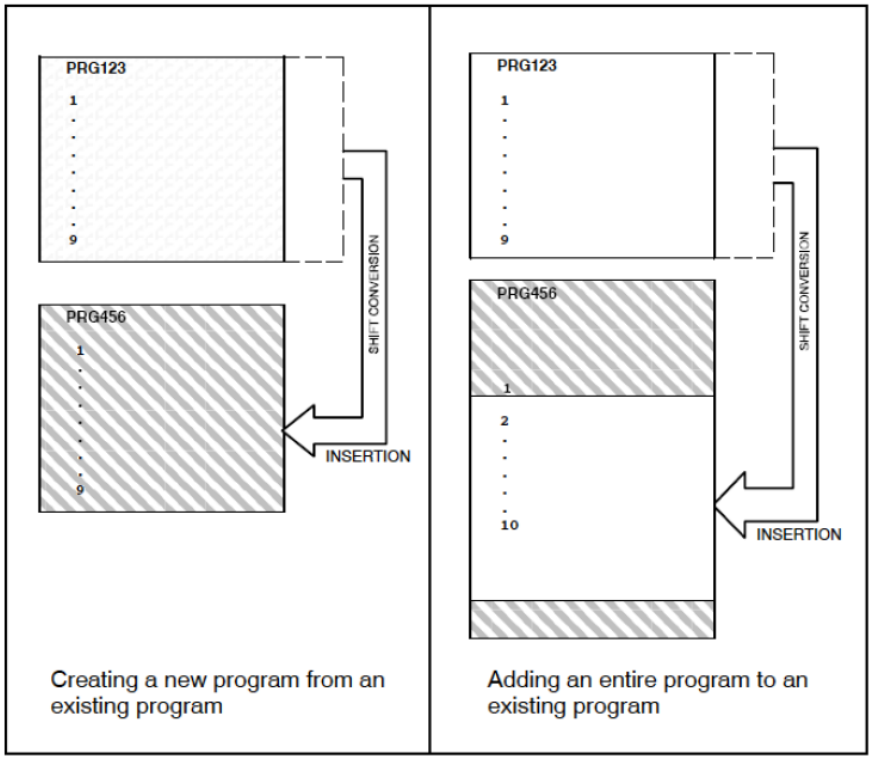


Fig. 24.4 (e) Mirroring an entire program

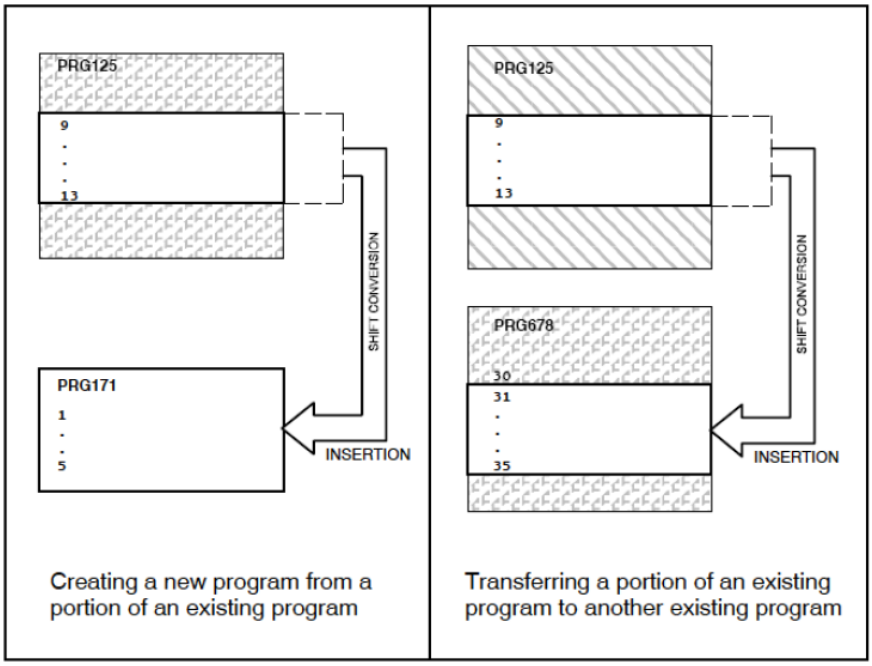


Fig. 24.4 (f) Mirroring a portion of a program

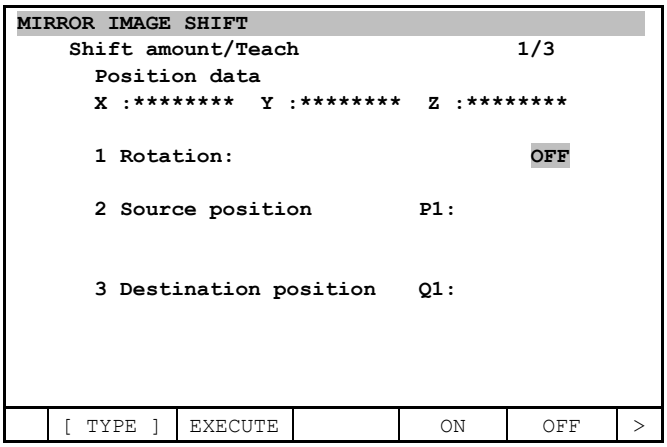
NOTE
The New Program can be the same as the original program, an already existing program, or a program that does not exist.

- **If you are inserting lines into the original program,** type in the name of the program and the corresponding line numbers.
- **If you are inserting lines into an existing program,** move the cursor to Insert line and type the line number at which you want to insert the shifted information.

9 Move the cursor to Image Shift Method.
- **To mirror the image using existing User, Jog, or World frames** and well-defined mirror frames such as Z-Y, Y-Z or X-Z, press F4, FRAME.
- **To mirror the image by teaching points to define the mirror plane,** press F5, TEACH. This method also mirrors the orientation.

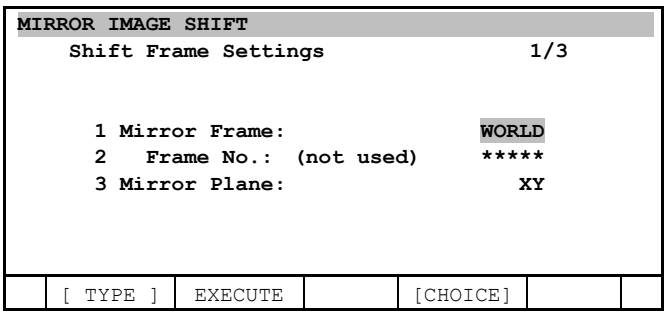
10 Move the cursor to Orientation Shift.
- **To control the orientation shift** in the mirrored program, press F4, FIXTURE.
- **To mirror the orientation** of the original program, press F5, MIRROR.

11 Press the down arrow key.
- **If you selected TEACH** for the Image Shift Method, the following Shift amount/Teach screen will be displayed:



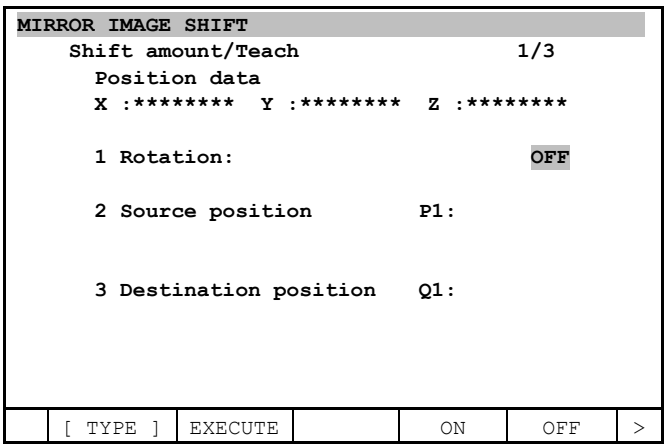
If you do not want to rotate the positions, continue to Step 12.
If you want to rotate the positions, continue to Step 13.

- **If you selected FRAME** for the Image Shift Method, the following Shift Frame Settings screen will be displayed:



Continue to Step 14.

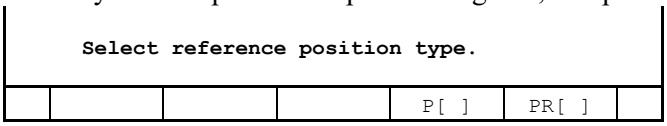
12 If you do not want to rotate the positions,
a. Move the cursor to Rotation, and press F5, OFF. The following screen will be displayed.



b. Move the cursor to source position (see Fig. 24.1 (a)).
c. Move the robot to the source position (P1) and either record or specify the position:

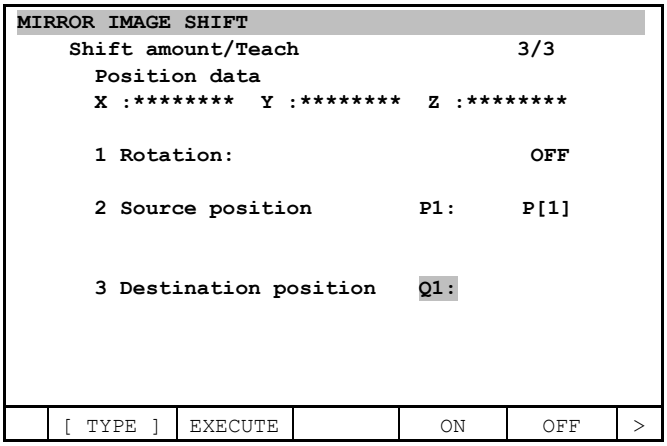
- **To record a position,** jog the robot to the position you want, press and hold in [SHIFT] key and press F5, RECORD.

- **To specify a previously recorded position or position register,** press F4, REFER. Type the number of a previously defined position or position register, and press [ENTER] key.

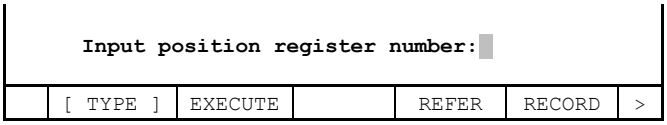


- **To clear a position setting,** press the [NEXT] key, >, and then press F1, CLEAR. At the prompt "Clear all data," press F4, YES, to confirm.

d. Move the cursor to Destination position. The following screen will be displayed.



- **To record a position,** jog the robot to the destination position (Q1). Press and hold in [SHIFT] key and press F5, RECORD.
- **To specify a previously recorded position register,** press the F4, REFER. Type the number of the position register and press the [ENTER] key.

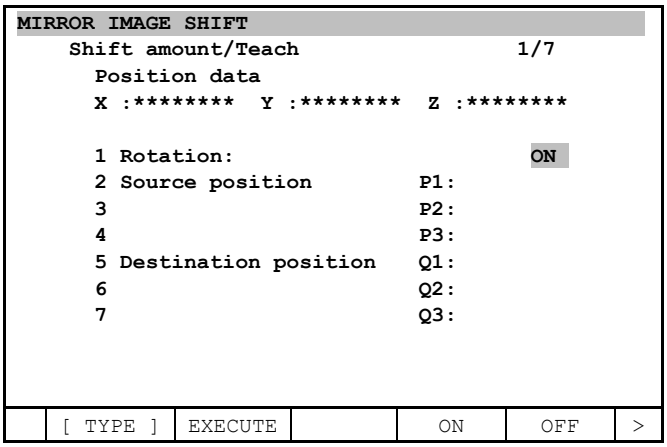


- **To clear a position setting,** press NEXT, >, and then press F1, CLEAR. At the prompt "Clear all data," press F4, YES, to confirm.

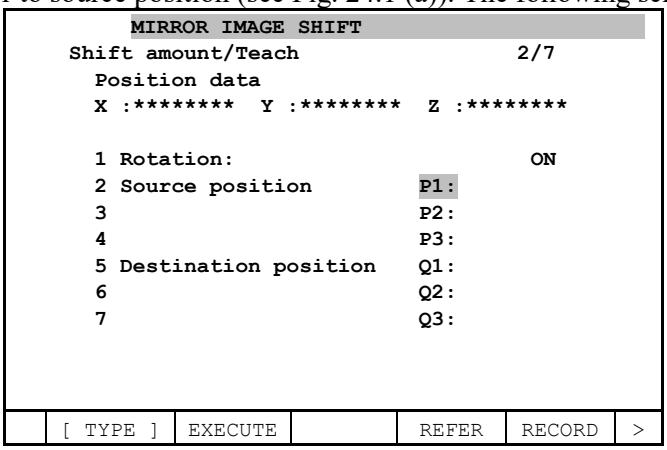
e. Record or specify the destination position.

13 If you want to rotate the positions,

a. Move the cursor to Rotation and press F4, ON. The following screen will be displayed.

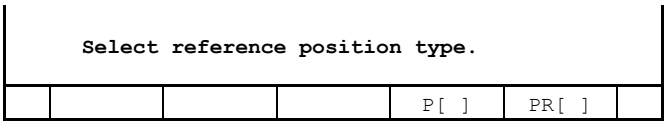


b. Move the cursor to source position (see Fig. 24.1 (a)). The following screen will be displayed.



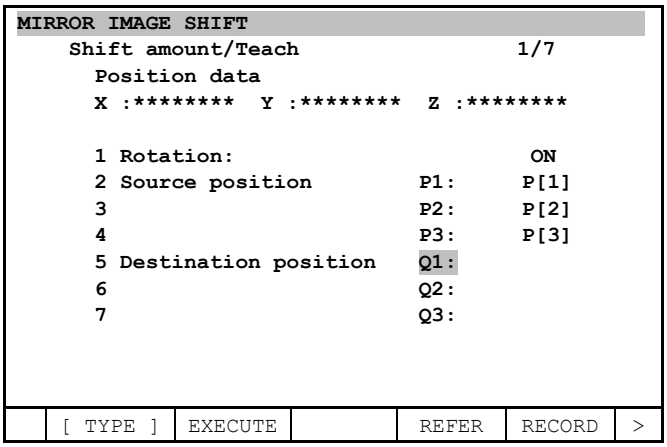
c. Move the robot to the first source position (P1) and either record or specify the position:

- **To record a position,** jog the robot to the position you want, press and hold in [SHIFT] key and press F5, RECORD.
- **To specify a previously recorded position or position register,** press F4, REFER. Select the position or position register.

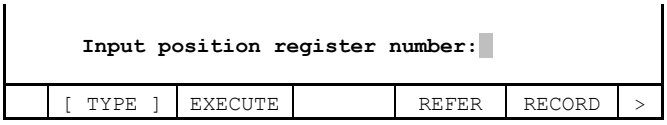


- **To clear a position setting,** press NEXT, >, and then press F1, CLEAR. At the prompt "Clear all data," press the F4, YES, to confirm.

d. Record or specify all the source positions.
e. Move the cursor to Destination position. The following screen will be displayed.



- **To record a position,** jog the robot to the destination position (Q1). Press and hold in the [SHIFT] key and press F5, RECORD.
- **To specify a previously recorded position register,** press the F4, REFER. Type the number of the position register and press the [ENTER] key.

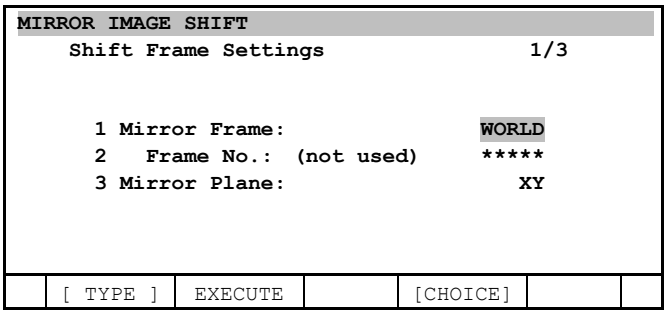


- **To clear a position setting,** press NEXT, >, and then press F1, CLEAR. At the prompt "Clear all data," press F4, YES, to confirm.

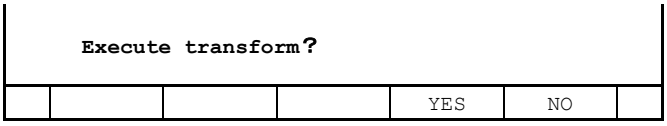
f. Record or specify all the destination positions.

NOTE
Pressing the F2, EXECUTE, will cause the positions you have selected to be mirrored and **will not** cause robot motion.

14 If you selected FRAME as the Image Shift Method, the following screen will be displayed.

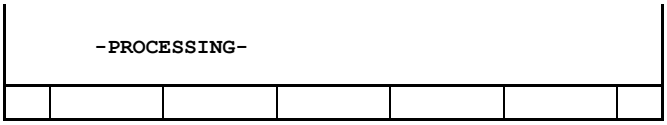


15 If you selected USER or JOG, move the cursor to Frame No. and type the number of the existing frame you want to use.
16 Move the cursor to Mirror Plane and press the F4, [CHOICE]. Select the plane you want to use and press the [ENTER] key.
17 **To mirror image the program,** press the F2, EXECUTE.

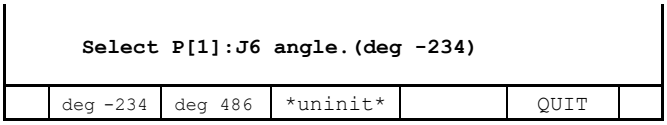


- **To execute the mirror image shift,** press the F4, YES.
- **If you do not want to execute the mirror image shift,** press the F5, NO.

18 Wait until software has finished processing the mirror image.



If the turn number will be changed, the following screen will be displayed.



The above example will be displayed when J6 turn number will be changed.

- If you do not want to change the turn number, press the F1, deg –234.
- If you want to change the turn number, press the F2, deg 486.
- If you want to make the position as un-initialized, press the F3, *uninit*.
- If you want to interrupt mirror image, press the F4, QUIT. At this time, the program has not been changed.

> ⚠ **CAUTION**
> Do not attempt to move the robot to a position that was not mirrored correctly. Please check that result is correct before it operates. Otherwise, you could injure personnel or damage equipment.

When the mirror image is complete, you must reteach each position that did not mirror correctly.

25 VIA INSTRUCTION

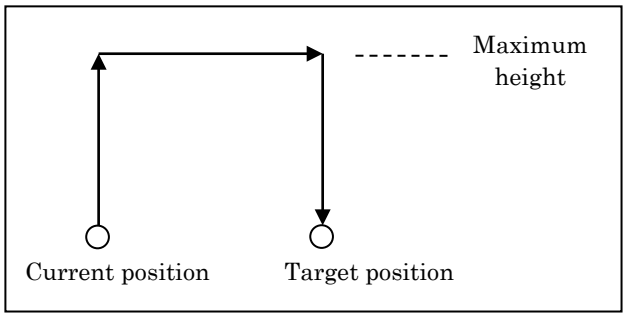
VIA instruction is one of additional motion instructions. This instruction makes it easy for user to teach the motion that goes up to the intended height, go horizontally, and go down to the target position without directly going to the target position.
(This instruction is for SCARA robot. It has been supported since software version 7DF1/20.)

25.1 HOW TO USE

VIA instruction is used by teaching after a motion statement as the following:
L P[1] 4000mm/s FINE VIA(FINE,Z=150)
VIA instruction has three types, DEFAULT, FINE and CNT.

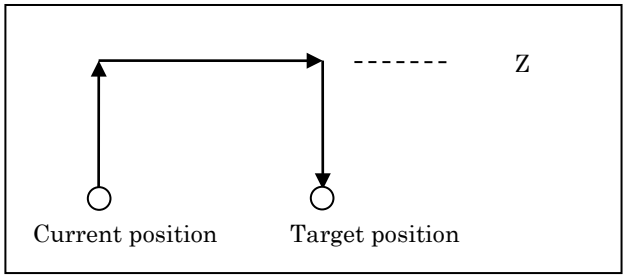
25.1.1 DEFAULT

VIA(DEFAULT)
Robot goes up to the maximum height. This is FINE motion.



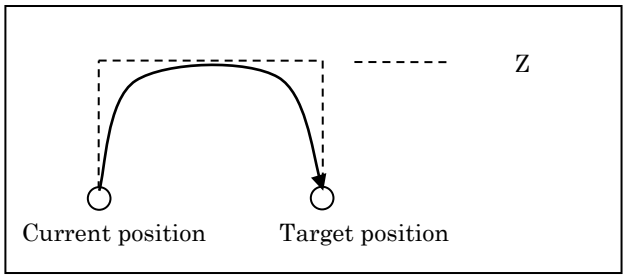
25.1.2 FINE

VIA(FINE,Z=n)
Robot goes up to the height that is specified by Z value. This is FINE motion.

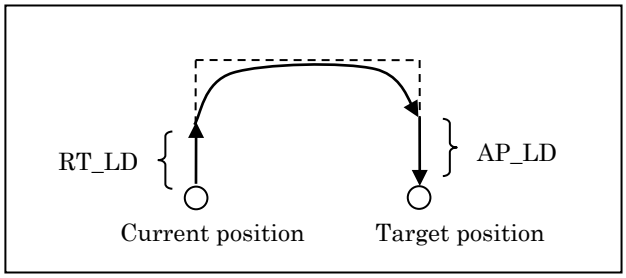


25.1.3 CNT

VIA(CNT,Z=n1,RT_LD=n2,AP_LD=n3)
Robot goes up to the height that is specified by Z value, go horizontally, and go down to the target position by CNT motion.



By specifying the distance to RT_LD, robot moves straight up from the current position to the specified distance.
By specifying the distance to AP_LD, robot moves straight down from the specified distance to the target position.



25.2 LIMITATIONS

- Z is the value of the world coordinate system. (It is not the value of the user coordinate system.)
- This instruction is usable with joint or linear motion. (It is unusable with circular or circle arc motion)
- RT_LD and AP_LD are usable with linear motion. (It is unusable with joint motion.)
- Program name in execution shows "-VIA-" when robot moves by VIA instruction.
- In single-step mode, robot stops not only target point but also two relay points.
- Backward execution to the motion statement with VIA instruction is not allowed.
- The following additional motion instructions are usable with VIA instruction:
  - Acceleration override instruction (ACC)
  - Time before/after instruction (TB/TA)
  - Distance before instruction (DB)
  - Signal output motion option (DO[]=, RO[]=, GO[]= and AO[]=)

  (The other additional motion instructions as Offset condition instruction (Offset) are unusable with VIA instruction.)

26 LEARNING PATH CONTROL

26.1 OVERVIEW

- Learning Path Control (LPC) is a software option that uses iterative learning techniques to improve path performance.
- LPC is an optional function.
- The LPC function requires the Learning Path Control option (R816).
- When LPC is used to improve small shapes performance, Small Circle Accuracy (J644) is recommended to be installed first.
- LPC recognizes the TP program generated by ShapeGen software (R801). LPC's setup will be completed automatically if learning targeted program is ShapeGen created TP program.

26.1.1 Procedure

To set up this function, please follow this sequence:

1. Understand necessary hardware, necessary software, and function limitations. (Detailed information in Ch.2)
2. Set up hardware and software. (Detailed information in Ch.3)
3. Perform learning procedure. (Detailed information in Ch.4)
4. Backup LPC schedule. (Detailed information in Ch. 5)

26.2 HARDWARE AND SOFTWARE

This chapter explains necessary hardware, necessary software, and software limitations.

26.2.1 Hardware

LPC requires the following hardware:

- 64MB D-RAM is necessary. LPC requires around 3MB available memory in D-RAM.

26.2.2 Software

Recommended Software Options

- Using LPC to improve small shape path performance, Small Circle Accuracy (J644) is recommended to be installed and ShapeGen software utility (R801) is recommended to be used to generate small shapes TP program.

NOTE
Small Circle Accuracy (J644) is installable with some specific robot models.

Limitations

- LPC will execute targeted TP program iteratively during learning. Please make sure process such as laser or water-jet will not be turned on automatically (or is safe to be turned on) by iterative TP program execution. It may be necessary to temporarily comment out the "process on" macro before learning.
- Learning results will not be applied (path will change) under following conditions by default:
  - Single Step program execution

- Backward program execution

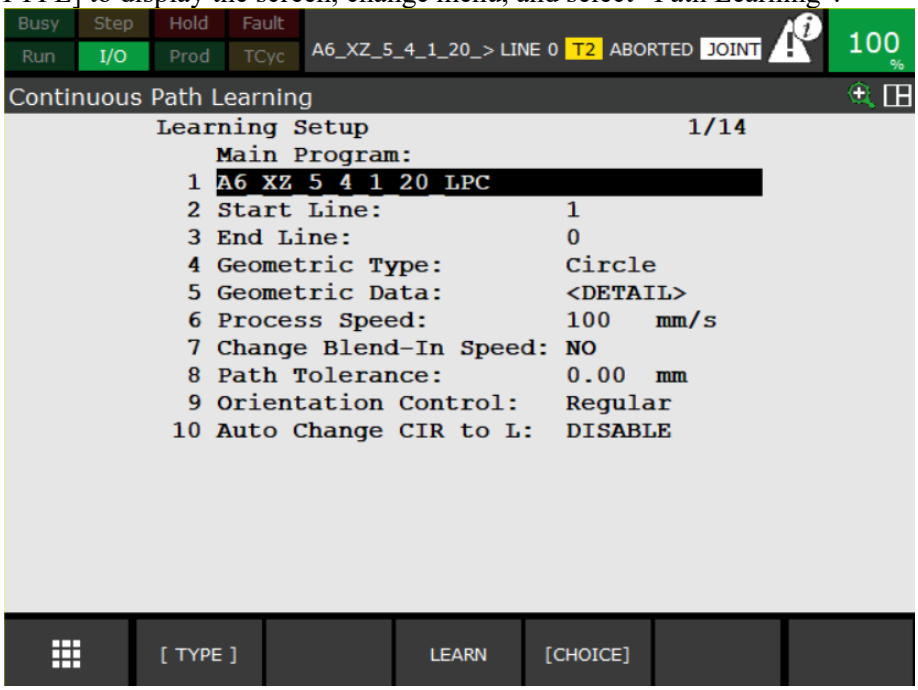
Robot will stop at the original taught TP position.

26.3 LPC SETUP MENU

This chapter explains how to set up LPC for learning. As the first step, LPC must be configured in the setup menu. The setup menu can be launched from TP editor's related view for easy access and better display(Please refer to Subsection 26.3.18). The following procedure is used to set up LPC.

Procedure 26-3-1 Displaying the LPC setup menu

1 Press the MENUS key.
2 Select "6 Setup".
3 Press F1, [TYPE] to display the screen, change menu, and select "Path Learning".



4 The LPC setup menu screen appears.

26.3.1 "Main Program" ITEM

This is the TP program for LPC to be executed during learning. The LRN motion modifier TP instruction will be automatically added from the Start Line to End Line to store any learned data in this targeted TP program.
If this targeted TP program is generated by ShapeGen utility, the geometric data and program related information will be extracted automatically and populate to the necessary menu items. Please double-check the setup items if the TP program was generated by ShapeGen and manually modified afterward.

26.3.2 "Start Line" ITEM

This menu item specifies the start TP line for learning process. LRN motion modifier will be added starting from this line.

26.3.3 “End Line” ITEM

This menu item specifies the end TP line for learning process. LRN motion modifier will be added before from this line.

26.3.4 “Geometric Type” ITEM

This menu item specifies the geometric type for learning process such as:

- Circle
- Slot
- Rectangle
- Free Form

NOTE
Currently LPC will improve the path performance based on theoretic data for “Circle” type only. LPC will improve the path performance based on the taught points information for the rest of geometric types.

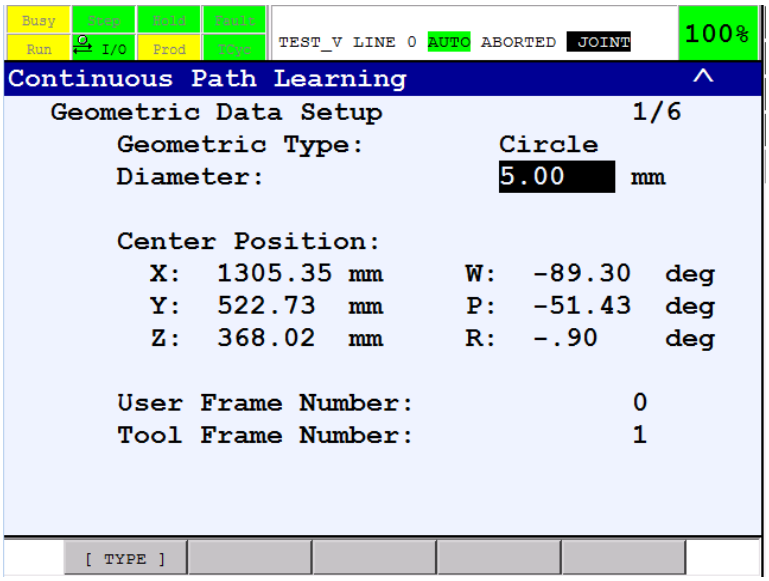
26.3.5 “Geometric Data” ITEM

This menu item specifies the detailed geometric type for learning process. Press Enter when cursor is at “<DETAIL>” to see the detailed data page.

“Geometric Data <DETAIL>” Item

The detailed geometric data for learning process is including:

- Diameter (when geometric type is Circle): the diameter of circle
- Length (when geometric type is Slot or Rectangle): the length from the origin of the shape
- Width (when geometric type is Slot or Rectangle): the width from the origin of the shape
- Center Position
- User Frame Number
- Tool Frame Number



26.3.6 "Process Speed" ITEM

This menu item specifies the programmed speed for the targeted TP program. Please note all linear and circular motion lines will be modified to this speed when learning starts.

26.3.7 "Change Blend-In Speed" ITEM

This menu item specifies whether to modify "Blend-In" motion line's programmed speed to be the above menu item. If the targeted TP program is not generated by ShapeGen, this menu item will not take effect.

In the beginning of learning process, LPC changes programmed speed of TP lines from the specified start line to the specified end line in the targeted TP program using menu item "Process Speed" automatically. During learning iteration, the targeted TP program will be executed repeatedly and robot will move from the end of TP program back to the first line of TP program iteratively. User can use this menu item to prevent LPC changing the speed of the first line of TP program (Blend-in PT line in ShapeGen generated programs) so the learning process will be smoother and vibration free when robot travels back from the end of program.

26.3.8 "Path Tolerance" ITEM

This menu item specifies the targeted path deviation during learning. Learning process will stop once path deviation reaches this number or maximum iteration reaches the menu item. For geometric type is circle, the path deviation will calculate based on the theoretic data to the center of the circle. For other geometric types, the path deviation will calculate based on the closed taught points.

26.3.9 "Orientation Control" ITEM

When LPC is used for small shapes, J1 may be fixed to reduce the backlash impact to the path performance. To use this special 5-axis motion for small shape to ensure the perpendicular cutting wall, please select "J1 Fixed" orientation control for learning process. LRNR-005 may be posted if this TP program's Tool orientation is not suitable for this special orientation control. Please simply change this menu item back to "Regular" and start the learning process, or spin about the tool's approach vector (jog robot using + or - in Tool Frame Z rotation direction) and re-generate the TP program with new tool approach vector then start the learning process.
When using "J1 Fixed" orientation control, it is also recommend to use linear motion TP lines instead of circular motion TP lines. To let LPC automatically change all circular motion lines from start to end to linear motion lines, please enable the next "Auto Change CIR to L" item.

26.3.10 "Auto Change CIR to L" ITEM

When this menu item is enabled, all motion lines between start line and end line will automatically change to L (Linear) motion type if motion type is C (Circular motion) during learning process.

26.3.11 "Max. Iterations" ITEM

This menu item specifies the maximum learning iterations. Learning process will stop once path deviation reaches "Path Tolerance" menu item or maximum iteration reaches this number.

26.3.12 "Process Disabled" ITEM

This menu item specifies the process and macro program are enabled/disabled during learning process. When this item is set to "TRUE", learning process will execute without the process and macro program.

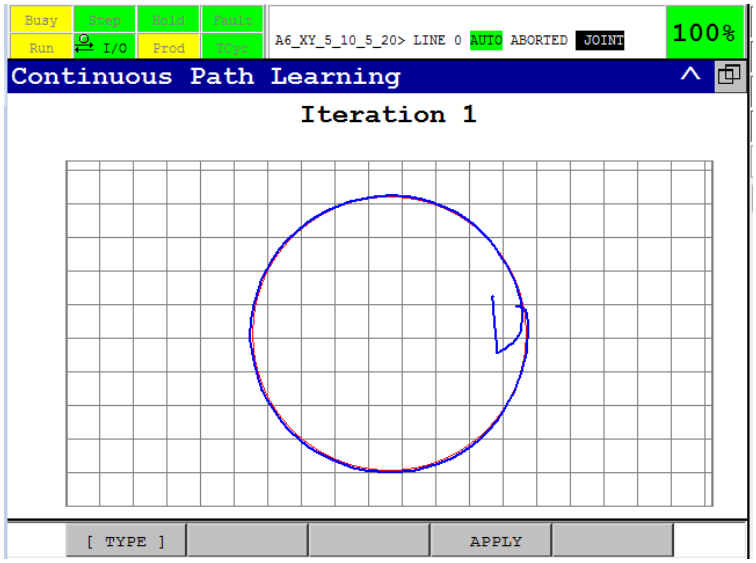
26.3.13 "LEARN" FUNCTION KEY

The "LEARN" function key begins the learning sequence.

- Please make sure robot is safe to run this TP program iteratively (is safe to run from the end of TP program back to the first line of TP program).
- System is in Auto mode and the speed override is at 100% and no fault present.
- When learning process starts, major keys will not be functional but any alarm/fault will stop the learning process automatically.
- Learning data will store in LRN motion modifier from start to end TP lines automatically.
- If learning menu is launched from related view, the TCP trace will be displayed in lower right pane for reach iteration automatically.
- When learning stops without errors, the final maximum path deviation will be displayed.

26.3.14 "RESULT" FUNCTION KEY

When learning process is complete, "RESULT" function key will show up. This function key will display the TCP trace of learning iterations. If the geometric type is circle, the theoretic circle will also display as in red for better comparison. Use Right or Left arrow key in this window to change the display of next or previous iteration respectively.



26.3.15 "APPLY" FUNCTION KEY

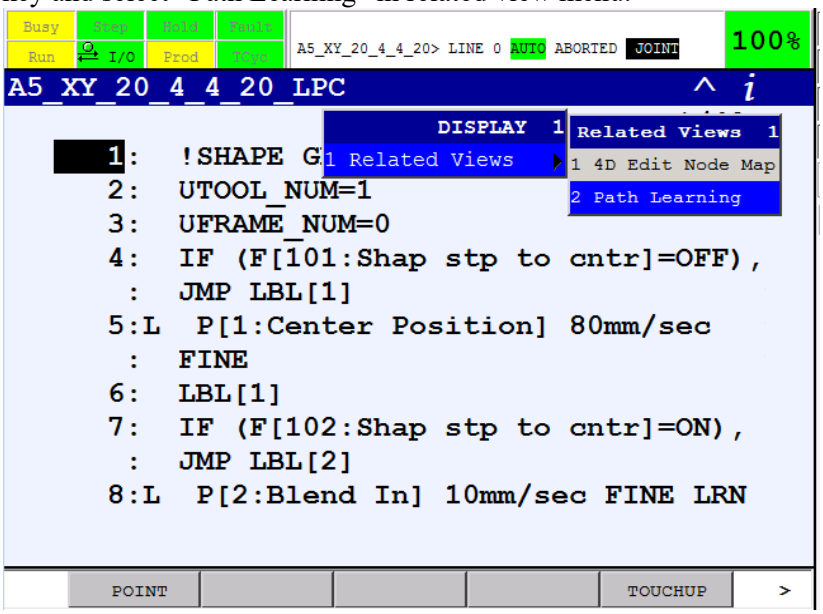
In the display window, "APPLY" function key will be available to use the displayed TCP trace as a final result and overwrite current learned data.

26.3.16 RELATED VIEW

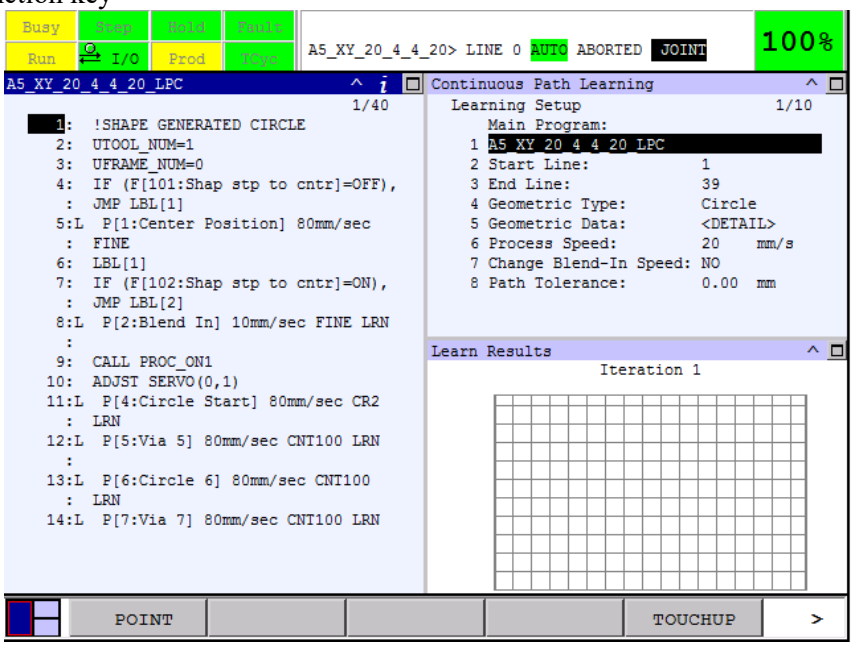
The setup menu can be launched easily when targeted TP program is opened in TP editor using related view. The TCP trace will display in the lower right pane real time during learning.

Procedure 26-3-2 Displaying the LPC related view menu

1 Open targeted TP program in editor.
2 Press *i* + Fctn key and select “Path Learning” in related view menu.



3 After Path Learning related view windows opened, move the focus to upper right pane for setup and LEARN function key



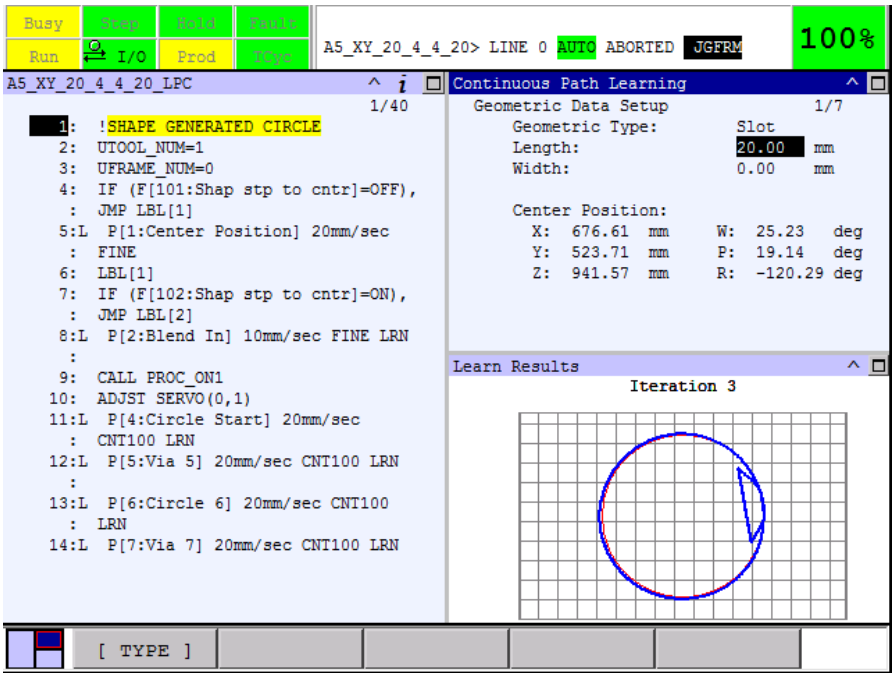
26.4 LPC LEARNING AND PLAYBACK

This chapter explains LPC learning and playback.

26.4.1 Learning

Learning will start when LEARN function key is pressed and all required conditions matched. The targeted TP program will be executed repeatedly until either path tolerance or maximum iteration is reached. During learning process, most keys on the teach pendant may not be functional. Learning can be stopped at any time when alarm occurs.

If learning is started from the related view, the TCP trace for iteration will be displayed automatically at lower right pane.



After learning is complete, the learned data is stored in LRN motion modifier for playback during production.

26.4.2 Limitations

Please make sure process such as laser or water-jet will not be turned on automatically (or is safe to be turned on) by iterative TP program execution. It may be necessary to temporary comment out the “process on” macro before learning.

26.4.3 Production with LPC Playback

The TP program with LRN motion modifier TP instruction will apply the learned data automatically when executed. Learned data will not be applied under following conditions:

- Single step mode
- Backward execution

Robot will stop at the original taught position in above conditions.

26.5 LPC BACKUP

The learned data is stored in the LRN TP instructions.
The regular TP program backup procedure will backup learned data (LRN TP instruction) automatically. However, ASCII format of LS file will not have learned data converted. Please do not backup TP program after LPC learning in LS (ASCII) format.

> ⚠ **CAUTION**
> Converting TP program with LRN motion modifier inside to LS file will lose learned data.

27 TORQUE CONTROL

The Torque Control function (R675) provides the following functions:

- Dynamic limit of the maximum motor torque including robot axes. (Refer to the 27.1 DYNAMIC TORQUE LIMIT).
- Permanent limit of the maximum motor torque for additional motor (Refer to the 27.2 PERMANENT TORQUE LIMIT FOR ADDITIONAL MOTOR).

27.1 DYNAMIC TORQUE LIMIT

This function works as follows:

- Provides schedules that allow you to assign torque limits for a specific auxiliary or robot axis.
- Automatically adjusts the error checking level to be suitable for the system when the deviation (error) between the taught position and the actual position is too large for the axis for which the torque limit is specified. You do not need to change and set this yourself.
- Automatically updates the program position to the contact position when Torque Control is used (Auto Position Update). The controller remembers the position where the specified axis stops when it makes contact with the workpiece (contact position). The destination position can be automatically updated to the contact position (Auto Position Update).
- Allows the robot to maintain the position where contact is made and increase the actual torque to overcome the motion dynamics limitation without crushing the workpiece.

This function automates some of the required settings, simplifying and improving programming for servo grippers and other servo-driven axes.
For example, as shown in Fig.27.1 (a), standard teaching requires you to teach the hand position according to the size of the workpiece. The Torque Control function limits the torque applied to the hand, making it possible to handle multiple workpieces using the closed hand position as the teaching position. In addition, the tool's floating mechanism can be omitted or simplified, because it can absorb variations in workpiece size. The Torque Control function allows fine adjustment of the gripping position and torque limit value after gripping the workpiece, so that different size workpieces can be gripped at the same teaching position without crushing them.

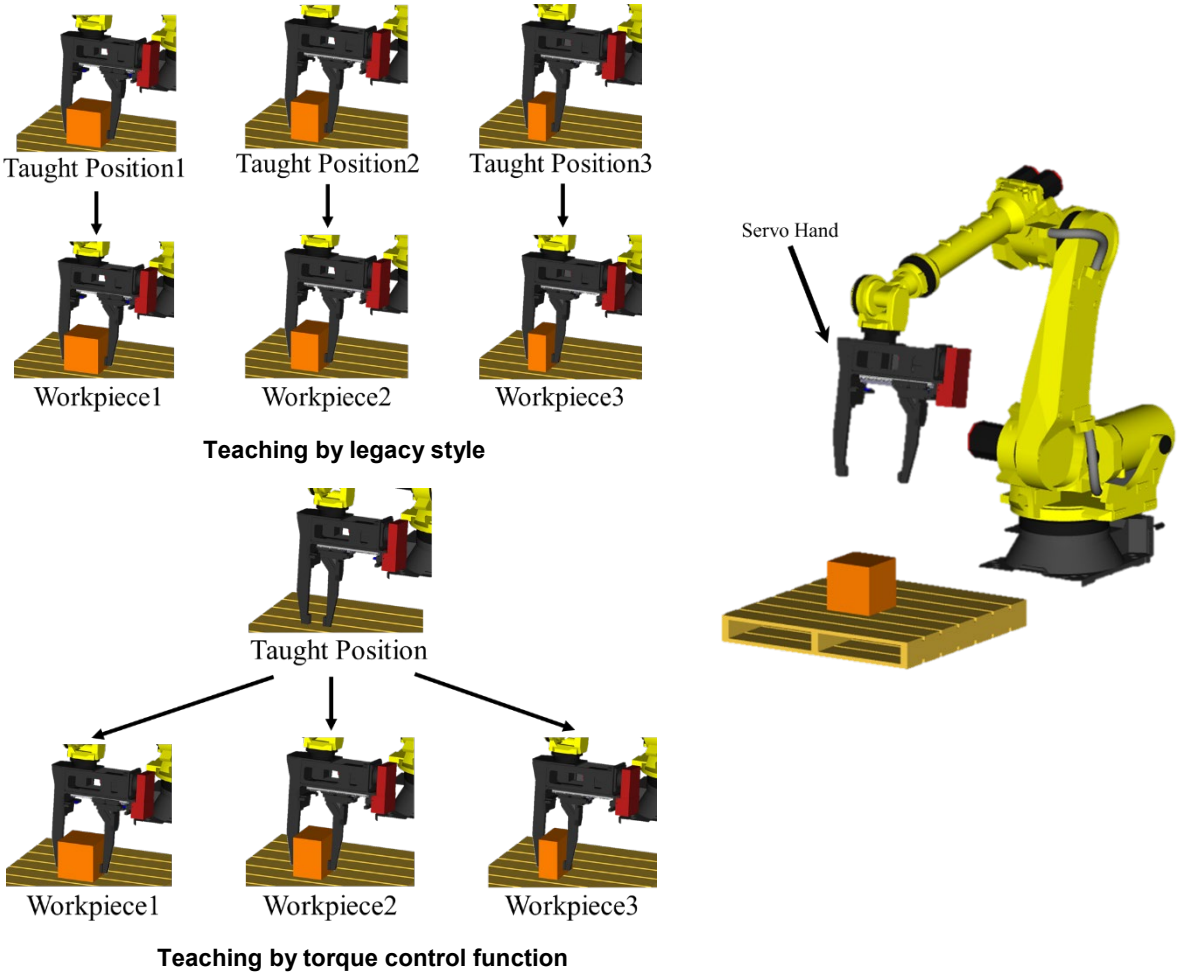


Fig. 27.1 (a) Example of using the Torque Control Function

You specify torque limits for auxiliary and robot axes using Torque Control schedules. To use a particular Torque Control schedule during program execution, include a TORQ_CTRL TP instruction in your TP program.

> ⚠ **WARNING**
> 1 Enabling torque limits on robot axes could cause damage equipment or injure personnel due to falling caused by gravity or deviation from its original trajectory.
> 2 Because regular error checking will be reduced when Torque Control is enabled. verify the path with extra caution.

27.1.1 SETTING UP THE DYNAMIC TORQUE LIMIT

Set up Torque Control schedules on the following SETUP Torque Control screens:
- Schedule screen
- Detailed Data screen

You can set up to 10 schedules for the Dynamic Torque Limit.

Procedure 27-1 Setting up Torque Control

Step

1. Press MENU.
2. Select SETUP.
3. Press F1, [TYPE] And select Torque Control.

4. Move the cursor to the schedule you want to set and press F2, DETAIL. The Detailed Data will be displayed.
5. Set the items as desired. To return to the Schedule List screen, press F2, LIST.

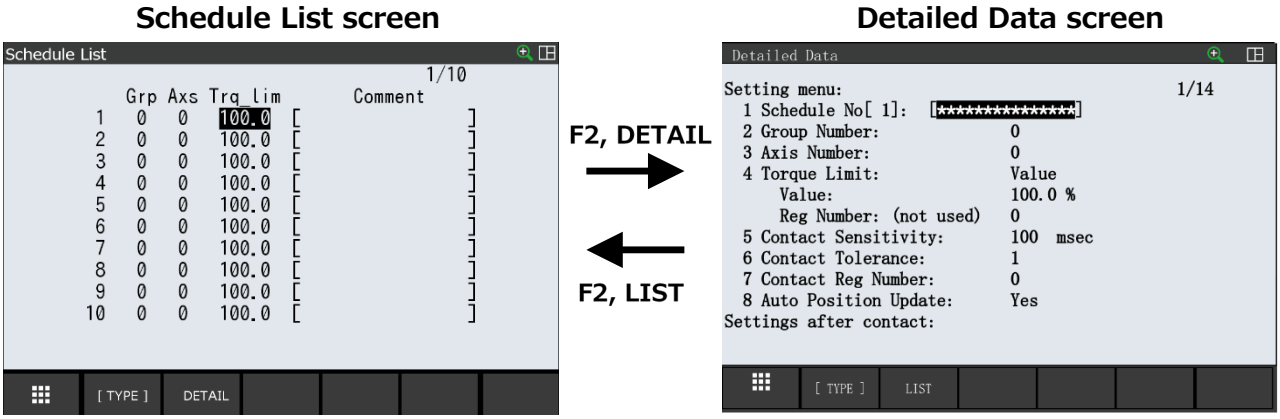


Fig. 27.1.1 (a) Torque Control setup screen

Setting items

You can set the schedule items listed in Table 27.2 (a) on the Detailed Data screen.

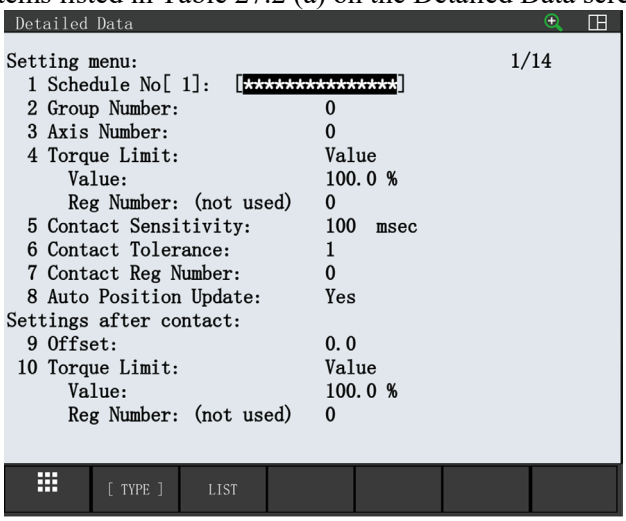


Fig. 27.1.1 (b) Torque Control Detail screen

Table 27.1.1 (a) Setting items of Torque Control function

Item		Default	Description
Setting menu:	Schedule No[ ]		This item specifies the Torque Control schedule number. If you press ENTER with the cursor on line 1, you can enter a comment for the schedule.
	Group Number	0	This item specifies the group number of the axes to be controlled.
	Axis Number	0	This item specifies the axis number whose torque you want to specify. With some exceptions, you cannot select robot axes. See Section 27.4 for details.
	Torque Limit Range : 0% to 100%	Value	This item specifies how to specify the torque limit value. - Value: The torque limit value specified on the detail screen is used. - Reg.: The torque limit value set in the register specified by the Reg. Number is used. If the maximum current of the amplifier is limited to protect the motor, this torque limit value identifies the percentage of the torque generated by the limited maximum current.
	Contact Sensitivity	100	This item specifies how long (msec) the servo axis remains idle. When the torque control axis is idle for longer than this time, the contact between the servo axis and the part will be determined, and the next instruction will be executed.
	Contact Tolerance	2	This item specifies the idle condition of the torque control axis in encoder counts. On the torque control axis, if the encoder count that changes in one control cycle is less than this value, it is considered to be idle. If it takes too much time to be judged as not idle, increase this value.
	Contact Reg Number	0	This item specifies the numeric register number for contact conditions after a TORQ_CTRL[ ] TP instruction is executed. The Contact Register values are as follows: 0: Torque Control has not started 1: the axis starts moving after a TORQ_CTRL TP instruction 2: the axis is in contact with the workpiece and the next motion line will start -1: the axis has not contacted the workpiece, but the destination position has been reached.
	Auto Position Update	Yes	If set to Yes, the position data of the specified axis between the TORQ_CTRL[ ] and TORQ_CTRL END instructions is updated automatically to the contact position. You do not need to use an offset command or PR[ ] value when teaching the Torque Control program. If set to No, the destination position will not be updated even if contact is made, and the original taught position will be used.

Item		Default	Description
Settings after contact:	Offset	0	This item specifies the position offset at which contact is made. A positive value means that the specific axis will be commanded to move an offset in the positive axis direction. This item can be used to control the tightness between the end effector and the workpiece. If Auto Position Update is set to NO, this item will be disabled and the original taught point will be used. Units : Rotary axis : rad Linear axis : mm
	Torque Limit Range: 0% to 100%	Value	This item specifies how to specify the new torque limit to be applied the workpiece after contact is made. This value can be used to control the tightness between the end effector and the workpiece. - Value: The torque limit value specified on the detail screen is used. - Reg.: The torque limit value set in the register specified by the Reg. Number is used.

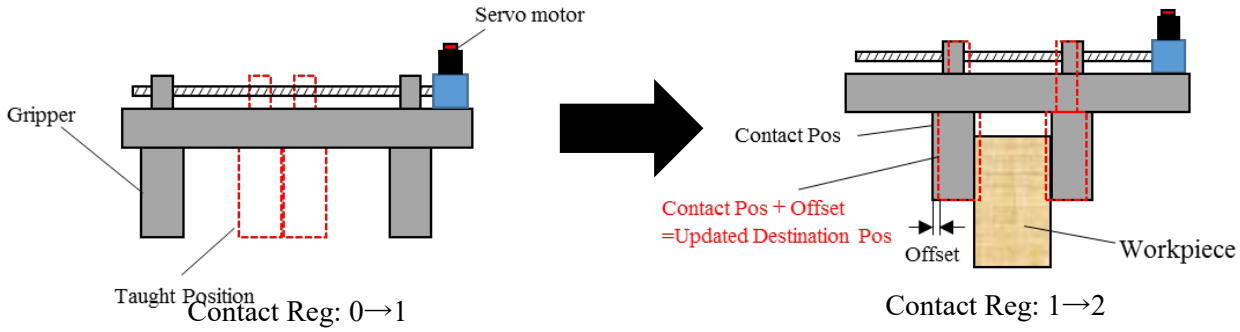


Fig. 27.1.1 (c) Example of contact / non-contact between servo hand and workpiece (Auto Position Update is set to Yes): With workpiece

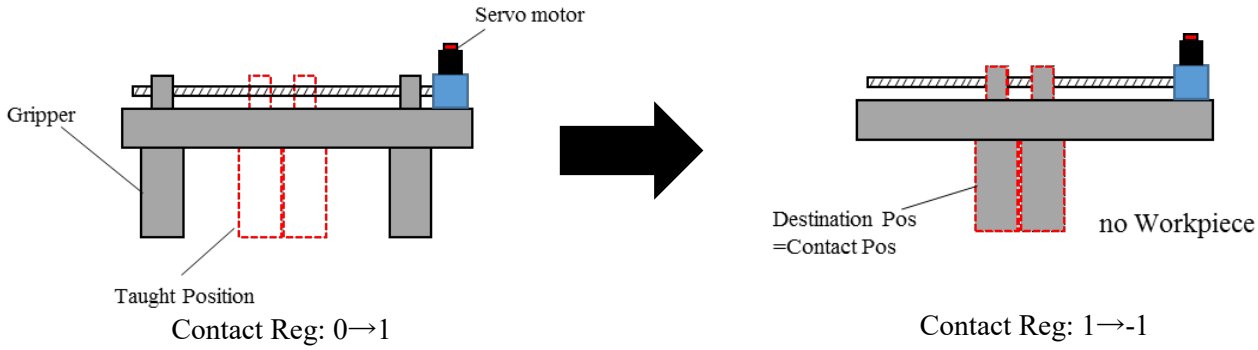


Fig. 27.1.1 (d) Example of contact / non-contact between servo hand and workpiece (Auto Position Update is set to Yes): No workpiece

27.1.2 PROGRAM INSTRUCTION

The Dynamic Torque Limit function includes the following program instructions:

- TORQ_CTRL[schedule number]
- TORQ_CTRL END / TORQ_CTRL END[schedule number]

TORQ_CTRL[schedule number]

When a TORQ_CTRL[schedule number] instruction enables the specified Torque Control schedule, the following occurs:

- The torque limit of the specified axis is enabled.
- The Torque Control function is enabled.
- When contact is made (for example, when a gripper contacts a workpiece), a quick stop is used and the deceleration stop processing of the normal motion is omitted. This prevents hesitation that could increase cycle time.
- When Auto Position Update is enabled, the contact position of the specified axis is updated to the destination position. If the taught point is reached without contact, the taught point will be updated as the destination position. This contact position (destination position) is applied until the TORQ_CTRL END instruction is executed.

Torque limits and position offsets can be kept in the subsequent motion instructions until the TORQ_CTRL END instruction is reached or the TP program execution status is ABORTED.

NOTE
When you execute TORQ_CTRL[schedule number] instruction, the group number specified in the schedule must be included in the group mask of the TP program, otherwise an alarm “INTP-216 Invalid value for group number” will be posted.

TORQ_CTRL END / TORQ_CTRL END[schedule number]

The TORQ_CTRL END instruction disables Torque Control. When a TORQ_CTRL END instruction is executed, the following occurs:

- The torque limit is reset to its default value.
- Error checking is reset to normal.
- The position when this instruction is executed is updated to the destination position.
- The TORQ_CTRL END[schedule number] can be used to disable Torque Control for a specific schedule.

NOTE
If multiple Torque Control enabled axes are configured in the same group, even though TORQ_CTRL END[schedule number] may be disabled independently, all motion in the same group must to be stopped at the same time; therefore all axes in the same group may still be affected.

NOTE
Execute the TORQ_CTRL END instruction while the axis for which the Torque Control is enabled is stopped. If the instruction is executed while the axis is moving, execution of the instruction is not complete until the axis stops.

TP Program Example

Figure 27.1.2 (a) shows an example of gripping a workpiece with the servo hand using the Torque Control function. The details of Figure 27.1.2 (a) are described as follows.

- The robot and servo tool are moved simultaneously.
- In line 4, the TORQ_CTRL[1] instruction enables the torque limit for the servo tool and enables the automatic updating of the contact position.
- In line 5, the taught position (P[3]) of the servo tool is completely closed, but the gripping position (P’[3]) of the workpiece is updated to the destination position of the servo tool using the Torque Control function.
- The gripping position of the servo tool is maintained until the TORQ_CTRL is executed, and the robot moves while gripping the workpiece.

```
:
:
4 : TORQ_CTRL[1]
5 : L P[3] 1500mm/s    FINE
6 : J P[4] 100%    CNT100
7 :TORQ_CTRL END
```

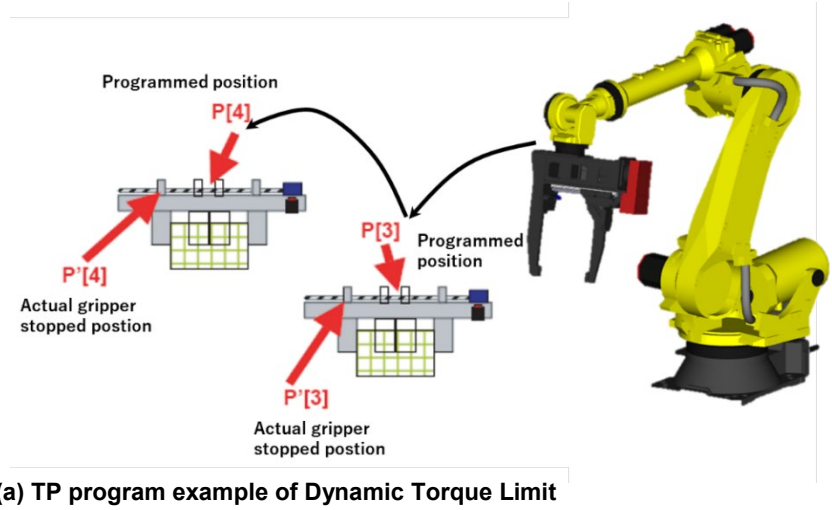


Fig. 27.1.2 (a) TP program example of Dynamic Torque Limit

27.1.3 LIMITATIONS

The Dynamic Torque Limit function has the following limitations:

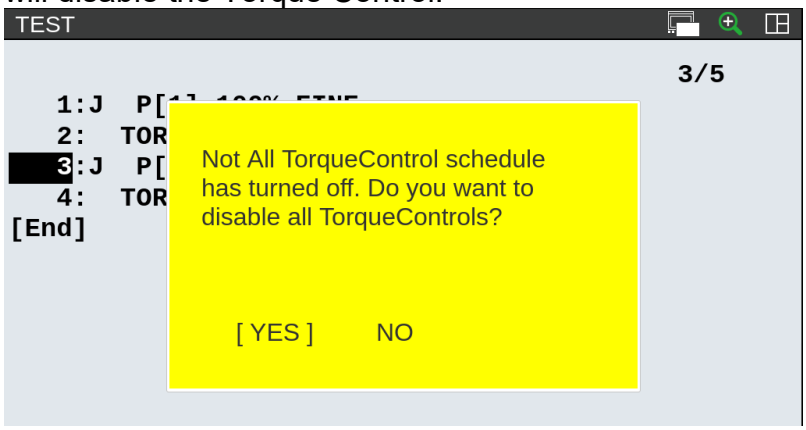
- In single step mode or when the program is paused, the TORQ_CTRL[ ] TP instruction will only activate the torque limit servo setting. Position updates and offsets will not be applied.
- When a program aborts, all torque limits will be reset to 100% value and error checking limits will be reset to their default values.
- As with other TP instructions, if the program resumes or skips to the motion instructions between the TORQ_CTRL[] and TORQ_CTRL END instructions; no Torque Control schedule will be used.
- Multi-arm multi-tasking applications are not supported.
- If DCS includes a Torque Control enabled axes, a DCS FB_CMP alarm may occur when or after Torque Control is used. Refer to DCS manual for details.

> ⚠ **WARNING**
> Motor torque is low and regular error checking is disabled when a program is paused while Torque Control is enabled.

⚠ WARNING

Even if the TP program is paused while the Torque Control is enabled, the Torque Control is not disabled and the torque limit value is not restored. Performing any of the following operations while the Torque Control is enabled will disable it.

- Execution of TRQ_CTRL END instruction.
- Cycling power of the controller.
- Ending/Aborting the TP program.
  - If you press ABORT (ALL) in FCTN key menu while the Torque Control is enabled, the following pop-up window appears, and selecting “YES” will disable the Torque Control.



- The Torque Control function cannot be enabled for robot axes, with exceptions.
  When TORQ_CTRL instruction is executed, the alarm “INTP-663 **(%s^4, %d^5)** Torque Limit cannot be used” will be posted.
  Exceptions:
  J1 axis of a robot whose J1 axis is a rail.
  J3 axis of a SCARA robot with 6kg or less payload capacity.
  Final axis of an opener robot for painting application.

27.1.4 CAUTIONS

- You can use Torque Control for robot axes, but do not enable Torque Control for any robot axes that can fall under their own weight.
- When Auto Position Update is disabled, the contact position will not be destination position even if it touches a workpiece. When an axis is stopped by a Torque Limit, the external force is equal to the limited motor torque, and the distance between the current (contact) position and destination position is considered to be a position error. If the torque limit after contact is increased, the axis moves to the destination with a larger torque than the torque before contact; therefore, you might break the servo tool or work piece.

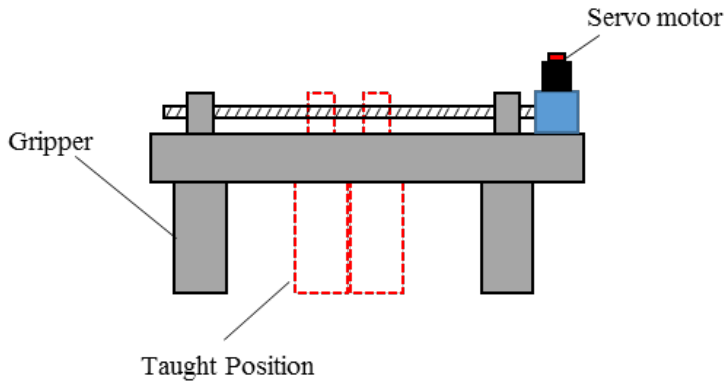


Fig. 27.1.4 (a) Taught Position of the servo hand

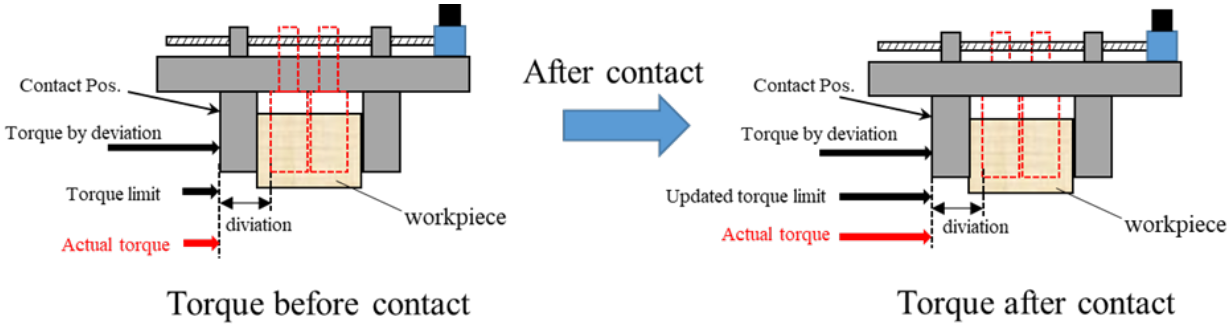


Fig. 27.1.4 (b) Auto position update disabled
If the torque limit after contact is large, the torque for the error amount is applied to the workpiece

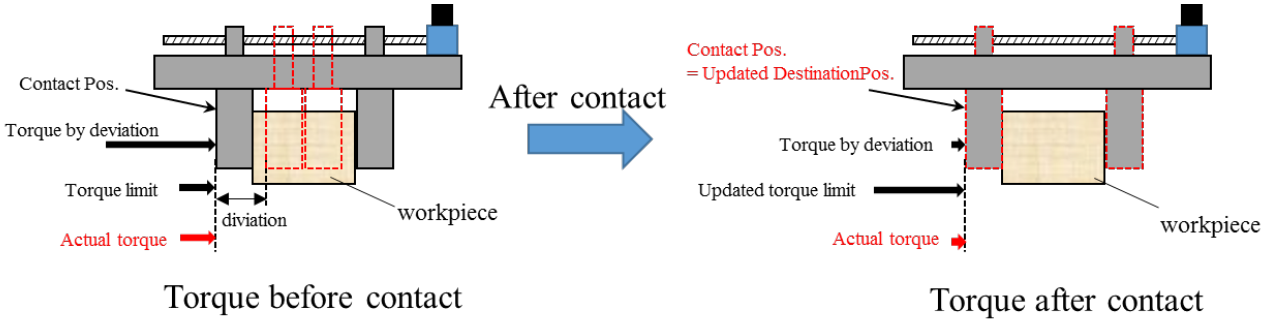


Fig. 27.1.4 (c) Auto position update enabled.
Even if the torque limit after contact is large, the actual torque is small.

27.2 PERMANENT TORQUE LIMIT FOR ADDITIONAL MOTOR

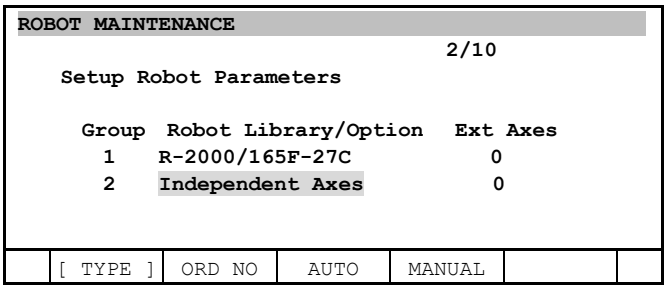
The maximum torque of the additional motor can be limited permanently for the Extended axes (J518), Independent axes (H895), Positioner (H874, H896) and Process axes(R689).
In this section, this function is referred to as the Permanent Torque Limit. Torque limit by the TORQ_CTRL TP instruction is referred to as Dynamic Torque Limit.

> ⚠ **WARNING**
> Please do NOT enable this function for any axes which may be fallen down by the gravity.

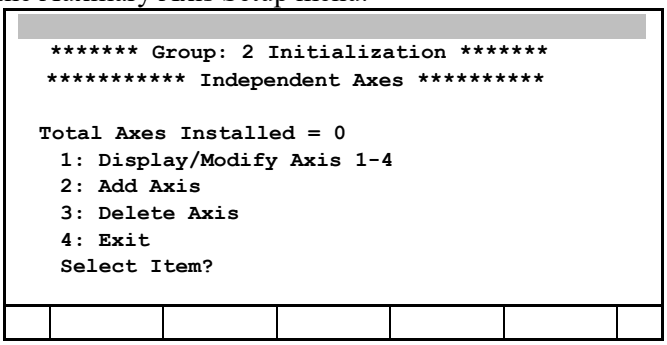
27.2.1 SETTING UP PERMANENT TORQUE LIMIT

Permanent Torque Limit can be set up at Auxiliary Axis Setup menu.
The following is an example of Independent axes.

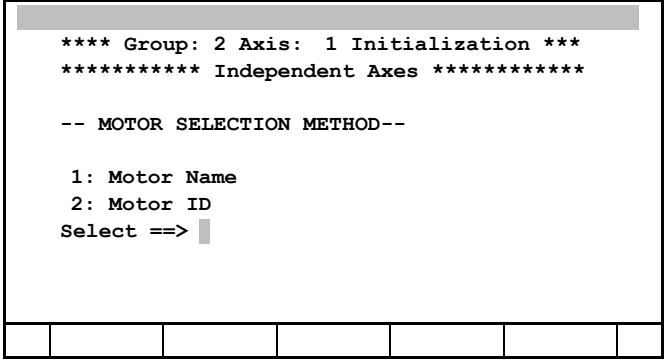
1. Open the Auxiliary Axis Setup menu.
   (a) In the Controlled Start, press [MENU] key and select "9. MAINTENANCE".
   (b) Move the cursor to the group to which you want to setup. Then press F4, MANUAL.



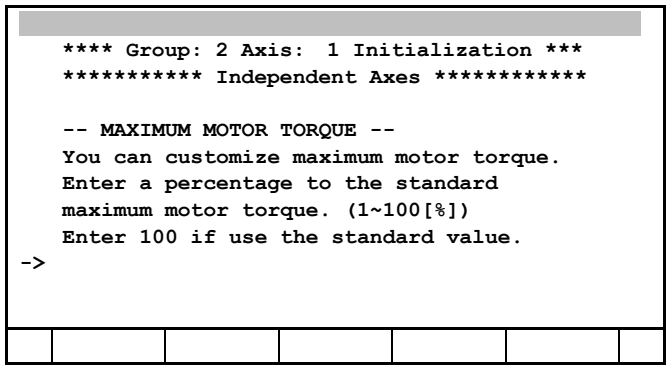
   (c) You will see the Auxiliary Axis Setup menu.



2. Select "2: Add Axis" on the Auxiliary Axis Setup menu, then you will see the following motor selection screen. Select the motor according to the mechanical specification.



3. You will see the maximum motor torque setting screen. If you need to limit the maximum motor torque, enter a percentage to the standard maximum motor torque. If need not to limit the maximum motor torque, enter 100.



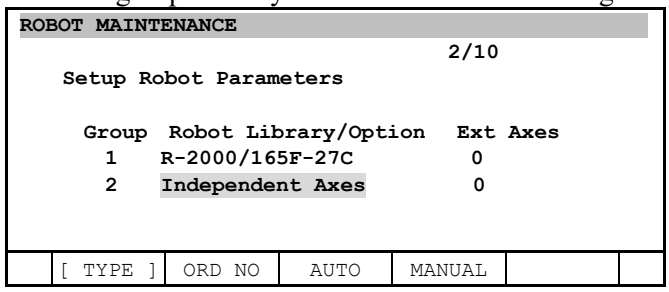
NOTE
This screen is only displayed if Torque Control function (R675) is installed.

4. The rest of the procedure is the same as for normal axis addition. Please refer to the following section in chapter B. SPECIAL OPERATION of the "OPERATOR'S MANUAL (Basic Function)" (B-84464EN) to set up the axis.
   - B.9 POSITIONER SETUP
   - B.10 EXTENDED AXIS SETUP
   - B.11 INDEPENDENT ADDITIONAL AXIS SETUP
   - B.12 PROCESS AXES SETUP

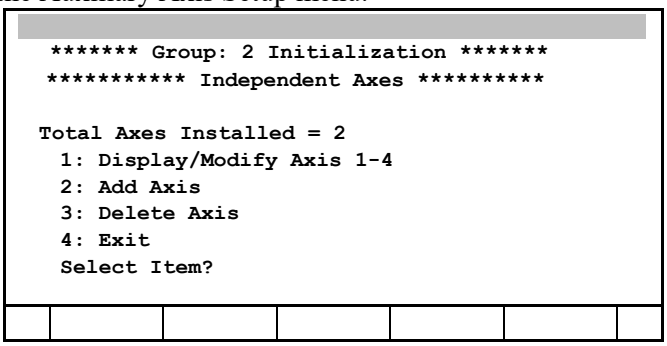
27.2.2 DISPLAY / MODIFY PERMANENT TORQUE LIMIT

The following is an example of Independent axes.

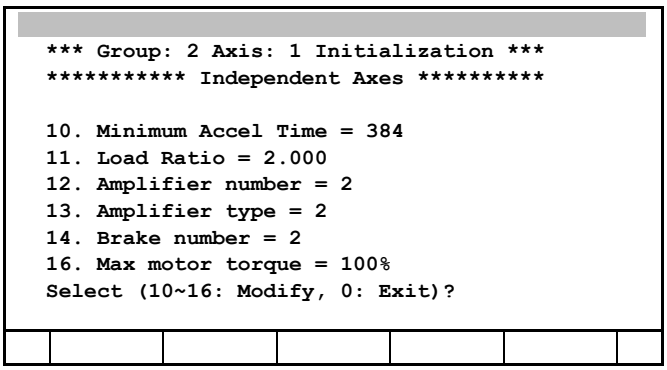
1. Open the Auxiliary Axis Setup menu.
   (a) In the Controlled Start, press [MENU] key and select "9. MAINTENANCE"
   (b) Move the cursor to the group where you want to check the settings. Then press F4, MANUAL.



(c) You will see the Auxiliary Axis Setup menu.

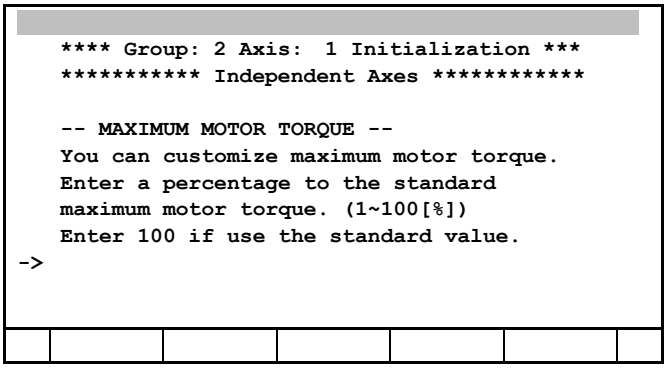


2. Select “1: Display/Modify Axis” on the Auxiliary Axis Setup menu, and then enter the axis number you want to confirm.
3. When the Display/Modify screen shows up, enter “0: Next Item” several times, and then confirm the "Max motor torque ".



[If modify Permanent Torque Limit]

Select "Max motor torque" item number, then you will see the following screen. You can change the maximum motor torque in this screen.



NOTE

If the Permanent Torque Limit value is changed for the axis that the Dynamic Torque Limit has been set already, the following screen will be displayed. Refer to the “CAUTIONS” in the section "COMBINED USE OF PERMANENT TORQUE LIMIT AND DYNAMIC TORQUE LIMIT" and adjust the torque limit value on the torque control setting screen.

```
**** Group: 2 Axis:  1 Initialization ***
*********** Independent Axes ************

-- MAXIMUM MOTOR TORQUE --
This axis is used in Torque Control.
If you change the maximum motor torque,
you should adjust the Torque Limit
in the Torque control setting menu
```

27.2.3 COMBINED USE OF PERMANENT TORQUE LIMIT AND DYNAMIC TORQUE LIMIT

When you set up both the Permanent Torque Limit and Dynamic Torque Limit and turn them on, the two limit values are multiplied. For example, under the following conditions, the combined torque limit value will be 20[%].

- Permanent Torque Limit 50[%]
- Dynamic Torque Limit 40[%]

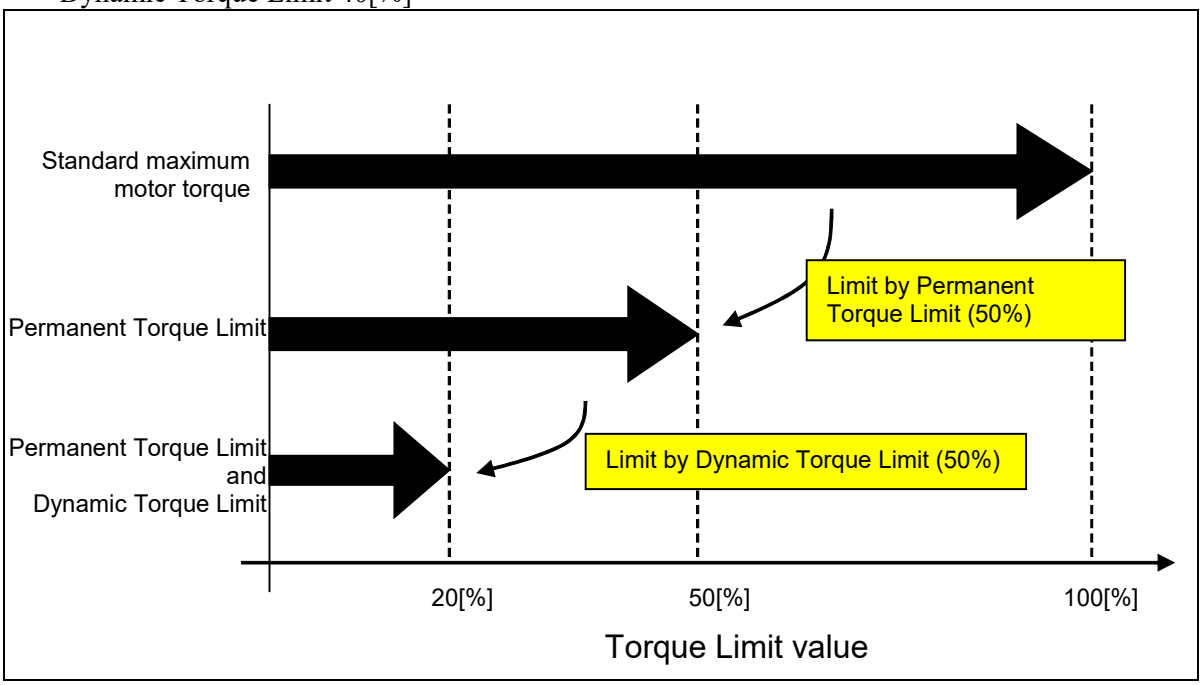


Fig. 27.2.3 (a) Example of combined use of Permanent Torque Limit and Dynamic Torque Limit

CAUTIONS

Since the value of Permanent Torque Limit and Dynamic Torque Limit are multiplied, if you change the Permanent Torque Limit value for the axis on which the Dynamic Torque Limit has been set, the combined limit value will change. If you want to keep the combined limit value, adjust the dynamic torque limit value at the torque control setting screen.

Example:

In the above example, the combined limit value is 20[%] under the following conditions.

- Permanent Torque Limit 50[%]
- Dynamic Torque Limit 40[%]

In this conditions, if you change the Permanent Torque Limit value from 50[%] to 80[%] without changing the Dynamic Torque Limit value, the combined limit value will change from 20[%] to 32[%]. In order to keep the combined limit value 20[%], change the Dynamic Torque Limit value to 25[%] at the dynamic torque control setting screen.

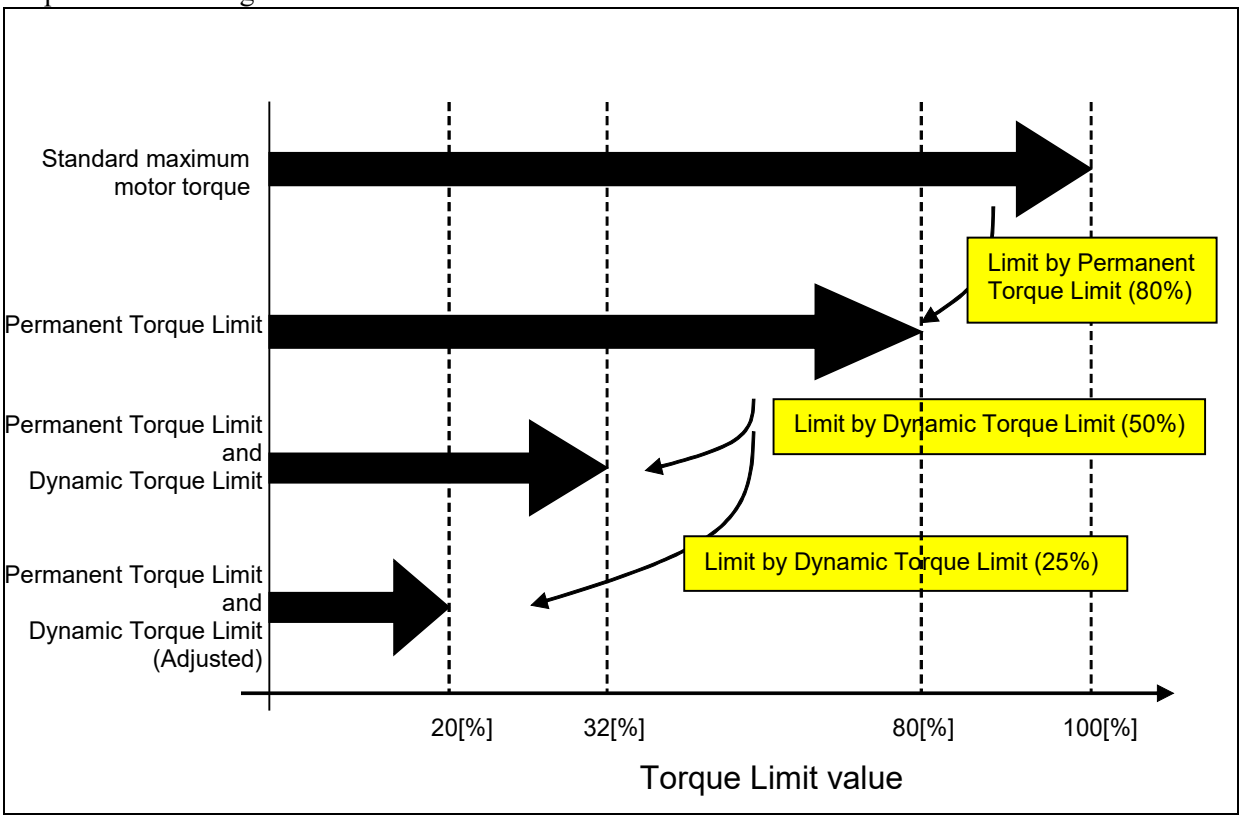


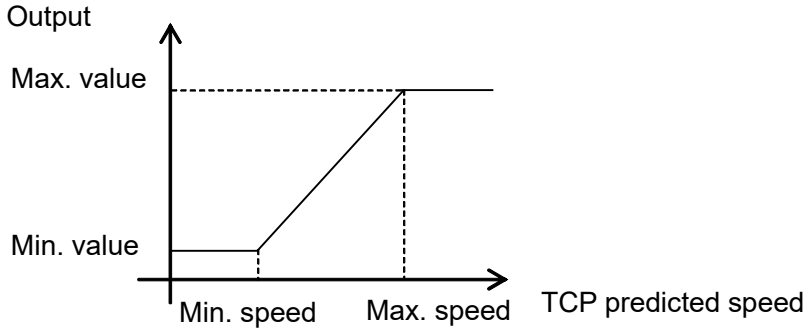
Fig. 27.2.3 (b) Adjustment of Dynamic Torque Limit after changing Permanent Torque Limit

28 TCP SPEED OUTPUT

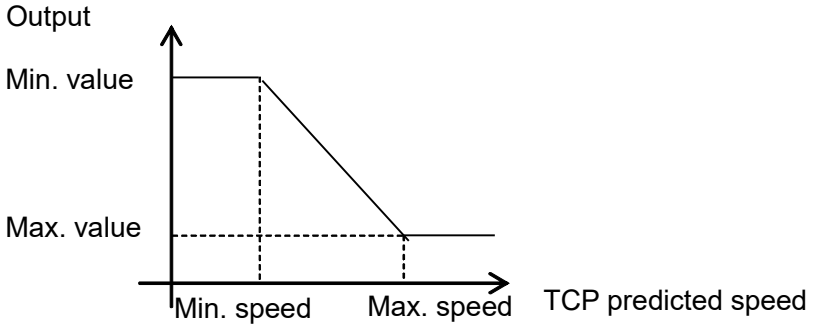
The TCP Speed Output function outputs the predicted speed of the tool center point (TCP) via analog output signals, group output signals, or registers. The TCP Speed Output software option is required to use this function. The TCP predicted speed is converted based on the output conditions defined by the following setup items:

- Minimum value
- Maximum value
- Minimum speed (mm/sec)
- Maximum speed (mm/sec)

The following figure shows the relationship between TCP predicted speed and output values.



The Min value is output when TCP predicted speed is the specified minimum speed or lower. The Max value is output when TCP predicted speed is the specified maximum speed or higher. The following figure shows the relationship between the predicted speed and the output value when you set the Max value to a value less than the Min value.



28.1 LIMITATIONS

The TCP Speed Output function has the following limitations:

- It cannot be used with DispenseTool or Dispense-enabled SpotTool+.
- It cannot be used with tracking functions such as Line Tracking.
- It cannot be used in T1 mode.
- The TCP predicted speed is output only while the robot is moving during program execution.
- The TCP predicted speed only considers group 1 motion.
- The TCP predicted speed does not consider the extended axis motion except for integrated extended axis.
- The accuracy of the TCP predicted speed might decrease when motor speed limit alarms are posted.
- The accuracy of the TCP predicted speed might decrease when the frame is being changed.

- The accuracy of the TCP predicted speed might decrease when the motion is being changed between normal and remote TCP motion.
- The accuracy of the TCP predicted speed might decrease when one of the following instructions is being executed:
  WAIT instruction
  Joint motion instruction
  Motion instruction with Approach/Retract (Linear Distance) instruction
  Motion instruction with Process Speed instruction
  Motion instruction with Max Speed instruction
  Motion control statement in a KAREL program
- Up to 10 output conditions can be defined.
- Only one prediction time can be specified. You cannot set a different prediction time for each output condition.

28.2 SETTING UP TCP SPEED OUTPUT

To use the TCP Speed Output function, you must set up the output conditions using Procedure 28-1.

Procedure 28-1 Setting up TCP speed output conditions

Step

1 Press MENU.
2 Select I/O.
3 Press F1, [TYPE].
4 Select TCP Speed. The following screen will be displayed.

```
TCP speed output
    List                                    1/10
      Enable  Target     Min (Val/Spd) Max
      1 OFF AO [    0]    0/   0      0/   0
      2 OFF AO [    0]    0/   0      0/   0
      3 OFF AO [    0]    0/   0      0/   0
      4 OFF AO [    0]    0/   0      0/   0
      5 OFF AO [    0]    0/   0      0/   0
      6 OFF AO [    0]    0/   0      0/   0
      7 OFF AO [    0]    0/   0      0/   0
      8 OFF AO [    0]    0/   0      0/   0
      9 OFF AO [    0]    0/   0      0/   0
     10 OFF AO [    0]    0/   0      0/   0

|  | [ TYPE ] | SETUP | DETAIL | ON | OFF |  |
```

5 Press F2, SETUP. The following screen will be displayed.

```
TCP speed output
    Setup                                   1/1
      1  Prediction time (msec) :     0

|  | [ TYPE ] | LIST |  |  |  |  |
```

6 Set the value for Prediction time.
7 Press F2, LIST. The following screen will be displayed.

```
TCP speed output
    List                                    1/10
      Enable  Target     Min (Val/Spd) Max
      1 OFF AO [    0]    0/   0      0/   0
      2 OFF AO [    0]    0/   0      0/   0
      3 OFF AO [    0]    0/   0      0/   0
      4 OFF AO [    0]    0/   0      0/   0
      5 OFF AO [    0]    0/   0      0/   0
      6 OFF AO [    0]    0/   0      0/   0
      7 OFF AO [    0]    0/   0      0/   0
      8 OFF AO [    0]    0/   0      0/   0
      9 OFF AO [    0]    0/   0      0/   0
     10 OFF AO [    0]    0/   0      0/   0

 | [ TYPE ] | SETUP | DETAIL | ON | OFF |
```

NOTE
Each line in this screen indicates the status of one output condition.

8 Move the cursor to the output condition you want to set up, and press F3, DETAIL. The following screen will be displayed.

```
TCP speed output
    TCP_SPD[1]                              1/6
      1 Enable:                  OFF
      2 Target:                AO [    0]
      3 Min. value:              0.000
      4 Max. value:              0.000
      5 Min. speed (mm/sec):     0.000
      6 Max. speed (mm/sec):     0.000

 | [ TYPE ] |  | LIST | ON | OFF |
```

9 Move the cursor to Target and select the output target.
10 Move the cursor to each item and set the value.
11 To start the output, set Enable to ON.

NOTE
You can also set Enable and Target using an TP instruction. Refer to Procedure 28-2 for more information.

12 Press F3, LIST. The following screen will be displayed.

```
TCP speed output
    List                                    1/10
      Enable  Target     Min (Val/Spd) Max
      1 ON  AO [    0]  100/  10  4095/2000
      2 OFF AO [    0]    0/   0      0/   0
      3 OFF AO [    0]    0/   0      0/   0
      4 OFF AO [    0]    0/   0      0/   0
      5 OFF AO [    0]    0/   0      0/   0
      6 OFF AO [    0]    0/   0      0/   0
      7 OFF AO [    0]    0/   0      0/   0
      8 OFF AO [    0]    0/   0      0/   0
      9 OFF AO [    0]    0/   0      0/   0
     10 OFF AO [    0]    0/   0      0/   0

 | [ TYPE ] | SETUP | DETAIL | ON | OFF |
```

NOTE
You can set each output condition on this screen to Enable.

13 Each setting items in the DETAIL of TCP speed output can also be configured with the following Data ID.

Table 28.1(a) Data ID of TCP speed output

Item	Corresponding Data ID
Enable	$TCP_SPEED_OUTPUT.TCP_SPEED_OUTPUT_ENABLE[a] / BOOLEAN / RW / a: group number
Type of Target	$TCP_SPEED_OUTPUT.TARGET_TYPE[a] / INTEGER / RW / a: Type of output / Data ID 1 corresponds to AO, 2 to GO, 3 to R(register)
Index of Target	$TCP_SPEED_OUTPUT.TARGET_INDEX[a] / INTEGER / RW / a: Index number of output
Min. value	$TCP_SPEED_OUTPUT.MIN_VALUE[a] / REAL / RW / a: group number
Max. value	$TCP_SPEED_OUTPUT.MAX_VALUE[a] / REAL / RW / a: group number
Min. speed (mm/sec)	$TCP_SPEED_OUTPUT.MIN_SPEED[a] / REAL / RW / a: group number
Max. speed (mm/sec)	$TCP_SPEED_OUTPUT.MAX_SPEED[a] / REAL / RW / a: group number

28.3 TCP SPEED OUTPUT INSTRUCTION

You can also set up the TCP Speed Output function using TP instructions.

Use the following instruction to start the output:

Target type[number]=(TCP_SPD[output condition number])

Use the following instruction to terminate the output:

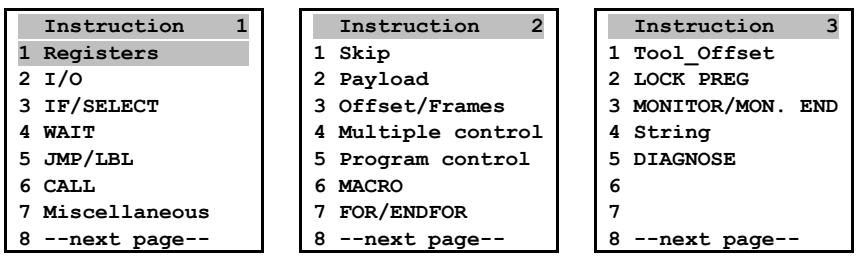
Target type[number]=constant value

Procedure 28-2 shows how to insert a TCP Speed Output instruction for a register type target.

Procedure 28-2 Inserting TCP speed output instruction

Step

1 Display the program edit screen.
2 Press NEXT- key.
3 Press F1, [INST]. The following menus will be displayed.



4 Select Registers. The following menu will be displayed.

```
REGISTER statement 1
1 ...=...
2 ...=...+...
3 ...=...-...
4 ...=...*...
5 ...=.../...
6 ...=...DIV...
7 ...=...MOD...
8 ...=(...)
```

5 Select …=(...). The following menus will be displayed.

```
  Mixed Logic   1
1 DO[  ]
2 R[  ]
3 F[  ]
4 GO[  ]
5 SO[  ]
6 AO[  ]
7 Parameter name
8 --next page--
```

```
  Mixed Logic   2
1 PR[i,j]
2 UO[  ]
3 RO[  ]
4 TIMER[  ]
5
6
7
8 --next page--
```

6 Select R[ ].

```
                                                    1/2
   1:   R[...]=(...)
[End]

|  |  | DIRECT | INDIRECT | [CHOICE] | [LIST] |  |
```

7 Set the register number for the output. The following menus are displayed.

```
  Mixed Logic  1
1 (
2 DI[  ]
3 DO[  ]
4 R[  ]
5 F[  ]
6 On
7 Off
8 --next page--
```

```
  Mixed Logic  2
1 Constant
2 GI[  ]
3 GO[  ]
4 SI[  ]
5 SO[  ]
6 AI[  ]
7 AO[  ]
8 --next page--
```

```
  Mixed Logic  3
1 Parameter name
2 AR[  ]
3 TIMER[  ]
4 TIMER_OVERFLOW
5 PR[  ]
6 UI[  ]
7 UO[  ]
8 --next page--
```

```
  Mixed Logic  4
1 RI[  ]
2 RO[  ]
3 TCP_SPD[  ]
4
5
6
7
8 --next page--
```

8 Select TCP_SPD[…].

```
                                                    1/2
   1:   R[1]=(TCP_SPD[...])
[End]

|  | <INSERT> | DIRECT | INDIRECT | [CHOICE] |  |  |
```

9 Set the output condition number.

```
                                                    1/2
   1:   R[1]=(TCP_SPD[1])
[End]

|  | <INSERT> |  |  |  |  |  |
```

29 TP DRAM/FILE STORAGE FUNCTION

TP DRAM/FILE Storage function enhances the capacity of program space by providing for storages outside of CMOS. In 1 motion group system, this function extends the available program space from 15000 points now to 100000 points. However, the maximum points will be fewer than 100000 points depending on system configuration. “TP DRAM/FILE Storage” option is necessary to use this function.

29.1 STORAGES

There are 4 kinds of storages, SHADOW, SHADOW ONDEMAND, FILE, and the existing storage CMOS. The “temporary” storage states mean the transition state when a program is moved between memories for edit or load of a program. The following tables are for the storages and its description.

Table 29.1(a) Storages

Storage	Description
CMOS	Stored into CMOS (non-volatile memory) in the same way as normal controller.
SHADOW	Program is in FROM (non-volatile memory). Loaded into DRAM (volatile memory) on startup.
SHADOW ONDEMAND	Program is in FROM. Loaded into DRAM when accessed.
FILE	Program is in a TP file in the directory specified in “File Path” in Program Configuration screen. Loaded into DRAM when accessed.

Table 29.1(b) Temporary storage status

“temporary” storage state	Description
SHAD (DRAM) ONDE	SHADOW ONDEMAND program is loaded into DRAM.
FILE (DRAM)	FILE program is loaded into DRAM.
SHAD (CMOS)	SHADOW program moved to CMOS for editing.
SHAD (CMOS) ONDE	SHADOW ONDEMAND program moved to CMOS for editing.
FILE (CMOS)	FILE program moved to CMOS for editing.

29.1.1 CMOS Programs

CMOS programs use CMOS (non-volatile memory) as the storage memory.
The programs in other storage are moved to CMOS at editing program temporarily.
The largest program size is dependent on CMOS size.

29.1.2 SHADOW Programs

SHADOW programs use FROM and DRAM as storage memory. As the size of DRAM is larger than that of CMOS, more programs than before will be available by using SHADOW. The capacity for programs that can be used for SHADOW is dependent on the rest capacity of FROM and DRAM.
A program stored in SHADOW is loaded from FROM into DRAM on start up. A program will be executable by loaded into DRAM.
The start up time will be long if there are too many SHADOW programs.
When SHADOW program is edited, it is moved to CMOS and to be “temporary” storage state SHAD (CMOS).

29.1.3 SHADOW ONDEMAND Programs

SHADOW ONDEMAND programs use FROM and DRAM as storage memory. SHADOW ONDEMAND is different form SHADOW in that SHADOW ONDEMAND is loaded when a program is accessed for edit or execution of the program. So a SHADOW ONDEMAND program has no effect to start up time.
The capacity of programs that can be used for SHADOW ONDEMAND is also dependent on the rest capacity of FROM and DRAM as with SHADOW.
When a SHADOW ONDEMAND program is loaded, its storage is moved to “temporary” storage state, SHAD(DRAM) ONDE. There is a slight time lag in the first access, but after that there is no time lag because SHAD(DRAM) ONDE, once loaded into DRAM, remains in DRAM until the power is off.
When a SHADOW ONDEMAND program is edited, its storage is moved to CMOS and to be “temporary” storage state SHAD (CMOS) ONDE.

29.1.4 FILE Programs

FILE programs use storage devices such as memory card as storage memory. In FILE storage, much more programs than in SHADOW can be available by using external storage devices that have large capacity. A FILE program is loaded into DRAM from external storage devices when first accessed as with SHADOW ONDEMAND.
The path to external storage device is specified in “File Path” in Program Configuration screen.
When a FILE program is loaded, its storage is moved into “temporary” storage state, FILE (DRAM).
The time lag in the first access is longer than that of SHADOW ONDEMAND programs, but after that there is no time lag as with SHADOW ONDEMAND because FILE (DRAM) remains in DRAM until the power is off.
When a FILE program is edited, its storage is moved to “temporary” storage state, FILE (CMOS).

NOTE
Use a dedicated directory in external device for FILE storage path. Do not use shared directly to save TP program files in other storage and backup files.

29.2 STORAGE CONFIGURATION

Procedure 29-1 Confirm storage device

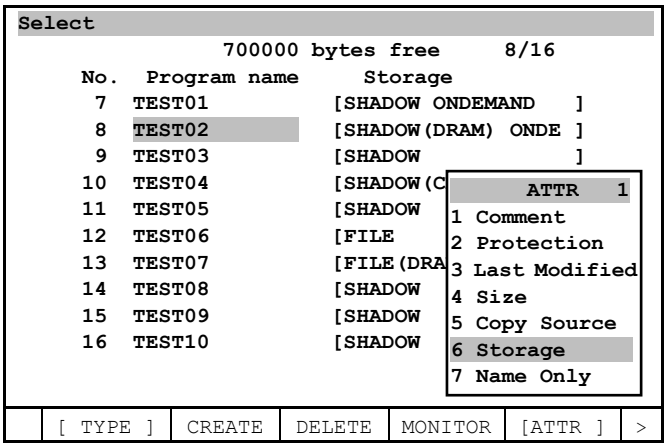
Step

1 Press the [SELECT] key. The following screen will be displayed. “Not Loaded” in the comment means that a SHADOW ONDEMAD or FILE program is not loaded into DRAM.

```
Select
                  700000 bytes free     8/16
     No.  Program name        Comment
      7   TEST01           [Not Loaded        ]
      8   TEST02           [                  ]
      9   TEST03           [                  ]
     10   TEST04           [                  ]
     11   TEST05           [                  ]
     12   TEST06           [Not Loaded        ]
     13   TEST07           [                  ]
     14   TEST08           [                  ]
     15   TEST09           [                  ]
     16   TEST10           [                  ]

    | [ TYPE ] | CREATE | DELETE | MONITOR | [ATTR ] | >
```

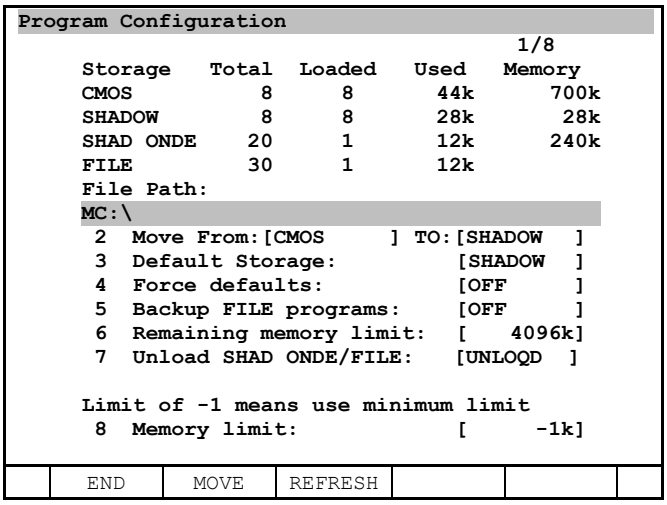
2 Press the F5, [ATTR], and select “Storage”. The following screen will be displayed.



Procedure 29-2 Display Program Configuration screen

Step

1 Press the [SELECT] key.
2 Press the F1, [TYPE], and select “Config…”.
3 The following screen will be displayed.



The table below describes the each item in Program Configuration screen.

Table 29.2 Program configuration screen

Item	Description
Storage	For the 4 storages, the number of programs and memory usage are displayed. Total : The number of programs that exist in the storage. Loaded: The number of programs that are actually loaded in the storage. Used : The memory usage for loaded programs. Memory: For CMOS, the capacity for programs is displayed. For others, the total size of the programs that exist in the storage.
File Path	Specify the device where FILE programs are stored.
Move From To	To and from locations for moving programs between storage types.
Default Storage	Set the storage for the new programs. Select from CMOS, SHADOW and SHADOW ONDEMAND. FILE storage cannot be selected.
Force defaults	If valid, loaded into the storage specified in the “Default Storage”. If invalid, loaded into the storage specified in the program file.
Backup FILE programs	Exclude or include FILE program types in backup.

Item	Description
Remaining memory limit	If remaining DRAM goes lower than the value set in “Remaining memory limit” in some operations such as creating a new program, an alarm occurs. Remained DRAM memory can be confirmed as temporary memory in Memory screen. You can set the value from 1024 to 10000.The default value is 4096. Usually, please do not change this value.
Unload SHAD ONDE/FILE	Memory of all SHADOW ONDEMAND and FILE programs that loaded to DRAM is released. This function is used to load other programs when DRAM memory is low. Move cursor to this item and push F5 [UNLOAD] to execute this function. These programs are deleted temporarily and they are loaded to DRAM again when they are selected. This function can be used after 7dc2 software.
Memory limit	This item decides the total memory size of SHADOW, SHADOW ONDEMAND and FILE programs that can be loaded into DRAM. If the total memory size goes over the setting value, an alarm occurs. You can set the value from –1 to 10000. The default value is –1. So there is no limitation. Usually, please do not change this value.
END	Finish Program Configuration screen and go back to Select screen.
MOVE	Move programs at a time according to the storage types set in “Move From To”.
REFRESH	Update FILE program list according to TP files in the directory specified in “File Path”. After this operation, there are updated FILE programs displayed in Select screen.

NOTE
The usage of TEMP is different from that displayed in Program Configuration screen. When you load a program into DRAM, it uses three times as much TEMP as the value of the memory displayed in Program Configuration screen.

NOTE
When Default Storage is SHADOW, storage of new programs is set SHADOW. But storage of program could be set CMOS automatically because of shortage of DRAM memory. In this case, please delete unnecessary programs in DRAM. And please recreate the program or change storage individually.

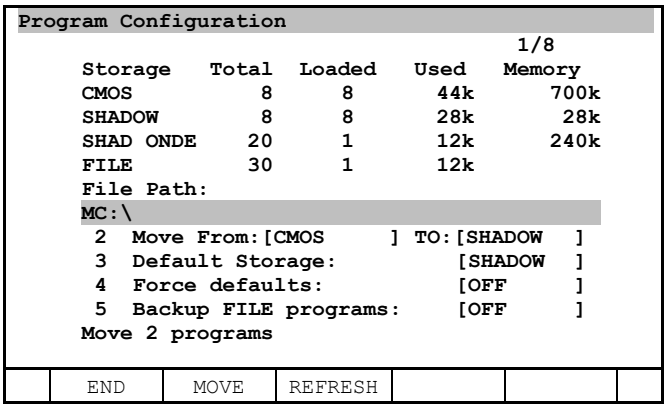
NOTE
These settings are reflected soon after changed.

Procedure 29-3 Move Storage

Step

- You can move all programs in the same storage to other storage at a time by using “MOVE” in Program Configuration screen.

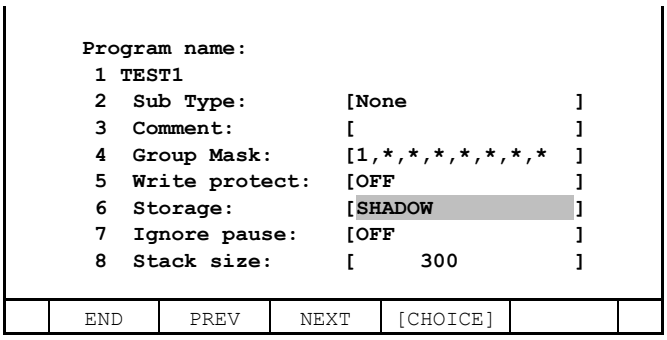
1 In the item “Move From To”, select the original storage in “From”, and select the new storage in “To”.
2 Press the F2, MOVE.
3 Press the F4, YES.
4 The following screen will be displayed and moving the storage finished.



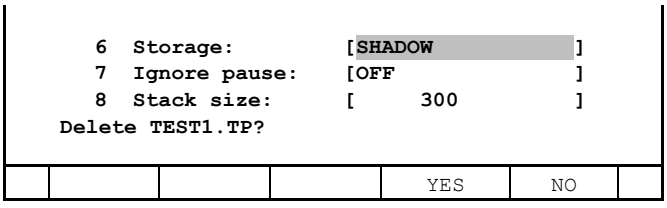
NOTE
Please confirm that there is enough memory space available in the new storage. If not, an alarm occurs and stops moving. Then, programs that are already moved do not go back the original storage. When you move large number of programs, please make backup in advance.

- You can also change the storage of a program individually.

1 Press the [SELECT] key and press the [NEXT] key.
2 Press the F2, DETAIL and the following screen will be displayed.
3 Move the cursor on Storage and press the [ENTER] key, and select the storage.



4 When the storage is changed from FILE, “Delete <file name>?” will be displayed.
If the F4 YES is selected, program file on file path is deleted.
If the F5 NO is selected, program file on file path is remained.



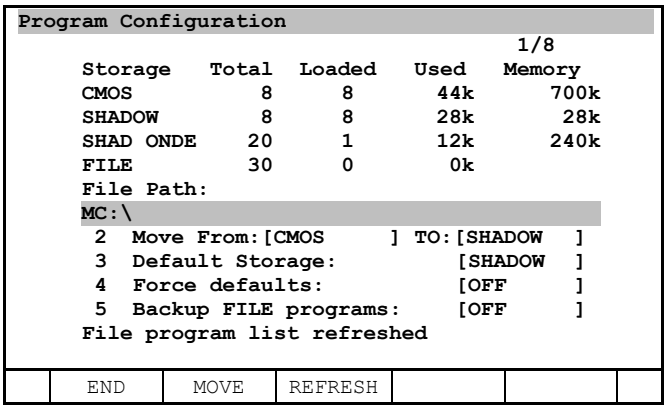
NOTE
Sometimes, the storage goes back to original storage soon after you change the storage. In such case, please confirm if the program is in edit, there is enough memory space in the new storage, or “File Path” is specified if FILE.

Procedure 29-4 Update FILE program list

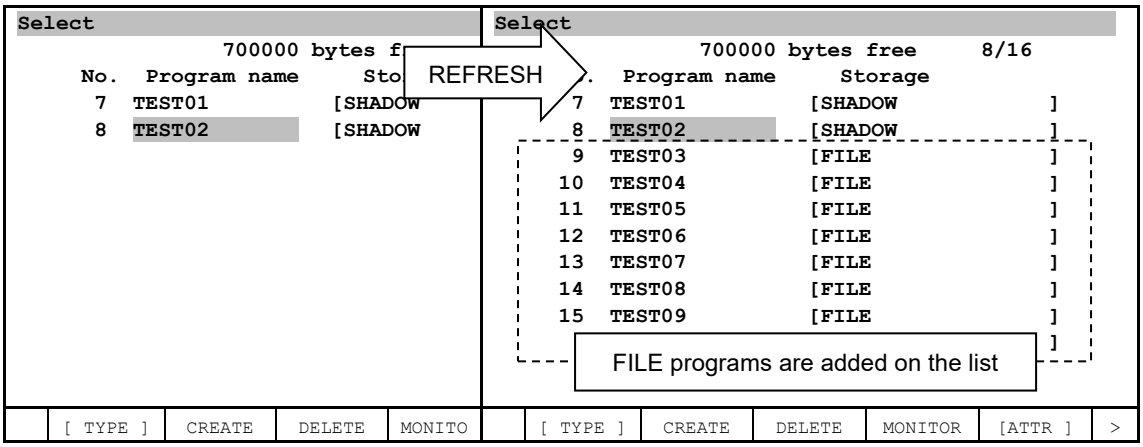
Step

- You can update the FILE program list. This operation is used to display FILE programs again or to use programs in other external storage devices.

1 Input the path of the device where FILE programs are stored, and press the [ENTER] key.
2 Press the F3, REFRESH.
3 Such screen as follows is displayed, and FILE program list is updated.



Updated FILE programs are listed in Select screen.



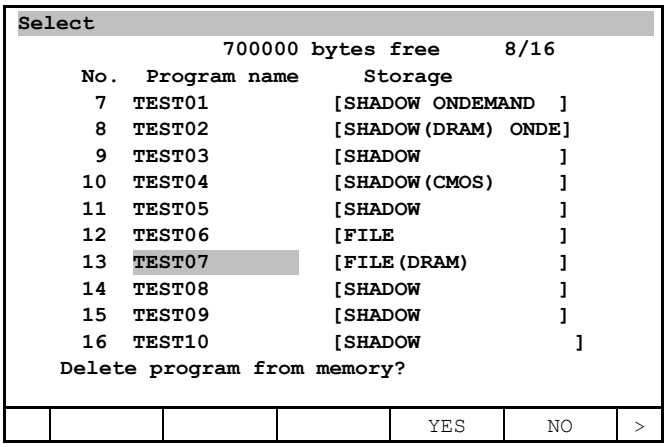
Procedure 29-5 Delete FILE, SHADOW ONDEMAND program

Step

- User programs can be deleted in Select screen. When operation to delete CMOS and SHADOW programs is executed, all program date is deleted.
  For example, when “SHADOW (CMOS)” program is deleted in Select screen, program data in CMOS and DRAM and program in FROM are deleted.
- For FILE and SHADOW ONDEMAND programs, you can choice to delete all program data or only temporary program data, in CMOS and DRAM.

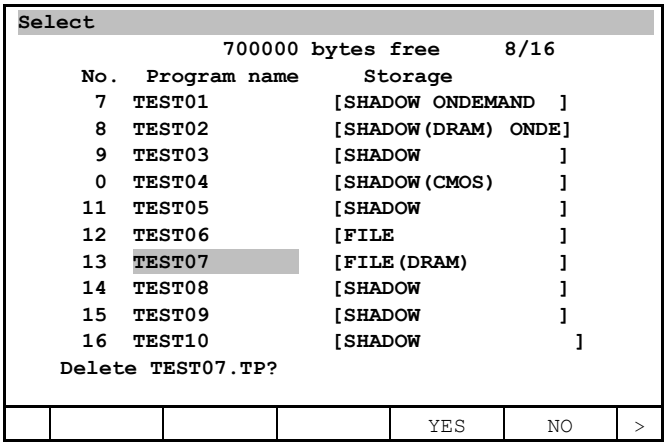
Please refer to the following process to delete FILE and SHADOW ONDEMAND program.

1 Move cursor to the program and press the F3 DELETE.
2 When program is loaded to DRAM, “Delete program from memory?” will be displayed.



3 If F4 YES is selected, program data in DRAM is deleted.
If F5 NO is selected, operation is terminated and program is not deleted.

4 If F4 YES is selected, following message is displayed continually.
For FILE programs, “Delete <file name>?” is displayed.
For SHDOW ONDEMAND programs, “Delete permanently?” is displayed.



5 If F4 YES is selected, list of the programs in Select screen are also deleted.
For FILE programs, program files on file path are also deleted.
For SHDOW ONDEMAND programs, programs in FROM are also deleted.
If F5 NO is selected, list of the programs in Select screen are remained. And program files on file path or programs in FROM are not deleted. When the programs are accessed, they can be loaded into DRAM again because programs on file path or in DRAM are remained.

29.3 SAVE / LOAD PROGRAMS

29.3.1 Save / Load TP Files

When a program is saved as TP file, information for storage is also preserved in it. So, when a TP file is loaded, the program is stored in the saved storage according to the information.
When “Force defaults” is “ON” in Program Configuration screen, the program is stored in the storage specified in “Default Storage”.
When the existing TP file that has no information for storage is loaded, the program is stored in the storage specified in “Default Storage”.

29.3.2 Copy Programs

When a program is copied, the storage of copied program is the same as that of original program.
However, when “Force defaults” is “ON” in Program Configuration screen, copied program is stored in the storage specified in “Default Storage”.

29.3.3 Save / Load LS Files

When a program is saved in ASCII format, a LS file is created. Then, new item “STORAGE” is added in the LS file. The storage of the program is printed in this item. When this LS file is loaded, the program is stored in the storage specified in the “STORAGE” item.
When a existing LS file that does not have the item “STORAGE” is loaded, the program is stored in the storage specified in “Default Storage”.

NOTE
”ASCII UPLOAD” option is necessary to load LS files.

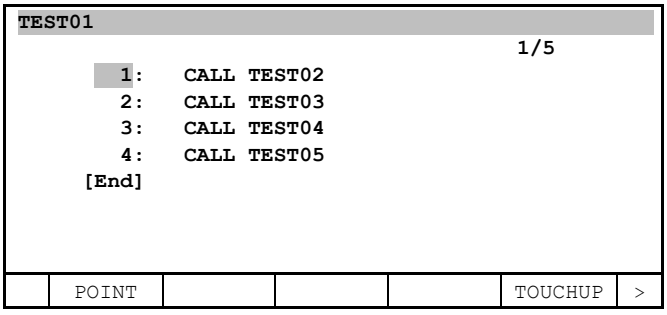
NOTE
”R709 TP DRAM/FILE Storage” option is necessary to load LS files that include “STORAGE” item. To load LS files into the controller without this option, you need to delete “STORAGE” item in advance.

29.3.4 Make Backup of Programs

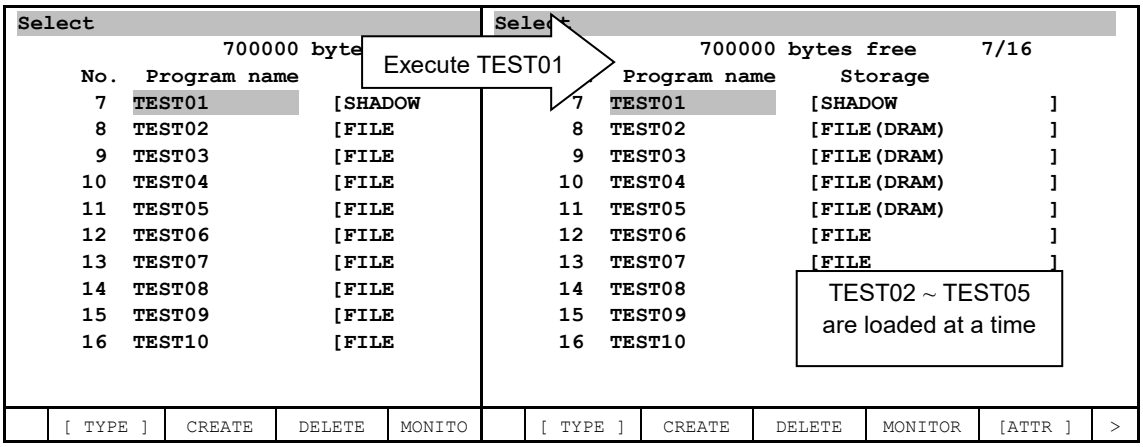
In File screen, you can make backup of SHADOW and SHADOW ONDEMAND programs as well as CMOS programs.
However, when you make backup of FILE programs, you need to set “Backup FILE programs” “ON” in Program Configuration screen.

29.4 LOADING PROCESS IN PROGRAM EXECUTION

The programs that are not loaded on startup such as SHADOW ONDEMAND and FILE programs have time lag until they are loaded, because they are loaded on the start of edit or execution.
For example, SHADOW program TEST01 execute FILE programs TEST02~TEST05 by using CALL instruction.

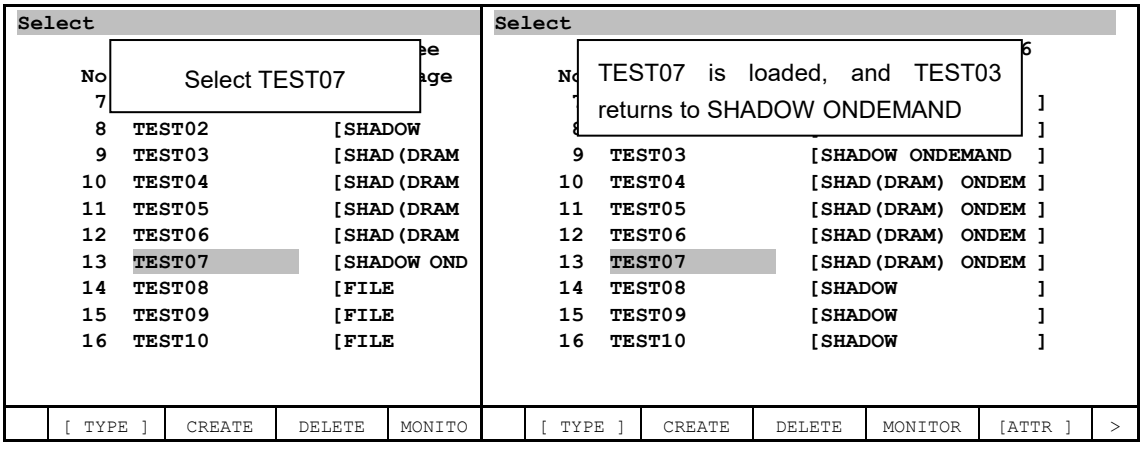


When TEST01 is executed, all the programs (TEST02~TEST05) that are called from TEST001 are loaded into DRAM.
If the size of a FILE program is large, it takes time to load it. So there is a time lag.
If these FILE programs are loaded in advance, TEST01 will start immediately.



29.5 PROGRAM EXCHANGE FUNCTION WITHOUT ENOUGH MEMORY SPACE

An error usually occurs when you load SHADOW ONDEMAND or FILE programs without enough memory space in DRAM. This function replaces old programs with newly loaded program in the same storage automatically.
Until there is enough memory space in DRAM for the newly loaded program, old programs are deleted from DRAM in order automatically.
Old programs are deleted, but they still exist in the original storage. So they can be loaded again.
The following figure indicates the behavior of loading a new SHADOW ONDEMAND program, TEST07 when there is not enough memory space in DRAM. When TEST07 is selected, TEST03 moves back to SHADOW ONDEMAND and TEST07 can move into the ”temporary” storage state SHAD(DRAM) ONDE.
So, there is a time lag when you load TEST03 again.
You can use this function by setting the item “Program exchange when lack memory”(MENU → SYSTEM → CONFIG → TP DRAM/File Storage → Program exchange when lack memory) to TRUE. By default setting is FALSE and this function is invalid.



29.6 PRECAUTION

29.6.1 Cause and Remedy for Alarm Occurrence

Cause and remedy for alarms that occur when you use this function are listed.

TPIF-036 Not enough memory

Cause:

1 Performed such operations as creating a new program, edit, or copy a program without enough memory space in DRAM and CMOS.
2 Performed such operations as creating, editing, or copying a SHADOW, SHADOW ONDEMAND or FILE program in the following situation.
"Memory limit" is valid and the memory usage exceeds the limit.
There is not enough memory space in CMOS.

Remedy:

1 Delete unnecessary programs using CMOS and DRAM, and make enough memory space in CMOS and DRAM.
2 Set larger value in "Memory limit" or set the limit invalid. Delete unnecessary programs using CMOS and make enough memory space in CMOS.

MEMO-126 No more available TPP memory

Cause:

1 Performed such operations as editing, copying a program or creating a new CMOS program without enough memory space in CMOS.

Remedy:

1 Delete unnecessary programs using CMOS and make enough memory space in CMOS.

MEMO-178 No more available DRAM (SHADOW/FILE)

Cause:

1 Performed such operations as creating a new program, edit, or copy a program without enough memory space in DRAM.
2 Performed such operations as creating, editing, or copying a SHADOW, SHADOW ONDEMAND or FILE program in the following situation.
"Memory limit" is valid and the memory usage exceeds the limit.

Remedy:

1 Delete unnecessary programs using DRAM and make enough memory space in DRAM.
2 Set larger value in "Memory limit" or set the limit invalid. Delete unnecessary programs using DRAM and make enough memory space in DRAM.

TPIF-239 %s loaded to CMOS

Cause:

Performed such operations as loading a TP or LS program file of SHADOW, SHADOW ONDEMAND or FILE program in FILE screen in the following either situation.
1 Memory space in DRAM is not enough.
2 "Memory limit" is valid and the memory usage exceeds the limit.

Remedy:

1 Delete unnecessary programs using DRAM and make enough memory space in DRAM.
2 Set larger value in "Memory limit" or set the limit invalid. Delete unnecessary programs using DRAN and make enough memory space in DRAM.

After these operations, change storage individually form CMOS or load the program file again.

NOTE
When this alarm occurs, the program is loaded to CMOS.

30 CYCLE TIME LOGGING

Cycle Time Logging function records and displays the cycle time of programs. Cycle times are recorded based on individual cycles and hourly averages. Multiple cycles can be observed to view trends and analyze data. The cycle time data is displayed graphically and raw cycle data in text format can also be displayed. By default, the cycle time is tracked from the start of a teach pendant program until the program ends. This is optional function.

Cycle Time Categories

Cycle Time is recorded based on multiple categories. The following table describes each category.

Table 30(a) Cycle time categories

Category	Description
MOTION	The amount of motion time in the cycle.
PROCESS	The amount of process time including welding time.
APPL	The amount of application-specific time executed such as SPOT instruction or ARC instruction.
WAIT	The amount of time spent waiting for I/O or a data event. Condition specified WAIT instruction is recorded.
DELAY	Time specified WAIT instruction is recorded.
KAREL	The amount of time spent executing KAREL logic.
LOGIC	The amount of time spent executing instructions except categories above (ex: Label, I/O).
HOMIO	The amount of time spent executing HOME_IO program. HOME_IO is automatically executed when the robot reaches the home position set in Reference Position screen and the program ran by SOP, UOP, or PROD_START ends. To execute HOME_IO, Reference position should be set correctly. If you do not use HOME_IO, home position setting is not necessary.
IDLE	The time, until the next cycle starts after the robot reaches home position and the program ends, is recorded.
FAULT	The amount of time spent during a program is paused by alarms or some other causes.
TOTAL	The total cycle time of categories above except HOMIO, IDLE and FAULT.

NOTE
The cycle time PROCESS, APPL, HOMIO, and IDLE is recorded only when the program is run by SOP cycle start, UOP_START, or PROD_START.

Cycle Time Screen Items

The following table describes the items displayed in STATUS Cycle Time screen.

Table 30(b) Items in cycle time screen

Item	Description
Program	The name of currently tracked program.
Mode	This item indicates whether the cycle is tracked per cycle or per hour. one, ten, or one hundred cycles can be displayed. It can also display one, ten, one hundred hours.
Date	The date of the currently highlighted cycle or hour recorded.
Time	The time of the currently highlighted cycle was executed.
Range (Total)	This indicates the total number of cycles as n/m where m is the total number of cycles or hours, and n is the currently displayed cycle or hour.
Samples	This item indicates the number of hourly samples used in the currently displayed average of hours.
[PROG]	This function key allows you to change the name of program for which cycle time information is displayed.
[ZOOM]	This function key allows you to change the displayed cycle time information. You can select to display one hour, ten hours, one hundred hours, one cycle, ten cycles, or one hundred cycles.

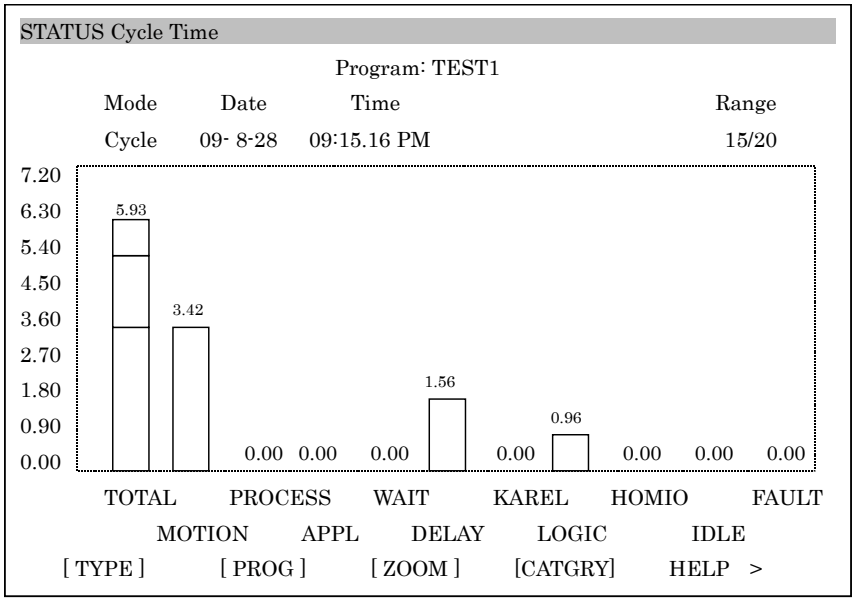
Item	Description
[CATGRY]	This function key allows you to choose which set of cycle time information is displayed. You can choose Total which displays the total of all of the categories using different color for each kind and the cycle time for each individual category. Or, you can choose an individual category to be displayed. When an individual category is displayed, depending on the currently selected mode, the category name will be displayed on the top of right corner, and cycle time information for that category will be only displayed. You can choose to display the following categories: TOTAL, MOTION, PROCESS, APPL, WAIT, DELAY, KAREL, LOGIC, HOMIO, IDLE, FAULT.
[PAGE]	This function key allows you to display cycle time data in a variety of formats. To display cycle time data on the teach pendant, Cycle Graphics should be displayed. Other formats Cycle Text and Average Text are displayed on web server screen. Line Text is only available in line-by-line mode. If you select Cycle Text or Average Text, the .DG files will be displayed. CYCLES.DG displays the cycle data, and CYCAVE.DG displays the hourly data. These files can be displayed using the Web Server on the teach pendant.
[UPDATE]	This function key is used for the 4 operations in the following. Each detail is described in 30.3.3. Load Cycle Data: Load the cycle time data. Clear Cycle Data: Clear the cycle time data. Clear Line Data: Clear the line data in line-by-line mode. Collect Line Data: Get the line data in line-by-line mode.
RECORD	This function key records the specific cycle time of a program. “RECORD” is displayed only in cycle mode when an individual category is selected. Detail is described in 30.3.5.

30.1 DISPLAY MODES OF CYCLE TIME

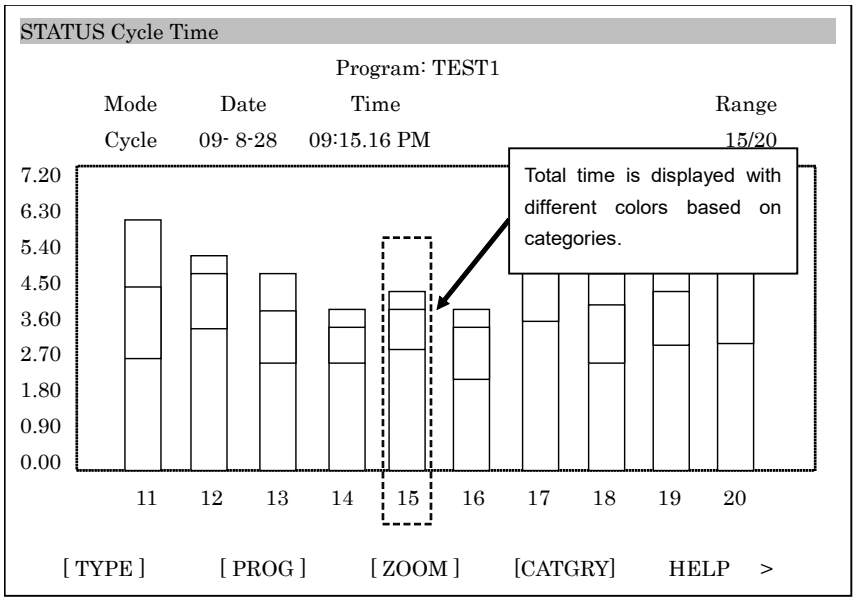
Cycle time data can be displayed in three modes by cycles, by hours, or line-by-line.

30.1.1 Cycle Mode

In cycle mode, cycle time is displayed by cycle. You can display cycle time data by one cycle, ten cycles, or one hundred cycles at a time. The following figure displays the total cycle time for the current cycle in one cycle mode. It also displays the total cycle time broken down by category. Each category is represented by a different color.

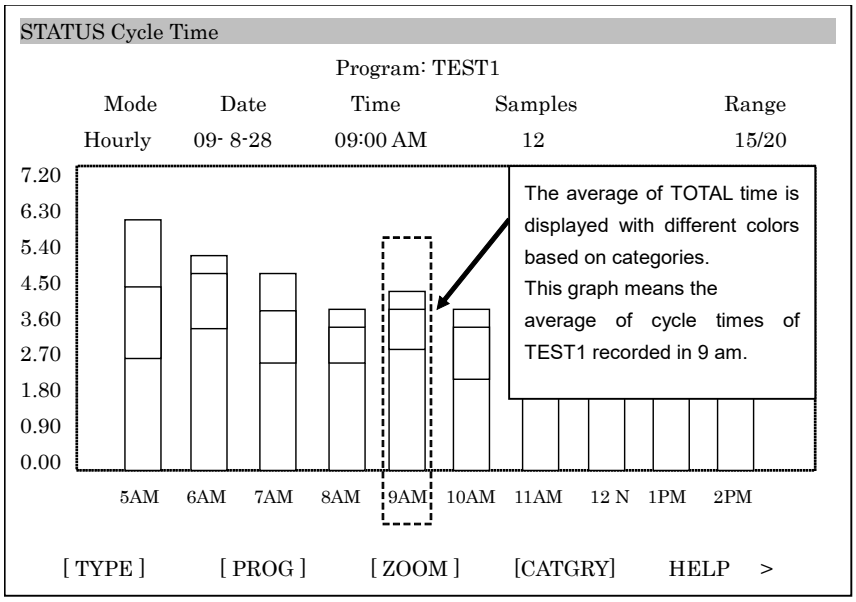


The following figure displays cycle times in ten cycles mode.



30.1.2 Hourly Mode

In hourly mode, the hourly average cycle time is displayed. You can display cycle time data by one hour, ten hours, or one hundred hours at a time. Hourly mode is available only when hourly data exists.
The following figure displays the cycle time in ten hours mode.



30.1.3 Line-by-Line Mode

In line-by-line mode, the cycle time in each line in a program is recorded. So you can see the cycle time of specific line number in the program. The data is saved in CYCLINE.DG in DRAM. So it is cleared when the power is off.

To use line by line mode, “Cycle Time Logging Settings” configuration item in SYSTEM/CONFIG screen should be configured.

- Line-by-Line mode Valid when it is TRUE. (FALSE by default)
- Number of lines Set the number you want to record. (0 by default)

After that,

1 In STATUS Cycle Time screen, press the [NEXT] key, press the F3, [UPDATE] and select Collect Line Data.
2 Execute a program of which you want the cycle time.
3 In STATUS Cycle Time screen, press the [NEXT] key, press the F2, [PAGE] and select Line Text.
4 The cycle time of each line is displayed in the Web Server as follows.

Record	Program	Line	Mncode	Duration-uS
0	TEST1	1	254	592680
1	TEST1	2	123	998858
2	TEST1	3	254	1208431

[TYPE] [PAGE] [HELP]

30.2 LOGGING CYCLE TIME

Procedure 30-1 Logging Cycle Time

Condition

- The program you want to track has been executed at least one time.

Step

1 Press the [MENU] key.
2 Press STATUS.
3 Press the F1, [TYPE].
4 Select Cycle Time.
5 Set the name of the program whose cycle time you want to display. By default, all programs are selected. To select a single program to display, press the F2, [PROG], and select the name of the program.
6 To change the cycle time display between cycles and hours, press the F3, [ZOOM] and select the kind of cycle time you want to display. After you have made a display selection (cycles or hours) you can quickly change the display by using the arrow keys. The up and down arrow keys move from one cycle or hour, to ten cycles or hours, then a hundred cycles or hours. The left and right arrow keys can move forward or backward through the cycles or hours. To move forward and backward quickly, press the [SHIFT] key and the left or right arrow key. If you are in ten cycle mode it will move by 5. If in one hundred cycle mode, it will move by 30.
7 To display cycle time for a specific program category, press the F4, [CATGRY] and select the category you want to display.

30.2.1 Change Display Format of Cycle Time Data

The format of displaying cycle time data can be changed. In STATUS Cycle Time screen, press the [NEXT] key, press F2, [PAGE], and select one from the following items.

- Cycle Graphics — Cycle time data is displayed graphically.
- Cycle Text — Cycle time data is displayed as raw text. The data is saved in CYCLES.DG.
- Average Text — Hourly average cycle time is displayed as raw text. The data is saved in CYCAVE.DG.
- Line Text — Line data of cycle time in line-by-line mode is displayed as raw text. The data is saved in CYCLINE.DG and is cleared with power off. Line text can be displayed only when line-by-line mode is valid.

NOTE
Text data is displayed on Web Server screen.

30.2.2 Save Cycle Time Data

To save the cycle time data, press the [FCTN] key and select SAVE. The cycle time data of cycle mode is saved in PGCYCCYC.TXT and that of hourly mode is saved in PGCYCHR.TXT. They are saved in the device selected in File screen.
F4, [BACKUP] in File screen can be also used.

30.2.3 Update Cycle Time Data

For update of cycle time data, there are 4 operations. In STATUS Cycle Time screen, press the [NEXT] key , press F3, [UPDATE], and select the operation you want to execute. The detail of each operation is described as follows.

- Load Cycle Data
  Clear the currently recorded cycle time data and load the previous cycle time data from the device that is selected in FILE screen.
  Loading PGCYCCY.TXT and PGCYCHR.TXT also can display the previous cycle time data.

- Clear Cycle Data
  Clear the currently recorded cycle time data.

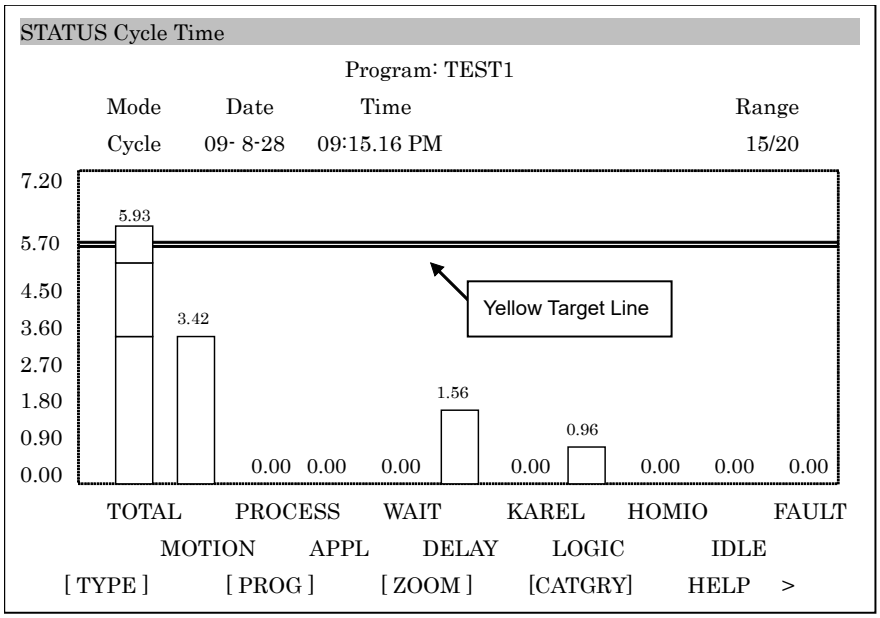
- Clear Line Data
  Clear the data for line by line mode. When you want to get a new line data, you should select Collect Line Data again.

- Collect Line Data
  Collect Line Data is an operation necessary for getting the line data for line-by-line mode. After this operation, execute a program whose cycle time you want to record. Then, in STATUS Cycle Time screen, press the [NEXT] key, press the F2, [PAGE], and select Line Text. The cycle time for each line is displayed on Web Server screen.

30.2.4 Display Target Cycle Time Line

Target cycle time can be displayed as a yellow line.
Target cycle time is displayed only in 1 cycle mode.

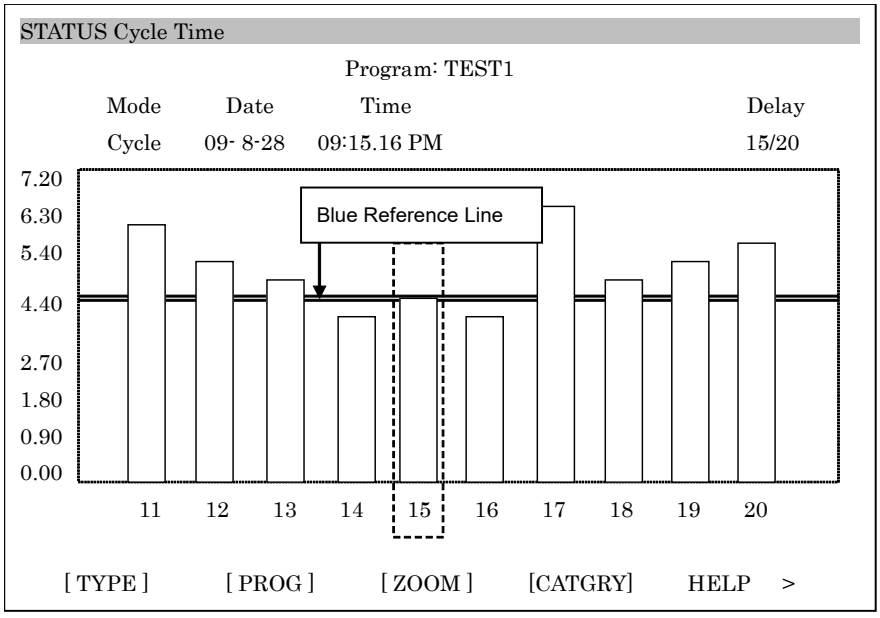
1 Press the [SELECT] key and select a program.
2 Press the [NEXT] key, and press the F2, DETAIL.
3 Press the F3, NEXT until Target Cycle Time (seconds) is displayed.
4 Press the [ENTER] key and type a new target cycle time.
5 Press the F1, END.
6 Press the [MENU] key, select STATUS, press the F1, [TYPE] and select Cycle Time if it is not already displayed.
7 Press the F2, [PROG] and select the program for which target cycle time is set.
8 Press the F3, [ZOOM] and select One Cycle. The following screen will be displayed.



30.2.5 Display Reference Cycle Time Line

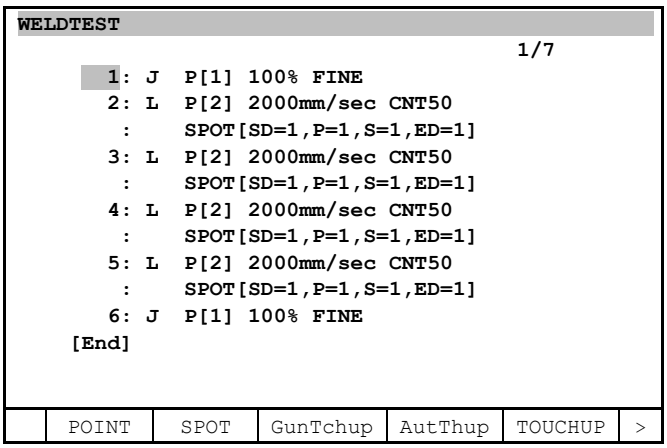
The cycle time of the current program can be recorded and be used as a reference. The recorded cycle time is displayed as a blue line. The reference cycle time line is displayed only in cycle mode and when an individual category is selected.

1 In STATUS Cycle Time screen, press F2, [PROG] and select the program whose cycle time you want to record.
2 Press the [NEXT] key, and confirm that F4, RECORD is displayed and press it.
3 You will be prompted to answer whether or not you want to overwrite. Select YES and the cycle time will be recorded.
4 Press the F4, [CATGRY] and select a category you want to display. DELAY is selected in the following example.
5 The recorded cycle time is displayed as a blue line. (not displayed when TOTAL is selected.)

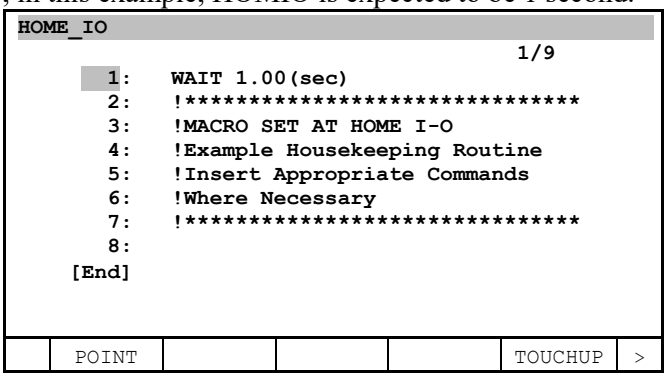


30.3 EXECUTION EXAMPLE OF SPOT PROGRAM

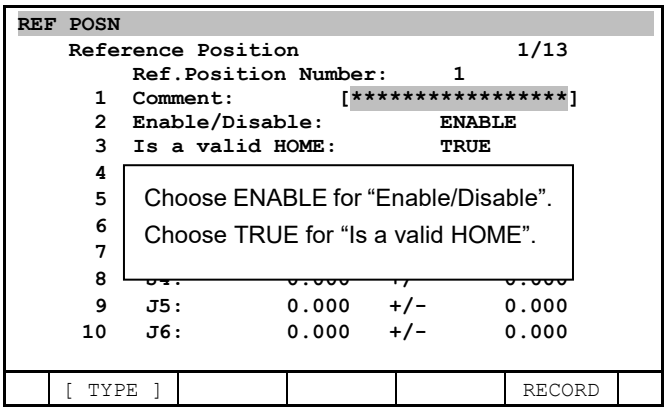
In this section, as an execution example of Cycle Time Logging function, the cycle time is recorded by executing the following Spot program, WELDTEST.
The cycle time of Spot program is not recorded by SHIFT-FWD execution. SOP cycle start, UOP_START, or PROD_START should be used.



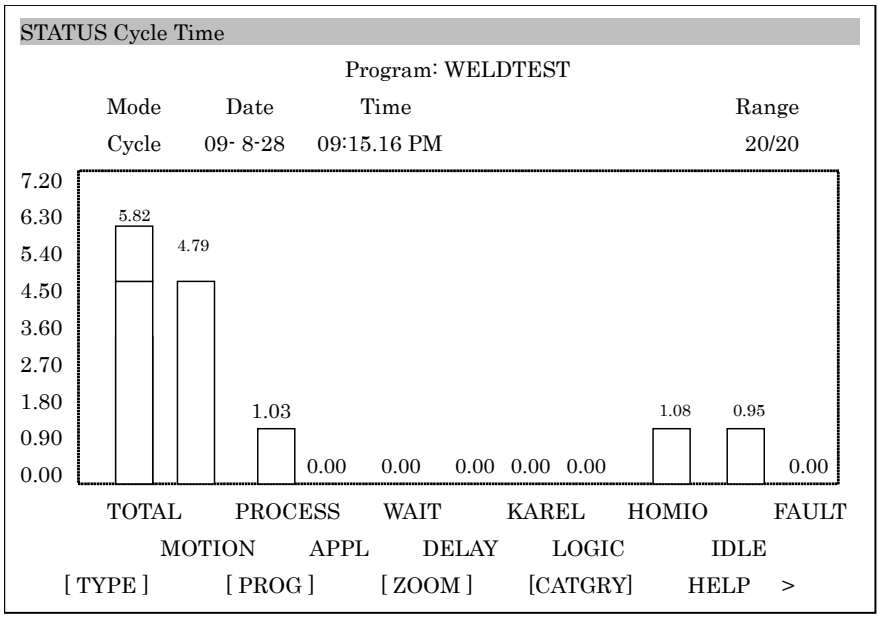
WELDTEST has 4 Spot instructions. In this example, the setting of welding time is 250 ms per one Spot instruction. So the total welding time is expected to 1 second and recorded as PROCESS cycle time.
The following HOME_IO program is also used. It has only one instruction of "WAIT 1.00 (sec)" and other statements are all comment. This is to make HOMIO cycle time easy to estimate. HOME_IO is automatically executed when the robot reaches the home position set in Reference Position screen and the program ran by SOP, UOP, or PROD_START ends. The execution time of HOME_IO is recorded as HOMIO cycle time. So, in this example, HOMIO is expected to be 1 second.



To execute HOME_IO, Reference position should be set correctly.



The following figure is the cycle time of WELDTEST.



As expected, PROCESS cycle time is 1.03 second and HOMIO cycle time is 1.08 second.
Here, IDLE cycle time is recorded with the value 0.95 second. IDLE cycle time records the time that the robot stays idle until next program starts after the program ends. In this example, WELDTEST was restarted soon after it ended. This cycle time data shows that it took 0.95 second to restart WELDTEST after it ended.
The Total cycle time includes MOTION and PROCESS cycle times. HOMIO, IDLE, and FAULT are not included in TOTAL cycle time.

31 MATH FUNCTION INSTRUCTION

- Math function instruction enables to calculate math function (ex: SIN COS and so on) in the TP program.
- Math function instruction can be used in assignment statement, conditional statement and wait command.
- To use this function, Math function option (J593) is required.

31.1 TYPE OF MATH FUNCTIONS

Usable instructions of math functions are as follows.

Table 31.1 Usable instruction of math function

Function	Explanation	Restriction of argument
SQRT[x]	Square root	0 <= x
SIN[x]	Trigonometric function Units are degrees.	None
COS[x]		None
TAN[x]		Except 90,270±360ｎ
ASIN[x]		-1 <= x <= 1
ACOS[x]		-1 <= x <= 1
ATAN[x]		None
ATAN2[x,y]		Except x=0,y=0
LN[x]	Natural Logarithm	0 < x
EXP[x]	Exponent	None
ABS[x]	Absolute	None
TRUNC[x]	Truncate	-2.1*10^9 <= x <=2.1*10^9
ROUND[x]	Round off	-2.1*10^9 <= x <=2.1*10^9

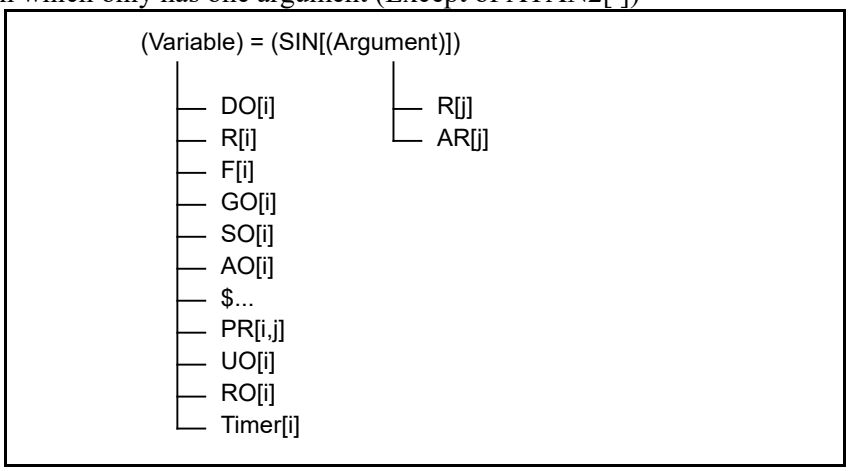
- Units of aforementioned functions are degrees.
- Conversion functions are not supported because the user can achieve them by multiplying a constant value (57.29579 or 0.017453 based on requirement).

31.2 INSTRUCTION FORMAT OF MATH FUNCTION

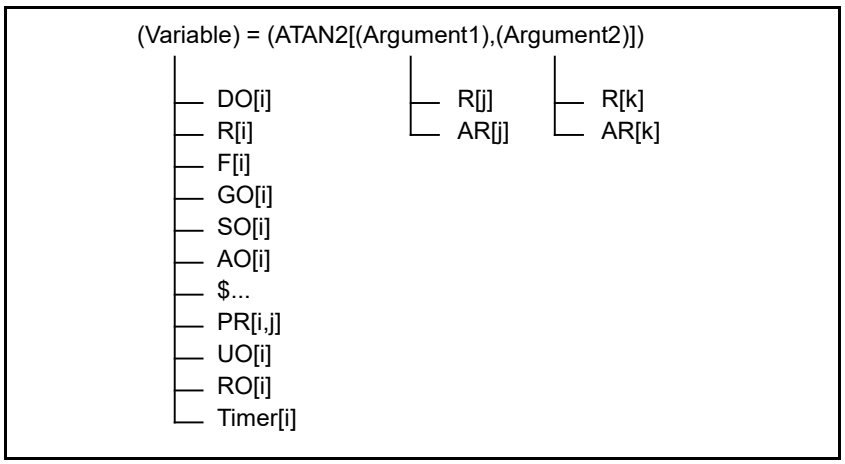
Instruction type of math functions is explained in the following example.

31.2.1 Instruction Format of Assignment Statements

- The function which only has one argument (Except of ATAN2[ ])

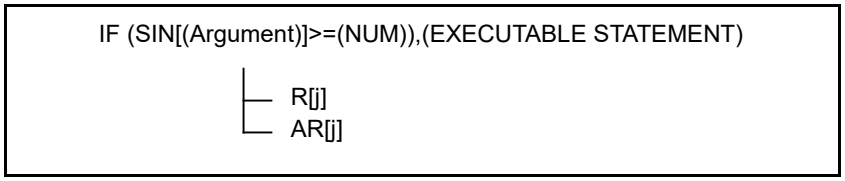


- The function which has two arguments (ATAN2[ ])



- The argument of math functions is “Register “ and “Argument Register”.
- Constant is not used directly.

31.2.2 Instruction Format of Relational Statements



- A mixed logic instruction expression can be used in the conditional statement of a conditional branch command.
- If the result of conditional statement is ON, the executable statement of the conditional branch command is executed.
- The argument of math functions is “Register “ and “Argument Register”.
- Constant is not used directly.

31.2.3 Instruction Format of Wait Command Statements

```
WAIT (SIN[(Argument)]>=(NUM))
                ├─ R[j]
                └─ AR[j]
```

- A mixed logic expression can be specified in the conditional statement of a wait command.
- Wait until the specified condition is satisfied.
- The argument of math functions is “Register “ and “Argument Register”.
- Constant is not used directly.

31.3 FUNCTION SPECIFICATION OF MATH FUNCTIONS

31.3.1 Square Root（SQRT）

Function name: SQRT[X]
Argument: A positive integer or a positive real value to calculate square root.
This function calculates the positive square root of the argument.
Example:

```
1 : R[1] = 4
2 : R[2] = 2
3 : R[3] = (SQRT[R[1]])
4 : R[4] = (SQRT[R[2]])
```

Execute this TP program then,

```
R[3] = 2
R[4] = 1.41421
```

If the argument is minus value, then the following error occurs.

```
INTP-674        Invalid argument
```

31.3.2 Trigonometric Function（SIN）

Function name: SIN[X]
Argument: An angle to calculate SIN (in degree). An integer or a real value.
This function calculates sine of the argument.
Example:

```
1 : R[1] = 90
2 : R[2] = 45
3 : R[3] = (SIN[R[1]])
4 : R[4] = (SIN[R[2]])
```

Execute this TP program then,

```
R[3] = 1
R[4] = 0.70710
```

31.3.3 Trigonometric Function（COS）

Function name: COS[X]
Argument: An angle to calculate COS (in degree). An integer or a real value.
This function calculates cosine of the argument.
Example:

```
1 : R[1] = 90
2 : R[2] = 45
3 : R[3] = (COS[R[1]])
4 : R[4] = (COS[R[2]])
```

Execute this TP program then,

```
R[3] = 0
R[4] = 0.70710
```

31.3.4 Trigonometric Function（TAN）

Function name: TAN[X]
Argument: An angle to calculate TAN (in degree). An integer or a real value.
This function calculates tangent of the arguments.
Example:

```
1 : R[1] = 0
2 : R[2] = 45
3 : R[3] = (TAN[R[1]])
4 : R[4] = (TAN[R[2]])
```

Execute this TP program then,

```
R[3] = 0
R[4] = 1
```

If the argument equals to 180*n+90 and execute, then the following Error will be posted.

```
INTP-674        Invalid argument
```

31.3.5 Inverse Trigonometric Function（ASIN）

Function name: ASIN[x]
Argument: An integer or a real value to calculate ASIN. (Return value is degree.)
This function calculates arc sine of the argument.

Example:

```
1 : R[1] = 1
2 : R[2] = 0.5
3 : R[3] = (ASIN[R[1]])
4 : R[4] = (ASIN[R[2]])
```

Execute this TP program then,

```
R[3] = 90
R[4] = 30
```

If the argument ranges are x>1 or x<-1 and execute program, then the following Error will be posted.

```
INTP-674        Invalid argument
```

31.3.6 Inverse Trigonometric Function（ACOS）

Function name: ACOS[x]
Argument: An integer or a real value to calculate ACOS. (Return value is degree.)
This function calculates arc cosine of the argument.
Example:

```
1 : R[1] = 1
2 : R[2] = 0.5
3 : R[3] = (ACOS[R[1]])
4 : R[4] = (ACOS[R[2]])
```

Execute this TP program then,

```
R[3] = 0
R[4] = 60
```

If the argument ranges are x>1 or x<-1 and execute program, then the following Error will be posted.

```
INTP-674        Invalid argument
```

31.3.7 Inverse Trigonometric Function（ATAN2）

Function name: ATAN2[x,y]
1st argument: x coordinates of the point to calculate ATAN2. An integer or a real value.
2nd argument: y coordinates of the point to calculate ATAN2. An integer or a real value.
This function calculates arc tangent of the 1st and 2nd arguments. (Return value is degree.)
Example:

```
1 : R[1] = -1
2 : R[2] = 0.5
3 : R[3] = (ATAN2[R[1],R[2]])
4 : R[4] = (ATAN2[R[2],R[1]])
```

Execute this TP program then,

```
R[3] = 153.4349
R[4] = -63.43
```

31.3.8 Inverse Trigonometric Function（ATAN）

Function name: ATAN [x]
Argument: An integer or a real value to calculate ATAN.
This function calculates arc tangent of the argument. (Return value is degree.)
Example:

```
1 : R[1] = 1
2 : R[2] = 0.5
3 : R[3] = (ATAN[R[1]])
4 : R[4] = (ATAN[R[2]])
```

Execute this TP program then,

```
R[3] = 45
R[4] = 26.56
```

31.3.9 Exponent

Function name: EXP[x]
Argument: An integer or a real value to calculate EXP.
This function calculates a value equal to e (approximately 2.71828) raised to the power specified by the argument.
Example:

```
1 : R[1] = 0
2 : R[2] = 1
3 : R[3] = (EXP[R[1]])
4 : R[4] = (EXP[R[2]])
```

Execute this TP program then,

```
R[3] = 1
R[4] = 2.71828
```

31.3.10 Natural Logarithm

Function name: LN[x]
Argument: An integer or a real value to calculate LN.
This function calculates the natural logarithm of the argument.
Example:

```
1 : R[1] = 1
2 : R[2] = 2.718281828
3 : R[3] = (LN[R[1]])
4 : R[4] = (LN[R[2]])
```

Execute this TP program then,

```
R[3] = 0
R[4] = 1
```

If the argument is x <= 0 and the program execute, then the following Error will be posted.

```
INTP-674        Invalid argument
```

31.3.11 Absolute（ABS）

Function name: ABS[x]
Argument: An integer or a real value to calculate ABS.
This function calculates the absolute value of the argument.
Example:

```
1 : R[1] = 5.5
2 : R[2] = -5.2
3 : R[3] = (ABS[R[1]])
4 : R[4] = (ABS[R[2]])
```

Execute this TP program then,

```
R[3] = 5.5
R[4] = 5.2
```

31.3.12 Truncate（TRUNC）

Function name: TRUNC[x]
Argument: A real value to calculate TRUNC.
This function converts the real argument to an integer by removing fractional part of the real value.
Example:

```
1 : R[1] = 5.5
2 : R[2] = -5.2
3 : R[3] = (TRUNC[R[1]])
4 : R[4] = (TRUNC[R[2]])
```

Execute this TP program then,

```
R[3] = 5
R[4] = -5
```

31.3.13 Round Off（ROUND）

Function name: ROUND[x]
Argument: A real value to calculate ROUND.
This function calculates integer value closest to the real argument.

Example:

```
1 : R[1] = 5.5
2 : R[2] = -5.2
3 : R[3] = (ROUND[R[1]])
4 : R[4] = (ROUND[R[2]])
```

Execute this TP program then,

```
R[3] = 6
R[4] = -5
```

31.4 BACKGROUND OPERATION OF MATH FUNCTION

Math function instructions are part of the mixed logic instruction.
Then Math function instruction can be used in background operation.

Table 31.4 Background operation of math function

Available data	SIN, COS, TAN, ASIN, ACOS, ATAN, ATAN2, SQRT, LN, EXP, ABS, TRUNC, ROUND
Available operators	(,), =, <>, <, <=, >, >=, +, -, *, /, DIV, MOD

31.5 TEACH MATH FUNCTION INSTRUCTION

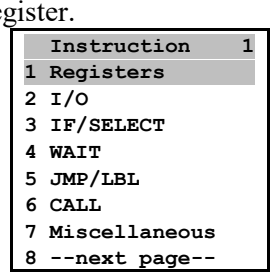
Math function can be used in mixed logic instruction.
As an example, the teaching process of following statement is as follows.

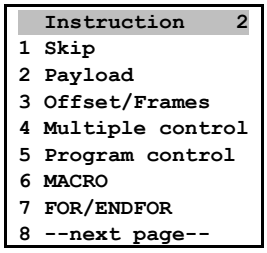
```
1 : R[1] = (SIN[R[2]])
```

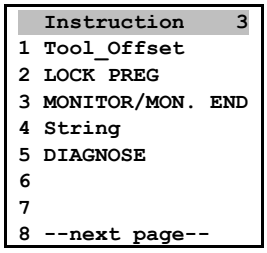
Procedure 31-1 Examples of teaching math function

Step

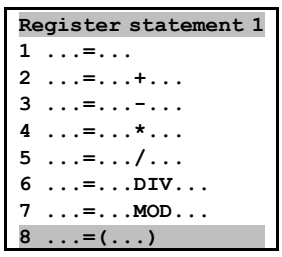
1 Move the cursor to a edited line, and press the F1, [INST] key.
2 Select Register.



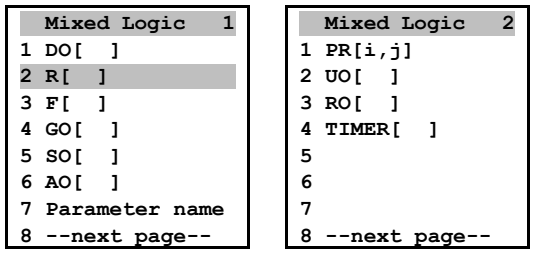




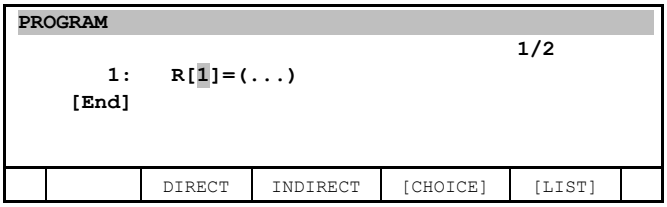
3 Select mixed logic instructions "…= (…)".



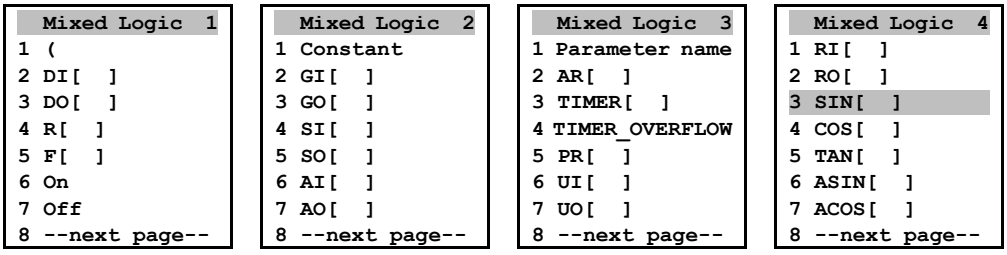
4 Select register “R[ ]”. Then the cursor moves in the square bracket.



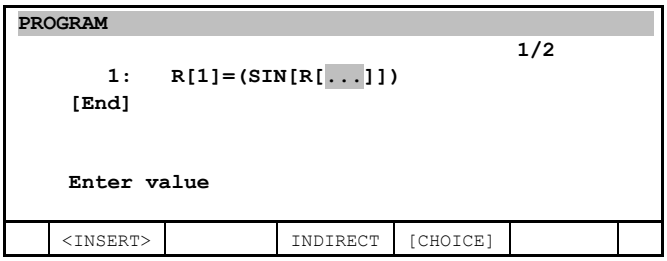
5 Input index 1.



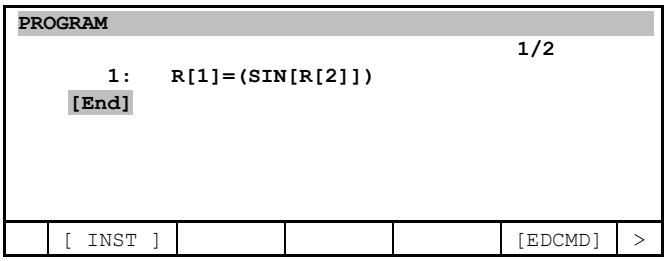
6 Then the cursor moves to the right side of equal sign.
7 Select “SIN[ ]”.



8 Then the cursor moves in the square bracket. And the default value R[…] is selected.
Arguments are “Register “ and “Argument Register”. (Not Constant)
The “Register” is default value of argument.
So to use “Argument Register”, press the F3, [CHOICE] then select “Argument Register”.
9 For example Register[2] is used here. Input value 2 then the cursor moves to the right parentheses.



10 Move down the cursor to finish teaching.
11 Teaching of SIN[R[2]] function completed.



31.6 RESTRICTION OF TEACHING MATH FUNCTION

- The argument of math functions is “Register “ and “Argument Register”.
- Constant is not used directly.
- If the user wants to use constant, input constant to a register, and use the register.
- You can put multiple math functions in a single line.
- You cannot put math functions in a math function.

Following statements are supported.

```
1 : R[1] = (SIN[R[2]])
2 : R[2] = (COS[AR[3]])
3 : R[3] = (SIN[R[4]] + COS[R]4)])
```

Following statements are not supported.

```
R[1] = (SIN[90])
        * Input constant directly.
R[3] = (SQRT[SIN[R[4]] * SIN[R[4]] + COS[R[4]]*COS[R[4]]])
        * SIN and COS are in SQRT.
```

- Math function might have a negligible calculation error of 10^{-7} due to internal calculation error.
- For example the result of COS[R[1]] (when R[1]=90), may be 10^{-8} that is not exactly zero. This behavior can be avoided by rounding off the result using ROUND function.

31.7 EXCEPTIONS AND RESTRICTION

Math function instruction posts the alarm in the following conditions.

- When the argument exceeds the domain of function definition.
  Example :LN[R[1]] (R[1] = -1)
  :ASIN[R[2]] (R[2] = 10)
  :SQRT[R[3]] (R[3] = -4)
  :TAN[R[4]] (R[4] = 90)

INTP-674	Invalid argument

- The value overflows when executed.
  Example :EXP[R[1]] (R[1] = 100)

INTP-323	Value overflow

- If the variables are divided by zero then, the following error will be occurred.
  Example :R[1] / SIN[R[2]] (R[2] = 0)
  :10/COS[R[3]] (R[3] = 90)

INTP-208	Divide by 0

32 SERVO TOOL CHANGE FUNCTION

32.1 OUTLINE

32.1.1 Feature of Function

The Servo Tool Change function:

- Enables you to change between two or more servo tools without powering off the robot controller.
- Allows you to change between air-driven and servo tools.

What is a servo tool?

A servo tool is a tool with an axis driven by a servo motor which is controlled by a robot controller.

32.1.2 Basic Specification

Total number of interchangeable servo tools	9 for a one-axis servo tool change 4 for a two-axis servo tool change 3 for a three-axis servo tool change Only one tool can be controlled at the same time.
Tool detaching time (default)	0.4 sec (Battery type) / 0.4 sec (Battery-less type) *1)
Tool attaching time (default)	4 sec (Battery type) / 6 sec (Battery-less type) *1)
Tool axis Pulsecoder backup battery	Both battery type and battery-less types are available.*2)

(*1) The time includes the internal processing time, but does not include the time required for robot tool-changing motion (*2) The batteryless type is not available for multi-axis servo tool change.

32.1.3 Restrictions, prohibitions, and precautions

Restrictions

Tool changing is possible only between servo tools that are set within the same motion group.

- Because servo tools without a battery for tool axis Pulsecoder value backup require calibration after they are attached, these tools will require longer processing time for tool changing than battery-equipped tools.
- The motor of the tool change must be controlled by the same amplifier.
- The tool change instruction cannot be used in a multi-tasking system, because the tool change instruction cannot be executed by multiple tasks at the same time.
- You cannot install both the Servo Tool Change and Servo Gun Change functions at the same time.
- In a system with the Servo Tool Change function installed, the Dynamic Brake Release function cannot be used for servo guns.
- Use the last motion group as the motion group for which the Servo Tool Change function is enabled.
- The servo tool can have a continuous turn axis, but calibration motion type 3 and 4 are not available if the servo tool is batteryless type.

Prohibitions

- Do not force detaching the tool while the controller is powered off.
- The Line Tracking function cannot be used in systems with the Servo Tool Change function.

Notes on using the battery-less servo tool type

- When calibration motion type 1 or 2 is selected, calibration cannot be interrupted by a hold or emergency stop. Program execution must not be stopped during the tool attachment process.
- When calibration motion type 3 or 4 is selected, the stroke limit must be expanded so that the tool axis touches the open limit.
  These types are not available for a continuous turn axis.
- When calibration motion type 5 or 6 is selected, a detection switch such as a limit switch or dog switch must be installed in the tool.
  If the servo tool has a continuous turn axis, that axis cannot restore the same position (phase of the rotation axis) as before it was detached.

For details on the calibration motion types, see Section 32.4, TOOL CHANGE SETUP.

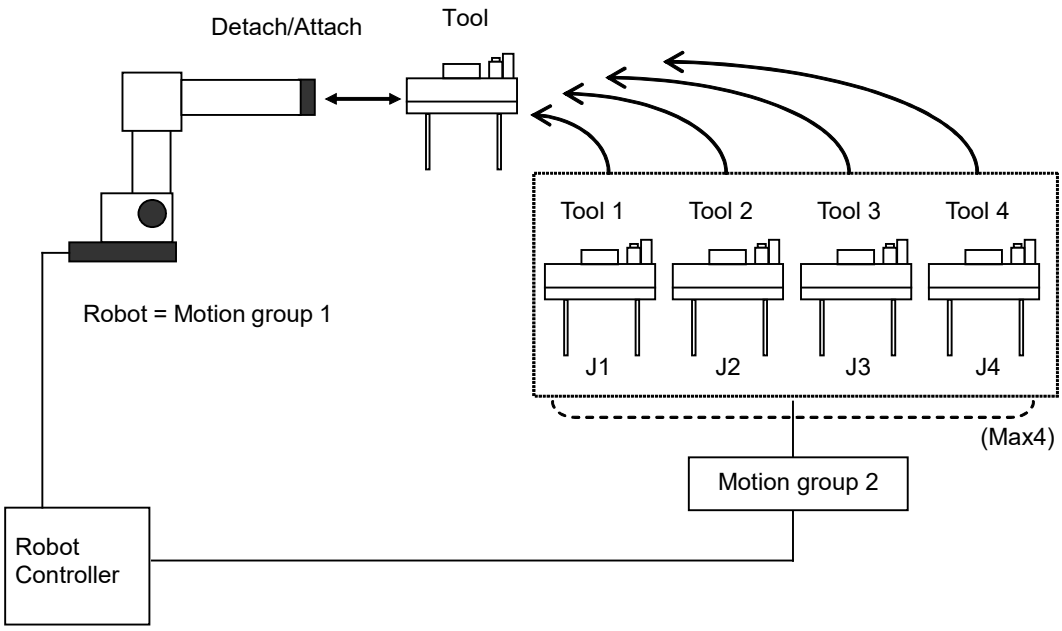
Precautions when using a multi-axis servo tool

- The servo tool number must be assigned to consecutive axes.
- The servo tool must have a Pulsecoder backup battery.
- To change to a two-axis servo tool, you must use a two-axis tool change unit.
- To change to a three-axis servo tool, you must use a three-axis tool change unit.
- Servo tools with four or more axes cannot be changed.

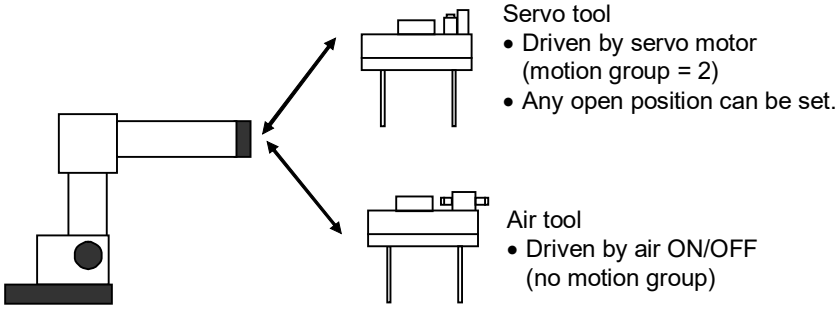
32.1.4 System Configuration

A one-axis Servo Tool Change system is constructed as follows.

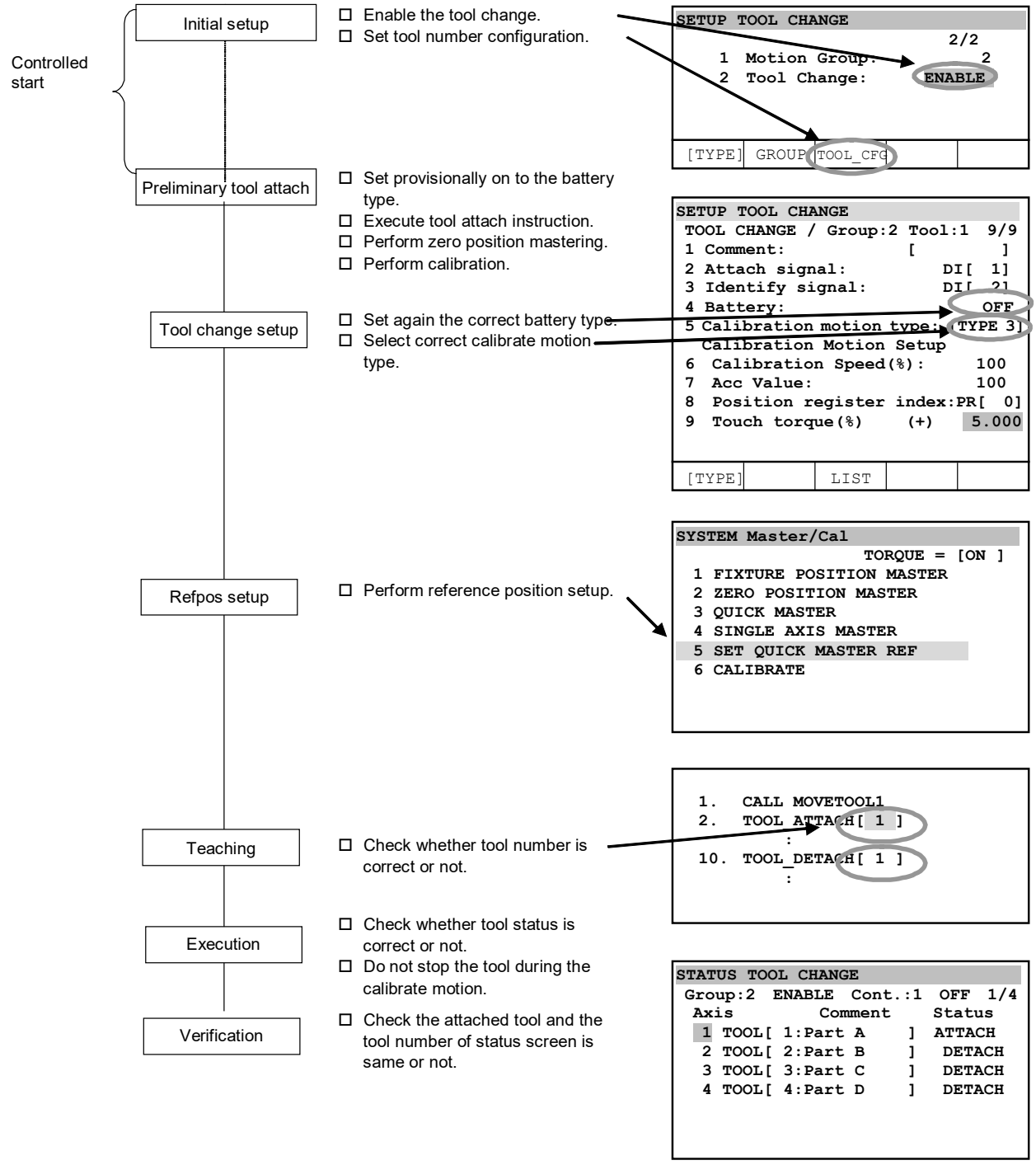
.



You can replace the servo tool with an air tool by changing it on the controller.



32.1.5 Outline of Servo Tool Change Installation



32.2 INITIAL SETUP

Perform the following initial setup to install a Servo Tool Change system.

- Set up the motion parameters for the servo tool axes
- Set up the configuration of the servo tool number
- Enable the Servo Tool Change function.

NOTE
his setup must be performed when the system is installed. However, since this setup is done only once at installation, there is no need to make these changes during normal use.

You perform tool change initial setup at controlled start. See Sections 32.9.1, 32.9.2, and 32.9.3 for detailed information.

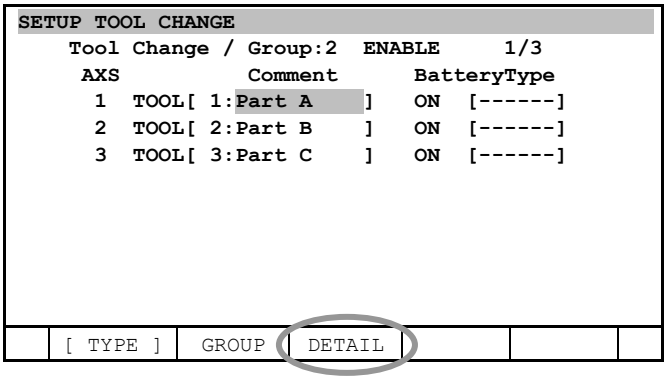
32.3 PRELIMINARY TOOL ATTACHING OPERATION

When Tool Change is enabled as described in Section 32.2, INITIAL SETUP, all tools assigned to the target motion group will be in the detached state.

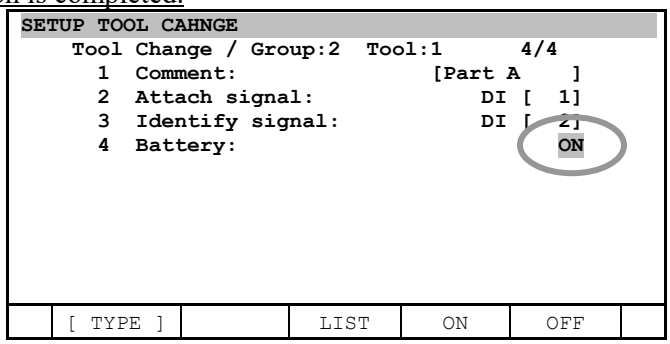
NOTE
If a system was already running with a servo tool before the Servo Tool Change software was installed, and the previously used tool is still attached, the tool is still considered to be attached. In this case, you do not have to perform the preliminary tool attachment in Steps 1 to 9.

When using a tool for the first time, be sure to perform the following steps for the preliminary attaching operation, then continue to Section 32.4, TOOL CHANGE SETUP.

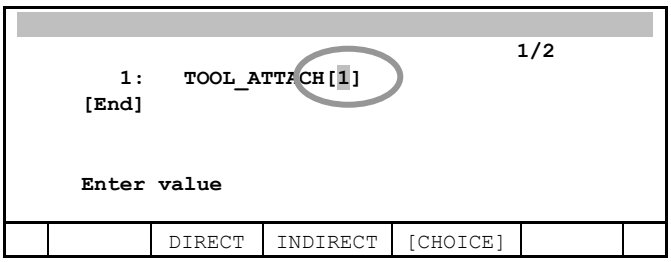
1 Press MENU.
2 Select SETUP.
3 Press F1, [TYPE].
4 Select Tool Change. The SETUP TOOL CHANGE list screen will be displayed.
5 From the list, select a tool and press F3, DETAIL. The SETUP TOOL CHANGE detail screen will be displayed.



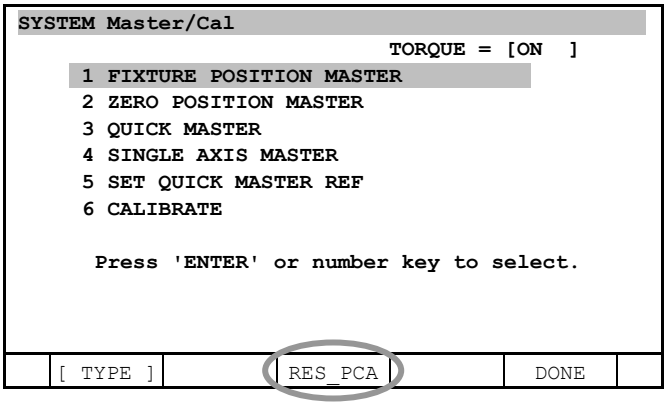
6 On the SETUP TOOL CHANGE detail screen, temporarily set the Battery item to ON. Even if the actual tool has no battery, select ON in this step. This item will be set again after the preliminary attaching operation is completed.



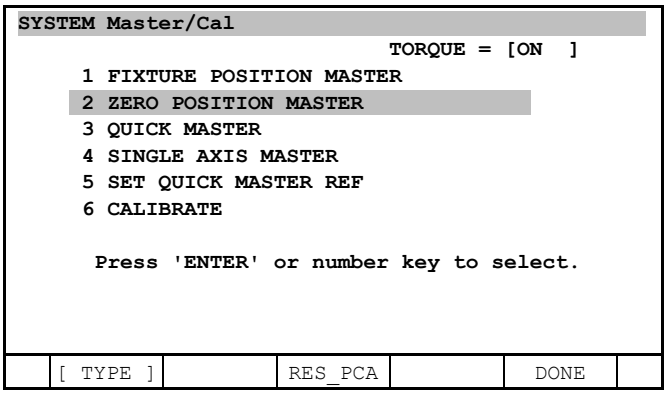
7 Attach a target tool to the tool changer.
8 Create a program to teach a TOOL ATTACH instruction.
9 Execute the TOOL ATTACH instruction to attach the tool temporarily.



10 After the TOOL ATTACH instruction is executed, the following alarms are issued:
SRVO-075 Pulse not established
SRVO-062 BZAL alarm
11 On the SYSTEM Master/Cal screen, press F3, RES_PCA and select YES.



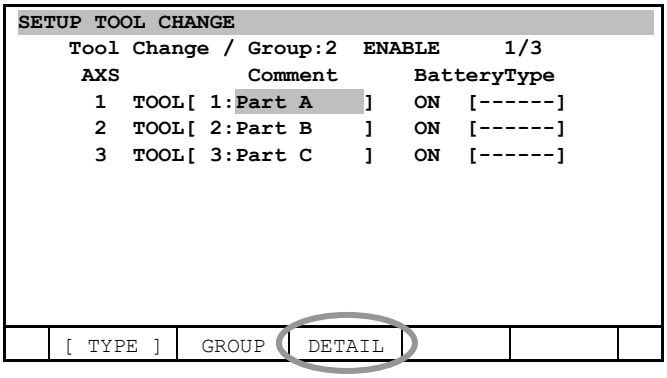
12 Cycle controller power.
13 Jog the servo tool axis.
Continue jogging the servo tool axis until the following message is displayed when the RESET button is pressed:
SRVO-075 Pulse not established.
14 In the same manner as for ordinary servo tools, perform zero-position mastering for the servo tool axis.

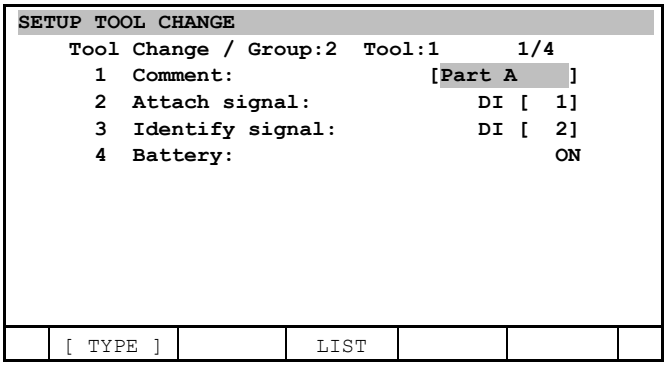


15 After completing tool change setup and reference position setting for the first tool, detach the first tool using the TOOL DETACH instruction, then perform the preliminary attaching operation for the second and subsequent tools.

32.4 TOOL CHANGE SETUP

You must perform tool change setup for each tool.





1 Press [MENU].
2 Select SETUP.
3 Press F1, [TYPE].
4 Select Tool change and the SETUP TOOL CHANGE screen will be displayed.
5 Select the tool from the list of tools, and press the F3, DETAIL. The detail screen will be displayed.

Common setup (common setup item for all tools)

Item	Description
Comment	his item is the comment for each tool.
Attach signal	This item is the signal used for tool set verification. Available I/O types are = DI and RI • This signal is checked before the tool attach/detach instruction. If this signal is OFF, an error occurs. • If the index is zero, the above checking is canceled.
Identify signal	This signal used for tool identification. Available I/O types are = DI, RI and GI • This signal is checked before the tool attach/detach instruction. If this signal is OFF, an error occurs. • If the index is zero, the above checking is canceled.
Battery	This item allows you to select the battery type. ON = Battery backup type OFF = Batteryless type NOTE To change a multi-axis servo tool, Battery must be set to ON.

(Select the following when Battery is set to OFF in the common settings.)

Item	Description
Calibration Motion Type	For battery-less tools, the position (pulse value) is lost when the tool is detached. For this reason, calibration must be performed when the tool is attached. Calibration involves the following two operations: • Turning the motor two turns for positioning the Pulsecoder. This operation will be referred to as "positioning" from here on. • Calibrating at a preset reference position You select from the following types, 1 to 6, to suit your system requirements. See Table 32.4 and Fig. 32.4. **⚠ CAUTION** In the case the tool is a rotational axis and Type 3 to 6, if the tool axis moves by an integer rotation, i.e., the number of rotation changes, between the detachment and the attachment, it cannot restore the same position (phase of rotation axis) as before the detachment. If you want to avoid this, use a motor with a brake.

Calibration motion type selection criteria

- Types 1, 2

Select these types for systems capable of holding the tool detachment position as a result of braking the servo tool axis motor until the next servo tool is attached.

⚠ CAUTION
For types 1 and 2, if calibration is interrupted by a hold or emergency stop, the reference position data will be lost. When this happens, see Section 32.11.2, Robot Stopped during Calibration"

For types 1 and 2, the tool detachment position is automatically stored as the reference position, and calibration is performed at that position after the tool is attached. For this reason, the position where the tool is attached must be the same as position where it was detached.
The allowable errors between the detach and attach positions must fall within the following ranges for linear and rotational axes respectively:

Linear axis:
±(gear ratio/2) mm or less around the detach position
Rotational axis:
±((360/gear ratio)/2) deg or less around the detach position

Note that if the above ranges are exceeded, positional misalignment will occur during tool attachment.

The difference between types 1 and 2 is in the motion direction during calibration. (See Table 32.4 and Fig. 32.4.)

Type 1:
Specify this type when the tool is detached at a position where it is impossible to move the tool in the positive direction when jogging two or more revolutions of the motor.

Type 2:
Specify this type when the tool is detached at a position where it is impossible to move the tool in the negative direction when jogging two or more revolutions of the motor.

- Types 3, 4

Select these types in systems that cannot hold the tool detachment position until the next servo tool attachment, because the servo tool axis motor does not have a brake. In this kind of system, the reference position must be set to another position, because the tool detachment position cannot be used as the reference position.

Types 3 and 4 use motor torque to detect when the tool axis contacts the open travel limit and sets the contact position as the reference position. This is because by default, the tool axis moves in the negative jogging direction during calibration motion. For details on how to set the reference position, see Section 32.5, SETTING THE REFERENCE POSITION.
The difference between types 3 and 4 is in the motion direction during calibration. See Table 32.4 and Fig. 32.4.

Type 3:
Specify this type when the tool is detached at a position where the distance to the open travel limit is two or more motor revolutions.
Type 4:
Specify this type when the tool is detached at a position where the distance to the open limit is two or less motor revolutions.

NOTE
1 In the calibration, to move the tool axis in the positive jogging direction (to jog to the positive limit), change the close direction.
2 With types 3 and 4, you must expand the stroke limit so that the tool axis touches the open limit.

NOTE
These types are not available for a continuous turn axis.

- Types 5, 6

As with types 3 and 4, select these types in systems that cannot hold the tool detachment position until the next servo tool attachment because the servo tool axis motor does not have a brake. With types 5 and 6, however, a detection signal must be provided such as a limit switch or dog switch, and the contact position must be set as the reference position. Install the detection switch so that it is turned on when the tool axis is jogged in the negative direction. This is because, by default, the calibration

motion causes the tool axis to move in the negative jogging direction. For details on setting the reference position, see Section 32.5, SETTING THE REFERENCE POSITION.
The difference between types 5 and 6 is in the motion direction during calibration. See Table 32.4 and Fig. 32.4.

Type 5:
Specify this type when the tool is detached at a position where the distance to the position where the detect signal turns ON is two or more motor revolutions.

Type 6: Specify this type when the tool is detached at a position where the distance to the position where the detect signal turns ON is two or fewer motor revolutions.

NOTE
1 In the calibration, to move the tool axis in the positive jogging direction, change the close direction.
2 With types 5 and 6, a switch such as a limit switch or dog switch must be installed on the tool.

NOTE
A continuous turn axis cannot restore the same position (phase of the rotation axis) as before it was detached.

Table 32.4 Calibration motion type

Type	Calibration Motion (Close direction is 1.)	Set Position of Reference position
Type 1	The tool is moved in the negative jogging direction by two motor revolutions and is subsequently moved in the positive direction by two revolutions, then the tool is calibrated. (NOTE 1) Set the setting item (1) on the next page.	Position when tool is detached Reference position is not required. (NOTE 2)
Type 2	The tool is moved in the positive jogging direction by two motor revolutions and is then moved in the negative direction by two revolutions, and then the tool is calibrated. Set the setting item (1) on the next page. (NOTE 1)	Position when tool is detached Reference position is not required. (NOTE 2)
Type 3	The tool is moved in the negative jogging direction until the touch torque is detected, then the tool is calibrated. Arrival at the limit is detected by the motor torque. Set the setting items (1), (2) and (3) on the next page.	Open limit position
Type 4	The tool is moved in the positive jogging direction by two motor revolutions and is then moved to the limit on the negative side until touch torque is detected, Then the tool is calibrated. Arrival at the limit is detected by the motor torque. Set the setting items (1), (2) and (3) on the next page.	Open limit position
Type 5	The tool is moved in the negative jogging direction until the detection signal is turned on, then the tool is calibrated. Set the setting items (1), (2) and (4) on the next page.	Position where the detection signal turns ON
Type 6	The tool is moved in the positive jogging direction by two motor revolutions and is then moved in the negative direction until the detection signal is turned on, then the tool is calibrated. Set the setting items (1), (2) and (4) on the next page.	Position where the detection signal turns ON

NOTE

1 When types 1 and 2 are selected, the tool mastering data is lost if the operation is interrupted due to a hold or emergency stop during calibration that is performed during the tool change sequence. If this happens, manually perform zero position mastering.

2 For types 1 and 2, the system automatically stores the position where the servo tool is detached as the reference position. This stored reference position is used in calibration the next time the same tool is attached. For this reason, when type 1 or 2 is selected, the detachment position and the attachment position must be the same. Do not select type 1 or 2 when a motor without a brake is used for the servo tool axis.

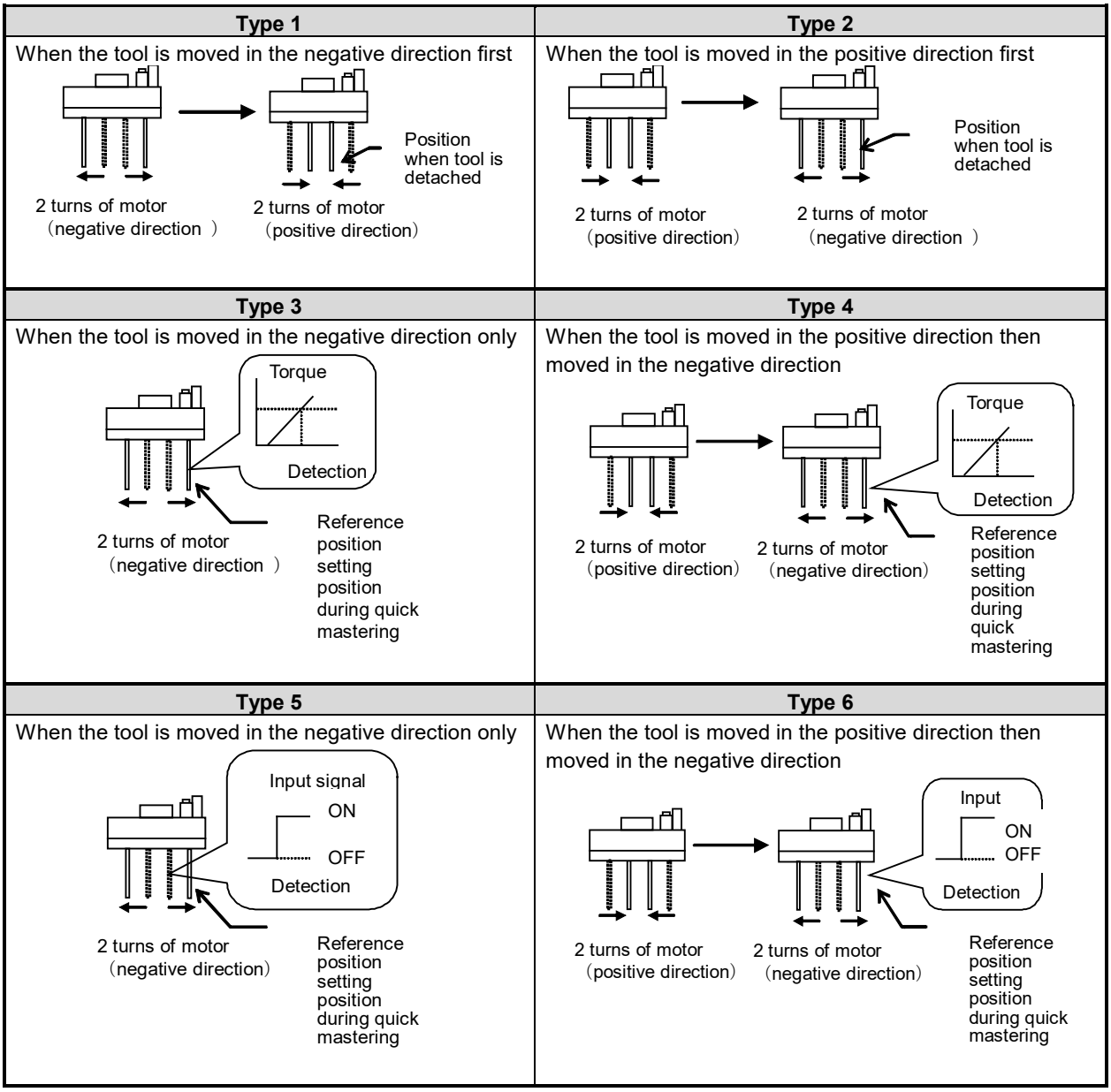


Fig. 32.4 Calibration motion type

The positive and negative directions in Fig. 32.4 are determined on the assumption that close direction = 1 (Data ID: $SERVOGUN_SETUP.CLOSE_DIRECTION_GUN[#]/RW/# is gun number.) is set. If this d data ID is set to 0, the directions are reversed.

Screen for calibration types 1, 2

```
SETUP TOOL CHANGE
    Tool Change / Group:2  Tool:1      5/7
      1  Comment:               [          ]
      2  Attach signal:              DI [  1]
      3  Identify signal:            DI [  2]
      4  Battery:                         OFF
      5  Calibration motion type:    [TYPE 2]
         Calibration Motion Setup
      6    Calibration Speed(%):         100
      7    Acc Value:                  ACC100

[ TYPE ]  |  LIST  [CHOICE]
```

Screen for calibration types 3, 4

```
SETUP TOOL CHANGE
    Tool Change / Group:2  Tool:1      5/9
      1  Comment:               [          ]
      2  Attach signal:              DI [  1]
      3  Identify signal:            DI [  2]
      4  Battery:                         OFF
      5  Calibration motion type:    [TYPE 4]
         Calibration Motion Setup
      6    Calibration Speed(%):         100
      7    Acc Value:                  ACC100
      8    Position register index:  PR[  1]
      9    Touch torque(%):     (+)    5.000

[TYPE]  |  LIST  [CHOICE]
```

Screen for calibration types 5, 6

```
SETUP TOOL CHANGE
    Tool Change / Group:2  Tool:1      5/9
      1  Comment:               [          ]
      2  Attach signal:              DI [  1]
      3  Identify signal:            DI [  2]
      4  Battery:                         OFF
      5  Calibration motion type:    [TYPE 6]
         Calibration Motion Setup
      6    Calibration Speed(%):         100
      7    Acc Value:                  ACC100
      8    Position register index:  PR[  1]
      9    Detect signal:            DI [  0]

[TYPE]  |  LIST  [CHOICE]
```

(1) Open limit motion settings - Set for all calibration types.

Calibration speed	Specify the calibration motion speed.
Accelerate rate	Specify the calibration motion acceleration rate. The default is 100(%).

(2) Position register No. setting
Set as follows when types 3 to 6 are selected in Calibration Setup.

Position register No	Specify the position register to hold the position at the moment when the reference position (open limit or signal input position) is detected to have been reached. The position register you specify here should be used exclusively for calibration. Do not use the same position register in a program, for example.

(3) Open limit detection torque setting
Set as follows when types 3 or 4 are selected in the Calibration Setup.

Touch torque (%)	Specify the threshold value of the motor torque for detecting that the open limit has been reached. Set the value within a range 0.0 to 100.0%.

(4) Detection signal setting - Set as follows when types 5 or 6 are selected in Calibration Setup.

Detection signal	Specify the type and No. of the input signal for detecting that the reference position has been reached. • DI • RI Specify the No. available for each of the signals as the signal No.

32.5 SETTING THE REFERENCE POSITION

32.5.1 Battery-less Tools

Battery-less tools lose their position (pulse value) when the tool is detached, so they must be calibrated when they are attached.
The position used for calibration is called the "reference position".
On battery-less tools, the reference position must be set before the tool is detached.
The method of setting the reference position differs according to the calibration motion type.

Types 1, 2

The system automatically sets the reference position when the servo tool is detached.

Types 3, 4

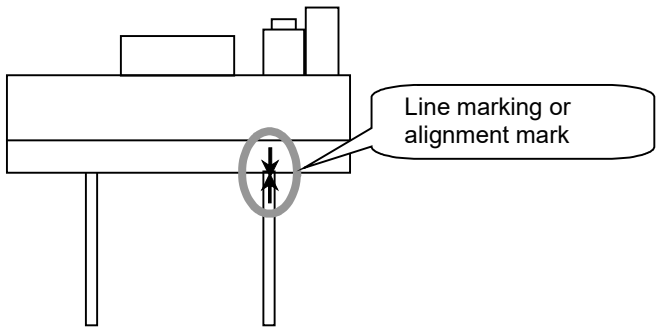
Create a program for detecting the reference position using touch torque, and execute this program to calculate the reference position. For details, refer to Section 32.10.1.

Types 5, 6

Create a program for detecting the reference position using the detection signal, and execute this program to calculate the reference position. For details, refer to Section 32.10.2.

32.5.2 Battery-Containing Tools

For battery-containing tools, the position (pulse value) is not lost even if the tool is detached. Therefore, a battery-powered tool does not require a reference position for calibration when the tool is attached, as with battery-less tools. However, it is recommended that you set a reference position for quick mastering by marking a line or alignment mark as shown in the following figure for extra safety in the event that the battery comes loose. For details on how to set the reference position for quick mastering, refer to Section 32.10.3.



32.6 TOOL CHANGE INSTRUCTION

The following instructions are provided for changing servo tools.

32.6.1 TOOL DETACH Instruction

This instruction detaches the servo tool without turning off power.

Format

TOOL DETACH[i]

i : Tool number (1 to 4)
Specify the tool number you want to detach.
R[ ] and AR[ ] can be used.

Ex) TOOL DETACH[1]
TOOL DETACH[R[2]]
TOOL DETACH[AR[1]]

32.6.2 TOOL ATTACH Instruction

This instruction attaches the servo tool without turning off power. For a battery-less tool, mastering and calibration are performed automatically.

Format

TOOL ATTACH[i]

i : Tool number (1 to 4)
Specify the tool number you want to detach.
R[ ] and AR[ ] can be used.

Ex) TOOL ATTACH[4]
TOOL ATTACH[R[3]]
TOOL ATTACH[AR[1]]

32.6.3 Example Program

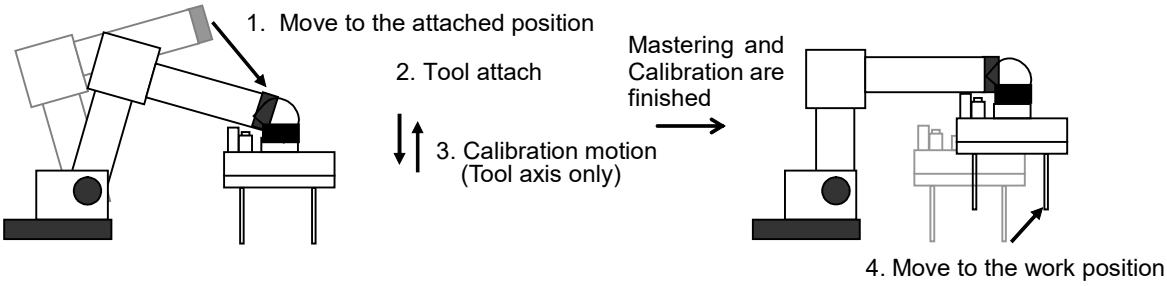
Example program

```
                                    [Explanation]
 1: J P[1:Approach] 100% FINE       1:  Move to approach position.
 2: L P[2:TOOL1 SET] 100mm/sec FINE 2:  Set tool axis to 0 position.
 3: TOOL DETACH[1]                  3:  Detach tool 1 (sequence execute).
 4: L P[1:Approach] 100mm/sec FINE  4:  Release  tool1 mechanically.
                  :
11: J P[11: Approach] 100% FINE
12: L P[12:TOOL2 SET] 100mm/sec FINE
13: TOOL ATTACH[2]                  13: Attach tool 2 (sequence execute).
14: UTOOL_NUM=2                     14:Change tool frame number.
15: L P[13: Approach] 100mm/sec FINE
                  :
99: END                             99: Normal end
```

NOTE
The above program is an example program. Create programs to suit the requirements of your system.

32.6.4 Forward Execution (FWD)

You must make sure that the tool number specified in the instruction is same as the actual tool. For the attach instruction, you must also set the calibration motion type to the correct type.
For battery-less tools, calibration motion is also performed (Step 3 in the following figure). See Fig. 32.4 for more details.



⚠CAUTION
When the calibration motion type is type 1 or 2, if the motion is stopped with an emergency stop or HOLD during the calibration motion, the mastering data of the tool is lost. If the mastering data is lost, manually perform zero position mastering and calibration again.
If you do not manually master and calibrate the tool,
1) The current position displayed on the TP will not match the actual position.
2) If a motion instruction is executed, the following alarm will occur:
"MOTN-049 Attempt to move w/o calibrated"
3) The TOOL DETACH instruction will not work until manual mastering and calibration are finished.

When a TOOL DETACH instruction is executed, both the robot and tool do not move at all.

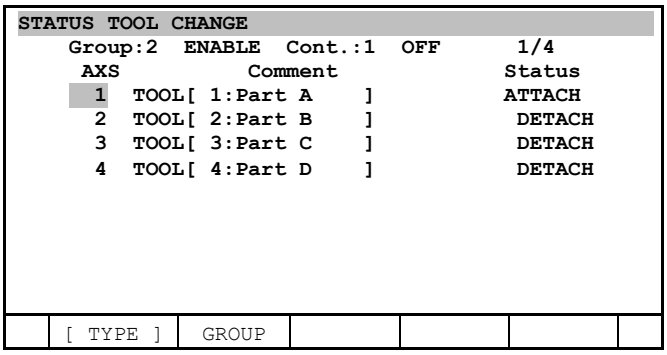
32.6.5 Backward Execution (BWD)

Both the TOOL ATTACH and TOOL DETACH instructions cannot be executed during backward execution. Therefore, tool change cannot be performed during backward execution.

32.7 TOOL CHANGE STATUS

Display the STATUS TOOL CHANGE screen to verify the currently attached or detached tool.

1 Press MENU.
2 Select NEXT and then STATUS.
3 Press F1, [TYPE].
4 Select Tool change to display the STATUS TOOL CHANGE screen.



The following status items are displayed.

Item	Description
AXS	This item is the axis number of the motion group assigned to the current motion group.
Tool	This item is the tool number assigned to the above axis.
Status	This item is the tool status: • ATTACH: The tool is attached. DETACH: The tool is detached.
Comment	This item is the comment for each tool.

32.8 TEACHING

In systems that perform tool changes, consider the information in this section during teaching.

32.8.1 Points to keep in mind when teaching

Position data

In a tool change system, each tool is assigned to each axis of the servo tool motion group. For example, if three kinds of tool are assigned to the tool change system, the servo tool motion group has data for three axes. This means that data for three axes data is recorded when the servo tool position is taught.

(Ex. The position data of tool1 which is assigned to first axis)

```
GP1:    UF : 0, UT : 1, Config : 'N, 0, 0, 0',
  X =   1879.25 mm,   Y = -1035.37 mm,   Z =    162.65 mm,
  W =   -178.67 deg,  P =   -13.68 deg,  R =    -37.32 deg
GP2:    UF : 0, UT : 1,
    J1 = 10.00 mm      J2 = 0.00 mm     J3 = 0.00 mm
```

Note that this position data has not only tool 1 data but also tool 2 and tool 3 data, which means that this position data can also move tool 2 and tool 3. If tool 2 is attached, tool 2 moves to the 0mm position. This specification allows for program flexibility. However, there is a risk of interference if you attach a different tool. To prevent this problem, teach each servo tool with a separate tool coordinate system number.
In this way, the position data for each tool has a different tool coordinate system number. Because of this, if the you execute a program with a different tool, the program will be stopped by the alarm “INTP-253 Tool frame number mismatch”.

Re-teaching after adding a servo tool

When you add a servo tool, you cannot use a program that was created before you added it. Re-teach the positions for attaching the corresponding tool.

32.8.2 Example Program

Example program

```
          :
 2:  TOOL ATTACH[1]
 3:  UTOOL_NUM=1
          :
11:J P[11] 100% FINE
12:L P[12] 100mm/sec FINE
13:J P[13:Approach] 100% FINE
14:L P[14:Tool1 set] 100mm/sec FINE
15:  TOOL DETACH[1]
16:L P[13:Approach] 100mm/sec FINE
          :
21:J P[21:Approach] 100% FINE
22:L P[22:Tool2] 100mm/sec FINE
23:  TOOL ATTACH[2]
```

```
24:  UTOOL_NUM=2
25:L P[23:Approach] 100mm/sec FINE
26:J P[24] 100% FINE
27:L P[25] 100mm/sec FINE
          :
99:  END
```

```
[Explanation]
2:Attach tool1.
3:Change tool frame number to 1.
        :
11:Handle with tool1.
12:Handle with tool1.
13:Move to approach position.
14:Set tool1 to tool stocker.
15:Detach tool1.
16:Release tool1.
        :
23:Attach tool2.
```

Move with tool number 1.

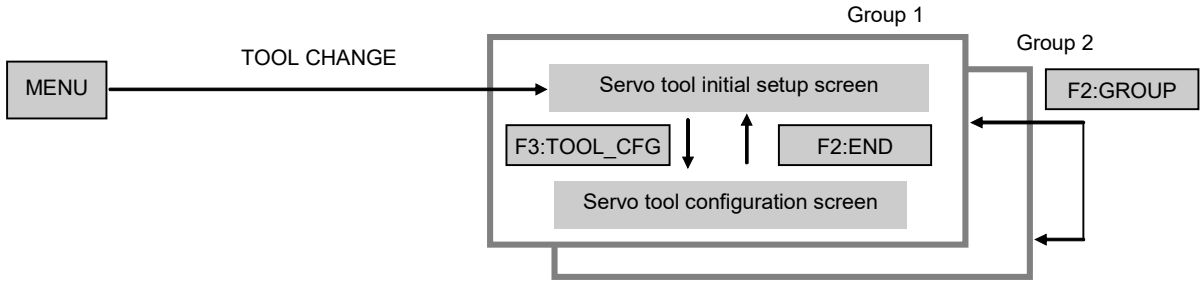
```
24:Change tool frame number to 2.
26:Handle with tool2.
27:Handle with tool2.
        :
99:Normal end
```

Move with tool number 2.

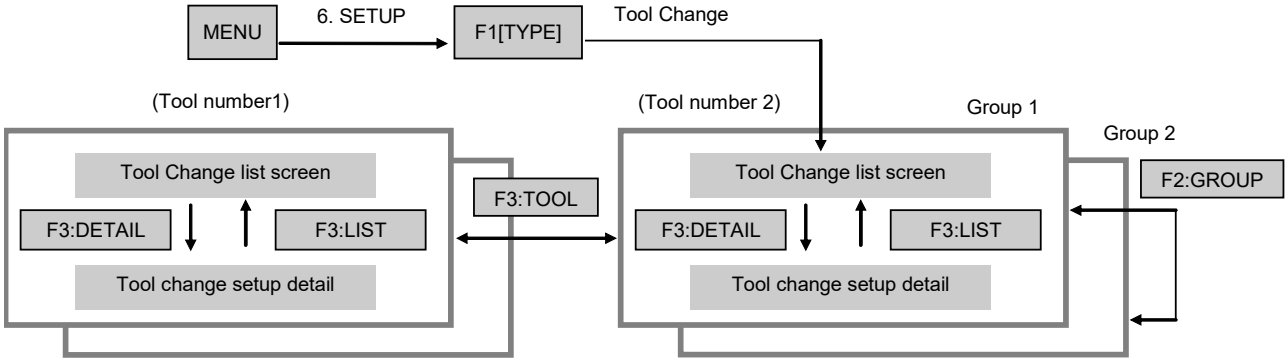
NOTE
This is an example program. Create a program related to your system.

32.9 SERVO TOOL CHANGE SCREENS

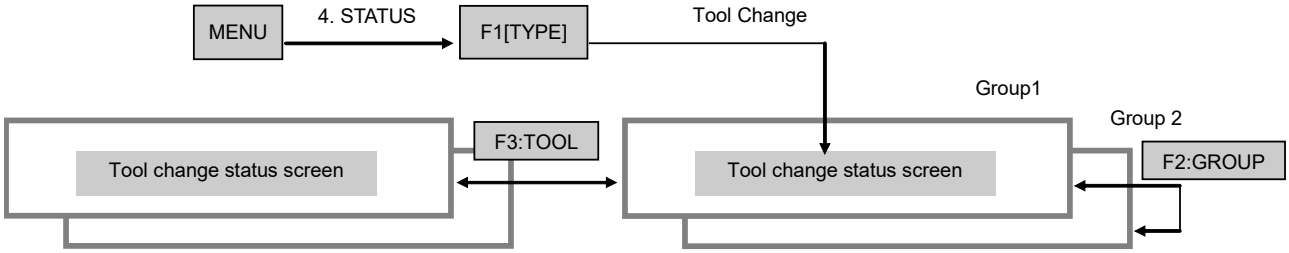
INITIAL SETUP SCREEN. This screen can be displayed from the CONTROLLED START Menu.)



SETUP SCREEN



STATUS SCREEN



32.10 TOOL CHANGE INITIAL SETUP

This section explains how to set up a tool change system.

NOTE
The setup procedures in this section only need to be performed once when the system is installed. Once these settings are made, there is no need to change them during normal use.

32.10.1 Setting Motion Parameters for Servo Tool Axes

Set the tool axes used for the Servo Tool Change function within each motion group according to the following:

- First to fourth axes: Independent axis setting procedure

32.10.2 Assigning Tool Numbers to Servo Tool Axes

Display the servo tool initial setup screen from the controlled start menu, and set the following:

1 Turn off power. While holding the PREV and NEXT keys, turn on power.
2 After a while, the following screen will be displayed. Select, Controlled start.

```
System version: V8.1035        2/10/2012

---------- CONFIGURATION MENU ----------

  1. Hot start
  2. Cold start
  3. Controlled start
  4. Maintenance

Select >
```

3 After a while, the controlled start screen will be displayed as follows.

```
Tool Setup                                              1/6
     1 F Number                                  F00000
     2 KAREL Prog in select menu                 YES
     3 Remote device:                            UserPanel
     4 Intrinsically safe TP:                    NO

[ TYPE ]                                         HELP
```

4 Press MENU.

5 Select NEXT and TOOL CHANGE. The servo tool initial setup screen will be displayed .

Motion Group	Input the motion group number for which the servo tools are set.
Tool Change (Data ID: TOOL_CHANGE_STATUS.TOOL_CHANGE_ENABLE[#]/$SHC_GRP[#].$SHC_ENB/RO/ # is group number)	Enable or disable the tool change function.

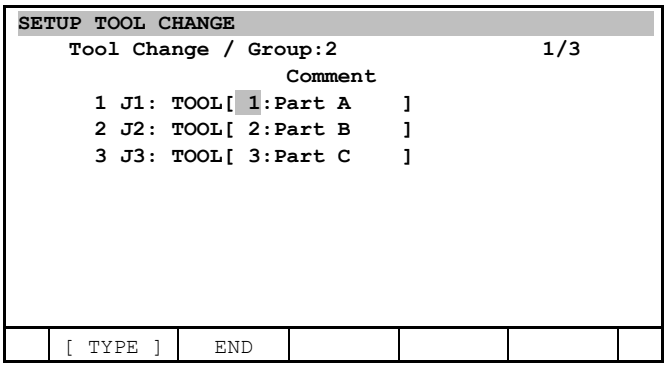
```
SETUP TOOL CHANGE                                       1/2

     1  Motion group:                                    2
     2  Tool Change:                              ENABLE

[ TYPE ]  GROUP  TOOL_CFG
```

The tool change function changes servo tools that are set within the same motion group. Note that you cannot change between tools that are set in different motion groups.

6 Press F3, TOOL_CFG. The SETUP TOOL CHANGE screen will be displayed.



Set the items listed in the following table for the tool axes included in the selected motion group.
If you are satisfied with the default settings, you do not need to change these items.

TOOL	Set a number from 1 to 4.
Comment	Set a comment for the tool number. You can also set a comment on the tool change setup screen after cold start.

To change to a multi-axis servo tool, assign the tool number to multiple axes.

NOTE
1 The tool number must be assigned to consecutive axes for multi-axis servo tool change.
2 Servo tools with four (or more) axes cannot be changed.

32.11 TOOL CHANGE REFERENCE POSITION SETUP METHOD (BATTERY-LESS SERVO TOOL TYPE)

32.11.1 Reference Position Setup for Calibration Types 3 and 4

Reference Position Setup Program

Create the following program to calculate the reference position based upon the sign of the calculated touch torque.

NOTE
Only the servo tool axis motion group can be used as the motion group of the program to be used for setting the reference position.
For example, if the servo tool axis motion group is the second group:
The motion group is [*,1,*,*,*,*,*,*].

Example Program Motion Group=[*,1,*,*,*,*,*,*]

```
1: J P[1:ZERO POS] 10% FINE

   (when sign of touch torque is plus)
2: SKIP CONDITION HPD_TRQ[ga] > 2 *1)
   (when sign of touch torque is minus)
2: SKIP CONDITION HPD_TRQ[ga] < -2 *1)

3: J P[2:TOOL OPEN END] 1% *2) FINE ACC 100
   *3)SKIP, LBL[1],PR[1]=JPOS
4: END
5: LBL[1]
6: UALM[1]
```

Explanation

1: Servo tool axis 0 position. The position need not necessarily be the 0 position.
2: As the touch torque in the skip condition, set the threshold value of the torque from which arrival at the limit can be detected when the tool moved in the negative direction of jogging touches the limit. Pay attention to the sign of the touch torque.
3: Move the tool in the negative direction of jogging to touch the limit. If detection fails, jump to LBL[1].
4: Normal end
5: Jump to here if torque cannot be detected.
6: Generate a user alarm.

NOTE

The above program is only an example. Create programs to suit the requirements of your system.
g: Motion group number of the currently attached tool
ga: Axis number of the currently attached tool
If the currently attached servo tool axis is the first axis of the second group, then g and ga are as follows: g=2, ga=1

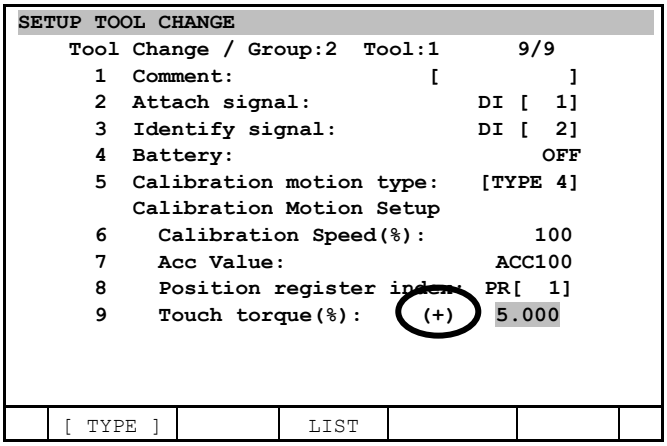
Program Operation Adjustment Method

Adjust the values of the shaded sections in the example program to adjust program motion. The motion adjusted here is used as the calibration motion performed when attaching the tool.

*1) Touch torque
Increase the value from about ±2, and calculate the value at which erroneous detection of contact does not occur.
The sign of touch torque is the sign that is currently displayed at in the Touch torque item on the following SETUP TOOL CHANGE screen.

NOTE

After confirming the sign in this screen, teach it in the program. Do not make a mistake when entering the sign. If you do so, contact with the open limit may not be detected.



*2) Program motion speed

To set the same calibration motion that is performed when attaching the tool, execute this program at the same override (normally, 100%) as when the attach instruction is executed for the tool used during actual production. For this reason, reduce the program motion speed so that contact is made with the open limit at a safe speed even if the program is executed at a high override.
For added safety, increase the program motion speed value successively from 1%.

*3) Acceleration rate
Decrease the acceleration rate if the motion is quick even if the program operation speed is 1%.

Reference Position Setup

After you have adjusted motionin the program, execute the program to bring the servo tool axis into contact with the open limit.
Set up the reference position according to the procedure described in Section 32.10.3 with the servo tool contacting the open limit.

Tool Change Setup

To set the motion in the newly adjusted program to the same motion as the calibration motion performed when attaching the tool, reflect the program motion speed, acceleration rate, and touch torque adjustment values on the SETUP TOOL CHANGE screen.

1 Press MENU.
2 Select SETUP.
3 Press F1, [TYPE].
4 Select Tool Change. The Tool Change Setup List screen will be displayed.
5 Select the tool to be adjusted from this list and press F3, DETAIL to display the Tool Change Details Setup screen.
6 Set the motion speed, acceleration rate, and touch torque adjustment values in the shaded sections on the following screen:

```
SETUP TOOL CHANGE
    Tool Change / Group:2  Tool:1      9/9
      1  Comment:               [          ]
      2  Attach signal:             DI [  1]
      3  Identify signal:           DI [  2]
      4  Battery:                       OFF
      5  Calibration motion type:   [TYPE 3]
         Calibration Motion Setup
      6    Calibration Speed(%):        100
      7    Acc Value:                 ACC100
      8    Position register index:  PR[  1]
      9    Touch torque(%):     (+)   5.000

 [ TYPE ]          LIST
```

32.11.2 Reference Position Setup for Calibration Types 5 and 6

Providing a Detection Signal

With calibration types 5 and 6, the position where the detection signal is input is used as the reference position during quick mastering.
You must install a limit switch for providing a detection signal capable of guaranteeing the absolute position of the tool.

Reference Position Setup Program

Create the following program to calculate the reference position based upon the installed detection signal.

NOTE
Only the servo tool axis motion group can be used as the motion group of the program to be used for setting the reference position.
For example, if the servo tool axis motion group is the second group, the motion group is [*,1,*,*,*,*,*,*].

Example Program otion Group=[*,1,*,*,*,*,*,*]

```
1:J P[1:ZERO POS] 10% FINE
2:SKIP CONDITION DI[ 1*1)] =ON
3:J P[2:SWITCH POS] 1% *2) FINE ACC100
   *3)SKIP,
  LBL[1],PR[1]=JPOS
4:END
5:LBL[1]
6:UALM[1]
```

[Explanation]
1: Servo tool axis 0 position. The position need not necessarily be the 0 position.
2: Set turning ON of the provided detection signal as the scan condition.
3: Move the tool to a position where the detection signal turns ON. (Jump to label[1] if the detection signal does not turn ON.)
4: Normal end
5: Jump to here if the detection signal does not turn ON.
6: Generate a user alarm

NOTE
The above program is an example. Create programs to suit the requirements of your system.

Program Motion Adjustment Method

Adjust the value of the shaded section in the example program to adjust program motion.
The motion adjusted here is used as the calibration motion when attaching the tool.

*1) Detection Signal
Assign the installed detection signal to DI[ ] or RI[ ].
*2) Program Motion Speed
To set the same calibration motion that is carried out when attaching the tool, execute this program at the same override (usually100%) as when the attach instruction is executed for the tool used during actual production. For this reason, reduce the program motion speed so that the switch will not be broken even if the program is executed at a high override.
For added safety, increase the program motion speed value successively from 1%.
*3) Acceleration rate
Decrease the acceleration rate if motion is too fast even if the program motion speed is 1%.

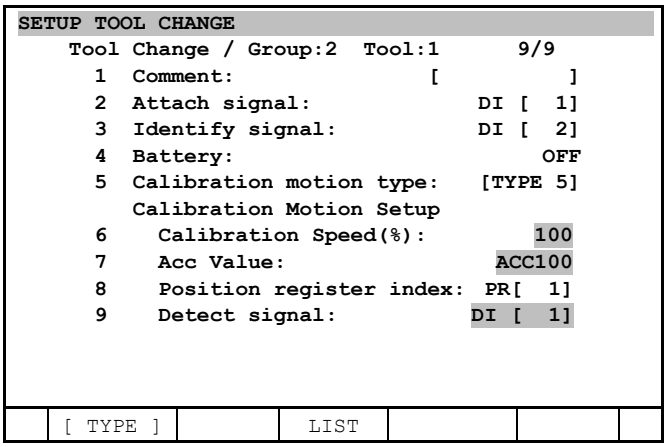
Reference Position Setup

After you have adjusted motion in the program, execute the program to stop the servo tool axis at the position where the detection signal turns ON.
With the robot still in this position, set up the reference position using the procedure in Section 32.10.3.

Tool Change Setup

To make the motion in the newly adjusted program the same as the calibration motion used when attaching the tool, the program motion speed, acceleration rate, and detection signal adjustment values are reflected in the tool change setting.

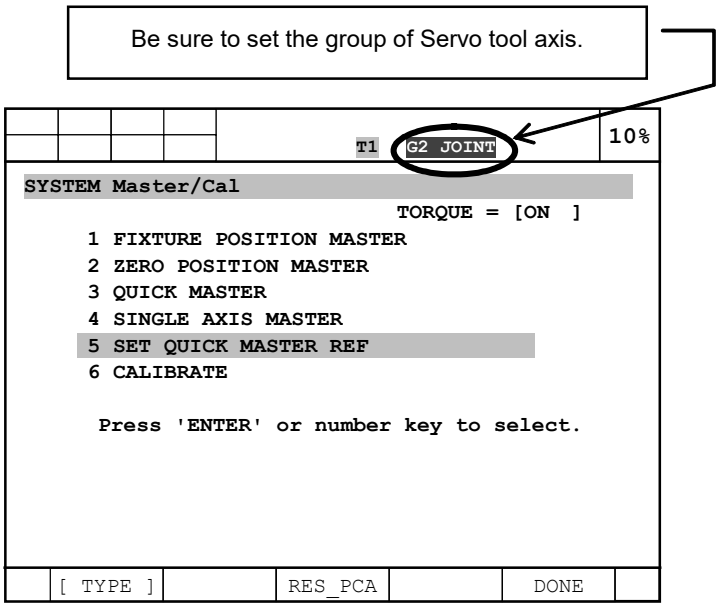
1 Press MENU.
2 Select SETUP.
3 Press F1, [TYPE].
4 Select Tool Change. The Tool Change Setup List screen will be displayed.
5 Select the tool to be adjusted from this list and press F3, DETAIL to display the Tool Change Details Setup screen.
6 Set the motion speed, acceleration rate, and detection signal adjustment values in the shaded sections on the following screen:



32.11.3 Quick Mastering Reference Position Setup

Calibration (quick mastering) is automatically performed when the servo tool is changed. However, the reference positions must be set to each of the tools to enable automatic calibration. The reference positions are set with the robot in a cold start state.
Use the following procedure to set the reference positions:

1 Press FCTN and select CHANGE GROUP. Change the motion group to the desired servo tool group (such as group 2).
2 Press MENU .
3 Select NEXT and then SYSTEM.
4 Press F1, [TYPE].
5 Select Master/Cal.
6 Select SET QUICK MASTER REF and then F4, YES.
7 The message "Quick master reference point is set." will be displayed. This indicates that the reference position setting has been completed.



32.12 TROUBLESHOOTING

32.12.1 Attach Instruction is Executed when the Tool is not Attached.

Symptom

The tool is not actually attached, so a communication alarm such as "SRVO-068 DTERR alarm" occurs on the Pulsecoder. Correct this using Remedy 1 or Remedy 2 depending on the situation.

.

Remedy 1

1 Turn machine lock ON for the servo tool motion group (usually group 2) on the Test Cycle screen.
2 Jog the robot so that the robot and tool are physically attached.
3 Cycle controller power.

Battery-less tool (calibration motion types 1, 2)

4 The tool pulse values and mastering data are lost. Jog the tool to cancel the "SRVO-075 Pulse not established" alarm.
5 Perform tool zero position mastering and calibration manually.

Battery-less tool (calibration motion types 3, 4, 5, 6)

6 The tool pulse values and mastering data are lost. Jog the tool to cancel the "SRVO-075 Pulse not established" alarm.
7 Perform tool zero position mastering and calibration manually. The tool 0 position can be an approximate position.
8 Execute the detach instruction to temporarily detach the tool. Also, physically detach the tool from the robot. Otherwise, the tool will not be calibrated during the next attach operation.
9 Execute the attach instruction again to attach the tool. Calibration this time is performed at the reference position to restore the position.

Battery mounted tool

10 The tool is attached and positioned, so it can be used as it is.

Remedy 2

1 Directly change the data ID to detach the tool. To do this, set the Detach all tools item to ENABLE.
2 Cycle controller power.
3 The robot starts up with the tool in a detached state. Jog the robot to physically attach the robot and tool.
4 Execute the attach instruction to attach the tool.

The tool can be used as it is after it is attached regardless of the battery type.

32.12.2 Robot Stopped during Calibration.

Symptom

This symptom occurs only on battery-less tools. Note that the symptom and remedy vary according to the preset calibration type.

Battery-less tool (calibration motion types 1, 2)

The "TOOL-012 Tool mastering data is lost" alarm occurs.

Battery-less tool (calibration motion types 3, 4, 5, 6)

The alarm does not occur.

Remedy

Calibration motion types 1, 2

1 The tool pulse values and mastering data are lost. So, first move the tool by manual feed to cancel the "SRVO-075 Pulse not established" alarm.
This alarm may not be occurring. If so, skip this step and proceed to step 2.
2 Then, carry out tool zero position mastering and calibration manually.

Calibration motion types 3, 4, 5, 6

1 Tool operation can be resumed as it is.

32.12.3 Calibration Motion Failed.

Symptom

The "TOOL-020 Calibrate motion is failed" alarm occurred.
This symptom occurs only when the calibration motion type is 3, 4, 5 or 6 on a battery-less tool.
This alarm occurs when the touch torque or detection signal conditions are not satisfied during calibration.
In both of the following Remedies 1 and 2, the tool is calibrated again. So, calibration motion will fail again if the touch torque or detection signal conditions are not satisfied. Before you remedy this symptom, make sure that the touch torque or detection signal conditions are correct.

Remedy 1

Perform Steps 6 to 9 in Section 32.11.1 Remedy 1.

Remedy 2

This is the same as Remedy 2 in Section 32.11.1.

32.12.4 Different Tool from that Specified by the Attach Instruction is Attached.

Symptom

The tool is attached to the robot, but the tool's current position is the wrong value. On a battery-less tool, an alarm may cause the robot to stop during calibration.

Remedy

The following describes an example where tool 2 (the actual attached tool) is mistakenly attached to tool 1 (tool No. taught by the attach instruction).

1 In this state, the tool cannot be detached from the robot with the detach instruction. So, directly change the data ID to detach the tool. To do this, set the Detach all tools item to ENABLE.
2 Cycle controller power.
3 The tool starts up in a detached state.
4 Execute the TOOL ATTACH[2] instruction to attach tool 2. In this state, tool 2 should be physically connected to the robot.

If tool 2 is a battery-less tool and the calibration motion type is 1 or 2:
If tool 2 is moved when attached by mistake instead of tool 1, tool 2 will move and be at a position different than when it was detached. So, tool 2 may be calibrated to the wrong position by the attach instruction executed in Step 3 above, resulting in misalignment between its actual and desired positions. In this case, perform tool zero position mastering and calibration manually.

If tool 2 is other than the above:
Tool 2 can be used as it is.

5 Tool 1 can be used as it is by executing the attach instruction to attach the tool.

32.12.5 Attached Tool has been Detached by Mistake (without Using the Detach Instruction).

Symptom

The tool is not actually attached, so a communications alarm such as "SRVO-068 DTERR alarm" occurs on the Pulsecoder. Correct this using Remedy 1 or Remedy 2 depending on the situation.

Remedy 1

This is the same as Remedy 1 in Section 32.11.1

Remedy 2

This is the same as Remedy 2 in Section 32.11.1

32.12.6 Tool Axis of a Detached Tool has Moved.

Symptom

Battery-less tool (calibration motion types 1, 2):
Misalignment has occurred. Misalignment occurs as a result of calibration at the wrong position when the tool is attached with the tool axis having moved to a different position. However, if the amount of axis movement is within the following tolerances, misalignment will not occur.
The allowable errors must fall within the respective following ranges for the linear and rotational axes:

Linear axis: ±(gear ratio/2)mm or less around the detach position
Rotational axis: ±((360/gear ratio)/2)deg or less around the detach position

Other instances:
No problem has occurred.

Remedy

Battery-less tool (calibration motion types 1, 2):
1 Execute the attach instruction to attach the tool.
2 If misalignment occurs, perform tool zero position mastering and calibration manually.

32.12.7 Battery Voltage has Dropped

Symptom

The "SRVO-065 BLAL" alarm has occurred with the tool in an attached state.

Remedy

1 Change the battery with the tool attached.
2 The tool can be used as it is.

32.12.8 Battery Ran Low while the Tool was Detached.

Symptom

The "SRVO-062 BZAL" and "SRVO-075 Pulse not established" alarms occurred when the tool was attached.

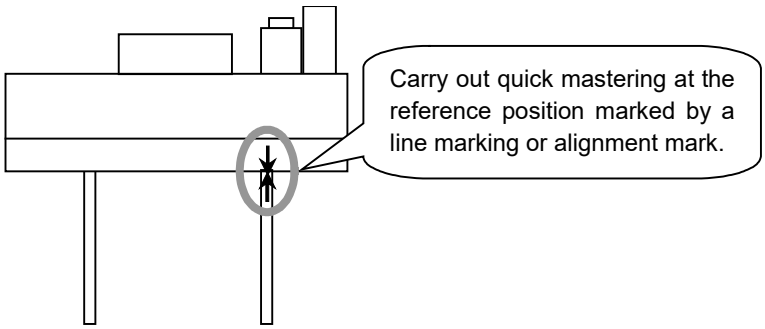
Remedy

If the reference position has not been set:

1 Press F3, RES_PCA on the Calibration screen and select YES.
2 Cycle controller power.
3 Jog the tool to cancel the "SRVO-075 Pulse not established" alarm.
4 Perform tool zero position mastering and calibration manually.

If the reference position has been set:

1 Press F3, RES_PCA on the Calibration screen and select YES.
2 Cycle controller power.
3 Jog the tool to cancel the "SRVO-075 Pulse not established" alarm.
4 Jog the tool to the reference position marked by a line marking or alignment mark.
5 Execute QUICK MASTER on the Calibration screen.



33 OPERATION WITHOUT SHIFT FUNCTION

This function makes it possible to operate without [SHIFT] key in case operation robot by Teach Pendant. Robot operation without shift function (J591) makes it possible to operate both jog operation and test execution operation without [SHIFT] key. And jog operation without shift function (J739) makes it possible to operate only jog operation without [SHIFT] key.

- The spec of the way to move the robot by jog feed differs with the spec that is described in "Procedure 5-1 Moving the robot by jog feed" of "FANUC Robot series OPERATOR's MANUAL(Basic Operation)"(B-84464EN) when "Robot operation without shift key function (J591)" or "Jog operation without shift function (J739)" is installed.
- The spec of the way of test execution differs with the spec that is described in "Procedure 5-7 Step test" and "Procedure 5-8 Continuous test (using the teach pendant)" of "FANUC Robot series OPERATOR's MANUAL(Basic Operation)"(B-84464EN) when "Robot operation without shift key function (J591)" is installed.

Those operations should be replaced with operations that is described following.
Operation without shift function is an optional function.

33.1 JOG OPERATION WITHOUT SHIFT

It is possible to operate jog without [SHIFT] key in case "J591 Robot operation without shift function" or "J739 Jog operation without shift function" is installed to the controller that differs from normal spec. Please submit to following procedure in case you operate jog.

Procedure 33-1 Moving the robot by jog feed (in case robot operation without shift key function or jog operation without shift key function is installed)

Condition

- The teach pendant must be enabled.
- The system must be in the operation enable state.
- Do not enter the operating area. Do not put any obstacles within the work area.
- Option software "J591 Robot operation without shift" or "J739 Jog operation without shift" is installed.

> **⚠WARNING**
> Before you jog the robot be sure that all safety requirements for the work area are satisfied. Otherwise, injury or property damage could occur.

Step

1 Press the [COORD] key to display a desired manual-feed coordinate system on the teach pendant.
2 Press the override key to adjust the jog feed rate displayed on the teach pendant.
3 Hold the teach pendant and press the deadman switch on the back of the teach pendant. Continue pressing the deadman switch during jog feed.
4 Turn on the teach pendant enable switch.

> **NOTE**
> 1 If the deadman switch is released when the teach pendant enable switch is on, an alarm occurs. To reset the alarm, press and hold down the deadman switch again, then press the [RESET] key on the teach pendant.
> 2 If the operator is not accustomed to the operation of the robot or is not sure about

the robot motions, low federate overrides should be set.

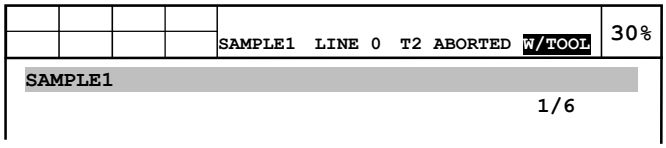
⚠WARNING
The robot starts its motion in the next step. If the jog feed of the robot needs to be stopped in an emergency in the order to avoid danger, the operator should release the deadman's switch or press the emergency stop button.

5 To move the robot by jog feed, press the jog key correspond to the desired robot motion direction. It is not necessary to press the [SHIFT] key, but robot will move if you press jog keys with pressing the [SHIFT] key. And robot is stopped if all jog keys are released but robot isn't stopped even if the [SHIFT] key is released.

NOTE
When the override is FINE or VFINE, press the jog key and release it every time for each motion.

Switch to wrist joint feed

6 Press the [FCTN] key. The function menu is displayed.
7 Select 5,TOGGLE WRIST JOG. The mark,[W/],is displayed to show the wrist joint jog mode. To release this mode, select 5,TOGGLE WRIST JOG again.



Switch to an extended axis

8 Press the [FCTN] key. The function menu will be displayed.
9 Select 4,TOGGLE SUB GROUP. The jog control is switched from the robot standard axes to an extended axis. The control will be returned when it is done.
10 To terminate jog feed, turn off the teach pendant enable switch and release the deadman switch.

33.2 TEST EXECUTION OPERATION WITHOUT SHIFT

It is possible to operate test execution without [SHIFT] key in case "J591 Robot operation without shift function" is installed to the controller that differs from normal spec. Please submit to following procedure in case you operate test execution.

Procedure 33-2 Step test (in case robot operation without shift key function is installed)

Condition

- The teach pendant must be enabled.
- The single-step must be set.(Step LED is ON)
- The system must be in the operation enable state.
- No one must be in the operation area. No obstacle must be placed in the operation area.
- Option software "J591 Robot operation without shift" is installed.

Step

1 Press the [SELECT] key. The program selection screen will be displayed.

2 Select the program to be tested and press the [ENTER] key. The program edit screen will be displayed.
3 Press the STEP key to select the step mode. The STEP LED lights. (Check that the STEP LED lights when the STEP key is pressed.)
4 Move the cursor to the program start line.
5 Press and hold down the deadman switch, then turn on the teach pendant enable switch.

⚠WARNING
The execution of the program instructions starts in the next step. The execution causes the robot to make a motion, witch may produce unpredictable results. The operator should check that no persons and no unnecessary equipment is in the work area and that each part of the protective fence is sound. Otherwise, injury or property damage would occur. If the program needs to be stopped before it terminates, the operator should release the [FWD]/[BWD] key or deadman switch or press the [HOLD] key or emergency stop button.

6 Start the program.
- To perform forward execution of the program, press and hold down the [FWD] key, it is not necessary to press the [SHIFT] key. Press and hold down the [FWD] key until execution of the program is completed. Program is paused if the[FWD] key is released during program execution.
- To perform backward execution of the program, press and hold down [BWD] key, it is not necessary to press the [SHIFT] key. Press and hold down the [BWD] key until execution of the program is completed. Program is paused if the [BWD] key is released during program execution.

7 After one line of the program is executed, the program is halted.
- When a motion instruction is executed, the cursor stops at the executed line. The next time forward execution of the program is performed, the next line of the program is executed.
- When a control instruction is executed, the cursor moves to the next line.

8 To release the step mode, press the [STEP] key.
9 Turn off the teach pendant enable switch, then release the deadman switch.

Procedure 33-3 Continuous test (using the teach pendant) (in case robot operation without shift key function is installed)

Condition

- The teach pendant must be enabled.
- The continuous mode must be set. (Step LED is ON)
- The system must be in the operation enable state.
- No one must be in the operation area. No obstacle must be placed in the operation area.
- Option software “J591 Robot operation without shift” is installed.

Step

1 Press the [SELECT] key. The program selection screen will be displayed.
2 Select the program to be tested and press the [ENTER] key. The program edit screen will be displayed.
3 Set the continuous mode. Check that the STEP LED is off. (If the STEP lamp is on, press the [STEP] key to turn it off.)
4 Move the cursor to the program start line.
5 Press and hold down the deadman switch, then turn on the teach pendant enable switch.

⚠WARNING
The execution of the program instructions starts in the next step. The execution causes the robot to make a motion, witch may produce unpredictable results. The operator should check that no persons and no unnecessary equipment is in the work area and that each part of the protective fence is sound. Otherwise, injury or property damage would occur. If the program needs to be stopped before it terminates, the operator should release the [FWD] key or deadman switch or press the [HOLD] key or the emergency stop button.

6 Press and hold down the [FWD] key (it is not necessary to press the [SHIFT] key). Hold down the [FWD] key until the execution of the program is completed. When the [FWD] key is released, the program is halted. The program is executed to the end, then forcibly terminated. The cursor is returned to the first line of the program.

34 HIGH SPEED SHIFT KEY FUNCTION

It is possible to operate “High speed SHIFT operation” that is operation that switches speed override to specific value temporary at T1 mode by pressing the [SHIFT] key. This function makes it enable to change moving speed of robot high in a moment and enhance operability. It is impossible to use this function at T2 mode.

This function is option software. To use this function, high speed SHIFT key function (J592) is required. In addition, to use high speed shift jog operation, robot operation without shift key function (J591) or jog operation without shift function (J739) is required.

34.1 SETUP FOR HIGH SPEED SHIFT KEY FUNCTION

It is possible to setup the speed override during high speed SHIFT operation. Please input the value of feed rate override (default value is 0) should be during high speed SHIFT operation to the value of “Override value for high speed shift” (MENU → SYSTEM → CONFIG → Override → High speed shift key Override value). It is possible to input 0-100 but 0 means disable high speed SHIFT function.

NOTE
Speed override is clamped to upper limit in case there is setting upper limit of speed override and the value of “Override value for high speed shift” is higher than this value. The speeds at the tool tip and flange surface are limited to 250 mm/sec or slower at T1 mode. So there is no situation that speed exceed 250 mm/sec by this function.

34.2 HIGH SPEED SHIFT JOG OPERATION

It is possible to change moving robot speed high during jog operation by pressing the [SHIFT] key. Please refer to following procedure.

NOTE
This operation is operation as its premise of installing option software “J591 Robot operation without shift function” or “J739 Jog operation without shift”, so this operation differs from normal spec.

Procedure 34-1 High speed SHIFT jog operation

Condition

- 3 Mode switch is T1 mode.
- The value of “Override value for high speed shift” (MENU → SYSTEM → CONFIG → Override → High speed shift key Override value) is 1-100.
- Speed override isn’t either FINE or VFINE.

Step

1 Hold the teach pendant and press the deadman switch on the back of the teach pendant. Continue pressing the deadman switch during jog feed.
2 Turn on the teach pendant enable switch.
3 Release the [SHIFT] key in case the [SHIFT] key is pressed and hold.

⚠WARNING
Next operation changes the speed of robot moving. Next operation must be executed after you ensure safety. Stop robot moving immediately in case the robot moves with speed that you don't forecast.

4 Execute jog operation by pressing a jog key and press the [SHIFT] key during jog operation. Speed override will be change to The value of "Override value for high speed shift" (default value is 100) with back color is changed from green to yellow during the [SHIFT] key is being pressed. This operation is called for "high speed SHIFT jog operation".

NOTE
1 It is impossible to operate high speed SHIFT jog operation in case speed override is "FINE" or "VFINE". Because "FINE" or "VFINE" jog is called inching jog, so these jogs are not sequence jog operation.
2 Override changing operation (ex. Pressing the override key) is canceled during high speed SHIFT operation.

5 High speed SHIFT operation is end and speed override is restored to the value of override before high speed SHIFT operation if the [SHIFT] key is released or jog operation is end during high speed shift operation. But high speed SHIFT operation is end without speed override is restored in case the value of override before high speed SHIFT operation is higher than during high speed SHIFT operation.

34.3 HIGH SPEED SHIFT TEST EXECUTION OPERATION

It is possible to switch speed of robot moving during test execution by pressing the [SHIFT] key in only T1 mode. Please refer to following procedure.

NOTE
This operation is operation as its premise of installing option software "J591 Robot operation without shift function", so this operation differs with normal spec.

Procedure 34-2 High speed SHIFT test execution operation

Condition

- 3 Mode switch is T1 mode.
- The value of "Override value for high speed shift" (MENU → SYSTEM → CONFIG → Override → High speed shift key Override value) is 1-100.

Step

1 Press the [SELECT] key, The program selection screen will be displayed.
2 Select the program to be tested and press the [ENTER] key. The program edit screen will be displayed.
3 Move the cursor to the program start line.
4 Press and hold down the deadman switch, then turn on the teach pendant enable switch.
5 Execute program by Pressing and holding down [FWD]/[BWD] key.
6 Release the [SHIFT] key in case the [SHIFT] key is pressed and hold.

⚠WARNING
Next operation may change the speed of robot moving. Next operation must be executed after you ensure safety. Stop robot moving immediately in case the robot moves with speed that you don't forecast.

7 Press the [SHIFT] key during test execution operation. Speed override will be change to the value of "Override value for high speed shift" (default value is 100) with back color is changed from green to yellow during the [SHIFT] key is being pressed. This operation is called for "high speed SHIFT execution operation".

NOTE
Override changing operation (ex. Pressing the override key) is canceled during high speed SHIFT operation.

8 High speed SHIFT operation is end and speed override is restored to the value of override before high speed SHIFT operation if the [SHIFT] key or the [FWD]/[BWD] key is released during high speed shift operation. But high speed SHIFT operation is end without speed override is restored in case the value of override before high speed SHIFT operation is higher than during high speed SHIFT operation.

35 *i*RDIAGNOSTICS

*i*RDiagnostics function consists of following three functions. To use this function, *i*RDiagnostics option (R765) is required.

Robot Condition Detection	Warn the users when abnormalities are detected in the reducers.
Motion Profiler	Show motion data (OVC, overheat, power consumption) of each program.

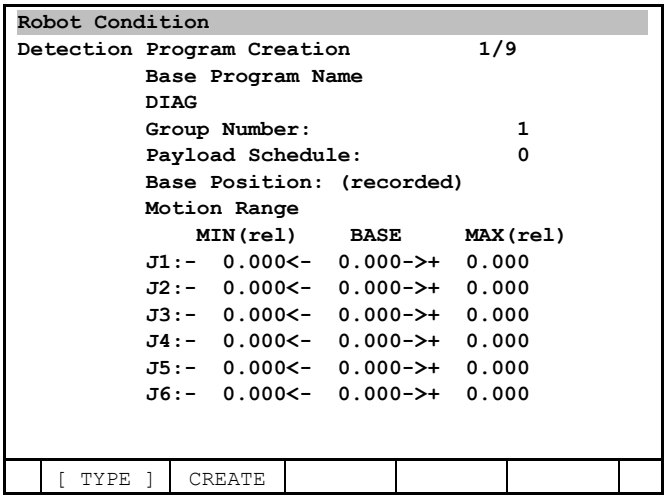
35.1 ROBOT CONDITION DETECTION

- This function detects abnormalities of reducers to help you decide when to perform preventive replacement to minimize downtime due to a problem in the reducers.
- By posting warning, it warns you of reducer abnormalities detected by periodically analyzing motion data.
- Robot Condition Detection requires the creation of a diagnostic program, registration of initial data, and periodic execution of the diagnostic program.
- The diagnostic program is created by giving the base position and allowable motion range.
- This function is not available for some robot models. Contact your local FANUC representative for information about compatible models.
- If a sufficient motion range is not given, motion data cannot be analyzed.
- In the case of impulsion-induced rapid degradation, or failure in certain parts, it might not be possible to detect a reducer failure in advance.
- A failure due to aging may also progress rapidly in the days leading up to failure. Run the diagnostic program periodically, about once a day.

35.1.1 Setup

CREATE
This function requires you to create a diagnostic program using the following procedure.

1 Jog the robot to the base position, where the diagnostic program starts and ends.
2 Press the [MENU] key.
3 Select “UTILITIES”.
4 Press the F1, [TYPE].
5 Select “Robot Condition”.

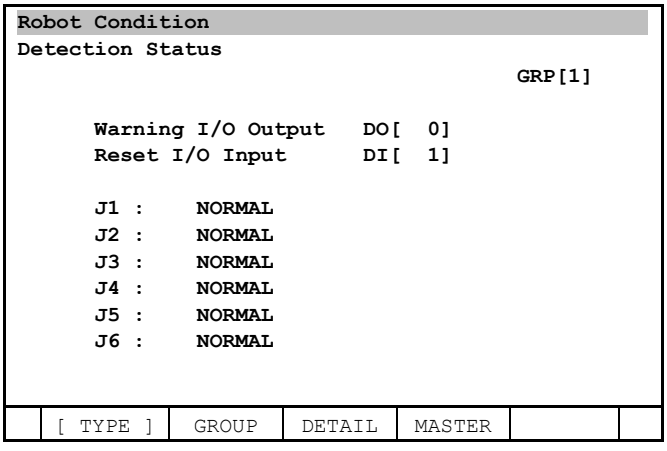


6 Enter the Base Program Name, which is the name of the diagnostic program to be created.
The group number will be added to the base name to create the program name.
The base program name should be 34 letters or fewer.
7 Enter the Group Number, which is the number of the group for analysis.
8 Enter the Payload Schedule, which is the number of the payload setting.
9 Move the cursor over Base Position, and press the F4, RECORD with [SHIFT] key.
(Data ID: $ROBOT_CONDITION.BASE_ANGLE[a, b] / REAL / RW / a: Group number b: Axis)
10 Enter the allowable motion range of each axis, in degrees.
For each axis, enter the allowable motion range relative to the base position, both MIN and MAX.
The robot will move axis by axis, one axis at a time.
An axis without wide enough motion range cannot be analyzed.
An axis with a motion range of 0 degrees will not be analyzed.
(Data ID: $ROBOT_CONDITION.MIN_ANGLE[a, b] / REAL / RW / a: Group number b: Axis)
(Data ID: $ROBOT_CONDITION.MAX_ANGLE[a, b] / REAL / RW / a: Group number b: Axis)11
Set to AUTO mode and set the teach pendant ON/OFF switch to OFF.
12 Set the override to 100%.
13 Press the F2, CREATE with [SHIFT] key.

A program will be created with the group number added to the Base Program Name.

NOTE
- Although the robot does not move during creation, robot motion is simulated inside the controller. Reset alarms and release holds and so forth to enable simulating robot motion with 100% override. Then, create a diagnostic program.
- Keep the override at 100% during creation.
- After creation, run the program with low a override to confirm the safety of the program.

MASTER
Next, perform the following procedure to register the initial diagnostic data.
1 Run the created program with 100% override (run with a low override beforehand to assure safety).
2 Press the [MENU] key.
3 Select “STATUS”.
4 Press the F1, [TYPE].
5 Select “Robot Condition”.



6 If upper right display of the motion GRP[] does not show the group for analysis, press the F2, GROUP and enter the group number.
7 Press the F4, MASTER key.

8 Enter –1 (all axes) when prompted to enter the axis number. However, if you are creating multiple programs for each axis, enter and register the target axes of the programs executed this time in order.
9 A message will ask you to confirm removal of previous data and use of current data as the base data.
"Please confirm to use current data as baseline data? Please note previous history will be removed after master."
Select [YES].

The latest analysis data is registered as the baseline data.
Axes for which no data was registered, shown with *s, were not analyzed due to insufficient motion range.

NOTE
Note the following items that may prevent this function from working properly.
- Do not edit the created diagnostic program.
- Creating a program with the same name will overwrite the old program. If overwritten after performing the MASTER procedure, the function will not work properly. If the program needs to be changed during setup, perform both the CREATE and MASTER procedures.
- Different programs may be created with arbitrary base positions. However, an axis can be analyzed with only one program. With different programs analyzing the same axis, this function will not work properly because of the different characteristics of the data collected. If you try to analyze an axis with a program different from the one used for the MASTER, the following warning will be posted.
DIAG-008 Baseline prog changed (G:g A:a)
where g is the group number and a is the axis number.
- Perform the MASTER procedure only for initial setup and when replacing a reducer.
- If it becomes impossible to use the program for some reason, such as moving the system or changing the robot's load, it will be necessary to perform the CREATE and MASTER. However, if a reducer is already reaching the end of its operating life, this function may not work properly.

35.1.2 Program Execution

After you have created baseline data, the latest data is analyzed to check for any abnormalities each time the diagnostic program is executed with 100% override.
Run the program to detect abnormalities periodically, for example, once a day.
Be sure to run the program under the same conditions as when the baseline data was recorded, such as payload.

If an abnormality is detected, the following warning will be posted,
SRVO-394 Reducer Abnormal (Gg, Aa)
where g is the group number and a is the axis number.
An abnormality is detected in the corresponding axis, so plan replacement accordingly.

35.1.3 Robot Condition Detection Status

You can check the current conditions on the Robot Condition Detection Status screen, as follows:
1 Press the [MENU] key.
2 Select "STATUS".
3 Press the F1, [TYPE] key.
4 Select "Robot Condition".

The Robot Condition Detection Status screen displays the condition of each axis.
Move the cursor to an axis and press F3, DETAIL to display the time the baseline data was registered and the time of the latest analysis data.

If the Warning I/O Output DO is set, the specified DO turns ON when an abnormality is detected.
If the Reset I/O Input DI is set, the Warning I/O Output DO will be cleared (turns OFF) when the specified DI turns ON.

35.1.4 After Replacement

When reducer is replaced, the corresponding baseline data need to be re-registered using the following procedure:

1 Run the diagnostic program with 100% override after you have run the program with low override beforehand to assure safety.
2 Press the [MENU] key.
3 Select “STATUS”.
4 Press the F1, [TYPE].
5 Select “Robot Condition”.
6 If the GRP[] number in the upper right corner is not the group for analysis, press the F2, GROUP and enter the group number.
7 Move the cursor to the axis of replacement and press the F3, DETAIL.
8 Confirm that the time of the Latest data is the time the diagnostic program was run in Step 1.
9 Press the F4 MASTER.
10 When prompted, enter the number of the replacement axis.
11 A message will ask you to confirm that the previous data will be deleted and the current data should be used as the base data.
“Please confirm to use current data as baseline data? Please note previous history will be removed after master.”
Select [YES].
12 Press the F3, DETAIL and confirm that the Base time has been updated.

The new base line data is registered.

35.2 MOTION PROFILER

The Motion Profiler is a motion option that provides important motion information for either over an entire cycle or line by line. The Motion Profiler allows you to easily analyze program performance for a selected program. Since the correspondence between the motion and program line is known, it is easy to adjust the program to achieve the desired motion performance.
The Motion Profiler function provides the following data for each execution cycle:

- Total Cycle Time
- Reducer Load Monitor (max. value)
- Joint Load Monitor (some models only, max. value)
- Reducer Estimated Life (L10 life)
- Power Consumption
- Regenerative Power (if hardware is available)
- Steady-State OVC (the prediction, if continuously run in the same path, non-stop)
- Overheat

The detailed data displayed in bar charts per program execution line includes:

- Execution Time per line
- Max. value of Reducer Load Monitor per line

- Power usage per line

Due to the memory size limitations of the controller, Motion Profiler data from programs that are too long may be overwritten and data may be missing. In this case, consider dividing a very long program into reasonable program sizes.

35.2.1 Setup

Set up the Motion Profiler using the following procedure.

1 Press the [MENU] key.
2 Select "STATUS".
3 Press the F1, [TYPE].
4 Select "Motion Profiler".

The Motion Profiler function is automatically disabled after the selected program is aborted. If you want to analyze the program again, re-enable the function again.
Ensure that the ambient temperature, special hardware option, and robot connection cable (RCC) length for each robot group are set correctly, because the prediction and calculation results are affected by these settings. The results will not reflect those of non-supported groups.
Set the ambient temperature on Motor/Amplifer of the System Config menu.

Table 35.3.1 Motion Profiler Setup menu items

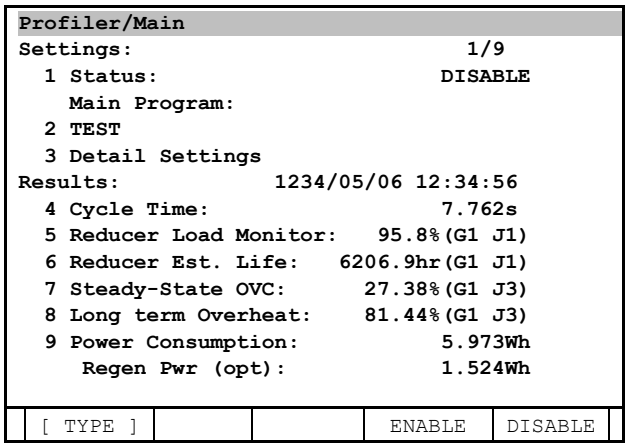
Item			Description
Status			This item is the enable or disable status of the Motion Profiler. This status will change to disable automatically after the main program is aborted.
Main Program			This item is the main TP program to be used by the Motion Profiler for motion analysis.
Detail Settings	Group Settings	Group Number	Group number
		Robot Connection Cable Length	This item is the length of the robot connection cable (RCC) for this robot. This number may be printed on the robot connection cable.
		Hardware Configuration	This item is any special hardware configuration for this robot. For example, motor fan, motor cover, and so forth.
	Data Storage	Device	This item is the device from which the comparison data is stored and read.
		Directory	This item is the directory from which the comparison data is stored and read. If not set, it is the root directory of the device.

35.2.2 Results

You can view the results of Motion Profiler function on either the Status screen or directly on the "related view" from the TP editor.

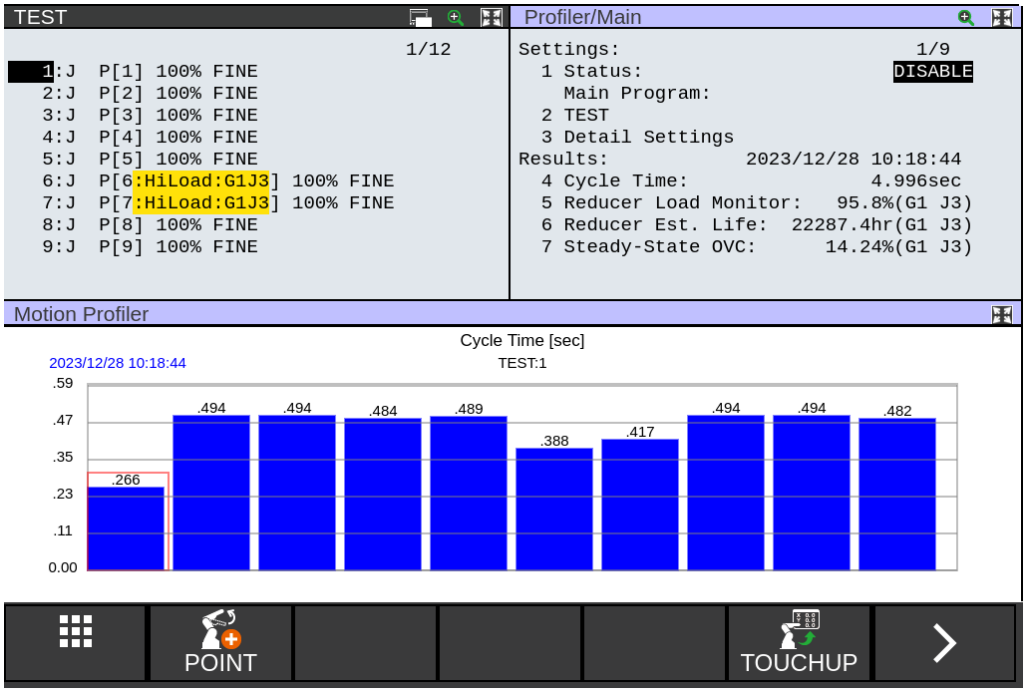
- To display Motion Profiler results on the Status screen:

1 Press the [MENU] key.
2 Select "STATUS".
3 Press the F1, [TYPE].
4 Select "Motion Profiler" to display the Profiler/Main screen.
5 Select data and press the F3, DETAIL to display the detail bar chart with program name: line number as the horizontal axis, or the data for each axis.



- To display Motion Profiler results on the Related view screen:

1 Press the [SELECT] key.
2 Select the TP program for which the Motion Profiler was enabled and press the [ENTER] key.
3 In the TP program editor, press the [i] + the [FCTN] keys to display the Related View menu.
4 Select Motion Profiler to review a triple pane with the TP editor, Motion Profiler Main screen, and Motion Profiler detailed chart. Scrolling the cursor in the TP editor automatically updates the Motion Profiler detailed charts. in the same operation and display can also be performed for the subprograms.



Unsupported axes or data:

If there are unsupported axes or data, “exclusions” will be displayed next to “Results:”. Selecting (exclusions) displays a list of unsupported axes or data.

Comparison Data Output:

You can output Motion Profiler data to a file for comparison, as follows:

1. Display the Profiler/Main screen.
2. Press F2 OUTPUT to save the file containing the Motion Profiler data.

A subdirectory with the Main Program name will be created (if not already created), in the directory of the device specified in the Detail Settings, and a new file will be saved in that directory.
If the file is successfully saved, the output file name will be displayed.
For example, “Output File: ~2018_07_08T12_34_56.dat”
If a file with the same name already exists, it will be overwritten.
If the file fails to be saved, the following message may be displayed:

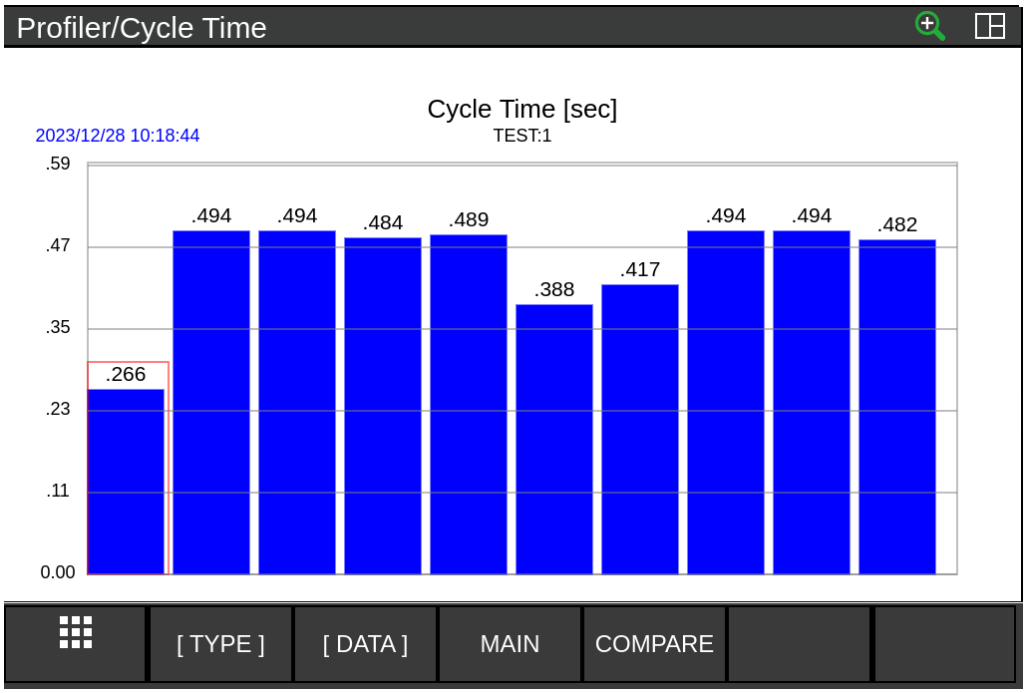
- “Comparison data path is uninitialized”
  The device to output the data is not specified. Specify the output device in the Detail Settings.
- “Subdirectory is too long”
  Due to the file system limitations, the directory names must be 40 characters or less.
- “Directory does not exist”
  The directory specified in the Detail Settings must exist on the device. Create the directory and try again.
- “Invalid device”
  The device specified in the Detail Settings is not accessible. Make sure that the device is correctly specified and inserted in the controller.

NOTE
Do not rename the subdirectory with the Main Program name or any of the data files created by this function, or those files may not be used for comparison.

35.2.2.1 Details

The main page displays the summarized data for the analyzed program. Move the cursor to each item and press the F3, DETAIL to display the detailed data for each item.

Cycle Time
Displays the execution time of the analyzed program. In DETAIL, the bar chart shows the execution time of each program line. Lines with very short execution times, including some commands, will not be displayed.

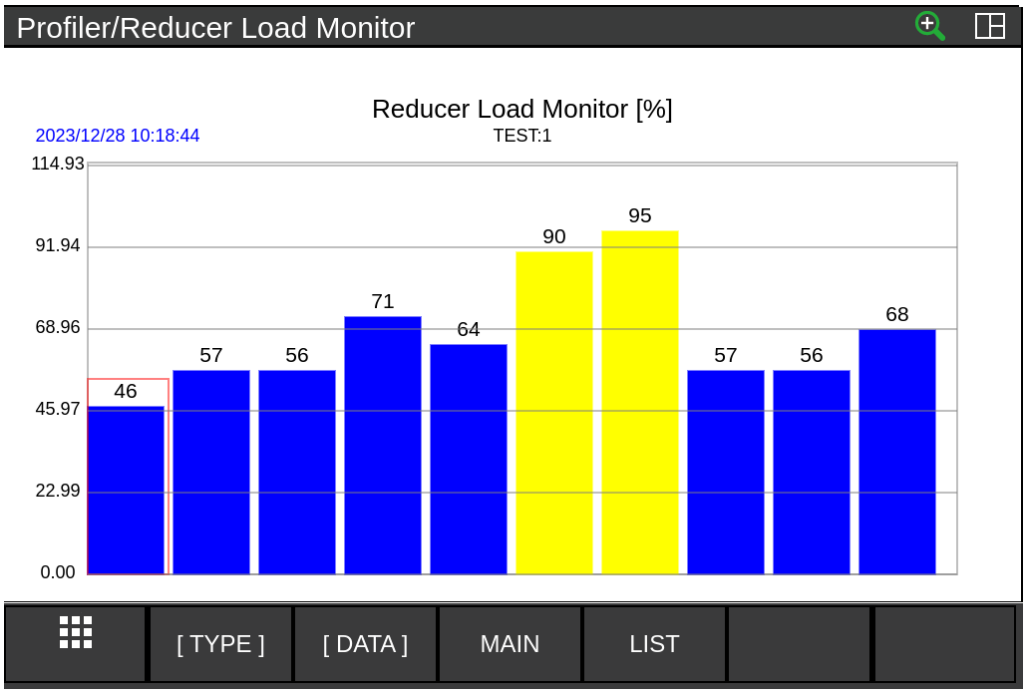


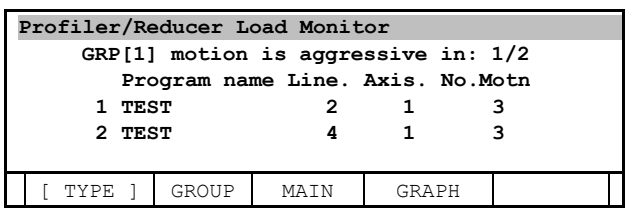
Reducer Load Monitor (shown only for supported models)

Calculates the load level applied to the drive element, such as a reducer or gear, from the motion command. If the load level continues to exceed 100%, the drive element may fail prematurely. If the load level exceeds 100%, edit the program to modify items such as acceleration override (ACC) or speed, to keep the load level below 100%.

The main page displays the maximum value within the analyzed program, and the corresponding group and axis. Pressing the F3, DETAIL displays the maximum values for each program line in a bar chart. Lines with aggressive motion will be colored yellow and red. Yellow indicates that the load level is near the limit, and red indicates that it exceeds the limit. Press the F4, LISTto display a list of aggressive motions with the program name, line number, axis number, and number of motion. If the motion command for the aggressive motion is derived from more than one line, the number of motions item displays number of lines involved. After analysis, if there are aggressive motions, the corresponding line will have the comment “HiLoad:G(group number)” added when EDIT is displayed with the teach pendant ON/OFF switch set to ON.

When the load level is over 100%, a “DIAG-005 Reducer load excess” warning will be posted in addition to the display with this option.





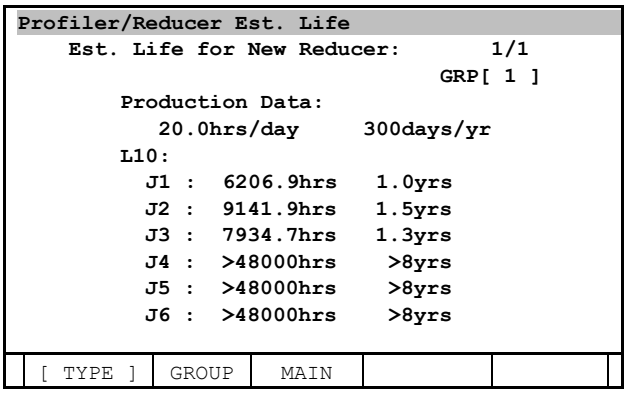
Joint Load Monitor (For certain models only. Shown only for supported models)

Displays the load level applied to the joints. If it exceeds 100%, a “MOTN-522 Joint load excess” alarm will stop the robot. There is no DETAIL screen for Joint Load Monitor.

Reducer Estimated Life (Not applicable for some axes. Shown only for supported models)

Displays the L10 life of a new reducer when the analyzed program is executed repeatedly. The Profiler/Main screen displays the shortest L10 life, with its group and axis. Press the F3, DETAIL to

display the L10 life for each axis. On the DETAIL screen, you can also change the production data, operating hours per day and operating days per year, used for L10 life calculation.
The L10 life is the time after which 10% of the reducers fail. Evaluate the L10 life in the context of the intended use.

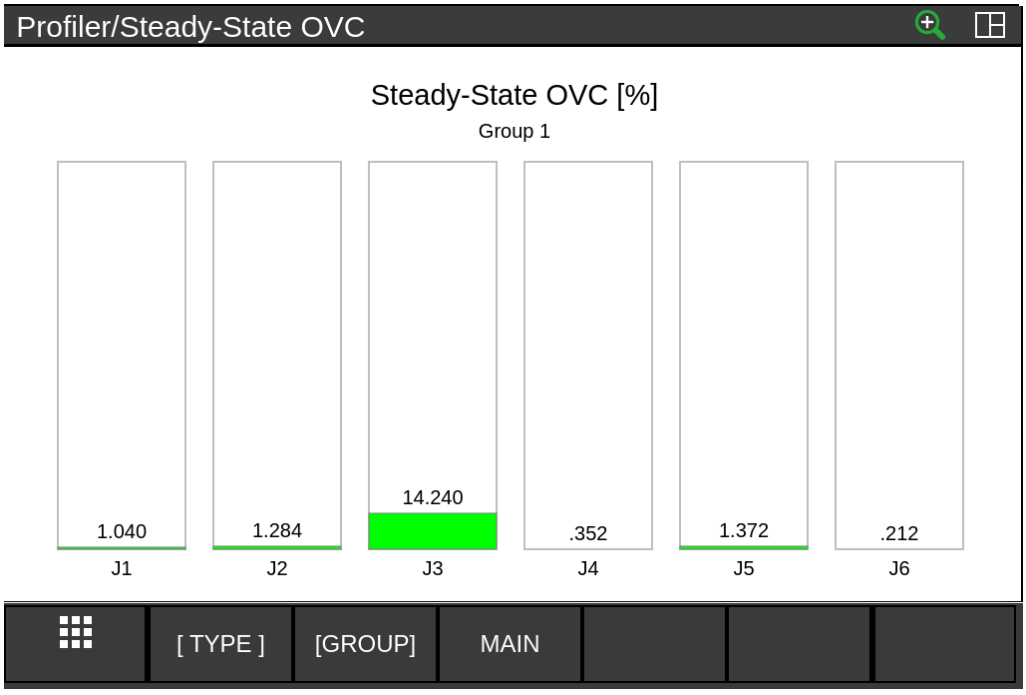


NOTE

Reducer Load Monitor, Joint Load Monitor, M430 Load Monitor, and Reducer Estimated Life require that the payload is set correctly. Otherwise, they will not work properly.
Also, the calculations are done with the assumption that there is no contact between the robot and other objects. Therefore, a program that includes contact and causes an external force (such as pushing a workpiece while mounting) will not be analyzed properly.

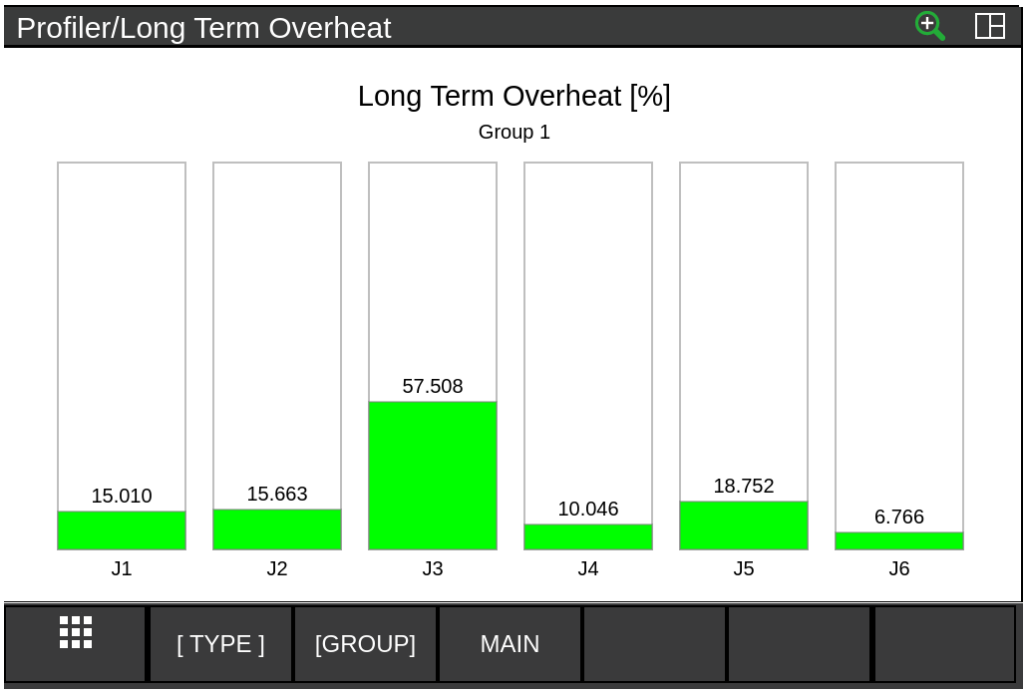
Steady-State OVC

Displays the OVC value estimated when the program was run repeatedly. The OVC value is the ratio of the motor temperature simulated by the software to the alarm threshold. On the Profiler/Main screen, the largest value among the axes is displayed. On the DETAIL screen, the values of each axis are displayed. A value over 100% will cause the "SRVO-046 OVC alarm (Group:i Axis:j)" to occur.



Overheat (for supported models)

The actual overheat alarm (SRVO-067 OHAL2 alarm (Grp:i Ax:j)) is triggered by a sensor in the motor, however, this function compares electric current data to experimental data to estimate whether an overheat alarm will occur. On the Profiler/Main screen, axis with the largest value is displayed. Pressing the F3, DETAIL displays the values for each axis. The values are expressed as a ratio of the calculated value against the thresholds. If the value exceeds 100%, an overheat alarm will occur. The estimation is affected by the Ambient Temperature and Hardware Configuration items set on the System Config and the Detail Settings screen, so be sure to set them accurately.



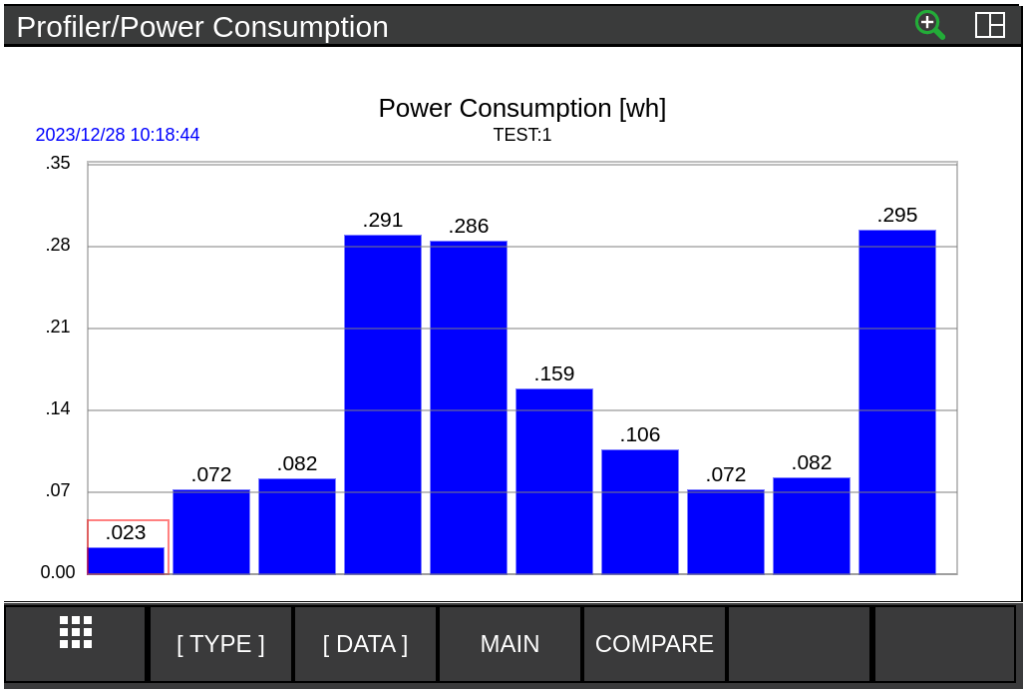
NOTE

Steady-State OVC and Overheat values are estimates and are not guaranteed to be 100% accurate. Use them as a guide only.

Their accuracy improves with appropriate ambient temperature setting and setting conditions closer to actual execution, such as raising the temperature of motor with warm-up. It is advised to execute in the most severe condition among the possibilities.

Power Consumption (for supported models)

Displays the power consumption while executing the analyzed program. Press the F3, DETAIL to display the power consumption per line. The power consumption values are estimated results and not measured data. If unsupported groups are included in motion groups of the program, for example, the results will not include their power consumption. The results are affected by the Robot Connection Cable Length item set on the Detail Settings screen, so be sure to set it accurately.



Regenerative Power

Displays the estimated regenerated energy while executing the analyzed program, if the hardware option is installed. The result is affected by the Robot Connection Cable Length item set on the Detail Settings screen, so be sure to set it accurately. There is no DETAIL screen.

35.2.2.2 Detailed Data Comparison

You can compare the Cycle Time and the Power Consumption data against previously saved data.

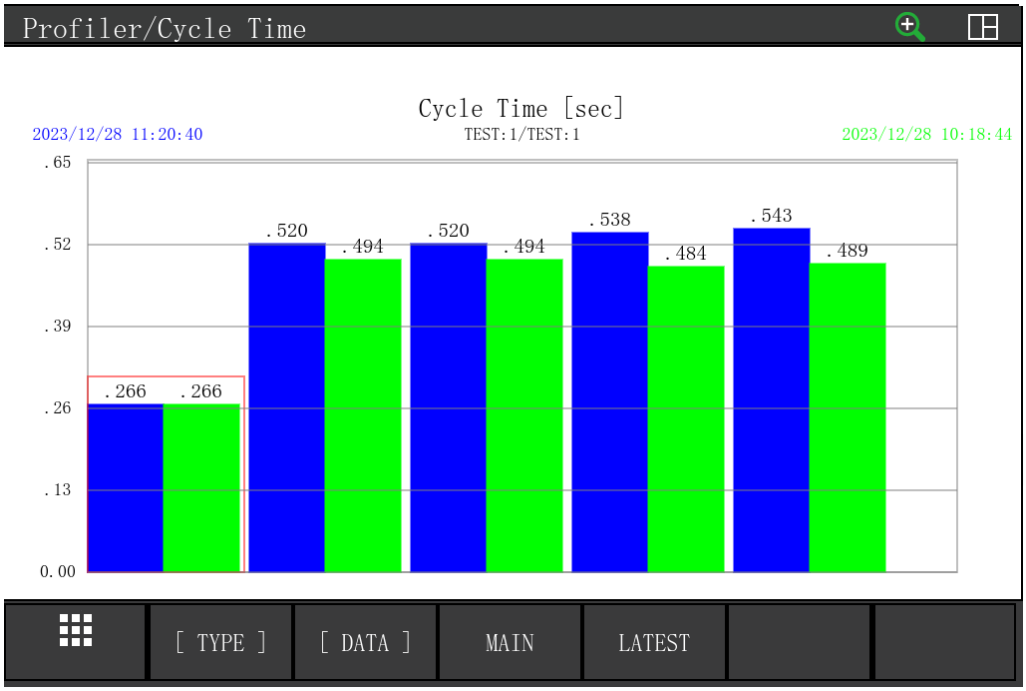
1. On the DETAIL bar charts for the Cycle Time or the Power Consumption, press F4 COMPARE.
   A list of the files in the directory named after the Main Program of the last recorded data under the directory specified on the Detail Settings screen will be displayed.
   The list is displayed by the date the data was recorded.
   Choose a file to compare to.

   While comparing, the comparison data will be shown in green. The program name and line number for the comparison data will be displayed after the program name and line number of the latest data separated by a slash "/".

NOTE
If the program has been edited to add or remove lines, the comparison data may not correspond to the same line in the latest data. If one program is longer than the other, the last data of the shorter data will display "N/A" for missing rows. In the Related View display, only up to the lines present in the current program can be displayed on the Edit screen.

2. To stop comparing, press F4 LATEST

NOTE
If many files exist in the specified directory, it may take some time to display the file list, during which time the teach pendant will appear non-responsive. Consider using different subdirectories if the number of files grows too large.


Profiler/Cycle Time
Cycle Time [sec]
2023/12/28 11:20:40
TEST:1/TEST:1
2023/12/28 10:18:44
.65
.52
.39
.26
.13
0.00
.266
.266
.520
.494
.520
.494
.538
.484
.543
.489
[TYPE]
[DATA]
MAIN
LATEST

36 MENU UTILITY

You can create your own teach pendant menus to help your application run more smoothly. To use your own menus, first perform menu setup, and then call the menu macro program from a teach pendant program and select the desired menu in the parameter.

See Fig. 36 (a) for an example program and example menus.

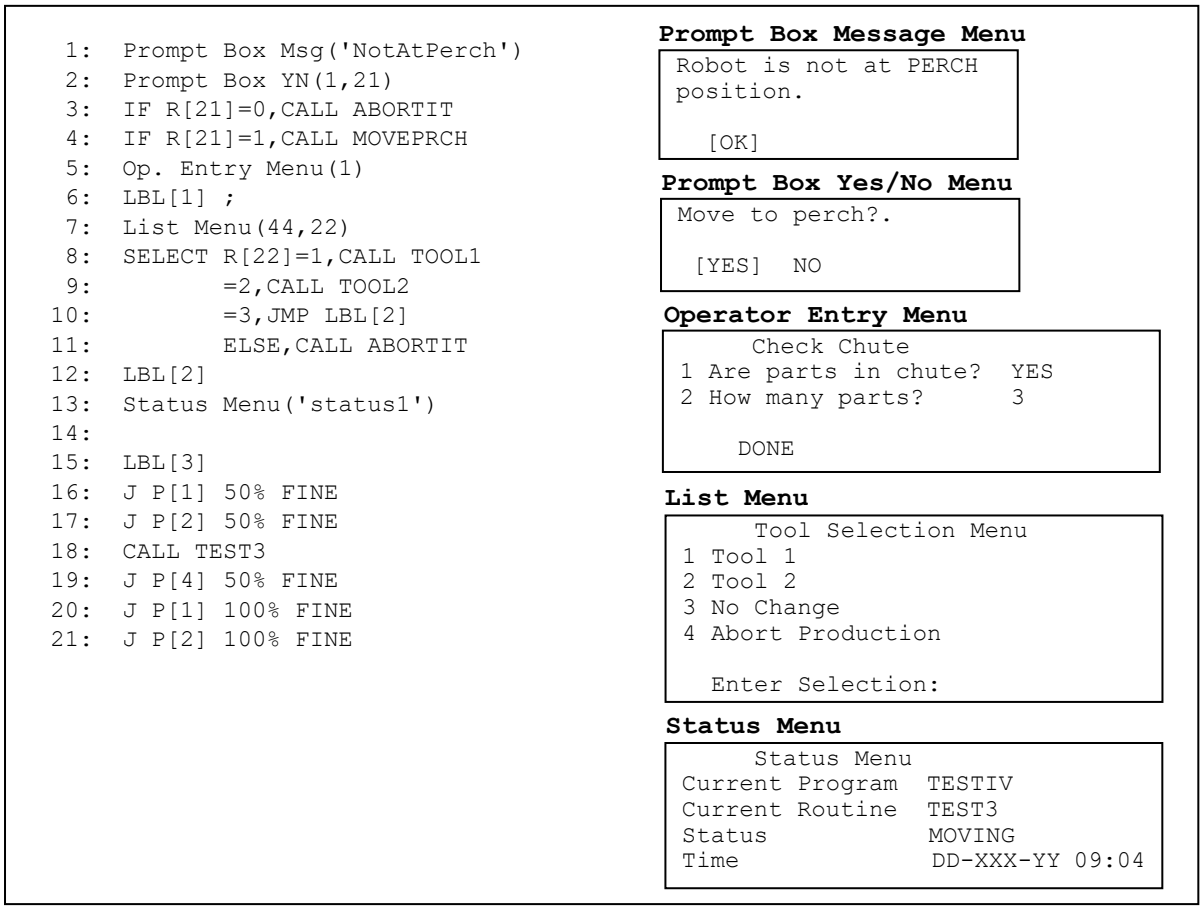


Fig.36 (a) Teach Pendant Menus Example

This section describes how to

- Create menus. Refer to Table 36 (a).
- Include instructions in a teach pendant program to display these menus.

The Teach Pendant Menus option provides some predefined menus that you can use as examples. You can use these sample menus as they are or you can change them as necessary.

Table 36 (a) SETUP Menu Utility Screen Items

ITEM	DESCRIPTION
Prompt box msg	This item is used to make sure the operator is aware of something. When a prompt box message screen is displayed, the teach pendant program pauses until the operator presses ENTER.
Prompt box yes/no	This item asks the operator a question that can be answered YES or NO. When a prompt box yes/no screen is displayed, the teach pendant program pauses until the operator selects YES or NO and presses ENTER.
Select from a list	This item asks the operator to select an item from a list of items. When a list menu is displayed, the teach pendant program pauses until the operator selects an item from the list.
Status menus	This item displays status information to the teach pendant screen. When a status menu is executed, the teach pendant displays the status menu and then continues executing. The values that are displayed are refreshed at least once every second.
Operator entry	This item requires the operator to enter INTEGER, REAL, BOOLEAN, or text values. When an operator entry menu is executed, the program pauses until the operator types the information and then presses F2, DONE.

36.1 PROMPT BOX MESSAGE MENU

A prompt box message menu is used to make sure the operator is aware of something. When a prompt box message screen is displayed, the teach pendant program pauses until the operator presses ENTER. In addition, when the prompt box is displayed, the operator cannot access any other menus. See Figure 36.1 (a) for an example of a prompt box message menu.

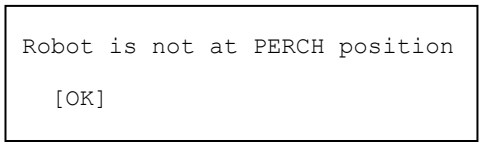


Fig.36.1 (a) Prompt Box Message Menu

Macro: Prompt Box Msg (menu)

A prompt box message menu uses the Prompt Box Msg (menu) macro.

Parameter 1: menu - This is the menu number or menu name. The menu can be chosen by number or name.

Errors

The following errors are possible:

- If the Prompt Box Msg (menu) macro was used in the teach pendant program without a parameter, an alarm is posted and the program is aborted.
- If the selected prompt box message menu does not exist when it is called from a teach pendant program, the “MHND-167 Selected menu does not exist” alarm appears.

Menu File Name

Each prompt box message menu is stored in its own variable file that begins with the prefix "MENU1." The next two characters of the filename are the menu number. For example, the prompt box message menu 1 is stored in the variable file named "MENU101.VR."

Examples
See Prompt Box Msg (menu) Macro Example 1 and Prompt Box Msg (menu) Macro Example 2 for examples of how to use the Prompt Box Msg (menu) instruction in a teach pendant program.

Prompt Box Msg (menu) Macro Example 1
PROG TEST1
1: Prompt Box Msg(1)

Prompt Box Msg (menu) Macro Example 2
PROG TEST2
1: Prompt Box Msg('NotAtPerch')

Setup Procedure
You can enter the text you want displayed in the prompt box from the Prompt box msg menus screen. Table 36.1 (a) lists the Prompt box msg Menu Screen Items. Table 36.1 (b) lists the Prompt box msg DETAIL Menu Screen Items.

Use Procedure 36-1 to create a prompt box message menu.

Table 36.1 (a) Prompt box msg menus Screen Items

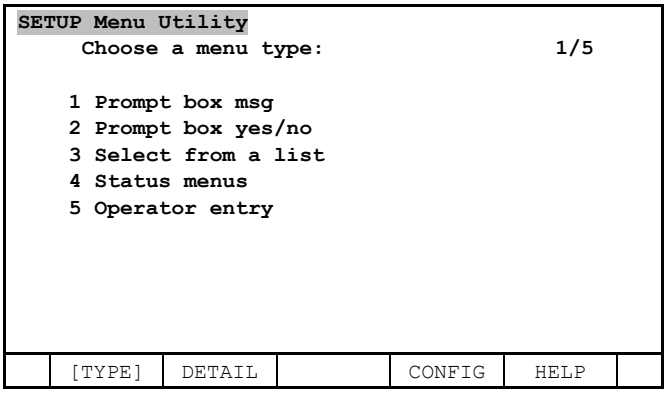
ITEM	DESCRIPTION
Item	This item refers to item numbers (1-99) always in sequence.
Menu number	This item refers to unique menu numbers (1-99).
Menu name	This item is the name of the menu.

Table 36.1 (b) Prompt msg menus DETAIL Screen Items

ITEM	DESCRIPTION
Prompt msg no:	This item refers to unique menu numbers (1-99).
Menu name	This item is the name of the menu.
Line	This item is the line number for the message. Maximum five line.
Text	This item is the display text used in the prompt box.

Procedure 36-1 Setup Prompt Box Msg

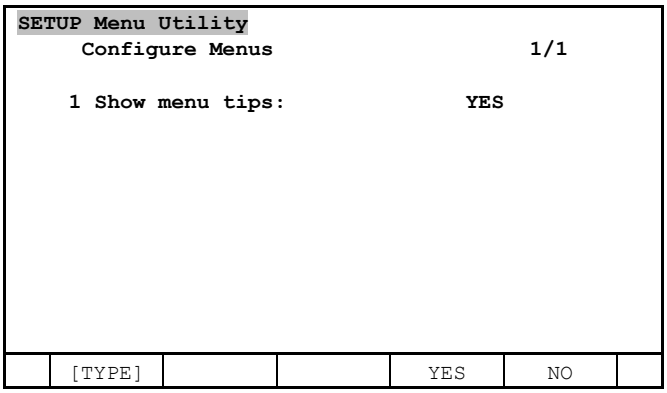
Steps
1 Press MENU.
2 Select SETUP.
3 PressF1, [TYPE].
4 Select Menu Utility. You will see a screen similar to the following.



5 To display menu tips:

Menu tips are prompt boxes that are displayed the first time you press the menu DETAIL key for each menu type. The prompt box contains information that describes the menus, macros, and parameters.

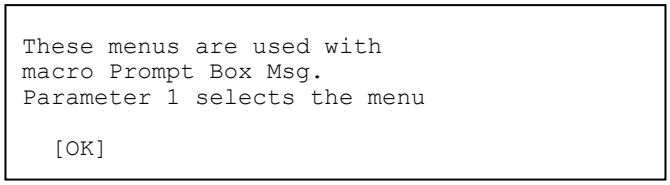
a. Press F4, CONFIG. You will see a screen similar to the following.



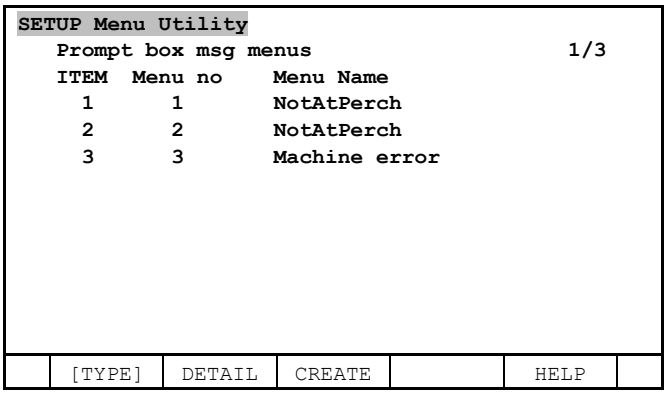
b. To turn off menu tips, press F5, NO.

To turn on menu tips, press F4, YES. You will see a screen similar to the following.

When you are finished viewing menu tips, press ENTER to select OK.



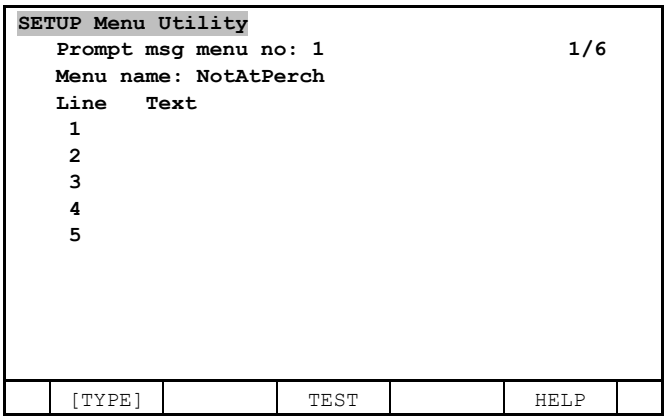
6 Move the cursor to Prompt box msg and press F2, DETAIL. You will see a screen similar to the following.



7 To create a new menu:

a. Press F3, CREATE.

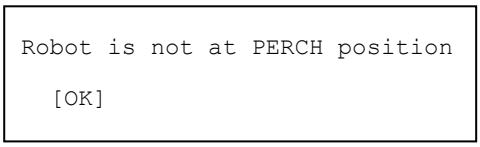
b. Type the number of the new menu and press ENTER. You will see a screen similar to the following.



c. Move the cursor to Menu name and press ENTER. Use the appropriate function keys to name the menu and press ENTER.

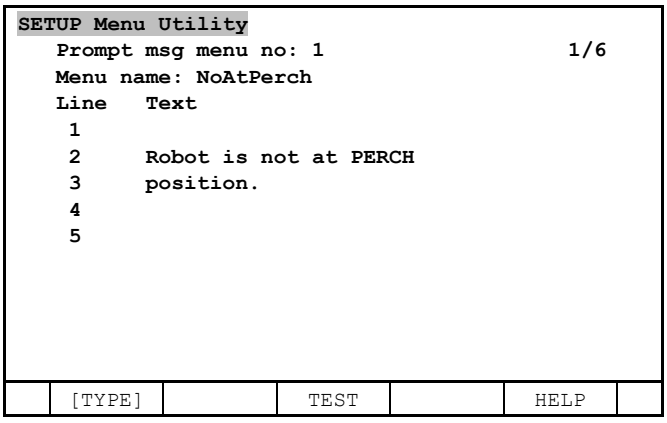
d. Move the cursor to each line and press ENTER. Use the appropriate function keys to type the menu text and press ENTER.

e. To test the menu you created, press F3, TEST. You will see a screen similar to the following.



8 To modify an existing menu:

a. Move the cursor to the menu name and press F2, DETAIL. You will see a screen similar to the following.



b. Move the cursor to the lines you want to change, press ENTER, and press the appropriate function keys to type the text. When you are finished with each line, press ENTER.

9 To save (back up) a menu to the default device:

a. Set the default device. Refer to Procedure 7-1 Changing file I/O devices in Section 7.1 FILE INPUT/OUTPUT UNITS in the *R-50iA CONTROLLER OPERATOR'S MANUAL (Basic*

Operation) (B-84464EN).

b. Move the cursor to the name of the menu you want to save.

c. Press NEXT, >, and press F1, SAVE. The menu will be saved to the default device.

NOTE
- If the file already exists on the default device, you will be prompted whether to replace the existing file with the one you are backing up. To replace the file, select YES. To cancel, select NO.
- You can also back up a menu using FILE, BACKUP APPLICATION. Refer to Section 7.4 SAVING FILES in the *R-50iA CONTROLLER OPERATOR'S MANUAL (Basic Operation)* (B-84464EN) for more information.
- Each prompt box message menu is stored in its own variable file that begins with the prefix " **MENU1**." The next two characters of the filename are the menu number. For example, the prompt box message menu 1 is stored in the variable file named "MENU1**01**.VR."

10 To load (restore) a menu:

a. Move the cursor to the name of the menu you want to load.

b. Press NEXT, >, and press F2, LOAD. The menu will be loaded from the default device.

NOTE
- If the file already exists on the controller, you will be prompted whether to replace the existing file with the one you are loading. To replace the file, select YES. To cancel, select NO.
- You can also load a menu using FILE, LOAD, *.VR. Refer to Section 7.5 LOADING FILES in the *R-50iA CONTROLLER OPERATOR'S MANUAL (Basic Operation)* (B-84464EN) for more information.

11 To clear a menu:

a. Move the cursor to the name of the menu you want to clear.

b. Press NEXT, >, and press F3, CLEAR.

c. Confirm the clear:

- To clear the menu, select YES and press ENTER.
- To cancel the clear, select NO and press ENTER.

12 To copy a menu:

a. Move the cursor to the name of the menu you want to copy.

b. Press NEXT, >, and press F4, COPY.

c. Type the menu number to which you want to copy the menu and press ENTER.

NOTE
If the new menu already exists, you will be prompted whether to replace the existing menu with the one you are copying. To replace the old menu with the new menu, select YES. To cancel the operation, select NO.

36.2 PROMPT BOX YES/NO MENU

A prompt box yes/no menu asks the operator a question that can be answered YES or NO. When a prompt box yes/no screen is displayed, the teach pendant program pauses until the operator selects YES or NO and presses ENTER. In addition, when the prompt box yes/no screen is displayed, the operator cannot access any other menus. See Figure 36.2 (a) for an example of a prompt box yes/no menu.

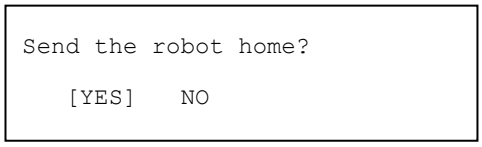


Fig.36.2 (a) Prompt Box Yes/No Menu

Macro: Prompt Box Y/N (menu, answer_reg)

A prompt box yes/no menu uses the Prompt Box YN (menu, answer_reg) macro.

Parameter 1: menu - This is the menu number or menu name. The menu can be chosen by number or name.

Parameter 2: answer_reg - This is the register that will contain the answer from the operator.

- R[answer_reg] = 0 if operator selected NO
- R[answer_reg] = 1 if operator selected YES

Errors

The following errors are possible:

- If the Prompt Box Y/N (menu, answer_reg) macro was used in the teach pendant program without two parameters, an alarm is given and the program is aborted.
- If the selected prompt box yes/no menu does not exist when it is called from a teach pendant program, “MHND-167 Selected menu does not exist” alarm appears.
- If the register answer_reg does not exist, then an alarm is given and the program is aborted.

Menu File Name

Each prompt box yes/no menu is stored in its own variable file that begins with the prefix "MENU2." The next two characters of the filename are the menu number. For example, the prompt box yes/no menu 1 is stored in the variable file named "MENU201.VR."

Examples

See Prompt Box YN (menu, answer_reg) Macro Example 1 and Prompt Box Y/N (menu, answer_reg) Macro Example 2 for examples of how to use the Prompt Box Y/N (menu, answer_reg) instruction in a teach pendant program.

Prompt Box Y/N (menu, answer_reg) Macro Example 1

```
PROG TEST1
1: Prompt Box Y/N(1,2)
2: IF R[2] = 0, CALL ABORTIT
```

Prompt Box Y/N (menu, answer_reg) Macro Example 2

```
PROG TEST2
1: Prompt Box Y/N('NotAtPerch',6)
2: IF R[6] = 1, JMP LBL[5]
```

Setup Procedure

Table 36.2 (a) lists the Prompt Box Yes/No Menu Screen Items. Table 36.2 (b) lists the Prompt Box Yes/No Menu DETAIL Screen Items. Use Procedure 36-2 to create a prompt box yes/no menu.

Table 36.2 (a) Prompt box Yes/No Menu Screen Items

ITEM	DESCRIPTION
Item	This item refers to item numbers (1-99) always in sequence.
Menu number	This item refers to unique menu numbers (1-99).
Menu name	This item is the name of the menu.

Table 36.2 (b) Prompt Box Yes/No Menu DETAIL Screen Items

ITEM	DESCRIPTION
Prompt Y/N menu no:	This item refers to unique menu numbers (1-99).
Menu name	This item is the name of the menu.
Line	This item is the line number for the message. Maximum five lines.
Text	This item is the display text used in the prompt box.

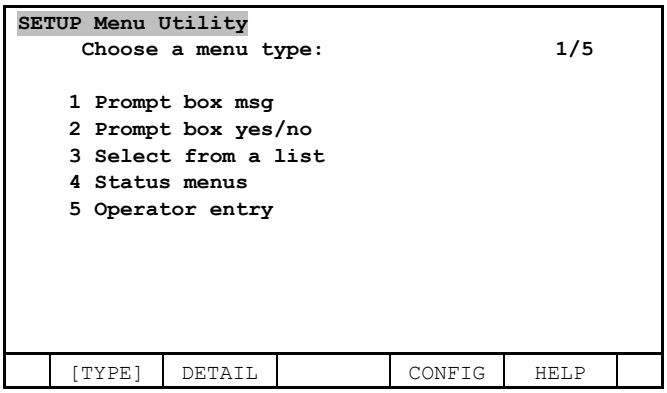
Procedure 36-2 Creating a Prompt Box Yes/No Menu

Conditions

- The default device is set. Refer to Procedure 7-1 Changing file I/O devices in Section 7.1 FILE INPUT/OUTPUT UNITS of the *R-50iA CONTROLLER OPERATOR'S MANUAL (Basic Operation)* (B-84464EN).

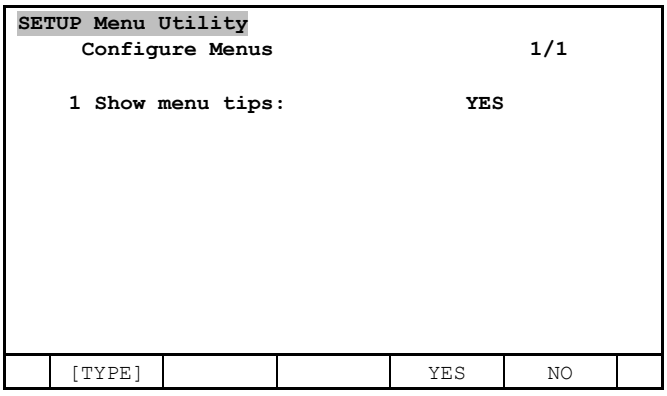
Steps

1 Press MENU.
2 Select SETUP.
3 PressF1, [TYPE].
4 Select Menu Utility. You will see a screen similar to the following.

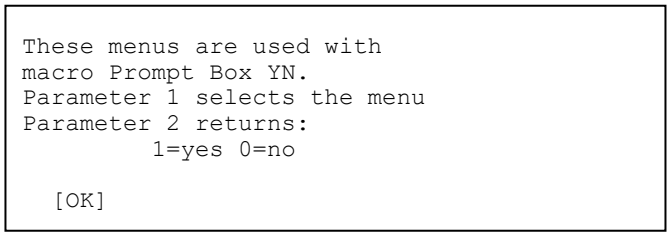


5 To display menu tips:
Menu tips are prompt boxes that are displayed the first time you press the menu detail key for each menu type. The prompt box contains information that describes the menus, macros, and parameters.

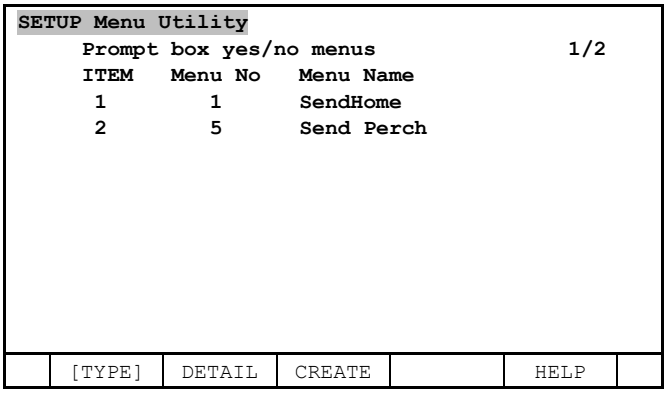
a. Press F4, CONFIG. You will see a screen similar to the following.



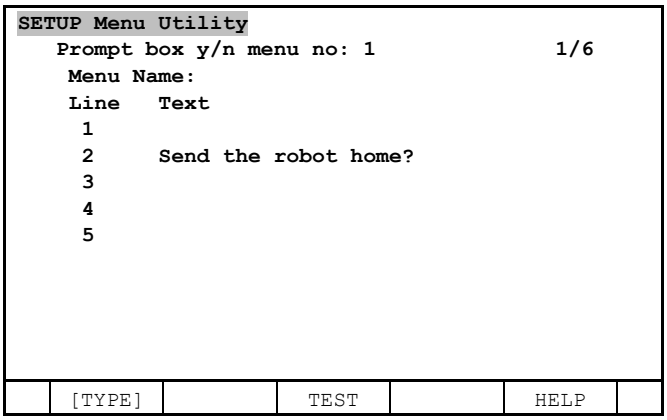
b. To turn off menu tips, press F5, NO.
To turn on menu tips, press F4, YES. You will see a screen similar to the following
When you are finished viewing menu tips, press ENTER to select OK.



6 Move the cursor to Prompt box yes/no and press F2, DETAIL. You will see a screen similar to the following.



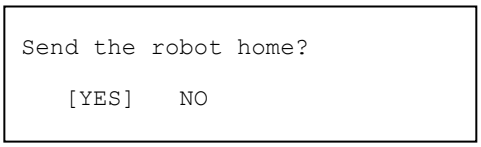
7 To create a new menu:
a. Press F3, CREATE.
b. Type the number of the new menu and press ENTER. You will see a screen similar to the following.



c. Move the cursor to Menu Name and press ENTER. Use the appropriate function keys to name the menu and press ENTER.

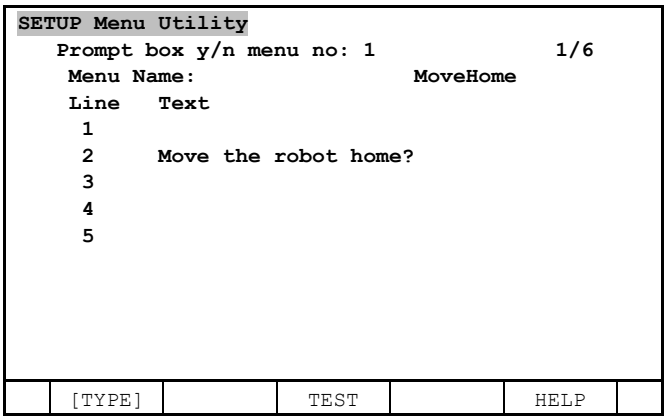
d. Move the cursor to each line and press ENTER. Use the appropriate function keys to type the menu text and press ENTER.

e. To test the menu you created, press F3, TEST. You will see a screen similar to the following.



8 To modify an existing menu:

a. Move the cursor to the Menu Name and press F2, DETAIL. You will see a screen similar to the following.



b. Move the cursor to the lines you want to change, press ENTER, and press the appropriate function keys to type the text. When you are finished with each line, press ENTER.

9 To save (back up) a menu to the default device:

a. Set the default device. Refer to Procedure 7-1 Changing file I/O devices in Section 7.1 FILE INPUT/OUTPUT UNITS in the *R-50iA CONTROLLER OPERATOR'S MANUAL (Basic Operation)* (B-84464EN).

b. Move the cursor to the name of the menu you want to save.

c. Press NEXT, >, and press F1, SAVE. The menu will be saved to the default device.

NOTE
- If the file already exists on the default device, you will be prompted whether to replace the existing file with the one you are backing up. To replace the file, select YES. To cancel, select NO.
- You can also back up a menu using FILE, BACKUP APPLICATION. Refer to Section 7.4 SAVING FILES in the *R-50iA CONTROLLER OPERATOR'S MANUAL (Basic Operation)* (B-84464EN) for more information.
- Each prompt box yes/no menu is stored in its own variable file that begins with the prefix " **MENU2**." The next two characters of the filename are the menu number. For example, the prompt box yes/no menu 1 is stored in the variable file named "MENU2**01**.VR."

10 To load (restore) a menu:

a. Move the cursor to the name of the menu you want to load.

b. Press NEXT, >, and press F2, LOAD. The menu will be loaded from the default device.

NOTE
- If the file already exists on the controller, you will be prompted whether to replace the existing file with the one you are loading. To replace the file, select YES. To cancel, select NO.
- You can also load a menu using FILE, LOAD, *.VR. Refer to Section 7.5 LOADING FILES in the *R-50iA CONTROLLER OPERATOR'S MANUAL (Basic Operation)* (B-84464EN) for more information.

11 To clear a menu:

a. Move the cursor to the name of the menu you want to clear.

b. Press NEXT, >, and press F3, CLEAR.

c. Confirm the clear:
   - To clear the menu, select YES and press ENTER.
   - To cancel the clear, select NO and press ENTER.

12 To clear a menu:

a. Move the cursor to the name of the menu you want to copy.

b. Press NEXT, >, and press F4, COPY.

c. Type the menu number to which you want to copy the menu and press ENTER.

NOTE
If the new menu already exists, you will be prompted whether to replace the existing menu with the one you are copying. To replace the file, select YES. To cancel, select NO.

36.3 LIST MENU

A list menu asks the operator to select an item from a list of items. When a list menu is displayed, the teach pendant program pauses until the operator selects an item from the list. See Figure 36.3 (a) for an example of a list menu.

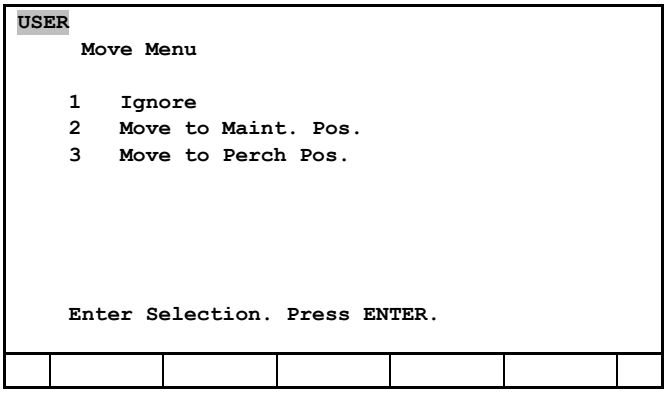


Fig.36.3 (a) List Menu

Macro: List Menu (menu, answer_reg)

A list menu uses the List Menu (menu, answer_reg) macro.

Parameter 1: menu - This is the menu number or menu name. The menu can be chosen by number or name.

Parameter 2: answer_reg - This is the register that will contain the answer from the operator.

- R[answer_reg] = item number the operator selected

Display

To display a list menu you create, use the List Menu macro. The menu will be displayed automatically on the USER screen. If the operator changes the teach pendant display to another screen during teach pendant execution of this menu but before making the selection, he can return to this screen by pressing MENU and selecting USER.

Setup

In a list menu, you can specify

- Up to eight list items from which to choose
- A macro program to execute when the item is selected, if desired; this is called the ActionTP program.

NOTE
The ActionTP program is a teach pendant program that is executed when the corresponding item number is selected. Do not use ActionTP teach pendant programs that use menu macros.

Errors

The following errors are possible:

- If the List Menu (menu, answer_reg) macro was used in a teach pendant program without two parameters, an alarm is given and the program is aborted.
- If the selected list menu does not exist when it is called from a teach pendant program, “MHND-167 Selected menu does not exist” alarm appears.

- If the register answer_reg does not exist, then an alarm is given and the program is aborted.

Menu File Name
Each list menu is stored in its own variable file that begins with the prefix "MENU3 ." The next two characters of the filename are the menu number. For example, the list menu 1 is stored in the variable file named "MENU301.VR."

Examples
See List Menu (menu, answer_reg) Macro Example 1 and List Menu (menu, answer_reg) Macro Example 2 for examples of how to use the List Menu (menu, answer_reg) macro instruction in a teach pendant program.

List Menu (menu, answer_reg) Macro Example 1
```
PROG TEST1
1: List Menu(2, 21)
2: SELECT R[21]=1,CALL TOOL1
3:              =2,CALL TOOL2
4:              =3,JMP LBL[2]
5:              ELSE,CALL ABORTIT
```

List Menu (menu, answer_reg) Macro Example 2
```
PROG TEST2
1: List Menu('ErrorRecov',2)
2: SELECT R[2] =1,JMP LBL[2]
3:             =2,CALL MOVEMANT
4:             =3,CALL MOVEPRCH
5:             ELSE,CALL ABORTIT
```

Setup Procedure
Table 36.3 (a) lists the Select from a list Screen setup items. Table 36.3 (b) lists the Select from a list DETAIL Screen setup items.

Use Procedure 36-3 to create a list menu.

Table 36.3 (a) Select from a list Screen Items

ITEM	DESCRIPTION
Item	This item refers to item numbers (1-99) always in sequence.
Menu number	This item refers to unique menu numbers (1-99).
Menu name	This item is the name of the menu.

Table 36.3 (b) List Menu DETAIL Screen Items

ITEM	DESCRIPTION
List menu no:	This item refers to unique menu numbers (1-99).
Menu name	This item is the name of the menu.
Line	This item is the line number for the message. Maximum eight lines.
Text	This item is the display text used in the prompt box.
ActionTP	This item is the TP program you want to run when the corresponding item is selected.
Title	This item is the title for the message.
PROMPT	This item allows you to enter a selection.

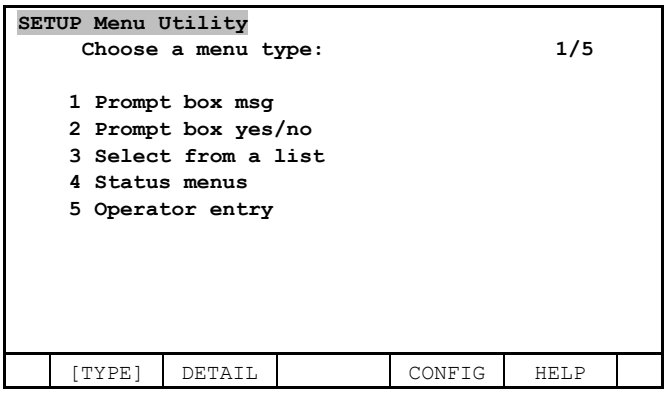
Procedure 36-3 Creating a List Menu

Conditions

- The default device is set. Refer to Procedure 7-1 Changing file I/O devices in Section 7.1 FILE INPUT/OUTPUT UNITS in the *R-50iA CONTROLLER OPERATOR'S MANUAL (Basic Operation)* (B-84464EN).

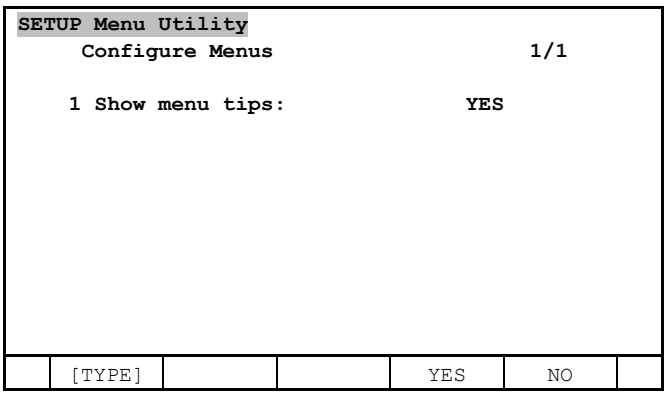
Steps

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



5 To display menu tips:
Menu tips are prompt boxes that are displayed the first time you press the menu DETAIL key for each menu type. The prompt box contains information that describes the menus, macros, and parameters.

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



b. To turn off menu tips, press F5, NO.
To turn on menu tips, press F4, YES. You will see a screen similar to the following.
When you are finished viewing menu tips, press ENTER to select OK.

```
Menus are used with macros
List Menu + Cheeck Cycle Stop.
Parameter 1 selects the menu.
Parameter 2 returns the
operators choice. (0=ERROR)

  [OK]
```

6 Move the cursor to Select from a list and press F2, DETAIL. You will see a screen similar to the following.

```
SETUP Menu Utility
     Select from a list menus                1/2
     ITEM   Menu No    Menu Name
      1         1      MoveMenu
      2        11      Cycle Stop

 [TYPE]  DETAIL  CREATE          HELP
```

7 To create a new menu:

a. Press F3, CREATE.

b. Type the number of the new menu and press ENTER. You will see a screen similar to the following.

```
SETUP Menu Utility
     LIST menu no:      1                   1/11
      Menu name:  MoveMenu
     Line    Text                        ActionTP
     TITLE: Move Menu
     1     Ignore
     2     Move to Maint. Pos.         MOVEMAINT
     3     Move to Perch Pos.          MOVEPRCH
     4
     5
     6
     7
     8
     PROMPT: Choose one and press ENTER:

 [TYPE]  [ALPH]  TEST          HELP
```

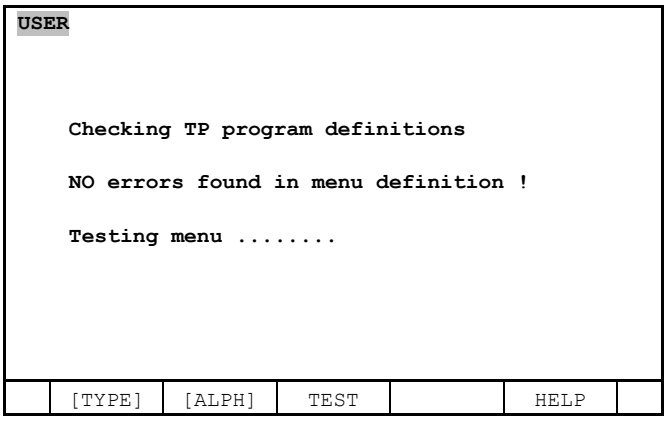
c. Move the cursor to Menu name and press ENTER. Use the appropriate function keys to name the menu and press ENTER.

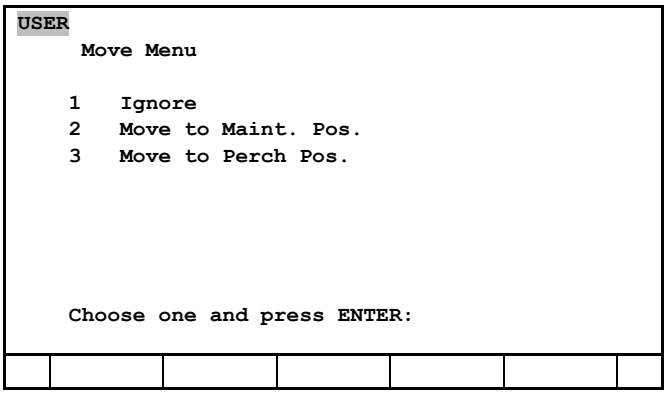
d. Move the cursor to each item and press ENTER. Use the appropriate function keys to type the list item text and press ENTER.

NOTE
The ActionTP program is a teach pendant program that is executed when the corresponding item number is selected. Do not use ActionTP teach pendant programs that use menu macros.

e. If you want to associate an ActionTP program with a list item, move the cursor to the item and
   - Press F2, [ALPH], use the appropriate function keys to type the teach pendant name, and press ENTER, or
   - Press F4, [CHOICE], choose the name of the teach pendant program, and press ENTER.

f. To test the menu you created, press F3, TEST.
   The list menu's display variables are checked to make sure that any ActionTP program listed exists.
   If no errors were found, you will see the following screen and then the menu will be displayed.





   If errors were found, they will be listed on the teach pendant screen and the menu will not be displayed. The previous menu will be displayed. Correct the ActionTP program names and press F3, TEST, again.

g. To test the ActionTP program, move the cursor to the ActionTP you want to test, press and continue pressing the TP Enabling Device (Deadman switch), turn the teach pendant ON/OFF switch to ON, and press SHIFT and F3, TEST.

8 To modify an existing menu:

a. Move the cursor to the menu name and press F2, DETAIL. You will see a screen similar to the following.

```
SETUP Menu Utility
     LIST menu no:      1                    1/11
      Menu name:  MoveMenu
     Line    Text                          ActionTP
     TITLE: Move Menu
     1      Ignore
     2      Move to Maint. Pos.           MOVEMAINT
     3      Move to Perch Pos.            MOVEPRCH
     4
     5
     6
     7
     8
     PROMPT: Choose one and press ENTER:

   | [TYPE] | [ALPH] | TEST |  | HELP |
```

b. Move the cursor to the lines you want to change, press ENTER, and press the appropriate function keys to type the text. When you are finished with each line, press ENTER.

9 To save (back up) a menu to the default device:

a. Set the default device. Refer to Procedure 7-1 Changing file I/O devices in Section 7.1 FILE INPUT/OUTPUT UNITS in the *R-50iA CONTROLLER OPERATOR'S MANUAL (Basic Operation)* (B-84464EN).

b. Move the cursor to the name of the menu you want to save.

c. Press NEXT, >, and press F1, SAVE. The menu will be saved to the default device.

NOTE
- If the file already exists on the default device, you will be prompted whether to replace the existing file with the one you are backing up. To replace the file, select YES. To cancel, select NO.
- You can also back up a menu using FILE, BACKUP APPLICATION. Refer to Section 7.4 SAVING FILES of the *R-50iA CONTROLLER OPERATOR'S MANUAL (Basic Operation)* (B-84464EN) for more information.
- Each list menu is stored in its own variable file that begins with the prefix "**MENU3**." The next two characters of the filename are the menu number. For example, the list menu 1 is stored in the variable file named "MENU3**01**.VR."

10 To load (restore) a menu:

a. Move the cursor to the name of the menu you want to load.

b. Press NEXT, >, and press F2, LOAD. The menu will be loaded from the default device.

NOTE
- If the file already exists on the controller, you will be prompted whether to replace the existing file with the one you are loading. To replace the file, select YES. To cancel, select NO.
- You can also load a menu using FILE, LOAD, *.VR. Refer to Section 7.5 LOADING FILES in the *R-50iA CONTROLLER OPERATOR'S MANUAL (Basic Operation)* (B-84464EN) for more information.

11 To clear a menu:
   a. Move the cursor to the name of the menu you want to clear.
   b. Press NEXT, >, and press F3, CLEAR.
   c. Confirm the clear:
      - To clear the menu, select YES and press ENTER.
      - To cancel the clear, select NO and press ENTER.

12 To copy a menu:
   a. Move the cursor to the name of the menu you want to copy.
   b. Press NEXT, >, and press F4, COPY.
   c. Type the menu number to which you want to copy the menu and press ENTER.

NOTE
If the new menu already exists, you will be prompted whether to replace the existing menu with the one you are copying. To replace the file, select YES. To cancel, select NO.

36.4 STATUS MENU

A status menu displays status information to the teach pendant screen. When a status menu is executed, the teach pendant displays the status menu and then continues executing. The values that are displayed are refreshed at least once every second. This is a read-only menu.

See Figure 36.4 (a) for an example of a status menu.

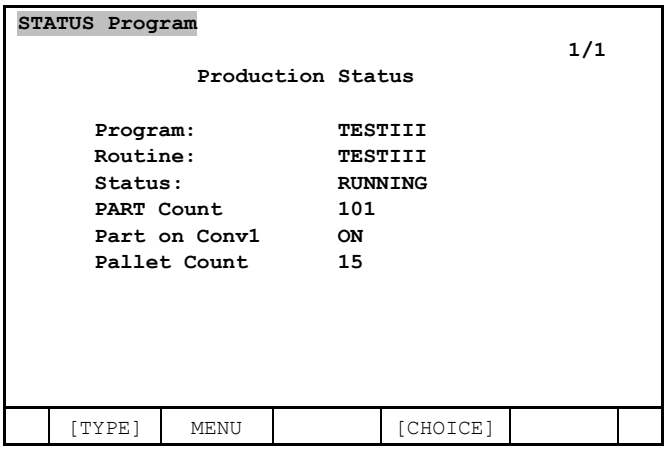


Fig.36.4 (a) Status Menu

Macro: Status Menu (menu)
A status menu uses the Status Menu (menu) macro.

Parameter 1: menu - This is the menu number or menu name. The menu can be chosen by number or name.

Display
When a status menu is executed, the teach pendant displays the status menu and then continues executing. The values that are displayed are refreshed.

A status menu can also be displayed as follows:

1. Press MENU.
2. Select STATUS
3. Press F1, [TYPE].
4. Select Prod. Status.

Several status menus can be displayed. If the status menu you want to see is not the one displayed, either press F2, MENU, and then select the number of the status menu you want to display, or press F4, [CHOICE], and select the menu by name.

Errors

The following errors are possible:

- If the Status Menu macro was used in a teach pendant program without a parameter, an alarm is displayed. However, the program is not aborted and the teach pendant screen is not changed.
- If a menu that was selected by name does not exist, or the menu is the wrong kind of menu, an alarm is given. However, the program is not aborted and the teach pendant screen is not changed.
- If the selected status menu does not exist when it is called for in a teach pendant program by number, an alarm is given and the teach pendant screen is changed to the default status menu, shown in Figure 36.4 (b).

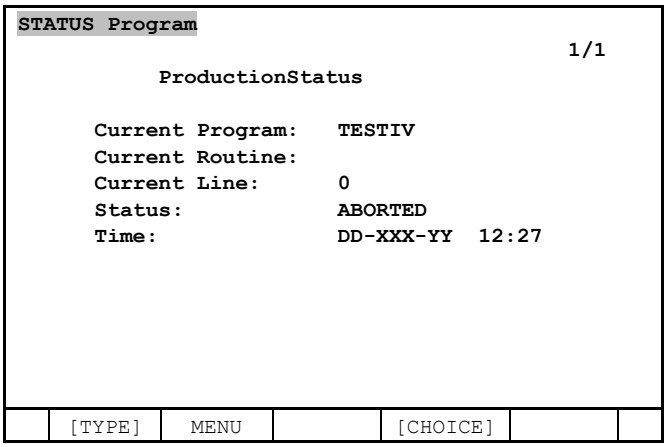


Fig.36.4 (b) Default Status Menu

Menu File Name

Each status menu is stored in its own variable file that begins with the prefix "MENU4." The next two characters of the filename are the menu number. For example, the status menu 1 is stored in the variable file named "MENU401.VR."

Setup

In a status menu, you can specify

- Up to nine status items
- The names of the variables that correspond to the status items, and detailed variable information

Table 36.4 (a) lists and describes the variable detail information.

Table 36.4 (a) Status Menu Variable Detail Information

<table>
<tr><th>ITEM</th><th colspan="3">DESCRIPTION</th></tr>
<tr><td>Display Text</td><td colspan="3">This item allows you to specify the text that will be displayed for the status item. You can type up to 17 characters of display text.</td></tr>
<tr><td rowspan="17">Variable Type</td><td colspan="3">This item allows you to specify the kind of variable for the status item:</td></tr>
<tr><td>Type of Variable</td><td>What to Define</td><td>Data Display Type</td></tr>
<tr><td>NONE</td><td>none</td><td>none</td></tr>
<tr><td>R[ ]</td><td>REG Number</td><td>INTEGER, REAL, BOOLEAN</td></tr>
<tr><td>DIN[ ]</td><td>I/O Number</td><td>INTEGER, BOOLEAN, TEXT</td></tr>
<tr><td>DOUT[ ]</td><td>I/O Number</td><td>INTEGER, BOOLEAN, TEXT</td></tr>
<tr><td>RDI[ ]</td><td>I/O Number</td><td>INTEGER. BOOLEAN, TEXT</td></tr>
<tr><td>RDO[ ]</td><td>I/O Number</td><td>INTEGER, BOOLEAN, TEXT</td></tr>
<tr><td>GIN[ ]</td><td>I/O Number</td><td>INTEGER</td></tr>
<tr><td>GOUT[ ]</td><td>I/O Number</td><td>INTEGER</td></tr>
<tr><td>KAREL VAR</td><td>Prog. Name, Var. Name</td><td>INTEGER, REAL, BOOLEAN, TEXT</td></tr>
<tr><td>TIME</td><td>none</td><td>special type - displays current time</td></tr>
<tr><td>CURR PROGRAM</td><td>none</td><td>special type - displays currently running program</td></tr>
<tr><td>CURR ROUTINE</td><td>none</td><td>special type - displays currently running routine</td></tr>
<tr><td>TIMER (SEC)</td><td>Prog. Name, Var. Name</td><td>Displays KAREL timer in seconds</td></tr>
<tr><td>TIMER (MIN)</td><td>Prog. Name, Var. Name</td><td>Displays KAREL timer in minutes</td></tr>
<tr><td>TIMER (HR)</td><td>Prog. Name, Var. Name</td><td>Displays KAREL timer in hours</td></tr>
<tr><td>PROG STATUS</td><td>none</td><td>special type - displays status of current program: RUNNING, MOVING, PAUSED, WAITING, WAITING FOR DI[ ], WAITING FOR RI[ ], ABORTED</td></tr>
<tr><td colspan="3">For DI[ ], DO[ ], RI[ ], and RO, the variable data types are as follows:
• INTEGER - displays as 1, 0
• BOOLEAN - displays as TRUE, FALSE
• TEXT - displays as ON, OFF</td></tr>
<tr><td>REG + I/O</td><td colspan="3">This item allows you to specify the number of the register or the I/O signal, when the variable type is R[ ], DI[ ], DO[ ], RI[ ], RO[ ], GI[ ], or GO[ ].</td></tr>
<tr><td>KAREL Program</td><td colspan="3">This item allows you to specify the name of the KAREL program that contains the variable you are displaying, when the variable type is KAREL VAR.</td></tr>
<tr><td>Variable Name</td><td colspan="3">This item allows you to specify the name of the variable you are displaying.</td></tr>
<tr><td>Data Type</td><td colspan="3">This item allows you to specify the variable data types:
• NONE
• INTEGER
• REAL
• TEXT
• BOOLEAN</td></tr>
</table>

Examples

See Status Menu (menu) Macro Example 1 and Status Menu (menu) Macro Example 2 for examples of how to use the Status Menu (menu) macro instruction in a teach pendant program.

Status Menu (menu) Macro Example 1

PROG TEST1
1: Status Menu(2)

Status Menu (menu) Macro Example 2

PROG TEST2
1: Status Menu('ProductionStatus')

Setup Procedure

Use Procedure 36-4 to create a status menu.

Table 36.4 (b) Status Menu Screen Items

ITEM	DESCRIPTION
Item	This item refers to item numbers (1-99) always in sequence.
Menu number	This item refers to unique menu numbers (1-99).
Menu name	This item is the name of the menu.

Table 36.4 (c) Status Menu DETAIL Screen Items

ITEM	DESCRIPTION
Status menu no:	This item refers to unique menu numbers (1-99).
Menu name	This item is the name of the menu.
Line	This item is the line number for the status items. Maximum of nine items
Text	This item is the display text used in the prompt box.

Table 36.4 (d) Status Menu Sub-DETAIL Screen Items

ITEM	DESCRIPTION		
Display Text	This item allows you to specify the text that will be displayed for the status item. You can [TYPE] up to 17 characters of display text.		
Variable [TYPE]	This item allows you to specify the kind of variable for the status item:		
	[TYPE] of Variable	What to Define	Data Display [TYPE]
	NONE	none	none
	R[ ]	REG Number	INTEGER, REAL, BOOLEAN
	DIN[ ]	I/O Number	INTEGER, BOOLEAN, TEXT
	DOUT[ ]	I/O Number	INTEGER, BOOLEAN, TEXT
	RDI[ ]	I/O Number	INTEGER, BOOLEAN, TEXT
	RDO[ ]	I/O Number	INTEGER, BOOLEAN, TEXT
	GIN[ ]	I/O Number	INTEGER
	GOUT[ ]	I/O Number	INTEGER
	KAREL VAR	PROG. Name, Var. Name	INTEGER, REAL, BOOLEAN, TEXT
	TIME	none	special [TYPE] - displays current time
	CURR PROGRAM	none	special [TYPE] - displays currently running program
	CURR ROUTINE	none	special [TYPE] - displays currently running routine
	TIMER (SEC)	Prog. Name, Var. Name	Displays KAREL timer in seconds
	TIMER (MIN)	Prog. Name, Var. Name	Displays KAREL timer in minutes
	TIMER (HR)	Prog. Name, Var. Name	Displays KAREL timer in hours

ITEM	DESCRIPTION		
Variable [TYPE]	PROG STATUS	none	special [TYPE] - displays status of current program: RUNNING, MOVING, PAUSED, WAITING, WAITINGFOR DI[], WAITING FORRI[],AT HOME, AT REPAIR, ABORTED
	For DI[ ], DO[ ], RI[ ], and RO, the variable data types are as follows: • INTEGER - displays as 1, 0 • BOOLEAN - displays as TRUE, FALSE • TEXT - displays as ON, OFF		
REG + I/O	This item allows you to specify the number of the register or the I/O signal, when the variable [TYPE] is R[], DI[], DO[], RI[], RO[], GI[], or GO[].		
KAREL Program	This item allows you to specify the name of the KAREL program that contains the variable you are displaying, when the variable [TYPE] is KAREL VAR.		
Variable Name	This item allows you to specify the name of the variable you are displaying.		
Data Type	This item allows you to specify the variable data [TYPE]s: • NONE • INTEGER • REAL • TEXT • BOOLEAN		

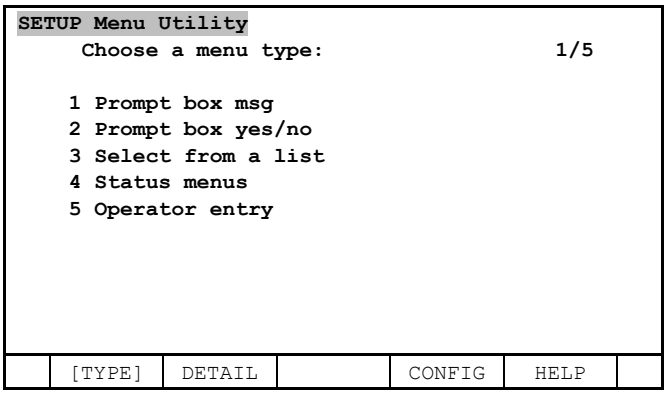
Procedure 36-4 Creating a Status Menu

Conditions

• The default device is set. Refer to Procedure 7-1 Changing file I/O devices in Section 7.1 FILE INPUT/OUTPUT UNITS in the *R-50iA CONTROLLER OPERATOR'S MANUAL (Basic Operation)* (B-84464EN).

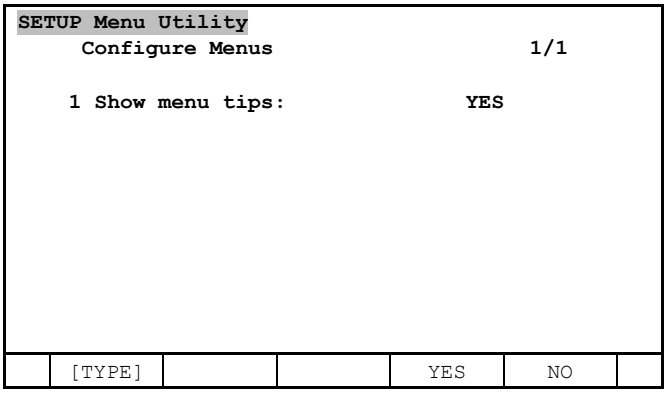
Steps

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



5 To display menu tips:
Menu tips are prompt boxes that are displayed the first time you press the menu DETAIL key for each menu type. The prompt box contains information that describes the menus, macros, and parameters.

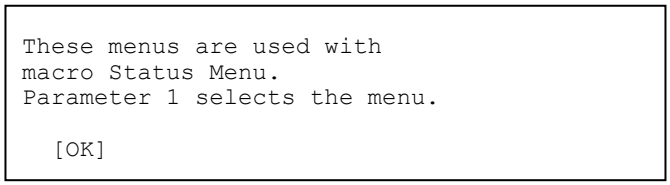
a. Press F4, CONFIG. You will see a screen similar to the following.



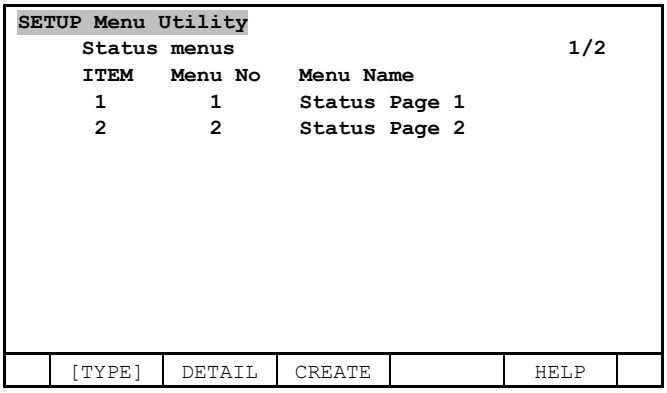
b. To turn off menu tips, press F5, NO.

To turn on menu tips, press F4, YES. You will see a screen similar to the following.

When you are finished viewing menu tips, press ENTER to select OK.



6 Move the cursor to Status menus and press F2, DETAIL. You will see a screen similar to the following.



7 To create a new menu:

a. Press F3, CREATE.

b. Type the number of the new menu and press ENTER. You will see a screen similar to the following.

```
SETUP Menu Utility
     STATUS menu no:      1                1/10
     Menu name/Title: ProductionStatus
       Line    Text
     1 Display : Current Program
     2 Display : Current Routine
     3 Display : Status
     4 Display : Current time
     5 Display : Execution time
     6 Display :
     7 Display :
     8 Display :
     9 Display :

   | [TYPE] | DETAIL | TEST |  | HELP |  |
```

c. Move the cursor to Menu name and press ENTER. Use the appropriate function keys to name the menu and press ENTER.

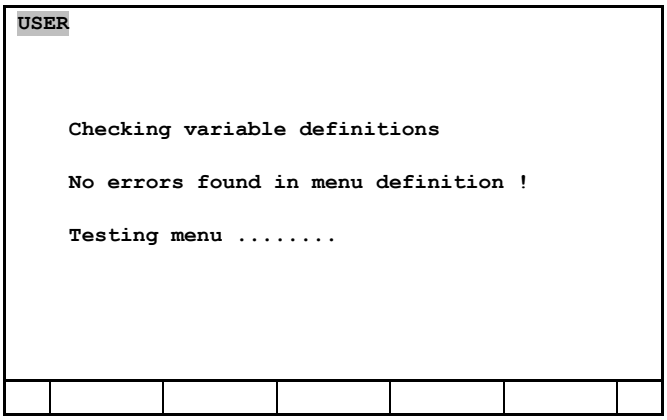
d. Move the cursor to each item and press F2, DETAIL. You will see a screen similar to the following.

```
SETUP Menu Utility
     STATUS menu no:     1                 1/6
     Menu name:                      Line no.:2
     1  Text:     Current Routine
     2  Var Type: CURR Routine
     3  REG + IO: 0
     4  KARELProg:
     5  Var Name:
     6  DataType: TEXT

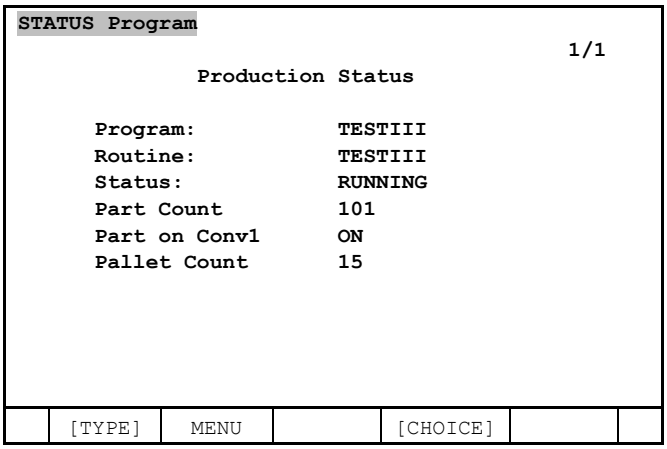
   | [TYPE] | NEXT V. | PREV V. |  | HELP |  |
```

e. Select each item and set as desired. To display the next variable, press F2, NEXT V. To display the previous variable, press F3, PREV V. Refer to Table 36.4 (a) for information on how to set each item. When you are finished setting variable detail information, press PREV.

f. To test the menu you created, press F3, TEST.
   The status menu's display variables are checked to make sure that the proper information was specified and that the variables exist.
   If no errors were found, you will see the following screen briefly.



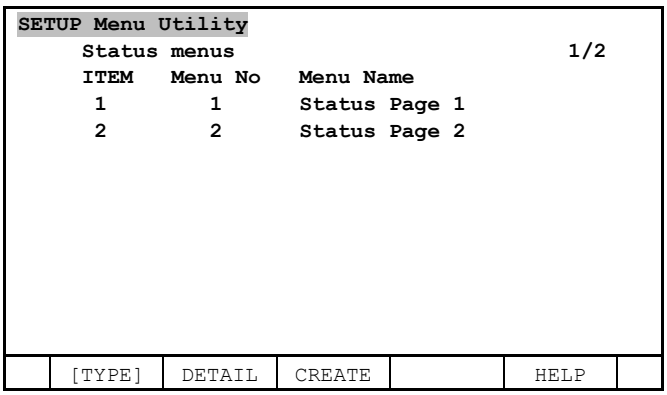
When testing is finished, you will see a screen similar to the following.



If errors were found, they will be listed on the teach pendant screen and the menu will not be displayed. The previous menu will be displayed. Correct the variable definitions and press F3, TEST, again.

8 To modify an existing menu:

a. Move the cursor to the menu name and press F2, DETAIL. You will see a screen similar to the following.



b. Move the cursor to the lines you want to change, press F2, DETAIL, and make the changes you want.

9 To save (back up) a menu to the default device:

a. Set the default device. Refer to Procedure 7-1 Changing file I/O devices in Section 7.1 FILE INPUT/OUTPUT UNITS in the *R-50iA CONTROLLER OPERATOR'S MANUAL (Basic Operation)* (B-84464EN).

b. Move the cursor to the name of the menu you want to save.

c. Press NEXT, >, and press F1, SAVE. The menu will be saved to the default device.

NOTE
- If the file already exists on the default device, you will be prompted whether to replace the existing file with the one you are backing up. To replace the file, select YES. To cancel, select NO.
- You can also back up a menu using FILE, BACKUP APPLICATION. Refer to Section 7.4 SAVING FILES in the *R-50iA CONTROLLER OPERATOR'S MANUAL (Basic Operation)* (B-84464EN) for more information.
- Each status menu is stored in its own variable file that begins with the prefix "**MENU4**." The next two characters of the filename are the menu number. For example, the status menu 1 is stored in the variable file named "MENU4**01**.VR."

10 To load (restore) a menu:

a. Move the cursor to the name of the menu you want to load.

b. Press NEXT, >, and press F2, LOAD. The menu will be loaded from the default device.

NOTE
- If the file already exists on the controller, you will be prompted whether to replace the existing file with the one you are loading. To replace the file, select YES. To cancel, select NO.
- You can also load a menu using FILE, LOAD, *.VR. Refer to Section 7.5 LOADING FILES in the *R-50iA CONTROLLER OPERATOR'S MANUAL (Basic Operation)* (B-84464EN) for more information.

11 To clear a menu:

a. Move the cursor to the name of the menu you want to clear.

b. Press NEXT, >, and press F3, CLEAR.

c. Confirm the clear:

- To clear the menu, select YES and press ENTER.
- To cancel the clear, select NO and press ENTER.

12 To copy a menu:

a. Move the cursor to the name of the menu you want to copy.

b. Press NEXT, >, and press F4, COPY.

c. Type the menu number to which you want to copy the menu and press ENTER.

NOTE
If the new menu already exists, you will be prompted whether to replace the existing menu with the one you are copying. To replace the file, select YES. To cancel, select NO.

36.5 OPERATOR ENTRY MENU

An operator entry menu requires the operator to enter INTEGER, REAL, BOOLEAN, or text values. When an operator entry menu is executed, the program pauses until the operator types the information and then presses F3, DONE.

See Figure 36.5 (a) for an example of an operator entry menu.

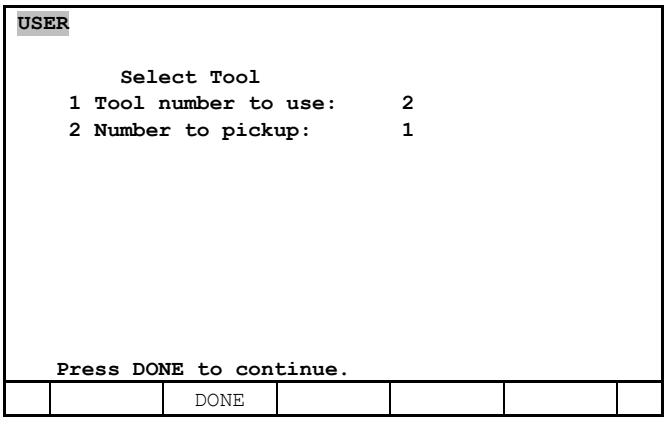


Fig.36.5 (a) Operator Entry Menu

The operator uses the:

- Numeric keys to type numbers
- Function keys to select YES or NO
- ENTER key and function keys to type letters

Macro: Op. Entry Menu (menu)
An operator entry menu uses the Op. Entry Menu (menu) macro.

Parameter 1: menu - This is the menu number or menu name. The menu can be chosen by number or name.

Display
To display an operator entry menu you create, use the Op. Entry Menu (menu) macro. The menu will be displayed automatically on the USER screen. If the operator changes the teach pendant display to another screen during teach pendant execution of this menu but before pressing DONE, the operator can press MENU and select USER to display this screen again.

Errors
The following errors are possible:

- If the Op. Entry Menu (menu) macro was used in a teach pendant program without a parameter, an alarm is given and the program is aborted.

- If the selected operator entry menu does not exist or the menu is the wrong kind of menu when it is called from a teach pendant program, “MHND-167 Selected menu does not exist” alarm appears.

Menu File Name

Each operator entry menu is stored in its own variable file that begins with the prefix "MENU5." The next two characters of the filename are the menu number. For example, the operator entry menu 1 is stored in the variable file named "MENU501.VR."

Setup

In a status menu, you can specify

- Up to 10 prompt items
- The names of the variables that correspond to the prompt items, and detailed variable information

Table 36.5 (a) lists and describes the variable detail information.

Table 36.5 (a) Operator Entry Menu Variable Detail Information

ITEM	DESCRIPTION		
Prompt	This item allows you to specify the text that will be displayed for the prompt item. You can type up to 17 characters of display text.		
Variable Type	This item allows you to specify the kind of variable for the value of the prompt item:		
	Type of Variable	What to Define	Data Display Type
	NONE	none	none
	R[ ]	REG Number	INTEGER, REAL, BOOLEAN
	KAREL VAR	Prog. Name, Var. Name	INTEGER, REAL, BOOLEAN, TEXT
REG Number	This item allows you to specify the number of the register, when the variable type is R[ ].		
KAREL Program	This item allows you to specify the name of the KAREL program that contains the variable you are using, when the variable type is KAREL VAR.		
Variable Name	This item allows you to specify the name of the KAREL variable you are using, when the variable type is KAREL VAR.		
Data Type	This item allows you to specify the variable data types: • NONE • INTEGER • REAL • TEXT • BOOLEAN		
Variable Minimum	This item allows you to specify the minimum value allowed for the INTEGER or REAL variable.		
Variable Maximum	This item allows you to specify the maximum value allowed for the INTEGER or REAL variable.		

Examples

See Op Entry Menu (menu) Macro Example 1 and Op. Entry Menu (menu) Macro Example 2 for examples of how to use the Op. Entry Menu (menu) macro instruction in a teach pendant program.

Op Entry Menu (menu) Macro Example 1

PROG TEST1
1: Ops. Entry Menu(2)

Op. Entry Menu (menu) Macro Example 2

PROG TEST2
1: Ops. Entry Menu('SetupTool')

Setup Procedure

Table 36.5 (b) through Table 36.5 (d) list the Operator Entry Menu Screen setup items.
Use Procedure 36-5 to create an operator entry menu.

Table 36.5 (b) Operator Entry Menu Screen Items

ITEM	DESCRIPTION
Item	This item refers to the ITEM number that you selected.
Menu number	This item refers to the menu number you selected
Menu name	This item refers to the name of the menu you selected

Table 36.5 (c) Operator Entry Menu DETAIL Screen Items

ITEM	DESCRIPTION
OP Entry Menu no	This item is the number of the menu (unique ID).
Title	This item is the name of the menu.
Prompt	This item has up to ten prompts.

Table 36.5 (d) Operator Entry Menu Sub-DETAIL Screen Items

<table>
<tr><th>ITEM</th><th colspan="3">DESCRIPTION</th></tr>
<tr><td>Prompt</td><td colspan="3">This item allows you to specify the text that will be displayed for the prompt item. You can enter up to 17 characters of display text.</td></tr>
<tr><td rowspan="4">Variable [TYPE]</td><td colspan="3">This item allows you to specify the kind of variable for the status item:</td></tr>
<tr><td>[TYPE] of Variable</td><td>What to Define</td><td>Data Display [TYPE]</td></tr>
<tr><td>None</td><td>None</td><td>None</td></tr>
<tr><td>R[ ]</td><td>REG Number</td><td>INTEGER, REAL, BOOLEAN</td></tr>
<tr><td></td><td>KAREL VAR</td><td>Prog. Name, Var. Name</td><td>INTEGER, REAL, BOOLEAN, TEXT</td></tr>
<tr><td>REG Number</td><td colspan="3">This item allows you to specify the number of the register, when the variable [TYPE] is R[].</td></tr>
<tr><td>KAREL Program</td><td colspan="3">This item allows you to specify the name of the KAREL program that contains the variable you are using, when the variable [TYPE] is KAREL VAR.</td></tr>
<tr><td>Variable Name</td><td colspan="3">This item allows you to specify the name of the KAREL variable you are using, when the variable [TYPE] is KAREL VAR.</td></tr>
<tr><td>Data Type</td><td colspan="3">This item allows you to specify the variable data [TYPE]s:
• NONE
• INTEGER
• REAL
• TEXT
• BOOLEAN</td></tr>
<tr><td>Variable Minimum</td><td colspan="3">This item allows you to specify the minimum value allowed for the INTEGER or REAL variable.</td></tr>
<tr><td>Variable Maximum</td><td colspan="3">This item allows you to specify the maximum value allowed for the INTEGER or REAL variable.</td></tr>
</table>

Procedure 36-5 Creating an Operator Entry Menu

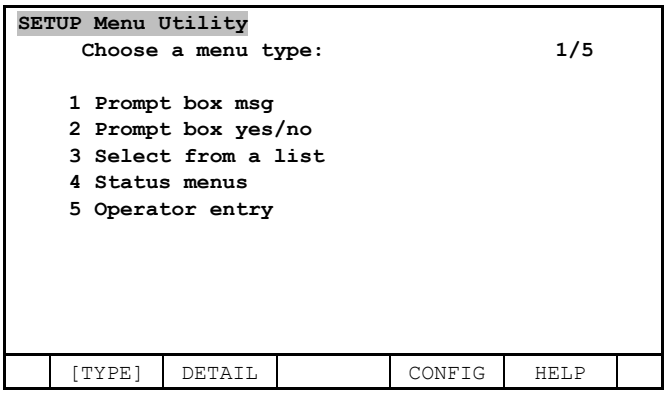
Conditions

- The default device is set. Refer to Procedure 7-1 Changing file I/O devices in Section 7.1 FILE INPUT/OUTPUT UNITS in the *R-50iA CONTROLLER OPERATOR'S MANUAL (Basic Operation)* (B-84464EN).

Steps

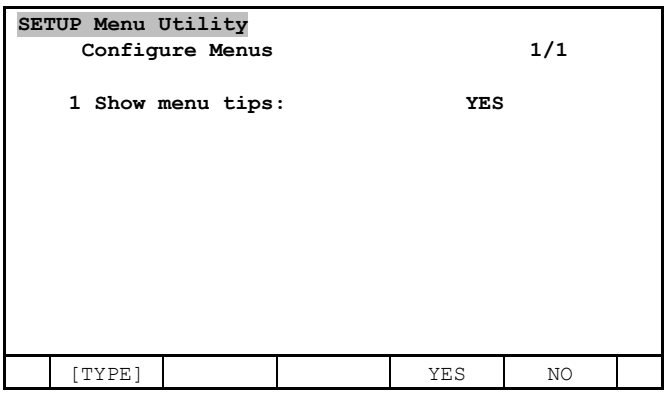
1 Press MENU.
2 Select SETUP.
3 Press F1, [TYPE].

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



5 To display menu tips:
Menu tips are prompt boxes that are displayed the first time you press the menu detail key for each menu type. The prompt box contains information that describes the menus, macros, and parameters.

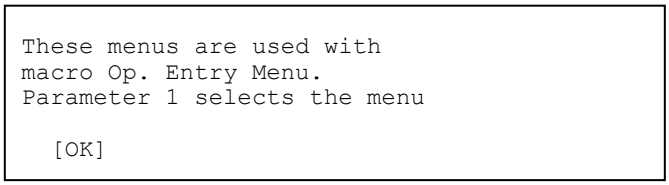
a. Press F4, CONFIG. You will see a screen similar to the following.



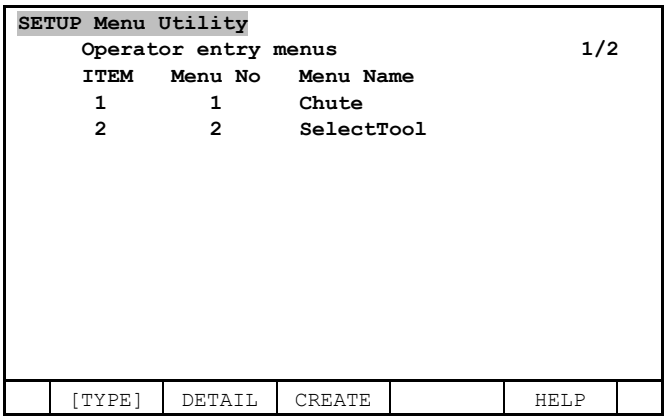
b. To turn off menu tips, press F5, NO.

To turn on menu tips, press F4, YES. You will see a screen similar to the following.

When you are finished viewing menu tips , press ENTER to select OK.



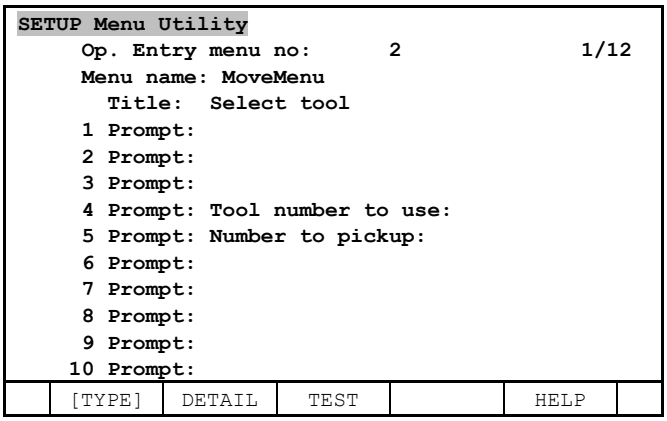
6 Move the cursor to Operator entry and press F2, DETAIL. You will see a screen similar to the following.



7 To create a new menu:

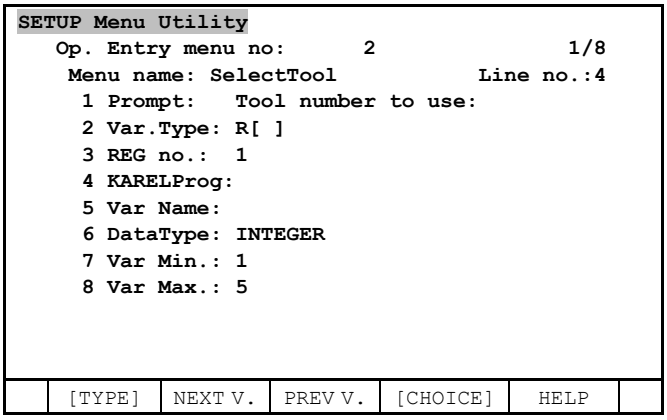
a. Press F3, CREATE.

b. Type the number of the new menu and press ENTER. You will see a screen similar to the following.



c. Move the cursor to Menu name and press ENTER. Use the appropriate function keys to name the menu and press ENTER.

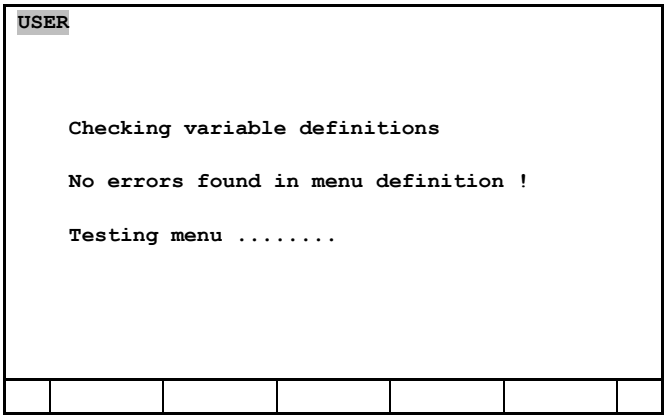
d. Move the cursor to each item and press F2, DETAIL. You will see a screen similar to the following.



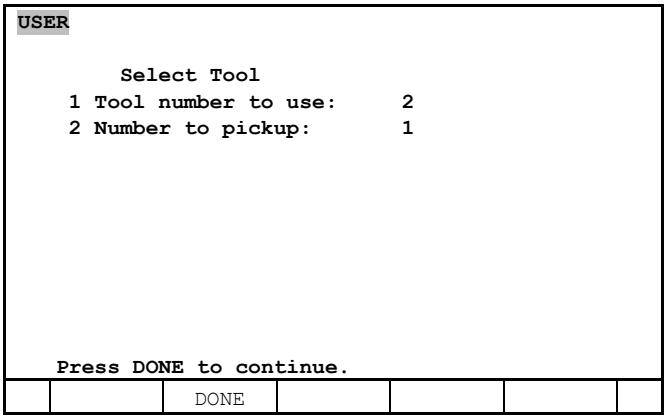
e. Select each item and set as desired. To display the next variable, press F2, NEXT V. To display

the previous variable, press F3, PREV V. Refer to Table 36.5 (a) for information on how to set each item. When you are finished setting variable DETAIL information, press PREV.

f. To test the menu you created, press F3, TEST.
The operator entry menu's display variables are checked to make sure that the proper information was specified and that the variables exist.
If no errors were found, you will see the following screen briefly.



When testing is finished, you will see a screen similar to the following.

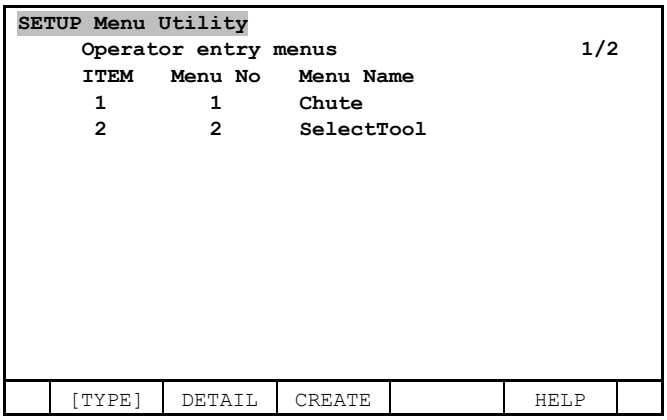


If errors were found, they will be listed on the teach pendant screen and the menu will not be displayed. Correct the variable definitions and press F3, TEST, again.

NOTE
If the variable information is not corrected before the menu macro is executed from a teach pendant program, the incorrect variables will be set to undefined and the values will not be able to be changed.

8 To modify an existing menu:

a. Move the cursor to the menu name and press F2, DETAIL. You will see a screen similar to the following.



b. Move the cursor to the lines you want to change, press F2, DETAIL, and make the changes you want.

9 To save (back up) a menu to the default device:

a. Set the default device. Refer to Procedure 7-1 Changing file I/O devices in Section 7.1 FILE INPUT/OUTPUT UNITS in the *R-50iA CONTROLLER OPERATOR'S MANUAL (Basic Operation)* (B-84464EN).

b. Move the cursor to the name of the menu you want to save.

c. Press NEXT, >, and press F1, SAVE. The menu will be saved to the default device.

NOTE
- If the file already exists on the default device, you will be prompted whether to replace the existing file with the one you are backing up. To replace the file, select YES. To cancel, select NO.
- You can also back up a menu using FILE, BACKUP APPLICATION. Refer to Section 7.4 SAVING FILES in the *R-50iA CONTROLLER OPERATOR'S MANUAL (Basic Operation)* (B-84464EN) for more information.
- Each operator entry menu is stored in its own variable file that begins with the prefix "**MENU5**." The next two characters of the filename are the menu number. For example, the operator entry menu 1 is stored in the variable file named "MENU5**01**.VR."

10 To load (restore) a menu:

a. Move the cursor to the name of the menu you want to load.

b. Press NEXT, >, and press F2, LOAD. The menu will be loaded from the default device.

NOTE
- If the file already exists on the controller, you will be prompted whether to replace the existing file with the one you are loading. To replace the file, select YES. To cancel, select NO.
- You can also load a menu using FILE, LOAD, *.VR. Refer to Section 7.5 LOADING FILES in the *R-50iA CONTROLLER OPERATOR'S MANUAL (Basic Operation)* (B-84464EN) for more information.

11 To clear a menu:
   a. Move the cursor to the name of the menu you want to clear.
   b. Press NEXT, >, and press F3, CLEAR.
   c. Confirm the clear:
      • To clear the menu, select YES and press ENTER.
      • To cancel the clear, select NO and press ENTER.

12 To copy a menu:
   a. Move the cursor to the name of the menu you want to copy.
   b. Press NEXT, >, and press F4, COPY.
   c. Type the menu number to which you want to copy the menu and press ENTER.

NOTE
If the new menu already exists, you will be prompted whether to replace the existing menu with the one you are copying. To replace the file, select YES. To cancel, select NO.

36.6 USING YOUR TEACH PENDANT MENUS

After you have set up a menu, you can use the menu macro instruction to display it.

When you edit a teach pendant program to use a teach pendant menu, you select the name of the menu from a subwindow. In the subwindow, you will see a list of all menus that were created using the Menu Utility SETUP screens. This list is updated at Cold start and when menus are added and deleted on the SETUP screen. Because there are five kinds of menus and each kind of menu can be used only with its own specific macro, you must choose the appropriate menu from the list.

A maximum of 20 menu names will be listed. If the menu you need is not listed, you can type it's name or number manually with the teach pendant number and letter keys.

Use Procedure 36-6 to use your teach pendant menus.

Procedure 36-6 Using Your Teach Pendant Menus

Conditions
- You have created a teach pendant program.

Steps
1 Select the program you want to edit from the Select menu.
2 Press ENTER.
3 Insert the menu macro instruction you want.
4 Press the right arrow key to move the cursor to the end of the instruction as shown in the following screen.

```
...
5:  Prompt Box Msg■
...
```

5 Press F4, [CHOICE].
6 Select String and then select Menus.
7 Select the menu you want.

- To select a menu from a list of already defined menus, move the cursor to the string you want and press ENTER.
- If the menu you want is not listed, press F5, String, press the appropriate function keys to type the string, and press ENTER.

37 4D GRAPHICS FUNCTION

The 4D GRAPHICS function allows you to display robots, tools, workpieces, and other work cells as 3D models on the *i*Pendant.
In addition, various invisible internal data, such as taught program positions, can be displayed on the *i*Pendant as 4D information.
A 4D graphic is the fusion of a 3D robot model and the robot's internal data (1D) as the fourth dimension. When the robot moves, the robot model on the 4D Graphics screen also moves. In machine lock, you can move only the robot model, allowing you to preview the robot's motion direction.

The 4D GRAPHICS option (R764) is required to use the 4D GRAPHICS function.

> ⚠ **WARNING**
> When moving the robot with the teach pendant enabled, be sure to watch the robot instead of watching the TP screen. Confirm that the robot is in a safe state before you view the pendant graphics.

> ⚠ **WARNING**
> The 4D GRAPHICS might not be an accurate representation of the real world. Confirm the robot path by actually executing the program.

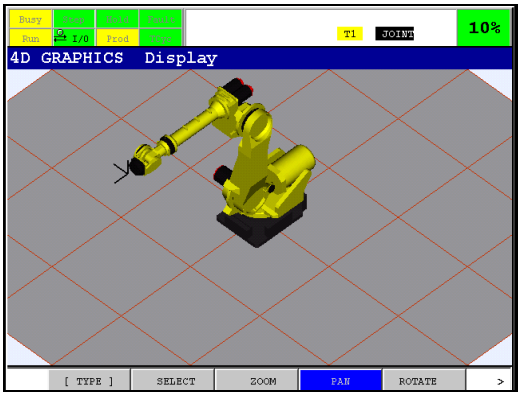
You can display the following invisible data on the *i*Pendant with the 4D Graphics function:
1 Logical tool center point (TCP). - 4D Graphics Display function
2 Frame settings - 4D Visual Jog function
3 Robot jog settings - 4D Jog Preview function
4 The points (nodes) in a TP program - 4D Node Map function
5 Exact path that the robot followed - 4D TCP Trace function
6 Position registers - 4D Position Register function
7 DCS safety zone (*1) - 4D DCS function.
*1: This function requires DCS Pos./Seed check (J567) option.

Use Procedure 37-1 to display the 4D GRAPHICS screen.

Procedure 37-1 Displaying the 4D GRAPHICS screen

Step

1 Press MENU.
2 Select NEXT and then select 4D GRAPHICS. The following screen will be displayed.
Alternatively, instead of Steps 1 and 2, you can press the [*i*] key and the POSN key at the same time.



37.1 OVERVIEW

37.1.1 Graphic Models

The 4D GRAPHICS screen displays the following graphic models.
- Robot model
- Internal data
- Cell floor
- Peripheral devices

Robot model
This displays a graphic model of the ordered robot.

Internal data
This displays various internal data such as the positions taught in a program or the tool center point.

Cell floor
This displays a reference floor with a 1-meter grid. You set up the position of the cell floor on the SETUP Frames screen.

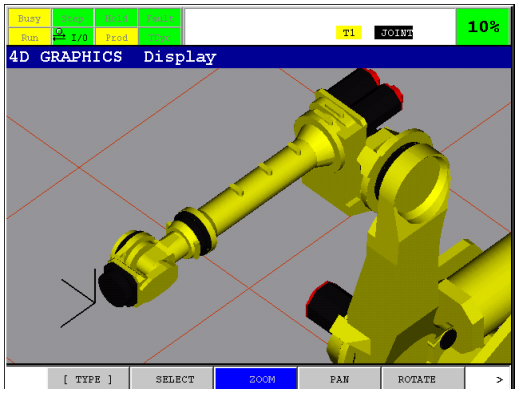
Peripheral devices
Peripheral device graphic models such as a workpieces, fences, and tools can be located anywhere in a 4D GRAPHICS scene using the 4D Editor function in ROBOGUIDE. Refer to Section 37.3 4D Editor Function for details.

37.1.2 Operation

On the 4D GRAPHICS screen, you can adjust the view using the following operation modes:
- ZOOM
- PAN
- ROTATE

You can also display the following pre-defined views:
- Preset Views
- User Views

ZOOM



In ZOOM mode, you can change the magnification of the viewpoint. Zooming in gives you a larger view of the model in question, but a narrower field of view. To use ZOOM mode, press F3, ZOOM. The ZOOM label will turn blue.

You can ZOOM using keys or the touch panel.

Zoom using keys

Zoom in (increase the magnification)

- Press the up arrow key on the *i*Pendant.
- For high magnification, press the SHIFT + up arrow keys on the *i*Pendant.

Zoom out (decrease the magnification)

- Press the down arrow key on the *i*Pendant.
- For high magnification, press the SHIFT + down arrow keys on the *i*Pendant.

Zoom using the touch panel

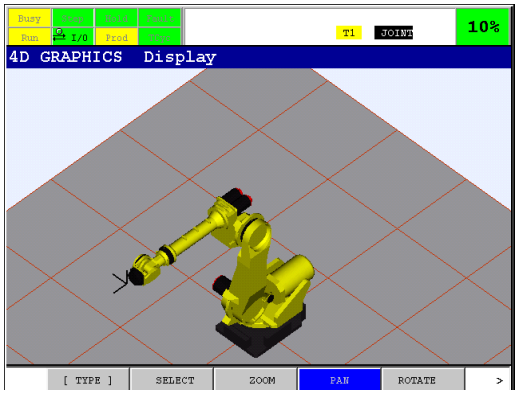
Zoom in (increase the magnification)

- Touch and release near the top of the screen.
- Touch the screen and drag your finger up or right.

Zoom out (decrease the magnification)

- Touch and release near the bottom of the screen.
- Touch the screen and drag your finger down or left.

PAN



In PAN mode, you can move up, down, left, and right without changing the angle of the viewpoint. To use PAN mode, press F4, PAN. The PAN label will turn blue.

You can PAN using keys or the touch panel.

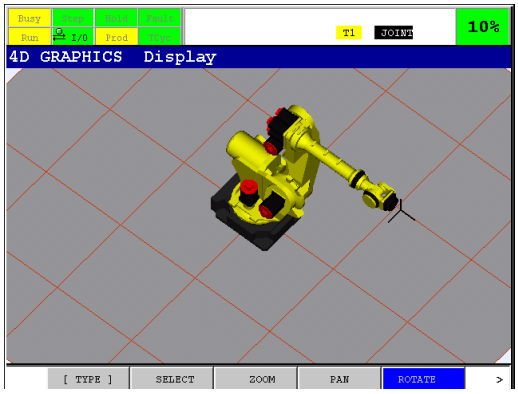
PAN using keys

- Press the arrow key on the *i*Pendant up, down, left, and right.

PAN using the touch panel

- Touch the screen and drag your finger up, down, left, and right.
- Click the screen so the clicked point becomes the center of the screen.

ROTATE



In ROTATE mode you can change the angle without changing the position of the viewpoint. To use ROTATE mode, press F5, ROTATE. The ROTATE label will turn blue.

You can ROTATE using keys or the touch panel.

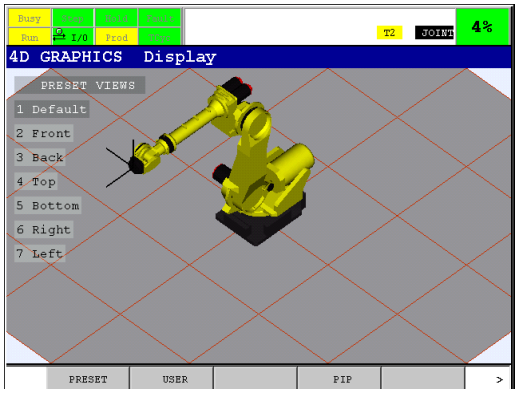
ROTATE using keys

- Press the arrow key on the *i*Pendant up, down, left, and right.
- Press the SHIFT + arrow keys on the *i*Pendant up, down, left, and right for a large amount of rotation.

ROTATE using the touch panel

- Touch the screen and drag your finger up, down, left, and right.

Preset Views



The 4D GRAPHICS screen provides seven preset views. The default view provides a view from 45 degrees, which is a good starting point for setting the view. The default view also puts information back on the screen that has been inadvertently lost. All preset views will center the floor in the middle of the view.

- Default Set the view to the default.
- Front Set the view in front of the robot.
- Back Set the view behind the robot.
- Top Set the view right above the robot.
- Bottom Set the view right below the robot.
- Right Set the view on the right side of the robot.
- Left Set the view on the left side of the robot.

To select Preset Views, press NEXT, > and press F2, [VIEWS]. Select one of the preset views.

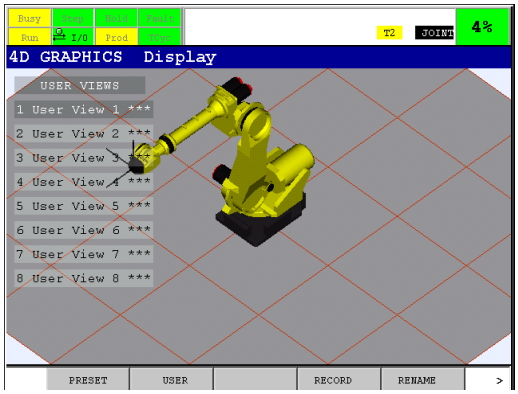
User Views

The 4D GRAPHICS screen provides up to eight user views that you can record. Once a view is recorded, you can retrieve the view by selecting it. You can also name the recorded user views.

Procedure 37-2 Recording User Views

Step

1 Move to the view you want to record.
2 Press NEXT, > and then F2, [VIEWS].
3 Press F2, USER to display the following screen.



4 Select the User View you want to record and press F4, [RECORD].

When a User View is recorded, "***" will be displayed to the right of the User View that was recorded. Once a User View is recorded, it is available to be retrieved at any time.
To rename a User View, press F5, RENAME on one of the User Views.

37.2 4D GRAPHICS SCENE

The 4D GRAPHICS function provides the following scenes:
- 4D GRAPHICS Display
- 4D GRAPHICS Edit Node Map
- 4D GRAPHICS Select Node Map
- 4D GRAPHICS Position Register
- 4D GRAPHICS TCP Trace
- 4D GRAPHICS Frame Display
- 4D GRAPHICS CDS (This function requires the DCS Pos./Seed check (J567) option.)

Pressing F1, [TYPE] in any 4D GRAPHIC scene and selecting a scene will display the selected scene.

37.2.1 4D GRAPHICS Display Scene

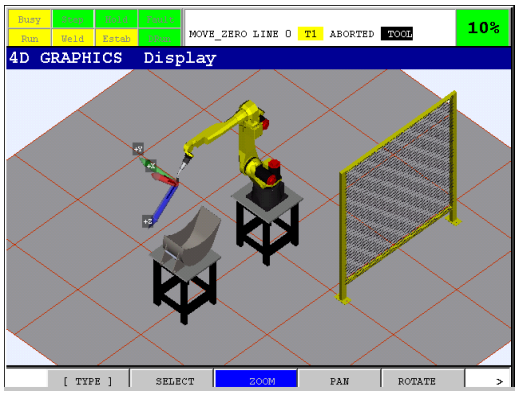
The 4D GRAPHICS Display scene is the default scene and can simultaneously display the data from multiple scenes.

The following functions are available in the 4D GRAPHICS Display scene:
- Visual Jog
- Jog Preview

37.2.1.1 Visual jog

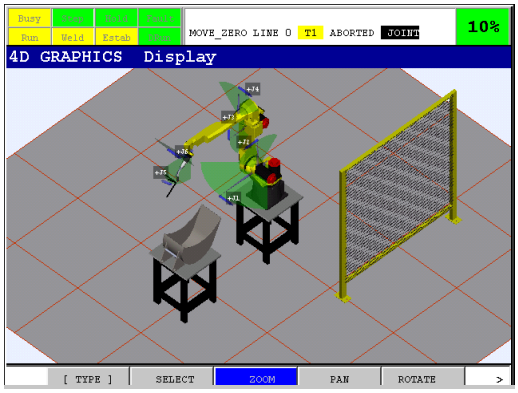
In Visual Jog, the 4D GRAPHICS Display shows the coordinate system and motion group selected for jogging. This allows you to preview the direction that the robot will move before actually jogging the robot. Visual Jog is available when the teach pendant is enabled.

Cartesian Visual Jog



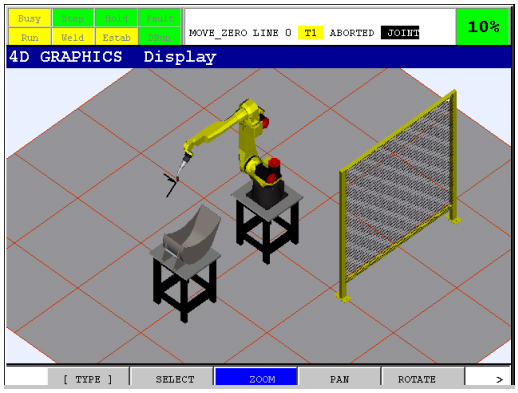
In Cartesian Visual Jog, the red, green, and blue axes correspond to the X, Y, and Z directions.

Joint Visual Jog



Each joint has an indicator consisting of a semi-transparent fan with a needle showing the current position. The fan is usually green and turns red as the robot is jogged closer to the limit.

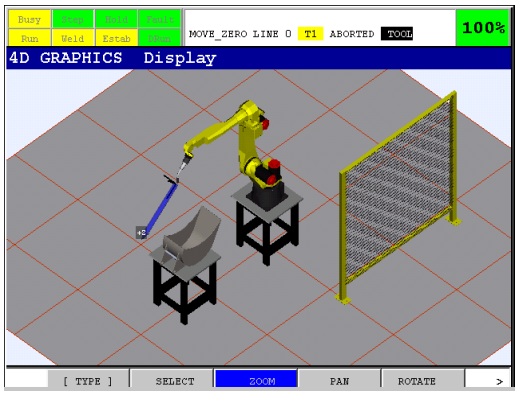
Pressing the *i* key and the COORD key simultaneously turn this function on and off.



37.2.1.2 Jog preview

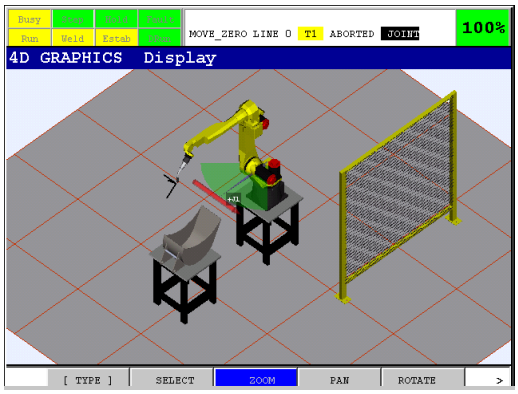
Jog Preview can preview the direction of robot motion when a particular jog key is pressed. It is activated when pressing a single jog key at the same time as the *i* key. Pressing more than one jog key will display the jog direction of the last key pressed. It is not necessary to press the TP Enabling Device (Deadman switch) or clear errors.
Jog Preview is available when the teach pendant is enabled.

Cartesian Jog Preview



Cartesian Jog Preview displays a single arrow indicating the direction the robot will move. The size of the arrow will change depending on the setting of the speed override. The figure shows an example of when the *i* and +Z jog keys are pressed.

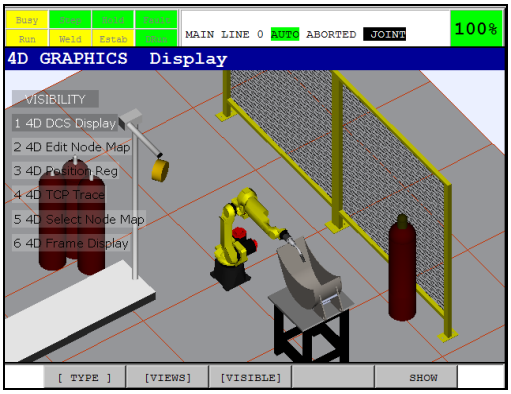
Joint Jog Preview



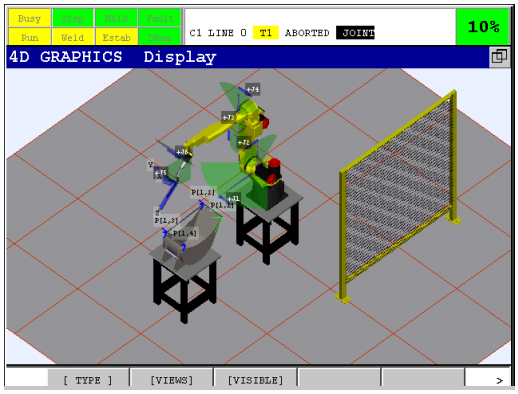
Joint Jog Preview displays a single red arrow tangent to the fan indicating the direction the joint will move. The size of the arrow will change depending on the setting of the speed override. The figure shows an example of when *i* and +J1 jog keys are pressed.

37.2.1.3 Set visibility: 4D GRAPHICS display

The VISIBILE function key controls the elements that are visible in the 4D GRAPHICS Display scene. Pressing F3, VISIBLE on page two of the function keys displays the following screen.



For example, selecting 4D Edit Node Map adds the node map to the scene. Select the same item again to make the element invisible. In the following figure, the node map and frame elements have been added.



NOTE
If too much information is displayed, the performance of the 4D GRAPHICS function will decrease.
When this happens, hide unnecessary data by making it invisible.

37.2.2 4D GRAPHICS Node Map

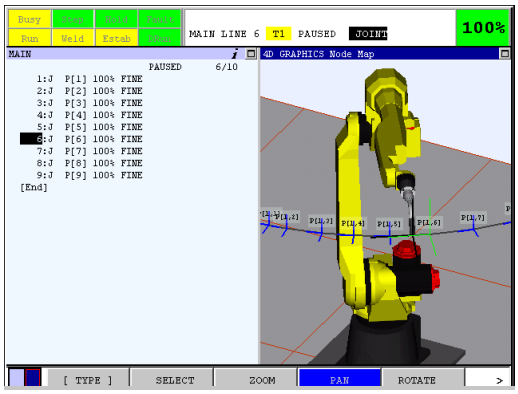
A node map is a 4D visualization of a TP program that displays all of the positions as nodes and connects them with lines. Each node also has a corresponding position number in the text label elements. The main purpose of a node map is to help you see the relationship between the positions in a program without having to move the robot.

NOTE
The Node Map connects two nodes in a program in descending order. Therefore, it is sometimes different from the real motion path.

NOTE
If INC, OFFSET, or Position Register are used in a motion instruction, or if the frame number is changed in a program, the node will be displayed in a different position from the actual position.

Each node is displayed in blue, green, or yellow. Blue is the normal node, green is the node of the position selected on the EDIT screen, and yellow is a node whose position has not yet been determined, such as a position register. Line segments between nodes are displayed in gray or red. A red line segment means there is an unspecified position between the two nodes.
The procedures for editing a node map and selecting a node map differ slightly from each other.

37.2.2.1 4D GRAPHICS edit node map



The 4D GRAPHICS Edit Node Map scene displays the node map for the currently selected program. It is linked to the EDIT screen, and the node that corresponds to the focused position is displayed in green on the node map. When the cursor is moved on the EDIT screen, the green node also moves. The node map will be updated immediately when you add, delete, or modify a position on the edit screen.

Procedure 37-3 4D GRAPHICS Editing a Node Map

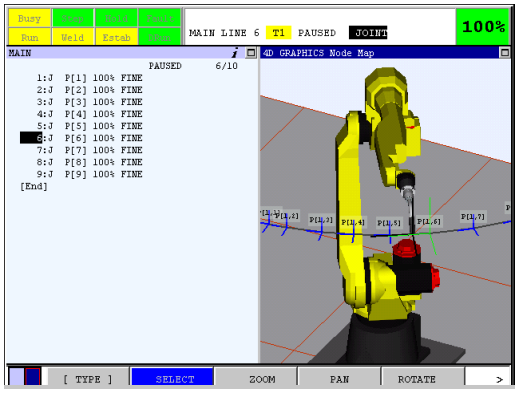
Step

In addition to editing the Node Map from the F1, [TYPE] menu, the following methods are available:

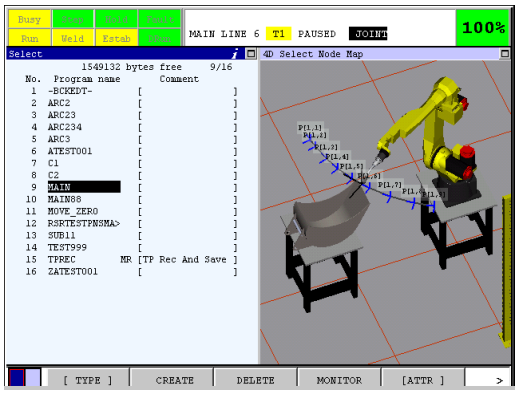
1 Press EDIT. If a program is selected, the EDIT screen will be displayed.
2 Press the *i* and FCTN keys on the EDIT screen. The Related View menu will be displayed.
3 Select 4D Edit Node Map.
4 Instead of Step 1 to Step 3, press the *i* and EDIT keys to display the node map.

SELECT Data

Selecting a node on the 4D graphics edit node map screen causes the cursor on the EDIT screen to move to the corresponding line. This allows you to check the line number of the selected node in the program. To select a node, touch the screen near the node. To use the SELECT Data function, press F2, SELECT. The SELECT label will turn blue.



37.2.2.2 4D GRAPHICS select node map



The 4D GRAPHICS Select Node Map scene displays the node map of the currently selected program on the SELECT screen. It is linked to the SELECT screen. When the cursor is moved on the SELECT screen, the node map of the selected program is displayed. When a program is selected on the SELECT screen, the EDIT screen of the program is displayed, and the corresponding 4D GRAPHICS Edit Node Map will also be displayed automatically.

Procedure 37-4 4D GRAPHICS Select Node Map

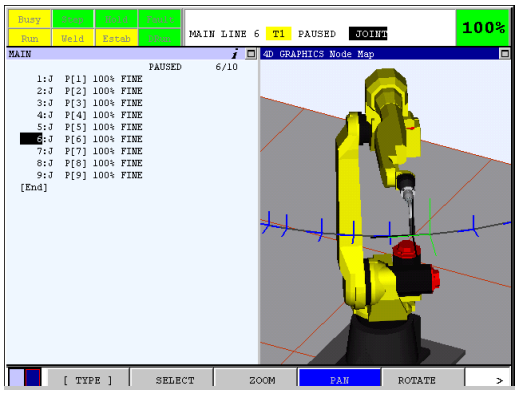
Step

In addition to selecting the Node Map from the F1, [TYPE] menu, the following methods are available:

1 Press SELECT. The SELECT screen will be displayed.
2 Press the *i* and FCTN keys on the SELECT screen. The Related View menu will be displayed.
3 Select 4D Display.
4 Instead of Step 1 to Step 3, press the *i* and SELECT keys to display the node map.

37.2.2.3 Set Visibility: node map position numbers

You can toggle the display of the position number element. by pressing NEXT, > and then F3, [VISIBLE]. The following figure shows a node map without position numbers.



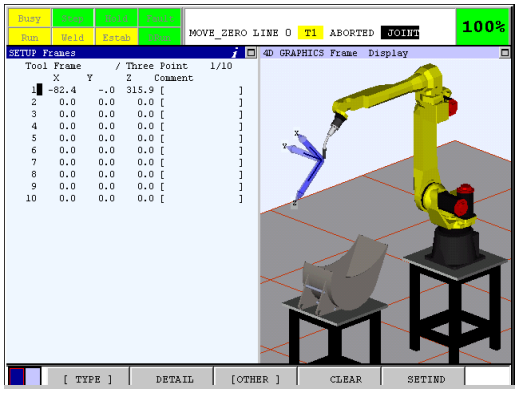
37.2.2.4 Node Map supported data

The following data are supported in a node map:

- Normal positions
- Positions specified by Position Registers
- Via positions and end positions in Circular motion
- Incremented positions by the INC motion modifier

37.2.3 4D GRAPHICS Frame Display

The 4D GRAPHICS Frame Display scene displays the following graphics for the current frame setup operation. The comment for the frame is also displayed.



Procedure 37-5 4D GRAPHICS Frame Display

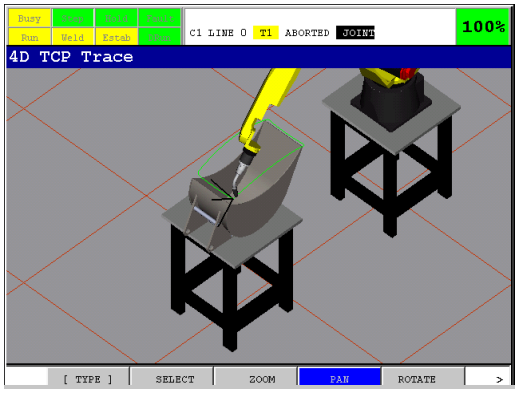
Step

In addition to displaying the 4D GRAPHICS Frame Display from the F1, [TYPE] menu, the following method is available:

1 Press MENU.
2 Select Setup.
3 Press F1 [TYPE].
4 Select Frames.
5 Press the *i* and FCTN keys on the Frame setup screen. The Related View menu will be displayed.
6 Select 4D Frame Display.

37.2.4 4D GRAPHICS TCP Trace

The 4D GRAPHICS TCP Trace scene provides the capability to view the path that the robot took when executing a program. The path is displayed in green and is updated every time the program is executed.

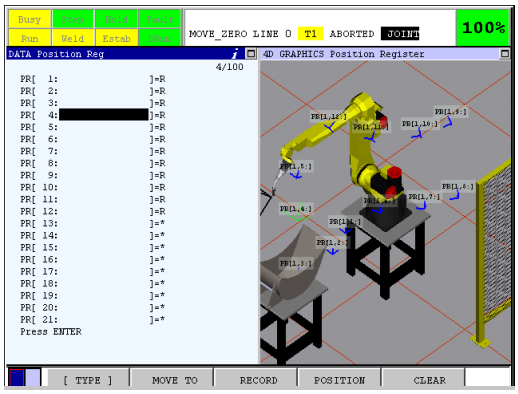


NOTE

The path displayed in the 4D GRAPHICS TCP Trace may differ from the actual path.

37.2.5 4D GRAPHICS Position Register

The 4D GRAPHICS Position Register scene displays position registers as nodes, like in a node map, helping you to understand where the position registers are located spatially. Each node also displays the corresponding position register number.
The 4D GRAPHICS Position Register screen is linked to the Position Register screen and the selected position register is displayed as a green node. When a position register is newly taught or modified, the graphics are automatically updated.



Procedure 37-6 4D GRAPHICS Position Register

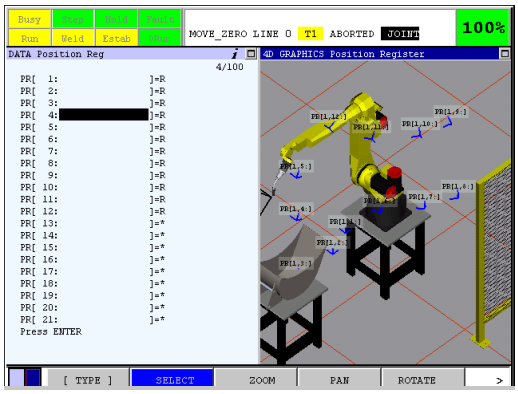
Step

In addition to displaying the 4D GRAPHICS Position Register from the F1, [TYPE] menu, the following methods are available:

1 Press DATA.
2 Press F1, [TYPE].
3 Select Position Reg.
4 Press the *i* and FCTN keys on the Position Register screen. The Related View menu will be displayed.
5 Select 4D Position Reg.
6 Instead of Step 1 to Step 5, press the *i* and DATA keys to display the screen.

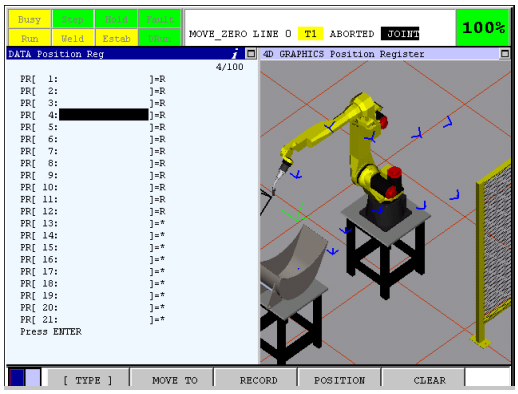
SELECT Data

Selecting a node causes the cursor in the Position Register screen move to the corresponding line. This allows you to check which position register the selected node corresponds to. To select a node, touch the screen near the node. To use the SELECT Data function, press F2, SELECT. The SELECT label will turn blue.



37.2.5.1 Set Visibility: position register numbers

You can toggle the position register number element by pressing NEXT, > and then pressing F3, [VISIBLE]. The following figure shows a node map without position register numbers.

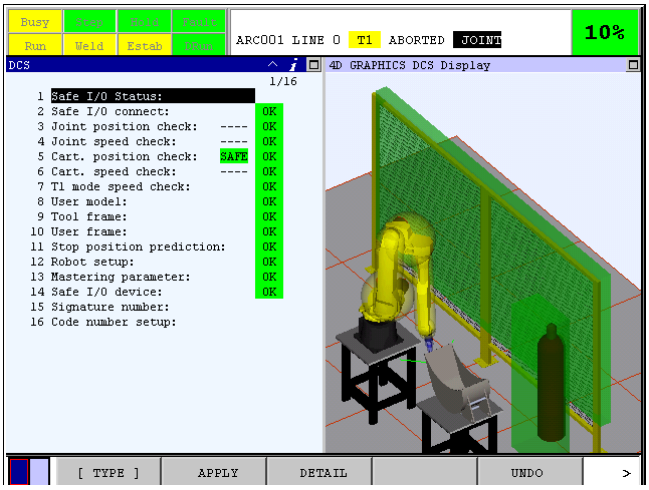


NOTE
Position registers are always considered to be in the “current” frame. The position register graphical display reflects the position of the position register in the current frame. This may not be how the position register is used. Changing the current frame will be reflected in the position of the position register triad in the graphics display.

NOTE
Position registers taught in JOINT will not be displayed.

37.2.6 4D GRAPHICS DCS

The 4D GRAPHICS DCS scene displays DCS safety zones, allowing you to set and check DCS safety zones graphically. This section describes how to display the 4D DCS screen.
The controller must have the DCS Pos./Speed check (J567) option installed to use this function.
Refer to Chapter 8 DCS VISUALIZATION in the *R-50iA CONTROLLER Dual Check Safety Function OPERATOR’S MANUAL* (B-84534EN) for details of the 4D Graphics DCS function.



Procedure 37-7 4D GRAPHICS DCS

Step

In addition to displaying the 4D GRAPHICS DCS from the F1, [TYPE] menu, the following method is available:

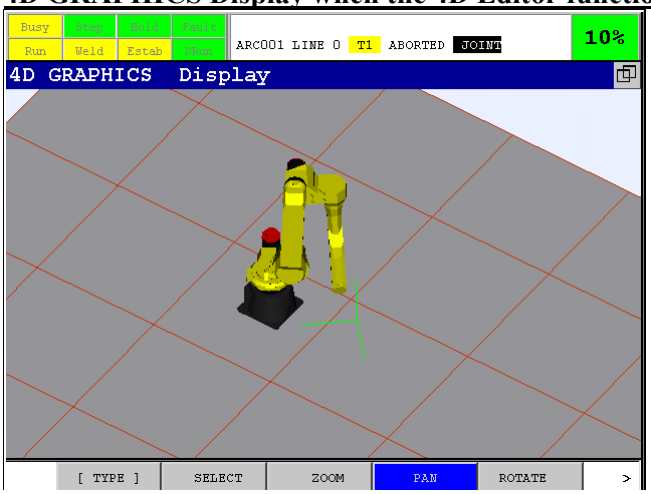
1 Press MENU.
2 Select NEXT, > and select SYSTEM.
3 Press F1, [TYPE] and select [DCS].
4 Press the *i* and FCTN keys on the DCS screen. The Related View menu will be displayed.
5 Select 4D DCS Display.

37.3 4D EDITOR FUNCTION

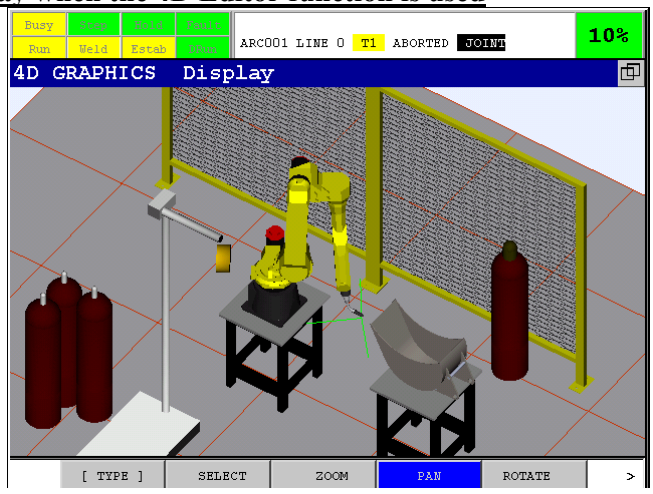
37.3.1 Overview

Peripheral device graphic models such as workpieces, fences and tools can be located anywhere in a 4D GRAPHICS scene using the 4D Editor function in ROBOGUIDE. This section describes the screen using the 4D Editor function.
Refer to the ROBOGUIDE 4D Editor help for details on the 4D Editor function.

4D GRAPHICS Display when the 4D Editor function is not used



4CD GRAPHICS Display when the 4D Editor function is used



The 4D Editor function allows for graphics that more closely resemble the actual system.

37.4 FULL SCREEN 4D DISPLAY

37.4.1 Overview

The Full Screen 4D Display function allows you to display a 4D graphics screen on a PC.
This function requires the Internet Conn/Cust (R558) and R764 4D GRAPHICS (R764) options.

NOTE
For early releases, display of 4D information requires that the PC have an advanced graphics card capable of displaying OpenGL. If your PC does not work, updating your PC graphics drivers might fix it.

37.4.2 Setup

37.4.2.1 Requirements

The Full Screen 4D Display function requires the following:

- The PC has an advanced graphics card capable of displaying OpenGL.
- The PC is connected to a network, and is properly configured to allow a TCP/IP connection to the controller with the *i*Pendant connected.
- The controller is connected to a network and properly configured for network access to the above PC.
- The controller has the Internet Conn/Custo (R558) option installed.
- The controller has the 4D Graphics (R764) option installed. The controller also has the DCS Pos./Speed check (J567) option installed to display the 4D DCS Display function.
- The HTTP Authentication is properly configured on the robot to allow *i*Pendant access. Refer to Section "HTTP AUTHENTICATION" in the "*Network and Security Function OPERATOR'S MANUAL"* (B-84604EN) for configuration information.

37.4.3 Operation

After you have properly configured your browser and verified that you can connect to the robot, you can display the Full Screen 4D Display screen. The following sections describe the operation and limitations of this feature.

37.4.3.1 Accessing the Full Screen 4D Display

This section describes how to connect to the controller and display the Full Screen 4D Display screen.

Procedure 37-8 Accessing the Full Screen 4D Display

Steps

1 Bring up your browser on the PC.
2 In your browser, type the following: http://<myrobot_name_ or IP_address> to access the robot HOME page. When input is completed, the robot homepage screen shown in Fig. 37.4.3.1 (a) is displayed.
Where <myrobot_name_ or_address> is either the DNS name of your robot (such as pderob111.frc.com) or the IP address of your robot (such as 192.168.1.100).
If the message "Full Screen 4D Display : Unavailable" is displayed, check whether the 4D Graphics (R764) option is installed.

If no message displayed but the robot homepage is not displayed, check whether the Internet Conn/Custo (R558) option is installed.

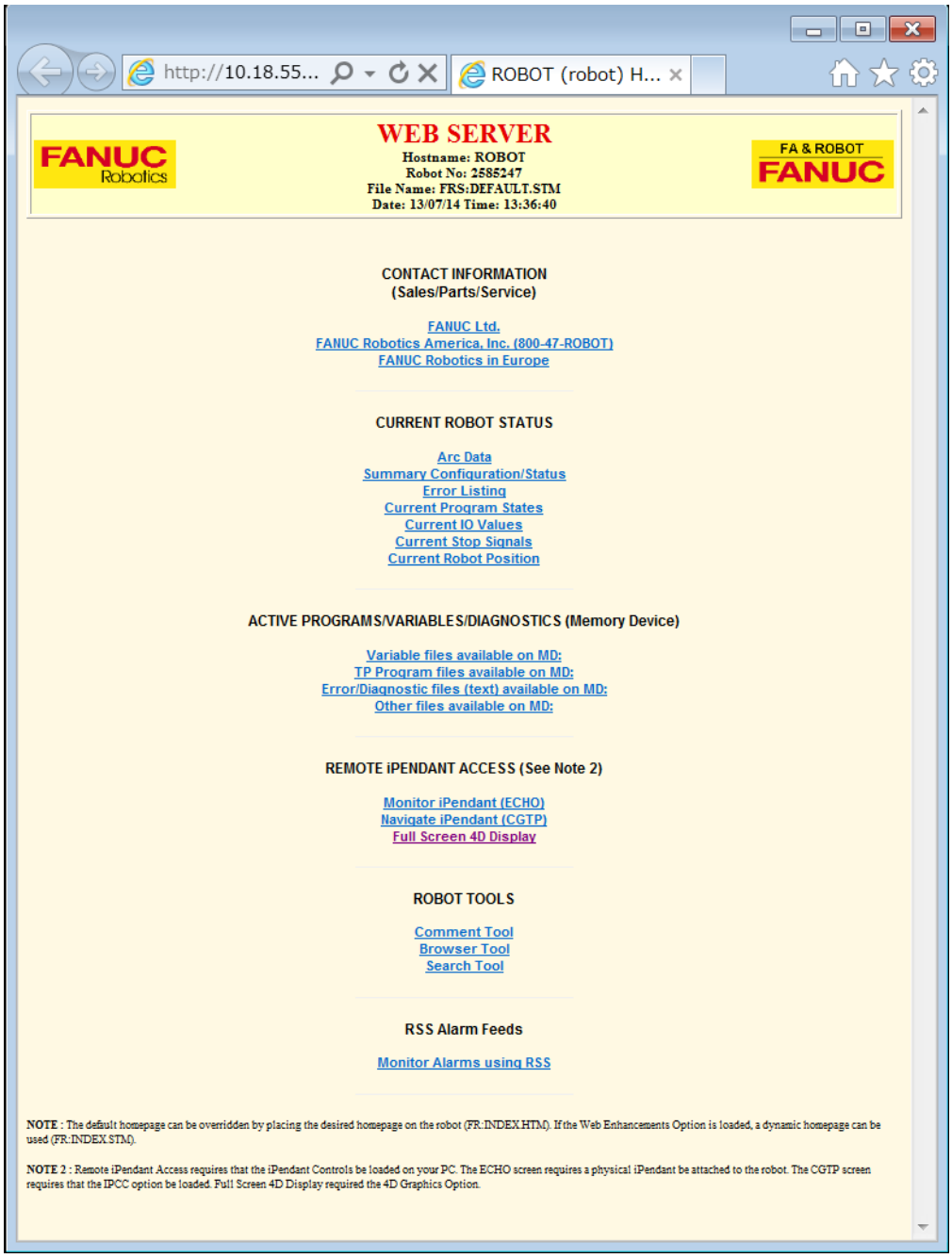


Fig. 37.4.3.1(a) Robot Home Page

3 From the HOME page, when you select Full Screen 4D Display, a 4D Graphics Display screen similar to the one shown in Figure 37.4.3.1(b) will be displayed on your PC.

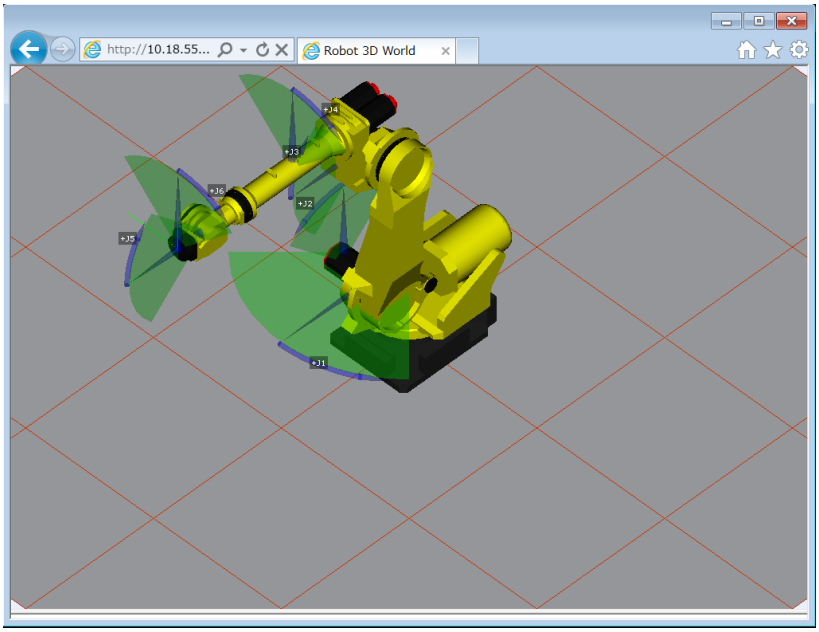


Fig. 37.4.3.1(b) Full Screen 4D Display Screen

37.4.3.2 View adjustment mode

You can adjust the view of a 4D scene on the PC using the mouse to PAN, ROTATE, or ZOOM the view.

Procedure 37-9 Selecting the View Adjustment Mode

Steps

1 Click with the right (alternate) mouse button. The 4D Graphic Display menus will be displayed in the upper left of the screen. Right-click again to dismiss the menu.

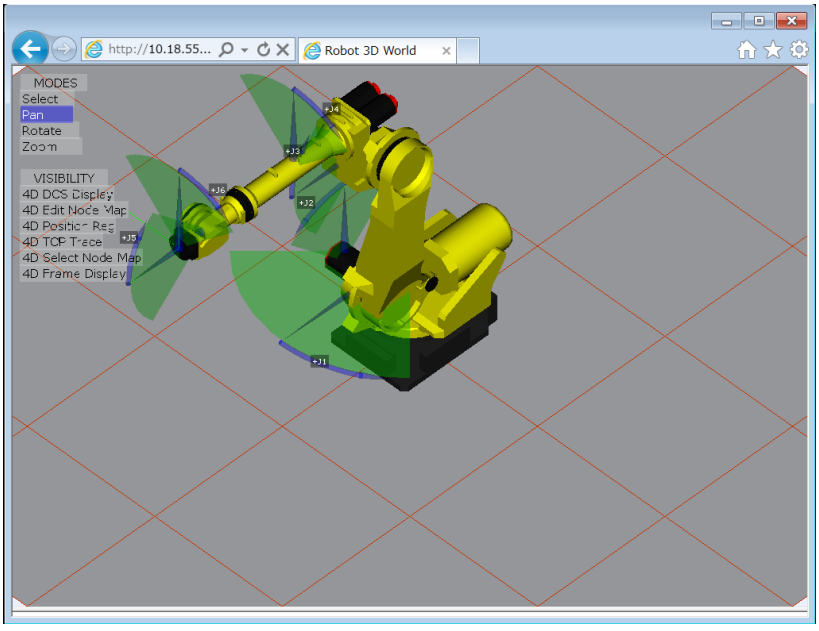


Fig. 37.4.3.2 Full Screen 4D Display Screen

2 Use either mouse button to select the view adjustment mode you want from the following modes:

- Pan - To move the view up and down and side to side.
- Rotate - To rotate the view.
- Zoom - To zoom the view in and out.

Click anywhere outside the menu to dismiss the menu.

3 Use the left (primary) mouse button to adjust the view according to the selection.

37.4.3.3 Scene visibility

Scene visibility allows you to select any combination of 4D display entities. The highlighted (white on blue) elements in the menu are the ones currently being displayed. Selecting an element toggles the visibility on or off.

Procedure 37-10 Selecting Scene Visibility

Steps

1 Click with the right (alternate) mouse button to display the menu.

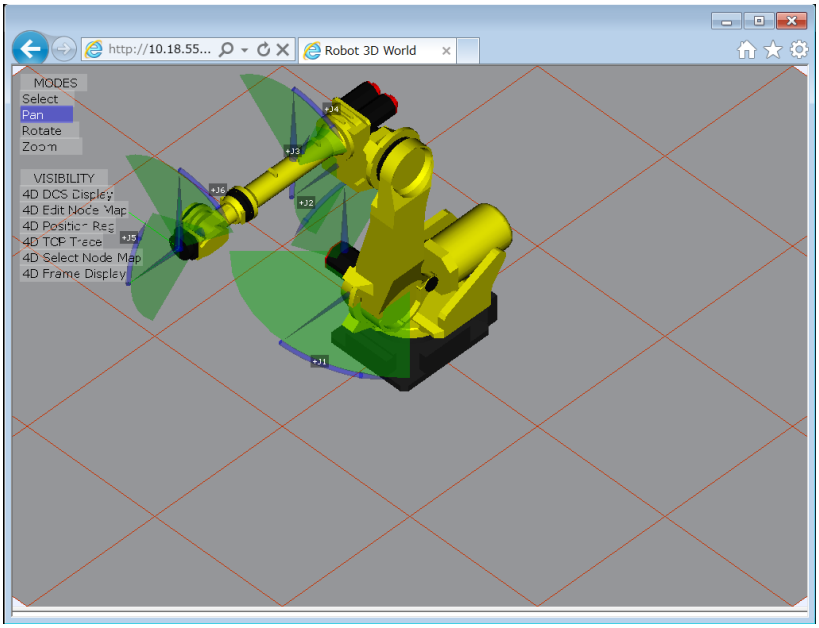


Fig. 37.4.3.3(a) 4D Graphics Node Map Scene

2 Use the left or right mouse button to select the scene you want to toggle:

- To change the visible/invisible status of each data item, select it.
- If you select multiple items, all selected data will be displayed on the 4D screen.

Refer to Section 37.2 4D GRAPHICS SCENE for details of each scene.

For example, in the system shown in Fig. 37.4.3.3(b), when you select 4D Select Node Map, the node map is displayed on your PC.

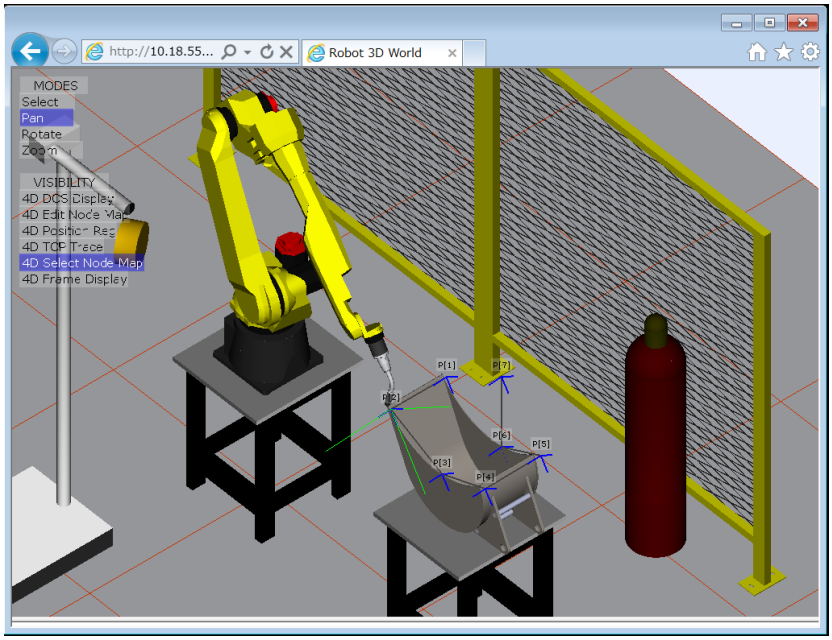


Fig. 37.4.3.3(b) 4D Graphics Node Map Scene

3 Click anywhere outside the menu to dismiss the menu

NOTE
If DCS Pos./Speed check (J567) option is not installed, the 4D DCS Display item will not be displayed.

38 DATA TRANSFER BETWEEN ROBOTS FUNCTION

This section explains the Data Transfer Between Robots optional function (J740).

This function enables you to transfer data between robots over Ethernet.
By calling a KAREL program, you can transfer registers and position registers between robot controllers.
There is no limit to the number of connections.

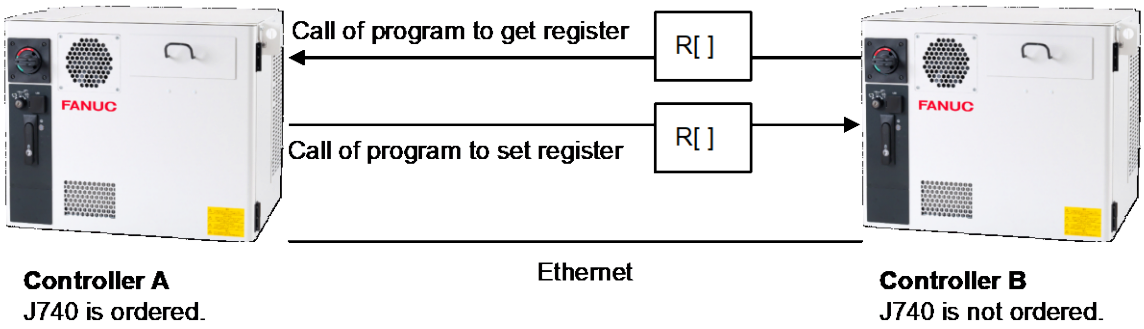


Fig. 38 Data Transfer Between Robots

In Fig. 38, Controller A runs a program to get registers from Controller B. In addition, Controller A calls a program to transfer registers to Controller B.
In Fig. 38, only Controller A starts the data transfer. Controller B is just responding to the request from Controller A. In this case, only Controller A needs this option.

NOTE
R-50iA controller cannot communicate with R-30iB Plus or earlier controller.

This function also provides KAREL built-ins for data transfer. Using a KAREL program, you can transfer the registers and I/O comments of a remote controller. You can also read and write I/O values. KAREL is a programming language used for building robot systems. For details about KAREL, refer to the *KAREL function OPERATOR'S MANUAL* (B-87004EN). Refer to Section 38.7 KAREL built-ins for the built-in routines provided by this function.

38.1 TERMINOLOGY

The following terms are used to simplify the description of the Data Transfer Between Robots function.

Client

Client is the robot controller that starts communication using this function. In Fig. 38, Controller A is the client. The client runs the program to request various services from another controller.

Server

Server is the robot controller that receives the request from the client and serves it. In Fig. 38, Controller B is the server.

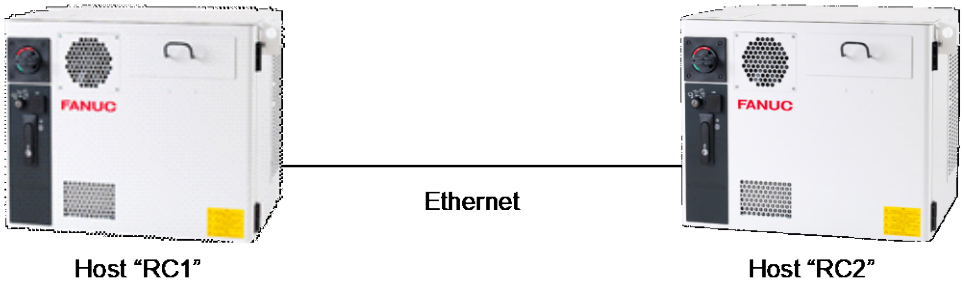
38.2 SETUP

Connect controllers using Ethernet. Set up TCP/IP parameters so that the controllers can communicate over Ethernet using TCP/IP.

1 Press MENU.
2 Select SETUP.
3 Select Host Comm.
4 Display the TCP/IP setup screen.
5 Enter the host names (Robot names) and IP addresses of the controllers that will communicate.
6 Cycle controller power.

For more details, refer to the *Network and Security Function OPERATOR'S MANUAL* (B-84604EN), "Setting up TCP/IP".

NOTE
Do not use underscores ("_") in the host name.




```
SETUP Host Comm
    TCP/IP                                1/40
        Robot name:                        RC1
        PORT#1 IP addr:          192.168.0.2
        Subnet Mask:             255.255.255.0
        Board address:           ********************
        Router IP addr:          ********************

      Host Name (LOCAL)  Internet Address
        1         RC1              190.168.0.2
        2         RC2              190.168.0.3
        3 **********     ********************
        4 **********     ********************

 [TYPE]  |  PORT  PING  HELP  >
```

```
SETUP Host Comm
    TCP/IP                                1/40
        Robot name:                        RC2
        PORT#1 IP addr:          192.168.0.3
        Subnet Mask:             255.255.255.0
        Board address:           ********************
        Router IP addr:          ********************

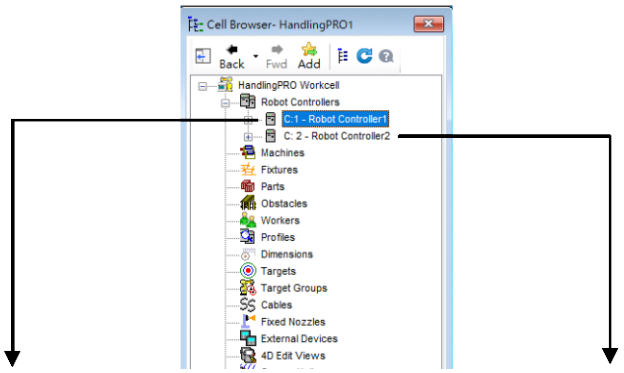
      Host Name (LOCAL)  Internet Address
        1         RC1              190.168.0.2
        2         RC2              190.168.0.3
        3 **********     ********************
        4 **********     ********************

 [TYPE]  |  PORT  PING  HELP  >
```

38.3 TCP/IP SETUP FOR ROBOGUIDE

You can simulate data transfer in a ROBOGUIDE work cell. Note that the IP address of each host must be 127.0.0.x.

The following figure is an example TCP/IP setup for ROBOGUIDE. There are two robot controllers in the work cell. The host name of the first controller is RC1 and the second is RC2.




```
SETUP Host Comm
    TCP/IP                                1/40
        Robot name:                       RC1
        PORT#1 IP addr:            127.0.0.2
        Subnet Mask:           255.255.255.0
        Board address:     ******************
        Router IP addr:    ******************

      Host Name (LOCAL)  Internet Address
        1         RC1              127.0.0.2
        2         RC2              127.0.0.3
        3 **********       ******************
        4 **********       ******************

 [TYPE]  |  PORT  PING  HELP  >
```

```
SETUP Host Comm
    TCP/IP                                1/40
        Robot name:                       RC2
        PORT#1 IP addr:            127.0.0.3
        Subnet Mask:           255.255.255.0
        Board address:     ******************
        Router IP addr:    ******************

      Host Name (LOCAL)  Internet Address
        1         RC1              127.0.0.2
        2         RC2              127.0.0.3
        3 **********       ******************
        4 **********       ******************

 [TYPE]  |  PORT  PING  HELP  >
```

You have to re-start the controllers after TCP/IP setup.

If you run the following program, the value and comment of R[1] of RC1 is transferred to R[5] of RC2.

```
1:  CALL RGETNREG('RC1',1,5,0) ;
```

RGETNREG is a KAREL program provided by this function. For more details, Refer to Section 38.4 STANDARD DATA TRANSFER PROGRAM.

38.4 STANDARD DATA TRANSFER PROGRAMS

This function includes the following programs to transfer registers and position registers.

Program	Function	Syntax
RGETNREG	Gets numeric registers	RGETNREG('host/IP address', source index, destination index, option)
RSETNREG	Sets numeric registers	RSETNREG('host/IP address', destination index, source index, option)
RGETPREG	Gets position registers	RGETPREG('host/IP address', source index, source group number, destination index, destination group, option)
RSETPREG	Sets position registers	RSETPREG('host/IP address', destination index, destination group, source index, source group number, option)

Each program is described in Sections 38.5.1 and later.

NOTE
If the arguments are not correct an error will occur. The following are examples of incorrect arguments:
- Type of argument (integer, string and the like) is wrong.
- The number of arguments is fewer than expected.
- The content of the argument is incorrect.

Use correct arguments in your called program.
When you restart executing the program, move the cursor to one line before the CALL instruction and restart from that line.

38.4.1 Program to Get a Numeric Register

Program name: RGETNREG

Overview

The client controller gets a register from the server and sets it to the client's register.

Syntax

RGETNREG (host_name, src_idx, dest_idx, option)
Input/Output parameters
host_name: [IN] String of host name or IP address of server.
src_idx: [IN] Index of the source register. This integer is the index of the server register.
dest_idx: [IN] Index of the destination register. This integer is the index of the client register.
option: [IN] This integer specifies function of this program.

Value	Description
0	Gets the register value and comment.
1	Gets the register value.
2	Gets the register comment.

Details

If the server does not have R[src_idx], the error "DTBR-014 Bad variable or register index (host name, index)" will be posted.
If the client does not have R[dest_idx], the error "VARS-024 Bad variable or register index" will be posted.

Example

The execution of the following program retrieves the value and comment of R[10] of the host "SERVER".

```
CALL RGETNREG (‘SERVER’, 10, 20, 0)
```

38.4.2 Program to Set a Numeric Register

Program name: RSETNREG

Overview

The client controller sets the client’s register to the server’s register.

Syntax

RSETNREG (host_name, dest_idx, src_idx, option)
Input/Output parameters
host_name: [IN] String of host name or IP address of server.
dest_idx: [IN] Index of destination register. This integer is the index of the server register.
src_idx: [IN] Index of source register. This integer is the index of the client register.
option: [IN] This integer specifies the function of this program.

Value	Destination
0	Sets the register value and comment.
1	Sets the register value.
2	Sets the register comment.

Details

If the server does not have R[dest_idx], the error “DTBR-014 Bad variable or register index (host name, index)” will be posted.
If the client does not have R[src_idx], the error “VARS-024 Bad variable or register index" will be posted.

Example

The execution of the following program sets value and comment of R[20] of the client to R[10] of the host “SERVER”.

```
CALL RSETNREG (‘SERVER’, 10, 20, 0)
```

38.4.3 Program to Get a Position Register

Program name: RGETPREG

Overview

The client controller gets a position register from the server and sets it to a position register on the client. Position data is obtained only for the specified group.

Syntax

RGETPREG (host_name, src_idx, src_grp, dest_idx, dest_grp, option)
Input/Output parameters
host_name: [IN] String of the host name or IP address of the server.
src_idx: [IN] Index of the source position register. This integer is the index of the PR on the server.
src_grp: [IN] Group number of the source position register. This integer is the group number of the server.
dest_idx: [IN] Index of the destination position register. This integer is the index of the PR on the client.
dest_grp: [IN] Group number of the destination position register. This integer is the group number of the client.
option: [IN] This integer specifies the function of this program.

Value	Description
0	Gets the position value and comment.
1	Gets the position value.

Value	Description
2	Gets the position comment.

Details

If the contents of the second through the fifth parameters are incorrect, the following errors will be posted.

Error	Cause
DTBR-014 Bad variable or register index (host name, index)	There is no PR[src_idx] on the server.
DTBR-015 Illegal group number (host name, index)	There is no group “src_grp” on the server.
VARS-024 Bad variable or register index	There is no PR[dest_idx] on the client.
ROUT-026 Illegal group number	There is no group dest_grp” on the client.

NOTE

The position data of the specified groups on the client and server may differ in the following ways:

- The number of axes
- Configuration of Cartesian positions
- Type of axis (rotary or linear)

You might not be able to use the position register that is obtained or set. Use the transferred data with caution.

NOTE

Even if the Cartesian position data is obtained and stored to a client’s group that does not support Cartesian data (independent axis, for example), Cartesian data will still be set.

Example

The execution of the following program by the client transfers the position data of group 2 of PR[10] of the host “SERVER” to the client’s group 1 of PR[20]. The comment is not acquired.

```
CALL RGETPREG (‘SERVER’, 10, 2, 20, 1)
```

38.4.4 Program to Set a Position Register

Program name: RSETPREG

Overview

The client controller sets the client’s position register to the server’s position register. Position data is set only for the specified group.

Syntax

RSETPREG (host_name, dest_idx, dest_grp, src_idx, src_grp, option)
Input/Output parameters
host_name: [IN] String of the host name or IP address of the server.
dest_idx: [IN] Index of the destination position register. This integer is the index of the server PR .
dest_grp: [IN] Group number of the position data of the destination server PR . This is an INTEGER.
src_idx: [IN] Index of the source position register. This integer is the index of client PR.
src_grp: [IN] Group number of the position data of the source client PR. This is an INTEGER.
option: [IN] This integer specifies the function of this program.

Value	Description
0	Sets the position data and comment.
1	Sets the position data.

Value	Description
2	Sets the position comment.

Detail

If the contents of the second through fifth parameters are incorrect, the following errors will be posted.

Error	Cause
DTBR-014 Bad variable or register index (host name, index)	There is no PR[dest_idx] on the server.
DTBR-015 Illegal group number(host name, index)	There is no group " dest_grp" on the server.
VARS-024 Bad variable or register index	There is no PR[src_idx] on the client.
ROUT-026 Illegal group number	There is no group "src_grp" on the client.

NOTE

The position data of the specified groups on the client and server may be different in its components. For example, the following points may differ in the following ways:

- The number of axes
- Configuration of Cartesian positions
- Type of axis (rotary or linear)

You might not be able to use position register that is obtained or set. Use the transferred data with caution.

NOTE

Even if the Cartesian position data is set to a server's group that does not support Cartesian data (independent axis, for example), Cartesian data will still be set.

Example

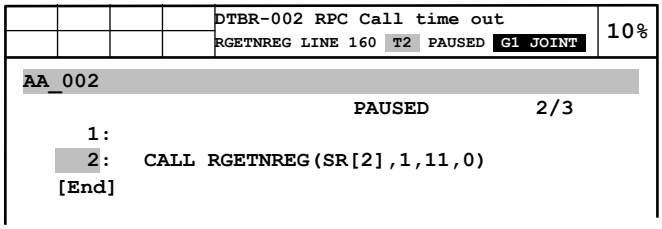
The execution of following program by the client transfers the position data of group 1 of PR[20] on the client to group 2 of PR[10] on host "SERVER".

```
CALL RSETPREG ('SERVER', 10, 2, 20, 1, 0)
```

38.5 RECOVERY FROM ERROR

When a standard data transfer program is executed and stops with an error, there are two major possible causes:

1 "DTBR-002 RPC Call timed out" is posted because communication failed or was not performed in time.

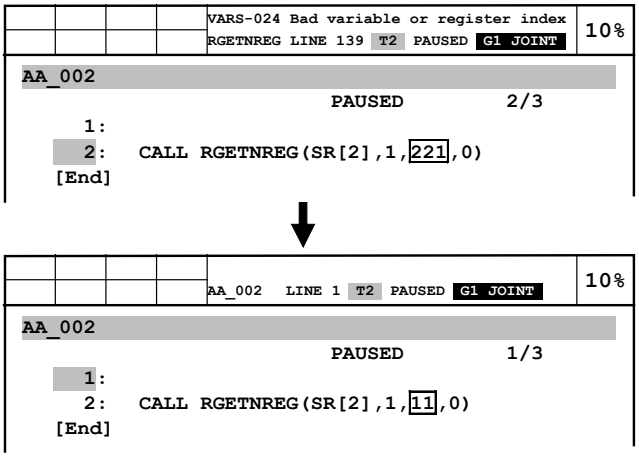


Check to see if communication between the controller (client) and specified host (server) is always stable. Perform a PING on the PING screen of the Host Comm setup screen.
Reduce the network load.

If standard data transfer programs or built-ins are running frequently, reduce the frequency.
If you are executing standard data transfer programs or built-ins continuously, leave an interval between them. If there is already an interval, increase it.

If you resume a program without moving the cursor, the same transfer will be performed again.
If there are no problems with the parameters and TCP/IP settings and there is still a network problem, resolve the network problem and resume execution.

2 Incorrect arguments
Modify the arguments to correct them. On the program edit screen, the cursor is on the line that calls the standard data transfer program. However, execution stopped inside the called program. The arguments were already passed to the subprogram (in the following figure, the subprogram is RGETNREG).
Even if you modify arguments on the program edit screen and resume from the same line, the problem is still not solved. This is because the argument values were already passed to the subprogram. If the arguments are registers or string registers, the value was passed at the subprogram call. Even if you change the value of the register or string register after an error occurs and you resume the program from the same line, the change will not be reflected.
After modifying arguments, move cursor to **one line before** the CALL instruction and resume from that line.



38.6 KAREL BUILT-INS

KAREL built-in routines (or functions) are pre-defined functions in the KAREL language. This function provides built-in routines to transfer data between robots. A KAREL program using these built-ins can

- Send and receive registers
- Send and receive position registers
- Send and receive string registers
- Read and write I/O values
- Read I/O simulation status
- Simulate and unsimulate I/O
- Read and write comments for the following:
  - Registers
  - Position registers
  - String registers
  - I/O

The following table lists the built-in routines provided by this function.

Built-in	Description
RGET_PORTCMT	Gets an I/O comment.
RGET_PORTSIM	Gets the I/O simulation status.
RGET_PORTVAL	Gets an I/O value.
RGET_PREGCMT	Gets a position register comment.
RGET_REG	Gets a register.
RGET_REG_CMT	Gets a register comment.
RGET_SREGCMT	Gets a string register comment.
RGET_STR_REG	Gets a string register.
RNREG_RECV	Transfers the server's register to the client's register.
RNREG_SEND	Transfers the client's register to the server's register.
RPREG_RECV	Transfers the server's position register to the client's position register.
RPREG_SEND	Transfers the client's position register to the server's position register.
RSET_INT_REG	Sets a register to an INTEGER value.
RSET_PORTCMT	Sets an I/O comment.
RSET_PORTSIM	Simulates I/O.
RSET_PORTVAL	Sets an I/O value.
RSET_PREGCMT	Sets a position register comment.
RSET_REALREG	Sets a register to a REAL value.
RSET_REG_CMT	Sets a register comment.
RSET_SREGCMT	Sets a string register comment.
RSET_STR_REG	Sets a string register.

38.6.1 RGET_PORTCMT Built-in ROUTINE

Purpose:

Allows a KAREL program to determine a comment set for a specified logical port of a remote host.

Syntax:

RGET_PORTCMT (host_port, port_type, port_no, comment_str, status)

Input/Output parameters

[in] host_port: STRING
[in] port_type: INTEGER
[in] port_no: INTEGER
[out] comment_str: STRING
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- **host_port** is the IP address or host name of remote robot controller.
- **port_type** specifies the code for the type of port whose comment is returned. Codes are defined in kliotyps.kl.
- **port_no** specifies the port number whose comment is being returned.
- **comment_str** is returned with the comment for the specified port. This should be declared as a STRING with a length of at least 16 characters. Only 16 characters are returned even if the server's comment is longer.
- **status** explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

The following program gets the comment of DO[10] of host RC1.

```
PROGRAM RGTPCMT
%COMMENT='RC1 DO[10] CMT'
```

```
%NOLOCKGROUP
%ENVIRONMENT RPCC
%INCLUDE kliotyps

VAR
  status  : INTEGER
  comment : STRING[16]
BEGIN
  RGET_PORTCMT('RC1', io_dout, 10, comment, status)
END RGTPCMT
```

38.6.2 RGET_PORTSIM Built-in ROUTINE

Purpose:

Gets port simulation status from a remote controller.

Syntax:

RGET_PORTSIM (host_port, port_type, port_no, simulated, status)
Input/Output parameters
[in]host_port: STRING
[in]port_type: INTEGER
[in]port_no: INTEGER
[out]simulated: BOOLEAN
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- **host_port** is the IP address or host name of the remote robot controller.
- **port_type** specifies the code for the type of port whose simulation status is returned. Codes are defined in kliotyps.kl.
- **port_no** specifies the port number whose simulation status is being returned.
- **simulated** returns TRUE if the port is being simulated, FALSE otherwise.
- **status** explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

The following program gets the simulation status of DI[8] of host RC1.

```
PROGRAM RGTPSIM
%COMMENT = 'RC1 DIN[8] SIM'
%NOLOCKGROUP
%ENVIRONMENT RPCC
%include kliotyps
VAR
  status  : INTEGER
  sim_stat: BOOLEAN
BEGIN
  RGET_PORTSIM('RC1', io_din, 8, sim_stat, status)
END RGTPSIM
```

38.6.3 RGET_PORTVAL Built-in ROUTINE

Purpose:

Allows a KAREL program to determine the current value of a specified logical port of a remote host.

Syntax:

RGET_PORTVAL (host_port, port_type, port_no, port_value, status)
Input/Output parameters
[in]host_port: STRING
[in]port_type: INTEGER
[in]port_no:INTEGER
[out]port_value:INTEGER
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- **host_port** is the IP address or host name of the remote robot controller.
- **port_type** specifies the code of the port whose value is being returned. Codes are defined in kliotyps.kl.
- **port_no** specifies the port number whose value is returned.
- **port_value** is returned with the current value (status) of the specified port. For BOOLEAN port types (DIN for example), this will be 0 = OFF, or 1 = ON.
- **status** explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

The following program gets the value of RO [3] of host RC1.

```
PROGRAM RGTPVAL
%COMMENT='GET RC1 RO[3]'
%NOLOCKGROUP
%ENVIRONMENT RPCC
%include kliotyps
VAR
  status  : INTEGER
  port_val: INTEGER
BEGIN
  RGET_PORTVAL('RC1', io_rdo, 3, port_val, status)
END RGTPVAL
```

38.6.4 RGET_PREGCMT Built-in ROUTINE

Purpose:

Retrieves a position register comment from a remote host.

Syntax:

RGET_PREGCMT (host_port, register_no, comment_str, status)
Input/Output parameters
[in]host_port: STRING
[in]register_no: INTEGER
[out]comment_str : STRING
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- **host_port** is IP address or host name of the remote robot controller.
- **register_no** specifies the position register from which to retrieve the comment. The comment of the given position register is returned in the **comment_str** parameter.
- **comment_str** should be declared as a STRING with a length of at least 16 characters.
- **status** explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

The following program gets the PR[5] comment of host RC1.

```
PROGRAM RGTPRCMT
%COMMENT='RC1 PR[5] CMT'
%NOLOCKGROUP
%ENVIRONMENT RPCC
VAR
  status  : INTEGER
  comment : STRING[16]
BEGIN
  RGET_PREGCMT('RC1', 5, comment, status)
END RGTPRCMT
```

38.6.5 RGET_REG Built-in ROUTINE

Purpose:

Gets an INTEGER or REAL value from the specified register of a remote host.

Syntax:

RGET_REG (host_port, register_no, real_flag, int_value, real_value, status)
Input/Output parameters
[in]host_port: STRING
[in]register_no: INTEGER
[out]real_flag: BOOLEAN
[out]int_value: INTEGER
[out]real_value: REAL
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- **host_port** is the IP address or host name of the remote robot controller.
- **register_no** specifies the register to get.
- **real_flag** is set to TRUE and**real_value**" is set to the register content if the specified register has a real value. Otherwise, **real_flag** is set to FALSE and **int_value** is set to the contents of the register.
- **status** explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

The following program gets value of R[2] of host RC1.

```
PROGRAM RGTNREG
%COMMENT='GET RC1 R[2]'
%NOLOCKGROUP
%ENVIRONMENT RPCC
VAR
  status   : INTEGER
  is_real  : BOOLEAN
  int_val  : INTEGER
  real_val : REAL
BEGIN
  RGET_REG('RC1', 2, is_real, int_val, real_val, status)
END RGTNREG
```

38.6.6 RGET_REG_CMT Built-in ROUTINE

Purpose:

Gets the comment from the specified register of a remote host.

Syntax:

RGET_REG_CMT (host_port, register_no, comment_str, status)
Input/Output parameters
[in] host_port: STRING
[in] register_no: INTEGER
[out] comment_str: STRING
[out] status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- **host_port** is the IP address or host name of the remote robot controller.
- **register_no** specifies the register from which to retrieve the comment. The comment of the given register is returned in **comment_str**.
- **comment_str** should be declared as a STRING with a length of at least 16 characters.
- **status** explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

The following program gets the R[2] comment of host RC1.

```
PROGRAM RGTNRCMT
%COMMENT='GET RC1 R[2] CMT'
%NOLOCKGROUP
%ENVIRONMENT RPCC
VAR
  status  : INTEGER
  comment : STRING[16]
BEGIN
  RGET_REG_CMT('RC1', 2, comment, status)
END RGTNRCMT
```

38.6.7 RGET_SREGCMT Built-in ROUTINE

Purpose:

Gets the comment from the specified string register of a remote host.

Syntax:

RGET_SREGCMT (host_port, register_no, comment_str, status)
Input/Output parameters
[in]host_port: STRING
[in]register_no: INTEGER
[out]comment_str: STRING
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- **host_port** is the IP address or host name of the remote robot controller.
- **register_no** specifies the string register from which to retrieve the comment.
- **comment_str** contains the comment of the specified string register.
- **comment_str** should be declared as a STRING with a length of at least 16 characters.
- **status** explains the status of the attempted operation. If not equal to 0, then an error occurred.

Examples:

The following program gets the SR[20] comment of host RC1.

```
PROGRAM RGTSRCMT
```

```
%NOLOCKGROUP
%COMMENT = 'GET CMT SR[20]'
%ENVIRONMENT RPCC
VAR
  status  : INTEGER
  comment : STRING[16]
BEGIN
  RGET_SREGCMT('RC1', 20, comment, status)
END RGTSRCMT
```

38.6.8 RGET_STR_REG Built-in ROUTINE

Purpose:

Gets the value from a specified string register of a remote host.

Syntax:

RGET_STR_REG (host_port, register_no, value, status)
Input/Output parameters
[in] host_port: STRING
[in] register_no: INTEGER
[out] value : STRING
[out] status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- **host_port** is the IP address or host name of the remote robot controller.
- **register_no** specifies the string register to get.
- **value** contains the value of the specified string register.
- **status** explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

The following program gets value of SR[10] of host RC1.

```
PROGRAM RGTSREG
%NOLOCKGROUP
%ENVIRONMENT RPCC
%COMMENT = 'GET SR[10]'
VAR
  status   : INTEGER
  sreg_val : STRING[253]
BEGIN
  RGET_STR_REG('RC1', 10, sreg_val, status)
END RGTSREG
```

38.6.9 RNUMREG_RECV Built-in ROUTINE

Purpose:

Transfers registers from a server (another controller) to registers in the client (the controller itself, which executes the built-in).

Syntax:

RNUMREG_RECV (host_port, src_idx, dest_idx, option, status)
Input/Output parameters
[in]host_port: STRING
[in]src_idx : INTEGER

[in]dest_idx : INTEGER
[in]option : INTEGER
[in]status : INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- **host_port** is the IP address or host name of the remote robot controller.
- **src_idx** specifies the index of the server register.
- **dest_idx** specifies index of client register. Acquired data is stored in this register.
- **option** specifies the function of this built-in, as follows.

Value	Description
0	Gets the register value and comment.
1	Gets the register value.
2	Gets the register comment.

- **status** explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

The following program gets the value and comment of R[3] of RC1 and sets them to R[12].

```
PROGRAM RNREGRCV
%COMMENT='RC1 R3->12'
%NOLOCKGROUP
%ENVIRONMENT RPCC

VAR
  STATUS : INTEGER
BEGIN
  RNUMREG_RECV('RC1', 3, 12, 0, STATUS)
END RNREGRCV
```

38.6.10 RNUMREG_SEND Built-in ROUTINE

Purpose:

Transfers registers from a client (the controller itself, which executes the built-in) to registers on the server (another controller).

Syntax:

RNUMREG_SEND (host_port, dest_idx, src_idx, option, status)
Input/Output parameters
[in]host_port: STRING
[in]dest_idx : INTEGER
[in]src_idx : INTEGER
[in]option : INTEGER
[in]status : INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- **host_port** is the IP address or host name of the remote robot controller.
- **dest_idx** specifies the index of the server register.
- **src_idx** specifies the index of the client register. This register data is sent to the server.
- **option** specifies the unction of this built-in, as follows.

Value	Descrition
0	Sets the register value and comment.

Value	Descrition
1	Sets the register value.
2	Sets the register comment.

- **status** explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

The following program sends the value of R[11] to R[2] of the host RC1.

```
PROGRAM RNREGSND
%COMMENT='R11->RC1 R1'
%NOLOCKGROUP
%ENVIRONMENT RPCC

VAR
  STATUS : INTEGER
BEGIN
  RNUMREG_SEND('RC1', 2, 11, 1, STATUS)
END RNREGSND
```

38.6.11 RPOSREG_RECV Built-in ROUTINE

Purpose:

Transfers the position register of a specified group of the server (another controller) to the position register of the specified group of the client (the controller itself, which executes the built-in).

Syntax:

RPOSREG_RECV (host_port, src_idx, src_grp, dest_idx, dest_grp, option, status)
Input/Output parameters
[in]host_port: STRING
[in]src_idx: INTEGER
[in]src_grp: INTEGER
[in]dest_idx: INTEGER
[in]dest_grp: INTEGER
[in]option: INTEGER
[in]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- **host_port** is the IP address or host name of the remote robot controller.
- **src_idx** specifies the index of the server position register.
- **src_grp** specifies the group number of the server. Position data of the specified group is transferred.
- **dest_idx** specifies the index of the client position register. The specified position register stores the acquired data.
- **dest_grp** specifies the group number of client. The position data of the specified group of PR[**dest_idx**] is changed to the received data.
- **option** specifies the function of this built-in, as follows.

Value	Description
0	Gets the position data and comment.
1	Gets the position data.
2	Gets the position comment.

- **status** explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

The following example program performs the following transfer, and also transfers the comment.
From: Group 1 of PR[5] of host RC1
To: Group 2 of PR[6]

```
PROGRAM RPREGRCV
%COMMENT='RC1 PR5G1->PR6G2'
%NOLOCKGROUP
%ENVIRONMENT RPCC

VAR
  STATUS : INTEGER
BEGIN
  RPOSREG_RECV('RC1', 5, 1, 6, 2, 0, STATUS)
END RPREGRCV
```

38.6.12 RPOSREG_SEND Built-in ROUTINE

Purpose:

Transfer a client position register of a specified group to a server position register of a specified group.

Syntax:

RPOSREG_SEND (host_port, dest_idx, dest_grp, src_idx, src_grp, option, status)
Input/Output parameters
[in]host_port: STRING
[in]dest_idx: INTEGER
[in]dest_grp: INTEGER
[in]src_idx: INTEGER
[in]src_grp: INTEGER
[in]option: INTEGER
[in]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- **host_port** is the IP address or host name of the remote robot controller.
- **dest_idx** specifies the index of the server position register.
- **dest_grp** specifies the server group number. The position data of the specified group is changed by data sent from the client.
- **src_idx** specifies the index of the client position register. Position data of group **src_grp** of PR[**src_idx**] is sent to the server.
- **src_grp** specifies the group number of the client.
- **option** specifies the function of this built-in as follows

Value	Description
0	Sets the position data and comment.
1	Sets the position data.
2	Sets the position comment.

- **status** explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

The following example program performs the following transfer, and also transfers the comment.
From: Group 2 of PR[7] of the client
To: Group1 of PR[8] of host RC1

```
PROGRAM RPREGSND
```

```
%COMMENT='PR7G2->RC1 PR8G1'
%NOLOCKGROUP
%ENVIRONMENT RPCC

VAR
  STATUS : INTEGER
BEGIN
  RPOSREG_SEND('RC1', 8, 1, 7, 2, 0, STATUS)
END RPREGSND
```

38.6.13 RSET_INT_REG Built-in ROUTINE

Purpose:

Stores an INTEGER value in the specified register of a remote host.

Syntax:

RSET_INT_REG (host_port, register_no, int_value, status)
Input/Output parameters
[in]host_port: STRING
[in]register_no: INTEGER
[in]int_value: INTEGER
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- **host_port** is the IP address or host name of the remote robot controller.
- **register_no** specifies the register into which **int_value** will be stored.
- **status** explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

The following program sets 100 to R[20] of host RC1.

```
PROGRAM RSTNRI
%COMMENT = 'RC R[20]=100'
%NOLOCKGROUP
%ENVIRONMENT RPCC
VAR
  status : INTEGER
BEGIN
  RSET_INT_REG('RC1', 20, 100, status)
END RSTNRI
```

38.6.14 RSET_PORTCMT Built-in ROUTINE

Purpose:

Allows a KAREL program to set the comment of a specified logical port of a remote host.

Syntax:

RSET_PORTCMT (host_port, port_type, port_no, comment_str, status)
Input/Output parameters
[in]host_port: STRING
[in]port_type: INTEGER
[in]port_no: INTEGER
[in]comment_str: STRING
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- **host_port** is the IP address or host name of the remote robot controller.
- **port_type** specifies the code of theport whose comment is being set. Codes are defined in kliotyps.kl.
- "**port_no** specifies the port number whose comment is being set.
- **comment_str** is a string whose value is the comment for the specified port. This must not be over 16 characters long.
- **status** explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

The following program sets a comment to DO[10] of the host RC1.

```
PROGRAM RSTPCMT
%COMMENT ='RC1 DO[10] cmt'
%NOLOCKGROUP
%ENVIRONMENT RPCC
%INCLUDE kliotyps
VAR
  status : INTEGER
BEGIN
  RSET_PORTCMT('RC1', io_dout, 10, 'RC1 DOUT[10]', status)
END RSTPCMT
```

38.6.15 RSET_PORTSIM Built-in ROUTINE

Purpose:

Simulates a port on a remote host.

Syntax:

RSET_PORTSIM (host_port, port_type, port_no, value, status)
Input/Output parameters
[in]host_port: STRING
[in]port_type: INTEGER
[in]port_no: INTEGER
[in]value: INTEGER
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- **host_port** is the IP address or host name of the remote robot controller.
- **port_type** specifies the code for the type of port to be simulated. Codes are defined in kliotyps.kl.
- **port_no** specifies the port number to be simulated.
- **value** specifies the initial value to set.
- **status** explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program simulates GO[3] of RC1 and sets its initial simulated value to 5.

```
PROGRAM RSTPSIM
%COMMENT ='RC1 SIM GO[3], 5'
%NOLOCKGROUP
%ENVIRONMENT RPCC
%INCLUDE kliotyps

VAR
  status  : INTEGER
```

```
BEGIN
  RSET_PORTSIM('RC1', io_gpout, 3, 5, status)
END RSTPSIM
```

38.6.16 RSET_PORTVAL Built-in ROUTINE

Purpose:

Allows a KAREL program to set a specified output (or simulated input) for a specified logical port.

Syntax:

RSET_PORTVAL (host_port, port_type, port_no, port_value, status)
Input/Output parameters
[in]host_port: STRING
[in]port_type: INTEGER
[in]port_no: INTEGER
[in]port_value: INTEGER
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- **host_port** is the IP address or host name of the remote robot controller.
- **port_type** specifies the code of the port whose value is being set. Codes are defined in kliotyps.kl.
- **port_no** specifies the port number whose value is being set.
- **port_value** indicates the value to be assigned to a specified port. If the **port_type** is BOOLEAN (DOUT for example), this should be 0 = OFF, or 1 = ON. This field can be used to set input ports if the port is simulated.
- **status** explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

The following program sets DO[3] of host RC1 to ON.

```
PROGRAM RSTPVAL
%COMMENT ='RC1 DO[3]=ON'
%NOLOCKGROUP
%ENVIRONMENT RPCC
%INCLUDE kliotyps

VAR
  status  : INTEGER
BEGIN
  RSET_PORTVAL('RC1', io_dout, 3, 1, status)
END RSTPVAL
```

38.6.17 RSET_PREGCMT Built-in ROUTINE

Purpose:

Sets the comment of a position register of a remote host.

Syntax:

RSET_PREGCMT (host_port, register_no, comment_str, status)
Input/Output parameters
[in]host_port: STRING
[in]register_no: INTEGER
[in]comment_str : STRING
[out]status: INTEGER

%ENVIRONMENT Group: RPCC

Detail:

- **host_port** is the IP address or host name of the remote robot controller.
- **register_no** specifies the position register of the remote robot controller.
- **comment_str** is the comment to be set. If **comment_str** exceeds 16 characters, it is truncated.
- **status** explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

The following program sets comment RC1 PR[5] to PR[5] of host RC1.

```
PROGRAM RSTPRCMT
%COMMENT ='RC1 PR[5] CMT'
%NOLOCKGROUP
%ENVIRONMENT RPCC
VAR
  status  : INTEGER
BEGIN
  RSET_PREGCMT('RC1', 5, 'RC1 PR[5]', status)
END RSTPRCMT
```

38.6.18 RSET_REALREG Built-in ROUTINE

Purpose:

Stores a REAL value in the specified register of a remote host.

Syntax:

RSET_REALREG (host_port, register_no, real_value, status)
Input/Output parameters
[in] host_port: STRING
[in] register_no: INTEGER
[in] real_value: REAL
[out] status: INTEGER
%ENVIRONMENT RPCC

Detail:

- **host_port** is the IP address or host name of the remote robot controller.
- **register_no** specifies the register into which the **real_value** will be stored.
- **status** explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

The following program sets -12.3 to R[20] of RC1.

```
PROGRAM RSTNRR
%COMMENT = 'RC1 R[20]=-12.3'
%NOLOCKGROUP
%ENVIRONMENT RPCC
VAR
  status : INTEGER
BEGIN
  RSET_REALREG('RC1', 20, -12.3, status)
END RSTNRR
```

38.6.19 RSET_REG_CMT Built-in ROUTINE

Purpose:

Sets the numeric register comment of a remote host.

Syntax:

RSET_REG_CMT (host_port, register_no, comment_str, status)
Input/Output parameters
[in] host_port: STRING
[in] register_no: INTEGER
[in] comment_str: STRING
[out] status: INTEGER
%ENVIRONMENT RPCC

Detail:

- **host_port** is the IP address or host name of the remote robot controller.
- **register_no** specifies which register for which to set the comment.
- **comment_str** represents the data used to set the comment of the given register.
  If **comment_str** exceeds 16 characters, the built-in will truncate the string.
- **status** explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

The following program sets comment “RC1 R[15]” to R[15] of host RC1.

```
PROGRAM RSTNRCMT
%COMMENT ='RC1 R[15] CMT'
%NOLOCKGROUP
%ENVIRONMENT RPCC
VAR
  status  : INTEGER
BEGIN
  RSET_REG_CMT('RC1', 15, 'RC1 R[15]', status)
END RSTNRCMT
```

38.6.20 RSET_SREGCMT Built-in ROUTINE

Purpose:

Sets the comment for the specified string register of a remote robot controller.

Syntax:

RSET_SREGCMT (host_port, register_no, comment_str, status)
Input/Output parameters
[in]host_port: STRING
[in]register_no: INTEGER
[in]comment_str : STRING
[out]status: INTEGER
%ENVIRONMENT Group:RPCC

Detail:

- **host_port** is the IP address or host name of the remote robot controller.
- **register_no** specifies the string register to set.
- **comment_str** contains the comment to set to the specified string register.
  If the **comment_str** exceeds 16 characters, the built-in will truncate the string.
- **status** explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

The following program sets the comment “RC1_STRING_REG_3” to SR[3] of host RC1.

```
PROGRAM RSTSRCMT
%COMMENT ='RC1 SR[3] CMT'
%NOLOCKGROUP
%ENVIRONMENT RPCC
VAR
  status  : INTEGER
BEGIN
  RSET_SREGCMT('RC1', 3, 'RC1_STRING_REG_3', status)
END RSTSRCMT
```

38.6.21 RSET_STR_REG Built-in ROUTINE

Purpose:

Sets the specified value for the specified string register of a remote host.

Syntax:

RSET_STR_REG (host_port, register_no, value, status)
Input/Output parameters
[in] host_port: STRING
[in] register_no: INTEGER
[in] value : STRING
[out] status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- **host_port** is the IP address or host name of the remote robot controller.
- **register_no** specifies the string register to set.
- **value** contains the value to set to the specified string register.
- **status** explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

The following program sets SR[4] of host RC1.

```
PROGRAM RSTSREG
%COMMENT ='Set RC1 SR[4]'
%NOLOCKGROUP
%ENVIRONMENT rpcc
CONST
  ERR_SUCCESS = 0
VAR
  status    : INTEGER
BEGIN
  RSET_STR_REG('RC1', 4, 'RC1 of SR[4] was set by remote host', status)
END RSTSREG
```

38.7 TIMEOUT AND RETRY

If communication is not successful within a certain period of time due to a network problem, this function disconnects the connection and tries to communicate again. If the communication is not successful after the second attempt, the default data transfer program will automatically post a "DTBR-002 RPC Call timed out" alarm. The status parameter of the built-in will be set to the corresponding value.
The default timeout value is 2 seconds. The number of retries is 2. If the communication fails completely, this function gives up after about 6 seconds.
If controller does not recognize the host name at all, communication might be abandoned before the timeout period elapses.
This function does stop trying to communicate and retrying even if the program is paused or aborted.

38.8 LIMITATIONS

- Communication performance is not guaranteed, because it depends on network traffic and the processing load on the controller.
- This function affects the performance of other functions that use Ethernet and is also affected by these other functions.
- Even if multiple tasks (programs) make communication requests, the data transfer is not processed in parallel, but are actually processed sequentially.
- Data transfer cannot be restricted by other functions, such as the password function.
- A robot controller cannot send requests to itself.
- The transfer of position registers pays attention to the number of axes or extended axes so that the data to be set at the destination does not exceed the axes on the destination controller. However, no other position data elements are considered.
  For example, the following position elements are not considered:
  - Whether the robot can have Cartesian data
  - Axis limits
  - Position configuration and turn number range
  - Type of axis (rotary or linear)
- Pausing or aborting the program does not stop communication once it has been started.
  The next request should be issued after the previous one has been completed.
- R-50iA controller cannot transfer with R-30iB Plus or earlier controller.

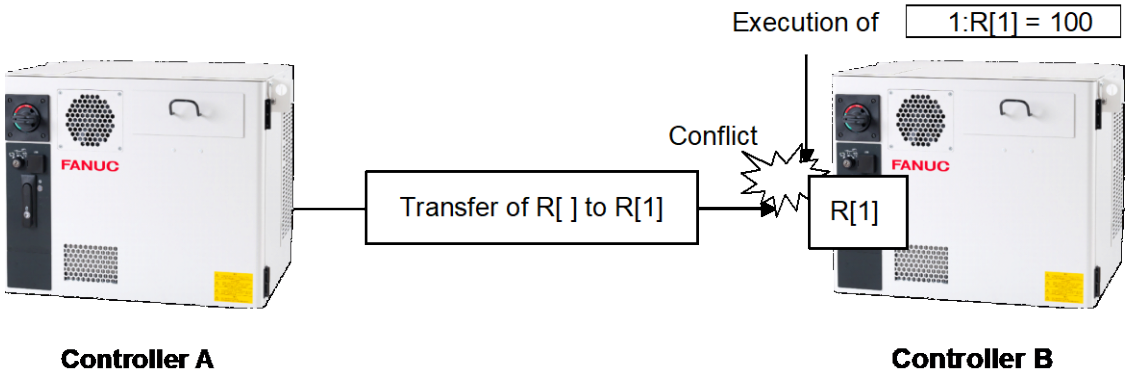
38.9 CAUTIONS

- This function does not replace hardwired I/O or field networks.
  This function offers the ability to read and write the I/O values of a remote controller. However, hardwired I/O and field network I/O are always much faster.
- Cyclic execution increases network load and controller processing load. Avoid continuous or cyclic execution of data transfer.
- Position register transfer cautions:
  The content of a position register depends on the robot type and position configuration.
  Use transferred position registers only after carefully examining their contents.
  Be especially careful when using transferred position registers as the destination positions in motion instructions.
  The server and client robots are not always the same type. The position register will be interpreted depending on the currently selected tool and user frames.

38.10 CONFLICT BETWEEN DATA WRITES

When transferring registers or position registers, note that only one controller changes a register or position register. This also applies when using string registers and I/O in the built-ins.

If both the server and client write to the same data, they will overwrite the value written by the other controller, making it look as if the data transfer was not working properly.
Specifically, if the timing of the client's and server's writes is very close, the earlier write might not be recognizable.



When the client writes a register, position register, or string register of the server, the value and comment are acquired first. Then the client overwrites the necessary part of the acquired data and requests the server to write the updated value as a whole.
Therefore, if the server writes slightly before the client, the value or comment may revert to the values that were initially acquired by the client.

38.11 TROUBLESHOOTING

"DTBR-002 RPC Call timed out" is posted.

Check to see if the communication between the controller (client) and specified host (server) is always stable. You can perform a PING on the PING screen of the Host Comm setup screen.
Reduce the network load.
If the standard data transfer program or built-ins are frequently executed, reduce the frequency.
If the standard data transfer program or built-ins are executed continuously, set a time interval between them. If there is already an interval, lengthen it.

Changes to program arguments are not reflected after resuming.

If you modified arguments in the default data transfer program, move the cursor to **one line before** the CALL instruction and resume the program from that line.

Communication takes time or the speed is unstable.

The data transfer speed of this function is not guaranteed.
Performance depends on network traffic and the CPU load on the client and server.
Reduce the network load.
Improve the timing of the data transfer. For example, data transfer that is not needed during robot motion can be done when the robot is not moving.

"INTP-320 (program name, line number) Undefined built-in" is posted.

If this symptom happens when the default data transfer program or built-in is executed, confirm whether the Data Transfer Between Robots optional function (J740) has been installed.

No timeout occurs and the program keeps running.

Pausing or aborting the program during data transfer does not stop the transfer once is has been started. Data transferring is attempted until the last retry fails.
The next data transfer request will not be processed until the previous data transfer completes (fails).

For example, consider the following case:

1 The Ethernet cable is not plugged in properly.
2 The standard data transfer program, RGETNREG, is called but the user aborted the program immediately on the FCTN menu.
3 The same program is executed again and RGETNREG is called immediately.

In this case, the second RGETNREG will time out after about twice the normal time has elapsed.
The default setting is about 12 seconds.
Once data transfer has started, it is recommended that you wait for the data transfer to be completed or an error to occur, rather than forcing an abort.

A timeout error occurs just after resuming the program.

Pausing a program does not stop data transfer once it has been started. If a timeout occurs while the program is paused, a timeout error will be posted when the program is resumed.

The value of the axis joint position of PR[ ] is not transferred properly.

If the axis type (rotary or linear) is different between the client's and server's corresponding axes, the transferred value will not be interpreted in the same way and will not be treated as the same value.

"INTP-311 (program name, line number) Uninitialized data is used" is posted when the transferred position register is used.

1 Confirm that the source data has been initialized.
2 The number of total axes or extended axes may be different.
If the source data does not have enough axes to set all axis data to the destination data, the axes that cannot be set because of the missing data will be uninitialized. In this case, set the data to the correct values before use.
For example, suppose group 1 of host RC1 has an extended axis and group 1 of RC2 has no extended axes.
If the Cartesian position of group 1 of RC2 is set to group 1 of RC1, E1 will be set to an uninitialized value.
You will have to manually set RC1 to a correct value. If 0 is appropriate, PR[1,7]=0 can be used.

The position data of the sent or received position register is partially uninitialized.

Refer to the previous item.
In a multi-group system, the position register screen will not display "R" unless the data of all groups has been initialized.
The standard transfer program and built-ins transfer only the specified groups.

"VARS-037 Position register is locked" is posted.

Check to see if the position register is locked on the server or client.

"HRTL-047 Address family not supported" is posted.

Make sure that you have not specified your own host name as the client.

This function cannot communicate with itself.

Data is sent to or received from a controller different than the one specified.

If you did not cycle power after changing the IP address and host name, cycle controller power.
Make sure that the host name does not include underscores.

“HRTL-049 Can't assign requested address” is posted.
“HOST-108 Internet address not found” is posted.

Verify that the specified host name or IP address is correct.
If you did not cycle power after changing the IP address and host name, cycle controller power.
Check whether the communication between the controller (client) and specified host (server) is always stable. You can perform PING on the PING screen of the Host Comm setup screen.

39 TOUCH SENSING

Touch sensing allows the robot to change a path automatically to compensate for workpiece displacement. Touch sensing consists of two phases: a search motion phase and a touch offset phase.

The Touch Sensing function (A05B-XXXX-J536) is required to use this function. This function was originally developed for ArcTool, so the explanations and figures in this chapter depict arc welding. You can also use this function with HandlingTool.

Search Motion

Search motion detects the current position of the workpiece, as follows:

- Moves the touch sensing part (for an arc welding robot, the wire tip is the TCP) toward the workpiece using predefined robot motion, speed, and direction.
- Uses an input signal to indicate that the robot has come into contact with the object.
- Stores the found location of the workpiece, or position offset information, in position registers.

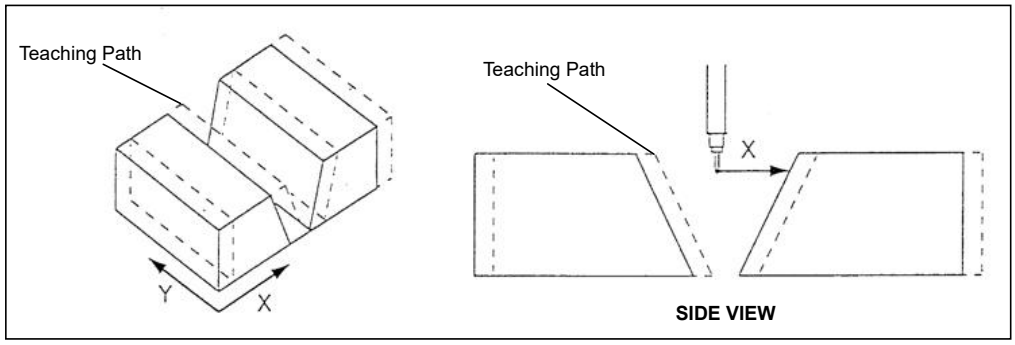


Fig. 39(a) Search motion for detecting a shift in the X direction

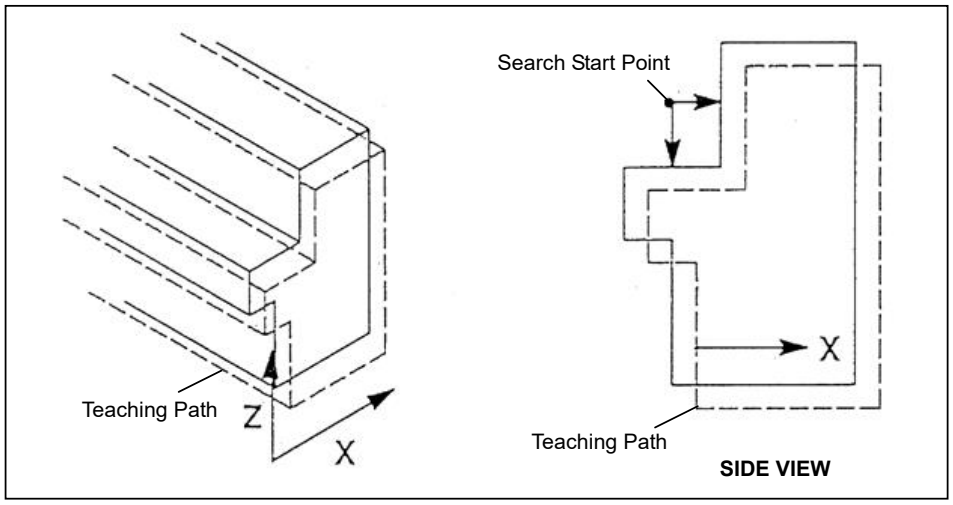


Fig. 39(b) Search motion for detecting a shift in the X and Z directions

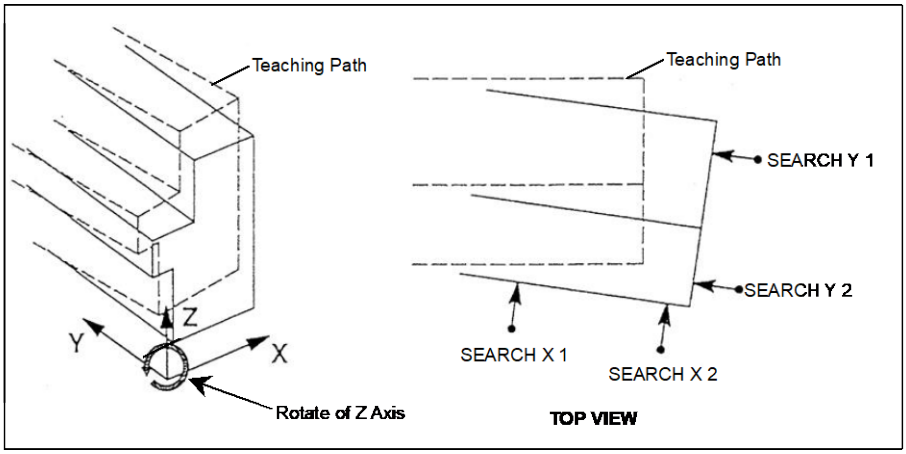


Fig. 39(c) Search motion for detecting a shift in the X and Y directions, and Z rotation

Touch Offset

Shift one or more taught positions in your program using the position stored in the position offset information.

Using Touch Sensing

To use touch sensing you must:

- Set up the robot Tool Center Point (TCP) correctly.
- Set up Touch Sensing hardware. The hardware monitors an input signal to determine when the robot comes into contact with the object. Some weld equipment models install touch sensing hardware by default, such as Lincoln weld equipment.

After finishing the above setup, perform the following setup, referring to specified sections in this chapter:

- **Assignment of Touch Sensing I/O (Section 39.1)**
  Assign I/O to enable and use the electrical interface circuit.
- **Setup of Touch Frame (Section 39.2)**
  Create a Touch Sensing frame to determine the search direction during search motion.
- **Touch Schedule (Section 39.4)**
  Set the method of approaching the workpiece (search pattern) and the method of storing the obtained position information (absolute position or position offset).
- **Touch Sensing Programming (Section 39.5)**
  Create a Touch Sensing program using Search and Touch Offset instructions.
- **Execution of Touch Sensing Program (Section 39.6)**
  Execute search motion with Master Flag set to ON to obtain the master position. After that, set Master Flag to OFF.

```
TOUCH_SENSING_1                                  13/13
    1:J  P[1] 50% FINE
    2:  Search Start[3] PR[3]
    3:J  P[2] 50% FINE
    4:J  P[3] 50% FINE Search[Y]
    5:J  P[4] 50% FINE
    6:J  P[5] 50% FINE Search[X]
    7:  Search End
    8:J  P[6] 50% FINE
    9:  Touch Offset PR[3]
   10:L  P[7] 100mm/sec FINE
     :  Weld Start[1,1]
   11:L  P[8] 80cm/min FINE
     :  Weld End[1,2]
   12:  Touch Offset End
[End]
```

Teach the Home Position
Start Search Motion. Use Touch Sch 3, PR[3]
Teach a search start position
Search in the Y direction
Teach another search start position
Search in the X direction
Finish Search Motion.
(If needed) Teach an intermediate point
The following points will be offset by PR[3]
P[7] is offset by PR[3]

P[8] is offset by PR[3]

Finish offsetting positions

Fig. 39(d) Example Touch Sensing program

39.1 TOUCH SENSING I/O ASSIGNMENT

To use Touch Sensing, you must assign the
- Input signal that the Touch Sensing circuit monitors to indicate when the robot has reached the object.
- Output signal that enables and disables the Touch Sensing circuit.

You must wire the necessary connections for the input and output signals to be used for Touch Sensing. The wire stick detection circuit on the process I/O board can also be used for Touch Sensing.

Input Signal (Touch Detect Signal)

The Touch Sensing input signal indicating contact with a workpiece is monitored by the Touch Sensing circuit. When the input is received, the current robot position is stored in a position register.
RI, DI, WI WS can be used as the Touch Sensing input signal.

REFERENCE: You can also set up Touch Sensing to monitor the condition of any RO or DO signal as an input signal. When the selected output turns on during a Touch Sensing routine, the controller reads this as a received input signal.

Output Signal (Touch Sensing Enable/Disable Command Signal)

RO, DO, WO, WSE can be used to enable the Touch Sensing circuit.

Assigning the Touch Sensing Inputs and Outputs

You must assign Touch Sensing inputs and outputs to match the hardware interface at your site. This involves assigning both input and output types and port numbers. Refer to Procedure 39-1.
After you have finished assigning Touch Sensing I/O, check whether Touch Sensing I/O works properly, using Procedure 39-2.

NOTE
To use a touch sensor circuit on the welding power supply for the Touch Sensing function, specify the port numbers of the I/O of the Touch Sensing signals on the Touch Sensing Setup screen after confirming them on the Weld I/O screen.

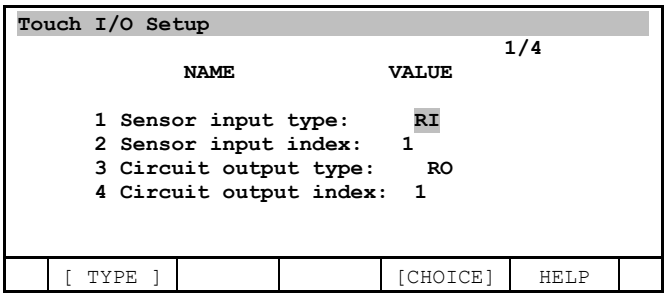
Procedure 39-1 Assigning Touch Sensing Input and Output Signals

Condition

- For multi group systems only, the motion group number is selected for assigning Touch Sensing I/O.

Step

1 Press MENU.
2 Select SETUP.
3 Press F1, [TYPE] and then select Touch I/O. The following screen will be displayed.



4 To assign Sensor (input) and Circuit (output) types:
  a Move the cursor to the line you want to assign.
  b Press F4, CHOICE.

c Move the cursor to the desired input or output type.
d Press ENTER.

5 To assign a Sensor and Circuit number:
a Move the cursor to the line you want to assign.
b Type the value and press ENTER.

NOTE
This screen shows the settings for the currently selected motion group. To view the settings for another motion group, change the motion group by selecting an CHANGE GROUP on the FCTN menu.

NOTE
After deciding which Touch Sensing I/O to use, include comments indicating that the signals are used for Touch Sensing. Perform this operation on each I/O signal screen.

NOTE
The ArcTool software checks the validity of the port type and port number when running a program that includes Touch Sensing. If the port type or number is invalid, the system will display an I/O invalid error message.

NOTE
Following Data IDs about Input Signal are supported.
- Data ID: $TOUCH_SENSING_IO_SETUP.SENSOR_INPUT_TYPE[g] / INTEGER / RO
The type of input signal that can be set is entered.
- Data ID: $TOUCH_SENSING_IO_SETUP.SENSOR_INPUT_INDEX[g] / INTEGER / RO
The number of input signal that can be set is entered.

Procedure 39-2 Testing Touch Sensing I/O

Condition

- A workpiece that can be detected by Touch Sensing is installed.
- Touch Sensing I/O has been assigned.

Step

1 Jog the robot near the workpiece.

⚠WARNING
If the Touch Sensing output signal is set ON in Step 2, current flow is generated in the Touch Sensing circuit. Therefore, do not touch the touch sensor part (on ArcTool, the wire tip). In addition, be sure to set the output signal to OFF.

2 Display the Touch I/O Setup screen for the Touch Sensing output signal set in Procedure 39-1. If the port type is DO, display the Digital Output signal screen. Move the cursor to the port number of the Touch Sensing output signal and press F4, ON. The signal will be output and the Touch Sensing circuit will be enabled.
3 Set the jog override speed to a low value. Jog the robot until the touch sensor part lightly touches the workpiece.

4 Display the Touch I/O Setup screen for the Touch Sensing Detect signal set in Procedure 39-1. If the port type is DI, display the Digital Input signal screen. Next, check the status of the port number for the Touch Sensing input signal. If the status is ON, Touch Sensing will work correctly. If the status is OFF even when the touch sensor part touches the workpiece, check the hardware connection or assignment of Touch Sensing I/O using Procedure 39-1.
5 Move the cursor to the output signal set ON in Step 2 and then press F5, OFF.

39.2 TOUCH FRAME SETUP

A touch frame determines the motion direction of the robot TCP (touch sensor part). A touch frame is defined by three points. The first point defines the origin or starting point. The second point defines the positive X direction of the touch frame. The third point defines the positive X-Y plane. The Z direction is created by the defined X, Y axes.
Fig. 39.2 shows how a touch frame is used in a Touch Sensing program. The orientation of the touch frame to the object is arbitrary in Fig. 39.2. The positive X axis could be aligned with the current Z direction. This would redefine positive Z to be in the opposite direction of the current positive X direction.

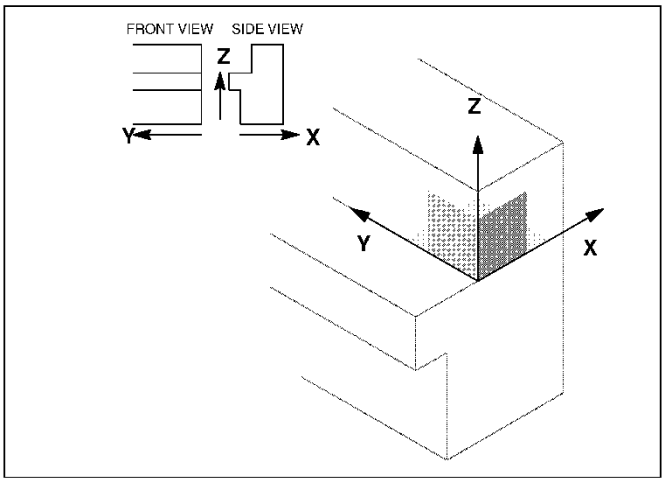


Fig. 39.2 Touch frame

NOTE
You can set up a maximum of 32 touch frames for each robot group. Touch frames are set up using the Touch Frame screen on the Setup menu.

NOTE
You must define a touch frame before you perform search motion in a program.

There are two ways to define touch frames: the teaching method and the direct entry method. The teaching method defines the touch frame by recording three points. The direct entry method defines the touch frame by the rotation angle values you enter on the Touch Sense Setup screen.
Normally, the Reference Group item on the Touch Frame Setup screen should be the same group number as the Robot Group, because the touch frame relates to the WORLD frame of the robot.
In a coordinated motion system, if you want to define a touch frame for the coordinated (reference) group, set the leader group number to the Reference Group and the follower group number to the Robot Group.
Table 39.2 lists and describes the items you must set to define a touch frame. For more details, refer to Section 39.9.

Table 39.2 Touch frame setup items

ITEM	DESCRIPTION
Frame Number	This item specifies the number of the touch frame you want to define.

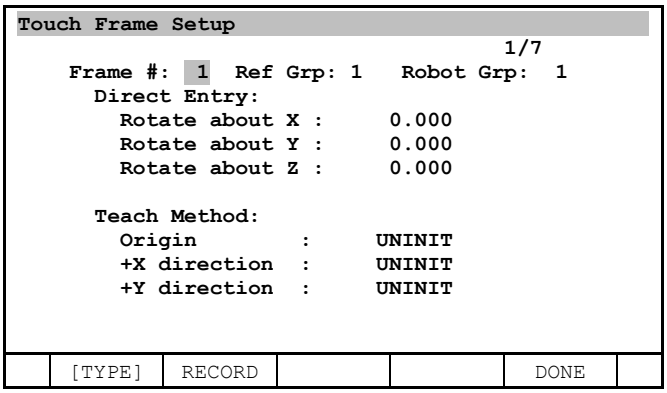
ITEM	DESCRIPTION
Reference Group	This item specifies the reference group you want to associate with the Touch Frame. Normally, you should specify the same group number as the Robot Group item described below. If you want to perform Coordinated Touch Sensing, refer to Section 39.9.
Robot Group	This item specifies the motion group for which the touch frame is used. You cannot select anon-robot group such as a positioner group. In a single group system, set the Robot Group to 1.
Direct Entry (Procedure 39-4)	
Rotate about X	This item specifies the rotation about X for the touch frame.
Rotate about Y	This item specifies the rotation about Y for the touch frame.
Rotate about Z	This item specifies the rotation about Z for the touch frame.
Teach Method (Procedure 39-3)	
Origin	This item allows you to record the origin of the touch frame.
+X direction	This item allows you to define the +X direction of the touch frame.
+Y direction	This item allows you to define the +Y direction of the touch frame.

Use Procedure 39-3 to define a touch frame using the teaching method. Use Procedure 39-4 to define a touch frame using the direct entry method.

Procedure 39-3 Setting Up of the Touch Frame Using the Teaching Method

Step

1 Press MENU.
2 Select SETUP.
3 Press F1, [TYPE].
4 Select Touch Frame. The following screen will be displayed.



5 Move the cursor to Robot Grp. Enter the robot group number and press ENTER.
6 Move the cursor to Frame Number. Enter the number of the frame to define and press ENTER.
7 Move the cursor to Ref Grp. Normally, enter the same group number as the Robot Group item, and press ENTER key. If you want to perform Coordinated Touch Sensing, enter the group number of the leader group.
8 Define the origin point of the Touch Frame:
  a Move the cursor to Origin.
  b Jog the robot TCP to the desired starting point (origin).
  c Press F2, RECORD. UNINIT will be changed to RECORDED.
9 Define the +X direction:
  a Move the cursor to +X direction.
  b Jog the robot TCP to a point along the +X axis of the touch frame.
  c Press F2, RECORD.
10 Define the +Y direction:

a Move the cursor to +Y direction.
b Jog the robot in the +Y direction of the touch frame, to a point on the X-Y plane.
c Press F2, RECORD.

11 Press F5, DONE to complete the definition of the frame. RECORDED will be changed to USED.

Procedure 39-4 Setting Up the Touch Frame Using Direct Entry Method

Step

1 Press MENU.
2 Select SETUP.
3 Press F1, [TYPE].
4 Select Touch Frame. The following screen will be displayed.

```
Touch Frame Setup
                                                1/7
    Frame #:  1  Ref Grp: 1   Robot Grp:  1
      Direct Entry:
        Rotate about X :      0.000
        Rotate about Y :      0.000
        Rotate about Z :      0.000

      Teach Method:
        Origin         :     UNINIT
        +X direction   :     UNINIT
        +Y direction   :     UNINIT

      [TYPE]  RECORD                          DONE
```

5 Move the cursor to Robot Grp. Enter the robot group number and press ENTER.
6 Move the cursor to Frame Number. Enter the number of the frame to define and press ENTER.
7 Move the cursor to Ref Grp. Normally, enter the same group number as the Robot Group item, and press ENTER. If you want to perform Coordinated Touch Sensing, enter the group number of the leader group.
8 Define the rotation angle about X:
   a Move the cursor to Rotate about X.
   b Enter the value (in degrees).
9 Define the rotation angle about Y:
   a Move the cursor to Rotate about Y.
   b Enter the value (in degrees).
10 Define the rotation angle about Z:
   a Move the cursor to Rotate about Z.
   b Enter the value (in degrees).
11 Press F5, DONE to complete the definition of the frame. UNINIT will be changed to USED.

> **⚠ CAUTION**
> Do not forget to press F5, DONE at the end of Procedures 39-3 and 39-4. If you forget this operation, the touch frame will not be defined.

NOTE

If you change the value of the Ref Grp for an initialized frame in Step 7 of Procedures 39-3 and 39-4, the following warning message will be displayed:

```
Frame data will be cleared!
[ ] [ ] [ ] [Yes] [No] [ ] [ ]
```

If you press F3, Yes, the frame data will be reinitialized.
If you set a different number than the Robot Group number to the Ref Grp, but coordinated pair calibration is not finished for the group number, the value of the Ref Grp will return to the original value and the following warning message will be displayed:

```
Referenced group does not exist
```

39.3 SEARCH PATTERN

The search pattern and pattern type determine the offset direction of the Touch Sensing function. Select the search pattern and pattern type after confirming the direction in which the workpiece may be shifted.
To detect a horizontal shift of the workpiece, one search motion is required in each search direction. In addition, to detect a rotational workpiece shift, two search motions are required in each search direction.

Search Pattern

The following search patterns are available. Select a search pattern based on the shape of the workpiece.

- Simple search
- Fillet/lap search
- V-Groove search
- Outside/inside diameter search

The selected search pattern determines the position data type to be stored in the position register. The position data type is either an absolute position or an offset value.
If you select a fillet/lap search, the search type should be determined according to the possible shift directions of the workpiece.

Simple Search

For a simple search, a two-dimensional search is executed to find the actual location of one position on a workpiece. A simple search stores the found absolute position (x, y, z, w, p, r) into a position register PR[ ]. Once completed, the robot is programmed to move to the position stored in that position register.

A simple search requires that:

- Surfaces being searched are perpendicular to each other.
- Searches must be performed in two different directions.
- The second search motion must be performed with the desired torch angle.

The first search defines the positional information for that search direction only (X, for example). The second search defines the other direction positional information (Z, for example). The starting position of the second search defines the remaining positional information, (Y, W, P, R, for example) that determines the torch angle for welding and, in this case, the Y value.

A simple search is typically used to find the starting point of a weld path that uses the Thru-Arc Seam Tracking (TAST) option or Automatic Voltage Control (AVC) Tracking option.
A two-dimensional search is programmed in the software as the only valid search pattern type when a simple search is used. Changing the search pattern type has no effect.

⚠WARNING
Do not use simple search when you use the Multipass option with Touch Sensing because both simple search and Multipass use position registers. Simple search stores the computed position in a position register. Multipass cannot use position registers to plan paths. Use the 2D fillet search pattern when using Multipass with Touch Sensing.

See Fig. 39.3(a) for an illustration of a simple search routine. Refer to Section 39.5 for example programs using simple search.

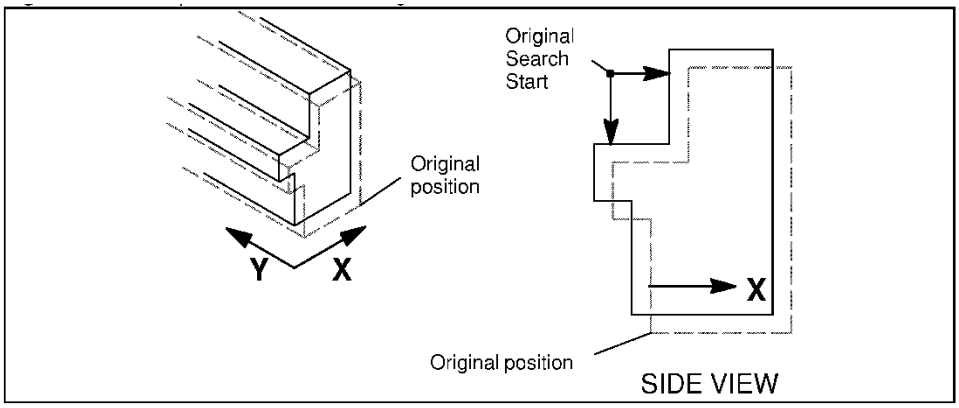


Fig. 39.3 (a) Simple search routine using searches in two directions

Fillet/Lap Search

For a fillet/lap search one, two, or three-dimensional searches are executed to obtain positional offset information. Fillet/lap search stores positional offset information in a position register PR[ ]. This offset can be applied to one or more positions in a programmed path.

1D Shift, 2D Shift, 3D Shift, 1D + Rot, 2D + Rot, 3D + Rot can be used as the search pattern type for fillet/lap search. You can offset in one, two, or three dimensions. In addition, a rotational offset for compensation can be used.

For example, when 1D + Rot is selected as the search pattern type, the fillet search can offset for a rotation about Z by searching two points in the X direction. Note that in this type of search, the first touch point is used as the arc start point. See Fig. 39.3(b).

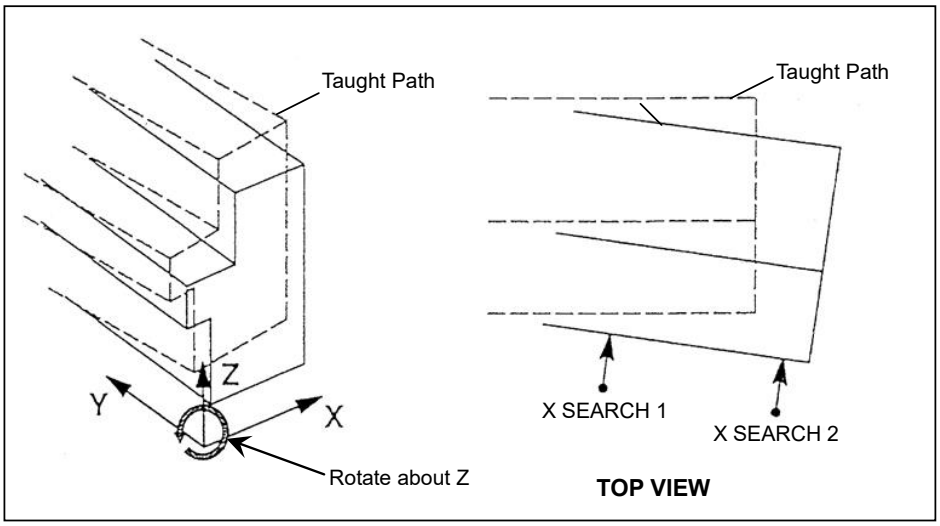


Fig. 39.3 (b) Fillet search in one direction (x) with rotation about Z

When 2D + Rot is selected as the search pattern type, the fillet search can offset for a rotation about Z by searching two points per one direction in the X and Y directions. See Fig. 39.3(c).

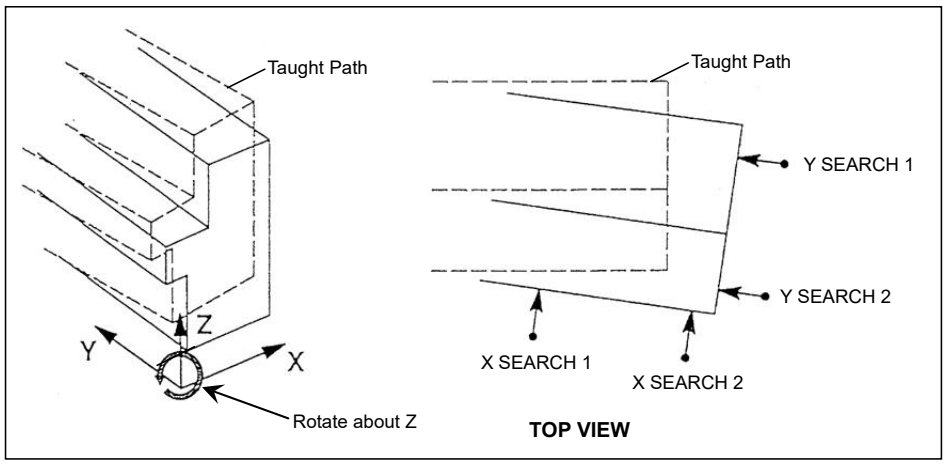


Fig. 39.3 (c) Fillet search in two directions (X and Y) with rotation about Z

When 3D + Rot is selected as the search pattern type, the fillet search can offset for rotations about all directions by searching three points in another direction in addition to the 2D + Rot search motion. See Fig. 39.3(d).

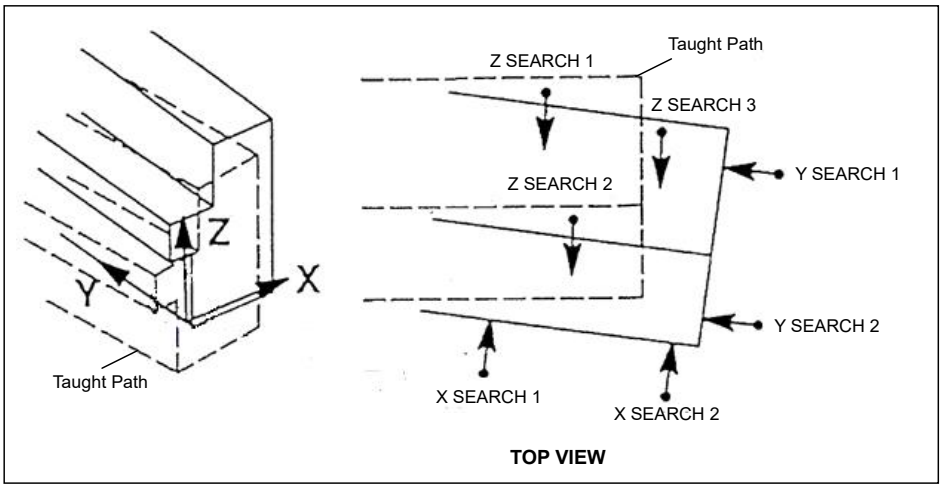


Fig. 39.3 (d) Fillet search in three directions (X, Y, Z) with rotation about all directions

Refer to Fig. 39.5(g)-(j) in Section 39.5 for example programs using fillet/lap search.

V-Groove Search

V-groove search a one-dimensional search executed to obtain positional offset information. A V-groove search stores position offset information in a position register [PR]. This offset can be applied to one or more positions in a programmed path. Only the 1D search pattern type is allowed for V-groove search.
See Fig. 39.3(e) for an illustration of a V-groove search.

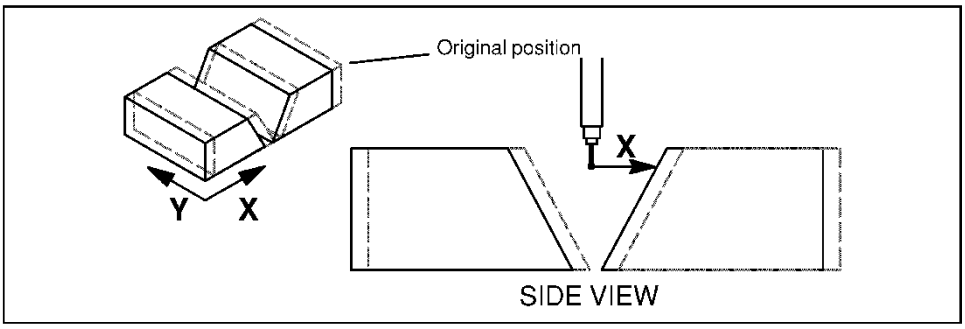


Fig. 39.3 (e) V-groove search

Outside/Inside Diameter Search (OD/ID)

In an outside/inside diameter search (OD/ID search), a two-dimensional search is executed to obtain the positional offset information of the center point of a circular path relative to the original (master) location. An outside/inside diameter search stores positional offset information in a position register [PR]. This offset can be applied to one or more positions in a programmed path.
Only the 2D search pattern type is allowed for an OD/ID search. For more details, refer to Table 39.4.

See Fig. 39.3 (f) for an illustration of an OD/ID search.

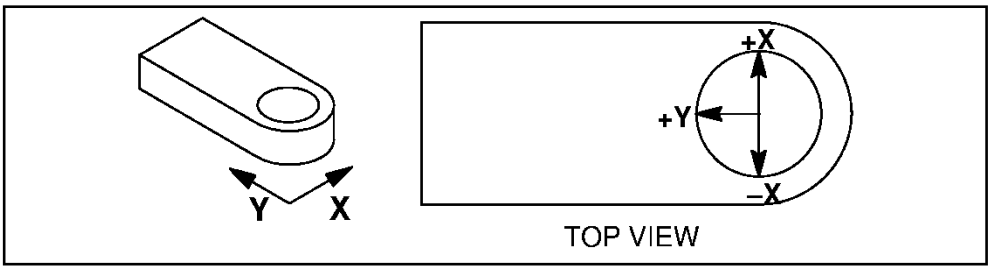


Fig. 39.3 (f) OD/ID search in two directions (X and yY

Table 39.3 shows a matrix of possible search patterns and valid pattern types. Select the combination you want to use on your application and verify that it will provide the proper results.

Table 39.3 Search patterns and valid pattern types

Search Patterns	Pattern Type 1_D	Pattern Type 2_D	Pattern Type 3_D	Pattern Type 1_D and Rotation	Pattern Type 2_D and Rotation	Pattern Type 3_D and Rotation
Simple Search	Not Valid	Requires 2 different search directions. 1 search per direction.	Not Valid	Not Valid	Not Valid	Not Valid
Fillet/Lap	Requires 1 search direction. 1 search.	Requires 2 different search directions, X and Y, X and Z, Y and Z. 1 search per direction.	Requires 3 different search directions, X, Y, and Z. 1 search per direction.	Requires 1 search direction. 2 searches per direction.	Requires 2 different search directions. 2 searches per direction.	Requires 3 different search directions. 3 searches in one direction (usually -Z) 2 searches in each of the remaining directions.
V-Groove	Requires 1 search direction. 1 search	Not Valid	Not Valid	Not Valid	Not Valid	Not Valid

Search Patterns	Pattern Type 1_D	Pattern Type 2_D	Pattern Type 3_D	Pattern Type 1_D and Rotation	Pattern Type 2_D and Rotation	Pattern Type 3_D and Rotation
OD/ID	Not Valid	Requires 3 different searches in 2 different directions. For example, +X, -X, +Y, NOT X, yY Z. 1 search per direction.	Not Valid	Not Valid	Not Valid	Not Valid

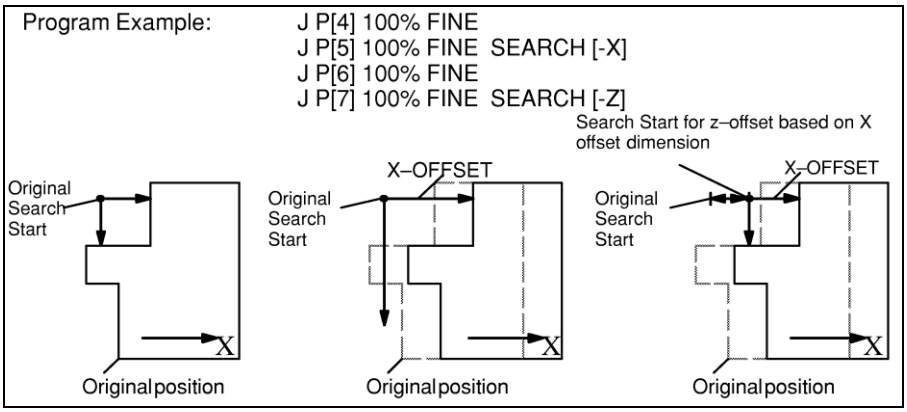
39.4 TOUCH SCHEDULES

A touch schedule is a series of conditions that control how the search motion is completed. a touch schedule includes the touch frame, search pattern, and pattern type.

Thirty-two touch schedules are available. You can access touch schedules from the DATA menu. There are two screens associated with touch schedules: the SCHEDULE screen and the DETAIL screen.
The SCHEDULE screen allows you to view and set limited information for all schedules at once. DETAIL allows you to view and set the complete information for a single schedule.
Table 39.4 lists and describes each DETAIL screen condition. The items on the SCHEDULE screen are also included on the touch sensing DETAIL screen.
Use Procedure 39-5 to define touch schedules.

Table 39.4 Touch schedule screen Items

ITEM	DESCRIPTION
Touch Schedule	This item indicates the number of the displayed schedule. A comment can also be entered here.
Master Flag Default=OFF	This item enables the search routine to be used as a mastering routine for those touch sensing programs that generate positional offset information. If set to ON, when the search routine is executed, the touched positions are recorded in the TP program internally as the reference positions to be used by future searches. This flag must be set to OFF after the master search is completed to generate the position offset information on the objects to be searched. When a search is performed with Master Flag ON, the position offset information in the position register is set to all zero values. The Master Flag condition has no effect on the simple search pattern.
Search Speed Default = 50.0 mm/sec	This item specifies how fast the robot will move when performing a search motion. The accuracy of the search will be improved when this speed is slow. **⚠ CAUTION** A search motion is programmed as a motion option at the end of a motion instruction. The speed at which the robot will move is determined by the search speed, not by what is indicated in the motion instruction. During testing, when dry run is in effect, this search speed is also used. The dry run speed has no effect.
Search Distance Default = 100 mm	This item defines how far the robot can move when it is performing a search. The error "THSR-017 No contact with part." is posted when this distance is reached without making contact with the workpiece.
Touch Frame Default = 1	This item defines the touch frame number to be used in the touch schedule. This determines the X, Y, and Z directions for the search motion. The same touch frame can be used in more than one touch schedule.

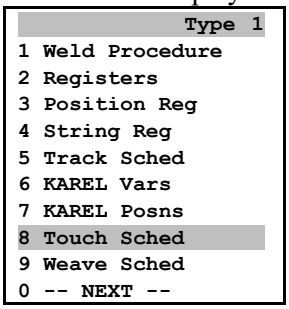
ITEM	DESCRIPTION
Search Pattern Default = SIMPLE	This item defines the type of object to be searched and causes the ArcTool software to compute the found position or positional offset information dependent on the search pattern selected. The computed data is stored in a position register. There are four available search patterns: - Simple - Fillet/Lap - V-Groove - OD/ID For more details, refer to Section 39.3.
Pattern Type Default = 1_D Shift • 1_D Shift • 2_D Shift • 3_D Shift • 1_D + Rot • 2_D + Rot • 3_D + Rot	This item selects the type of offset to be stored in the position register for the fillet/lap search pattern. Six pattern types are available: Stores a one-dimensional offset. Offsets can be in the X, Y, or Z direction. Stores a two-dimensional offset. Offsets can be in two of the X, Y, or Z direction. Stores a three-dimensional offset to a program. Offsets are in all of the X, Y, and Z directions. Stores a one-dimensional offset with rotation about the axis of which the search is not performed. Stores a two-dimensional offset with rotation about the axis of which no searches are performed. For example, if the workpiece is being searched for an offset in both the X and Y directions, a 2_D Shift & Rotate search can offset for a rotation about the Z axis. Stores a three-dimensional offset with rotation about all of the axes. NOTE: Simple, OD/ID, and V-groove search patterns are pre-defined. Changing the pattern type for these searches has no effect.
Incremental Search Default = ON	This item offsets the starting position of the second and subsequent searches in a search routine by the amount of offset found by the first search motion. If set to OFF, the robot returns to the original starting position. The following illustration shows how the incremental search affects the search routine. Incremental search requires a different position number for each search. Program Example: J P[4] 100% FINE J P[5] 100% FINE SEARCH [-X] J P[6] 100% FINE J P[7] 100% FINE SEARCH [-Z]  Without incremental search, the robot found the X-offset but cannot find the Z-offset. **NOTE** Simple search does not support incremental search.
Auto Return Default = ON	This item moves the robot back to the search start position when contact is made with the object. If set to OFF, the robot stops at the contact point and moves straight to the next position.
Return Speed Default = 100 mm/sec	This item specifies the speed at which the robot will return to the search start position upon making contact with the part.
Return Distance Default = 2000 mm	When Auto Return is set to ON, Return Distance specifies the distance the robot will return automatically. If the return distance passes the initial search start position, the robot will return to the initial start position.

ITEM	DESCRIPTION
Reference Group	This item specifies how the offset is recorded. If the Search Pattern is not simple search, you should specify the same group number as Ref Grp on the selected Touch Frame. If the Search Pattern is simple search, you should specify the same group number as Robot Group item on this schedule. **NOTE** When search pattern is not simple search, if Reference Group is not equal to Ref Grp on Touch Frame, an error message "THSR-039 Reference grp mismatch," will be posted. When search pattern is simple search, if Reference Group is not the same as the specified number of Robot Group on this touch schedule, an error message, "THSR-037 Illegal motion ref. grp," will be posted.
Return Term Type Default = FINE	This item specifies the termination type the robot will use to return to the search start position. Four Return Term Types are available: - FINE - CNT20 - CNT40 - CNT100
Contact Record PR Default = 32	This item is the search output position register used as a temporary buffer to hold the last search contact position. The purpose of this temporary position register buffer is to provide the ability to look at the positional data of an individual search, or to extract data from the buffer in a program. By default, this register is position register 32. **⚠ CAUTION** The data in the position register is overwritten at each search motion so the same position register should not be used to store the final positional data from the search motion. Also, the contents of this temporary buffer is a real position, not an offset. Do not program motion instructions to use this position register data as an offset.
Error on Failure Default = ON	This item posts the error “THSR - 017 No contact with part, if the search move exceeds the distance set in Search Distance. When this item is OFF, the program execution continues with the next instruction even if the Search Distance is exceeded. Programming Hint: If this is set to OFF, the next instruction in the program looks at the contents of the Error Register and branches accordingly.
Error Register Number Default = 32	When Error On Failure is set to OFF, this register is set to 1 when the search distance is exceeded. A successful search sets this register to 0.
Robot Group Default = [1,*,*,*,*,*,*,*]	This item specifies the robot group that uses the touch sensing schedule.

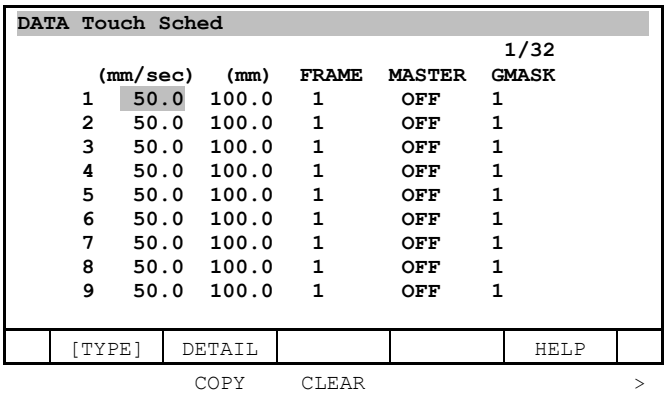
Procedure 39-5 Setting Up Touch Schedules

Step

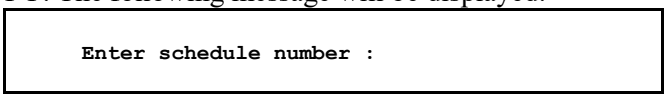
1 Press DATA.
2 Press F1, [TYPE]. The following menu will be displayed.



3 Select Touch Sched. The following menu will be displayed.

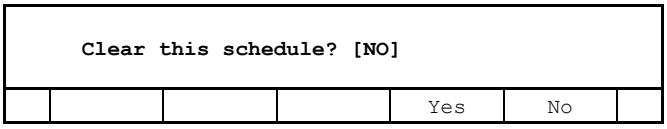


4 To copy schedule information from one schedule to another:
   a Press NEXT, >.
   b Move the cursor to the schedule you want to copy.
   c Press F2, COPY. The following message will be displayed.

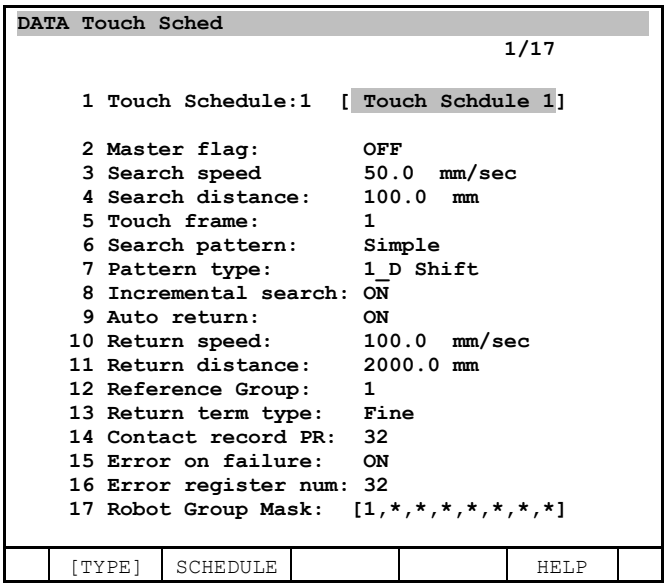


   d Enter the schedule number from which you want to copy the data.
   e Press ENTER. The data will be copied, but the comment will not be copied.

5 To clear the information you have entered for a schedule:
   a Move the cursor to the schedule.
   b Press NEXT, >.
   c Press F3, CLEAR. The following message will be displayed. If F4, Yes is pressed, the data will be cleared, but the comment will remain.



6 Move the cursor to the desired schedule number.
7 To display more information about the schedule, press F2, DETAIL. See the following screen for an example.



8 Set each schedule item as desired.
9 To add a comment:
  a Move the cursor to the to the comment line and press ENTER.
  b Select a method of naming the comment.
  c Press the appropriate function keys to add the comment.
  d When you are finished, press ENTER.
10 To set the Search pattern, Pattern type, and Return Term Type:
  a Move the cursor to each item and press F4, SELECT.
  b Select the item and press ENTER.

NOTE
A touch schedule is only a means to perform a search. Master position information is never saved in the touch schedule – it is saved in the TP program. For this reason, you can use the same touch schedule in multiple programs and multiple locations.

39.5 TOUCH SENSING PROGRAMMING

A Touch Sensing routine consists of search instructions to locate an object and offset instructions to displace programmed positions.

⚠WARNING
Recorded positions and position registers are affected by UFRAME, and UFRAME has an affect during playback. If you change UFRAME, any recorded positions and position registers will also change.

NOTE
Any changes to the tool frame affect the search start position.

Touch Sensing Instructions

The following touch sensing instructions are provided:
- Search Start [ ] PR [ ] (Search Start instruction)
- Search End (Search End Instruction)
- Touch Offset PR[ ] (Touch Offset Start instruction)
- Touch Offset End (Touch Offset End instruction)

In addition, the following instruction is used a motion modifier between Search Start – Search End instructions.
- Search [ ] (Search instruction)

⚠CAUTION
Do not use Replace in the [EDCMD] menu on the program edit screen in a program that has Touch Sensing instructions. This will cause a memory write error. If you want to replace motion instructions, insert Touch Sensing instructions after deleting the motion instruction you want to replace.

Search Start/End Instructions

The Search Start instruction commands the start of search motion. Specify a touch schedule number for this search motion, and also specify a position register number for storing Touch Sensing positional information.

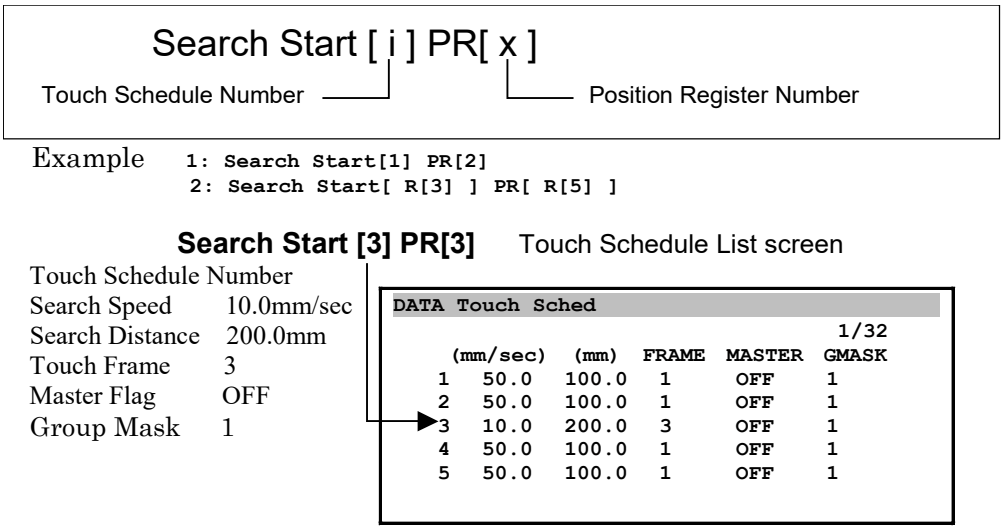


Fig. 39.5(a) Search start instruction

One Search End instruction is required for one Search Start instruction. If you want to execute a Search Start instruction again for A new search motion, you must execute A Search End instruction to finish the previous search motion.

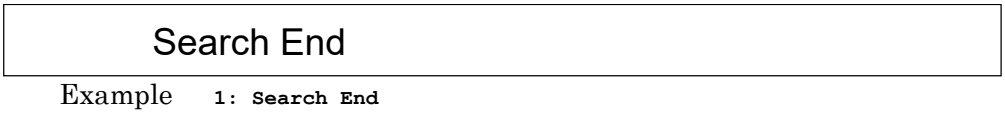


Fig. 39.5(b) Search end instruction

Search instructions must be taught between Search Start and Search End instructions.

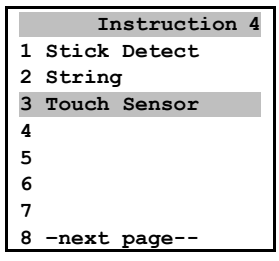
Procedure 39-6 Teaching Search Start and End Instructions

Condition

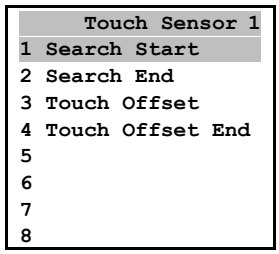
- The program edit screen is displayed.
- The teach pendant Enabling Device (Deadman switch) is pressed and the ON/OFF switch is set to ON.
-

Step

1 Press NEXT, > and press F1, [INST]. The following menu will be displayed.



2 Select Touch Sensor. A list of touch sensing instructions will be displayed. If you want to teach a Search Start instruction, select Search Start.



3 Enter a touch schedule number in the [] just after Search Start. Also, enter a position register number in the [] of PR.
4 To teach a Search End instruction, select Search End in Step 2.

Search Instruction

The Touch Sensing motion modifier, Search[ ], directs the motion of the robot (in a positive or negative X,Y, or Z direction) to search for the object. The X, Y, and Z vectors are defined by the touch frame assigned in the touch schedule.

The taught position in a motion instruction with a search motion modifier has no meaning and it not used, so the move to the search start position must be taught in a separate motion instruction. In addition, the motion speed in a motion instruction with a search motion modifier has no meaning and is not used. Instead, the search speed specified in the touch schedule (refer to Table 39.5) is used as the motion speed. When the touch sensor part touches the workpiece during search motion, the current position is recorded in the Contact Record PR.

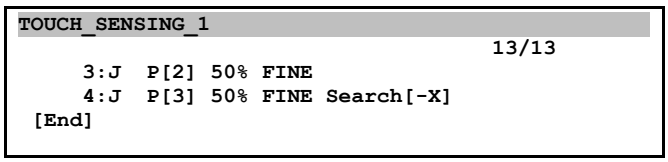


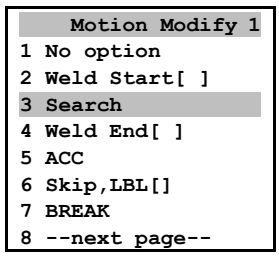
Fig. 39.5(c) Program example using a search motion modifier

Use Procedure 39-7 to teach a Search instruction.

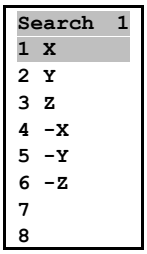
Procedure 39-7 Teaching a Search Instruction

Step

1 Jog the robot to the search start position and teach a motion instruction.
2 Teach another motion instruction again at the same position.
3 Move the cursor to the end of the motion instruction taught in Step 2, and press F4, [CHOICE] to display the motion modifier choices.



4 Select Search.
5 Select the direction of the search to be performed and press ENTER.



⚠CAUTION
Teach the Search instruction at the last motion modifier position in the motion instruction. Also, the motion instruction for moving to the search start position (taught just before the motion instruction that has the Search motion modifier) must use FINE termination type. If CNT termination type is used for these instructions that move to the search start position, correct offset information cannot be calculated.

Touch Offset Start and Touch Offset End Instructions

Touch Offset Start/End instructions are used for shifting program positions using a position offset which obtained by the search motion. The amount of offset in the Touch Offset instruction is defined in the position register specified in the Touch Offset Start instruction. Normally, specify the position register number that recorded position offset in the Search Start/End instructions.

Touch Offset PR[x]
Position Register Number

Example
```
1: Touch Offset PR[3]
2: Touch Offset PR[ R[5] ]
```

Fig. 39.5(d) Touch Offset start instruction

Touch Offset End

Example
```
1: Touch Offset End
```

Fig. 39.5(e) Touch Offset End instruction

Use Procedure 39-8 to teach Touch Offset Start and End instructions.

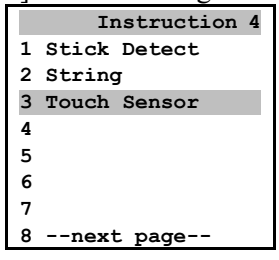
Procedure 39-8 Teaching Touch Offset Start and End Instructions

Condition

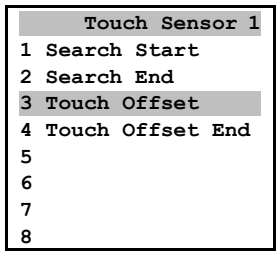
- The program edit screen is displayed.
- The teach pendant Enabling Device (Deadman switch) is pressed and the ON/OFF switch is set to ON.

Step

1 Press NEXT, > and press F1, [INST]. The following menu will be displayed.



2 Select Touch Sensor. A list of touch sensing instructions will be displayed. If you want to teach a Touch Offset Start instruction, select Touch Offset.



3 Enter a position register number in the [] of the PR.
4 To teach a Touch Offset End instruction, select Touch Offset End in Step 2.

> **⚠ CAUTION**
> Do not use a CALL instruction between Touch Offset Start and Touch Offset End instructions. The Touch Offset is disabled in the motion instructions after the CALL instruction is executed.

Simple Search Program Example

After a simple search routine, the Touch Sensing software will calculate an absolute (real) position (X,Y,Z,W,P,R) and put the data in the position register specified in the Search Start instruction.

Since this is a real position, the robot will be commanded to move to the position in the position register.

Example: J PR [4] 100% FINE Weld Start [1,1]

The above is an example of when position register 4 is the position register specified in the simple search routine.

Simple search differs from other search patterns in the following ways:

- Master Flag is always OFF.
- The position values stored in the position register is the absolute position data.
- Touch Offset Start/End instructions are never required.

```
SIMPLE_SEARCH
                                    13/13
    1:J  P[1] 50% FINE
    2:  Search Start[4] PR[4]
    3:J  P[2] 50% FINE
    4:J  P[3] 50% FINE Search[Y]
    5:J  P[4] 50% FINE Search[-Z]
    6:  Search End
    7:J  PR[4] 50% FINE
     :  Weld Start[1,1]
```

Teach Home Position
Start Search Motion. Use Touch Sch 4, PR[4]
Teach a search start position
Search to Y direction
Search to -Z direction
Finish Search Motion
Move the robot to PR[4] position,
And start Arc Welding

Fig. 39.5(f) Program example of simple search (Refer to Fig. 39.4(a))

1D Search Program Example

In Touch Sensing programs other than simple search, the offset information captured by the search motion is used in the Touch Offset instruction to offset the taught path.
In 1D Search, only one point in one direction that may shift the workpiece will shift is searched, and the Touch Offset instruction is used to compensate the taught path for the shift. Refer to the following program example.

```
1D_SEARCH
                                    12/13
    1:J  P[1] 50% FINE
    2:  Search Start[1] PR[1]
    3:J  P[2] 50% FINE
    4:J  P[3] 50% FINE Search[Y]
    6:  Search End
    7:J  P[4] 50% FINE
    9:  Touch Offset PR[1]
   10:J  P[5] 100mm/sec FINE
     :  Weld Start[1,1]
   11:L  P[6] 80cm/min FINE
     :  Weld End[1,2]
   12:  Touch Offset End
```

Teach Home Position
Start Search Motion. Use Touch Sch 1, PR[1]
Teach a search start position
Search in the Y direction
Finish Search Motion
(If needed) Teach an intermediate point
Following points will be offset by PR[1]
P[5] is offset by PR[1]

P[6] is offset by PR[1]

Finish offset of positions

Fig. 39.5(g) Program example of 1D search (Fillet/Lap search and V-Groove search)

2D Search Program Example

In 2D search, one point is searched in each of two directions. Then, the taught path is compensated for the shift using a Touch Offset instruction. Refer to the following program example.

```
2D_SEARCH
                                    12/13
    1:J  P[1] 50% FINE
    2:  Search Start[1] PR[2]
    3:J  P[2] 50% FINE
    4:J  P[3] 50% FINE Search[Y]
    5:J  P[4] 50% FINE
    6:J  P[5] 50% FINE Search[X]
    7:  Search End
    8:J  P[6] 50% FINE
    9:  Touch Offset PR[2]
   10:L  P[7] 100mm/sec FINE
     :  Weld Start[1,1]
   11:L  P[8] 80cm/min FINE
     :  Weld End[1,2]
   12:  Touch Offset End
```

Teach Home Position
Start Search Motion. Use Touch Sch 1, PR[2]
Teach a search start position
Search in the Y direction
Teach another search start position
Search in the X direction
Finish Search Motion
(If needed) Teach an intermediate point
Following points will be offset by PR[2]
P[7] is offset by PR[2]

P[8] is offset by PR[2]

Finish offset of positions

Fig. 39.5(h) Program example of 2D search (Fillet/Lap search)

2D + Rotation Search Program Example

In 2D + rotation search, two points are searched in each of the two directions. The taught path is then compensated using a Touch Offset instruction. Refer to the following program example.

```
2D+ROT_SEARCH
                                    25/26
    1:J  P[1] 50% FINE
    2:  Search Start[3] PR[3]
    3:J  P[2] 50% FINE
    4:J  P[3] 50% FINE Search[X]
    5:J  P[4] 50% FINE
    6:J  P[5] 50% FINE Search[X]
    7:J  P[6] 50% CNT100
    8:J  P[7] 50% FINE
    9:J  P[8] 50% FINE Search[Y]
   10:J  P[9] 50% FINE
   11:J  P[10] 50% FINE Search[Y]
   12:  Search End
   13:J  P[1] 50% FINE
   14:  Touch Offset PR[3]
   15:L  P[11] 100mm/sec FINE
     :  Weld Start[1,1]
   16:L  P[12] 80cm/min CNT100
   17:L  P[13] 80cm/min FINE
     :  Weld End[1,2]
   18:  Touch Offset End
```

Teach Home Position
Start Search Motion. Use Touch Sch 3, PR[3]
Teach a search start position
Search to X direction
Teach another search start position
Search in the X direction
Teach an intermediate position
Teach another search start position
Search in the Y direction
Teach another search start position
Search in the Y direction
Finish Search Motion
Move to Home Position
Following points are offset by PR[3]
P[11] is offset by PR[3]

P[12] is offset by PR[3]
P[13] is offset by PR[3]

Finish offset of positions

Fig. 39.5(i) Program example of 2D + rotation search (Fillet/Lap search)

3D + Rotation Search Program Example

A 3D + rotation search is a 2-D + rotation search plus a three-point search in another direction. The amount of rotation in all three directions can also be obtained as offset data.

```
3D+ROT_SEARCH
                                        25/26
     1:J  P[1] 50% FINE
     2:  Search Start[3] PR[3]
     3:J  P[2] 50% FINE
     4:J  P[3] 50% FINE Search[X]
     5:J  P[4] 50% FINE
     6:J  P[5] 50% FINE Search[X]
     7:J  P[6] 50% CNT100
     8:J  P[7] 50% FINE
     9:J  P[8] 50% FINE Search[Y]
    10:J  P[9] 50% FINE
    11:J  P[10] 50% FINE Search[Y]
    12:J  P[11] 50% CNT100
    13:J  P[12] 50% FINE
    14:J  P[13] 50% FINE Search[Z]
    15:J  P[14] 50% FINE
    16:J  P[15] 50% FINE Search[Z]
    17:J  P[16] 50% FINE
    18:J  P[17] 50% FINE Search[Z]
    19:  Search End
    20:J  P[1] 50% FINE
    21:  Touch Offset PR[3]
    22:L  P[18] 100mm/sec FINE
      :  Weld Start[1,1]
    23:L  P[19] 80cm/min CNT100
    24:L  P[20] 80cm/min FINE
      :  Weld End[1,2]
    25:  Touch Offset End
```

Teach Home Position
Start Search Motion. Use Touch Sch 3, PR[3]
Teach a search start position
Search in the X direction
Teach another search start position
Search in the X direction
Teach an intermediate position
Teach another search start position
Search in the Y direction
Teach another search start position
Search in the Y direction
Teach an intermediate position
Teach another search start poison
Search in the Z direction
Teach another search start poison
Search in the Z direction
Teach another search start poison
Search in the Z direction
Finish Search Motion
Move to Home Position
Following points are offset by PR[3]
P[18] is offset by PR[3]

P[19] is offset by PR[3]
P[20] is offset by PR[3]

Finish offset of positions

Fig. 39.5(j) Program example of 3D + rotation search (Fillet/Lap search)

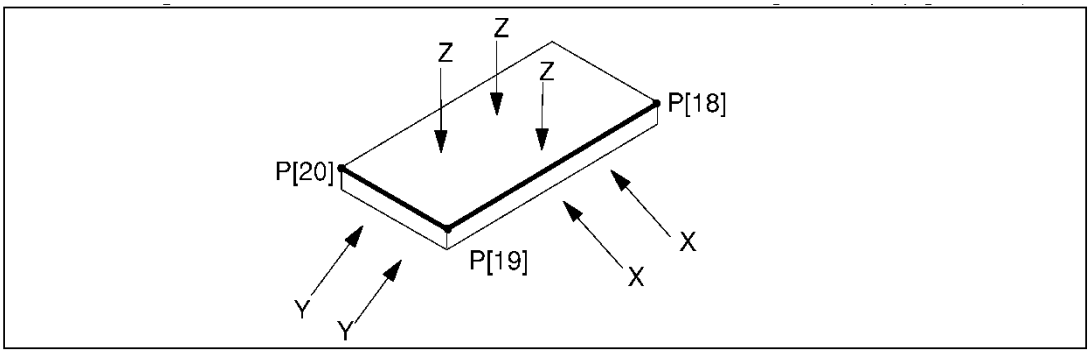


Fig. 39.5(k) Outline of 3D + rotation search

⚠CAUTION

For Fillet/Lap, V-Groove, OD/ID search pattern programs, perform search motion (Search Start – Search End section) first with Master Flag ON to store the master position information in the TP program. This is necessary to establish master positions for all search motion. After you have done this, execute the Touch Sensing program with Master Flag OFF.
If the program has never been executed with Master Flag ON, the Touch Sensing program will not run properly (refer to Section 39.6).

39.6 EXECUTION OF TOUCH SENSING PROGRAM

Touch Sensing Mastering

In search pattens other than simple search, Touch Sensing calculates offsets relative to the position obtained by mastering. An offset is computed by comparing the position of the current workpiece (offset position) with the original workpiece position (mastering position).
Therefore, before executing a Touch Sensing program, you must perform mastering to obtain the original workpiece position. Use Procedure 39-9 to perform mastering.

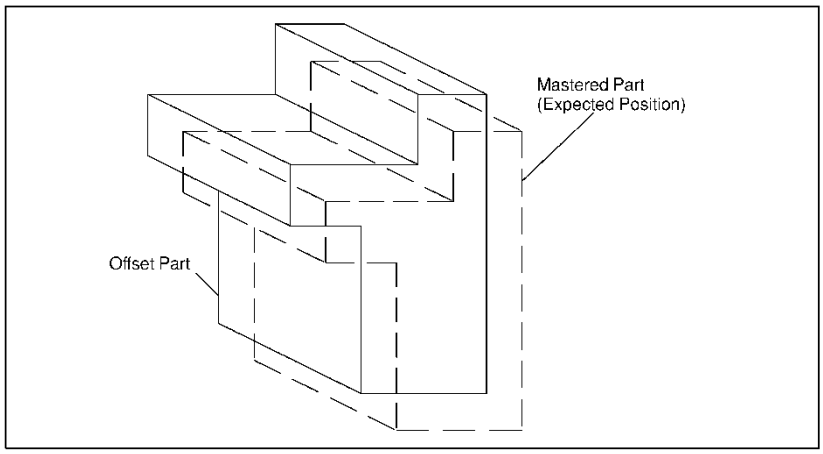


Fig. 39.6(a) Part in mastered position and offset position

> **NOTE**
> A simple search does not require mastering since it produces an absolute position stored in a position register.

Procedure 39-9 Touch Sensing Mastering

Condition

- The workpiece is placed in a position without any shift (placed on the taught path).
- The Touch Sensing Program has been created.
- A search pattern other than simple search pattern is specified on the Touch Schedule.

Step

1 Display Touch Schedule Detail screen for the touch schedule number used by referring to Procedure 39-5.
2 Move the cursor to Master Flag and press F4, ON. Master Flag turns ON.
3 Display the program edit screen for the mastering program.
4 Execute the program from the Search Start to the Search End instruction (single step execution is permitted). The mastering position is stored in the TP program and the data in the position register specified in the Search Start instruction all change to zero.
5 Display the Touch Schedule Detail screen, move the cursor on Master Flag, and press F5, OFF. Master Flag turns OFF.

> **NOTE**
> Incremental search is disabled while the Master Flag is turned ON. If incremental search is not performed, check the Master Flag.

Executing a Touch Sensing Program after Mastering

After mastering has been completed, the taught path is corrected each time by executing search motion and touch offset. Fig 39.6(b), (c) provide a detailed description of the process when executing a Touch Sensing program after mastering This is a program example of performing a 2D fillet search.

```
2D_FILLET_SEARCH                                    12/13
     1:J  P[1] 50% FINE
     2:  Search Start[3] PR[1]
     3:J  P[2] 50% FINE
     4:J  P[3] 50% FINE Search[X]
     5:J  P[2] 50% FINE
     6:J  P[4] 50% FINE Search[-Z]
     7:  Search End
     8:
     9:  Touch Offset PR[1]
    10:J  P[5] 50% FINE
    11:L  P[6] 50cm/min CNT100
    12:L  P[7] 50cm/min FINE
    13:  Touch Offset End
```

Fig. 39.6(b) Program example of 2D fillet search

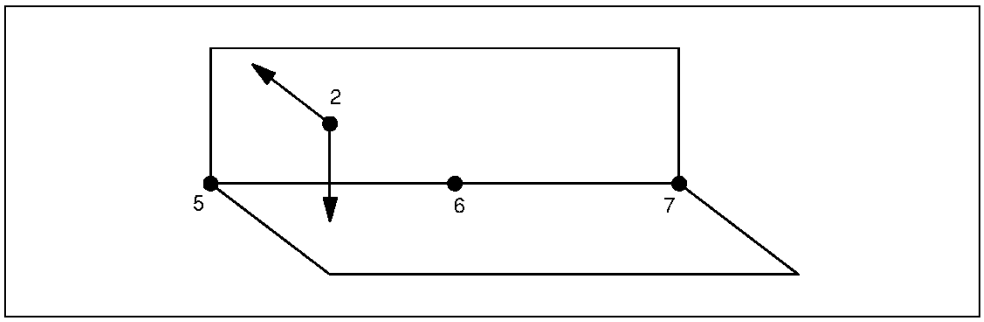


Fig. 39.6(c) Part with one Touch Start position, 2, and three points along a path, 5, 6, 7

- After mastering is completed, the mastering position and current position (offset position) obtained by the search motion are compared and offset data is calculated when search motion (between the Search Start – End instructions) is performed with Master Flag OFF. The offset data obtained by this search motion is the relative amount between the mastering position and offset position.

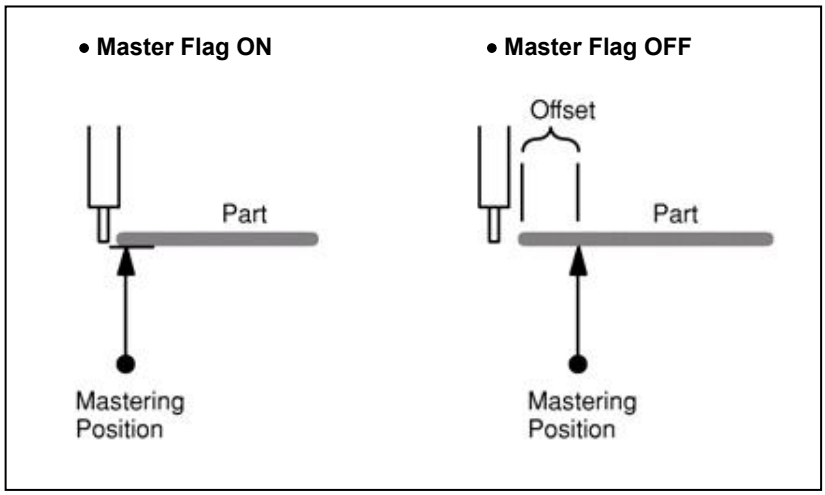


Fig. 39.6(d) Offset value

- After the Search End instruction is executed, the offset data is set in the position register specified in the Search Start instruction.

- After that, the offset amount is applied to all motion instructions between the Touch Offset Start – End instructions (position 5, 6 and 7 in the program example), resulting in compensated path motion to match the workpiece at the offset position.

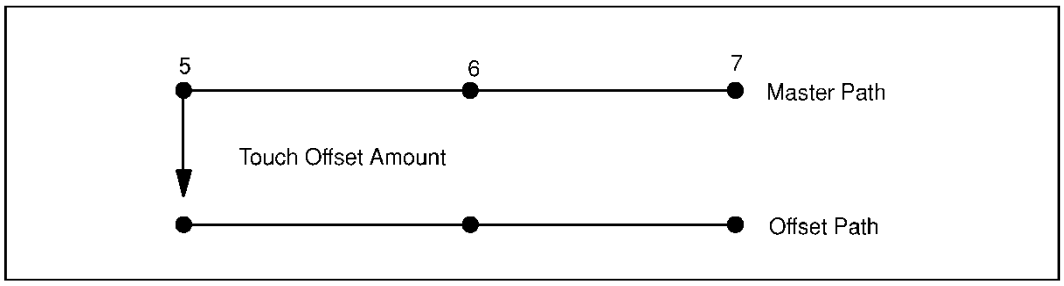


Fig. 39.6(e) Illustration of the path when an offset is applied

Touch Sensing Program Execution Details

- If BWD execution is performed for a Search Start instruction, search motion becomes temporarily disabled. To resume the search motion, restart the program again from the line just before the Search Start instruction using FWD execution.
- Touch offset finishes when a Touch Offset End instruction is executed, or the program is aborted. However, touch offset continues when the program is paused and then resumed.
- If BWD execution is performed for a Touch Offset Start instruction, touch offset becomes temporarily disabled. To resume touch offset, restart the program again from the line just before the Touch Offset Start instruction using FWD execution
- After the FWD execution of a Touch Offset Start instruction, if the cursor is moved to a line other than a line between the Touch Offset Start and End instructions and the program is resumed, the offset specified in the Touch Offset Start instruction is applied (that is, the offset is applied).

Remastering

The touchup procedure described in Section 39.7 should work for most instances where the search start positions do not need to be moved or if the workpieces do not change drastically.
Remastering is required if the search start positions must be taught again. If the path must be changed significantly, it is recommended that you remaster to ensure a correct path.

The procedure for remastering is no different than normal mastering (refer to Procedure 39.9). After finishing remastering, turn Master Flag OFF.

39.7 TOUCHING UP OF TOUCH SENSING PROGRAM

Mastering position data is stored internally in the Touch Sensing TP program. For this reason, be careful when modifying this program. This section explains how to touch up positions in a Touch Sensing program, as shown in Fig. 39.7 (a) and (b).

Touching Up Path Positions

Occasionally, the workpiece or its placement on a fixture will change, requiring adjustment of the path. The entire process of remastering is not needed to accommodate these changes.
When touching up motion instructions placed between Touch Offset instructions is required, you must consider the current offset data of the workpiece while teaching the new master path. Therefore, touching up must be performed after the completion of accurate search motion during touch offset (offset is compensated). In this situation, if touchup is performed by jogging the robot, only the difference information from the original offset position is reflected in the position data in the program, and a new, correct master path is generated (refer to Fig. 39.7(a), (b)).

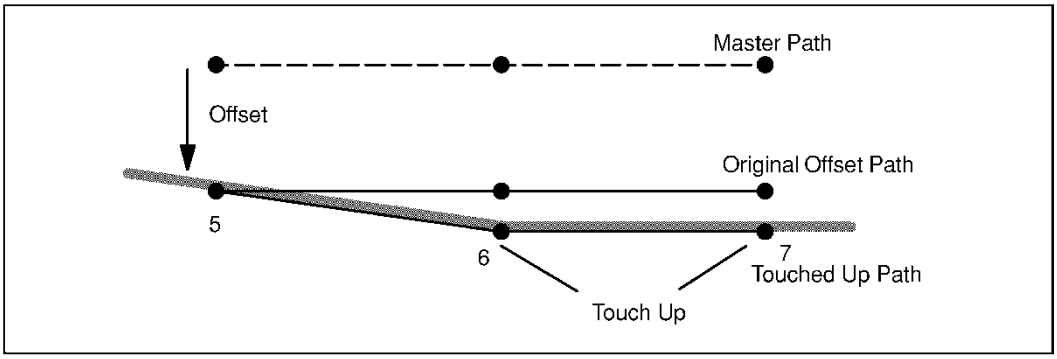


Fig. 39.7 (a) Touchup of position 6 and 7

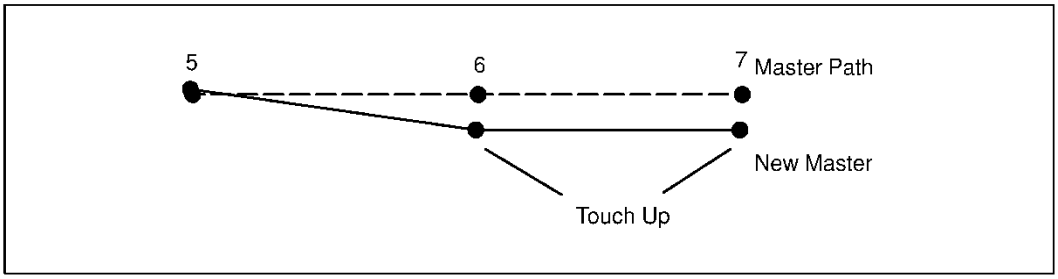


Fig. 39.7 (b) New master path after touching up position 6 and 7

Use Procedure 39-10 to touch up in a Touch Sensing program. Fig. 39.7(c) shows the example that requires Procedure 39-10 during touchup.

```
2D_FILLET_SEARCH
                                        9/14
     9:  Touch Offset PR[1]
    10:J  P[5] 50% FINE
    11:L  P[6] 50cm/min CNT100
    12:L  P[7] 50cm/min CNT100
    13:  Touch Offset End
```

When you touch up these positions, Procedure 39-10 is required.

Fig. 39.7(c) Example of touch up during touch offset

Procedure 39-10 Touching up a Touch Sensing Program

Condition

- The Program edit screen is displayed.
- The teach pendant Enabling Device (Deadman switch) is pressed and the ON/OFF switch is set to ON.
-

Step

1 Execute search motion (between Search Start – End instructions) and store the offset data in a position register.
2 Execute a Touch Offset Start instruction.
3 Run the program in single step mode to the motion instruction you want to touch up.
4 Jog the robot to the position you want to touch up. Then, move the cursor to the position number of the current line and press SHIFT and F5, TOUCHUP. Touching up is done.
5 Touch up all required positions between the Touch Offset Start – End instructions using Step 4.

> **⚠ CAUTION**
> Do not touch up after BWD execution during single step execution between the Touch Offset Start – End instructions. Otherwise, the offset data will be incorrect.

Incorrect Touchup

A common error is to alter the path without the correct offset being applied. Touchup must be done after executing the search and while the touch offset is applied.

An example of an incorrect touchup is as follows:

- Execute the program without executing search motion.
- Touch up the positions between the Touch Offset Start – End instructions without executing a Touch Offset Start instruction.
- Perform touchup with a position register which has incorrect offset data.

If incorrect touchup is performed, the new master position become incorrect. Therefore, you cannot obtain the correct compensated path even if search motion and touch offset have been done properly.

Adding New Positions

You can add new positions using the same touchup Procedure 39-10.

- The search must be completed, and an accurate offset must be generated.
- This operation should be done during touch offset.
- A new position should be added to the offset path.

Fig. 39.7(d) illustrates adding a point to a path.

NOTE

If the program is ABORTED while adding new positions, the offset is cancelled. New positions will be taught as actual locations rather than positions with an offset applied. The results would be similar to Incorrect Touchup and correct path compensation cannot be obtained.

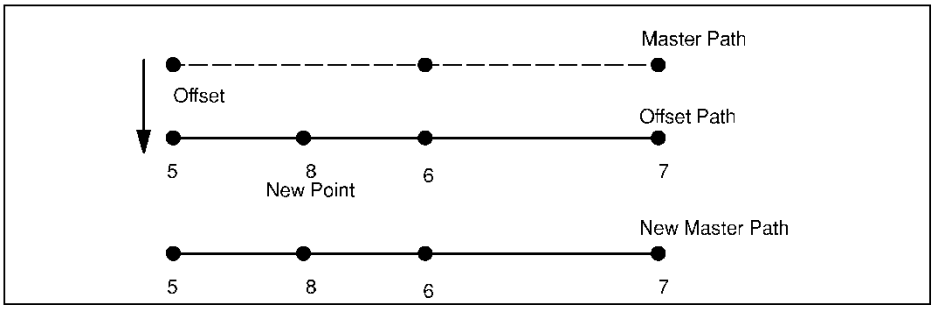


Fig. 39.7(d) Adding a new point while executing touch offset

Touching Up a Search Start Position

Touching up a search start position is different than touching up a touch offset path position. If the search start position is touched up, remastering must be executed. However, if the search start position is moved along the axis of the search direction, remastering is not required.

Example that does not require remastering

If the search start position is too close to the workpiece due to poor programming, changes in the workpiece, or a change in the workpiece location, then the search start position only needs to be moved back along the search direction. This can be accomplished with no effect on the path positions and remastering will not be required.

The example shown in Fig. 39.7(e) shows a workpiece and search start position.

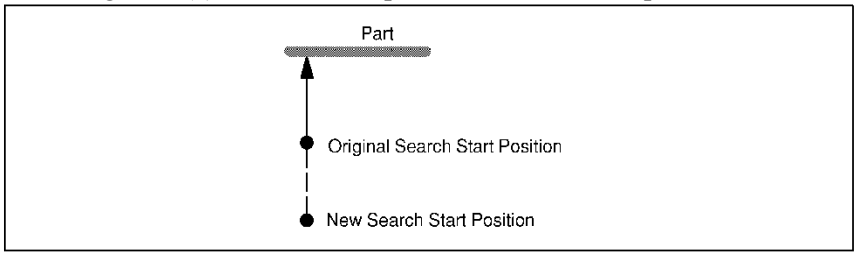


Fig. 39.7(e) Moving a search start position along the search direction

Example that requires remastering
The example shown in Fig. 39.7(f) shows a search start position moved to a position off the axis of the search direction. If the search position is moved off the axis of the search direction, remastering is required. Refer to Section 39.6 for information about remastering.

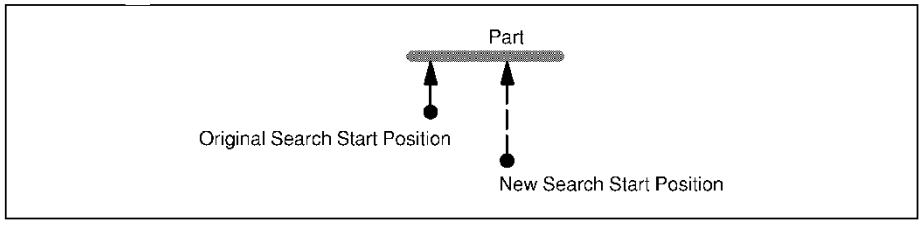


Fig. 39.7(f) Search start position is separated from the original search direction axis

39.8 MULTIPLE SEARCHES

Complex programs can have multiple searches generating several offsets.

The program example shown in Fig. 39.8(a) shows two searches that can be performed for complex shapes. When performing a search for such as complex workpiece, search motion may not be able to be completed by one touch frame. Therefore, multiple searches are required.
- The first search stores the offset data in position register 1 with positions 10, 11, 12 using the offset.
- The second search stores offset data in position register 2 with positions 13, 14, 15 using the offset.

```
MULTIPLE_SEARCHES
                                                25/26
     1:J  P[1] 50% FINE
     2:  Search Start[3] PR[1]
     3:J  P[2] 50% FINE
     4:J  P[3] 50% FINE Search[X]
     5:J  P[4] 50% FINE Search[-Z]
     6:  Search End
     7:J  P[5] 50% FINE
     8:  Search Start[4] PR[2]
     9:J  P[6] 50% FINE
    10:J  P[7] 50% FINE Search[-X]
    11:J  P[8] 50% FINE
    12:J  P[9] 50% FINE Search[-Z]
    13:  Search End
    14:
    15:  Touch Offset PR[1]
    16:J  P[10] 50% FINE
    17:L  P[11] 50cm/min CNT100
    18:L  P[12] 50cm/min CNT100
    19:  Touch Offset End
    20:
    21:  Touch Offset PR[2]
    22:J  P[13] 50% FINE
    23:L  P[14] 50cm/min CNT100
    24:L  P[15] 50cm/min CNT100
    25:  Touch Offset End
```

Fig. 39.8(a) Program example of multiple searches

If a position of the taught path is to be touched up, the corresponding search must be performed.

- The first search must be executed to obtain an accurate offset.
- The offset is applied and positions 10, 11, and 12 can be touched up as normal.
- The master position will be correctly updated.

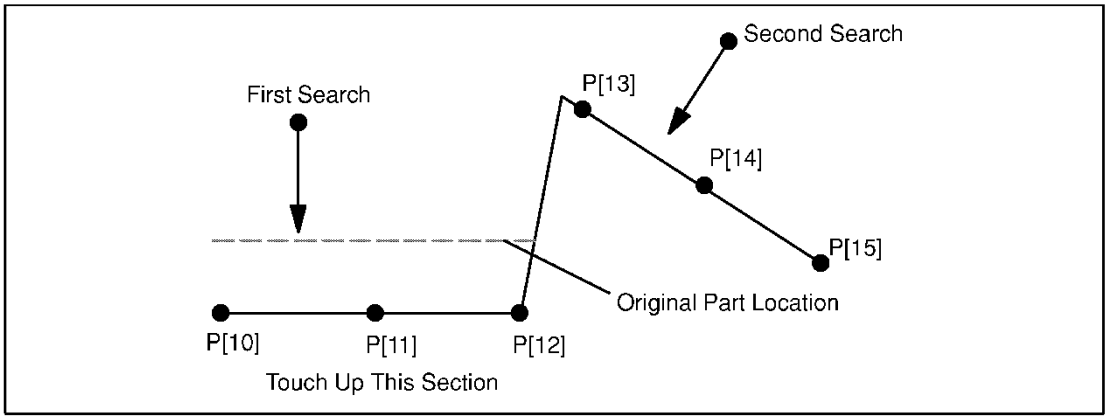


Fig. 39.8(b) Touchup on multiple searches

39.9 COORDINATED TOUCH SENSING

Touch offset using Touch Sensing can also apply to a motion instruction that has a coordinated motion motion modifier. To achieve this, you must set the touch frame and touch schedule for Coordinated Touch Sensing. The setup and programming differ depending on the search pattern.

Coordinated Touch Sensing for search patterns other than Simple Search

Fig. 39.9(a) shows the search motion of leader group (Group 2: Positioner) and follower group (Group 1: Robot) for 2D+rotation with coordinated motion. Search motion is done based on the coordinated frame of leader group. Touch offset motion after search motion is shown in Fig. 39.9(b).

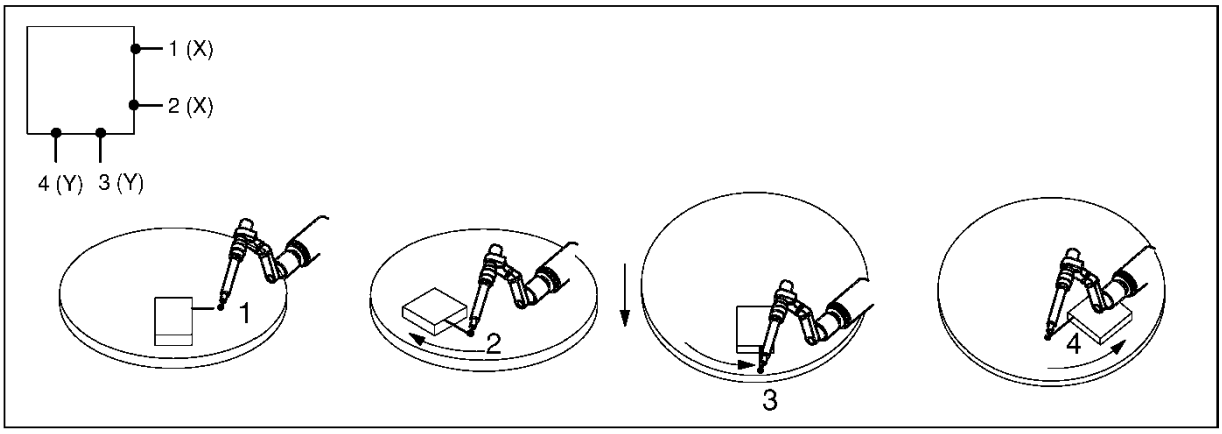


Fig. 39.9(a) 2D+rotation search with coordinated motion

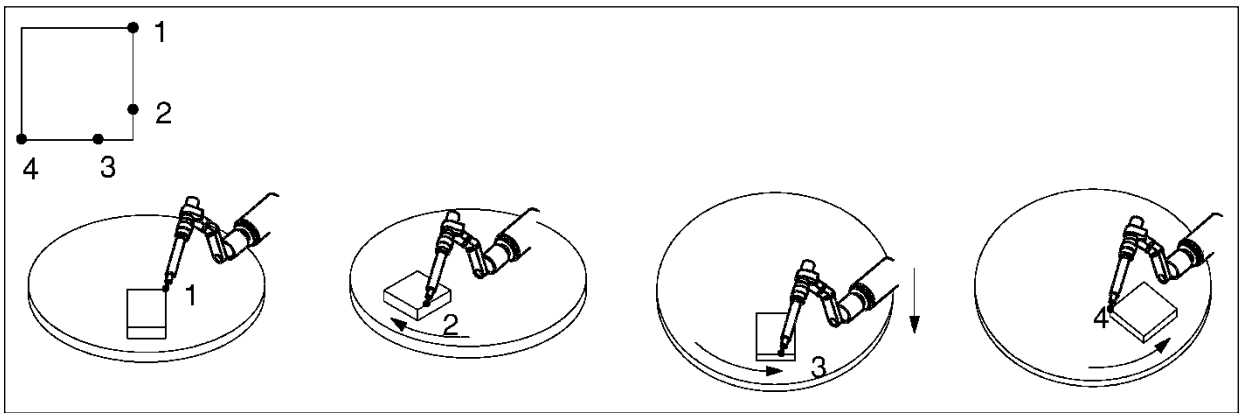


Fig. 39.9(b) 2D+rotation Touch offset with coordinated motion

Thetouch frame setup is as follows.
Enter the group number of the robot group (follower group) that performs search motion and touch offset to in the RoBot Grp item. After that, enter the group number of the coordinated motion leader group in the Ref Grp item. This means that, when Coordinated Touch Sensing is performed, the Ref Grp number and Robot Grp number in the touch frame are different. For more details, refer to Procedures 39-3 and 39-4.
Before the setup, you should already have completed the calibration of the coordinated pair between leader group and follower group in coordinated motion. If you change the Ref Grp item in the touch frame without calibrating the coordinated pair, the message "Referenced group does not exist" will be displayed. For information on calibrating coordinated pairs, refer to the *Coordinated Motion Function OPERATOR'S MANUAL*.

In addition, the Ref Grp number and Robot Grp number in the touch frame must also be entered in the touch schedule. Set the robot group for Touch Sensing to the Robot Group Mask item on the Touch Schedule Detail screen. Next, enter the touch frame number in the Touch Frame item. Then, enter the same number as the Ref Grp number (Leader group number) in the touch frame in the Reference Group item.

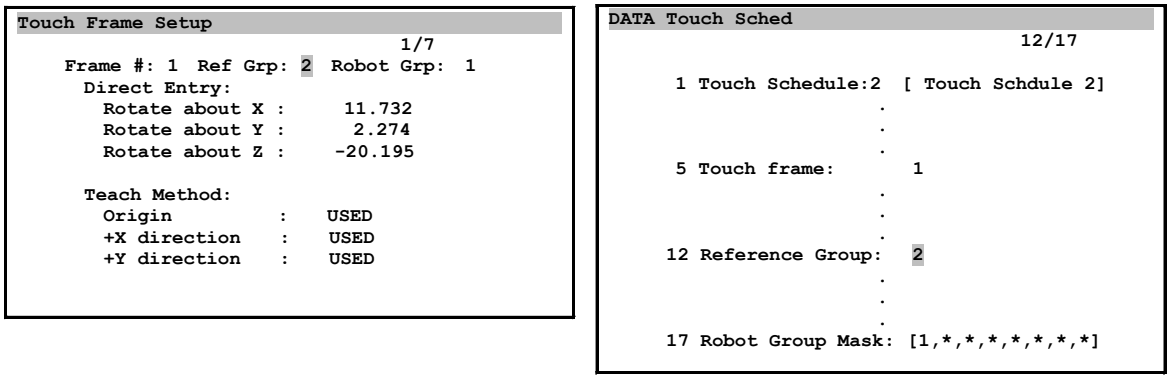


Fig. 39.9(c) Setup example of touch frame and touch schedule (Coord, except simple search)

Fig. 39.9(d) is the program example. The search direction is derived from the direction of the leader group (positioner in the figure). Even if the leader group moves, the search direction will not be changed relatively. It is also possible to teach motion instructions that are not related to search motion (refer to Line 7 of Fig. 39.9(d)). The offset data obtained by search motion is compensated by the direction of the leader group.

Group Mask in Program[1,1,*,*,*,*,*,*]

```
COORD_SEARCH
                                        19/20
    1:J  P[1] 20% FINE
    2:  Search Start[2] PR[2]
    3:J  P[2] 50% FINE
    4:J  P[3] 50% FINE[X]
    5:J  P[4] 50% FINE
    6:J  P[5] 50% FINE Search[X]
    7:J  P[6] 50% CNT100
    8:J  P[7] 50% FINE
    9:J  P[8] 50% FINE Search[Y]
   10:J  P[9] 50% FINE
   11:J  P[10] 50% FINE Search[Y]
   12:  Search End
   13:J  P[1] 50% FINE
   14:  Touch Offset PR[2]
   15:L  P[11] 100mm/sec FINE
     :  COORD Weld Start[1,1]
   16:L  P[12] 80cm/min CNT100
     :  COORD
   17:L  P[13] 80cm/min FINE
     :  COORD Weld End[1,2]
   18:  Touch Offset End
   19:J  P[1] 50% FINE
```

Teach Home Position (Leader/Follower groups)
Start Search Motion: Touch Sch 2, PR[2]
Teach a search start position (Leader/Follower)
Search in the X direction (Only follower group)
Teach another search start point (Leader/Follower)
Search in the X direction (Only follower group)
Teach an intermediate point (Leader/Follower)
Teach another search start point (Leader/Follower
Search in the Y direction (Only follower group)
Teach another search start point (Leader/Follower
Search in the Y direction (Only follower group)
Finish Search Motion
Move to Home Position
Following points are offset by PR[2]
P[11] is offset by PR[2]

P[12] is offset by PR[2]

P[13] is offset by PR[2]

Finish offset of positions
Teach Home Position

Fig. 39.9(d) 2D + rotation search with coordinated motion

NOTE
When reference group setup is performed for coordinated motion, mastering information of all groups related to the reference group are stored in the TP program. If you change the reference group in the touch frame or touch schedule, perform remastering.

Coordinated Touch Sensing with Simple Search

The touch frame for simple search can be based on either the world frame of the follower group or coordinated frame of the leader group.
If you select the world frame of the follower group as the basis, enter the group number of the follower group. If you select the coordinated frame of the leader group as the basis, enter the group number of the leader group. In this case, you must complete the calibration of the coordinated pair. In addition, the absolute position data in the position register obtained by the search motion is based on the follower group.

In the touch schedule for a simple search, the reference group number must be the same as the Robot Group Mask.
Fig. 39.9(e) is a program example of simple search with coordinated motion.

Group Mask of TP Program[1,1,*,*,*,*,*,*]

```
COORD_SEARCH2
                                        11/12
     1:J  P[1] 50% FINE
     2:  Search Start[2] PR[2]
     3:J  P[2] 50% FINE
     4:J  P[3] 50% FINE Search[X]
     5:J  P[4] 50% FINE
     6:J  P[5] 50% FINE Search[Y]
     7:  Search End
     8:J  P[6] 50% FINE
     9:L  PR[2] 100mm/sec FINE
      :  Weld Start[1,1]
    10:L  P[7] 80cm/min FINE
      :  COORD
    11:  Weld End[1,2]
```

Teach Home Position (Leader/Follower groups)
Start Search Motion, Touch Sch 2, PR[2]
Teach a search start position (Leader/Follower)
Search in the X direction (Only follower group)
Teach another search start point (Leader/Follower)
Search in the Y direction (Only follower group)
Finish Search Motion
Teach an intermediate point (Leader/Follower groups)
Move the robot to PR[2]
Start Arc Welding
Coordinated motion

Fig. 39.9(e) Program example of simple search with coordinated motion

⚠CAUTION
Search motions and touch offsets for coordinated motion cannot be mixed with non-coordinated motion. That is, you cannot perform searches for coordinated motion, then perform searches for non-coordinated motion, and then use the offset from the coordinated motion search. You must use the offsets for coordinated motion searches before performing any non-coordinated searches. Similarly, you must use any offsets from non-coordinated motion searches before performing any coordinated motion searches.

NOTE
The leader group is not allowed to move between the searches.

40 TOUCH SKIP FUNCTION

The Touch Skip function monitors the disturbance values of multiple axes simultaneously, and performs a high-speed skip when the disturbance value of any axis exceeds the threshold value.
You set the disturbance threshold values on the SETUP/ TOUCH SKIP screen.

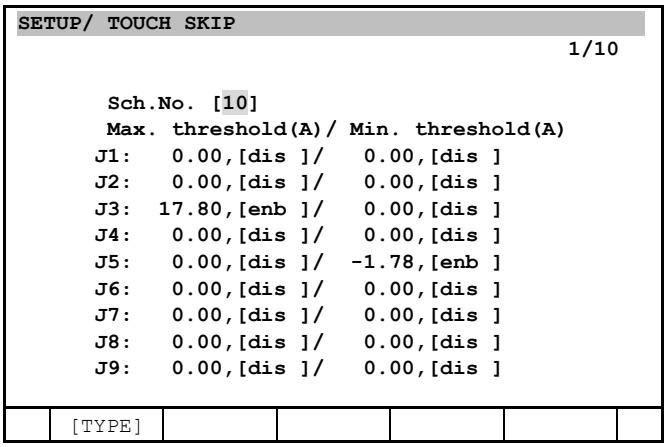
When the Touch Skip function is installed,
- The SETUP/ TOUCH SKIP screen can be displayed.
- Touch Skip programs can be called.

This function requires the Touch Skip function option (A05B-2700-J921).
Some robot models do not support this function.

40.1 TOUCH SKIP SCREEN

You can set up the Touch Skip thresholds on the SETUP/TOUCH SKIP screen.

1. Display the SETUP/ TOUCH SKIP screen: MENU → SETUP → F1, [TYPE] → Touch Skip.



2. To change the schedule number, enter a value in the Sch. No. field.
3. In the subsequent lines, set the values and enable or disable the disturbance thresholds for the axes you want to set.
4. You can check the disturbance during motion on the STATE → AXES → DISTURBANCE screen and set thresholds with added margins if necessary.

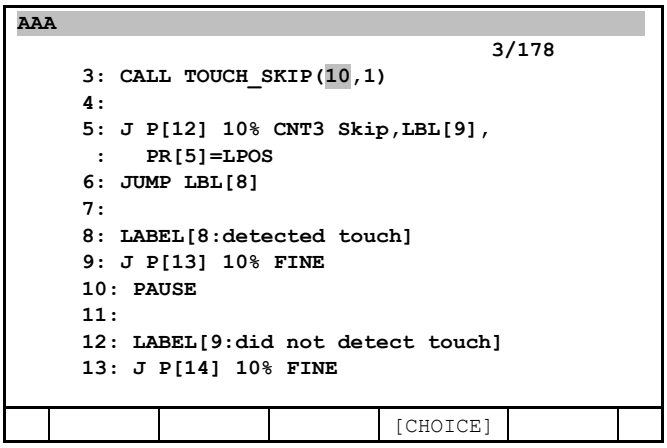
In the screen shown above, a schedule with a J3 disturbance larger than 17.8[A] or J5 disturbance is smaller than –1.78[A] is set.

40.2 TOUCH SKIP PROGRAM

Include the following program instruction to set a skip condition.

- TOUCH_SKIP(s, g)
  This program sets a skip condition as when the disturbance on one of the axes in group [g] exceeds the threshold of schedule number [s] set on the SETUP/ TOUCH SKIP screen.

For example, you can use this program as follows.



- In line 3, a Touch Skip condition is specified as when the disturbance on one of the axes in group [1] exceeds the threshold of schedule number [10] set on the SETUP/ TOUCH SKIP screen.
  In the same way as for a SKIP CONDITION instruction, this condition is effective until the program ends or another skip condition is enabled.
- A motion with a high-speed skip instruction in line 5 works in the same way as a standard high-speed skip instruction.
  If the disturbance exceeds the threshold on one of the axes in group [1], the robot stops and then the next line (line 6) is executed.
  If the disturbance does not exceed the threshold on any axis in group [1] and the robot reaches the destination point, the jump to label [9] is executed.

40.3 NOTES / LIMITATIONS

- To decrease the impact of stopping when a skip condition is satisfied, the speed of the motion with a Touch Skip instruction is limited by an upper speed limit. This upper speed limit is adjusted for each model. If the taught speed exceeds the limit, the speed is limited automatically and the following warning will be posted:

  MOTN-560 Hspd.skip speed limit (G:group number)

- Some models are designed to decelerate to a stop during high-speed skip instruction execution. With such models, you cannot use Touch Skip. If motion with a Touch Skip instruction is executed in such models, the following alarm will be posted:

  MOTN-561 Touch skip NOT available (G:group number)

- Touch Skip cannot be used with the Skip instruction (Skip, LBL). If Touch Skip is executed with a Skip instruction (Skip, LBL), the following alarm will be posted. When you use Touch Skip, you must use the high-speed skip instruction (Skip, LBL, PR).

  MOTN-561 Touch skip NOT available (G:group number)

- On the R-50iA controller and later, Touch Skip can be used on auxiliary axes. Note that if a robot model does not support Touch Skip, Touch Skip cannot be used on auxiliary axes on these robot groups either.

41 MROT INSTRUCTION

The MROT (Minimal Rotation) Instruction is a motion modifier that can be used with the Joint motion type, or Cartesian motion type with the WJNT motion modifier. It generates the shortest joint angle move for the wrist axes within the axis limits during Joint and WJNT motion, and ignores the turn numbers of the wrist. For example, if the start and destination angles are 150° and -130°, the shortest move is the motion from 150° to 230°. This is because 230° and -130° are the same position for the axis physically, and the 230° position requires less rotation to reach.

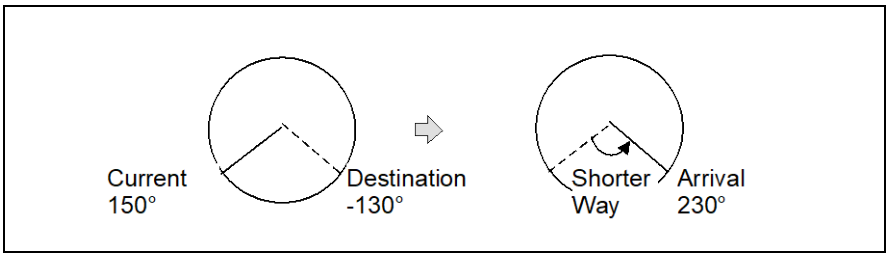


Fig. 41 MROT instruction

To use this function, the MROT Instruction option is required.

41.1 HOW TO USE MROT

Add an MROT motion modifier to Joint or WJNT motion.

J P[1] 100% FINE MROT
L P[2] 2000mm/sec CNT100 WJNT MROT

41.2 LIMITATIONS

The MROT motion modifier has the following limitations:

- This function is available only with Handling Tool.
- The MROT motion modifier takes effect when the destination position of the Joint or WJNT motion is represented in Cartesian space. If the destination position is represented in Joint angles, the wrist axes always reach to the taught joint angles even if the MROT motion modifier is added.
- MROT generates the shortest joint move within the axis limits. For example, if the upper limit of wrist axis is 180°, and the start and destination angles are 150° and -130°, the warning "MOTN-330 MROT Limit Warn" will occur and the axis will move from 150° to -130° because 230° is out of limit.

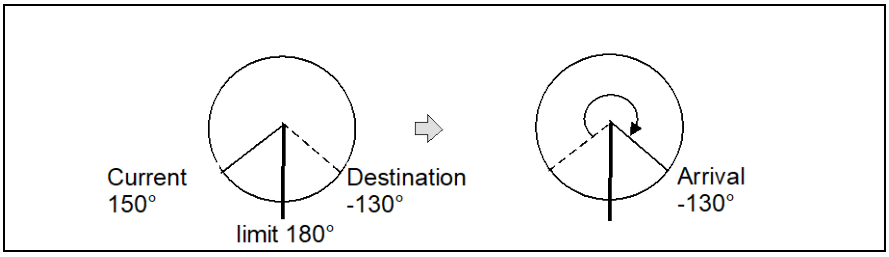


Fig. 41.2 Shorter way is out of limit.

- Single step backward motion may not be the same as forward motion in terms of path.

For example:
1: J P[1] 100% FINE
2: J P[2] 100% FINE MROT
3: J P[3] 100% FINE

If the actual reached joint angles at P[2] are different than the taught joint angles (in terms of turn number), then the backward motion from P[3] to P[2] will be different than the forward motion from P[2] to P[3] in terms of path. Because the backward motion from P[3] to P[2] is not MROT motion, the arrival position is the joint angles taught in P[2].

42 SMALL CIRCLE ACCURACY IMPROVEMENT

The Small Circle Accuracy Improvement function allows you to achieve path accuracy improvements around small circles in relatively low-speed operations, such as laser and waterjet applications.

The software option A05B-2700-J644 is required to use this function.
This function is available only for specific robots.

> ⚠ **WARNING**
> Overshoot might occasionally increase during high-speed motion (for example, air cutting motion) when this option is used.
> In this case, reduce the teaching speed or reduce the acceleration using the ACC override modifier.

42.1 SPECIFICATIONS

The Small Circle Accuracy Improvement option has the following benefits:
1) Path accuracy in relatively low-speed motion can be improved.
2) The compensation value can be changed using a macro program prepared in advance, to select suitable compensation values based on the robot position and posture.

The Small Circle Accuracy Improvement option is composed of following:
- Automatic setting of special parameters for backlash and friction compensation
- Macro program to change compensation
- TP screen for setting compensation schedule data

> ⚠ **CAUTION**
> Damage to the robot and tools might occur if the compensation gain is set to an improper value.
> Be careful when you change the compensation gain value.
> Change the compensation gain value a little bit at a time to minimize the risk of damage.

42.2 SETTINGS

42.2.1 PARAMETER SETTINGS

When this option is installed, the default backlash and friction compensation values are automatically set.

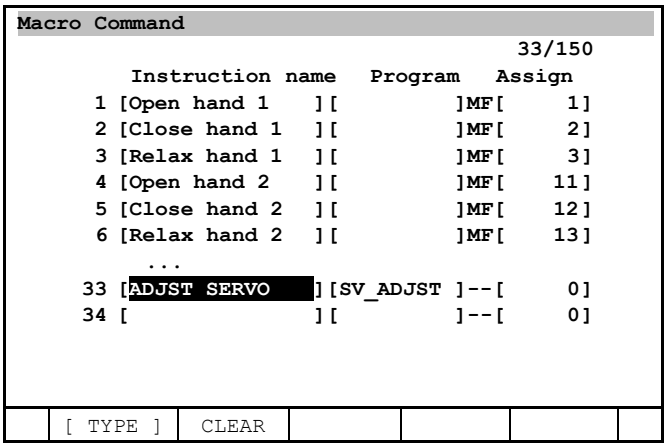
Therefore, you do not have to set these compensation values.

42.2.2 MACRO PROGRAM : SV_ADJST.TP

This macro program is used to change the compensation values of backlash and friction.

This program requires one argument setting, so it cannot be executed individually.

By default, this macro program is registered as shown in the following screen.
If necessary, change the macro name and so forth on the Setup/Macro screen.

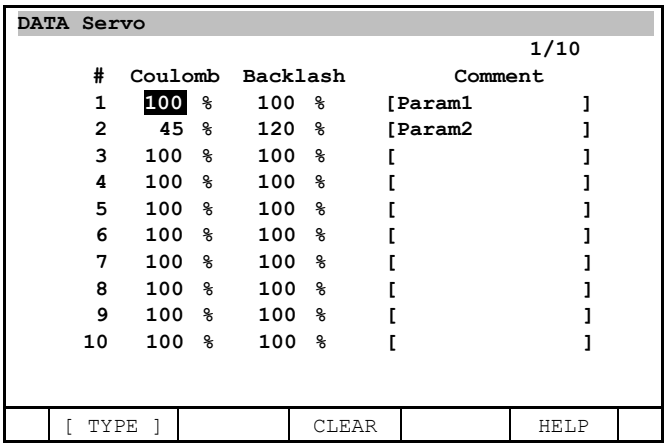


> **⚠ CAUTION**
> Do not execute this macro program on the fly, such as between CNT motions. Be sure to execute this macro program during robot stops, after FINE or CNT0 motion.
> Otherwise, the robot and tools might be damaged.

42.2.3 COMPENSATION VALUE SETTING SCREEN

You can use the servo schedules on this screen to fine-tune cut quality if needed. Tuning small shapes involves optimizing both the friction compensation and backlash compensation gains.

To display following screen :
DATA → F1, [TYPE] → Select Servo Sched



> ⚠ **CAUTION**
> For robots with wall or angle mount configurations, backlash compensation can make the trajectory worse. In these cases, set the backlash compensation gain to 0%.
> Press F5, HELP to check the valid range of gain values.

42.3 PROGRAMMING

The following example program changes the compensation values in lines 4-12.

Program	Description
1:J P[1] 100% FINE 2:J P[2] 100% FINE 3: **4: ADJST SERVO(1)** 5: 6: SDO[1]=ON 7:L P[3] 37mm/s CNT100 8:C P[4] : P[5] 37mm/s CNT100 9:C P[6] : P[3] 37mm/s FINE 10: SDO[1]=OFF 11: **12: ADJST SERVO(0)** 13: 14:J P[1] 100% FINE [End]	Changes the compensation values set in schedule #1. Resets the compensation values to default values.

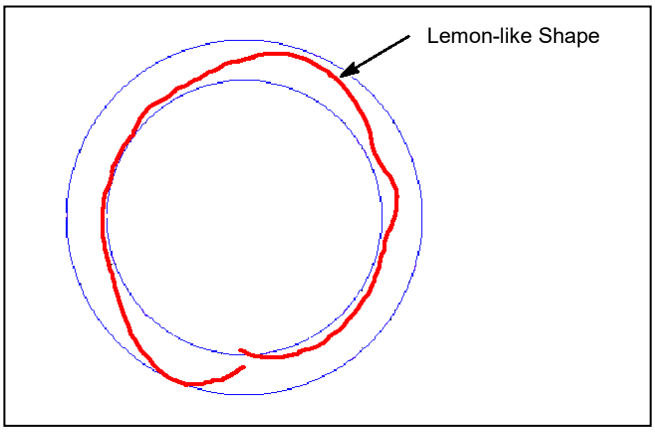
> ⚠ **WARNING**
> It is very important to reset the servo gains to their default values after the cut is performed.
> Also, if the program is aborted or paused in lines 5-11, the compensation value is retained by schedule #1 even if “Program Abort” is executed.

42.4 TROUBLE SHOOTING (TUNING GUIDELINES)

Use following guidelines for the following shapes:

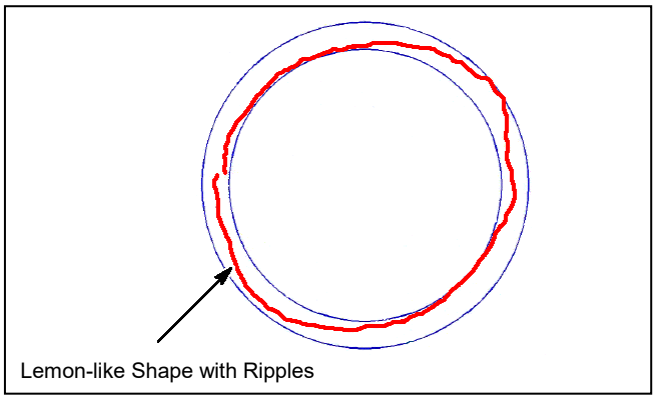
- Lemon shape circle
- Lemon shape circle with ripples
- Circle with flat spot
- Circle with inward bump

42.4.1 LEMON SHAPE CIRCLE



To improve the lemon-like shape of the circle, increase the backlash compensation gain.

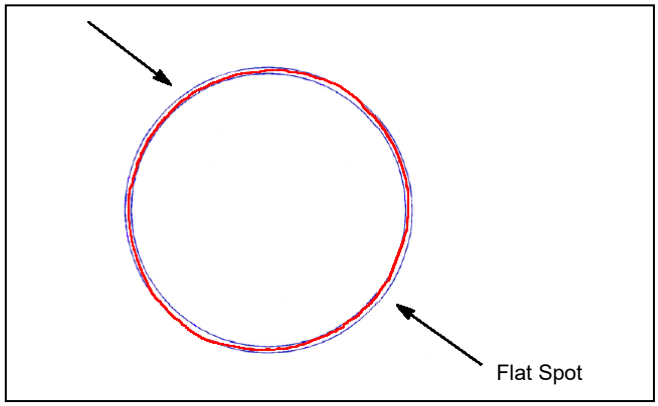
42.4.2 LEMON SHAPE CIRCLE WITH RIPPLES



To avoid a lemon shape circle with ripples:

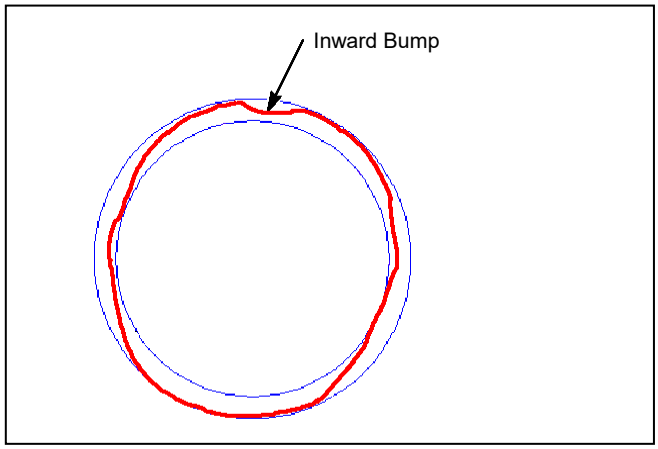
- Lower the backlash compensation gain.
- Adjust the Coulomb friction compensation gain up or down.

42.4.3 CIRCLE WITH FLAT SPOT



To avoid a circle with a flat spot, lower the Coulomb friction compensation gain to reduce or eliminate the flat spots.

42.4.4 CIRCLE WITH INWARD BUMP



To avoid this shape, lower the backlash compensation gain to reduce or eliminate the inward bump.

43 SHAPE GENERATION FUNCTION

43.1 OVERVIEW

NOTE
This function is an option that requires the Shape Generation Function 3 (A05B-xxxx-R801).

The Shape Generation function simplifies the creation of machining programs that cut two-dimensional shapes. This function also reduces online teaching time by automatically generating a simple robot motion program using a graphic interface.
You can use the Shape Generation function to generate robot motion programs for the following shapes.:

- Circles
- Slots
- Hexagons
- Rectangles
- Keyholes
- Pommels
- Customs

NOTE
Since cutting tools move only in the X-Y plane, pommel shapes cannot be used.

To perform machining operation using this function, you must do the following:
1. Input shape settings and data on the SETUP Shapes screen.
2. Generate shape programs from the UTILITY Shape screen.
3. Teach process instructions and the correct program in the generated program.
4. Run the program.

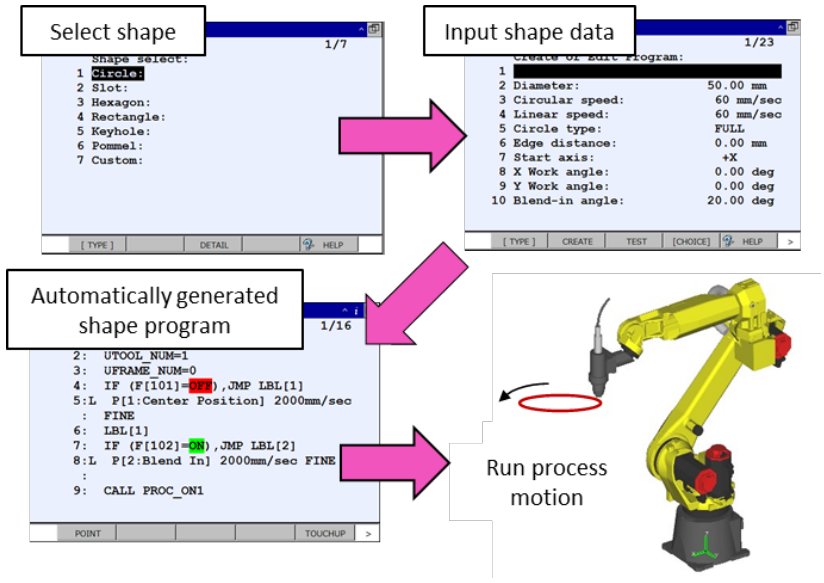


Fig. 43.1 Shape Generation overview

> ⚠ **WARNING**
> When an error occurs, make sure that the program stops the machining process (such as laser cutting or water jet cutting). The Shape Generation function does not automatically stop the machining process when an error occurs. Design a system in which the machining process is properly stopped when an error occurs. If the machining process is not properly stopped, injury to personnel or damage to equipment could occur.

43.1.1 Shape Generation Programs

The following TP programs are required for machining using the Shape Generation function:

・Shape program
This program is generated based on information entered on the UTILITIES Shape screen. This program defines the path for shape generation and performs the process defined in the process program. This program includes only program instructions for these purposes.

・Production program
This is the master program for calling the shape program. In this program, teach all required program instructions except shape generation instructions.

> ⚠ **WARNING**
> In the production program, make sure that the calling sequence of the shape program is performed correctly for the proper processing.
> When an error occurs, make sure that the production program stops the machining process (such as laser cutting or water jet cutting). The Shape Generation function does not automatically stop the machining process when as error occurs. Design a system in which the machining process is properly stopped when an error occurs. If machining process is not properly stopped, injury to personnel or damage to equipment could occur.

・Process program（optional）
This program is used to define the machining process. When creating a shape program other than for a custom shape, you can use PROC_ON#.TP and PROC_ON#.TP as model programs (# is the group number).
If you do not use the model programs, you can create the machining process by directly editing the automatically generated shape program.

43.2 SHAPE GENERATION SETUP

You perform basic Shape Generation setup on the SETUP Shapes screen. Perform common setup for all shapes and default setup for each specific shape .
The Shape Generation function allows you to create robot motion programs for the following shapes.
・Circles
・Slots
・Hexagons
・Rectangles
・Keyholes
・Pommels
・Custom shapes

You set the default settings for each shape on the detail setup screen for each shape. The settings you make on the SETUP Shape screen are used as the default settings when using the Shape Generation utility. Adjust the Shape Generation based on these settings.

Procedure 43-1 Setting Up Shape Generation

Step

1 Press MENU.
2 Select SETUP.
3 Press F1, [TYPE].
4 Select Shape. The following screen will be displayed.

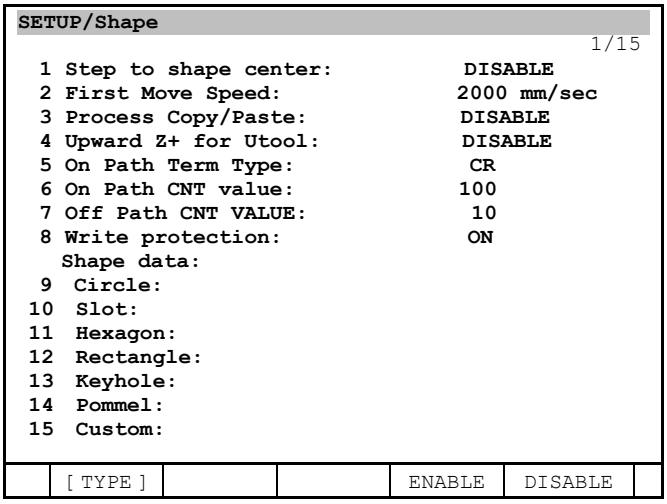


Fig. 43.2 SETUP Shape screen

5 Select the desired shape type (Circle, Slot, Hexagon, Rectangle, Keyhole, Pommel, Custom), and press F3, DETAIL.
6 When you are finished changing setup items, press F2, SAVE to apply the new settings as the default setting values.

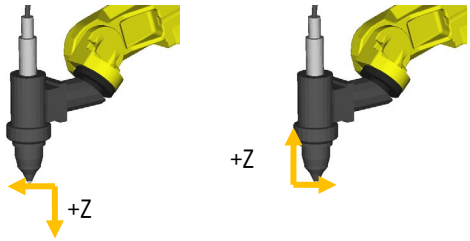
Refer to Table 43.2.1 for the basic Shape Generation setup items, Table 43.2.2 for setup items common for all shapes except custom shapes, and Table 43.2.3 to Table 43.2.9 for shape-specific setup items.

43.2.1 Basic Setup

The Shape Generation basic setup items are listed in Table 43.2.1.

Table 43.2.1 Shape Generation basic setup items

ITEM	DESCRIPITION
Step to shape center	This item defines whether the robot steps to the shape center (Enabled) or steps directly to the pierce position (Disabled).
First Move Speed	This item indicates the motion speed of the robot to the first position in the program.
Process Copy/Paste	When this item is enabled, the information contained in the PROCESS_ONx.TP and PROCESS_OFx.TP programs is copied to the shape program. When this is disabled, the CALL to the process programs is used. For details, refer to "Process Copy/Paste " below.

ITEM		DESCRIPITION
Upward Z+ for Utool		This item is used when the standard ArcTool UTOOL defines the tool Z direction to go into the TCP, instead of away from the TCP.  invers : invaluid invers : valid
On Path Term Type		This item is used to control the termination type used to get onto the initial part of the shape, where the choice will be CR or CNT. If CNT is specified, the On Path CNT value will be used. If CR is specified, On Path CNT value is ignored.
On Path CNT value		This value is used if the On Path Term Type CNT is specified. If CR is specified, this value is ignored.
Off Path CNT VALUE		This item is used to adjust the CNT value as the motion leaves the shape path.
Write Protection		This item automatically sets the shape to be write-protected when the shape is created. It is set to TRUE by default.
Shape data	**Circle**	Selecting this item allows you to set up default shape information for Circles.
	Slot	Selecting this item allows you to set up default shape information for Slots.
	Hexagon	Selecting this item allows you to set up default shape information for Hexagons.
	Rectangle	Selecting this item allows you to set up default shape information for Rectangles.
	Keyhole	Selecting this item allows you to set up default shape information for Keyholes.
	Pommel	Selecting this item allows you to set up default shape information for Pommels.
	Custom	Selecting this item allows you to set up default shape information for Custom shapes.

Process Copy/Paste

When Process Copy/Paste is disabled, the program calling PROC_ON1.TP and PROC_OF1.TP is generated as follows.

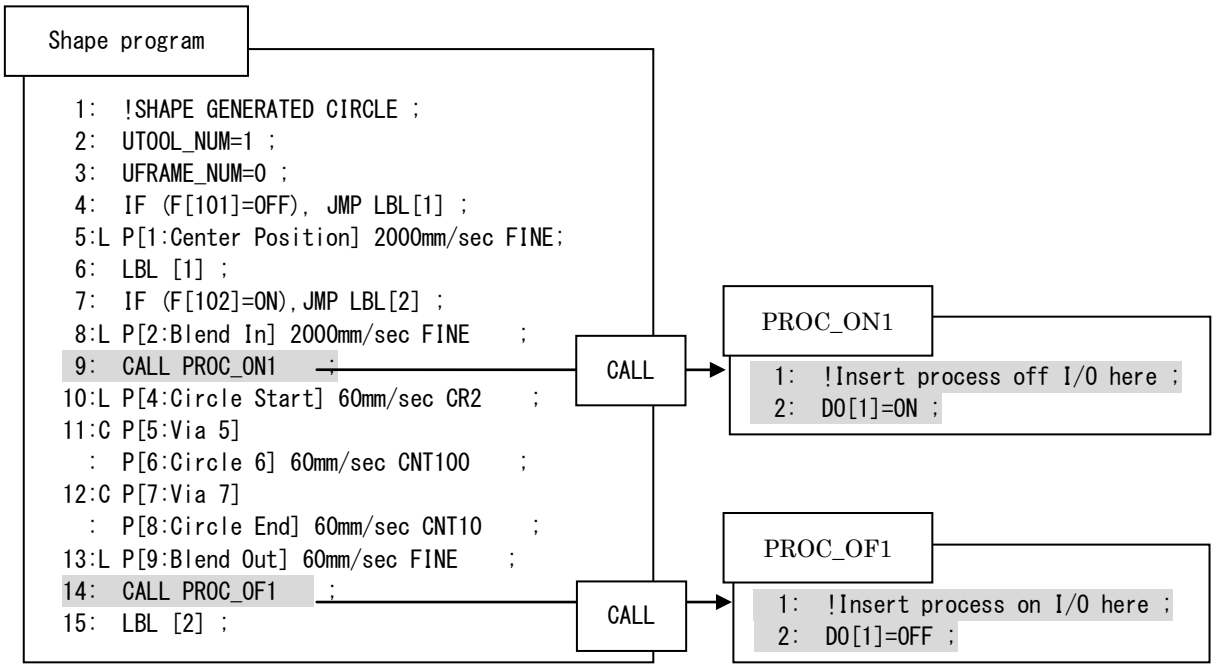


Fig. 43.2.1(a) Shape program example (Process Copy/Paste is disabled)

When Process Copy/Paste is enabled, the program in which program processing defined in PROC_ON1.TP and PROC_OF1.TP is automatically inserted in the default location as follows.

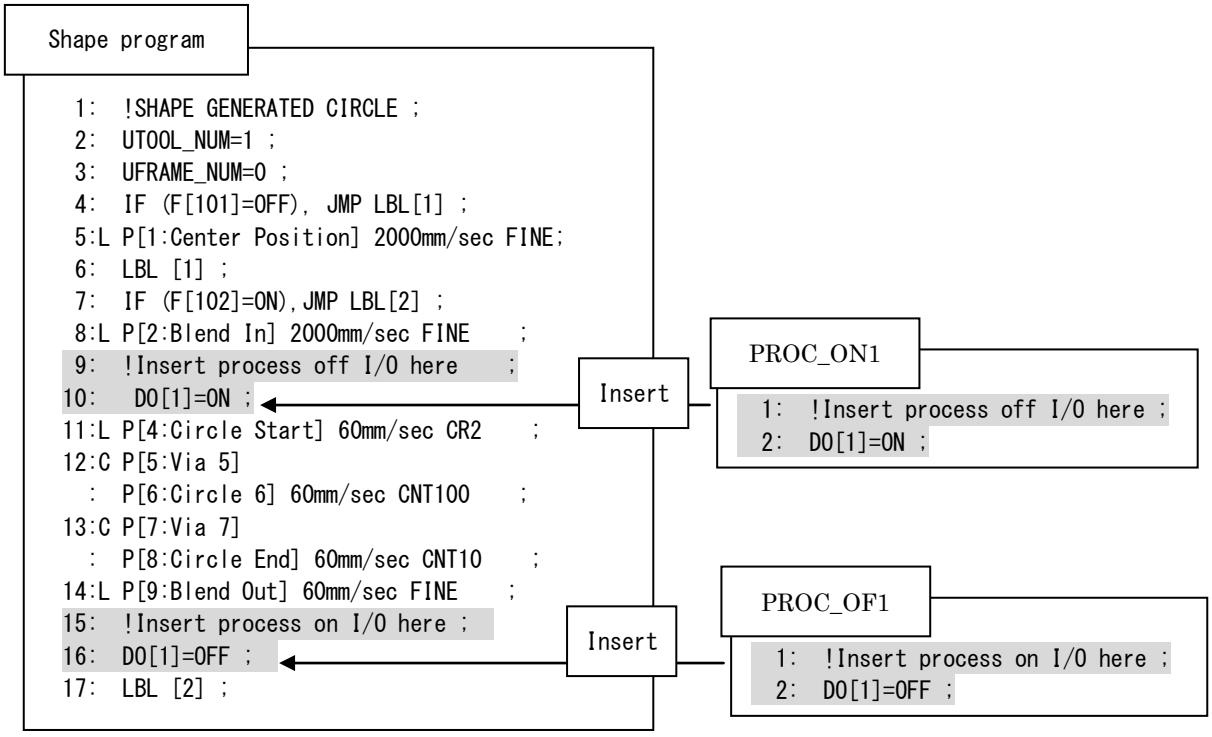


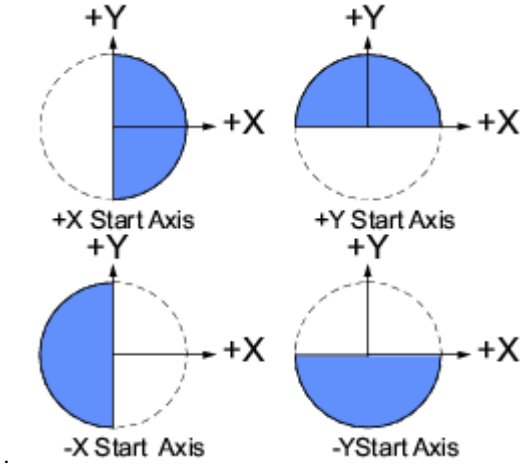
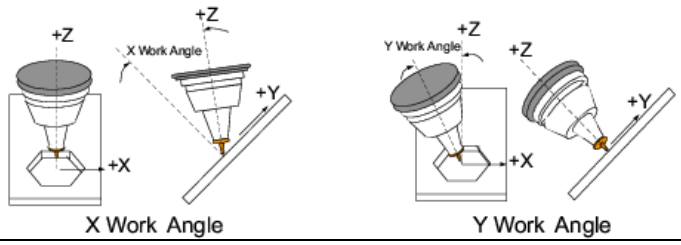
Fig. 43.2.1(b) Shape program example (Process Copy/Paste is enabled)

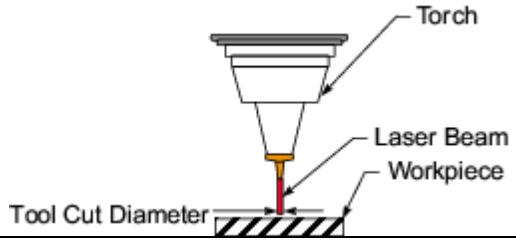
NOTE
The generated shape program automatically includes call instructions for PROC_ON and PROC_OF. If these programs do not exist, the system software will automatically create them. Once created, the system software will not overwrite them, so you can freely edit the instructions for process processing.

43.2.2 Common Settings for All Shapes

Table 43.2.2 lists the Shape Generation settings common to all shapes except Custom shapes.

Table 43.2.2 Shape Generation common settings

ITEM	DESCRIPTION
Start Axis (+/-X or +/-Y)	This item is the axis on the shape at which the robot will begin its cut. Each of the four start axis choices are shown. Note: Half circle orientation depends on which start axis is selected 
X-Work angle Range : ±45°	The X and Y Work angles allow you to program beveled cuts. The angles are measured from normal around the corresponding shape axis (X or Y). 
Y-Work angle Range : ±45°	
Blend-in angle Range : ±45°	This item is the angle from the start axis to the start position.
Blend in distance Range : ±20 mm	This item is the distance from the shape edge to the start position.
Overlap angle Range : ±45°	This item is the angle between the start axis and the end position.
Blend-off angle Range : ±45°	This item is the angle from the start axis to the end position inside the shape.
Blend-off distance Range : ±20 mm	This item is the distance from the shape edge to the end position.

ITEM	DESCRIPTION
Number of circle points This item appears only in Circle, Slot, Keyhole, Rectangle, and Pommel shapes	This item is the number of points you choose to define around the circle. For rectangle shapes, normally this is set to 0 and use CR with a single point in the corner. If this is a non-zero value, the corner will be turned into circular motion and the number indicates how many points will be used in each corner.
Cut direction	This item is the direction the TCP travels around the shape.
Rotation	This item allows the shape to rotate around the TCP.
Tool cut diameter	This item is the diameter around the TCP when cutting. 
Plunge distance	This item is the distance the tool travels down into the workpiece.
Plunge speed	This item is the speed of the tool as it travels down into the workpiece.
Retract distance	This item is the distance the tool travels up out of the workpiece when the cut is completed.
Retract speed	This item is the speed of the tool as it travels up out of the workpiece when the cut is completed.

NOTE

The cutting tool moves only in the X-Y plane, so the following items are not available:

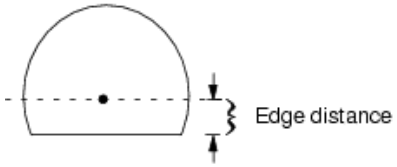
- X-Work angle
- Y-Work angle
- Plunge distance
- Plunge speed
- Retract distance
- Retract speed

43.2.3 Circle Shape Setup

Table 43.2.3 lists the specific circle shape setup items.

Table 43.2.3 Circle shape setup items

ITEM	DESCRIPTION
Diameter **Range : 1 - 500 mm**	This item is the diameter of the current circle or half circle.
Circular speed **Range : 1 - 9999 mm/sec**	This item is the cutting speed of the circle or half circle.
Linear Speed **Range : 1 - 9999 mm/sec**	This item is the cutting speed of the line.
Circle Type **(Full/D_Shape)**	This item specifies whether the circle is a full circle or D-shape circle.

ITEM	DESCRIPTION
Edge Distance **Range : -250 - 250 mm**	This item specifies the distance from the circle center to where the D-shape edge will be. Edge distance is used only when the Circle Type is set to D_shape. 

Refer to Fig.43.2.3 for the correspondence of the setup items in the generated shape.

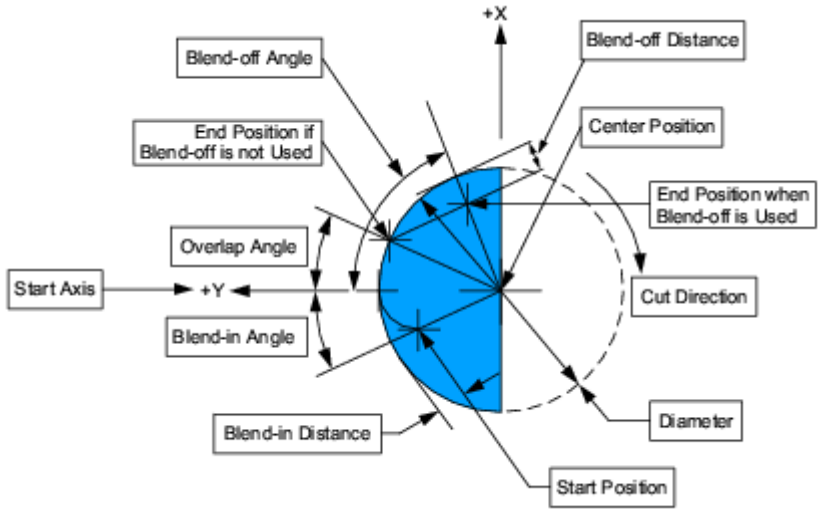


Fig.43.2.3 Circle shape setup

43.2.4 Slot Shape Setup

Table 43.2.4 list the specific slot shape setup items.

Table 43.2.4 Slot shape setup items

ITEM	DISCRIPTION
Length **Range : (1 - 500 mm)**	This item is the length across the longer side of the current slot.
Width **Range : (1 - 500 mm)**	This item is the width across the shorter side of the current slot.
Speed **Range : (1 - 9999 mm/sec)**	This item is the cutting speed of the current slot.

Refer to Fig. 43.2.4 for the correspondence of the setup items in the generated shape.

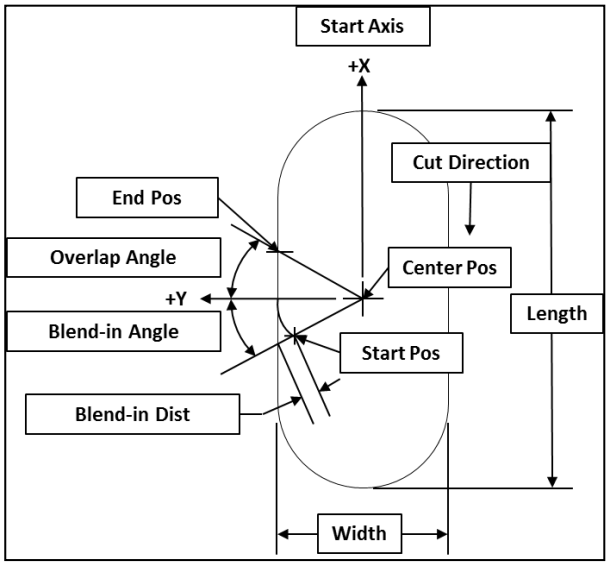


Fig.43.2.4 Slot shape setup

43.2.5 Hexagon Shape Setup

Table 43.2.5 list the specific hexagon shape setup items.

Table 43.2.5 Hexagon shape setup items

ITEM	DISCRIPTION
Diameter **Range : (1 - 500 mm)**	This item is the diameter across the corners of the current hexagon.
Corner Radii	This item specifies the radius of the corners (all corners are identical).
Speed **Range : (1 - 9999 mm/sec)**	This item is the cutting speed of the current hexagon.

Refer to Fig.43.2.5 for the correspondence of the setup items in the generated shape.

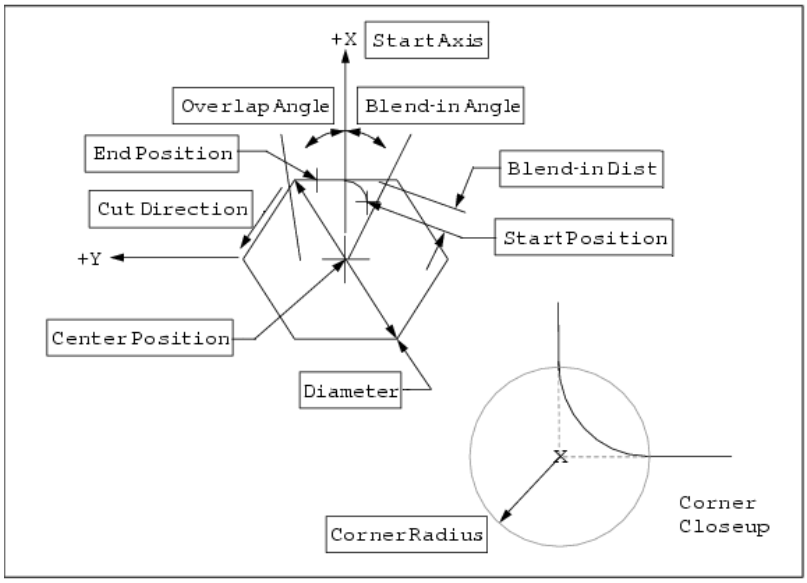


Fig.43.2.5 Hexagon shape setup

43.2.6 Rectangle Shape Setup

Table 43.2.6 lists the specific rectangle shape setup items.

Table 43.2.6 Rectangle shape setup items

ITEM	DISCRIPTION
Length **Range : (1 - 500 mm)**	This item is the length across the longer side of the current rectangle.
Width **Range : (1 - 500 mm)**	This item is the width across the shorter side of the current rectangle.
Speed **Range : (1 - 9999 mm/sec)**	This item is the cutting speed of the current rectangle.
Radius 1	This item specifies the radius of the first corner encountered in the rectangle depending on the cut direction.
Radius 2	This item specifies the radius of corner 2.
Radius 3	This item specifies the radius of corner 3.
Radius 4	This item specifies the radius of corner 4.

Refer to Fig.43.2.6 for the correspondence of the setup items in the generated shape.

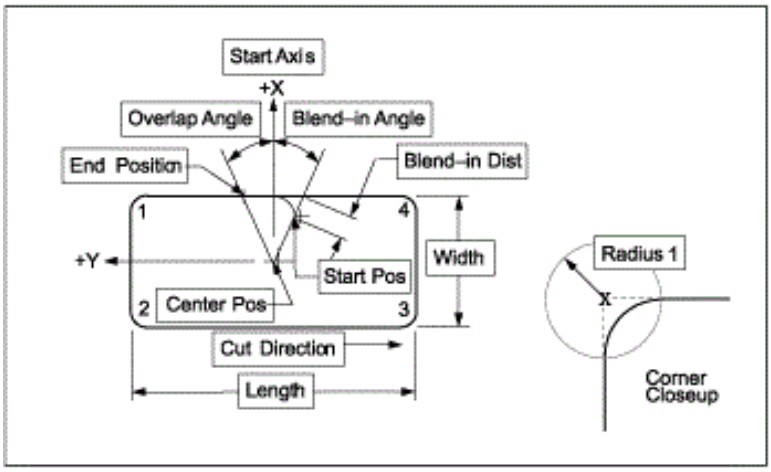


Fig.43.2.6 Rectangle shape setup

43.2.7 Keyhole Shape Setup

Table 43.2.7 lists the specific keyhole shape setup items.

Table 43.2.7 Keyhole shape setup items

ITEM	DISCRIPTION
Major Diameter **Range : (0 - 500 mm)**	This item specifies the diameter of the large circle.
Minor Diameter **Range : (0 - 500 mm)**	This item specifies the diameter of the small radius, which is also the width of the slot.
Length **Range : (0 - 500 mm)**	This item specifies the overall length of the keyhole.
Circular Speed **Range : (1 - 9999 mm/sec)**	This item specifies the cutting speed at circular portion of the current keyhole. You can also specify the speed units: mm/sec, cm/min, and in/min.
Linear Speed **Range : (1 - 9999 mm/sec)**	This item specifies the cutting speed at the linear portion of the current keyhole. You can also specify the speed units: mm/sec, cm/min, and in/min.

Refer to Fig.43.2.7 for the correspondence of the setup items in the generated shape.

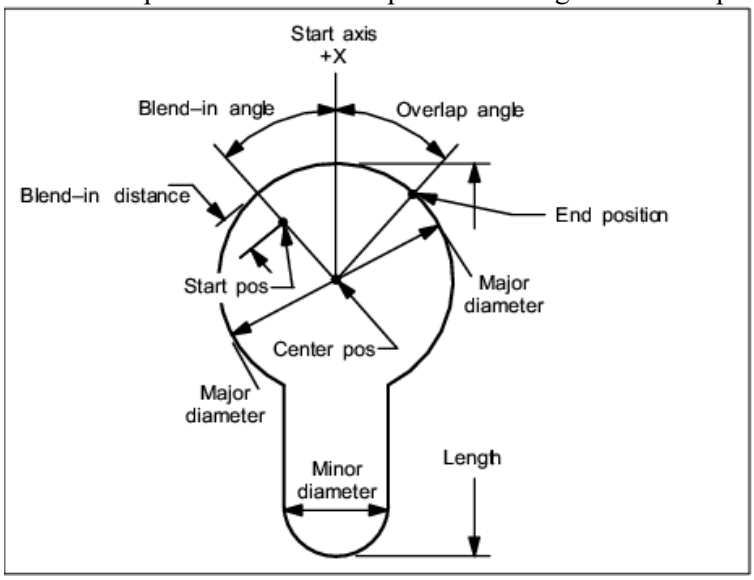
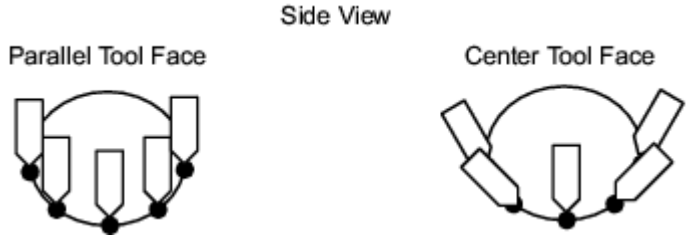


Fig.43.2.7 Slot shape setup

43.2.8 Pommel Shape Setup

Table 43.2.8 lists the specific pommel (cut of pipe) shape setup items.

Table 43.2.8 Pommel shape setup items

ITEM	DISCRIPTION
Diameter **Range : (0 - 500 mm)**	This item specifies the diameter of the circle to process on the pipe.
Object Diameter **Range : (0 - 500 mm)**	This item specifies the diameter of the pipe. This must be larger than the diameter of the circle.
Tool-Face toward **Default: Parallel**	This item specifies the direction the face of the tool will point during rotation: Parallel: keeps the face of the tool parallel to the pipe's X-Y plane. Center: directs the face of the tool to the origin of the pipe's X-Y plane. 
Circular Speed **Range : (1 - 9999 mm/sec)**	This item specifies the cutting speed at circular portion. You can also specify the speed units: mm/sec, cm/min, and in/min.
Linear Speed **Range : (1 - 9999 mm/sec)**	This item specifies the cutting speed at linear portion. You can also specify the speed units: mm/sec, cm/min, and in/min.

Refer to Fig.43.2.8 for the correspondence of the setup items in the generated shape.

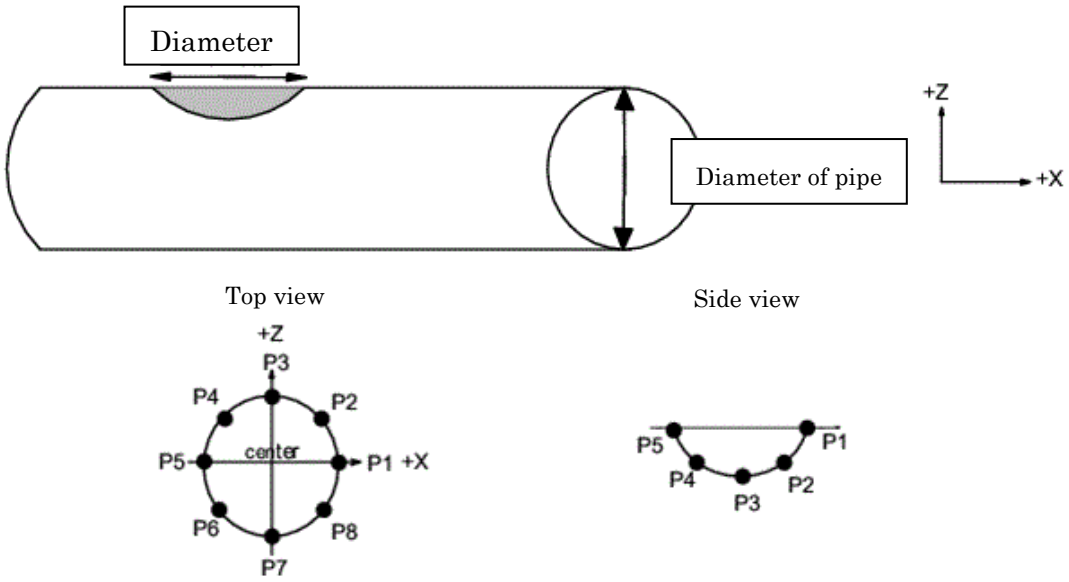


Fig. 43.2.8 Pommel shape setup

43.2.9 Custom

Selecting a CUSTOM shape allows you to create a shape program using any shape template.
Perform the following steps to create a shape program for a custom shape.

Step

1. Define the process path.
   A process path is position data in the X, Y, Z coordinate system. This path must have a reference position, a process start position, and a process end position. It can also include interpolation points connecting each point, and, if necessary, a preliminary process at the start and overlap at the end.

2. Create a custom shape template program.
   Create a program based on the X, Y, Z coordinate system of the created machining path. Define each coordinate system data in the program's position data. It is not necessary to specify orientation data (W, P, R).
   Also, the robot TCP position does not need to be considered at this point. The robot TCP position is considered when creating the shape program.
   The template program can contain any TP instructions.

NOTE
1 The first point in the template program (the position data with the smallest line number) is the reference position (TCP at the time of creation) when creating the shape program. If you do not want to include the reference position in the process path, create a template program that skips the first point with a JMP LBL instruction.
2 Use FINE, CNT, and CR termination types. Other termination types cannot draw a shape with accuracy.
3 Use Linear and Circular motion types. Since orientation data (W,P,R) is not used, set these items to 0.
4 You can define up to 64 points in the path. The number of program lines is unlimited.

> ⚠ **WARNING**
> The template program does not consider the robot's position data. Therefore, do not execute the template program or directly modify the position data (considering the robot position) on the edit screen

For example, define a rhomboid shape with P1 as the reference position as shown in Fig. 43.2.9(a). The units are mm.

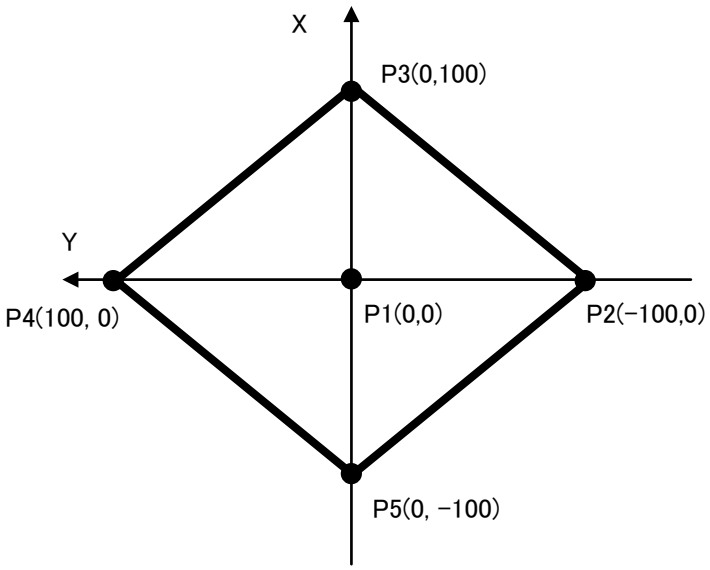


Fig. 43.2.9(a) Template shape example

For such a shape, refer to the following as an example of creating a simple template program.

When passing through the reference position	When not passing through the reference position
1: L P[1] 100mm/sec FINE ; 2: L P[2] 100mm/sec FINE ; 3: CALL PROC_ON ; 4: L P[3] 100mm/sec FINE 5: L P[4] 100mm/sec FINE ; 6: L P[5] 100mm/sec FINE ; 7: CALL PROC_OFF ;	1: JMP LBL[100] 2: L P[1] 100mm/sec FINE ; 3: LBL[100] 4: L P[2] 100mm/sec FINE ; 5: CALL PROC_ON ; 6: L P[3] 100mm/sec FINE 7: L P[4] 100mm/sec FINE ; 8: L P[5] 100mm/sec FINE ; 9: CALL PROC_OFF ;

Position No.	Position Data(X,Y,Z)
P[1]	X = 0.000 mm, Y = 0.000 mm, Z = 0.000 mm,
P[2]	X = 0.000 mm, Y = -100.000 mm, Z = 0.000 mm
P[3]	X = 100.000 mm, Y = 0.000 mm, Z = 0.000 mm,
P[4]	X = 0.000 mm, Y = 100.000 mm, Z = 0.000 mm,
P[5]	X = -100.000 mm, Y = 0.000 mm, Z = 0.000 mm,

Fig.43.2.9(b) Template shape example program

3. Set up custom shape detail data.
   Table.43.2.9 lists specific setting items for creating custom shapes.

Table 43.2.9 CUSTOM shape setup items

ITEM	DESCRIPTION
Template name	This item specifies the template program name to be used.
X Work angle	This item specifies the bias angle of the X axis for diagonal cutting.
Y Work angle	This item specifies the bias angle of the Y axis for diagonal cutting.
Rotation	This item specifies the rotation angle of shapes about the TCP.

43.3 SHAPE GENERATION UTILITY

43.3.1 UTILITIES Shape Settings

You specify detailed Shape Generation settings on the UTILITIES Shape screen. The default values of the items on the UTILITIES Shape screen is the content set on SETUP Shape screen. Changes made on the UTILITIES Shape screen take precedence over the settings made on the SETUP Shape screen.

NOTE
Changes made on the UTILITIES Shape screen do not affect the default values.

In addition to the setting items on the SETUP Shape screen (common settings + shape-specific settings) the UTILITIES Shape screen includes the two items listed in Table 43.3.1. Refer to Table 43.2.2 through Table 43.2.9 for the common settings and shape-specific settings.

Table 43.3.1 Additional UTILITIES Shape settings

ITEM	DESCRIPTION
Create or Edit Program	This item allows you to create a new program name for the shape program that will be created. At this item you can also select an existing program to edit. Press F4, [CHOICE] to display a list of existing programs.
Center position	This item appears near the end of the table. It indicates whether a center position has been recorded. For an existing program, you can touch up the position and regenerate the program by pressing F4, TOUCHUP. And you can also move the robot to the recorded center position by pressing F5, MOVETO.

NOTE
In addition to the settings you enter on the UTILITIES Shape DETAIL screens, the system will use the following information to generate a shape program:
- Currently selected jog GROUP (for multi-group systems)
- Current TOOL frame of the selected group
- Current USER frame of the selected group
- The location of the TCP of the selected GROUP. This will be the center point.

43.3.2 Creating and Editing a Shape program

Use the following procedure to create a shape program on the UTILITIES Shape screen.

STEP
1 Press MENU.
2 Select UTILITIES.
3 Press F1, [TYPE].
4 Select Shape Gen and press ENTER. The following screen will be displayed:

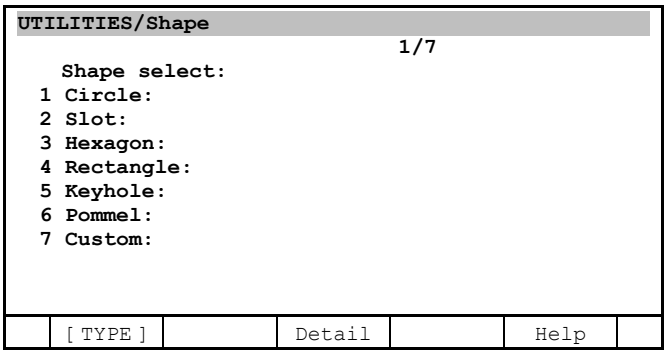


Fig.43.3.2(a) UTILITIES screen

5 Select the shape to create or modify and press F3, DETAIL.
6 Create a new program, select an existing program to edit, or copy an existing program:

To create a new program:

a. Move the cursor to Create or Edit Program and press ENTER.
b. Type a name for the shape program that will be created and press ENTER.

If all the default shape data is correct for the new program, go to Step 8 to generate the program.
To modify the default data for the new program, go to Step 7.

T**o edit an existing program**:

a. Move the cursor to Create or Edit Program and press F4, [CHOICE].
b. Select the name of the program to modify and press ENTER.

To copy an existing program:

a. Move the cursor to Create or Edit Program and press F4, [CHOICE].
b. Select the name of the program to modify and press ENTER.
The new program data will be copied from the existing one name.
c. Move the cursor to Create or Edit Program and enter the new program name.

To generate the new program from copied shape data, go directly to Step 8.
To generate the new program from the data modified from copied shape data, go to Step 7.

7. Modify shape data as desired. The modifications you make here will affect only the current shape. The default shape data will not change. Refer to Sections 43.2.1 – 43.2.9 for shape-specific settings.

8. **To generate the program** when all the information is correct, press CREATE. This will create the new shape program or re-create the existing edited program.

NOTE
When you press CREATE to generate a new program, the system will always use the current location of the group that is selected. However, if you are editing an existing program, the TCP that was recorded in the existing program will always be used. Confirm that the TCP is correct when it creates the shape program.

9. Add application and process-specific instructions to the PROC_ON#.TP and PROC_OFF#.TP programs if Process Copy/Paste is disabled

10. Test run the program:

a. Press the teach pendant Enabling Device (Deadman switch), set the ON/OFF switch to ON, and make the program ready to run.
b. To run the generated program, press and hold SHIFT and press F3, TEST.

NOTE
Single step execution is not supported from the TEST key. If you need to single step the program, go to the SELECT program screen, select the shape program you want to run, and run it in the same way as usual program execution.

If the shape is not exactly the way you want it to be, you can change the data in the screen, then press CREATE again to re-create that shape.

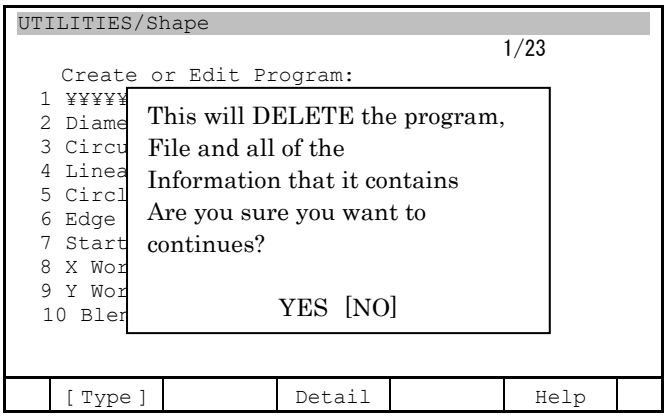


Fig.43.3.2(b) Message at program re-creation

⚠ CAUTION
Pressing CREATE again will completely erase the current program and re-create it with the same name, using the data you set in the UTILITIES Shape DETAIL screen. The instructions you enter in the PROC_ON and PROC_OFF instructions are never changed.
Any other modifications made to the program logic will be lost when you press CREATE.

44 MATERIAL HANDLING GRIPPER OPTION

The Material Handling Gripper option provides the controller with the ability to set up, configure, and control the functioning of vacuum or clamp-style material handling valves. You define the tooling valve type, characteristics, and I/O. Then you can control the tooling valve using specific macro commands that are called from within a teach pendant program, run from the Manual Functions screen, or run by pressing a teach pendant user key.

The following terms are used in the descriptions of setting up and operating the Material Handling Tool. Review them for a better understanding of the Material Handling Gripper option.

Tooling
This is a "gripper-like" mechanism controlled by valves.

Vacuum Tool
This is a mechanism that picks up objects using vacuum suction cups.

Clamp Tool
This is a mechanism that picks up objects using clamps.

Part Present
This is a sensor that indicates whether an object is being held by a tool.

Vacuum Input
This is a sensor that indicates whether the vacuum suction is on.

Valve n to A
This is a signal that indicates whether the tooling valve is in the A position. When the signal is on, the tooling valve is in the A position. The A and B positions must be opposite.

Valve n to B
This is a signal that indicates whether the tooling valve is in the B position. When the signal is on, the tooling valve is in the B position. The A and B positions must be opposite.

NOTE
Some descriptions in this chapter might apply only to the use of this option with the SpotTool+ product and not the HandlingTool product.

44.1 SETTING UP A MATERIAL HANDLING TOOL

You set up a Material Handling Tool by doing the following:

1 Select Handling Config during Controlled start (for SpotTool+ only).
2 Set up tooling valve information.
3 Set up tooling valve signals.
4 Configure tooling valve I/O.

After you have set up this information, you can also control valve I/O.

If you are familiar with the configuration of a Material Handling Tool, you can set up the gripper information using an mhgripcfg.xml file which you can import. This is not recommended for beginner users. This is described in Section 44.1.6.

44.1.1 Selecting the Handling Config (SpotTool+ only)

If you have SpotTool+, before you can set up and use a Material Handling Tool, you must do the following:

- Select and configure the material handling application at Controlled start - Procedure 44-1
- Select the material handling application as the current application for robot operation - Procedure 44-2

Procedure 44-1 Selecting and Configuring the Handling Config at Controlled Start

Steps

1 Perform a Controlled start.

a. Turn the power disconnect circuit breaker to ON.

If you are using a Paint Robot, also press the ON button.

b. Immediately press and hold the PREV and NEXT keys on the teach pendant.

c. Select Controlled start and press ENTER. You will see a screen similar to the following.

```
SPOT CONFIG                  CONTROLLED START MENUS
                                          1/50
 1 F Number:                          F00000 >
 2 Load Spottool Macros:              DISABLED
 3 Number of equipments:              1
 4 Number of weld controllers:        1
 5 Current equipment, gun             1, 1
 6   Weld interface:                  DIGITAL
 7   Number of guns:                  1
 8   Studwelder:                      DISABLED
 9     Enable Backup gun:             DISABLED
10 Tool changer gun:                  DISABLED
11 Soft touch gun:                    DISABLED
Press FCTN then START (COLD) when done
```

2 Select the application:

a. Press MENU.

b. Select Handling Config. You will see a screen similar to the following.

```
Appl selection CONTROLLED START MENUS  1/3
 1 Spot Weld          TRUE
 2 Handling           FALSE
 3 Dispense           FALSE
```

c. Move the cursor to Handling and press F4, TRUE. To disable an application, press F5, FALSE.

NOTE
- If you have the Multiple Application option in SpotTool+, you can enable up to three loaded applications on the controller.
- After you have enabled an application, you can use the enabled application instructions in a teach pendant program.

3 Configure the Handling application:

a. Press MENU.

b. Select Handling Config. You will see a screen similar to the following.

c. Set each item as desired, and press ENTER. Refer to Table 44.1.1 (a).

```
Handling Signals            CTRL START MENU
 1 Number of MH Tools:              1 <-- See Note
 2 Number of Valves per Tool:       4 <-- See Note
 3 Number of Clamp Inputs:          34
 4 Number of Part Present Inputs:   12
 5 Number of Vacuum Inputs:         4
 6 Use Active Tool for Arguments:   NO
```

NOTE
The combination of these two items is restricted by size and can be determined by multiplying one by the other. However, the resulting value cannot exceed 210. For example, 29 tools by 7 valves (203) would be the limit for that setup. As a second example, 22 tools by 9 valves (198) would be the limit for that setup.

Table 44.1.1 (a) Handling Config Menu Options

Item	Description
Number of MH Tools	Use this item to configure the number of tools you need for your system. The maximum number of tools is 29. At cold start, you will then have the specified number of tools available in the MH Valves menu under Setup.
Number of Valves Per Tool	Use this item to configure the number of valves you have in your tool. The maximum number of valves is 29. At cold start, you will then have the specified number of valves available in the MH Valves menu under Setup. You can then configure any valve to be a vacuum or clamp style tool and set up the required I/O for each.
Number of Clamps	Use this item to configure the number of clamps you need for your tool. The maximum number of clamps is 64. At Cold start, you will then have the specified number of clamps available in the MH Valves menu under Setup to assign to a valve.
Number of Part Present Inputs	Use this item to configure the number of Part Present Inputs you need for your tool. The maximum number of Part Present Inputs is 29. At cold start, you will then have the specified number of Part Present Inputs available in the MH Valves menu under Setup to assign to a valve.
Number of Vacuum Inputs	Use this item to configure the number of Vacuum Inputs you need for your tool. The maximum number of Vacuum Inputs is 29. At cold start, you will then have the specified number of Vacuum Inputs available in the MH Valves menu under Setup to assign to a valve.

Item	Description
Use Active Tool for Arguments	Use this item to change the behavior of the tool macros when no parameters are passed to them. When this is set to YES, the number of valves set in the tool SETUP are used to operate that number of valves for that tool. When set to NO, they act like the original specification.

Procedure 44-2 Selecting the Current Application for Robot Operation

Conditions

- The applications you want have been enabled at controlled start. Refer to the *Spot Welding Function OPERATOR'S MANUAL* (B-84464EN-4) for more information.

Steps

1 Cold start the controller.
   a. Press FCTN and select CYCLE POWER. Select YES, and press ENTER.
   b. When the BMON screen is displayed on the teach pendant, press and hold the SHIFT and RESET keys. Or, on the operator panel, press and hold RESET.
   c. Release all of the keys.
2 Press the FCTN key.
3 Select CHANGE APPL-TOOL.
The next enabled application will be selected. The currently selected application is displayed in the middle of the screen name line on the teach pendant screen, as shown in Figure 44.1.1 (a).
4 Repeat Step 2 and Step 3 until the material handling application is selected.

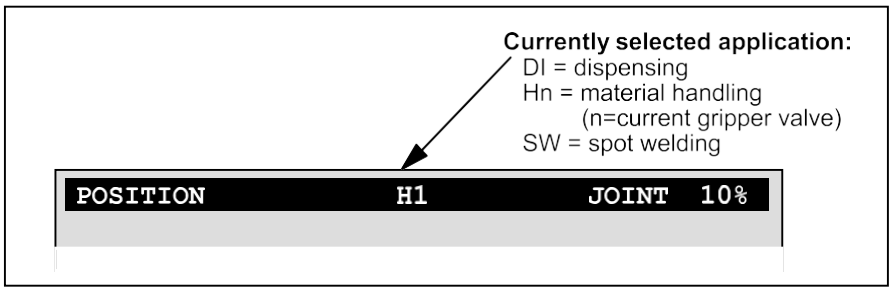


Fig.44.1.1 (a) Currently Selected Application

44.1.2 Setting Up Tooling Valves

You can use from 1 to x Tooling valves, depending on how many you enabled at Controlled start (Procedure 44-1). Use Procedure 44-3 to set up tool valves.

NOTE
You cannot change any of the tooling valve setup information when the controller is in AUTO mode.

Procedure 44-3 Setting Up an MH Tooling Valve

Steps

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

```
SETUP Mat.Handling
       Select tool to setup        1/1
       TOOL   VALVES NAME
        1       1    TOOL 1
```

5 Select the Tool you are using, and press ENTER. You will see a screen similar to the following.

```
SETUP Mat.Handling
Select valve to setup Tool: 1       1/3
VALVE  TYPE          VALVE NAME
 1     CLAMP         DOOR RING
 2     VACUUM        HINGE
 3     *****         VALVE3
```

6 Move the cursor to the valve you want to set up and press F2, DETAIL.

Table 44.1.2 (a) Valve DETAIL Screen Items

Item	Description
Vacuum Valve	
Valve #n	This item is the name of the valve you are setting up.
Valve type	This item specifies the type of valve, which is Vacuum.
Vac Feedback Default: NO	This item indicates whether the vacuum valve has a feedback sensor. • YES means the vacuum valve has a feedback sensor. • NO means the vacuum valve does not have a feedback sensor.
Ignore Feedback Min: 0 ms Max: 10000 ms Default: 750 ms	When Vac Feedback is YES, this item specifies the amount of time to wait after the vacuum was turned on to start checking whether feedback has been received.
Continuous Check Default: NO	When Vac Feedback is YES, this item specifies whether to check that feedback stays on until the vacuum is turned off. • YES checks that feedback stays on until the vacuum is turned off. • NO does not check that feedback stays on until the vacuum is turned off.
Max Feedback Delay Min: 0 ms Max: 10000 ms Default: 750 ms	When Vac Feedback is YES, this item specifies the amount of time to wait before giving an alarm if no feedback has been received. When Vac Feedback is NO, this item specifies the amount of time to wait after turning on or off the vacuum to make sure the vacuum is in the desired state before continuing.
Blow off time Min: 0 ms Max: 10000 ms Default: 1000 ms	When releasing a part manually, this item specifies the amount of time to pulse the vacuum valve to position B.
Recovery Blow off time Min: 0 ms Max: 10000 ms Default: 1000 ms	When releasing a part as part of error recovery, this specifies the amount of time to pulse the vacuum valve to position B.

Item	Description
No. of parts present Min: 0 Max: 6 Default: 0	This item specifies the number of parts present sensors used by this valve. **Note** If you change the number of parts present, the Material Handling Valve I/O setup will be changed. Increasing the number of parts present will add an entry with default data on the tool's I/O screen. The existing data will be left unchanged. The newly added data will remain unchanged. Decreasing the number of parts present will remove the last entry for that data type. The rest of the data will remain unchanged.
Continuous Check Default: NO	After a part has been picked up, this item specifies whether to check that the part present sensor stays on until the part is put back down. • YES checks that part present sensor stays on until a picked up part is put back down. • NO does not check that the part present sensor stays on until the part is put back down.
Toggle Retry Enabled (Grip) Default: FALSE	When this item is enabled, and the vacuum made state input is not sensed, the Alarm Recovery menu is displayed and provides a choice “Toggle I/O and recheck”. When this recovery choice is selected, the vacuum cups will toggle off, delay and then re-close. If the vacuum made sensor is in the proper state after toggling, the cycle will continue, if it is not, the alarm recovery menu will return.
Auto Retry (Grip) Default = 0 Min = 0 Max = 5	When the Toggle Retry (Grip) is enabled, then you can select the number of auto retries for toggling the vacuum cups automatically before the alarm recovery is executed when the vacuum made input is not being sensed. If the toggle retry (Grip) is DISABLED, you cannot edit the Auto Retry (Grip) value.
Toggle Retry Enabled (Release)	When this item is enabled, and the vacuum made input is sensed when it should not be, the Alarm Recovery menu is displayed and provides a choice “Toggle I/O and recheck”. When this recovery choice is selected, the vacuum cup will toggle on, delay and then toggle back off. If the vacuum made input is in the proper state after toggling, the cycle will continue, if it is not, the alarm recovery menu will return.
Auto Retry (Release) Default = 0 Min = 0 Max = 5	When the Toggle Retry (Release) is enabled, then you can select the number of auto retries for toggling the vacuum cups automatically before the alarm recovery is executed when the vacuum made input is being sensed when it should not be. If the toggle retry (Release) is DISABLED, you cannot edit the Auto Retry (Release) value.
Cancel Recover Option (Grip) Default = FALSE.	When Cancel Recover (Grip) is TRUE, then during alarm recovery for the vacuum sensor alarms, a choice is presented to the user to “Cancel and Recover”. If this choice is used, then it cancels the current operation and reverses it. For example, on a “grip” fault, the robot will “ungrip”. To use this option, the recovery path must be programmed in the pick or drop program. When this option is selected, all valves that were closed with the “Vacuum On (x,y,z) macro are attempted to be reversed to starting state. If the valves all have the opposing Cancel Recovery enabled, then all valves will be opened and confirmed to be in the open state as long as the “Check Opened” item is also enabled. The Flag[40] is set to 1, this is to be used by the user for indicating that the program must use the user’s programmed recovery to exit or back away. If the “Vacuum On” macro has multiple vacuum outputs, it is recommended that ALL outputs have the same setup, or exit recovery may not be able to execute. For instance, if a “Vacuum On (1,2,4)” is executed, vacuum fault occurs on Vacuum gripper #4. When you select Cancel and Recover, the vacuum outputs are toggled off in the opposing order they were turned on and checked to proper state (if the sensor checking is enabled). If any of the vacuums have the Toggle option DISABLED the recovery will write a message to the top of the teach pendant “'Gripper# x Vacuum# y Toggle is DISABLED'”

Item	Description
Cancel Recover Option (Grip) Default = FALSE	delay 2 seconds then, post an error message 'Open Grip to Exit NOT Possible' indicating that the recovery is not available to be completed due to the current setup of all valves are not allowed to exit recovery. The flag 40 will be set to zero in this case. It is recommended that you use the Auto Backward Exit option (R861) when using the Cancel Recovery. This option allows the robot's exit path to be recorded. with minimal Path programming. See example program tmptpick.tp.
Cancel Recover Option (Release) Default = FALSE.	When Cancel Recover (Release) is TRUE, then during alarm recovery for the Vacuum alarm, a choice is presented to the user to "Cancel and Recover". If this choice is used, then it cancels the current operation and reverses it. For example, on a "vacuum off" fault, the robot will return to the vacuum on state. To use this option, the recovery path must be programmed in the pick or drop program. When this option is selected, all valves that were opened with the "Vacuum off (x,y,z) macro are attempted to be reversed to starting state. If the valves all have the opposing Cancel Recovery enabled, then all vacuums will be turned on and confirmed to be in the vacuum made state as long as the "Check Closed" item is also enabled. The Flag[40] is set to 1, this is to be used by the user for indicating that the program must use the users programmed recovery to exit or back away. If the "Vacuum off" macro has multiple vacuum grippers, it is recommended that ALL vacuum grippers have the same setup, or exit recovery may not be able to execute. For example "Vacuum off (1,2,4)" is executed and a vacuum fault occurs on vacuum gripper #4. When the user selects Cancel and Recover, the vacuums are turned back on in the opposing order they were turned off and checked to proper state (if the sensor checking is enabled). If any of the vacuum grippers have the Toggle option DISABLED the recovery will write a message to the top of the teach pendant "'Gripper# x Vacuum# y Toggle is DISABLED'" delay 2 seconds then, post an error message Close Grip to Exit NOT Possible' indicating that the recovery is not available to be completed due to the current setup of all valves are not allowed to exit recovery. The flag 40 will be set to zero in this case. It is recommended that you use the Auto Backward Exit option (R861) option when using the Cancel Recovery. This option allows the robot's exit path to be recorded with minimal Path programming. See example program tmptpick.tp.
Clamp Valve	
Valve #n	This item is the name of the valve you are setting up.
Valve type	This item specifies the type of valve, which is CLAMP.
No. of clamps Min: 0 Max: 7 Default: 0	This item specifies the number of clamp sensor pairs used by this clamp valve. **Note** If you change the number of clamps, the Material Handling Valve I/O setup will be changed. Increasing the number of clamps will add an entry with default data on the tool's I/O screen. The existing data will be left unchanged. The newly added data will remain unchanged. Decreasing the number of clamps will remove the last entry for that data type. The rest of the data will remain unchanged.
Check Opened Default: NO	Before a part has been picked up and after a part has been put down, this item specifies whether to automatically check that the valve's clamps are all open.
Check Closed Default: NO	After a part has been picked up, this item specifies whether to automatically check that the valve's clamps are closed.
Operation Timeout Min: 0 ms Max: 9999 ms Default: 750 ms	The Operation Timeout description is the time in milliseconds the sensor is checked before posting a fault. As soon as the sensor is checked and in a valid state, the Over Stroke Delay is executed.

Item	Description
Over Stroke Delay Min: 0 ms Max: 9999 ms	After the close sensor becomes active, the gripper will wait this amount of time in milliseconds and check the sensor again. This is used when a gripper may go past the "closed" switch to an overstroke condition if there is not a part present.
No. of parts present Min: 0 Max: 6 Default: 0	This item specifies the number of parts present sensors used by this valve. **Note** If you change the number of parts present, the Material Handling Valve I/O setup will be changed. Increasing the number of parts present will add an entry with default data on the tool's I/O screen. The existing data will be left unchanged. The newly added data will remain unchanged. Decreasing the number of parts present will remove the last entry for that data type. The rest of the data will remain unchanged.
Continuous Check Default: NO	After a part has been picked up, this item specifies whether to check that the part present sensor stays on until the part is put back down. • YES checks that part present sensor stays on until a picked up part is put back down. • NO does not check that the part present sensor stays on until the part is put back down.
Toggle Retry Enabled (Grip) Default: FALSE	When this item is enabled, and the clamp closed state input is not sensed, the Alarm Recovery menu is displayed and provides a choice "Toggle I/O and recheck". When this recovery choice is selected, the clamp will toggle through open, delay and then re-close. If the clamp is in the proper state after toggling, the cycle will continue, if it is not, the alarm recovery menu will return.
Auto Retry (Grip) Default = 0 Min = 0 Max = 5	When the Toggle Retry (Grip) is enabled, then you can select the number of auto retries for toggling the clamp automatically before the alarm recovery is executed when the Clamp Closed input is not being sensed. If the toggle retry (Grip) is DISABLED, you cannot edit the new Auto Retry (Grip) value.
Toggle Retry Enabled (Release)	When this item is enabled, and the clamp open state input is not sensed, the Alarm Recovery menu is displayed and provides a choice "Toggle I/O and recheck". When this recovery choice is selected, the clamp will toggle through close, delay and then re-open. If the clamp is in the proper state after toggling, the cycle will continue, if it is not, the alarm recovery menu will return.
Auto Retry (Release) Default = 0 Min = 0 Max = 5	When the Toggle Retry (Release) is enabled, then you can select the number of auto retries automatically before the alarm recovery is executed when the Clamp Open input is not being sensed. If the toggle retry (Release) is DISABLED, you cannot edit the new Auto Retry (Release) value.
Cancel Recover Option (Grip) Default = FALSE.	When Cancel Recover (Grip) is TRUE, then during alarm recovery for the Clamp close alarm, a choice is presented to "Cancel and Recover". If this choice is used, then it cancels the current operation and reverses it. For example, on a "grip" fault, the robot will "ungrip". To use this option, the recovery path must be programmed in the pick or drop program. When this option is selected, all valves that were closed with the "Grip Part (x,y,z) macro are attempted to be reversed to starting state. If the valves all have the opposing Cancel Recovery enabled, then all clamps will be opened and confirmed to be in the open state as long as the "Check Opened" item is also enabled. The Flag[40] is set to 1, this is to be used by the user for indicating that the program must use the user's programmed recovery to exit or back away. If the "Grip Part" macro has multiple clamps, it is recommended that ALL clamps have the same setup, or exit recovery may not be able to execute. For instance, if a "Grip Part (1,2,4)" is executed, clamp fault occurs on clamp#4. When the user selects Cancel and Recover, the clamps are opened in the opposing order they were closed and checked to proper state (if the sensor checking is enabled). If any of the clamps closed have the Toggle option DISABLED. The recovery will write a message to the top of the teach pendant "'Gripper# x Clamp# y Toggle is DISABLED'" delay 2 seconds then, post

Item	Description
Cancel Recover Option (Grip) Default = FALSE.	an error message 'Open Grip to Exit NOT Possible' indicating that the recovery is not available to be completed due to the current setup of all valves are not allowed to exit recovery. The flag 40 will be set to zero in this case. It is recommended that you use the Auto Backward Exit option (R861 when using the Cancel Recovery. This option allows the robot's exit path to be recorded. with minimal Path programming. See example program tmptpick.tp.
Cancel Recover Option (Release) Default = FALSE.	When Cancel Recover (Release) is TRUE, then during alarm recovery for the Clamp Open alarm, a choice is presented to the user to "Cancel and Recover". If this choice is used, then it cancels the current operation and reverses it. For example, on a "ungrip" fault, the robot will "grip". To use this option, the recovery path must be programmed in the pick or drop program. When this option is selected, all valves that were opened with the "Release Part(x,y,z) macro are attempted to be reversed to starting state. If the valves all have the opposing Cancel Recovery enabled, then all clamps will be closed and confirmed to be in the close state as long as the "Check Closed" item is also enabled. The Flag[40] is set to 1, this is to be used by the user for indicating that the program must use the users programmed recovery to exit or back away. If the "Grip Part" macro has multiple clamps, it is recommended that ALL clamps have the same setup, or exit recovery may not be able to execute. For instance, if a "Release Part (1,2,4)" is executed, clamp fault occurs on clamp#4. When you select Cancel and Recover, the clamps are closed in the opposing order they were opened and checked to proper state (if the sensor checking is enabled). If any of the clamps closed have the Toggle option DISABLED. The recovery will write a message to the top of the teach pendant "'Gripper# x Clamp# y Toggle is DISABLED'" delay 2 seconds then, post an error message Close Grip to Exit NOT Possible' indicating that the recovery is not available to be completed due to the current setup of all valves are not allowed to exit recovery. The flag 40 will be set to zero in this case. It is recommended that you use the Auto Backward Exit option (R861) option when using the Cancel Recovery. This option allows the robot's exit path to be recorded with minimal Path programming. See example program tmptpick.tp.
Undefined Valve (*****)	
Valve #n	This item is the name of the valve you are setting up.
Valve [TYPE]	This item specifies the type of valve, which is NONE.

If you selected a vacuum type valve, you will see a screen similar to the following.

```
SETUP Mat. Handling
Setup Vacuum Information    Tool: 1       1/17

 1 Valve #1:                [      VALVE 1]
 2 Valve Type:                      Vacuum
 3 No. of Vacuum Sensor:            8
 4 Vac Feedback:                    NO
 5  Ignore Feedback:                750 (ms)
 6  Continuous Check:               NO
 7 Max Feedback Delay:              750 (ms)
 8 Blow off time:                   1000 (ms)
 9 Recovery Blowoff Time:           1000 (ms)
10 No. of parts present:            4
11 Continuous Check:                NO
- Toggle Retry Enabled(Grip):      YES
13 Auto Retry (Grip):               3
- Toggle Retry Enabled(Release):   YES
15 Auto Retry(Release):             3
16 Cancel/Recover Option(Grip):    NO
17 Cancel/Recover Option(Release): NO
```

If you select a clamp type valve, you will see a screen similar to the following.

```
SETUP Mat. Handling
Setup Clamp Information                  1/15

 1 Valve #1:          [      DOOR   RING]
 2 Valve Type:                      Clamp
 3 No. of clamps:                   3
 4 Check Opened:                    YES
 5 Check Closed:                    YES
 6  Operation Timeout:              750 (ms)
 7  Over Stroke Delay:              750 (ms)
 8 No. of parts present:            0
 9  Continuous Check:               NO
- Toggle Retry Enabled(Grip):      YES
11  Auto Retry (Grip):                    2
- Toggle Retry Enabled(Release):   YES
13  Auto Retry (Release):           2
14 Cancel/Recover Option(Grip):    NO
15 Cancel/Recover Option(Release): NO
```

If you select a ***** (NONE) type valve, you will see a screen similar to the following.

```
SETUP Mat. Handling
Setup Valve Information            1/2
 1 Valve #3          [   VALVE 3]
 2 Valve Type:              None
```

NOTE
You cannot change any of the valve setup information when the controller is in AUTO mode.

7 Move the cursor to each item and set as desired.

NOTE
If you change the number of clamps or number of parts present, the Config setup will be changed when you press APPLY, as follows:
- Increasing the number of clamps or number of parts present will add an entry with default data on the valve's I/O screen. The existing data will be left unchanged. The newly added data will remain unchanged.
- Decreasing the number of clamps or number of parts present will remove the last entry for that data type. The rest of the data will remain unchanged.

8 When you are finished making changes to the setup information and want to update the settings, press F3, APPLY.

9 To display detail information for another valve, press F2, VALVE#, type the number of the valve you want, and press ENTER.

10 To save all of the MH valve settings to the MHGRIPDT.VR file on the default device,
- To continue the save, select YES and press ENTER.
- To cancel the save, select NO and press ENTER.

NOTE
If the file MHGRIPDT.VR file already exists on the default device, you will be asked whether to overwrite the file.

- To overwrite the MHGRIPDT.VR file on the default device, select YES and press ENTER.
- To cancel the save, select NO and press ENTER.

11 To restore and MHGRIPDT.VR file from the default device to controller memory (load will overwrite the current MHGRIPDT.VR file in controller memory),

a. Press NEXT, >, and then press F2, LOAD.

b. Confirm the restoration of the MHGRIPDT.VR file:
   - To continue the restoration, select YES and press ENTER.
   - To cancel the restoration, select NO and press ENTER.

12 To set up the signals used by the tooling valves, refer to Section 44.1.3.

44.1.3 Setting Up Tooling Valve Signals

You need to set up the following valve signals to use the Material Handling Tool:

- Clamp inputs
- Part present inputs
- Vacuum inputs
- Valve outputs

The type of signals you need to set up depends on the type of valve you have and the number of valves you defined at Controlled start.

Table 44.1.3 (a) and Table 44.1.3 (b) list these I/O signals. Use Procedure 44-4 to set up valve signals.

Table 44.1.3 (a) Valve Signals

I/O Signal	Description
Clamp inputs - signals referred to by name on the I/O Mat. Handling screens	
Clamp n open n = 1 to 64	This signal indicates that the specified clamp n is open. You can give the signal any 16-character name. The signal can be either an RI or DI.
Clamp n closed n = 1 to 64	This signal indicates that the specified clamp n is closed. You can give the signal any 16-character name. The signal can be either an RI or DI.
Part present (PP) inputs - signals referred to by name on the I/O Mat. Handling screens	
Part present n = 1 to 29	This signal indicates that the part present sensor is on. You can give the signal any 16-character name. The signal can be either an RI or DI.
Vacuum (VacMade) inputs - signals referred to by name on the I/O Mat. Handling screens	
Vacuum n made n = 1 to 29	This signal indicates that the vacuum has been made for the specified tool n. n is any of the tools that were defined at Controlled start. You can give the signal any 16-character name.
Valve outputs - signals referred to by name on the I/O Mat. Handling screens	
Valve n to A n = 1 to 29	This signal indicates that the valve is in the A position. You can give the signal any 16-character name.
Valve n to B n = 1 to 29	This signal indicates that the valve is in the B position. You can give the signal any 16-character name.

Table 44.1.3 (b) SETUP MH Tool Signal DETAIL Screen Items

Item	Description
ID #	This item is the number which is assigned to the signal.
Input or Output Signal	This item represents the signal being used.
Type	This item represents the type of digital input/output that is assigned.
Number	This item represents the digital input or output number.

Procedure 44-4 Setting Up Tooling Valve Signals

Conditions

- You have defined tools at Controlled Start. Refer to the *Spot Welding Function OPERATOR'S MANUAL* (B-84464EN-4).

Steps

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

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

```
SETUP Mat. Handling
Select valve to setup  Tool: 1      1/2
VALVE  TYPE          VALVE NAME
 1     CLAMP         DOOR RING
 2     VACUUM        HINGE
```

5 Press F3, SIGNALS. You will see a screen similar to the following.

```
SETUP MH CONFIG
       I/O TYPES
 1     Clamp Inputs
 2     Part Present Inputs
 3     Vacuum Inputs
 4     Valve Outputs
```

6 Move the cursor to the type of I/O you want to set up and press F2, DETAIL.

For clamp inputs, you will see a screen similar to the following.

```
SETUP Mat. Handling
Clamp Signals for all valves        1/40
ID#    INPUT SIGNAL         TYPE #
 1     Clamp 1 open         DI[ 425]
 1     Clamp 1 closed       DI[ 426]
 2     Clamp 2 open         DI[ 427]
 2     Clamp 2 closed       DI[ 428]
 3     Clamp 3 open         DI[ 429]
 3     Clamp 3 closed       DI[ 430]
 4     Clamp 4 open         DI[   0]
 4     Clamp 4 closed       DI[   0]
 5     Clamp 5 open         DI[   0]
```

For part present inputs, you will see a screen similar to the following.

```
SETUP Mat. Handling
PP Signals for all valves          1/10
ID#    INPUT SIGNAL         TYPE #
 1     Part present 1       DI[ 437]
 2     Part present 2       DI[ 438]
 3     Part present 3       DI[ 439]
 4     Part present 4       DI[   0]
 5     Part present 5       DI[   0]
 6     Part present 6       DI[   0]
 7     Part present 7       DI[   0]
 8     Part present 8       DI[   0]
 9     Part present 9       DI[   0]
```

For vacuum inputs, you will see a screen similar to the following.

```
SETUP Mat. Handling
VacMade Inputs for all valves              1/2
ID#    INPUT SIGNAL         TYPE #
 1     VACUUM 1 MADE        DI[ 440]
 2     VACUUM 2 MADE        DI[   0]
```

For valve outputs, you will see a screen similar to the following.

```
SETUP Mat. Handling
Setup Valve Signals                1/20
ID#    OUTPUT SIGNAL        TYPE #
 1     Valve 1 to A         DO[ 429]
 1     Valve 1 to B         DO[ 430]
 2     Valve 2 to A         DO[ 425]
 2     Valve 2 to B         DO[ 426]
 3     Valve 3 to A         DO[   0]
 3     Valve 3 to B         DO[   0]
 4     Valve 4 to A         DO[   0]
 4     Valve 4 to B         DO[   0]
 5     Valve 5 to A         DO[   0]
```

7 Select the signals you want to set and set them as desired. You can change the signal name, type, and digital input or output number.
8 After you have set up the signals for each valve, you can configure the I/O for each valve. Refer to Section 44.1.4.

44.1.4 Configuring Tooling Valve I/O

When you configure tooling valve I/O, you specify which of the signals you set up in Section 44.1.3 will be used with each of the valves you set up in Section 44.1.2. Use Procedure 44-5 to configure valve I/O.

Procedure 44-5 Configuring Valve I/O

Steps

1 Press MENU.
2 Select SETUP.
3 Press F1, [TYPE].
4 Select MH Valves.
Move the cursor to the valve for which you want to configure I/O and press F2, DETAIL. You will see a screen similar to the following.

```
SETUP Mat. Handling
Setup Vacuum Information     Tool: 1       1/17

 3 Valve #1:                 [      VALVE 1]
 4 Valve Type:                      Vacuum
11 No. of Vacuum Sensor:            8
12 Vac Feedback:                    NO
13  Ignore Feedback:                750 (ms)
14  Continuous Check:               NO
15 Max Feedback Delay:              750 (ms)
16 Blow off time:                   1000 (ms)
17 Recovery Blowoff Time:           1000 (ms)
18 No. of parts present:            4
11 Continuous Check:                NO
- Toggle Retry Enabled(Grip):       YES
13 Auto Retry (Grip):               3
- Toggle Retry Enabled(Release):    YES
15 Auto Retry (Release):            3
16 Cancel/Recover Option(Grip):     NO
17 Cancel/Recover Option(Release):  NO
```

5 Press F5, CONFIG.

For an output signal, you will see a screen similar to the following.

```
SETUP MHandling
Valve 1:       DOOR RING
   TYPE        OUTPUT SIGNAL TYPE #
 1 Valve 1 to A              DO[  14]
 2 Valve 1 to B              DO[  15]
 Cannot change a valve's output signal
```

NOTE
The items on the output screen are for display only; you cannot make any changes on this screen.

For an input signal, you will see a screen similar to the following.

```
SETUP MHandling
Valve 1: DOOR RING
   TYPE       INPUT SIGNAL  TYPE #
 1 CLOPEN:Clamp 1 open       DI[   0]
 2 CLCLSD:Clamp 1 closed     DI[  11]
 3 CLOPEN:Clamp 20 open      DI[   0]
 4 CLCLSD:Clamp 20 closed    DI[   0]
 5 PARTPR:Part present 4     DI[  45]
 Select the signals used by this value
```

In this example screen, the clamp valve has two clamp pairs and one part present.

6 To select the input signal you want to configure,

a. Move the cursor to the input signal you want and press F4, [CHOICE].

A subwindow will be displayed that contains the signal names you set up in Section 44.1.3. See the following screen for an example.

```
SW
 1 Unassigned PP      5 Part Present 4
 2 Part present 1     6 Part Present 5
 3 Part present 2     7 Part Present 6
 4 Part present 3     8 -- NEXT --
 4 CLCLSD:    Clamp 20 closed      DI[  0]
 5 PARTPR:    Unassigned PP        DI[  0]
 Select the signals used by this valve
```

b. Select the appropriate input signal name.

The input TYPE and # columns will be filled in automatically based on the signal name you selected.

44.1.5 Controlling Valve I/O

After you have set up and configured valve I/O, you can monitor, simulate, and force I/O. Refer to Table 44.1.5 (a). Use Procedure 44-6 to control valve I/O.

Table 44.1.5 (a) I/O Mat. Handling Screen Items

Item	Description
VALVE#	Enter a valve number to view the I/O status for that valve.
IN/OUT	This item represents the signal being used. You can change the view between the Input and Output signals.
TYPE	This item represents the type of digital input/output that is assigned.
NUMBER (#)	This item represents the digital input or output number.
SIM/UNSIM	To simulate, press F4, SIMULATE. To unsimulate, press F5, UNSIM.
TOOL#	Enter the Tool number to choose a different tool.

Procedure 44-6 Controlling Valve I/O

Steps

1 Press MENU.
2 Select I/O.
3 Press F1, [TYPE].
4 Select MH Valves.

For clamp inputs, you will see a screen similar to the following.

```
I/O MatHandling DI
Valve 1: DOOR RING
   INPUT SIGNAL       TYPE #  SIM   STATUS
 1 Clamp 1 open       DI[ 425] U    ***
 2 Clamp 1 closed     DI[ 426] U    ***
 3 Clamp 2 open       DI[ 427] U    ***
 4 Clamp 2 closed     DI[ 428] U    ***
 5 Clamp 3 open       DI[ 429] U    ***
 6 Clamp 3 closed     DI[ 430] U    ***
 7 Part present 1     DI[ 437] U    ***
 8 Part present 2     DI[ 438] U    ***
```

For clamp outputs, you will see a screen similar to the following.

```
I/O MatHandling DO
Valve 1: DOOR RING
   OUTPUT SIGNAL      TYPE #  SIM   STATUS
 1 Valve 2 to A       DO[ 425] U    ***
 2 Valve 2 to B       DO[ 426] U    ***
```

For vacuum inputs, you will see a screen similar to the following.

```
I/O MatHandling DI
Valve 2: HINGE
   INPUT SIGNAL       TYPE #  SIM   STATUS
 1 Vacuum 1 made      DI[ 440] U    ***
 2 Part present 3     DI[ 439] U    ***
```

For vacuum outputs, you will see a screen similar to the following.

```
I/O MatHandling DO
Valve 2: HINGE
   OUTPUT SIGNAL      TYPE #   SIM    STATUS
 1 Valve 2 to A       DO[ 425] U      ***
 2 Valve 2 to B       DO[ 426] U      ***
```

5 To change the display between inputs and outputs, press F3, IN/OUT.
6 To display I/O information for another valve, press F2, VALVE#, type the number of the valve you want, and press ENTER.

NOTE
You cannot change the simulation of a signal, or turn the signal ON or OFF from the teach pendant, when the controller is in AUTO mode.

7 To simulate a signal,
   a. Move the cursor to the SIM column of the signal you want to simulate.
      - U means the signal is not simulated or unsimulated.
      - S means the signal is simulated.
   b. Simulate or unsimulate the signal.
      - To simulate, press F4, SIMULATE.
      - To unsimulate, press F5, UNSIM.
   c. To unsimulate all simulated signals press FCTN and then select UNSIM ALL I/O.

8 To turn a signal ON or OFF,
   a. Move the cursor to the STATUS column.

NOTE
If you want to turn ON or OFF an input signal, you must first simulate the input. If you want to turn ON or OFF an output signal, it can be either simulated or unsimulated.

   b. Turn ON or OFF the signal:
      - To turn ON the signal, press F4, ON.
      - To turn OFF the signal, press F5, OFF.

44.1.6 Importing and Exporting Material Handling Gripper xml

Starting with the R-50iA controller, the ability to import and export Material Handling Gripper configurations to or from an xml file is supported. The xml file must be named “mhgripcfg.xml” and it may be imported from any device the FILE menu supports. The file must be on the root directory of the device when imported, and is exported to the root directory when exported. For example, ud1:\mhgripcfg.xml when importing from ud1:\. You can set the device to import from or export to via the FILE menu (MENU → FILE) as shown in Fig.44.1.6 (a).

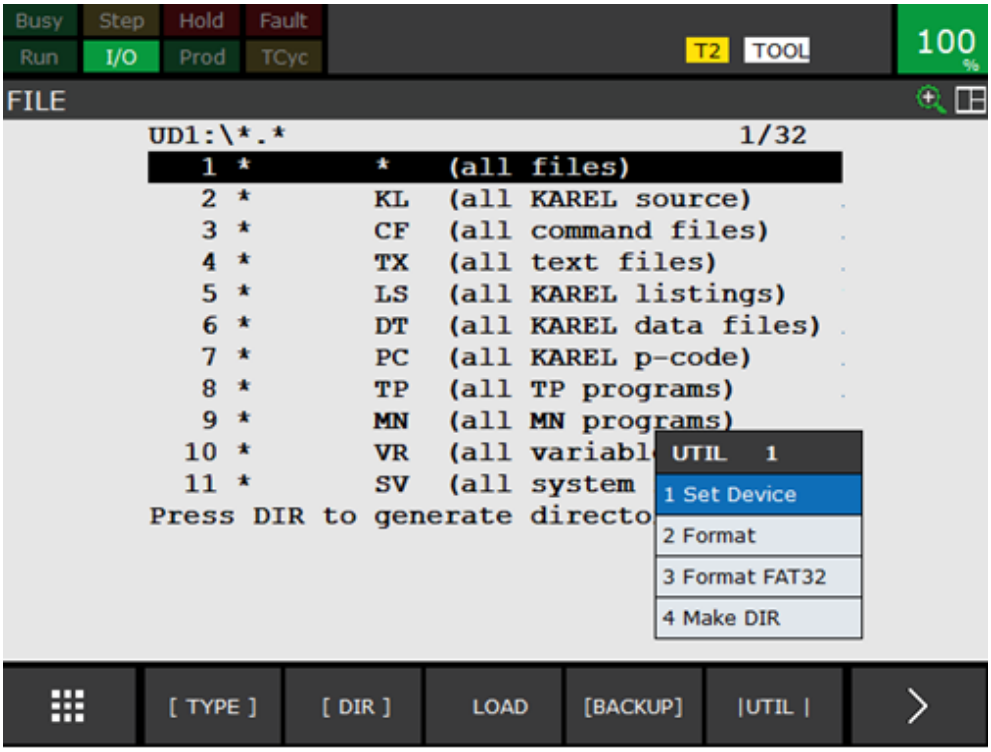


Fig.44.1.6 (a) Set Device

Once you have set the device from the FILE menu, you can import or export an mhgripcfg.xml file on that device by going to the MH Gripper setup menu (MENU → SETUP → MH Valves), then press F2, DETAIL to display the tool detail page, then press NEXT, >,to switch to the second set of function keys (F6-F10). Import and Export are the F9 and F10 keys. This screen is shown in Fig.44.1.6 (b).

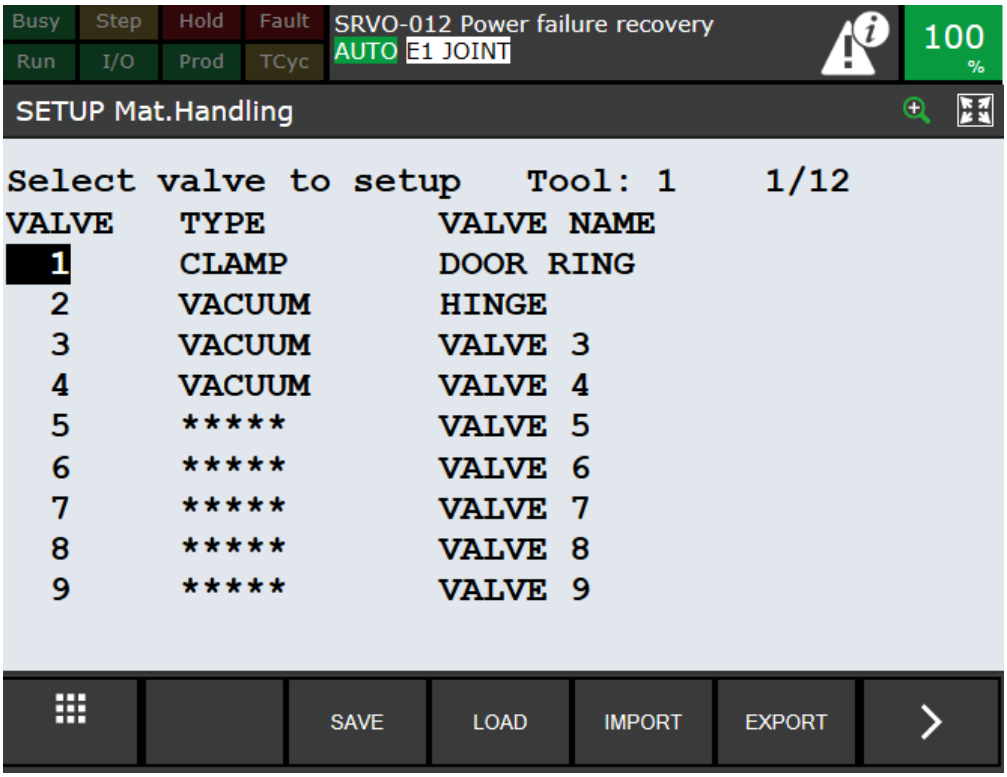


Fig.44.1.6 (b) Import/Export function keys

The mhgripcfg.xml file can completely set up all data associated with the MH Valve setup screens, including resizing data arrays. The data in the xml file is organized similarly to the screens that are normally used to set up the data on the teach pendant. The example configuration shows a setup that has four tools, six grippers per tool, up to six clamps per clamp type gripper, up to four part presence sensors per gripper, and up to six vacuum made inputs per vacuum gripper. It also defines which inputs and outputs are used for the gripper configuration. A value of 0 is used if that I/O value is not defined yet. If there are syntax errors, missing tags, missing name-value pairs, and so forth, an error will be posted when the file is imported. If there is an error in the imported xml file, the previous settings will be restored.

44.1.7 Example mhgripcfg.xml file

```
<Tool_max Total="4" />
<Gripper_max Total="6" Max_clamps="6" Max_Part_Present="4" Max_Vacuum_Inputs="6" />
<Global_Settings RetryActiveDO="0" Cancel_Recovery_Flag="0" Manual_Cycle_Confirm="NO"
UseActiveToolForArguments="NO" />
<Gripper_Inputs >
  <Part_Present Number="1" Name="Part present 1" Port_type="1" Port_Number="1" />
  <Part_Present Number="2" Name="Part present 2" Port_type="1" Port_Number="2" />
  <Part_Present Number="3" Name="Part present 3" Port_type="1" Port_Number="3" />
  <Part_Present Number="4" Name="Part present 4" Port_type="1" Port_Number="4" />
  <Clamp_InputsA Number="1" Name="Clamp 1A open" Port_type="1" Port_Number="5" />
  <Clamp_InputsB Number="1" Name="Clamp 1B closed" Port_type="1" Port_Number="6" />
  <Clamp_InputsA Number="2" Name="Clamp 2A open" Port_type="1" Port_Number="7" />
  <Clamp_InputsB Number="2" Name="Clamp 2B closed" Port_type="1" Port_Number="8" />
  <Clamp_InputsA Number="3" Name="Clamp 3A open" Port_type="1" Port_Number="9" />
  <Clamp_InputsB Number="3" Name="Clamp 3B closed" Port_type="1" Port_Number="10" />
  <Clamp_InputsA Number="4" Name="Clamp 4A open" Port_type="1" Port_Number="11" />
  <Clamp_InputsB Number="4" Name="Clamp 4B closed" Port_type="1" Port_Number="12" />
  <Clamp_InputsA Number="5" Name="Clamp 5A open" Port_type="1" Port_Number="13" />
  <Clamp_InputsB Number="5" Name="Clamp 5B closed" Port_type="1" Port_Number="14" />
  <Clamp_InputsA Number="6" Name="Clamp 6A open" Port_type="1" Port_Number="15" />
  <Clamp_InputsB Number="6" Name="Clamp 6B closed" Port_type="1" Port_Number="16" />
  <Vacuum_made Number="1" Name="VACUUM 1 MADE" Port_type="1" Port_Number="17" />
  <Vacuum_made Number="2" Name="VACUUM 2 MADE" Port_type="1" Port_Number="18" />
  <Vacuum_made Number="3" Name="VACUUM 3 MADE" Port_type="1" Port_Number="19" />
  <Vacuum_made Number="4" Name="VACUUM 4 MADE" Port_type="1" Port_Number="20" />
  <Vacuum_made Number="5" Name="VACUUM 5 MADE" Port_type="1" Port_Number="21" />
  <Vacuum_made Number="6" Name="VACUUM 6 MADE" Port_type="1" Port_Number="22" />
</Gripper_Inputs>
<Gripper_Outputs >
  <Valve_A Number="1" Name="Valve 1 to A" Port_type="2" Port_Number="1" />
  <Valve_B Number="1" Name="Valve 1 to B" Port_type="2" Port_Number="2" />
  <Valve_A Number="2" Name="Valve 2 to A" Port_type="2" Port_Number="3" />
  <Valve_B Number="2" Name="Valve 2 to B" Port_type="2" Port_Number="4" />
  <Valve_A Number="3" Name="Valve 3 to A" Port_type="2" Port_Number="5" />
  <Valve_B Number="3" Name="Valve 3 to B" Port_type="2" Port_Number="6" />
  <Valve_A Number="4" Name="Valve 4 to A" Port_type="2" Port_Number="7" />
  <Valve_B Number="4" Name="Valve 4 to B" Port_type="2" Port_Number="8" />
  <Valve_A Number="5" Name="Valve 5 to A" Port_type="2" Port_Number="9" />
  <Valve_B Number="5" Name="Valve 5 to B" Port_type="2" Port_Number="10" />
  <Valve_A Number="6" Name="Valve 6 to A" Port_type="2" Port_Number="11" />
  <Valve_B Number="6" Name="Valve 6 to B" Port_type="2" Port_Number="12" />
</Gripper_Outputs>
<Tool Name="TOOL 1" Tool_number="1" >
  <Grip_Info Number="1" Type="Clamp" Name="DOOR RING" >
    <Clamp_Info Num_Clamps="3" Check_Opened="YES" Check_Closed="NO"
Operation_Timeout_ms="1000" Overstroke_Delay="1000" Number_part_present="2"
Continuous_check="YES" >
      <Retry_config Toggle_Retry_grip_enable="NO" Grip_num_retries="0"
Toggle_retry_release_enable="NO" Release_num_retries="0" Canc_Recov_grip="NO"
Canc_Recov_Release="NO" />
      <Outputs >
        <Valve_numbers A="1" B="1" />
        <Valve_numbers A="2" B="2" />
        <Valve_numbers A="3" B="3" />
      </Outputs>
      <Inputs >
        <Part_present_number Number="1" />
```

```
        <Part_present_number Number="2" />
      </Inputs>
    </Clamp_Info>
  </Grip_Info>
  <Grip_Info Number="2" Type="Vacuum" Name="HINGE" >
    <Vacuum_Info Num_Sensors="2" Feedback="NO" Ignore_feedback_ms="750"
Continuous_check="YES" Max_Feedback_Delay_ms="1000" Blowoff_ms="1000"
Recovery_blowoff="1000" Number_part_present="1" Part_pres_Cont_check="NO" >
      <Retry_config Toggle_Retry_grip_enable="NO" Grip_num_retries="0"
Toggle_retry_release_enable="NO" Release_num_retries="0" Canc_Recov_grip="NO"
Canc_Recov_Release="NO" />
      <Inputs >
        <Vac_made_index Number="1" />
        <Vac_made_index Number="2" />
        <Part_present_number Number="3" />
      </Inputs>
    </Vacuum_Info>
  </Grip_Info>
  <Grip_Info Number="3" Type="Vacuum" Name="VALVE 3" >
    <Vacuum_Info Num_Sensors="1" Feedback="NO" Ignore_feedback_ms="750"
Continuous_check="NO" Max_Feedback_Delay_ms="750" Blowoff_ms="1000"
Recovery_blowoff="1000" Number_part_present="0" Part_pres_Cont_check="NO" >
      <Retry_config Toggle_Retry_grip_enable="NO" Grip_num_retries="0"
Toggle_retry_release_enable="NO" Release_num_retries="0" Canc_Recov_grip="NO"
Canc_Recov_Release="NO" />
      <Inputs >
        <Vac_made_index Number="1" />
      </Inputs>
    </Vacuum_Info>
  </Grip_Info>
  <Grip_Info Number="4" Type="Vacuum" Name="VALVE 4" >
    <Vacuum_Info Num_Sensors="1" Feedback="NO" Ignore_feedback_ms="750"
Continuous_check="NO" Max_Feedback_Delay_ms="750" Blowoff_ms="1000"
Recovery_blowoff="1000" Number_part_present="0" Part_pres_Cont_check="NO" >
      <Retry_config Toggle_Retry_grip_enable="NO" Grip_num_retries="0"
Toggle_retry_release_enable="NO" Release_num_retries="0" Canc_Recov_grip="NO"
Canc_Recov_Release="NO" />
      <Inputs >
        <Vac_made_index Number="1" />
      </Inputs>
    </Vacuum_Info>
  </Grip_Info>
</Tool>
<Tool Name="GRIP 2" Tool_number="2" >
  <Grip_Info Number="1" Type="Vacuum" Name="VALVE 1" >
    <Vacuum_Info Num_Sensors="1" Feedback="YES" Ignore_feedback_ms="750"
Continuous_check="NO" Max_Feedback_Delay_ms="750" Blowoff_ms="1000"
Recovery_blowoff="1200" Number_part_present="0" Part_pres_Cont_check="NO" >
      <Retry_config Toggle_Retry_grip_enable="NO" Grip_num_retries="0"
Toggle_retry_release_enable="NO" Release_num_retries="0" Canc_Recov_grip="NO"
Canc_Recov_Release="NO" />
      <Inputs >
        <Vac_made_index Number="1" />
      </Inputs>
    </Vacuum_Info>
  </Grip_Info>
  <Grip_Info Number="6" Type="Clamp" Name="VALVE 6" >
    <Clamp_Info Num_Clamps="2" Check_Opened="YES" Check_Closed="YES"
Operation_Timeout_ms="750" Overstroke_Delay="1200" Number_part_present="2"
Continuous_check="YES" >
      <Retry_config Toggle_Retry_grip_enable="YES" Grip_num_retries="2"
Toggle_retry_release_enable="YES" Release_num_retries="3" Canc_Recov_grip="YES"
Canc_Recov_Release="YES" />
      <Outputs >
        <Valve_numbers A="6" B="6" />
        <Valve_numbers A="7" B="7" />
      </Outputs>
      <Inputs >
        <Part_present_number Number="5" />
        <Part_present_number Number="6" />
      </Inputs>
    </Clamp_Info>
  </Grip_Info>
```

```
</Tool>
<Tool Name="GRIP 3" Tool_number="3" >
  <Grip_Info Number="2" Type="Clamp" Name="VALVE 2" >
    <Clamp_Info Num_Clamps="2" Check_Opened="YES" Check_Closed="YES"
Operation_Timeout_ms="750" Overstroke_Delay="1000" Number_part_present="2"
Continuous_check="YES" >
      <Retry_config Toggle_Retry_grip_enable="YES" Grip_num_retries="3"
Toggle_retry_release_enable="YES" Release_num_retries="1" Canc_Recov_grip="YES"
Canc_Recov_Release="YES" />
      <Outputs >
        <Valve_numbers A="6" B="6" />
        <Valve_numbers A="7" B="7" />
      </Outputs>
      <Inputs >
        <Part_present_number Number="1" />
        <Part_present_number Number="2" />
      </Inputs>
    </Clamp_Info>
  </Grip_Info>
</Tool>
<Tool Name="GRIP 4" Tool_number="-4" >
  <Grip_Info Number="1" Type="Clamp" Name="VALVE 1" >
    <Clamp_Info Num_Clamps="2" Check_Opened="YES" Check_Closed="YES"
Operation_Timeout_ms="750" Overstroke_Delay="0" Number_part_present="2"
Continuous_check="NO" >
      <Retry_config Toggle_Retry_grip_enable="NO" Grip_num_retries="0"
Toggle_retry_release_enable="NO" Release_num_retries="0" Canc_Recov_grip="NO"
Canc_Recov_Release="NO" />
      <Outputs >
        <Valve_numbers A="0" B="0" />
        <Valve_numbers A="0" B="0" />
      </Outputs>
      <Inputs >
        <Part_present_number Number="0" />
        <Part_present_number Number="0" />
      </Inputs>
    </Clamp_Info>
  </Grip_Info>
</Tool>
```

44.2 CONTROLLING A TOOLING VALVE IN MANUAL MODE

You can operate a tooling valve manually using teach pendant user keys when Handling is the currently selected application. The status screen in Figure 44.2 (a) shows how the user keys are used for HandlingTool.

```
Appl status          H1
   HandlingTool
      USERKEYS                    LEDS on TP
      U1: VALVE +                 L1: NO USE
      U2: VALVE ON/OFF            L2: NO USE
      U3: BLOWOFF TIMED ON        L3: NO USE
      U4: SETUP
      U5: STATUS
      U6: I/O
      U7: POSITION
```

Fig.44.2 (a) HandlingTool User Keys

Table 44.2 (a) lists and describes the manual operations you can perform using the user keys.

Table 44.2 (a) Handling Manual Operations

User Key	Description of Operation
Teach Pendant Enabled SHIFT + USER KEY 1	The next defined tooling valve becomes the currently selected valve. It is displayed on the teach pendant status line as "H," followed by a number that corresponds to the valve number.
Teach Pendant Enabled SHIFT + USER KEY 2	The state of the currently selected tooling valve is changed from position A to position B, or from position B to position A.
Teach Pendant Enabled SHIFT + USER KEY 3	If the currently selected tooling valve is a vacuum valve, the blowoff is timed on for the Blow off time specified on the SETUP Mat. Handling screen (Section 44.1.2).

44.3 MATERIAL HANDLING VALVE MACRO PROGRAMS

The macro programs listed in Table 44.3 (a) are provided with the Material Handling option. You can call these macros from other teach pendant programs or invoke them from the Manual Functions screen.

Table 44.3 (a) Material Handling Valve Macro Programs

Instruction Name	Program
Grip Part	GRIPPART
Release Part	DROPPART
Part Present	CHKPART
Check No Part	CHKNOPRT
Prepare to Pick	OKTOPICK
Clear to Proceed	CLR2PRCD
Turn ON Vacuum	VACUUMON
Turn OFF Vacuum	VACUUMOF
Turn OFF Blowoff	BLOWOFF
Set CurrentValve	SETVALVE
Set MH Tool	SET_TOOL

Note the following macro information:

- All of the macros are system macros.
- You set up all of the tooling valve control variables on the SETUP Mat. Handling screens (Section 44.1.2).
- You set the current tooling valve using SHIFT USERKEY 1, or using the Set CurrentValve macro. When a macro is called with no parameters, the current tooling valve is used by default.
- You can disable each MH PAUSE alarm using the Disable for 20 Cycles feature (Section 44.5.2).

NOTE

- There are no material handling program instructions; macros are used instead. You can combine material handling macros with either SpotTool+ application program instructions or DispenseTool application program instructions.
- Refer to section 44.3.1 – 44.3.12 for information on setting up and executing macros.

44.3.1 Examples

Teach Pendant Program with Material Handling Clamp Valve Macros shows an example teach pendant program that uses Material Handling clamp valve macros.

Teach Pendant Program with Material Handling Clamp Valve Macros

```
; Move above part 1
Prepare to Pick(1)      ** Check all clamps open for VALVE #1
; Move to part
Part Present(1)         ** Check part present #1 is ON
Grip Part(1)            ** Shift valve for VALVE #1 to A Position
; Move above part 2
Prepare to Pick(2)      ** Check all clamps open for VALVE #2
; Move to part
Part Present(3)         ** Check part present #3 is ON
Grip Part(2)            ** Shift valve for VALVE #2 to A Position
; Move to dropoff
Release Part(1,2)       ** Shift VALVEs #1 and #2 to B Position
; Move above release
Check No Part(1,2,3)    ** Check part presents #1, #2, and #3 to be OFF
; Move to home
```

44.3.2 Grip Part

Refer to Table 44.3.2 (a) and Table 44.3.2 (b) for details on the Grip Part macro.

Table 44.3.2 (a) Grip Part Macro

Item	Description
Program Name	GRIPPART.TP
Macro Name	Grip Part
KAREL Program	MHGRIPPT.KL
Description	This macro picks up parts using a CLAMP type valve. For each valve specified, the B side outputs are turned OFF and the A side outputs are turned ON. The macro program then waits for the clamp valve's "Operation Timeout" period. If the specified valves use different delays, the macro program will use the longest delay. If "Check Closed" is required, an alarm will alert you if any of the clamp valves failed to close.
Parameters	Up to nine parameters can be used. Each parameter is an INTEGER with a value from 1 to 16, which defines the CLAMP type valve to use. If no parameter is given, the macro program uses the current valve.
Errors	All errors will cause the program to display an error and pause: MHND-250 ABORT "Parm %s is not an integer" MHND-251 ABORT "Parm %s is not between 1-16" MHND-252 ABORT "Valve %s is not setup" MHND-246 ABORT "Valve %s is not a clamp gripper" MHND-256 PAUSE "Clamp %s state error" MHND-257 PAUSE "Clamp %s state error"
Program Logic	• Check parameters • Close requested CLAMP valves • Turn on A side valve and turn OFF B side valve • Wait CLAMP "Operation Timeout" period • If CLAMP "Check Closed" is YES • Check that each of the valve's Clamp open inputs are OFF and Clamp closed inputs are ON • Send an alarm if an error occurs

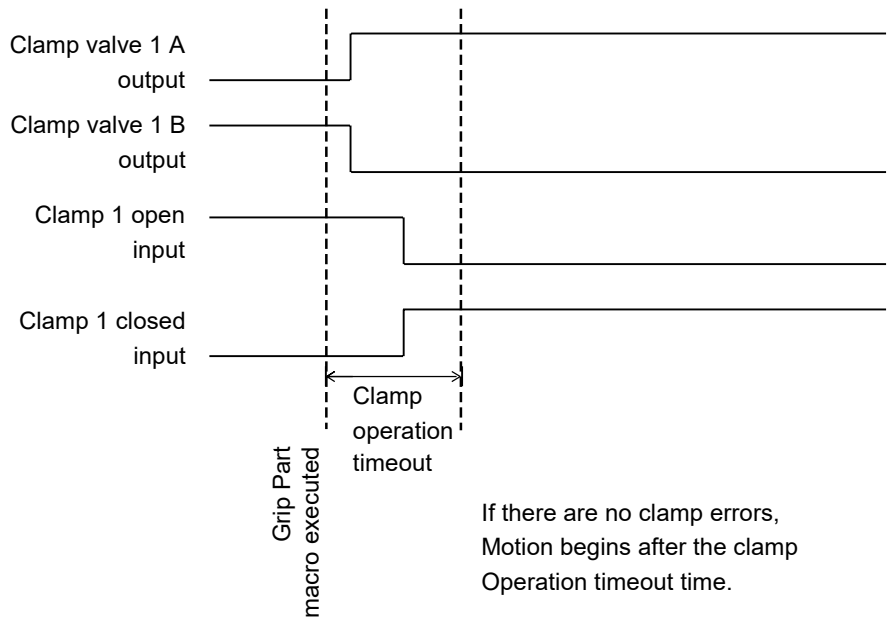
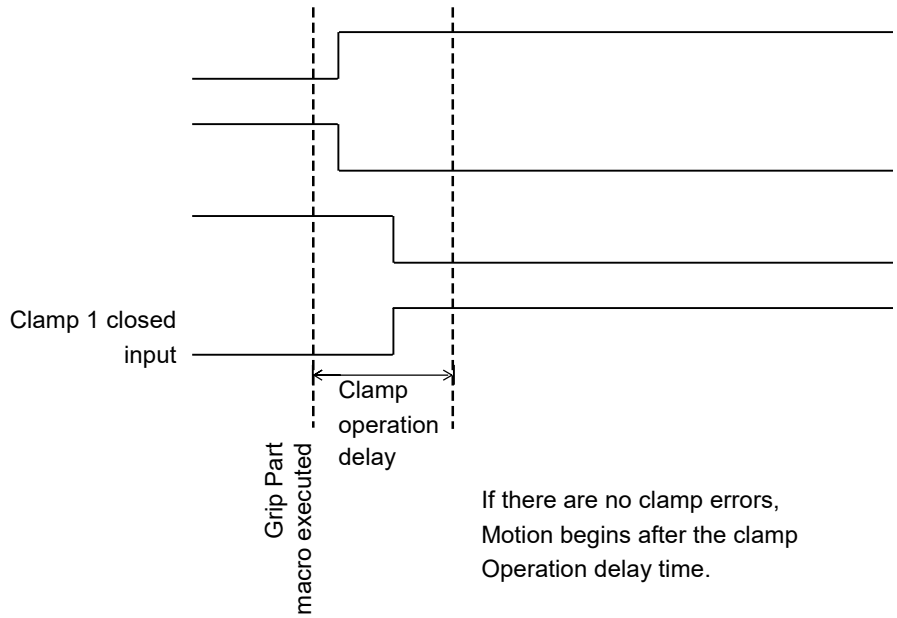
Item	Description
Timing Diagram	

Table 44.3.2 (b) Grip Part Macro

Item	Description
Program Name	GRIPPART.TP
Macro Name	Grip Part
KAREL Program	MHGRIPPT.KL
Description	This macro picks up parts using a CLAMP type valve. For each valve specified, the B side outputs are turned OFF and the A side outputs are turned ON. The macro program then waits for the clamp valve's "Operation Delay" period. If the specified valves use different delays, the macro program will use the longest delay. If "Check Closed" is required, an alarm will alert you if any of the clamp valves failed to close.
Parameters	Up to nine parameters can be used. Each parameter is an INTEGER with a value from 1 to 16, which defines the CLAMP type valve to use. If no parameter is given, the macro program uses the current valve.
Errors	All errors will cause the program to display an error and pause: MHND-250 ABORT "Parm %s is not an integer" MHND-251 ABORT "Parm %s is not between 1-16" MHND-252 ABORT "Valve %s is not setup" MHND-246 ABORT "Valve %s is not a clamp gripper" MHND-256 PAUSE "Clamp %s state error" MHND-257 PAUSE "Clamp %s state error"
Program Logic	• Check parameters • Close requested CLAMP valves • Turn on A side valve and turn OFF B side valve • Wait CLAMP "Operation Delay" period • If CLAMP "Check Closed" is YES • Check that each of the valve's Clamp open inputs are OFF and Clamp closed inputs are ON • Send an alarm if an error occurs

Item	Description
Timing Diagram	

44.3.3 Release Part

Refer to Table 44.3.3 (a) and Table 44.3.3 (b) for details on the Release Part macro.

Table 44.3.3 (a) Release Part Macro

Item	Description
Program Name	DROPPART.TP
Macro Name	Release Part
KAREL Program	MHDROPPT.KL
Description	This macro drops off parts using a CLAMP type valve. For each valve specified, the A side outputs are turned OFF and the B side outputs are turned ON. The macro program then waits for the clamp valve's "Operation Timeout" period. If the specified valves use different delays, the macro program will use the longest delay. If "Check Opened" is required, an alarm will alert you if any of the valves failed to open.
Parameters	Up to 9 parameters can be used. Each parameter is an INTEGER with a value from 1 to 16, which defines the CLAMP type valve to use. If no parameter is given, the macro program uses the current tool.
Errors	All errors will cause the program to display an error and pause: MHND-250 ABORT "Parm %s is not an integer" MHND-251 ABORT "Parm %s is not between 1-16" MHND-252 ABORT "Valve %s is not setup" MHND-246 ABORT "Valve %s is not a clamp gripper" MHND-256 PAUSE "Clamp %s state error" MHND-257 PAUSE "Clamp %s state error"

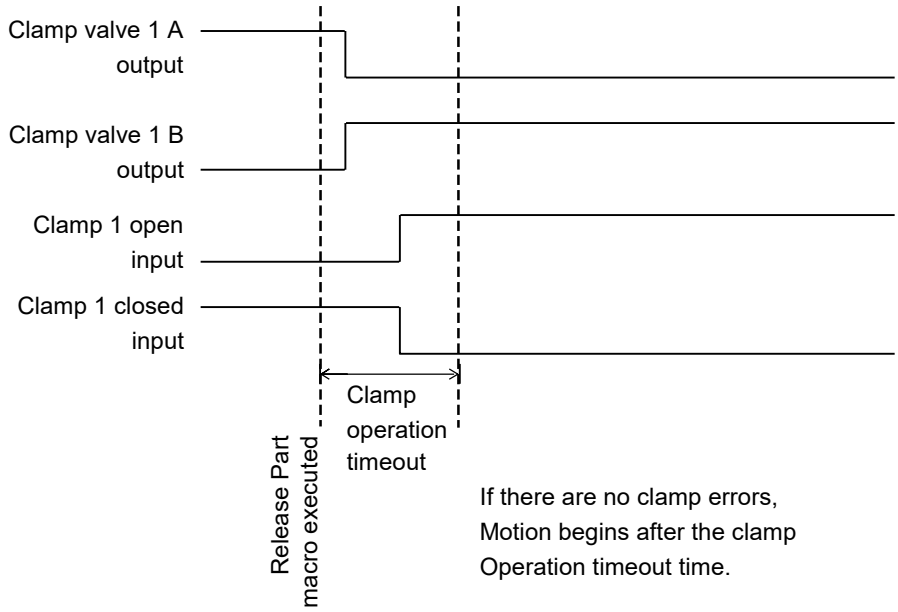
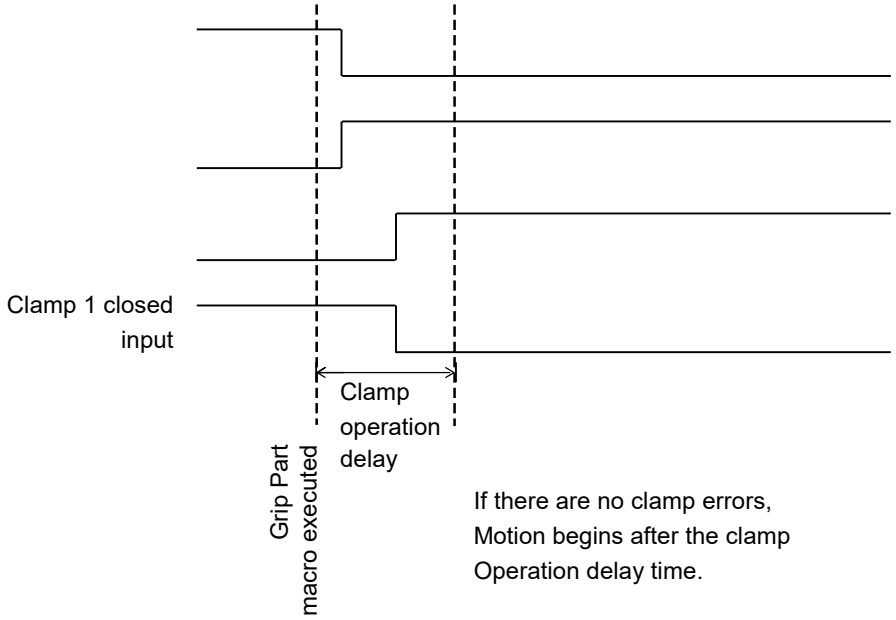
Item	Description
Program Logic	• Check parameters • If the continuous check of part presents is active, turn it off • Close the requested CLAMP valves • Turn on the B side valve and turn off the A side valve • Wait CLAMP "Operation Timeout" period • If CLAMP "Check Opened" is YES • Check that each of the valve's Clamp open inputs is ON and Clamp closed inputs is OFF • Send an alarm if an error occurs
Timing Diagram	

Table 44.3.3 (b) Release Part Macro

Item	Description
Program Name	DROPPART.TP
Macro Name	Release Part
KAREL Program	MHDROPPT.KL
Description	This macro drops off parts using a CLAMP type valve. For each valve specified, the A side outputs are turned OFF and the B side outputs are turned ON. The macro program then waits for the clamp valve's "Operation Delay" period. If the specified valves use different delays, the macro program will use the longest delay. If "Check Opened" is required, an alarm will alert you if any of the valves failed to open.
Parameters	Up to 9 parameters can be used. Each parameter is an INTEGER with a value from 1 to 16, which defines the CLAMP type valve to use. If no parameter is given, the macro program uses the current tool.
Errors	All errors will cause the program to display an error and pause: MHND-250 ABORT "Parm %s is not an integer" MHND-251 ABORT "Parm %s is not between 1-16" MHND-252 ABORT "Valve %s is not setup" MHND-246 ABORT "Valve %s is not a clamp gripper" MHND-256 PAUSE "Clamp %s state error" MHND-257 PAUSE "Clamp %s state error"

Item	Description
Program Logic	• Check parameters • If the continuous check of part presents is active, turn it off • Close the requested CLAMP valves • Turn on the B side valve and turn off the A side valve • Wait CLAMP "Operation Delay" period • If CLAMP "Check Opened" is YES • Check that each of the valve's Clamp open inputs is ON and Clamp closed inputs is OFF • Send an alarm if an error occurs
Timing Diagram	

44.3.4 Part Present

Refer to Table 44.3.4 (a) for details on the Part Present macro.

Table 44.3.4 (a) Part Present Macro

Item	Description
Program Name	CHKPART.TP
Macro Name	Part Present
KAREL Program	MHPTPRES.KL
Description	This macro checks that specified Part Present inputs are ON. If tryout mode is FALSE, for each part present input specified, this macro checks that the input is ON. If tryout mode is TRUE, for each part present input specified, this macro checks that the input is OFF.
Parameters	Up to ten parameters can be used. Each parameter is an INTEGER with a value from 1 to 12, which defines the Part Present input to check. If no parameter is given, an error occurs.
Errors	MHND-250 ABORT "Parm %s is not an integer" MHND-260 ABORT "Parm %s is not between 1-12" MHND-253 ABORT "No parameters passed to routine" MHND-254 PAUSE "%s not sensing part" MHND-248 PAUSE "TRYOUT MODE: %s reading part"

Item	Description
Program Logic	• Check parameters • For each specified Part Present input: • If the input is OFF, tryout mode is FALSE, and the alarm is not disabled, send the alarm "signal not reading part" • If the input is ON, tryout mode is TRUE, and the alarm is not disabled, send the alarm "TRYOUT MODE: signal reading part" • If Part Present "Continuous Check" is required, turn on a condition handler to watch that Part Present inputs stay on.

44.3.5 Check No Part

Refer to Table 44.3.5 (a) for details on the Check No Part macro.

Table 44.3.5 (a) Check No Part Macro

Item	Description
Program Name	CHKNOPRT.TP
Macro Name	Check No Part
KAREL Program	MHNOPART.KL
Description	This macro checks that the specified Part Present inputs are OFF. Tryout mode is not used by this macro. This macro handles any errors encountered if they have not been disabled by the user.
Parameters	Up to ten parameters can be used. Each parameter is an INTEGER with a value from 1 to 12, which defines the Part Present input to check. If no parameter is given, an error occurs.
Errors	MHND-250 ABORT "Parm %s is not an integer" MHND-260 ABORT "Parm %s is not between 1-12" MHND-252 ABORT "Valve %s is not setup" MHND-253 ABORT "No parameters passed to routine" MHND-255 PAUSE "%s still reading part"
Program Logic	• Check parameters • For each specified Part Present input: – If the input is ON and the alarm is not disabled, send the alarm "signal still reading part".

44.3.6 Prepare to Pick

Refer to Table 44.3.6 (a) for details on the Prepare to Pick macro.

Table 44.3.6 (a) Prepare to Pick Macro

Item	Description
Program Name	OKTOPICK.TP
Macro Name	Prepare to Pick
KAREL Program	MHOKPICK.KL
Description	This macro checks the required conditions before picking up a part for both CLAMP and VACUUM type valves. For each valve specified, if "Check Opened" is required, an alarm will alert you if any of the clamps are not open. For each valve specified, an alarm will alert you if any of the valve's Part Present inputs are on. Then, for ease of customization, the teach pendant program CLR2PRCD.TP is called.
Parameters	Up to nine parameters can be used. Each parameter is an INTEGER with a value from 1 to 9, which defines the tooling valves to check. If no parameter is given, the current valve is used.

Item	Description
Errors	MHND-250 ABORT "Parm %s is not an integer" MHND-251 ABORT "Parm %s is not between 1-16" MHND-252 ABORT "Valve %s is not setup" MHND-246 ABORT "Valve %s is not a clamp gripper" MHND-256 PAUSE "Clamp %s state error" MHND-257 PAUSE "Clamp %s state error" MHND-259 PAUSE "%s vacuum is not off" MHND-255 PAUSE "%s still reading part"
Program Logic	• Check parameters • For each specified valve – If CLAMP "Check Opened" is YES, check that each of the valve's Clamp open inputs are ON and Clamp closed inputs are OFF. – If VACUUM "Vac Feedback" is YES, check that each of the valve's Vacuum Made inputs is OFF. – Check that the valve's Part Present inputs are OFF • Call CLR2PRCD.TP

44.3.7 Clear to Proceed

Refer to Table 44.3.7 (a) for details on the Clear to Proceed macro.

Table 44.3.7 (a) Clear to Proceed Macro

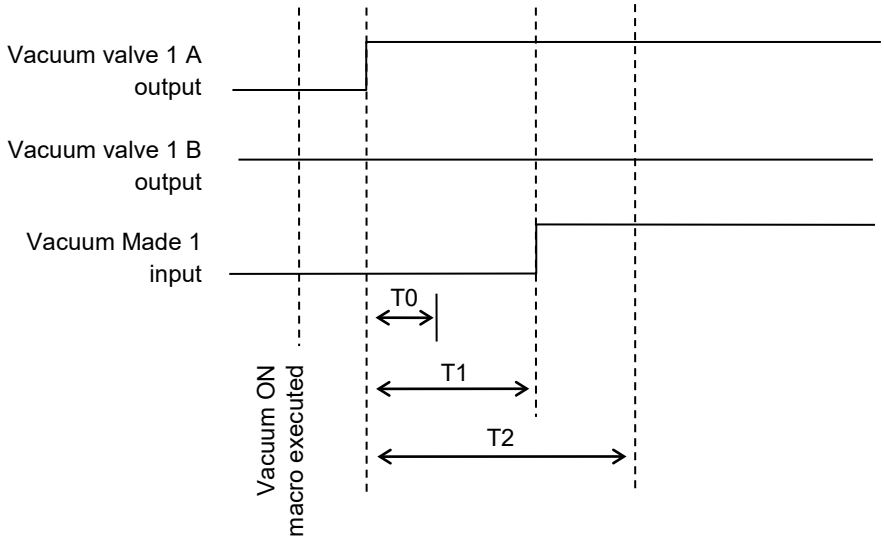
Item	Description
Program Name	CLR2PRCD.TP
Macro Name	Clear to Proceed
KAREL Program	N/A
Description	This macro is a teach pendant program you customize to define any other conditions that need to be checked before picking up the part. This macro is called from the Prepare to Pick macro (OKTOPICK.TP). This macro is used by both types of tooling valves.
Parameters	None
Errors	N/A (You define the UALMs.)
Program Logic	TPE defined by user /PROG CLR2PRCD Macro 1: !This macro is called by oktopick ; 2: !tto allow the user to add any ; 3: !additional criteria for the ; 4: !robot to wait on before picking ; 5: !up the part. ; 6: ; 7: !Does not use the parameters ; 8: !passed to oktopick. ;

44.3.8 Turn ON Vacuum

Refer to Table 44.3.8 (a) for details on the Turn ON Vacuum macro.

Table 44.3.8 (a) Turn ON Vacuum Macro

Item	Description
Program Name	VACUUMON.TP
Macro Name	Turn ON Vacuum
KAREL Program	MHVACON.KL
Description	This macro picks up parts using a VACUUM type valve. For each valve specified, the B side outputs are turned OFF and the A side outputs are turned ON. If "Vac Feedback" is YES for any selected valve, the logic waits for the valve's "Ignore Feedback" period. If the specified valves use different delays, the macro program will use the longest delay. Then the macro program will wait for up to the "Feedback Timeout" for the vacuum made inputs to turn ON. As soon as the inputs are ON, the macro program continues. If the inputs never turn ON, an alarm is given. If "Vac Feedback" is NO for all selected valves, the macro program waits for the valve's "Feedback Timeout" period. If "Continuous Check" is required, an alarm will alert you if the vacuum made input turns OFF.
Parameters	Up to two parameters can be defined. Each parameter is an INTEGER with a value from 1 to 2, which defines the VACUUM type valve to use. If no parameter is given, the macro program uses the current valve.
Errors	MHND-250 ABORT "Parm %s is not an integer" MHND-247 ABORT "Parm %s is not between 1-2" MHND-252 ABORT "Valve %s is not setup" MHND-245 ABORT "Valve %s is not a vacuum gripper" MHND-258 PAUSE "%s did not make vacuum"
Program Logic	• Check parameters • Close the required VACUUM valves – Turn on the A side valve and turn off the B side valve • If "Vac Feedback" is YES – Wait for VACUUM Ignore Feedback – For up to ["Feedback Timeout" - "Ignore Feedback"] ms, watch for the Vacuum Made input to turn ON. – If the Vacuum Made input does not turn ON, give an error • If "Vac Feedback" is NO – Wait for VACUUM Feedback Timeout • If Vac Feedback "Continuous Check" is required and tryout mode is FALSE, turn on a condition handler to watch that the Vacuum Made inputs stay ON.

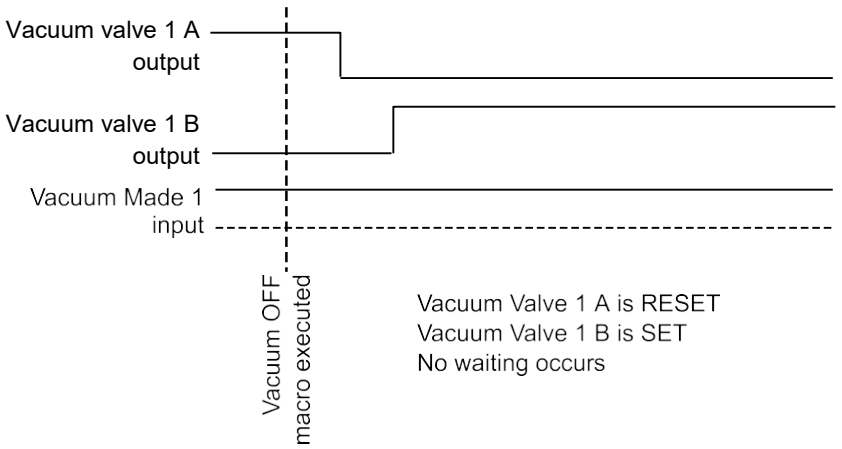
Item	Description
Timing Diagram	 T0 – Time to ignore the Vacuum Made signal T1 – Time to achieve vacuum feedback (Maximum Feedback Timeout) T2 – Time to achieve vacuum with no feedback (Feedback Timeout) Motion begins after T1 or T2, depending on the “Vac Feedback” setting

44.3.9 Turn OFF Vacuum

Refer to Table 44.3.9 (a) for details on the Turn OFF Vacuum macro.

Table 44.3.9 (a) Turn OFF Vacuum Macro

Item	Description
Program Name	VACUUMOF.TP
Macro Name	Turn OFF Vacuum
KAREL Program	MHVACOFF.KL
Description	This macro puts down parts using a VACUUM type valve. If "Continuous Check" is active, the macro disables the condition that is watching for the Vacuum Made input to turn OFF. For each valve specified, the B side outputs are turned ON and the A side outputs are turned OFF.
Parameters	Up to two parameters can be defined. Each parameter is an INTEGER with a value from 1 to 2, which defines the VACUUM type valve to use. If no parameter is given, the macro program uses the current valve.
Errors	MHND-250 ABORT "Parm %s is not an integer" MHND-247 ABORT "Parm %s is not between 1-2" MHND-252 ABORT "Valve %s is not setup" MHND-245 ABORT "Valve %s is not a vacuum gripper" MHND-259 PAUSE "%s vacuum is not off"
Program Logic	• Check parameters • If the continuous checking of part presents is active, turn it off • If the continuous checking of vacuum made is active, turn it off • Open the requested vacuum valves – Turn on the B side valve and turn off the A side valve

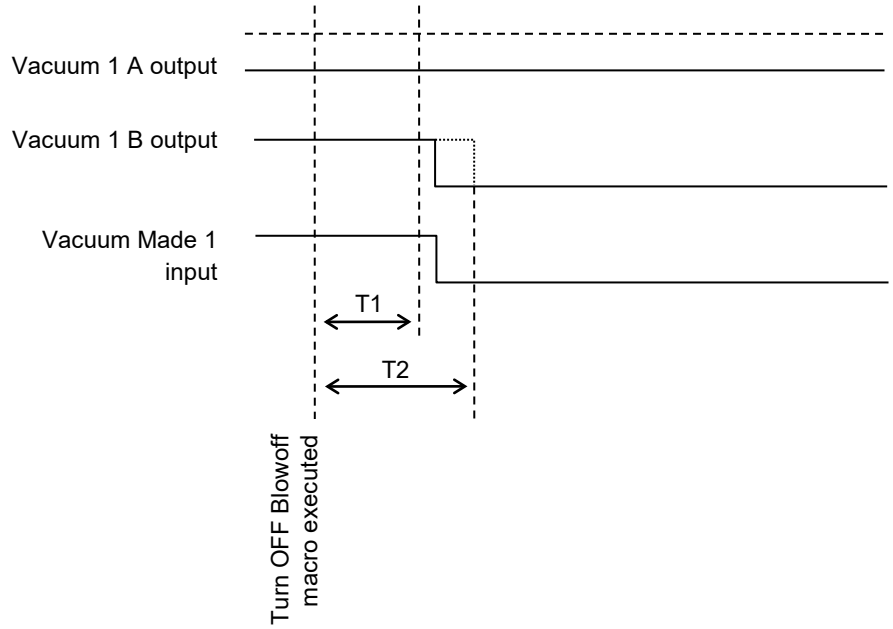
Item	Description
Timing Diagram	

44.3.10 Turn OFF Blowoff

Refer to Table 44.3.10 (a) for details on the Turn OFF Blowoff macro.

Table 44.3.10 (a) Turn OFF Blowoff Macro

Item	Description
Program Name	BLOWOFF.TP
Macro Name	Turn OFF Blowoff
KAREL Program	MHBLOWOF.KL
Description	This macro turns off the blowoff output when using a VACUUM type valve. If "Vac Feedback" is YES for any selected valve, the macro program will wait for up to the valve's "Blow off time" period, for the Vacuum Made inputs to go OFF. When the inputs are OFF, the macro program continues. If the inputs never go OFF, an alarm is issued. If "Vac Feedback" is NO for all selected valves, the macro program waits for the valve's "Blow off time" period.
Parameters	Up to two parameters can be defined. Each parameter is an INTEGER with a value from 1 to 2, which defines the VACUUM type valve to use. If no parameter is given, the macro program uses the current valve.
Errors	MHND-250 ABORT "Parm %s is not an integer" MHND-247 ABORT "Parm %s is not between 1-2" MHND-252 ABORT "Valve %s is not setup" MHND-245 ABORT "Valve %s is not a vacuum gripper"
Program Logic	• Check parameters • If "Vac Feedback" is YES – For up to "Blow off time" ms, watch for the Vacuum Made input to turn OFF. – If the Vacuum Made input does not turn OFF, issue an error. • If "Vac Feedback" is NO – Wait for VACUUM "Blow off time" ms. • Turn OFF Blow off output – Turn off B side valve.

Item	Description
Timing Diagram	 T1 – Time for vacuum made to be lost (maximum T2) T2 – Time for vacuum to be released without feedback (Blowoff Time Motion begins after T1 or T2, depending on “Vac Feedback” setting)

44.3.11 Set CurrentValve

Refer to Table 44.3.11 (a) for details on the Set CurrentValve macro.

Table 44.3.11 (a) Set CurrentValve Macro

Item	Description
Program Name	SETVALVE.TP
Macro Name	Set CurrentValve
KAREL Program	MHSETVLV.KL
Description	This macro sets the "Current Valve," which is used as a default by most the Material Handling macros.
Parameters	One macro can be used. The parameter is an INTEGER with a value from 1 to 16, which defines the valve to use as a default.
Errors	MHND-250 ABORT "Parm %s is not an integer" MHND-251 ABORT "Parm %s is not between 1-16" MHND-252 ABORT "Valve %s is not setup" MHND-253 ABORT "No parameters passed to routine"

44.3.12 Set MH Tool

Refer to Table 44.3.12 (a) for details on the Set MH Tool macro.

Table 44.3.12 (a) Set MH Tool

Item	Description
Program Name	SET_TOOL.TP
Macro Name	Set MH Tool
KAREL Program	MHSETGRP.KL
Description	This macro sets the "Current Tool", which is used as a default by most the Material Handling macros.
Parameters	The parameter is an INTEGER with a value from 1 to 16, which defines the tool to use as a default.
Errors	MHND-250 ABORT "Parm %s is not an integer" MHND-251 ABORT "Parm %s is not between 1-16" MHND-252 ABORT "Valve %s is not setup" MHND-253 ABORT "No parameters passed to routine"

44.4 EXAMPLE PROGRAM

It is recommended that you use the Auto Backward Exit option (R861) when using Cancel Recovery. This option allows the robot's exit path to be recorded with minimal Path programming. See the example below.

Teach Pendant Program: tmptpick.tp

```
1:  ;
2: !Set Recovery Skip Input ;
3: SKIP CONDITION DI[27:OFF:StyleBit4]=ON   ;
4: !Start Recording Recovery Path ;
5: Rec Path Start ;
6: ;
7: !Check tooling in Open State ;
8: !Value is Valve#(s) to Check ;
9: Prepare to Pick(0) ;
10: ;
11: !Monitor for Skip Jump Cond ;
12:J P[1] 100% CNT100 SkipJump,LBL[999]     ;
13:L P[2] 2000mm/sec CNT100 SkipJump,LBL[999]      ;
14:L P[3] 1500mm/sec CNT10 SkipJump,LBL[999]       ;
15:L P[4] 400mm/sec FINE SkipJump,LBL[999]  ;
16: !Stop Recording Recovery Path ;
17: Rec Path End       ;
18: !Grip Part Pass Valve#(s) to use ;
19: Grip Part(2) ;
20: !If didnt Grip part,Jump to Exit ;
21: IF (F[40:OFF:MH Grip Cancel]=ON),JMP LBL[999] ;
22: !Update Payload settings ;
23: PAYLOAD[2] ;
24: ;
25:L P[5] 400mm/sec FINE      ;
26:J P[6] 100% CNT50   ;
27:J P[7] 100% CNT50   ;
28: ;
29: END ;
30: ;
```

```
31: LBL[999:Recov From Path] ;
32: !Execute Path Recorded to Exit ;
33: Do Bwd Exit ;
```

44.5 PRODUCTION

When you have the Material Handling Plug-in, production operation has the following features, if they have been enabled:

- Tryout mode
- Disable for 20 cycles
- Alarms

44.5.1 MH Tryout Mode

MH Tryout Mode enables and disables the ability of the robot to execute material handling commands in a tryout, or dry run, mode. You enable and disable MH Tryout Mode on the TEST CYCLE Spot Weld menu after you set up the cell interface I/O. Refer to Section 4.1 CELL INTERFACE I/O SIGNALS in the *Spot Welding Function OPERATOR'S MANUAL* (B-84464EN-4) for more information.

When MH Tryout Mode is enabled, it does the following:

- Disables checking for feedback signals.
- Changes the checking for part present signals. When MH Tryout Mode is enabled on the TEST CYCLE Spot Weld menu, the part present signals must never be on.
- Allows the robot to run the program with all other material handling alarms and conditions enabled.

44.5.2 Disable for 20 Cycles

When you use the Disable Fault Checking feature, you can specify that the robot should ignore tooling valve inputs, clamps, and part present alarms for 20 cycles. After 20 cycles, an error message will be re-enabled and, if the alarm occurs again, will pause the program until it is reset by the operator.

You can use the Disable Fault Checking feature to disable each tooling valve PAUSE alarm for 20 cycles. When a material handling pause alarm occurs, you will be able to choose Disable Alarm on the error recovery screen to disable this particular alarm for 20 cycles. Refer to Section A.5 DISABLE FAULT CHECKING in the *Spot Welding Function OPERATOR'S MANUAL* (B-84464EN-4) for more information about using the Disable Fault Checking feature to do this.

44.5.3 Alarms

All Material Handling alarms turn on the MH Fault output, if the controller is not in MH Tryout Mode. The output will be turned off when a FAULT RESET is given.

The MH Alert output turns ON when a fault that has been disabled for 20 cycles occurs. The output will be turned OFF when a FAULT RESET is given.

45 SPECIAL JOG SEQUENCE

This function enables Jog operation without pressing the [SHIFT] key.
Pressing only Jog keys, the robot moves in low speed. So you can confirm the direction the robot moves in under safe condition.
Normal Jog operation is also possible by pressing the [SHIFT] key additionally.
It is possible to order this function independently.
In that case, this function is available in all tools.

45.1 USAGE

When this function is valid, Jog operation is possible in the following conditions.

Press only Jog keys:
The robot moves in the speed corresponds to override 5%. When the override is less than 5%, the robot moves in the speed corresponds to the override.

Press both Jog keys and [SHIFT] key:
The robot moves in the normal Jog speed. The Jog speed corresponds to the override specified in Teach Pendant.

When this function is invalid, Jog operation is possible in the following conditions.

Press only Jog keys:
The robot does not move and the following message is displayed.
"JOG-007 Press shift key to jog"

Press both Jog keys and [SHIFT] key:
The robot moves in the normal speed. The Jog speed corresponds to the override specified by Teach Pendant.

45.2 SETUP

This function can be set up by the setting item "Special Jog sequence val."(MENU → SYSTEM → CONFIG → Override → Special Jog sequence val.).

Special Jog sequence val.:
0: This function is invalid.
1-5: This function is valid (Default is 5). When only Jog key is pressed, the robot moves in the speed corresponds to the override specified here. Default value is 5%.

45.3 APPLICATION

Automatic Override setup when changing frames:

Normally, override is automatically initialized to 10% when Jog frame is changed in Teach Pendant. With this function, as you can confirm the direction in low speed, you may want to make this automatic override setup invalid.
Setting the following setting item(MENU at controlled start→ CONFIG → Override → Coordinate Override), the current override can be kept when frame is changed.

Coordinate Override = 100 (Standard value 10)

Meaning of this setting:

When the current override is larger than the value specified here, the override is changed into the value specified here.

46 MOTION INSTRUCTION ENHANCED EDITING

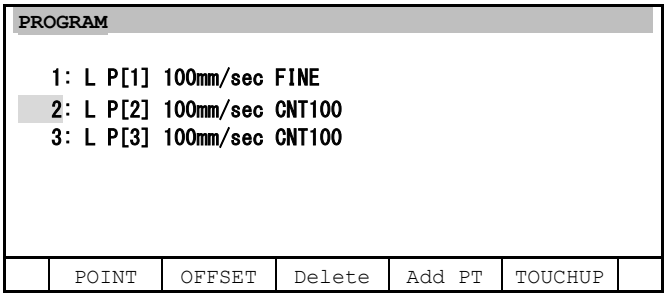
This function includes two functions, “Motion instruction insert and instruction delete function” and “Data offset function”. These are described below.
It is possible to order Enhanced Edit option (J560) independently.
In that case, this function is available only in handling tool and LR handling tool.

46.1 MOTION INSTRUCTION INSERT AND INSTRUCTION DELETE FUNCTION

This function enables you to insert motion instructions or delete instructions in a program with function keys in the edit screen.

46.1.1 Usage

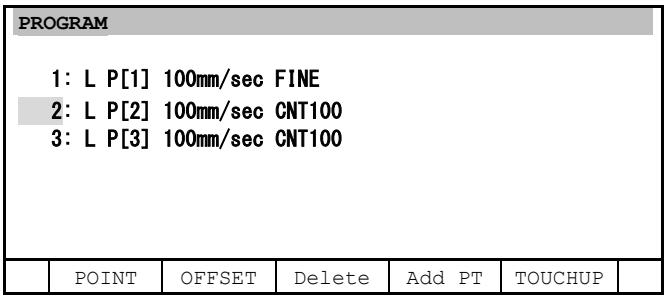
Function keys are displayed as follows in EDIT screen.



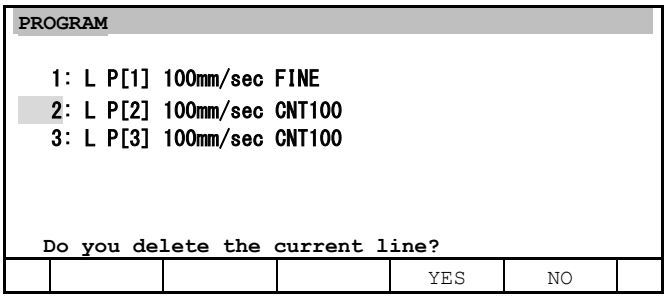
Procedure 46-1 Delete instructions

Step

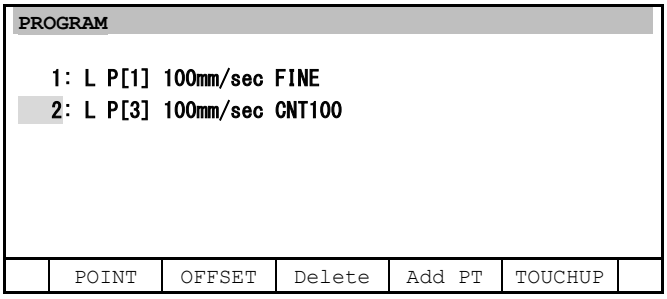
1 Move the cursor on the line you want to delete.
2 Enable Teach Pendant, and press the [SHIFT] key and the F3, Delete.



3 F4, YES and F5, NO will be displayed. Press YES to delete. When NO is pressed, the deleted is cancelled and edit screen will be displayed .



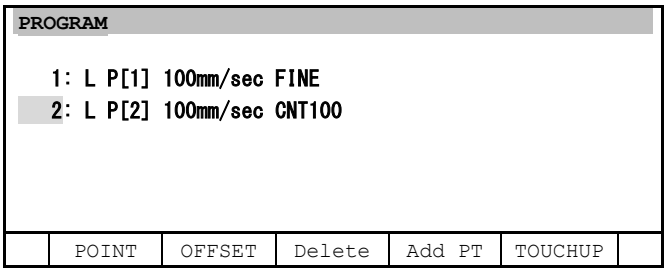
4 The line that the cursor was on is deleted.



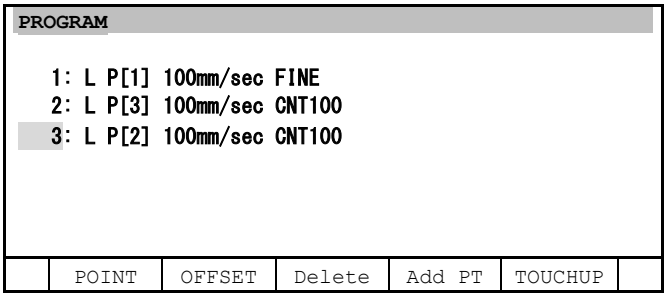
Procedure 46-2 Insert motion instructions

Step

1 Move the cursor on the line you want to delete.
2 Enable Teach Pendant, and press the [SHIFT] key and the F4, Add PT.



3 Motion line is inserted, and the cursor moves to the next line.



46.1.2 Precautions

- Standard motion instruction that is selected in F1, POINT menu is inserted. This is the same motion instruction that is taught by pressing the [SHIFT] key and POINT.
- When Circular motion is selected as standard motion instruction, Circular motion is inserted. In this case, current position is recorded in passing point and the target point is unspecified.
- After operations with this function, you cannot undo the operation by Undo in [EDCMD].

- This function cannot be used in quick menu.

46.2 DATA OFFSET FUNCTION

Data offset function enables you to change multiple position data or motion speed at once in a program that you are editing. This function includes 3 functions, “Feed rate conversion”, “Position data conversion (direct method)”, and “Position data conversion (2 point teach method)”.

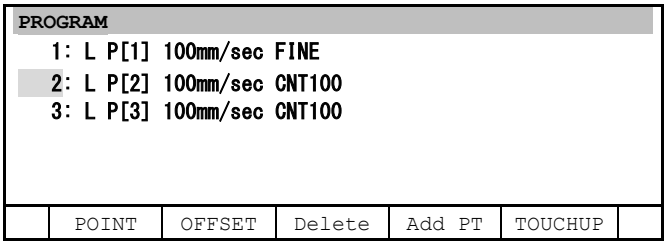
46.2.1 Feed Rate Conversion

This function converts the feed rate of multiple motion instructions at once.
- You can convert a whole or part of the program you are editing.
- You can specify the offset to current feed rate.
- You can convert the feed rate of all motion instructions into specified value regardless of current feed rate.
- You can convert the motion instructions whose current feed rate is equal to specified value into the value you want to give.
- You can specify offset rate (%) to current feed rate.
- Except offset rate specification, only one motion type of Joint, Linear or Circular motion can be converted at once.

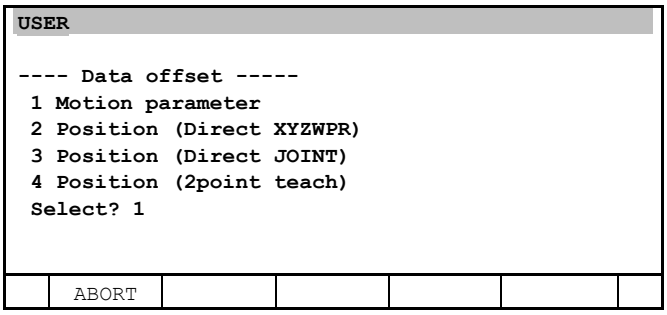
Procedure 46-3 Feed rate conversion

Step

1 Press the F2, OFFSET in EDIT screen.



2 Input “1” to select “1 Motion parameter”.



3 To convert a whole program, input “1”. Then go to step 5. To convert a part of a program, input “2”. Then go to step 4.

```
USER
 << Motion parameter >>

---- Program range -----
 1 Whole program
 2 Part of the program
  Select? 2


 [ ] [ABORT] [ ] [ ] [ ] [ ] [ ]
```

4 Input start line number and last line number.

```
USER
 << Motion parameter >>

---- Program range -----
 1 Whole program
 2 Part of the program
Select? 2

Start line number:1

Last line number:


 [ ] [ABORT] [ ] [ ] [ ] [ ] [ ]
```

5 Select target motion type or All (Rate specify). If you select All (Rate specify), go to step 11.

```
USER
 << Motion parameter >>
 ---- Target motion type ----
1 JOINT
2 Linear
3 Circular
4 Arc
5 Liner,Circular,Arc
6 All (Rate specify)
Select?

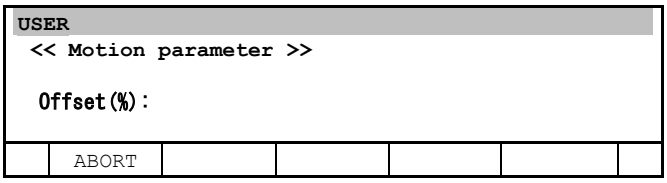

 [ ] [ABORT] [ ] [ ] [ ] [ ] [ ]
```

6 Select the method for converting feed rate.
Input “1” to specify the offset to the current feed rate. Then go to step 7.
Input “2” to convert feed rate to specified value. Then go to step 8.
Input “3” to specify the rate to the current feed rate. Then go to step 11.

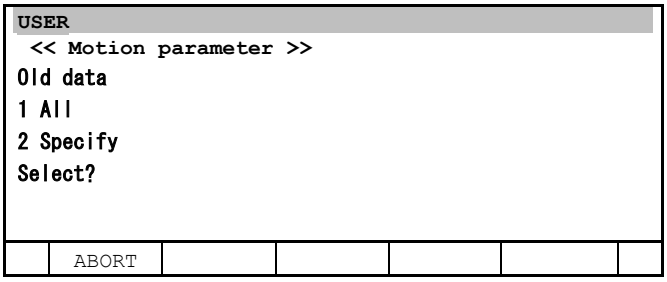
```
USER
 << Motion parameter >>
---- Method ----
1 Offset
2 Absolute
3 Rate
Select?


 [ ] [ABORT] [ ] [ ] [ ] [ ] [ ]
```

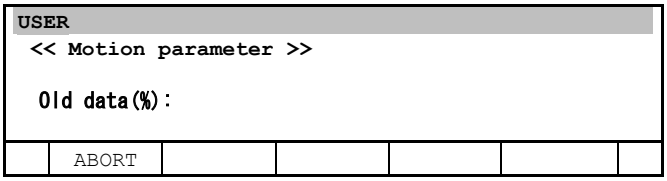
7 Input the offset. When you want to decrease the value, input negative value. The unit is % when Joint motion or mm/sec when Linear motion or Circular motion. Go to step 12.



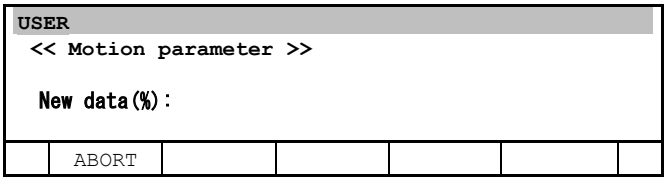
8 Input “1” to convert all the motion instructions. Then go to step 10.Input “2” to convert only the motion instructions whose current feed rate is equal to the value specified by operation 9. Then go to step 9.



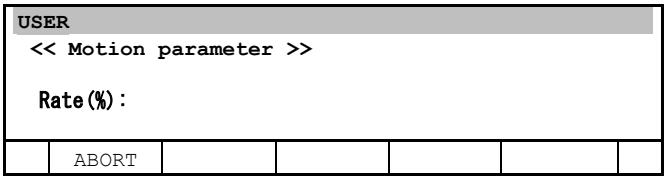
9 Input current feed rate of the motion instructions that you convert. The unit is % when Joint motion or mm/sec when Linear or Circular motion.



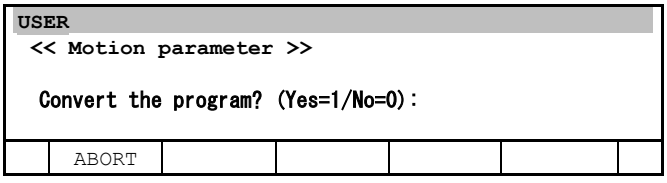
10 Input new value of feed rate. The unit is % when Joint motion or mm/sec when Linear or Circular motion. Go to step 12.



11 Specify the rate (-100 to 100).



12 Input “1” to convert the program. Input “0” to cancel the conversion.



13 Edit screen will be displayed after conversion.

46.2.2 Position Data Conversion (direct method)

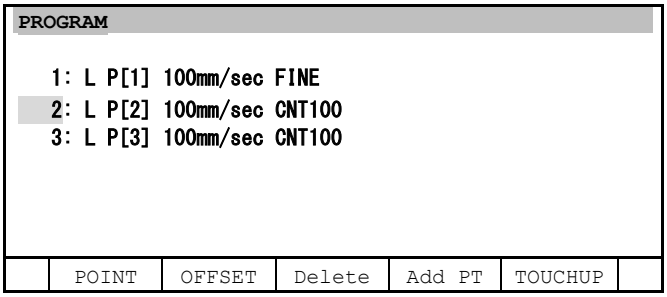
Position data conversion (2 point teach method) converts multiple position data in the program at once. Translation is specified by teaching 2 point, the original position and the destination position.

- You can convert a whole or part of the program you are editing.
- You can specify the offset to current position data.
- You can convert the position data of all motion instructions into specified value regardless of current position data.
- You can convert the motion instructions whose current position data is equal to specified value into the value you want.
- You can specify the number of user frame to specify the value.
- You can select the type, Cartesian (X, Y, Z, W, P, R) or Joint (J1 to J9).
- Only one element of X, Y, Z, W, P, R or J1 to J9 can be converted at once.

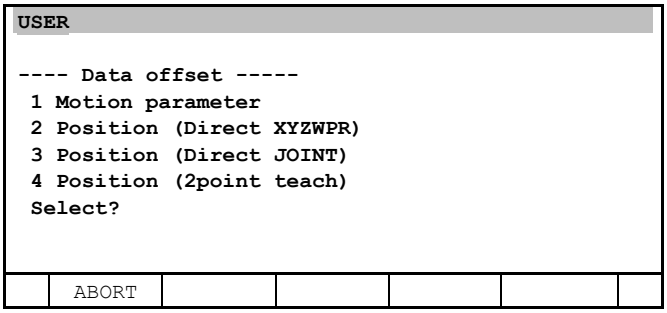
Procedure 46-4 Position data conversion (direct method)

Step

1 Press the F2, OFFSET in EDIT screen.



2 Input “1” to select “2 Position (Direct XYZWPR). Input “2” to select “3 Position (Direct JOINT).



3 Input “1” to convert a whole program. Then go to step 5. Input “2” to convert a part of a program. Then go to step 4.

```
USER
 << Position (Direct XYZWPR) >>

---- Program range -----
 1 Whole program
 2 Part of the program
  Select?

[ ] [ABORT] [ ] [ ] [ ] [ ] [ ]
```

4 Input start line number and last line number.

```
USER
 << Position (Direct XYZWPR) >>

---- Program range -----
 1 Whole program
 2 Part of the program
  Select?  2

Start line number: 1

Last line number:

[ ] [ABORT] [ ] [ ] [ ] [ ] [ ]
```

5 Input the number of the element or axis number to convert.

When “Direct XYZWPR” is selected

```
USER
<< Position (Direct XYZWPR) >>

 ---- Target motion type ----
1 X  2 Y  3 Z
1 W  2 P  3 R
Select?

[ ] [ABORT] [ ] [ ] [ ] [ ] [ ]
```

When “Direct JOINT” is selected

```
USER
<< Position (Direct JOINT) >>

 ---- Target axis ----
Enter axis number?

[ ] [ABORT] [ ] [ ] [ ] [ ] [ ]
```

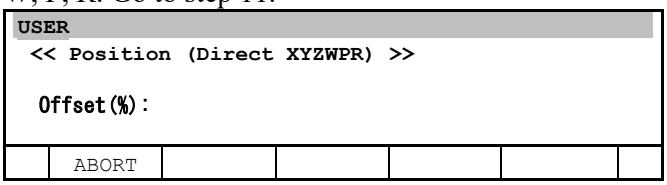
6 Input “1” to specify the offset to current position data. Then go to step 7. Input “2” to specify the position data into specified value. Then go to step 8.

```
USER
 << Position (Direct XYZWPR) >>

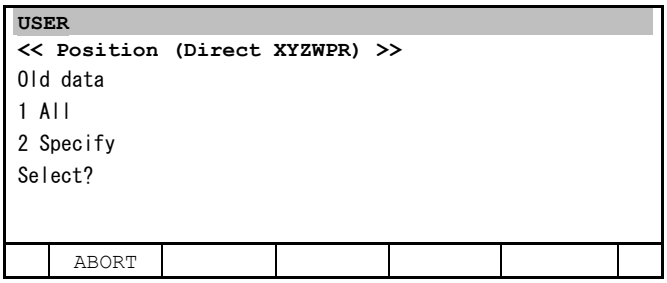
---- Method ----
1 Offset
2 Absolute
Select?

[ ] [ABORT] [ ] [ ] [ ] [ ] [ ]
```

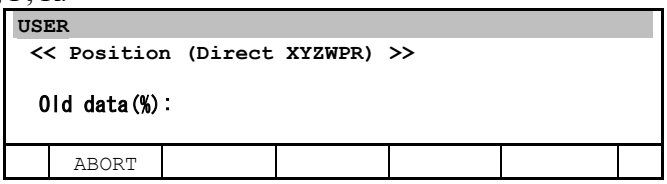
7 Input the offset. When you want to decrease the value, input negative value. The unit is mm when X, Y, Z or deg when W, P, R. Go to step 11.



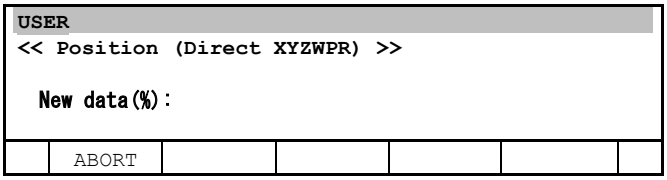
8 Input “1” to convert all the motion instructions. Then go to step 10. Input “2” to convert only the motion instructions whose current feed rate is equal to the value specified by operation 9. Then go to step 9.



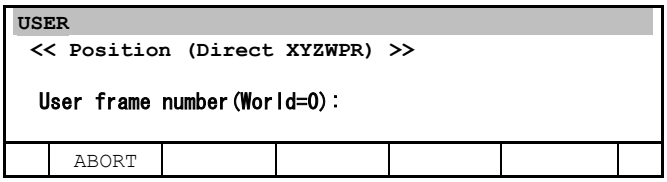
9 Input the current position data of the motion instruction that you convert. The unit is mm when X, Y, Z or deg when W, P, R.



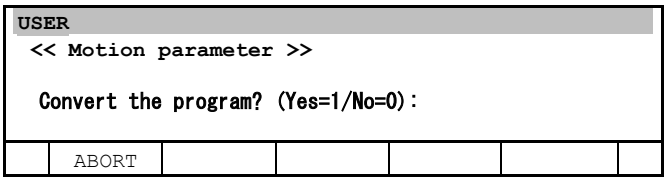
10 Input new position data. The unit is mm when X, Y, Z or deg when W, P, R.



11 Input the user frame number that is the base of conversion. When world frame is used, input “0”.



12 Input “1” to convert the program. Input “0” to cancel the conversion.



13 Edit screen will be displayed after conversion.

46.2.3 Position Data Conversion (2 point teach method)

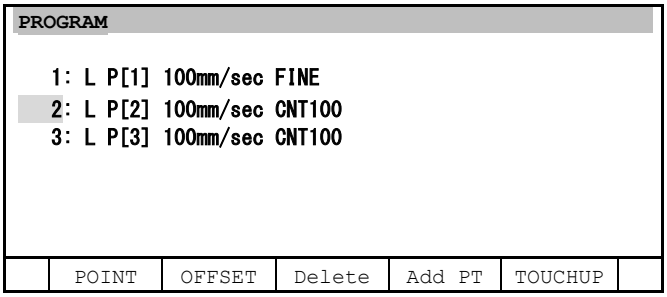
Position data conversion (2 point teach method) converts multiple position data in the program at once. Translation is specified by teaching 2 point, the original position and the destination position.

- You can convert a whole or part of the program you are editing.
- Parallel element (X, Y, Z) of the difference between the two points is converted on world frame. Rotation element (W, P, R) of the difference between the two points is converted on tool frame.

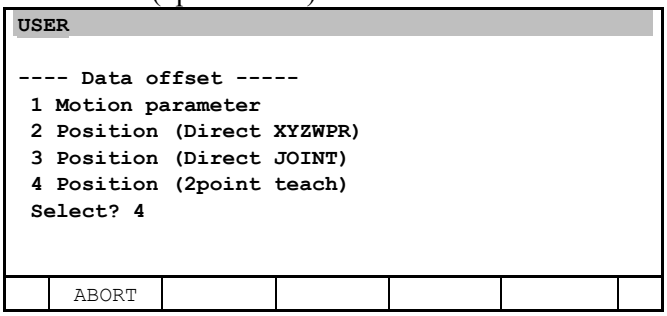
Procedure 46-5 Position data conversion (2 point teach method)

Step

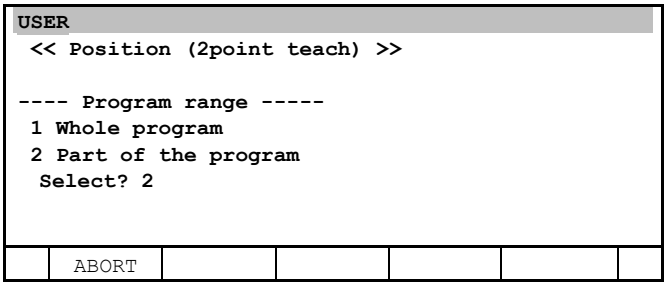
1 Press the F2, OFFSET in EDIT screen.



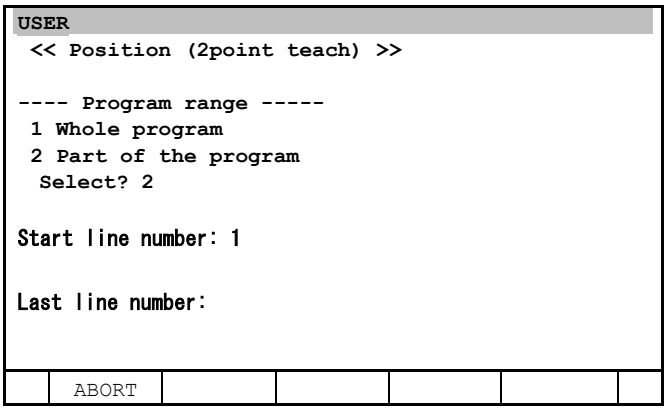
2 Input “4” to select “4 Position (2point teach)”.



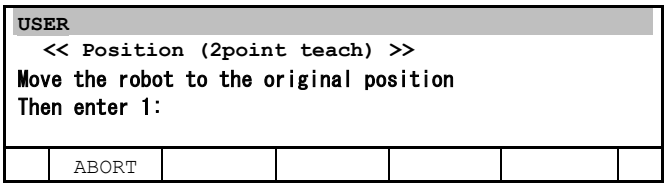
3 Input “1” to convert a whole program. Then go to step 5. Input “2” to convert a part of a program. Then go to step 4.



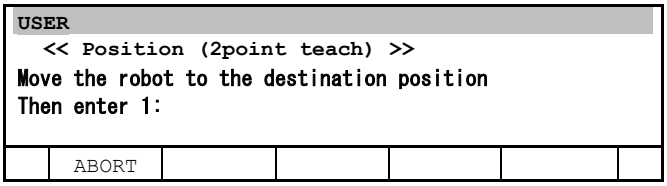
4 Input the start line number and last line number.



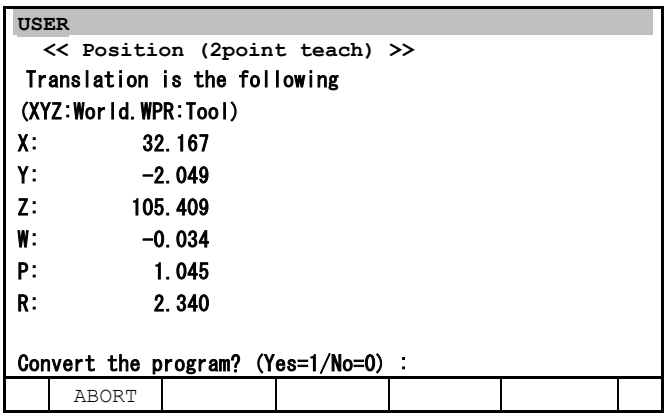
5 Input “1” after moving the robot to the original position.



6 Input “1” after moving the robot to the destination position.



7 Each translation is displayed.
The value X, Y, Z is the translation on world frame.
The value W, P, R is the translation on tool frame.
Input “1” to convert the program.
Input “0” to cancel the conversion.



8 Edit screen will be displayed after conversion.

46.2.4 Precautions

- After operations with this function, you cannot undo the operation by Undo in [EDCMD].
- The original program is copied to YFDTOFBK. When you want to use the original program, copy YFDTOFBK.
- A program whose name is YFDTOFBK is overwritten after conversion. Please do not create YFDTOFBK.
- In the multi group system, only the position data of first group can be converted.
- When the position data is converted in a system with extended axes, only the position data of robot axes are converted. The position data of extended axes are not converted.
- When another screen is displayed (ex current position display) during the operation of data offset function, you can return to the operation of data offset function by pressing the [EDIT] key, or selecting “9 USER” in MENU.
- Press the F1, ”ABORT” when you want to abort data offset function and return to EDIT screen. When you press the F2, “OFFSET” again after this operation, you can restart data offset function from the beginning.
- When teach pendant is disabled, data offset function does not work by pressing the F2, “OFFSET”. When teach pendant is turned disabled during the operation of data offset function, the operation is aborted and EDIT screen is displayed.
- When the position data is singular point or out of stroke limit after conversion, the position data will be unspecified data. In this case, the number of the unspecified data will be displayed and displayed as follows “Error occurred(1:OK 0:Cancel)”. When you select 1, the position data that have no error will be normally converted and the position data that have errors will be unspecified data. When you select 0, the conversion will be canceled.
- In conversion of position data by “Position (Direct XYZWPR)” or “Position (2point teach)”, the amount of the change in all axes is automatically adjusted not to exceed plus and minus 180 degree.
- When position data is converted, the position data is recorded as Cartesian coordinates.
- When position register is used in the motion instruction, this function does not convert the position data of the position register.
- This function does not convert the position data of Palletizing function.
- This function does not convert the position data of Asynchronous operation group instruction or Synchronous operation group instruction.
- This function does not convert the position data of Incremental instruction correctly. Please do not use this function to convert the position data of Incremental instruction.
- This function does not change the feed rate of the following motion instruction correctly. Please do not use this function to change the feed rate of the following motion instruction.
  - Feed rate is specified by a register.
  - Feed rate unit of Joint motion instruction is not %.
  - Feed rate unit of motion instruction except Joint is not mm/sec.

47 FINISHING FUNCTION PACKAGE

The description of option software “Finishing function package(J558)” is written in this chapter. This software is package of two optional software of Special Jog Sequence (J559) and Motion Instruction Enhanced Editing(J560).

NOTE
Finishing Function Package can be used only in handling tool.

- Please refer to Chapter 45 SPECIAL JOG SEQUENCE on this manual if you want to know about “Special Jog Sequence(J559)”.
- Please refer to Chapter 46 MOTION INSTRUCTION ENHANCED EDITING on this manual if you want to know about “Motion instruction Enhanced Editing(J560)”.

48 AUTO BACKWARD EXIT

Auto Backward Exit (R861) allows a programmer or operator to back a robot out of a machine or from within a workpiece safely and easily without teaching a path or jogging. The robot starts recording its path at a specified point in a program, and moves back along the recorded path to return to the original start point under program control.

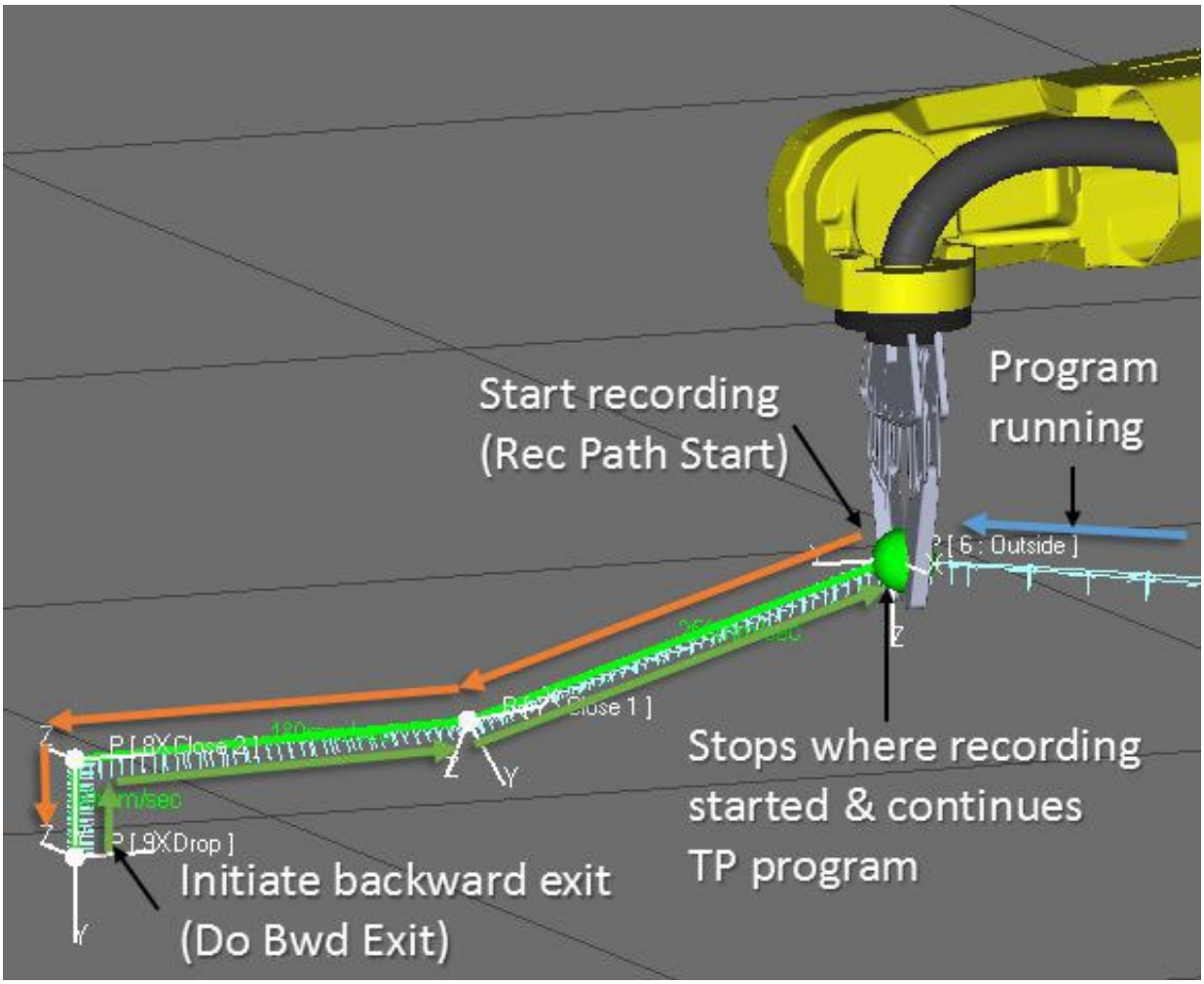


Fig. 48 Auto Backward Exit Application

48.1 Specification Overview

Auto Backward Exit can be enabled at Controlled Start. Some setup items can be set at both Cold Start and Controlled Start.

Auto Backward Exit is controlled through a system of macro type teach pendant programs. All except **Do Bwd Exit** can be executed without stopping the robot. Recording results will change however, depending on movement speed and timing. The provided macros are listed in Table 48.1.

Table 48.1 Auto Backward Exit Macros

MACRO	DESCRIPTION
Rec Path Start	Starts recording of robot path.
Rec Path End	Stops recording if currently recording.
Rec Path Pause	Pauses recording of the robot path until resumed or ended.
Rec Path Resume	Resumes paused path recording.
Rec Path Clear	Clears existing recorded path
Do Bwd Exit	Starts exit motion backward along the recorded path.

48.2 Setup

Auto Backward Exit can be set up and configured via the Auto Bwd Exit menu entry at Controlled Start. Refer to Table 48.2 for information on each item you can set. Changes take effect when F3, APPLY is pressed, or at the next Controlled Start. Some setup items can also be set at Cold Start.

Table 48.2 Auto Exit Setup Items

ITEMS	DESCRIPTION
Controlled Start Setup Items	
Enable Auto Exit: (TRUE or FALSE)	Enables the feature and allocates recording memory or disables feature and deallocates memory.
Max Record Distance: (1–12000 mm)	Defines the maximum path recording length. Longer distances use more memory.
Recording Interval: (1–100 mm)	Defines the point recording interval during robot motion recording. Shorter distances use more memory.
Start From Closest Node: (YES or NO)	When set to YES, backward exit path starts from the recorded node closest to robot's current position. When set to NO, backward exit path starts from last recorded node.
Move to Node using F3 or DI[#] (0 = not enabled)	Defines a DI[] that is used for user-controlled motion start of backward exit path when system is in AUTO mode.
Warning Distance: (0-1000 mm)	When Start From Closest Node: is set to YES, define a threshold distance between robot's current position and the closest recorded node beyond which warning screen of Fig. 48.3 (c) appears and user-controlled motion is required. When Start From Closest Node: is set to NO, define a threshold distance between robot's current position and the last recorded node beyond which prompt screen of Fig. 48.2 (c) appears.
Warning Angle: (0-360 deg)	Define a threshold angle difference between robot's current oriantation and that of the last recorded node beyond which warning screen of Fig. 48.3 (c) appears and user-controlled motion is required.

Cold Start or Controlled Start Setup Items	
ITEMS	**DESCRIPTION**
Auto Exit Speed: (1–250 mm/sec)	Defines the robot speed for backward exit motion up to 250 mm/sec.
Warning Distance: (0-1000 mm)	When Start From Closest Node: is set to NO, this item can be set in Cold Start.
G1 Available DO[1–4096] (0 = not enabled)	A digital output can be turned on when a path has been recorded for this group.
G1 Executing DO[1–4096] (0 = notenabled)	A digital output can be turned on when 'Do Bwd Exit' is executing for this group.



Fig. 48.2 (a) Controlled Start Setup Menu

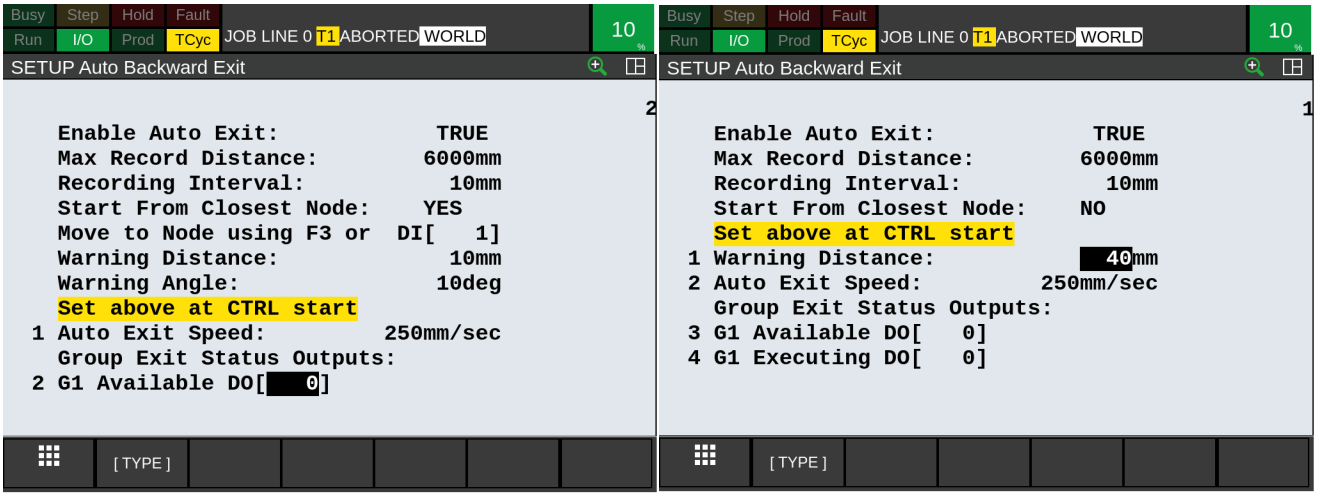


Fig. 48.2 (b) Cold Start Setup Menu

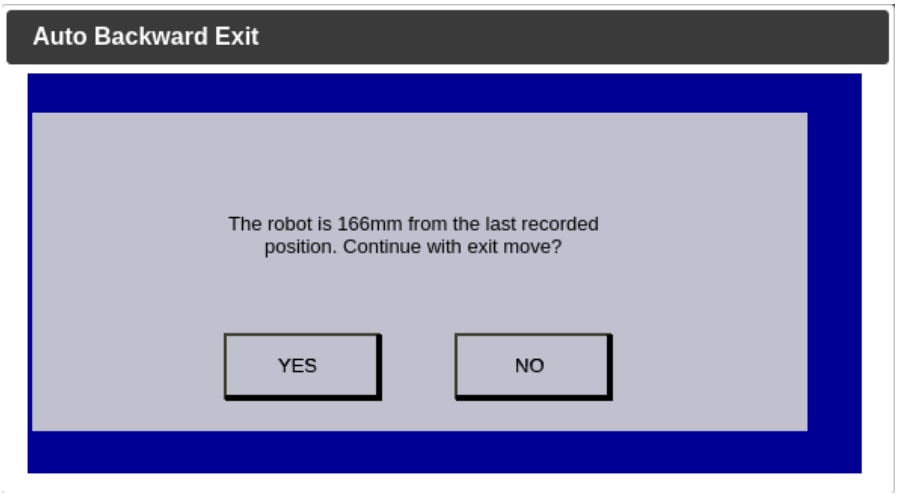


Fig. 48.2 (c) Warning Screen when Start From Closest Node: NO

Procedure to Set Up Auto Backward Exit

1. Perform a Controlled Start. (Turn the power ON with "Prev" and "Next" keys pressed, then select "3 Controlled start")
2. Press MENU, select 0 until you can select Auto Bwd Exit, and press ENTER.
3. Press F4, TRUE to set Auto Backward Exit to TRUE. Set all other items as necessary. Refer to Table 48.2 .
4. When you are done, press F3, APPLY.

NOTE

Additional group status outputs will appear on multi group systems. If multiple groups are recorded in one teach pendant program, only the status outputs for the lowest group are set.

RECORDING NOTES

On a Multi-arm system, positional data will be recorded for all groups / axes specified by the recording program's group mask. All groups will move together when executing the backward exit motion.

While the a path is being recorded, the system will only record programmed motion. Other motions such as jog motion, or the motion from original path resume, etc. will not be recorded. Recording is only based on the set recording interval, and as such time or speed are not factors. The only exception is where a higher speed changes the forward path of the robot. In these cases, the backward path will mimic the recorded forward path, but at the backward exit speed.

If the recorded path exceeds the maximum distance, the system will fault with the message "Reached Maximum Path Distance". Auto Bwd Exit is still available, but recording cannot be continued.

When the Rec Path Clear macro is executed, any recorded path data becomes unavailable, so recovery cannot be executed until recording is performed once again.

48.3 Programming

Auto Backward Exit can be implemented in different ways for a wide variety of error recovery situations.

The following are simple examples of how the feature can be used.

Example Program #1 : Make it easy to teach exit motion

In the first example shown in Fig. 48.3 (a), the robot starts recording at P[2] before moving to P[3]. At the end of the program, 'Do Bwd Exit' macro will move the robot all the way back to P[2] where recording started. The program would then continue to P[1]. Note that recording is not stopped at any point during this program. Therefore, this is an example where Auto Backward Exit is used as a convenient way to program the robot to execute its most recent motion in reverse without having to insert the previously taught points.

```
 1: L    P[2:Outside]  500mm/sec  CNT50
 2:
 3:  ! Start recording
 4: Rec Path Start
 5: L    P[3:Close 1]  250mm/sec  CNT 25
 6: L    P[4:Close 2]  180mm/sec  CNT 10
 7: L    P[5:Drop]  50mm/sec  FINE
 8: CALL DROP_PART
 9: ! Initiate backward exit
10: Do Bwd Exit
11: J    P[1:HOME]  100%  FINE
```

Fig. 48.3 (a) Example Program #1

Example Program #2 : Start exit motion from recorded exit path

It is also possible to stop or pause the recording, and execute further motion before running Do Bwd Exit, as shown in Fig. 48.3 (b) .

In this case, when the Do Bwd Exit macro is executed, the software will first check the setting of the “Start From Closest Node” item from the setup menu (which can only be set at Controlled Start).

Note that “Node” in this menu item does NOT refer to a taught point in the TP program; it refers to a position that was recorded by Auto Backward Exit while running the program with recording on.

If this item is set to NO, the robot will move through the entire recorded path, beginning with the last recorded point. If the distance from the robot’s current position to the last recorded point is beyond the distance specified in the setup menu item “Warning Distance”, then a warning will appear on the TP, giving the user the option of whether or not to still execute the backward path.

If the “Start from Closest Node” item is set to YES, then the robot will move through the recorded path beginning at the recorded position that is closest to the robot’s current position. If the distance between the current robot position and the closest exit path recorded position is larger than what is specified in the setup menu, or if the orientation difference between these positions is larger than what is specified in the menu, then the screen shown in Fig. 48.3 (c) will appear.

```
 1: L    P[2:Outside]  500mm/sec  CNT50
 2:
 3:  ! Start recording
 4: Rec Path Start
 5: L    P[3:Close 1]  250mm/sec  CNT 25
 6: L    P[4:Close 2]  180mm/sec  CNT 10
 7: L    P[5:Drop]  50mm/sec  FINE
 8: CALL DROP_PART
 9: Rec Path End
10: L    P[6:Drop]  50mm/sec  FINE
11: ! Initiate backward exit
12: Do Bwd Exit
13: J    P[1:HOME]  100%  FINE
```

Fig. 48.3 (b) Example Program #2

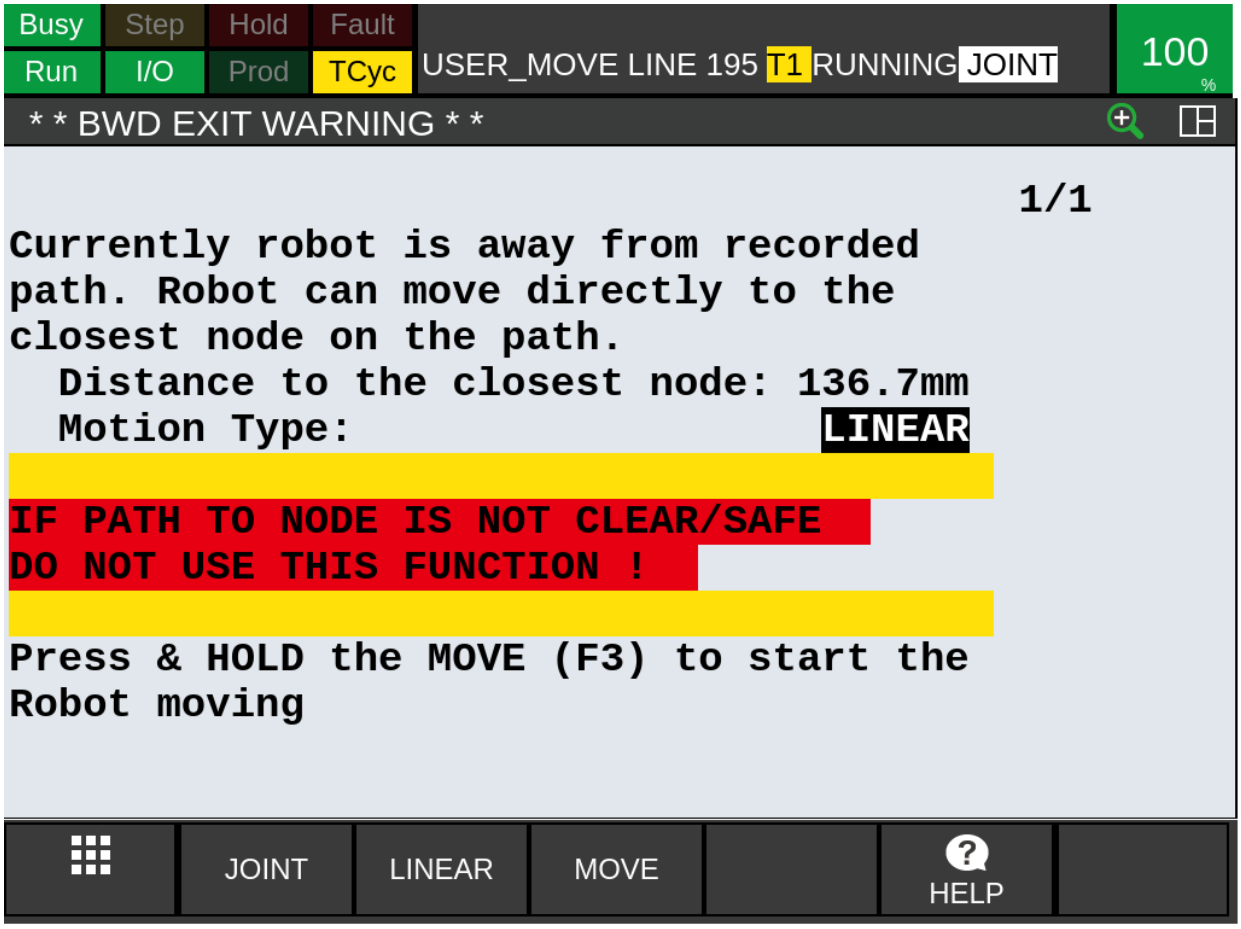


Fig. 48. 3 (c) BWD EXIT WARNING

As long as this screen is present, if system is in T1 or T2 mode, the user must keep the F3 key depressed on the Teach Pendant in order to allow the robot to move. If system is in AUTO mode, the DI[] specified in the setup menu must turn on and stay on in order to allow the robot to move. This condition and screen will persist until the distance and/or orientation difference mentioned above falls below the thresholds specified in the setup menu, at which time the robot will execute the remainder of the exit path to completion without the need for user input. Note that this screen defaults to displaying LINEAR as shown, but it can be changed to JOINT by the F1 key, in case the user prefers the robot to move to the exit path using joint motion instead of linear.

Note that this behavior would be the same whether line 9 in Program #2 from Fig. 48.3 (b) uses the Rec Path End or Rec Path Pause macro. That is because in both cases the software is no longer recording after that line, therefore the robot position will be away from the recorded exit path.

Example Program #3 and #4 : Separate recording program and exit program

The examples so far have shown the Do Bwd Exit macro being inserted in the same program as the path recording macros such as Rec Path Start, Rec Path End, etc. However, this is not required. Do Bwd Exit could be inserted into a separate program and executed from there. For example, consider the programs in Fig. 48.3 (d) .

In Fig. 48.3 (d), Program #3 could be a main program running in production, and Program #4 could be a recovery program. In this case, Program #3 could be executing and an error could occur, leading the program to be paused and then aborted by the user. In this case, the user may or may not jog the robot after aborting the program. Then, the user could execute Program #4 to recover.

If Program #3 was aborted before Rec Path End had executed, and the user had not moved the robot, then the behavior will be the same as for Program #1. That is, when the Do Bwd Exit macro finally executs, the robot will move through the recorded Backward Exit Path until the robot reaches P[2].

If Program #3 was aborted after the Rec Path End had executed, or the user jogged the robot before executing Program #4, then the behavior is the same as explained earlier for Program #2. That is, the same warning screens will appear with the same required user input, before the Backward Exit motion is executed. Note that in this example from Fig. 48.3 (d), in order to execute Program #4 for recovery, Program #3 must be ABORTED.

Example Program #3

```
 1: L    P[2:Outside]  500mm/sec  CNT50
 2:
 3:  ! Start recording
 4: Rec Path Start
 5: L    P[3:Close 1]  250mm/sec  CNT 25
 6: L    P[4:Close 2]  180mm/sec  CNT 10
 7: L    P[5:Drop]  50mm/sec  FINE
 8: CALL DROP_PART
 9: Rec Path End
10: J    P[1:HOME]  100%  FINE
```

Example Program #4

```
 1: OVERRIDE=45 %
 2:
 3: ! Initiate backward exit
 4: Do Bwd Exit
 5: DO[1] = PULSE, 0.5sec
```

Fig. 48.3 (d) Example Program #3 and #4

Example Program #5 and #6: Pausing and resuming recording

The next two examples in Fig. 48.3 (e) and Fig. 48.3 (g) show how Rec Path Pause and Rec Path Resume could be used. In Program #5, none of the motion from P[3] to P[4] to P[5] and back to P[3] in lines 8-10 will be recorded as part of the exit path. So, when Do Bwd Exit is executed at line 15, the backward exit motion will behave as shown in Fig. 48.3 (f).

```
 1: L   P[1:Outside]  500mm/sec  CNT50
 2:
 3: ! Start recording
 4: Rec Path Start
 5: L   P[2:Close 1]  250mm/sec  FINE
 6: L   P[3:Close 2]  180mm/sec  FINE
 7: Rec Path Pause
 8: L   P[4:PERCH 1]  100mm/sec  FINE
 9: L   P[5:PERCH 2]  100mm/sec  FINE
10: L   P[3: Close 2]  100mm/sec  FINE
11: Rec Path Resume
12: L   P[6]  100mm/sec  CNT25
13: L   P[7]  100mm/sec  CNT25
14: ! Initiate backward exit
15: Do Bwd Exit
```

Fig. 48.3 (e) Example Program #5

In Fig. 48.3 (f), the blue dotted line is the main path executed before Do Bwd Exit, and the red line is the exit path. In this figure, it can be seen that the motion between Rec Path Pause and Rec Path Resume is skipped during Backward Exit motion.

Also note the “X” in Fig. 48.3 (f). If execution of the main program is stopped and aborted at this location while executing line 12, the Do Bwd Exit macro could be run separately as described in Examples 3 and 4. In this case, if the robot is not jogged away, the Backward exit motion would still move along the red line in Fig. 48.3 (f), starting at the location of “X”.

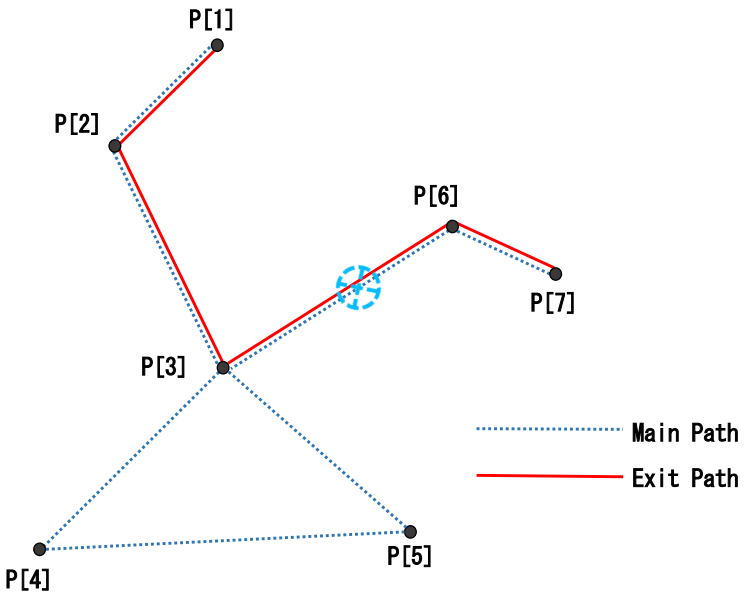


Fig. 48.3 (f) Main Path and Exit Path for Example Program #5

Note that it is not required to execute Rec Path Pause and Rec Path Resume at the same physical location as shown in Fig. 48.3 (e). However, if the user does not program it this way, then the Backward Exit Motion is affected as shown in the next example.

```
 1: L   P[1:Outside]  500mm/sec  CNT50
 2:
 3: ! Start recording
 4: Rec Path Start
 5: L   P[2:Close 1]  250mm/sec  FINE
 6: L   P[3:Close 2]  180mm/sec  FINE
 7: Rec Path Pause
 8: L   P[4:PERCH 1]  100mm/sec  FINE
 9: L   P[5:PERCH 2]  100mm/sec  FINE
10: Rec Path Resume
11: L   P[6]  100mm/sec  CNT25
12: L   P[7]  100mm/sec  CNT25
13: ! Initiate backward exit
14: Do Bwd Exit
```

Fig. 48.3 (g) Example Program #6

In Fig. 48.3 (h), once again the blue dotted line is the main path executed before Do Bwd Exit, and the red line is the exit path. But now when the Do Bwd Exit macro on line 14 is executed, because Rec Path Pause and Rec Path End are at different physical positions, during the backward exit motion, the robot will move linearly from P[6] to P[3], and that motion was not part of the original main path.

Also, if program execution is stopped and aborted at the “X”, and then Do Bwd Exit is executed from a separate program as in Examples 3 and 4, then in this example the warning screens will be shown even if the robot is not jogged away.

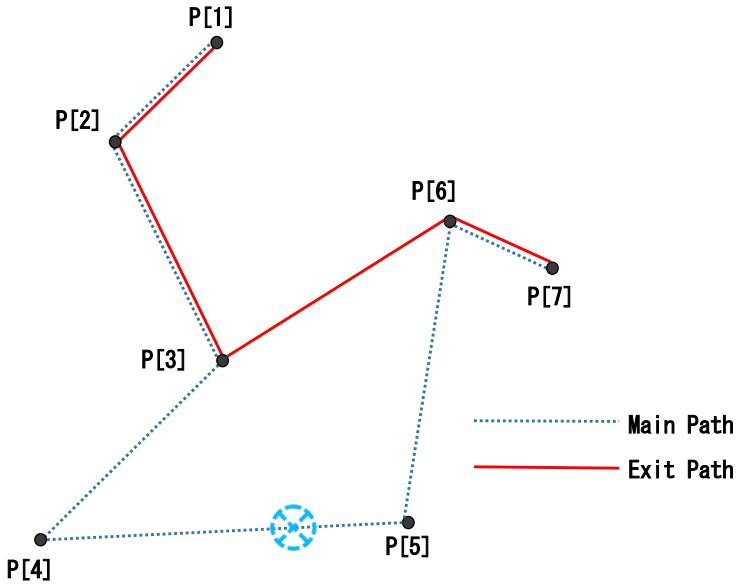


Fig. 48.3 (h) Main Path and Exit Path for Example Program #6

Example Program #7 : Clear recorded exit path

Finally, there may be some cases where an exit path has been recorded, but subsequent motions put the robot into a position in which it is actually not safe or not desirable to run the exit path. In such a case, the user may want to clear the recorded path to make sure it cannot be used accidentally. The Rec Path Clear macro can be used for this purpose, as shown in the example program in Fig. 48.3 (i).

```
 1: L    P[2:Outside]  500mm/sec  CNT50
 2:
 3:  ! Start recording
 4: Rec Path Start
 5: L    P[3:Close 1]  250mm/sec  CNT 25
 6: L    P[4:Close 2]  180mm/sec  CNT 10
 7: L    P[5:Drop]  50mm/sec  FINE
 8: CALL DROP_PART
 9: Rec Path End
10: L    P[6]  100mm/sec  CNT 100
11: L    P[7]  100mm/sec  CNT 100
12: L    P[8]  100mm/sec  CNT 100
13: Rec Path Clear
14: J    P[1:HOME]  100%  FINE
```

Fig. 48.3 (i) Example Program #7

In this example program, if the program were stopped and aborted at any point before line 13, the Do Bwd Exit macro could be executed to run the exit path as described in earlier examples. However, once the Rec Path Clear macro is executed on line 13 of this program, if the user then attempts to execute Do Bwd Exit, it will post an error and the robot will not move, as if no path had been recorded. The user would place the Rec Path Clear macro in their program based on where they feel it is no longer safe to use the exit path.

Note that Rec Path Clear assumes that Rec Path End has already been executed at some time. If that is not true, and Rec Path Clear is executed while the software is still currently recording the exit path, then a warning alarm is posted, program execution continues and the exit path already recorded will not be cleared.

NOTE
Only one recorded exit path is maintained in memory at any given time. So when Rec Path Start is executed, if a recorded path already exists at that time, it will be cleared and replaced with the new recording.

49 EXPANDED REGISTERS FUNCTION

The Expanded Registers function expands the maximum number of numeric registers to 5,000 and position registers to 2,000. However, the maximum number of position registers depends on the number of groups. Refer to Table 49 for the maximum numbers of numeric and position registers based on the number of groups.

Table 49 Maximum numbers of registers based on the number of motion groups

Groups	Position registers	Numeric registers
1	2000	5,000
2	1000	
3	666	
4	500	
5	400	
6	333	
7	285	
8	250	

When this option is installed, the numbers of numeric and position registers are expanded as shown in Table 49 automatically. The expanded numbers of registers are available in all register instructions. Refer to Section 3.4 REGISTER INSTRUCTIONS in the *FANUC Robot series OPERATOR'S MANUAL (Basic Operation)* (B-84464EN) for details on numeric and position registers.

NOTE
The Expanded Registers function requires 3MB of NVRAM memory.

49.1 SETTING THE NUMBER OF REGISTERS

The maximum numbers of numeric and position registers can be changed. However, the numbers cannot be set beyond the numbers listed in Table 49.
Use Procedure 49-1 to set the number of registers.

Procedure 49-1 Setting up the maximum number of numeric and position registers

Step

1 Turn ON the controller with the PREV and the NEXT keys pressed. Then select Controlled start.
2 Press MENU.
3 Select 0 – NEXT -- and select PROGRAM SETUP, to display the Program Limits screen.

Program Limits screen



4 Move the cursor to numeric registers or position registers and enter a new value.
5 Perform a cold start.

> **⚠WARNING**
> If the number of numeric or position registers is decreased, the reduced register data will be lost. For example, if the number of numeric registers is reduced from 5,000 to 4,000, the data in numeric registers data 4,001 to 5,000 will be lost.

49.2 SAVING AND LOADING FILES

Executing backup all on the FILE screen saves numeric register and position register data as files.
When backup all is executed, numeric register data is saved in a file named NUMREG.VR and position register data is saved in a file named POSREG.VR. When you load these files, numeric and position register data is overwritten to the saved data. Refer to Section 7.4 SAVING FILES and 7.5 LOADING FILES in the *FANUC Robot series OPERATOR'S MANUAL (Basic Operation)* (B-84464EN) for details.

However, there are some limitations when loading these files. These limitations are described in the following subsections.

49.2.1 LOADING A NOT EXPANDED xxxREG.VR FILE

NUMREG.VR and POSREG.VR files saved in a system without the Expanded Registers (R830) function can be loaded into a system that has the Expanded Registers (R830) function.

However, the number of numeric and position register files will be converted to the number of registers in the loaded files.

> **⚠CAUTION**
> When you load these files, if the warning message "VARS-206 ***REG.VR will load at startup" is displayed, you must reboot the controller to apply the register data.

NOTE
The number of registers can be changed again on the Program Limits screen at controlled start. Refer to Subsection 49.1 SETTING THE NUMBER OF REGISTERS for more information.

49.2.2 LOADING AN EXPANDED xxxREG.VR FILE

Loading to the system that has the Expanded Registers option

NUMREG.VR and POSREG.VE files saved in a system with the Expanded Registers (R830) option will be allowed to load in a system that also has the Expanded Registers (R830) option.

However, if the file to be loaded has been saved on a system in which the number of registers has been changed, the number of numeric and position registers will be converted to the number of numeric and position registers in the loaded file.

⚠ CAUTION
When you load these files, if the warning message “VARS-206 ***REG.VR will load at startup” is displayed, you must reboot the controller to apply the registers data.

NOTE
The number of registers caqn be changed again on the Program Limits screen at controlled start. Refer to Subsection 49.1 SETTING THE NUMBER OF REGISTERS for more information.

Loading to the system that does not have the Expanded Registers option

NUMREG.VR and POSREG.VE files saved in a system that has the Expanded Registers (R830) option are disabled when loaded into a system that does not have the Expanded Registers (R830) option.

However, if the system loading the Expanded Registers has the numbers of registers set to the values listed in Table 49.2.2, the saved file will be allowed to load.

Table 49.2.2 The maximum numbers of registers in each group system

Groups	Position registers	Numeric registers
1	372	999
2	372	
3	248	
4	186	
5	148	
6	124	
7	106	
8	93	

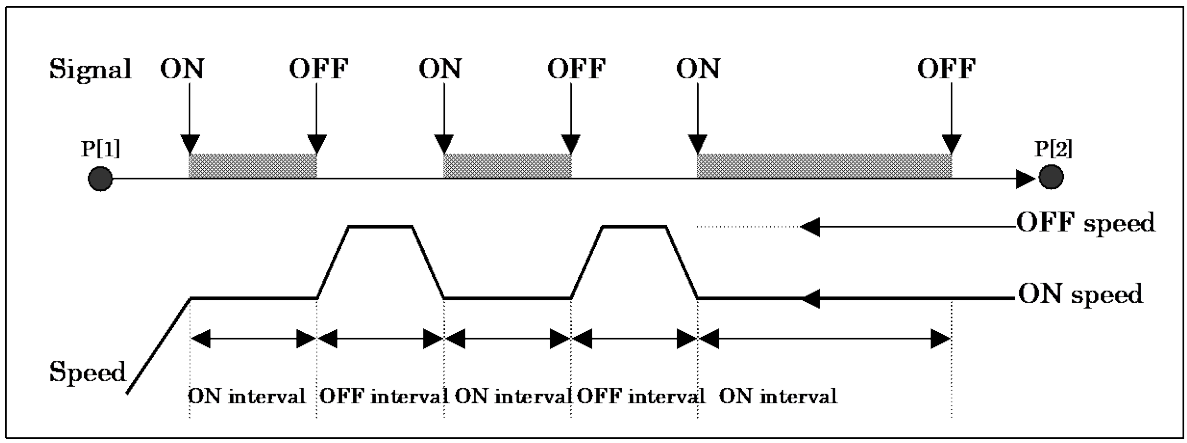
The numbers of numeric and position registers are converted to the numbers of registers in the loaded files.

⚠ CAUTION
When you load these files, if the warning message “VARS-206 ***REG.VR will load at startup” is displayed, you must reboot the controller to apply the registers data.

NOTE
The number of registers can be changed again on the Program Limits screen at controlled start Refer to Subsection 49.1 SETTING THE NUMBER OF REGISTERS for more information.

50 STITCH FUNCTION

The Stitch function enables you to easily teach stitch-like sealing and laser welding applications as on and off intervals by turning a signal ON and OFF at each Stitch interval.



50.1 SPECIFICATIONS

50.1.1 Instructions

Use the following instructions to specify the interval in which the Stitch function is used.
This interval will be referred to as the Stitch interval from now on.

- **STITCH[i]**
  - Starts the Stitch process (change of speed, signal control) based on a Stitch schedule.
  - The Stitch process continues until the STITCH END instruction is executed.
  - Stitch schedules are specified on the STITCH DATA screen, described in Section 50.1.2.
  - The STITCH instruction can be taught only as a single instruction. It cannot be taught as a motion modifier.
- **STITCH END**
  - Stops the Stitch process and turns off the signal.

Example

```
1: J P[1] 100% FINE
2:    STITCH[1]
3: L P[2] 2000mm/sec CNT100
4: L P[3] 2000mm/sec FINE
5:    STITCH END
```

Stitch interval.
Only linear and circular motions are allowed. If joint motion type is used, an error is posted.

50.1.2 Stitch Schedules

- You set Stitch schedules on the following STITCH DATA screens: STITCH DATA LIST, STITCH DATA DETAIL, and STITCH DATA COMMON. Refer to 50.4 APPENDIX 1 for the screen structure.
- The screen items are described in the following sections.

STITCH DATA LIST screen

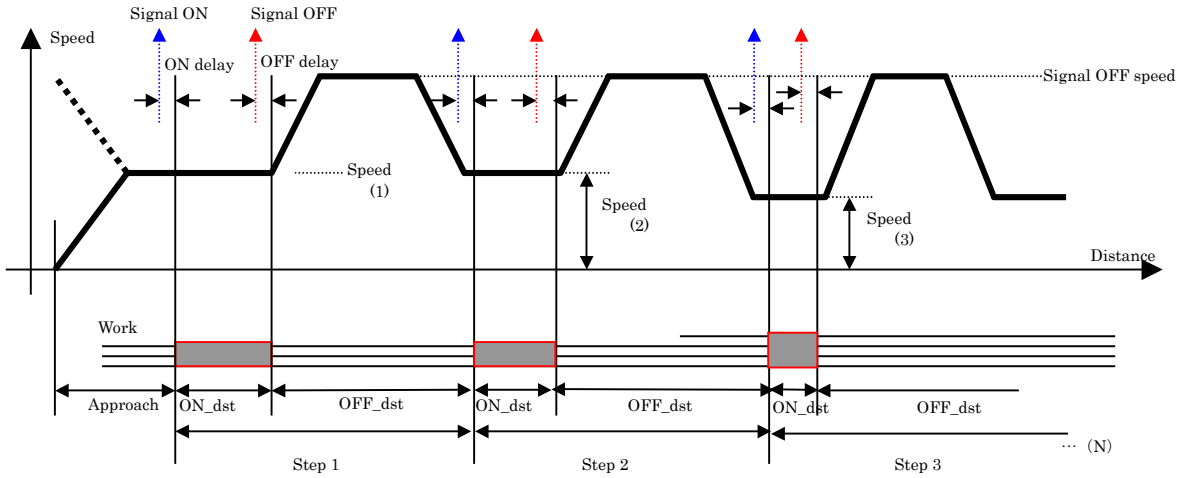
This screen displays the list of Stitch schedules. The STITCH instruction specifies the schedule using the number (1-20) displayed on this screen. To display the DETAIL screen foe each schedule, move the cursor to a schedule and press F4, DETAIL.

Function Key	Description
F3（COMMON）	This key displays the STITCH DATA COMMON screen.
F4（DETAIL）	This key displays the DETAIL screen for the selected schedule (the schedule at the current cursor position).
F5（COPY）	This key allows you to copy the selected schedule to another schedule. **CAUTION: Even if the conditions for the selected schedule to be copied have already been set, they will be overwritten.**

STITCH DATA DETAIL screen

◆ You can set the following items on this screen.

Item	Description	Range
Comment	This item displays the comment for each schedule.	
Equipment condition	This item is the type of equipment signal (DO/RO/GO/AO) and index number.	DO/RO/GO/AO 1-32766
Signal ON delay time	This item is the delay time between the signal turning ON and the equipment starting.	0-9999[msec]
Signal OFF delay time	This item is the delay time between the signal turning OFF and the equipment stopping.	0-9999[msec]
Approach distance	This item is the approach distance to stabilize the speed.	0-9999[mm]
Signal OFF speed.	This item is the speed in the OFF intervals. The units are specified in the Speed unit item on the STITCH DATA COMMON screen.	0.1 - max speed [mm/sec]
Specify the following items for each step.		
ON_dst	This item specifies the distance of the signal ON interval	0-9999[mm]
Speed	This item specifies the speed of the signal ON interval. The units are specified on the STITCH DATA COMMON screen.	0.1 - max speed [mm/sec]
OFF_dst	This item specifies the distance of the signal OFF interval.	0-9999[mm]
Count	This item specifies how many times this step is executed. If you want to repeat a sequence, enter the number of repetitions. If set to 0, this step will be ignored and the next step with a non-zero counter will be executed.	0-99[times]
Cond.	This item specifies the value to be output to the equipment condition signal. This value is output in the ON interval, and 0 is output in the OFF interval. If the equipment signal is DO or RO, set this value to 1. For GO, the value specified here is output as a binary code.	1-9999
F2（DATA）	This key displays the STITCH DATA DETAIL screen for a different schedule.	
F3（LIST）	This key returns to the STITCH DATA LIST screen.	



NOTE

- Do not change the ON and OFF speeds while the program is running. Pause the program and then change the speed. If you change the speed during program execution, the robot may not operate at the specified speed.
- When stitch motion is performed by a controller installed with the TCP Speed Estimation function, the warnings “TCPP-018 Begin Error Mode at line” or “TCPP-019 Speed Ovrd Mode at line” could occur. No remedies are required for these warnings.

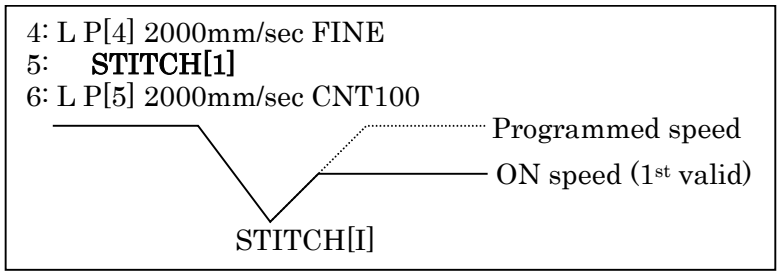
STITCH DATA COMMON screen

◆ You can set the following items on this screen.

Item	Description	Range
Stitch enable	The stitch process (speed change and signal control) is performed only when the specified signal is ON. When this signal is OFF, the motion speed in the stitch interval is the programmed speed. If the index is set to 0, the signal is ON.	DI/RI 1-32766
Equipment enable	The equipment signal is turned on only when the specified signal is ON. If this signal is OFF, only speed control is performed. If the index is set to 0, the signal is ON.	DI/RI 1-32766
Speed unit	This item specifies the units of the Signal OFF speed and ON speed.	cm/min mm/sec inch/min

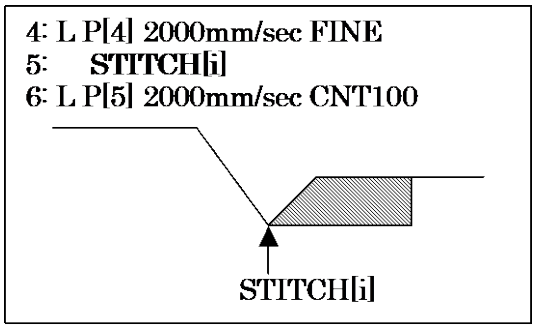
50.1.3 Stitch Process Flow

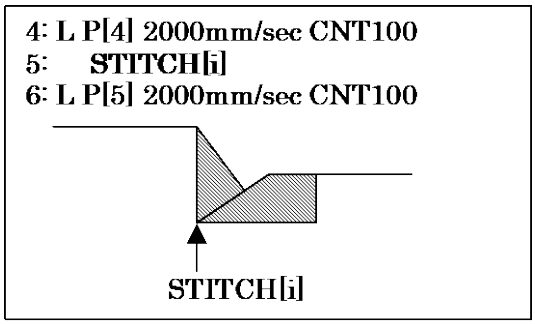
1 After executing a STITCH[i] instruction, the TCP speed is changed to the ON speed of the first valid step of Stitch Schedule i. A valid step is a step with a non-zero count.
The speed specified by the program has no meaning.



2 When the distance the TCP traveled after the STITCH[i] instruction exceeds {approach distance – ON delay (converted to distance)}, the controller switches the signal ON.

NOTE
If a STITCH[i] instruction is executed before motion execution has completed, such as a STITCH[I] instruction just after CNT motion, the approach distance will be shorter than expected.





NOTE
ON delay time is interpreted after conversion to distance. ON delay time is converted to the distance that the TCP travels at the ON speed with 100% override. The timing of the signal output is based on this converted ON delay time. For this reason, be sure to adjust the ON delay time with 100% override.

3 After this, the speed is changed, and the signal is turned ON/OFF according to the total distance the TCP traveled.

4 If the motion instruction doesn't complete after completion of all steps, TCP moves at OFF speed until STITCH END instruction is executed.

50.1.4 Other Specifications

◆ Several motions in a Stitch interval.
The Stitch function does not care how many motion instructions are in a Stitch interval. In the following example, the TCP moves from P[1] to P[3] according to Stitch schedule 1.

```
1: J P[1] 100% FINE
2:    STITCH[1]
3: L P[2] 2000mm/sec CNT100
4: L P[3] 2000mm/sec FINE
5:    STITCH END
```

◆ Changing the Stitch schedule in a Stitch interval.
When a STITCH[i] instruction is executed in a Stitch interval, a new Stitch schedule is used from that point on. The approach distance for the new Stitch schedule is also executed.

- Pause and resume
  When the program is paused, the equipment signal is turned OFF.
  After resuming, the rest of the Stitch sequence is executed. However, the timing of the Stitch might be different from the non-paused case due to the effects of acceleration and deceleration.

- Single step execution
- Backward execution
- Aborting a of program
- Changing the program line

If any of the preceding actions are performed when a Stitch is in process, the Stitch function will be automatically disabled and the equipment signal will be turned OFF.
If the Stitch function is disabled, the programmed speed will be used.
After resuming, the Stitch process will not be performed. To use the Stitch function, move the cursor to one line before the STITCH[i] instruction and run the paused (aborted) program.

- T1 Mode

If one of the followings is executed in T1 mode, the error INTP-427 WARN (program, line) Stitch disable(T1 mode) will occurs:
- execute STITCH[]
- Pause during Stitch, then change to T1 mode and resume

NOTE
If the program is paused during the Stitch process, then the execution mode is changed to T1 and the program resumed, the resume speed will be the same speed as when the Stitch function is enabled.

- Saving Stitch schedules
  Stitch schedules can be saved to the following file using “All of above” on the FILE screen: SYSSTCH.SV

50.2 ADJUSTMENTS

Speed changes require acceleration and deceleration time. For this reason, to maintain the specified speed in the ON interval, it is desired to start deceleration at the appropriate timing. Robot speed depends on the specified speed. Appropriate acceleration and deceleration time depends on the robot posture, even if the specified speed is the same. Keep in mind the following information when changing the speed:

- Timing of deceleration from the OFF interval to the ON interval
  By advancing the timing of speed switching from the OFF section to the ON section, the motion speed of the ON section can be attained earlier.
  To make the adjustment, press the MENU and open the SYSTEM / CONFIG Stitch Function screen.
  The following parameters are available:

Division Number 1-10（default value: 2）
The OFF interval is divided into the number of sections specified here in order to change the deceleration rate in divisions.

Speed Rate [1-10] 1-100（[1]=100, [2]=0）
For each division, the speed is set to the following speed.

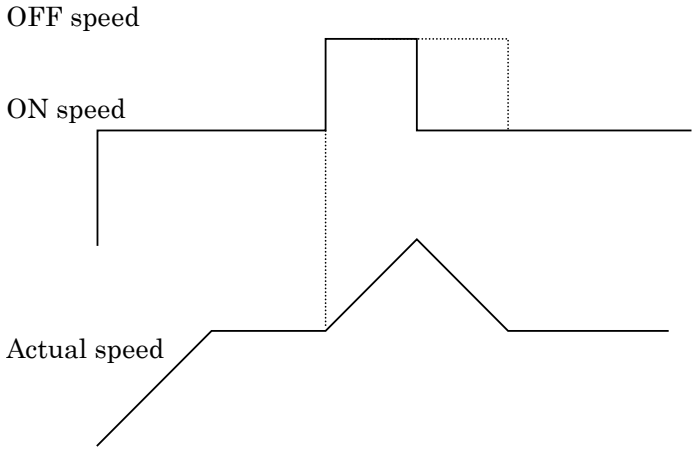
ON speed + (OFF speed – ON speed) * Speed Rate / 100

This means that Speed Rate=100 is the OFF speed and Speed Rate=0 is the ON speed.

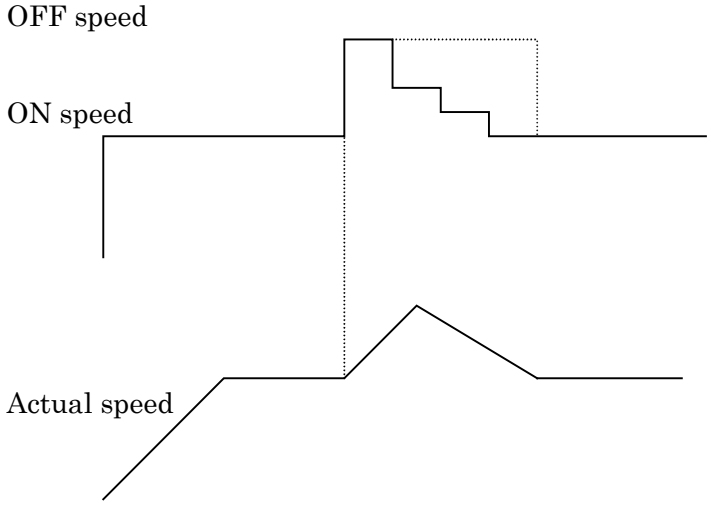
NOTE
The DEC_RATE value should be in the range from 0 to 100.

Division Number and Speed Rate [ ] Examples

Division Number =2
Speed Rate [1]=100, [2]=0



Division Number =4
Speed Rate [1]=100, [2]=50, [3]=25, [4]=0



50.3 SYSTEM CONFIG MENU

Stitch function displays on Program Instructions of System config menu when “Stitch Function” (J683) is installed. This item allows you to set up Division Number and Speed Rate.

Setting hierarchy 2	Setting hierarchy 3	Descriptions

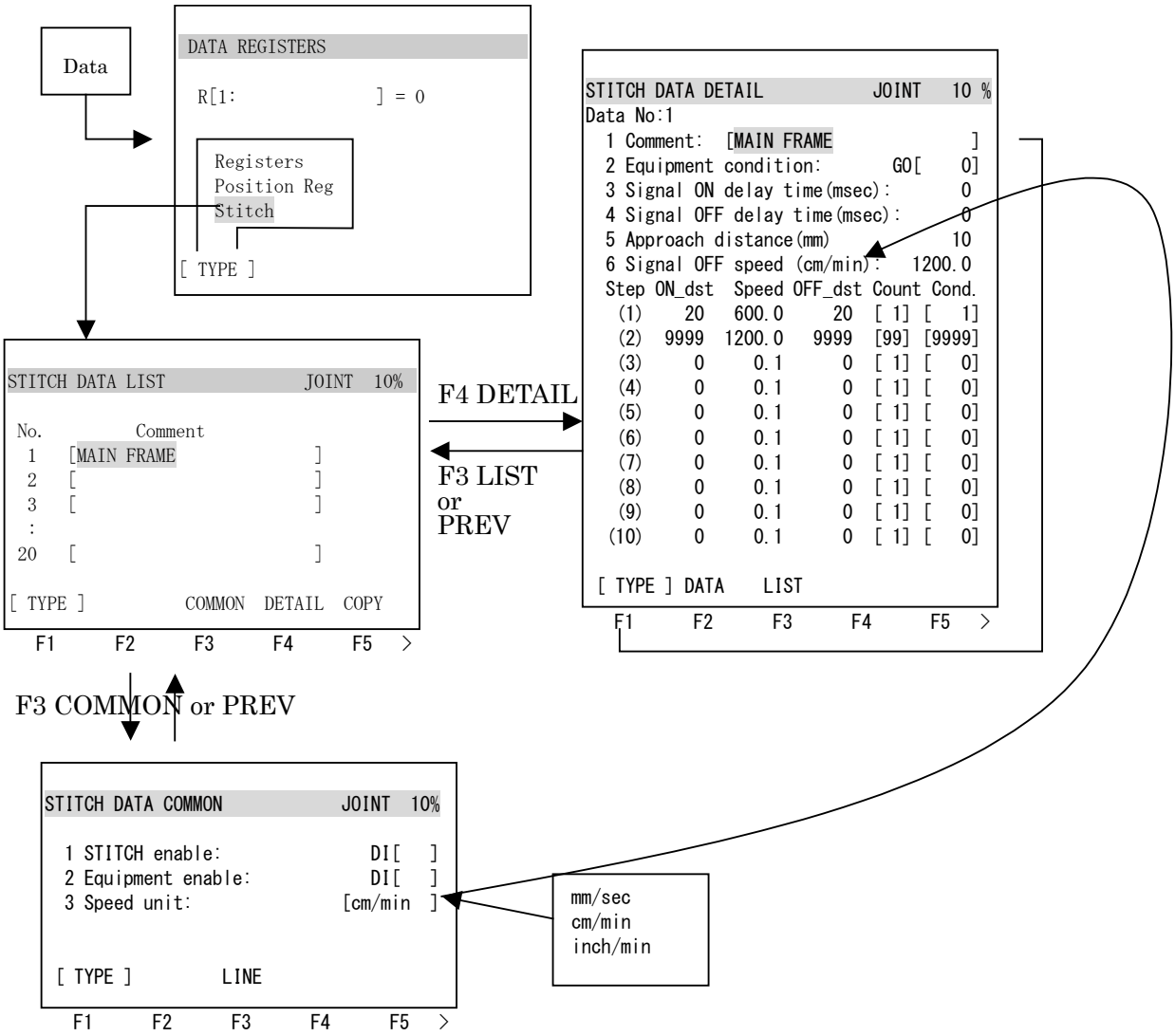
Stitch Function	Division Number	The number of Division where the Speed rate is changed from the OFF section to the ON section.
	Speed Rate	This is the rate of speed for each division.

50.4 LIMITATIONS

- After an E-stop, distance cannot be calculated correctly. After resuming, the distance that the TCP moved is calculated. However, there may be errors due to overrun after E-stop.
- JOG does not accumulate the distance of Stitch intervals.
- The operations are not available in a Stitch interval:
  - JOINT motion
  - Changing the tool frame or user frame.
- The Stitch function cannot be used with the following functions:
  - ✓ Continuous Turn
  - ✓ Line tracking
  - ✓ Coordinated motion
  - ✓ Remote TCP
  - ✓ Robots that do not have Cartesian coordination (Independent axes)
  - ✓ Extended axes
  - ✓ Distance Before (In stitch interval, Distance Before is not available.)
  - ✓ Constant Path function

50.5 STITCH FUNCTION SCREEN STRUCTURE

Appendix1 Structure of screens for stitch function



51 VISUAL DIAGNOSTICS

Visual Diagnostics uses the *i*Pendant animated picture display capability to provide a rich visual presentation of status information. A Visual Diagnostic (VD) screen is one that displays a reference image as the background, and small indicator images strategically placed and activated in the foreground. The appearance of the indicator images is driven by the state of information within the controller. This can be useful, for example, in an application in which the robot is welding. Each weld position could be monitored and visually represented so that the success or failure of a weld is reported by green and red indicators. See Fig.51 for another example.

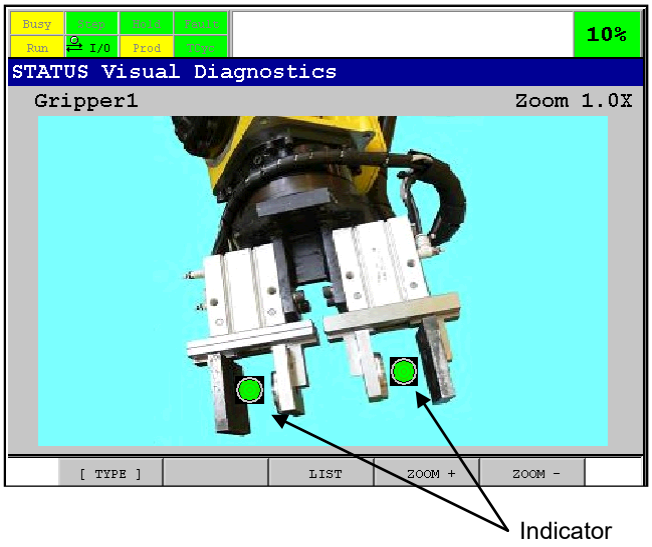


Fig.51 Simple Gripper Example

The background is a picture of a gripper with several actuators. In the foreground near many of the actuators are status indicators which are pictures of an LED to indicate the status of the actuator. The LED is green when the actuator is open and red when it is closed.

The foreground indicators are not limited to pictures of red and green LEDs. In fact, each indicator can have multiple states, each with a unique picture to display the state graphically. For example, a clamp indicator could use images of the clamp in the opened, closed and intermediate positions. If properly scaled and oriented, the clamp on the VD screen would appear to represent the configuration of the clamp accurately.

An indicator is displayed on or near an associated detail in the background image to specify its status. An indicator can have up to four states and each state can be represented by a different image. A Visual Diagnostic screen can have as many indicators as you want. However, be careful when adding indicators because each takes controller resources in terms of space and time, as well as screen clutter.

An indicator can have up to four states. An indicator's state is defined by the value of its one or two state control elements at run-time. If only one state control element (SCE) is defined (ie: number 2 is marked "**") then it can be a Boolean I/O (0=False, 1=True), Numeric I/O, Data Register or *i*Rvision Register.

When the state control element is a vision register VR[x], the model ID from view 1 is used. If, at run-time, a numeric state control element evaluates to a number other than 0 through 3, the base 4 modulus of the number is used. If both state control elements are used, then they must both be Boolean I/O types. The indicator state is then determined as shown in Table 51.

Table 51 Indicator States

SCE2	SCE1	State
0	0	0
0	1	1
1	0	2
1	1	3

Each of the four indicator states has an associated image that is displayed. If there is no state image associated with the number evaluated, the no image is shown and the background image shows through. The State Image: fields enable you to set up this state-to-image relationship.

51.1 CREATING AND EDITING VISUAL DIAGNOSTIC SCREENS

You can create custom Visual Diagnostic screens on the robot controller. Table 51.1(a) and Table 51.1(b) list items you can set when setting up a Visual Diagnostic screen. Use Procedure 51-1 to create and edit a Visual Diagnostic screen.

Procedure 51-1 Creating and Editing Visual Diagnostic Screens

Conditions

- You have graphic files accessible on the controller that you want to use as your background and indicator images. These files should be stored as a .gif or .jpg and you must know where they are located on the controller.

  The most common place to store these images is on FR:. If these are stored on MC:, then the memory card must be inserted to view the Visual Diagnostic screen properly.

  The screen area allocated for the background image is 630 pixels wide by 330 pixels high. For maximum performance, the background image should not be larger than this.

Steps

1 Press the [MENU] key.
2 Select SETUP.
3 Press the F1, [TYPE].
4 Select Visual Diagnostics. If this is the first time you are creating a Visual Diagnostic screen, the following screen will be displayed.

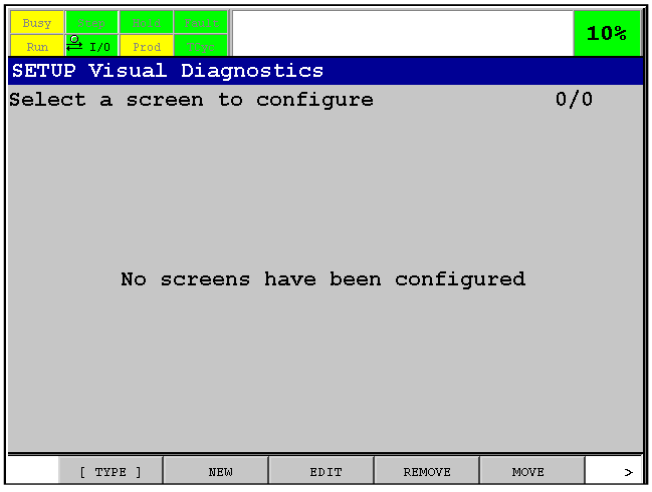


Fig. 51.1 (a) Select a Screen to Configure

If this is not the first time you have created a Visual Diagnostic screen, thumbnails of Visual Diagnostic screens currently configured on the controller are listed. To edit an existing screen, select the screen and press the F3, EDIT, go to Step 6.

5 To create new screen, press F2, NEW. Fig.51.1(b) will be displayed. Refer to Table 51.1(a) and Table 51.1(b) to set up the new screen.

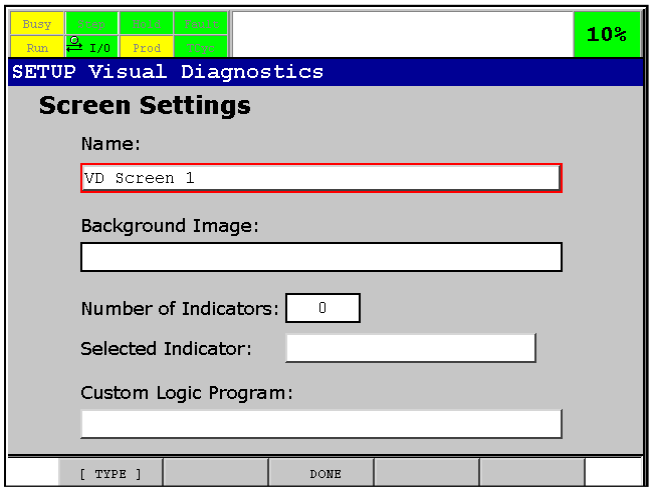


Fig. 51.1 (b) Screen Settings Page

The red box on the screen indicates the selected field. You can select a field using the arrow keys. If you have a touch screen-enabled *i*Pendant, you can select a field by touching it. The function keys are adjusted to reflect operations that can be performed on the selected field. F3 is always DONE which, along with PREV, returns you to the previous SETUP screen.

Table 51.1 (a) SETUP Visual Diagnostics Screen Settings Setup Items

Item	Description
Name	The item is the text that shows up on the SETUP and STATUS list screens so it must be unique. It can be from 1 to 40 characters and it can contain spaces. You can modify the name directly on the SETUP Visual Diagnostics Screen Settings screen.
Background Image	The large picture that covers most of the Visual Diagnostic screen is called the background image. It is stored as a file. When set, the file name is shown on the SETUP Visual Diagnostics Screen Setting screen.
Number of Indicators	This item specifies the number of indicators that have been set up. You cannot modify the number directly. Only the function keys are active when you are on this field.
Selected Indicator	Only one indicator can be configured at a time. The Selected Indicator field specifies which one is currently selected.

Item	Description
Custom Logic Program	You can identify a program that will run each time the Visual Diagnostic screen is refreshed. Use this feature if complex logic is required to determine the state of an indicator. A typical custom logic program will combine the status of various elements in the controller and set the indicators state control elements accordingly. A custom logic program can be implemented in either TPP or KAREL. It must not contain any motion statements. Use the following guidelines to create a custom logic program: • The program must exist before setting this field but you can modify it later to change the logic. The most recent version of the program is run each time the screen is update. • If it is a teach pendant program, the attribute should be set as follows: BUSY_LAMP_OFF = 3; ABORT_REQUEST = 3; PAUSE_REQUEST = 7; DEFAULT_GROUP = *, *, *, *, *, *, *, * By specifying 3 in the BUSY_LAMP_OFF setting, the custom logic program runs and the lamp which indicates the program running will not be lit. By specifying 3 in the ABORT_REQUEST setting and 7 in the PAUSE_REQUEST setting, when the program running as the same time as the custom logic program aborts or pauses, continue the execution of the custom logic program. The custom logic program does not contain any motion statements to run at the same time as the other program, specify * in the DEFAULT_GROUP setting. These settings can only be made by editing the TP program in the ASCII format (.LS). In order to load the ASCII program, ASCII Program Loader option (A05B-2600-R796) is required. • The program must run quickly and terminate. If the program takes more than 1 second to complete an warning alarm will be posted but Visual Diagnostics will proceed. One second is very long. A typical program should complete in less than a millisecond.

6 To change the VD screen name,
   a. Select the Name field, and press the [ENTER] key.
   b. Type the name of the screen, and press OK.
7 To select the background image,
   a. Select the Background Image field and press the F4, [CHOICE].
   See the screen shown in Fig. 51.1 (c) for an example.

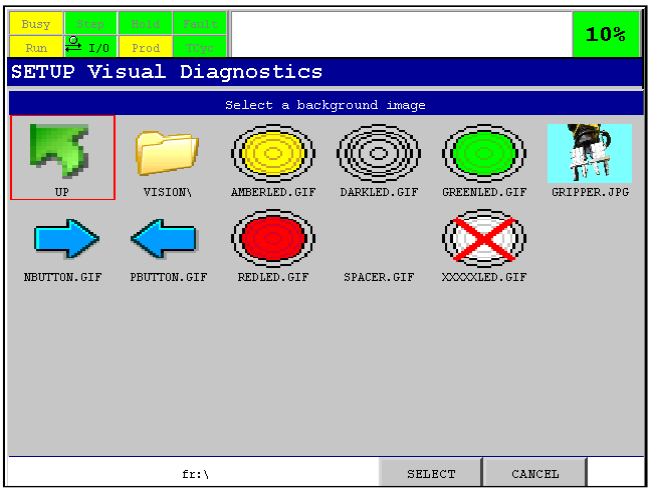


Fig. 51.1 (c) Selecting an Image File

b. The red box on the screen indicates the item with focus. Use the arrow keys to move the red box when more than one item is displayed. Press the [ENTER] key or the F4 SELECT to select the item with focus.
If you have a touch screen *i*Pendant, touching the item gives it focus and selects it in one step. Selecting the green arrow displays the images and folders in the parent folder. The top most parent is a list of devices. See Fig. 51.1 (d). Selecting a folder or a device displays the folders and image files within it. Selecting an image file returns that name to the field being edited in the Visual Diagnostics setup screen.

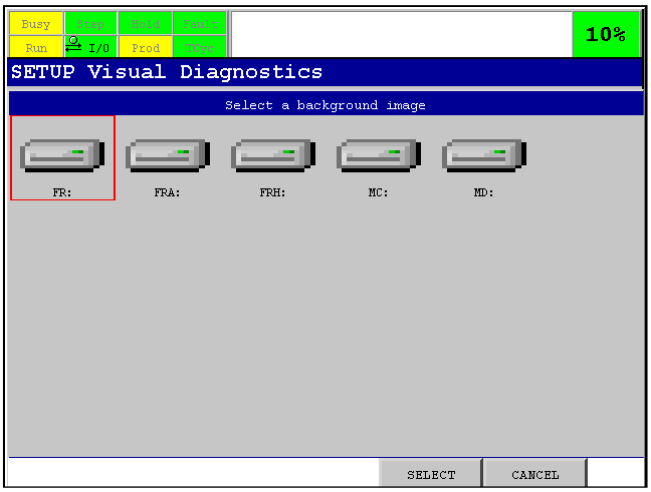


Fig. 51.1 (d) File Devices

8 To adjust background image,
a. Select the Background Image field and press the F2, ADJUST. The selected graphic will be displayed in the center of the screen as shown in Fig. 51.1 (e).

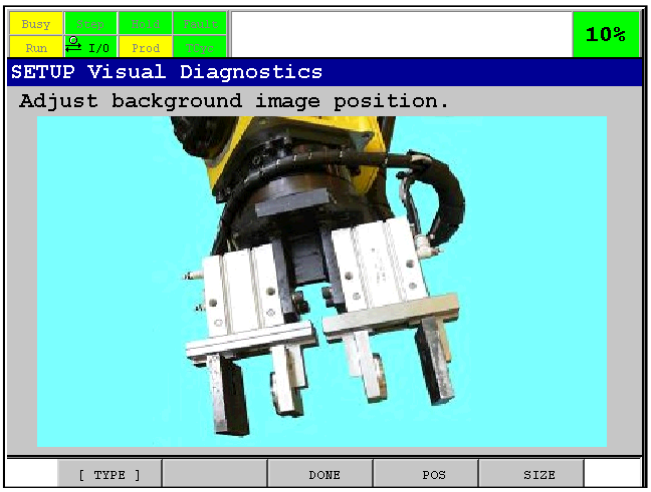


Fig. 51.1 (e) Adjust

b. F4, POS and F5, SIZE act like sticky radio buttons. Only one or the other is active at a time. When F4, POS is active, the arrow keys reposition the indicator image. When F5, SIZE is active, the up and down arrow keys grow and shrink the height of the image. The right and left arrow keys grow and shrink the width.

c. To position or resize the graphic in larger increments, press the [SHIFT] key and then use the arrow keys.

d. To complete the operation, press the F3, DONE or the [PREV] key.

9 To add in indicator, select the Number of Indicators field and press the F2, NEW. A new indicator with default settings will be created and you will immediately be placed into the edit mode as described below and shown in Fig. 51.1 (f).

10 To edit an existing indicator,

a. Select the Selected Indicator field and press the F2, EDIT.

b. Only one indicator can be edited at a time. If the name of the indicator you want to edit is not the one displayed, press the F4, [CHOICE] and select it from the displayed list.
Indicator editing is done on a screen similar to the one shown in Fig. 51.1 (f).
Refer to Table 51.1(b) for information on setting up items on the Indicator Setting screen.

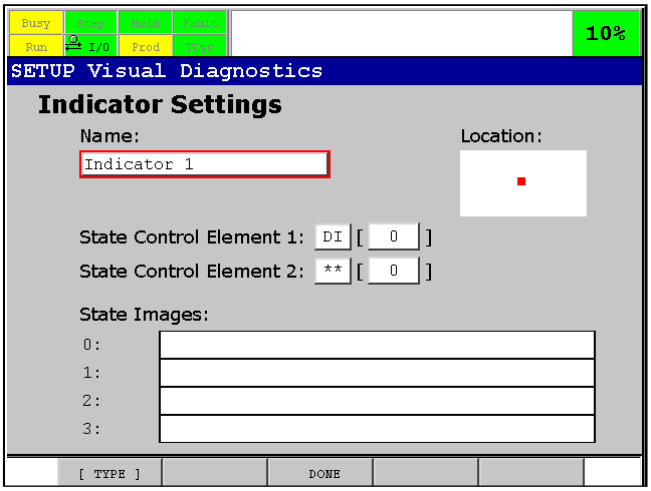


Fig. 51.1 (f) Indicator Settings Page

Table 51.1 (b) Setup Visual Diagnostics Indicator Setup Items

Item	Description
Name	This item specifies the name of the indicator. It provides you with a way to reference the indicator for setup. It is not displayed on the Visual Diagnostic STATUS screen.
Location	The Location field shows a graphic that provides a rough idea where the indicator is located.
State Control Elem 1	This item specifies the first element that controls the indicator. It can be an I/O signal, data register, or vision register.
State Control Elem 2	This item specifies an optional second element to combine with the first one to control the indicator.
State Images	These are the images to display when the State Control Element(s) evaluate to the corresponding number.

11 To name the indicator,
   a. Select the Name field, and press the [ENTER] key.
   b. Type in the new indicator name and press OK.
12 To adjust the location of the indicator on the graphic,
   a. Select the Location field and press the F2, ADJUST.
      The background image will be displayed with a small red box placed on it showing where the indicator is located. See the figure below.
   b. Use the arrow keys to move the small red square to the location you want. If you have a touch screen, you can touch on a new location. Press the F3, DONE or PREV to complete the operation.
13 To change the State Control Element type,
   a. Select the left field in State Control Element line, and press the F4 [CHOICE].
      Each State Control Element line has two fields. The type field is displayed as one or two characters to the left of the left bracket. The index field is between the brackets.
      Valid types for this field are:
      - Boolean I/O - DI, DO, F, RI, RO, SI, SO, UI, UO, WI, WO
      - Numeric I/O - AI, AO, GI, GO
      - Data Register - R
      - *i*RVision Register – VR
      - Not used - ** (Only valid for SCE2)
   b. Select a type, and press the [ENTER] key.
14 To change the State Control Element index,
   a. Select the right field and press the [ENTER] key.
   b. Enter the index number directly. No validation is done at this time. An invalid type or index is detected only when the Visual Diagnostic screen is viewed in the STATUS menu (refer to Section 51.3).
15 To associate an image to a state,
   a. Select the image name field to the right of the desired state number and press the F4, [CHOICE].
   b. Use the file thumbnail browser to select the image. See Fig. 51.1 (g) for an example. Fig. 51.1 (g) shows the associated images to states 0 and 1.

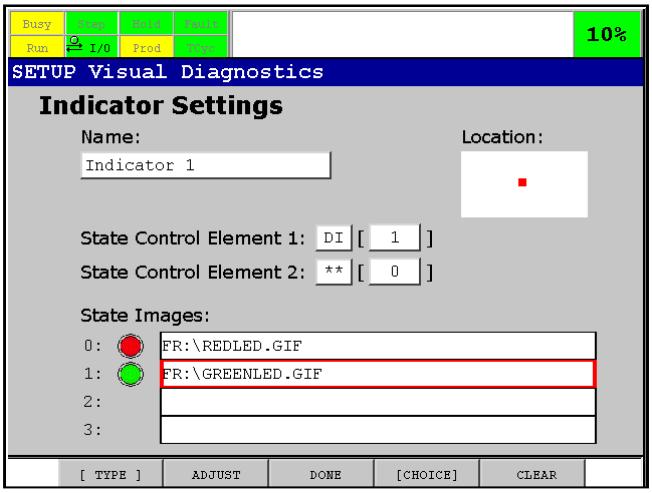


Fig. 51.1 (g) Indicator Settings

16 To make fine position and scale adjustments to the state image,
   a. Select the State Image field and Press the F2, ADJUST.
   b. The background image with only the currently selected indicator state image overlaid on top will be displayed. See Fig. 51.1 (h).

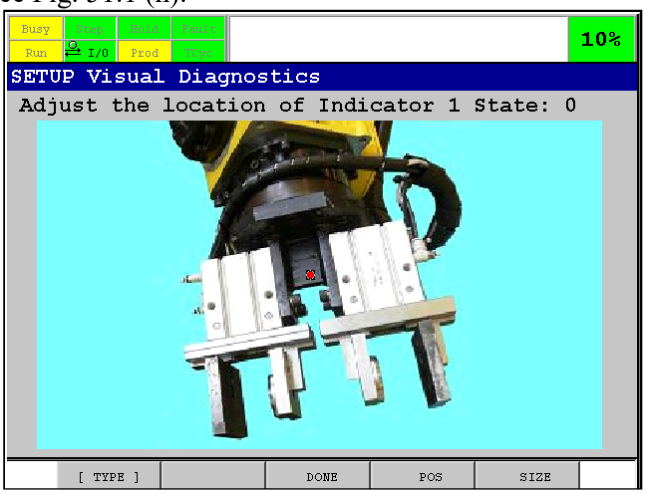


Fig. 51.1 (h) Adjust the Location of LED Indicator State: 0

   The position adjustment is relative to the grid location chosen for the indicator. These adjustments apply to the specific indicator state on the specific Visual Diagnostic screen being set up. They do not alter the indicator image file itself.
   c. F4, POS and F5, SIZE act like sticky radio buttons. Only one or the other is active at a time. When F4, POS is active, the arrow keys reposition the indicator image. When F5, SIZE is active, the up and down arrow keys grow and shrink the height of the indicator image. The right and left arrow keys grow and shrink the width. To position or resize the graphic in large increments, press the [SHIFT] key and then use the arrow keys.
   d. Press the F3, DONE or the [PREV] key to complete the operation.
17 When you are finished setting up the indicators on the screen, press the F3, DONE or the [PREV] key. This will return you to the Screen Setting page on which the following procedures are performed.
18 To remove the selected indicator from a screen,
   a. Select the Selected Indicator field and press F5, REMOVE.
   b. A confirmation box will be displayed so that you are aware that the indicator will be permanently removed by this function.
19 To assign a custom logic program,
   a. Select the Custom Logic Program field and press the F4, [CHOICE]. This will display the list of programs currently loaded on the controller.

b. Select the program you want from this list, and press the [ENTER] key. See the Fig.51.1 (i) for an example.

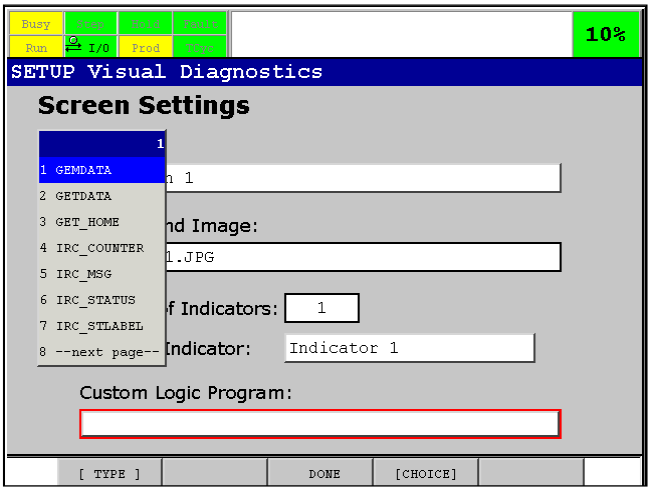


Fig. 51.1 (i) Custom Logic Program Field

20 To clear the Custom Logic Program assignment, select the Custom Logic Program field and press the F5, CLEAR.

51.2 MANAGING SCREENS

You can perform functions such as remove, move, copy, and load a new screen.

If the selected item is currently being displayed in any other *i*Pendant pane, the EDIT and REMOVE function keys will not let you operate on it.

Procedure 51-2 Managing Screens

Condition

- Visual Diagnostic screens have already been set up.

Step

1 Press the [MENU] key.
2 Select SETUP.
3 Press the F1, [TYPE].
4 Select Visual Diagnostics. The following screen will be displayed.

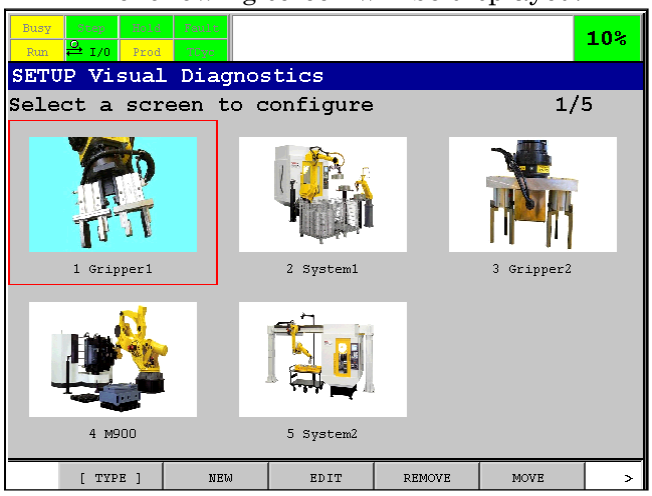


Fig.51.2(a) SETUP Visual Diagnostics

5 To remove a screen, press the F4, REMOVE. A confirmation box will be displayed so that you are aware that the screen will be permanently removed by this function.

6 To reposition a screen,

a. Press the F5, MOVE. The following screen will be displayed.

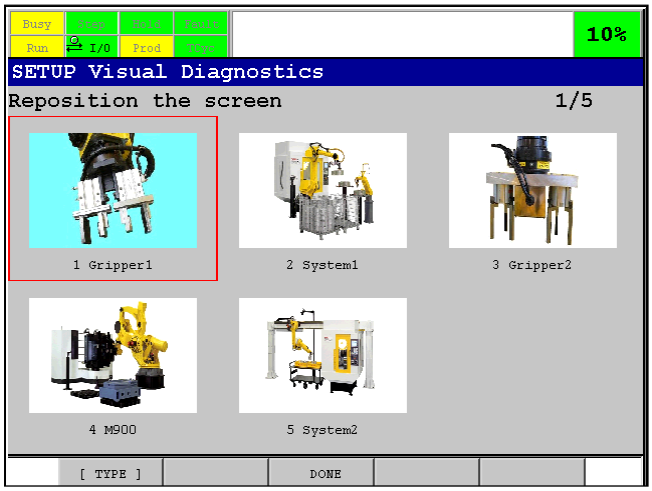


Fig.51.2(b) Reposition the Screen

b. Use the arrow keys to move the selected screen to its new position.

c. Press the F3, DONE to finish the operation and return to the screen selection page.

7 To make a new screen as an exact copy of the currently selected screen but with a different screen name,

a. Press >, NEXT, and then press the F2, COPY. A new screen will be created and the Screen Settings page shown so that you can edit the new screen.

b. Use Procedure 51-1 to edit the screen.

8 To load Visual Diagnostic screens developed on another controller onto this controller,

a. Press >, NEXT and then press the F3, LOAD.

b. Using the file browser, find and select the file named “vmscreens.xml” on any file device. When selected, the system merges the screens defined in it with the screens that currently reside on the controller. The file named “vmscreens.xml” is saved by the file backup operation in FILE screen. When the file backup operation is done in FILE screen, if “Application” or “All of above” is selected in the backup menu, the file named “vmscreens.xml” is saved to the designated device.

c. If a duplicate screen name is encountered, you are prompted with the option of replacing or keeping the one resident on the controller.

51.3 VIEWING VISUAL DIAGNOSTIC SCREENS

Use Procedure 51-3 to view Visual Diagnostic Screens.

Procedure 51-3 Viewing Visual Diagnostic Screens

Condition

- The Visual Diagnostic screen you want has already been set up.

Step

1 Press the [MENU] key.

2 Select STATUS.

3 Press the F1, [TYPE].

4 Select Visual Diagnostics. The following screen will be displayed.

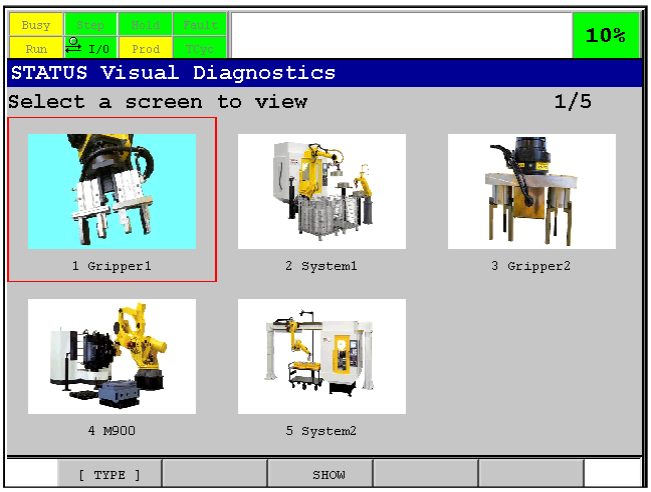


Fig. 51.3 (a) Select a Screen to View

Use this menu item to display a Visual Diagnostic screen. Thumbnails of the set of screens configured on the controller are shown on the list selection screen.
The red box on the screen indicates the selected screen. You can select a screen using the arrow keys. If you have a touch screen-enabled *i*Pendant, you can select a screen by touching it. The numbers in the upper right hand corner indicate how many screens are available and which is currently selected.

5 To display a screen, select it and press the F3, SHOW.
If the selected item is currently being edited in any other *i*Pendant pane, F3, SHOW will not let you view it.

6 If the screen you have selected has an invalid signal set, the following screen will be displayed.

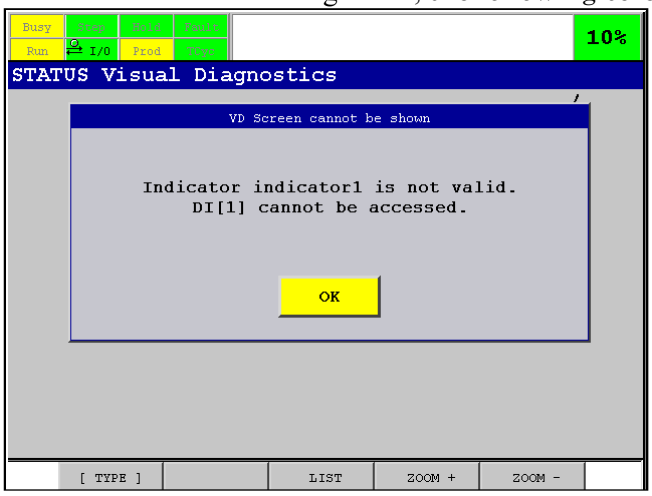


Fig.51.3(b) Invalid Indicator

a. Press OK.
b. Go to the SETUP menu to redefine the signal for that indicator. See Procedure 51-1.

7 After you select a Visual Diagnostic screen, the following screen will be displayed.

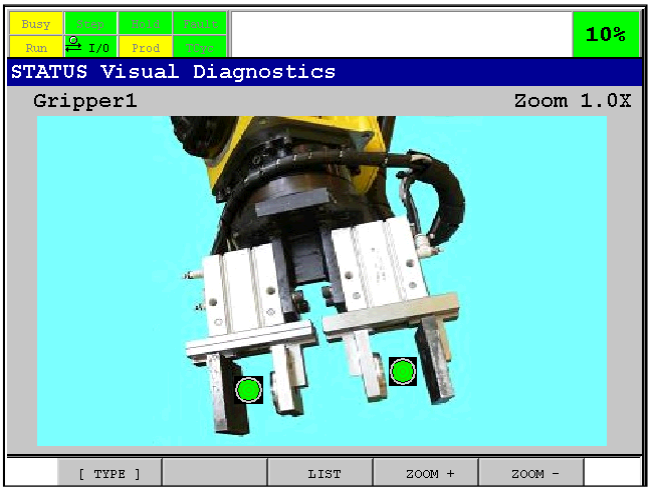


Fig.51.3(c) Visual Diagnostics Screen Display

The top line displays the title of the screen. In this example, the title is Gripper1.
Adjustments (since controller power up) to the magnification and picture center are remembered for each Visual Diagnostic screen. So, if you leave the screen and then return, the view is restored.

8 To increase magnification of the image,
   a. Press the F4, ZOOM+.
   b. The magnification is increased keeping the point of the image in the center of the display fixed in place.

9 To decrease magnification of the image,
   a. Press the F5, ZOOM-. The point of the picture in the center of the display is kept in place until one of the edges of the image touches the edge of the display area. The image will then be shifted in that direction to keep the image edge touching the display edge.
   b. Magnification will not be reduced below 100%.
   c. F2, FULL, restore the image to full view.

10 To adjust how the image is centered in the display when the magnification is greater than 100%,
   a. Press the right and left arrow keys to shift the view to the right and left.
   b. Press the up and down arrow keys to shift the view up and down.
   c. To move the view in larger increments, hold [SHIFT] key while pressing the arrow key.

11 To display the Visual Diagnostic Screen list selection page, press the F3, LIST. This key has the same function as the [PREV] key.

52 PDF VIEWER FUNCTION

The *i*Pendant PDF Viewer function (J838) allows viewing of PDF documents in proprietary file formats. The following file formats are supported:

Adobe® Acrobat® Portable Document Format (PDF) v1.3 and later.

52.1 OPENING A PDF DOCUMENT

PDF documents can be opened on any local storage device into robot controller.

Procedure 52-1 Open PDF Document

1 Press the [MENU] key.
2 Select FILE.
3 Press the F5, [UTIL]
4 Select Set Device.
5 Select a device contains pdf file.
6 Press F2, [DIR]
7 Select *.PDF.
8 Select the file to view.
9 Press the [ENTER] key.

NOTE
It is required to change the display type of the file screen to “Wide” before selecting file name when pdf file size is larger than 10 Mbyte. Press the [NEXT] key, press the F5[VIEW] and select “Wide” to change to “Wide” screen.

NOTE
It may take a lot of time to display or scroll when pdf file size is very large.

52.2 NAVIGATING WITH THE PDF VIEWER

Fig. 52.2 shows a typical single pane view of a document open in the PDF Viewer.

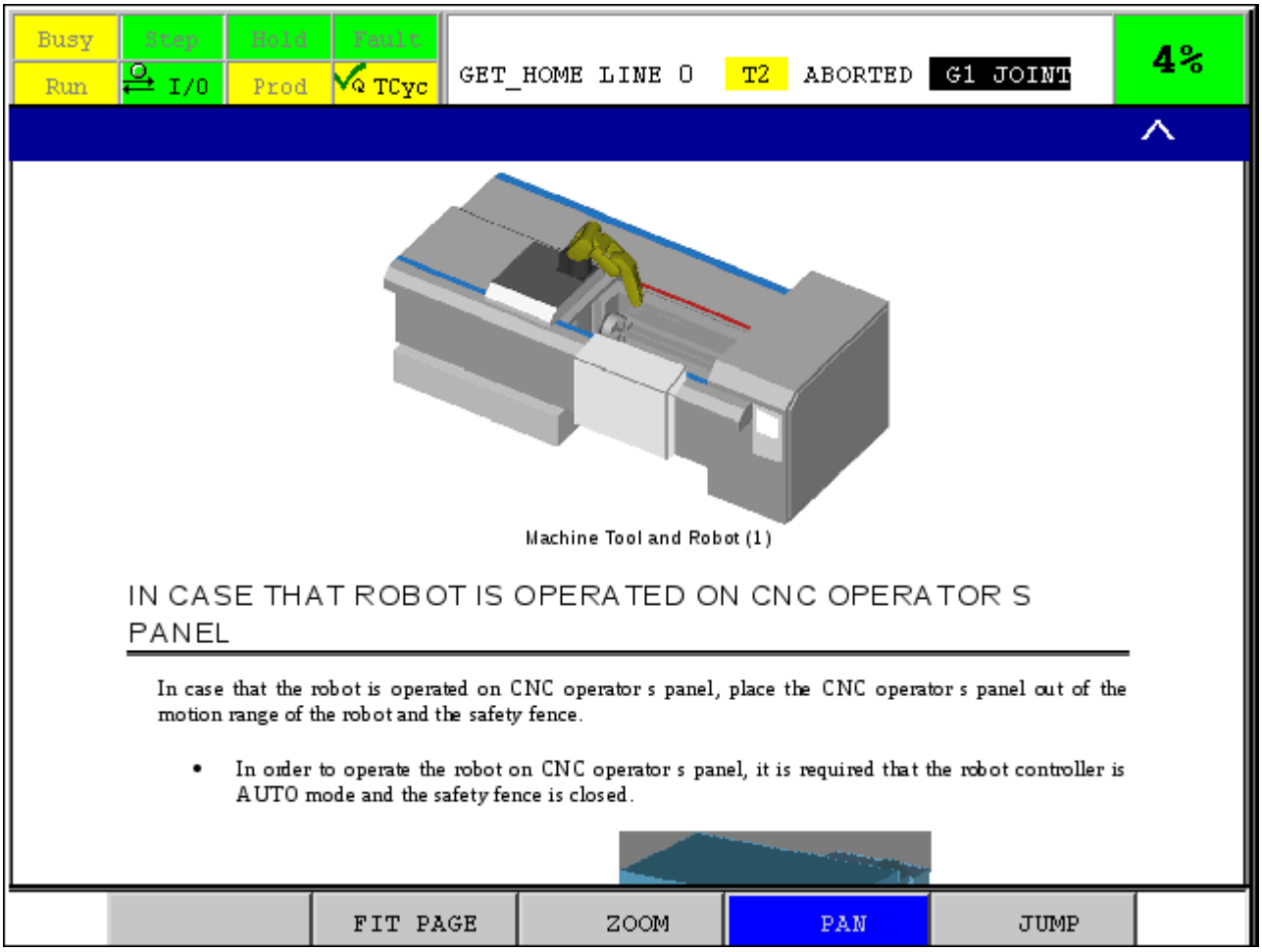


Fig. 52.2 PDF Viewer

Three function keys are available when the PDF Viewer has focus.

Table 52.2 PDF Viewer Function Keys

Function Key	Description
FIT PAGE	Zoom so that the entire page is visible within the PDF Viewer window. this mode allows quick navigation between pages of a document.
ZOOM	The up and down arrow key will increase and decrease the zoom level respectively.
PAN	The arrow keys will pan the display. Touch screen gestures will always pan the display.
JUMP	Table of contents will be displayed and can jump to each page if PDF file has the table of contents.

53 HELP AND DIAGNOSTICS DISPLAY

You can use the DIAG/HELP key to display help for the data displayed in the current window on *i*Pendant or diagnostics for an alarm.

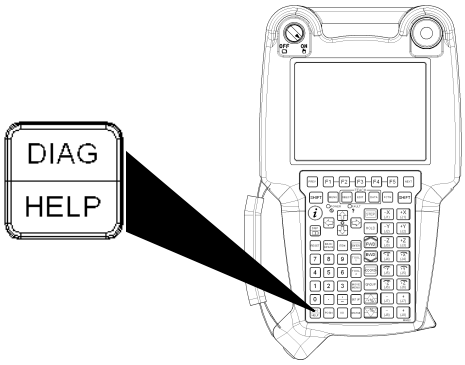


Fig. 53 (a) DIAG/HELP key

- If the [DIAG/HELP] key is pressed, help for the current screen will be displayed.
- If the [DIAG/HELP] key is pressed while holding down the [SHIFT] key, diagnostics for an active alarm or the specified alarm is displayed.

To use these functions, the following software options are required.

- Help Display: Online Help (A05B-2700-R664)
- Diagnostics Display: Alarm Cause/Remedy (A05B-2700-R665)

53.1 ONLINE HELP FUNCTION

If the [DIAG/HELP] key on the teach pendant is pressed, the help to explain the operation in the current screen is displayed. For example, if the [DIAG/HELP] key is pressed while the program shift screen is displayed, the help for the program shift screen is displayed as follows.

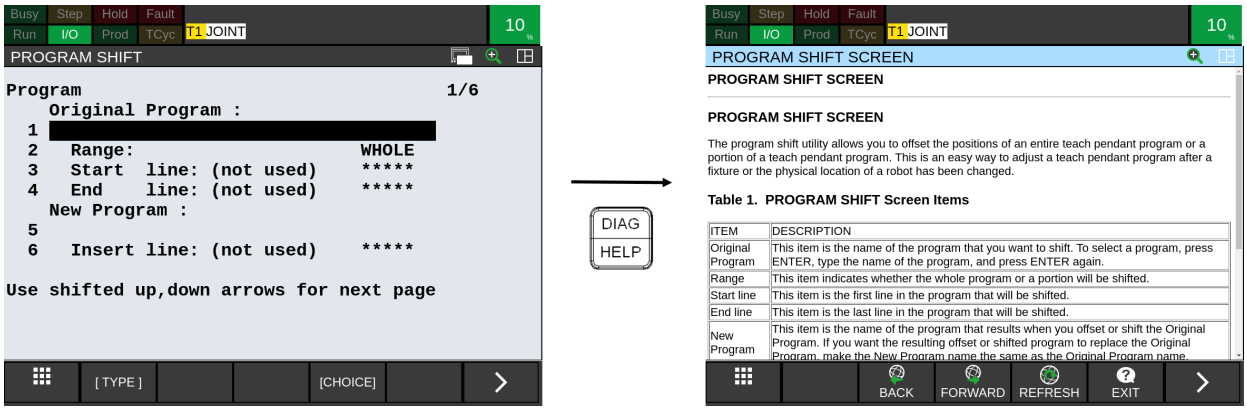


Fig. 53.1 (a) Help for Program Shift Screen

To return to the previous screen, press the [PREV] key or the [DIAG/HELP] key when the help screen is displayed. You can also display the previous screen by pressing the [DIAG/HELP] key with the [SHIFT] key.

53.2 ALARM CAUSE/REMEDY DISPLAY FUNCTION

If the [DIAG/HELP] key on the teach pendant is pressed with the [SHIFT] key, the alarm cause and remedy can be displayed as the diagnostic information. According to the displayed screen and alarm occurring status, the displayed contents of the alarm cause and remedy differ as follows.

1 In case that the alarm screen (the active alarm screen or the alarm log screen) is displayed, the cause and remedy for the specified alarm by the cursor in the alarm screen is displayed.
2 In case that the active alarm screen is displayed, if no alarm is displayed in the screen, the cause and remedy for the selected alarm in the alarm list is displayed.
3 In case that the screen other than the alarm screen is displayed, when an alarm occurs, the cause and remedy for the active alarm is displayed.
4 In case that the screen other than the alarm screen is displayed, when no alarm occurs, the cause and remedy for the selected alarm in the alarm list is displayed.

To return to the previous screen, press the [PREV] key or the [DIAG/HELP] key when the cause and remedy is displayed. You can also display the previous screen by pressing the [DIAG/HELP] key with the [SHIFT] key.

Procedure 53-1 Diagnostics display when alarm screen is displayed.

Conditions

- The alarm screen (the active alarm screen or the alarm log screen) is displayed on teach pendant.
- The alarm is displayed in the screen.

Steps

1 Move the cursor to the alarm that you want to display the cause and remedy.
In case that only one alarm is displayed in the active alarm screen, the cursor cannot be moved. The cause and remedy for the displayed alarm in the active alarm screen is displayed.
2 Press the [DIAG/HELP] key while holding down the [SHIFT] key.
3 The cause and remedy for the specified alarm is displayed.

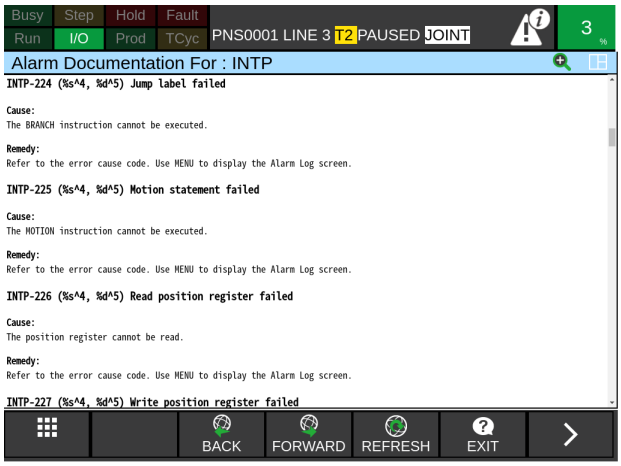


Fig. 53.2 (a) Alarm Cause and Remedy Display

Procedure 53-2 Diagnostics display when the active alarm screen is displayed and no alarm is displayed in the screen.

Conditions

- The active alarm screen is displayed on teach pendant.

- No alarm is displayed in the screen.

Steps

1 Press the [DIAG/HELP] key while holding down [SHIFT] key.
2 The list screen for alarm is displayed as follows.

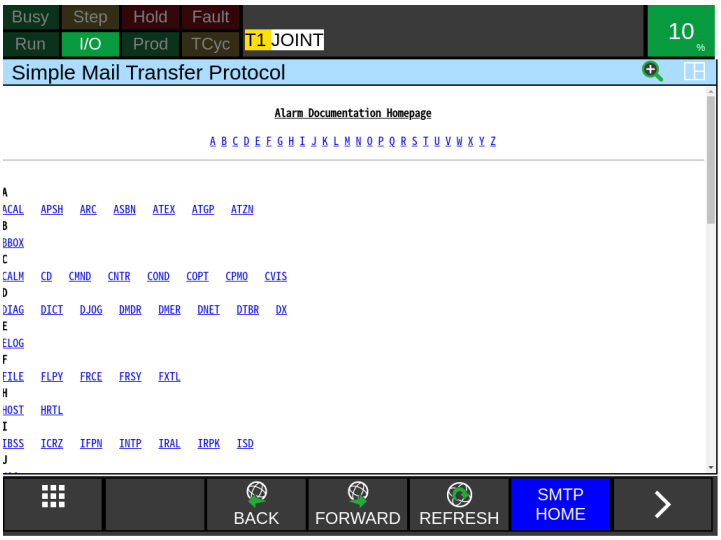


Fig. 53.2 (b) List screen for DIAG alarm

3 You can display the list of the alarm that you want to display the cause and remedy by selecting facility name in this screen.
4 Select the alarm to display the cause and remedy in the alarm list screen.

Procedure 53-3 Diagnostics display when the screen other than alarm screen is displayed and an alarm occurs.

Conditions

- The screen other than the alarm screen is displayed on teach pendant.
- An alarm occurs.

Steps

1 Press the [DIAG/HELP] key while holding down [SHIFT] key.
2 The following popup screen will be displayed.



Fig.53.2 (c) Popup screen (When an alarm occurs)

3 In case that the “CONTINUE” button is pressed, the cause and remedy for the active alarm is displayed as follows. In case that the “CANCEL” button is pressed, the popup screen disappears and the previous screen will be displayed.

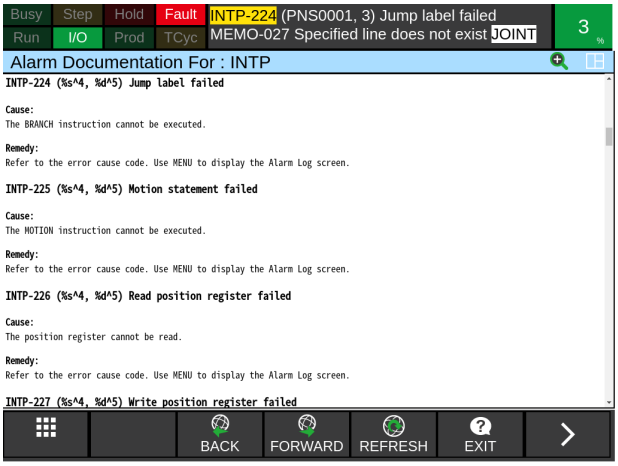


Fig.53.2 (d) Alarm Cause and Remedy Display

Procedure 53-4 Diagnostics display when the screen other than alarm screen is displayed and no alarm occurs.

Conditions

- The screen other than the alarm screen is displayed on teach pendant.
- An alarm does not occur.

Steps

1 Press the [DIAG/HELP] key while holding down [SHIFT] key.
2 The following popup screen will be displayed.

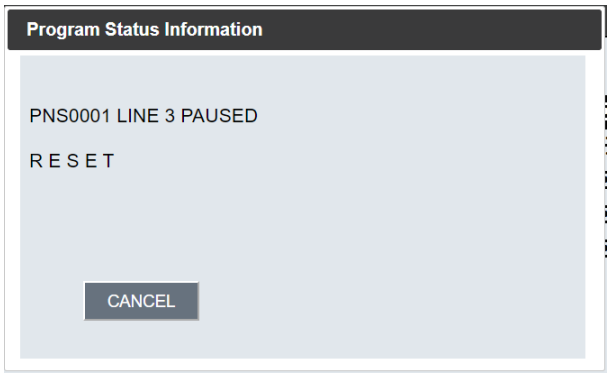


Fig.53.2 (e) Popup screen (When no alarm occurs)

53.3 HELP/DIAGNOSTICS SCREEN

The operations in the online help screen and the alarm cause and remedy screen are as follows.

Change of the display by the link

By the selecting the link, you can display the figure, the table, or the other page specified by the link.

- To highlight the link that you want to select, use the up and down cursor keys.
- If the teach pendant equips the touch panel, touch the link to select.

Scroll of the screen

If the screen exceeds the window, the scroll bar is displayed. Please scroll the screen by the following ways.

- To scroll the screen, press the up or down cursor key while holding down [SHIFT] key.
- If the teach pendant equips the touch panel, touch the scroll bar.

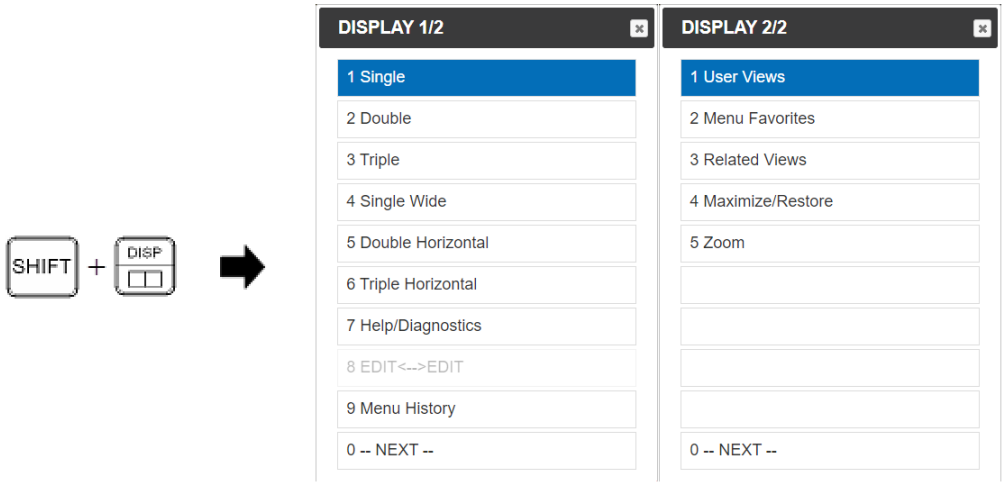
If the window on the teach pendant is divided, the Help/Diagnostics screen will be displayed as follows.

Table 53.3 (a) Help/Diagnostics Display

Window mode	Display Location
Single	The Help/Diagnostics information replaces the current information in the window.
Status/Single	The Help/Diagnostics information replaces the current information in window on the right.
Double	The Help/Diagnostics information replaces the current information in the inactive window (window without focus).
Triple	The Help/Diagnostics information replaces the current information in the window without focus. For example in triple mode, the left window is window 1, the upper right window is window 2, and the lower right window is window 3. Then if the active window is window 2, the Help/Diagnostics information replaces the contents of window 3. If the active window is window 3, the Help/Diagnostics information replaces the contents of window 1, and so forth.

53.4 HELP/DIAGNOSTICS MENU

You can display an online help screen and an alarm cause and remedy screen by selecting an item in the Help/Diagnostics menu. The Help/Diagnostics menu will be displayed when the Help/Diagnostics item is selected in the DISPLAY menu that is displayed by pressing the [DISP] key with the [SHIFT] key.



(The items in the menu and the order of the items may differ depending on the software and version.)

Fig. 53.4 (a) DISAPLY menu by pressing the [DISP] key with the [SHIFT] key

When the Help/Display item is selected in the DISPLAY menu, the Help/Diagnostics menu will be displayed.

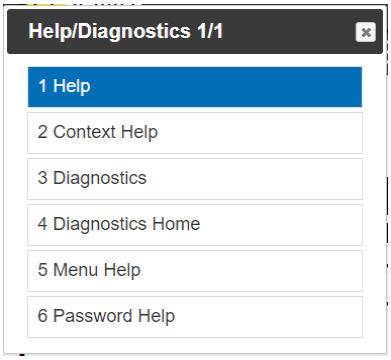


Fig. 53.4 (b) Help/Diagnostics menu

Table 53.4 describes the displayed screen by selecting each item is the Help/Diagnostics menu.

Table 53.4 (a) Help/Diagnostics menu items

Item	Description
Help	This item displays help for the currently active window.
Context Help	This item displays context sensitive help, if exists, for the currently active window. In other cases, the same screen as Help is displayed.
Diagnostics	When the alarm screen is displayed, this item displays the cause and remedy for the designated alarm in the alarm screen. When the active alarm screen is displayed, if there is no active alarm, the DIAG alarm list screen to display Diagnostics Home Page is displayed. When the screen other than the alarm screen is displayed, the alarm cause and remedy screen is not displayed by this item.
Diagnostics Home	This item displays the DIAG alarm list screen to display Diagnostics Home Page.
Menu Help	This item displays the menu tree and provides links to all the help files other than the current menu.

54 MAINTENANCE REMINDER

The **Maintenance Reminder** function reminds you when it is time to perform periodic maintenances or part replacement in the robot system. When predefined and customized maintenance time is approaching, you will be notified by TP warnings, output to a DO, and a Notification Iconto help you maintain the robot system at appropriate intervals.
You can configure the maintenance items. Some robot models have predefined items setup for recommended maintenance tasks, called Required Maintenances.
To use this function, the Maintenance Reminder option (A05B-2700-J766) is required. When installed with the Maintenance Reminder/ Intelligent Cable Change Reminder Add-on (A05B-2700-S511), the maintenance items will be changed to Intelligent Cable Change Reminder.
In addition, when used with Zero Down Time (ZDT), you can check maintenance items on mobile devices other than the teach pendant.

54.1 INTELLIGENT CHANGE REMINDER

For supported models, Maintenance Reminder changes to Intelligent Change Reminder.

54.1.1 Intelligent Grease Change Reminder

The Intelligent Grease Change Reminder (referred to as i-grease change from now on) estimates and notifies you of the appropriate grease replacement timing by considering operating conditions, instead of by just using the conventional elapsed time and running time data. Under severe conditions, replacing grease earlier will prevent premature failure of the reducers. Under moderate conditions, i-grease can prolong the grease replacement interval.
i-grease change displays the estimated remaining life as a percentage (%). It also displays the year and month in which the remaining life is expected to reach 0%, considering the recent usage trend. Use these values to plan grease replacement.
i-grease change requires the ambient temperature to be set. Refer to 54.3.3 to set the ambient temperature.
The i-grease change function does not require any additional options. Contact your local FANUC representative for supported robot models and software versions.

54.1.2 Intelligent Cable Change Reminder (option: S511)

Intelligent Cable Change Reminder (referred to as i-cable change from now on) estimates and notifies you of the appropriate mechanical unit cable replacement timing by considering operating conditions, instead of by just using the conventional elapsed time and running time data.
i-cable change displays the estimated remaining life as a percentage (%). It also displays the year and month in which the remaining life is expected to reach 0%, considering the recent usage trend. Use these values to plan a cable replacement.
The i-cable change function requires the Intelligent Cable Change Reminder Add-on (A05B-2700-S511).
Contact your local FANUC representative for supported robot models and software versions.

54.2 MAIN MENU

Use the following procedure to display the Maintenance Reminder screen.
1 Press the [MENU] key.
2 Select “STATUS”.
3 Press the F1, [TYPE].

4 Select “Reminder”.

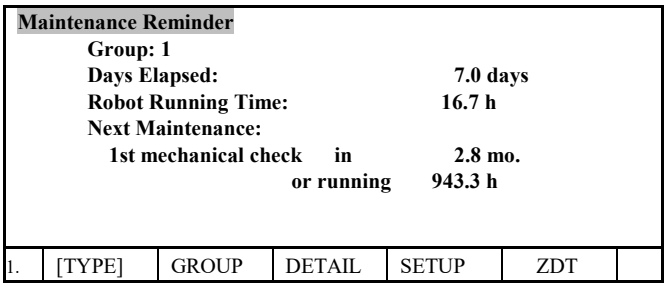


Table 54.2 Maintenance Reminder screen items

Item	Description
Group	This item is the group number of the currently displayed maintenance items.
Days Elapsed	This item displays the days that have elapsed since the robot started being used. It also includes the time when the power was off. Adjusting the clock of the controller may result in more elapsed time than actually occurred. All of the groups in the controller share the same elapsed time.
Robot Running Time	This item displays the total time the robot has been moving. It does not include waiting time. Each group has an individual counter.
Next Maintenance	Among the enabled maintenance items, the item with the shortest remaining time to the next maintenance will be displayed with the remaining time. The items displayed are limited to those setup with “Days Elapsed” or “Robot Running Time”.

If multiple groups exist, to specify the motion group number, press F2 [GROUP].
To display the screen for checking maintenance time and completing maintenance, press F3[DETAIL] (Section 54.4).
To display the setup screen, press F4[SETUP] (Section 54.3).
To display the ZDT setup screen, press F5[ZDT] (Section54.5).
To display the Monitored Out screen, press [NEXT], > key and then F3 “OUT LIST”. To display the Maintenance Record screen, press [NEXT], > and then F4 “RECORD”.

54.3 MAINTENANCE REMINDER SETUP

This section describes how to set up the Maintenance Reminder function:

- Common settings – settings that are common to groups and items, such as when to send reminder notices.
- Maintenance item settings –settings that are apecific to each maintenance item, such as maintenance intervals.
- Passcode settings – if a passcode, described in the next section, is set, displaying the setup menu will prompt you for a passcode. Set a passcode in cases where maintenance personnel and operators are not the same.
- Ambient temperature settings

54.3.1 Common Settings

The Maintenance Reminder SETUP screen is for settings common among all controller motion groups, such as when to output a maintenance reminder warning.

Use the following procedure to display the Maintenance reminder SETUP screen.

1 Display the main Maintenance Reminder screen in Section 54.2.
2 Press the F4, [SETUP］.

3 If a passcode has been set, the prompt line will ask “Please enter the passcode: “. Enter the passcode to display the SETUP screen.

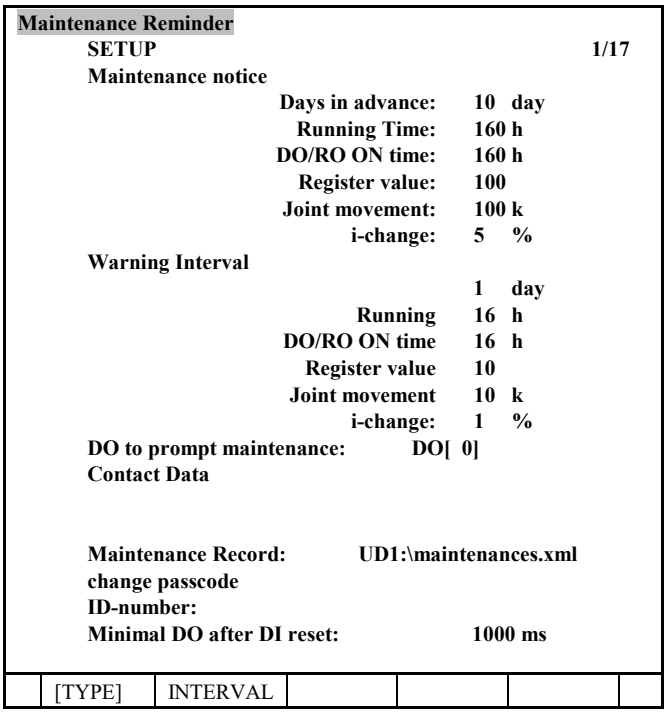


Table 54.3.1 Maintenance Reminder SETUP Common setting items

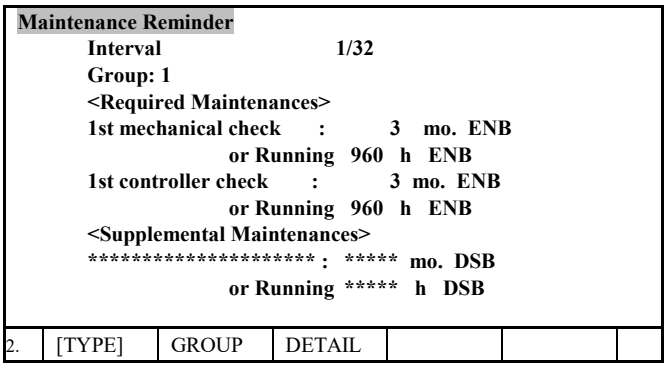
Item	Description
Maintenance notice	Set the timing in which to post warnings and output DO. In addition to Days, Running Time, DO/RO ON time, Register value, and Joint movement, if the groups include an i-grease supported robot model, or if i-cable change (S511) has been installed, i-change will be added. This is for i-grease change and i-cable change. When the remaining time or value to the next maintenance reaches the set time, warnings, DO, and Notification screen begin to remind you that maintenances will soon be necessay. The units for Joint movement are thousand revolutions (kRev) for rotational axes, and km for linear axes. If item-specific setup, described in a later section, is performed, the item-specific value will be used instead of this common setting value.
Warning Interval	Set the interval to post warnings after the maintenance notice time, explained above, has been reached. If item-specific setup, described in a later section, is performed, the value will be used instead of this common setting value.
DO to prompt maintenance	Set the DO to be output after the maintenance notice time, explained above, has been reached. DO will be ON when any item reaches the maintenance notice time, and will stay ON until “COMPLETE” operation, to be explained in a later section. Setting DO[0] will disable DO output. An item-specific DO can be set up, which will be described in a later section.
Contact Data	Any text up to 37 letters can be set for these two lines. When set, they will be displayed on the main menu in Section 54.2. These lines will not be displayed when no data is set. Any data, such as contact data for maintenance personnel can be set.
Maintenance Record	The Maintenance Record can be output to a file. This item sets the destination file for the record. The Maintenance Record will be output only after the file is created on the Maintenance Record screen, which will be described in a later section.
change passcode	Select this item and press ENTER to set a passcode. (If a passcode has already been set,) After entering the current passcode, enter the new code twice to set the new passcode. Set the passcode to “1111” to disable passcode protection. Do not forget the passcode, or settings cannot be changed and “COMPLETE” operation, to be explained in a later section, cannot be executed.

Item	Description
ID-number	This item allows you to set any text of up to 10 letters as the controller ID. When set, it will be displayed on the Maintenance Reminder main menu shown in Section 54.2. It will not be displayed when no ID has been set. The ID-number can be used for any purpose, such as for maintenance management.
Minimal DO after DI reset	Maintenance items can be reset by DI, which will be described in a later section. This item sets the minimal time length of a DO output when a DI reset is performed. When checking the DO using an external PLC or other device, set the length to be longer than the cycle length of the checking. The actual time length of the DO output may take longer depending on the processing status of the controller.

54.3.2 Maintenance Item Settings

Use the following procedure to display the Maintenance Reminder Interval screen.

1 Display Maintenance Reminder SETUP screen in Section 54.3.1.
2 Press the F2 [INTERVAL］.



There are two types of maintenance items: Required Maintenances and Supplemental Maintenances. Required Maintenances are preset for each robot models, and you can add more maintenance items to Supplemental Maintenances. You set up each type of maintenance item differently, but they operate in the same way after setup.

Table 54.3.2(a) Maintenance Item settings

Item	Description
Group	This item is the group number of the currently displayed maintenance items.
Name	This item is the name of the maintenance item. The names of the Required Maintenances are preset and cannot be changed. The names of Supplemental Maintenances can be set to any string up to 36 letters.
Interval	This item is the interval for performing the maintenance. The intervals can be set for Days Elapsed and/or Robot Running Time, DO/RO ON Time, Register value, or Joint movement. The units forDays Elapsed are days (day) or months (mo.) which can be changed by pressing F4 or F5 keys to select the units. The units for Robot Running Time and DO/RO ON Time are hours (h). The units for Joint movement are thousand revolutions (kRev) for rotational axes, and km for linear axes. The intervals for Required Maintenances have already been set up. They can be shortened, but cannot be made longer. The intervals for i-grease change and i-cable change will not be displayed and cannot be changed.
ENB/DSB	This item enables or disables the interval type of the maintenance item. Select the item and press F4 or F5 key to change whether the maintenance is enabled. Required Maintenances are enabled and cannot be disabled. However, the maintenances that are meant to be done once, such as 1st mechanical check, will be disabled by “COMPLETE” operation, to be explained in a later section.

If multiple motion groups exist, to change the motion group number to set up, press F2 [GROUP].
To set up the type of interval and notification details, press F3 DETAIL to display the Maintenance Item Setup screen.

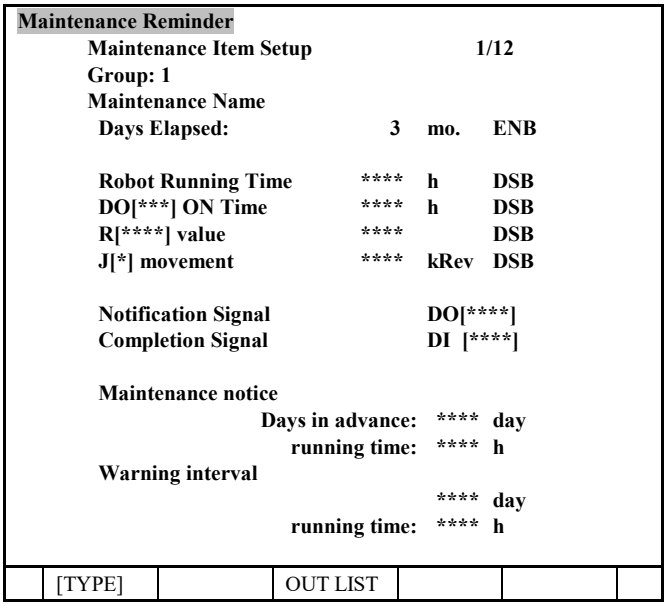
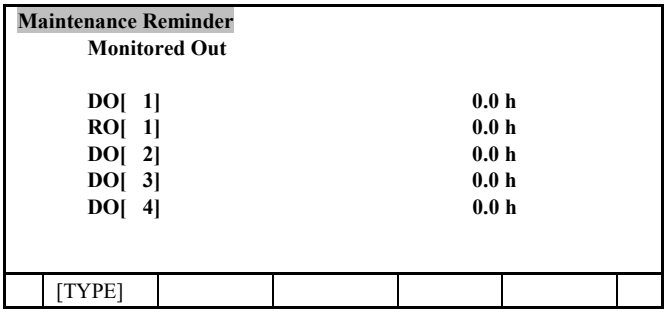


Table 54.3.2(b) Maintenance Item Setup screen items

Item	Description
Group	This item is the group number of the currently displayed maintenance item.
Name	This item is the name of the maintenance item. The names of the Required Maintenances are preset and cannot be changed. The names of Supplemental Maintenances can be set to any string up to 36 letters.
Interval	This item is the interval for performing maintenance. The intervals can be set for Days Elapsed and/or Robot Running Time, DO/RO ON Time, Register value, or Joint movement. The intervals for i-grease change and i-cable change will not be displayed and cannot be changed.
DO/RO ON Time	This item count the time that the specified DO or RO is ON, and notifies you of the maintenance period based on the time length. To change the type of output, move the cursor to DO (RO) and press F5 RO (F4 DO). Enter the number of the output for which to monitor the ON time inside [] , and length of the interval to the right. Up to five different outputs can be monitored.
Register value	You can notify the maintenance period based on the increase in value of the register you specify in this item. Enter the number of the register inside [], and the length of the interval, increase of the register value, to the right. At the completion process, described later, the register value itself will not be reset. The next interval starts from the value at the time, and the notification will be based on the difference between the current value and that value.
Joint movement	You can notify the maintenance period based on the amount of motion of the specified axis in this item. Enter the number of the axis inside [], and length of the interval, amount of motion, to the right. The units of interval are thousand revolutions (kRev) for rotational axes, and km for linear axes.
ENB/DSB	This item enables or disables the interval type of the maintenance item. Select the item and press F4 or F5 key to change whether the maintenance is enabled. Required Maintenances are enabled and cannot be disabled. However, the maintenances that are meant to be done once, such as 1st mechanical check, will be disabled by a “COMPLETE” operation, to be explained in a later section. Robot Running Time, DO/RO ON Time, Register value, and Joint movement can only be enabled when the other three items are disabled.
Notification Signal	This item sets an item-specific DO to output when the item reaches the notification time. To disable the item-specific DO, set this item to 0.
Completion Signal	Completion of Maintenance, described in a later section, can be processed through the DI input you specify here. To process completion with a DI, the Notification Signal item described above needs to be set. The Completion Signal needs to be input until the falling edge of the Notification Signal is detected. If the Completion Signal is input when the Notification Signal is ON, the Notification Signal will be OFF after the process. If the Completion Signal is input when the Notification Signal is OFF, the Notification Signal will be ON after the process, then OFF after the time set in “Minimal DO after DI reset”, described in Section 54.3.1.

Item	Description
Maintenance Notice	This item sets the timing to post warnings and output DO. The timing for Days Elapsed and for the enabled type of interval can be set. When the remaining time or value to the next maintenance period reaches the set time, warnings, DO, and Notification screen will start to remind maintenance. If not set, the common settings described in Section 54.3.1 will be used.
Warning interval	This item set the interval to post warnings after the Maintenance Notice time explained above has been reached. If not set, the common settings described in Section 54.3.1 will be used.

To display the list of DO/ROs for which ON time is counted, press F3 "OUT LIST". This screen is the same screen as the one displayed when "OUT LIST" is pressed on the main Maintenance Reminder screen.



54.3.3 Ambient Temperature Setting

i-grease change requires ambient temperature setting. Set the ambient temperature in SYSTEM/CONFIG screen.

1 Press the [MENU] key.
2 Select "SYSTEM".
3 Press the F1, [TYPE].
4 Select "CONFIG".
5 Select "Motor/Amplifier".
6 Select the number of the group to set.
7 Enter the ambient temperature in Celsius.

Enter the ambient temperature at operation.

NOTE
The ambient temperature setting is shared with Motion Profiler.

54.4 CHECK MAINTENANCE TIME AND COMPLETE MAINTENANCE

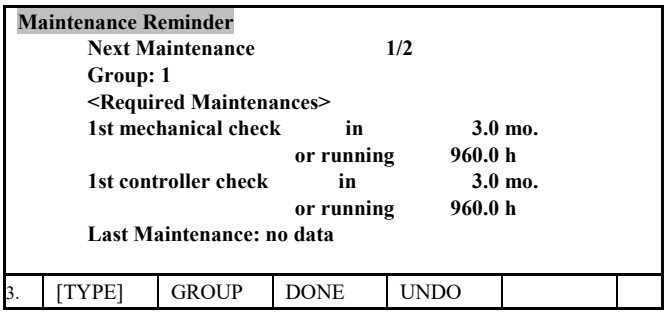
On the DETAIL screen, the time remaining to the next maintenance period is displayed for each maintenance items. Also on this screen, an item can be "COMPLETE", after the completion of the corresponding maintenance, to reset the counter to start counting for the next maintenance period.

54.4.1 Checking the Maintenance Time

Use the following procedure to display the Maintenance Reminder DETAIL screen.

1 Display the Maintenance Reminder main screen in the Section 54.2.
This menu only displays the item in the group with the shortest remaining days in Days Elapsed and the item with the shortest remaining hours in Robot Running Time, if different from the item with the shortest remaining days.

2 Press the F3 [DETAIL]



On the DETAIL screen, all the enabled items in the group are displayed with the remaining time, beginning with Required Maintenances followed by Supplemental Maintenances. The upper lines represent the remaining time in Days Elapsed and the lower lines represent the remaining time in Robot Running Time, DO/RO ON Time, Register value, or Joint movement. If only one type of the counter is enabled, the other type will display *'s.

Supported robot models also display i-grease changes. For i-grease changes, in addition to remaining life as a %, the estimated month the grease reaches its life will be displayed. This estimation is based on recent usage. The estimation month will not be calculated until it has been used for some period. The grease life for i-grease change will be estimated once a day, with the usage information from the day before. It will not be updated other than on a once-a-day estimation.

For supported models with i-cable change (S511) installed, i-cable change that shows the life of a mechanical cable is displayed. As with i-grease change, it will display the remaining life as a % and the estimated month the cable reaches the end of its life. The cable life is calculated while in motion and will be updated when the screen is displayed.

If multiple motion groups exist, press F2 [GROUP] to change the displayed group.

54.4.2 Maintenance Reminder Notification

When the maintenance time approaches, and the remaining time or value is less than the Maintenance notice item configured in Section 54.3.1 or 54.3.2, the following warning will be posted:

"DIAG-009 G(group number) (Name of the maintenance) in (type of interval)(remaining time/value) (unit)".

This warning is posted repeatedly at the Warning Interval set in Section 54.3.1 or 54.3.2 until the maintenance is processed "COMPLETE" in the following section.

Also, if an output DO is set, the DO will turn ON. This DO will turn back ON even if it is turned OFF in the I/O menu, unless the maintenance item is marked "COMPLETE".

The Notification screen will notify you of the need for maintenance in addition to the warnings and DO.

54.4.3 Completing Maintenance

When the Required Maintenances reach their maintenance time, follow the instructions in the mechanical unit operator's manual to complete the corresponding maintenance, or contact your local FANUC representative. For Supplemental Maintenances, complete the corresponding maintenanceaccording to the design intent. Upon completion of the maintenance, you must mark the maintenance item as COMPLETE. Otherwise, the maintenance reminder warning will continue to be posted at the set intervals, and the next maintenance period cannot be appropriately reminded.

Perform the following steps to mark a maintenance item as COMPLETE.

1 Display the Maintenance Reminder DETAIL screen in Section 54.4.1.
2 Select the item for which maintenance has been completed.
3 Press F3 [COMPLETE].
4 If passcodes are being used, enter the passcode, when asked "Please enter the passcode:".
5 The prompt line will ask, "done?" Press F4 [yes].
If the wrong item is selected, press F5 [no].
The selected item is marked as COMPLETE and the count to the next maintenance period starts.
For an item that requires maintenance only once, such as "1st mechanical check," the count will not restart, but displays "done".

If you have inadvertently marked a maintenance task COMPLETE without having it completed, you can undo it only for one time. Use the following procedure to cancel the COMPLETE.

1 Display the Maintenance Reminder DETAIL screen in Section 54.4.1.
2 Select the itemmarked COMPLETE by mistake.
3 Press the F4 [UNDO].
4 If passcodes are being used, enter the passcode, when asked "Please enter the passcode:".
5 The prompt line will ask, "undo the last maintenance?" Press F4 [yes].
If the wrong item is selected, press F5 [no].
The remaining time changes back to when it was not marked COMPLETE.

After marking the item COMPLETE, selecting the item on the DETAIL screen displays the date of COMPLETE at the bottom of the screen after Last Maintenance:. If the item is never COMPLETE or if COMPLETE is canceled with an UNDO, it displays "no data". If an item displays "no data", it cannot be undone.

Maintenance completion can also be processed using the Completion Signal DI input. When the DI set in Fig. 54.3.2 is input, the item will be marked COMPLETE. The DI must be input until the completion process is confirmed. If the Notification Signal DO is ON, it will be OFF when the process is finished. If it is OFF, it will be ON when the process is finished, then OFF after "Minimal DO after DI reset" time set in Section 54.3.1. In either case, the process can be confirmed by detecting the falling edge of the Notification Signal DO.
Fig. 54.4.3 shows how you should set up your system to utilize completion processing using the Completion Signal DI input. If maintenance is done after the notification, with the DO ON, the DI should be input until the DO turns OFF. If maintenance is done before the notification, with the DO OFF, the DI should be input until the DO turns ON, then OFF. The DO would be ON at least for the time set in "Minimal DO after DI reset". This time should be long enough so that your system can detect the DO ON. Since the signal for stopping the DI input is always the DO change from ON to OFF, the system for the completion process can be set up as inputting the DI until the falling edge of the corresponding DO is detected, regardless of whether the DO was ON during maintenance.
If Completion Signal DI is input by mistake, it needs to be undone on the teach pendant using the UNDO process described above.

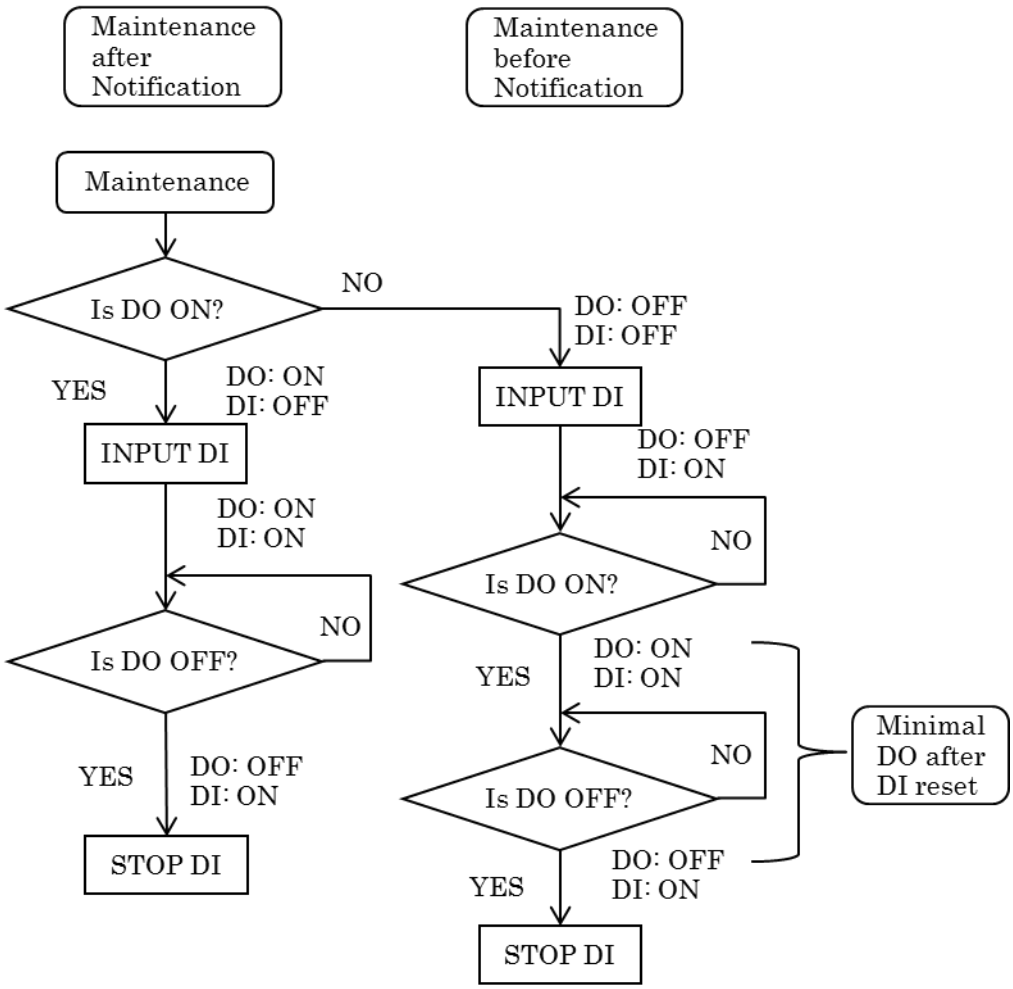


Fig. 54.4.3 A user system that utilizes the completion process by DI input

54.5 ZDT

Zero Down Time (ZDT) collects and analyzes data to prevent unexpected robot downtime, via a network. When used with ZDT, Maintenance Reminder enables checking maintenance timing remotely from devices other than the teach pendant, such as PCs or smart devices. It sends periodic data and reminder data to the configured server. The data can be checked from a web portal or ZDT application. Refer to the ZDT manual for information on setting up ZDT.

54.5.1 Maintenance Reminder setup

To check Maintenance Reminder information on ZDT, in addition to ZDT server and other setup, some setup is required on the Maintenance Reminder side.
Use the following procedure to display the ZDT settings menu.

1 Display the main menu in Section 54.2.
2 Press the F5 [ZDT]
If ZDT option is not installed, [ZDT] will not appear.

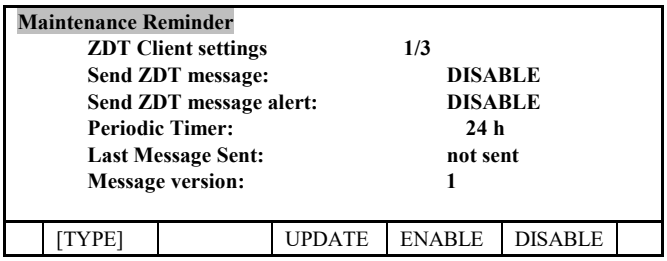


Table 54.5.1 Maintenance Reminder setup items for ZDT

Item	Description
Send ZDT message	Enables and disables the sending of Maintenance Reminder messages to ZDT. If ZDT is disabled, this item cannot be set to ENABLE.
Periodic Timer	This item sets the interval at which ZDT messages are to be sent when Send ZDT message is set to ENABLE.
Last Message Sent	This item displays the last time Maintenance Reminder created a ZDT message. If it is different from the time stamp on the Maintenance Reminder screen for the ZDT application, it indicates that the message was not sent or received properly.
Message version	This item is the version of the message sent to the ZDT server. If your are using Zero Down Time package version 1.0, set this item to 0.

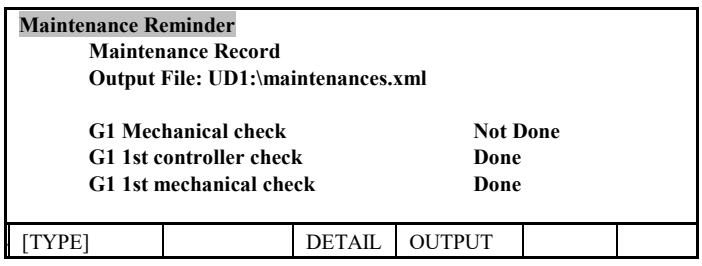
This setup is valid for all groups and maintenance items.
If for some reason, such as a server problem, message is not sent or received properly and information on ZDT is not updated (when the time stamp for ZDT and the Last Message Sent item mentioned above do not correspond), a new Message can be created and sent at any time by pressing F3 [UPDATE].

54.6 MAINTENANCE RECORDS

Maintenance records are saved. Maintenance records are created at the time of the first warning, and are updated when maintenance is overdue or maintenance is completed. You can view up to the 10 most recent maintenance records on the teach pendant. If you create a maintenance record file in advance, old maintenance records can be saved and viewed as an XML file.

54.6.1 Displaying Maintenance Record

Use the following procedure to display the Maintenance Record screen.
1 Display the main menu in Section 54.2.
2 Press the [NEXT] key.
3 Press the F4 RECORD



To display the detailed information for the selected item, press F3 DETAIL.
To create the specified file and write the information for the items that are COMPLETE, press F4 OUTPUT. Once the file is created, the information on the COMPLETE maintenance itemswill be added to the file as the maintenance items are completed.

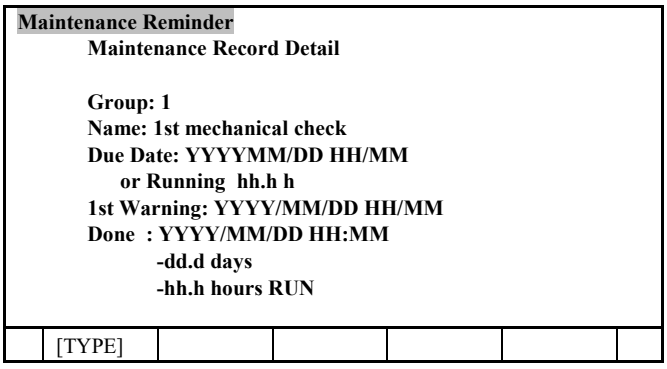


Table 54.6.1 Maintenance Record items

Item	Description
Group	This item is the group number of the currently displayed maintenance item.
Name	This item is the name of the displayed maintenance item.
Due Date	This item is the due date of the maintenance item. The first line shows the due date calculated from the interval set in “Days Elapsed”. If not set, *’s will be displayed. The second line shows remaining time/value of the other type of interval. A negative value indicates that the time/value is overdue. If not set, the second line will not be displayed.
1st Warning	This item displays the date the first warning for the item was posted. If no warnings have been posted before being marked COMPLETE, *’s will be displayed.
Done	This item displays the date the item is marked COMPLETE. The second and third lines show the amount of time/value overdue at completion. A negative value shows the remaining time/value when the item was marked COMPLETE before it was due.

54.6.2 Maintenance Record File

The Maintenance Record file is created in XML format in the specified directory with the specified name.

Table 54.6.2 Items in the Maintenance Record File

Node/Element	Description
MAINT_RECORD	Maintenance Record Data
CONT	The Controller
ID	The ID set in Section 54.3.1. If not set, it is another type of ID that is automatically set.
ROBOT	The group of the Maintenance Record Data
Model	The name of the model of the group such as R-2000/165F-27C
Group	The group number
ITEM	The maintenance Item
ID	The ID number of the maintenance item. This corresponds to the order the item is shown on the screen described in Section 54.3.2.
Name	The name of the maintenance item
Type	The type of maintenance item expressed in numbers. 1: “Days Elapsed” only. 2: “Robot Running Time” is set. 3: “DO/RO ON Time” is set. 4: “Register value” is set. 5: “Axis motion” is set. With supported models, 7: “i-grease change” is set. Supported models with i-cable change (S511) installed, will have 8: “i-cable change” set.
HIST	The record of maintenance
DueAct	The due date and time in terms of “Days Elapsed”
DueOther	Due for other type of timer. A negative value is the time/value till due. A positive value is the time/value overdue. If the item is COMPLETE, it is same as DonePast.

Node/Element	Description
DueType	The type of timer that came due, expressed as a number. 0:Not due. 1: “Days Elapsed”. 2: “Robot Running Time”. 3: “DO/RO ON Time”. 4: “Register value”. 5: “Axis motion”. With supported models, 7: “i-grease change”. With i-cable change (S511) installed, 8: “i-cable change”.
Warn	The date and time of the first warning.
DoneDate	The date and time when the item is COMPLETE.
DonePast	The difference between due and the timer when it is COMPLETE. This is the same as DueOther.

Example:

```
<?xml version="1.0"?>
<MAINT_RECORD>
<CONT ID="TEST ">
 <ROBOT Group="1" Model="R-2000/165F-27C">
  <ITEM ID="1" Name=" 1st mechanical check" Type="2">
   <HIST DueAct="2016/3/16 16:08" DueOther="-3839.8 hours RUN" DueType="0" Warn=""
DoneDate="2015/03/18 15:44:38" DonePast="-3839.8 hours RUN"/>
  </ITEM>
  <ITEM ID="11" Name="TEST1" Type="5">
   <HIST DueAct="" DueOther=" -100.0 kRev J[1]" DueType="0" Warn="2015/03/18 15:43:50"
DoneDate="2015/03/18 15:44:06" DonePast=" -100.0 kRev J[1]"/>
  </ITEM>
 </ROBOT>
</CONT>
</MAINT_RECORD>
```

54.7 ROBOT SETTINGS

When software is updated, the required maintenance items may have been changed. In that cases, robot setup at Controlled Start will have new required maintenance items set. The maintenance items are usually set without being overwritten, but if the sum of the numbers of the required maintenances and the supplemental maintenances exceeds the maximum number, the maintenance items will be overwritten.

54.7.1 i-grease Change

i-grease change is supported differently depending on the software version and robot models.
If the software is updated from a version that does not support i-grease to one that does, and robot setup is performed, it will switch from the conventional grease change to the i-grease change. (If robot setup is not executed after a backup is restored, the conventional grease change will remain.)
In this case, the ratio of the remaining hours until the maintenance in the conventional grease change will be used as the initial value of the remaining life of the i-grease change after robot setup. (After a grease change, when the Completion process is done, the remaining life will be reset to 100%.) Therefore, if the robot is switched to i-grease in the middle of the process, the maintenance reminder will be at a different time than if i-grease were used from the beginning, even if the usage conditions are the same.

54.7.2 i-cable Change

i-cable change is supported differently depending on the software version and robot models.
When i-cable change (S511) is added, or for multi-arm configuration, when the software is updated from a version that does not support one of the robot models to one that does, the conventional cable change will be changed to the i-cable change.
In this case, as with i-grease change, the ratio of remaining time will be converted to the initial value of the life of the i-cable change. Therefore, similarly, changing to i-cable in the middle of the process will result in different timing of the maintenance due compared to i-cable change were used from the beginning, even if the usage conditions are the same.

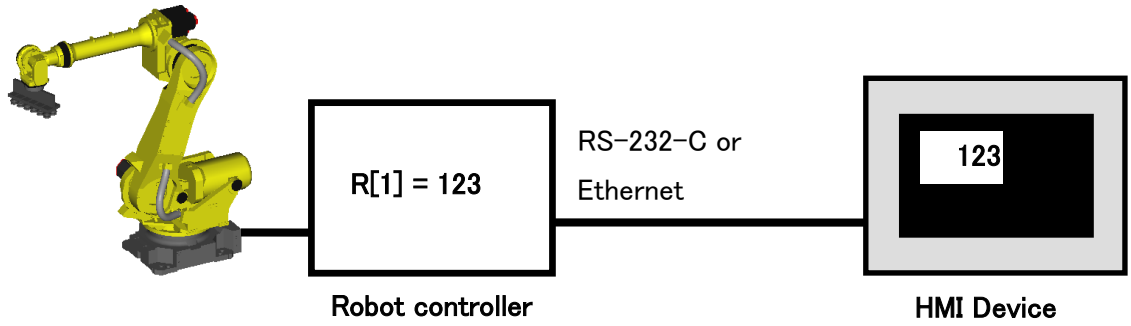
55 HMI DEVICE COMMUNICATION

The HMI Device Communication function allows the robot to communicate with external HMI devices etc. connected via RS-232-C or Ethernet using Modbus/TCP or OPC UA. This function allows various robot data, such as DI, DO, Register and so on, to be read or written from and to an HMI device etc.

This function performs the following two types of communication:

- Operates as a Modbus slave, communicating with an HMI device etc. that is the Modbus master
- Operates as an OPC UA Server, communicating with OPC UA Clients. (Robot can't become OPC UA client.)

The HMI Device Communication function allows you to design the screen by drawing lamps, switches, and so forth on the screen. You can assign robot data to the lamps, switches, and so forth as desired. For example, by assigning a robot register to the display panel on the HMI device, you can display the register value on the screen of the HMI device.



The HMI Device Communication function is standard when the Standard setting (R651) is selected, but the following version conditions must be met to use each function. When the North America Setting (R650) is selected, the HMI device (R553) option is required to use this function.

The response time for this function is often around 3-18ms but may take 100 ms or more, which is not guaranteed.
The response time varies depending on the robot controller load.
To obtain the current position in real time, use the HIGH SPEED POSITION OUTPUT in Chapter 59 in this manual.

The OPC UA server functionality is supported for Publisher and MonitoredItem, so subscribers can use the client side.
The minimum data acquiring interval for MonitoredItem is 100 ms.
In the OPC UA server function, the OPC UA client will read/write large arrays of data, so it is recommended that you use client software that supports index range, which enables to read/write only a specific range of data.

⚠ WARNING
This function checks the received data and performs the proper process only when the robot controller receives the data sent from the HMI device etc. This function cannot detect a situation in which the communication from an HMI device etc. is stopped accidentally. For this reason, do not use this function to transfer data that may affect safety. For example, even if HMI device etc. sends a STOP request to the robot, there is no guarantee that the robot will be stopped by the request. Therefore, it is extremely dangerous to use this function to create a safety-related system.

NOTE
When using this function to communicate with an HMI device etc., test the entire robot system including the HMI device etc.

Portions copyright © by OPC Foundation, Inc. and licensed under the Reciprocal Community License (RCL).

Connection tested HMI devices

Schneider Electric Japan Holdings Ltd. GP4000 Series

Set the following to the “Manufacturer” and “Series”.
RS-232-C : "Manufacturer: Modbus-IDA", "Series: MODBUS RTU SIO master"
Ethernet: "Manufacturer: Modbus-IDA", " Series: MODBUS TCP master"

KEYENCE CORPORATION VT3 Series

Set the following to the “Manufacturer” and “Model”.
RS-232-C: "Manufacturer: MODBUS protocol", "Model: RTU mode (1:N)"
Ethernet: "Manufacturer: MODBUS protocol", "Model: MODBUS/TCP(Ethernet)"

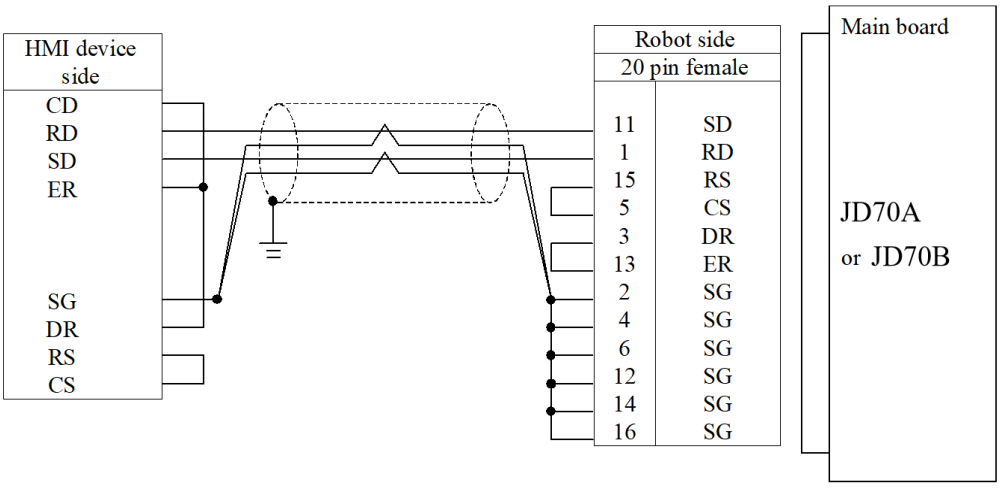
55.1 CONNECTION AN HMI DEVICE

55.1.1 Modbus RS-232-C Connection

Cable

The following diagram shows the connection to an RS-232-C port. Refer to the HMI device manual for information about the connection on the HMI device side.

The cable on the ROBOT CONTROLLER side is connected to the JD70A (Port 1) or JD70B (Port 2) connector on the main board.



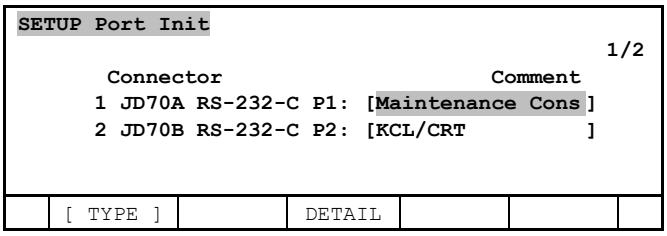
The specifications of the connectors and cable are as follows:

•Connector	PCR-E20FS (Honda Tsushin Kogyo)
•Housing	PCR-V20LA (Honda Tsushin Kogyo), or compatible connector
•Shielding quality	Entirely shielded twisted pair cable is recommended. For protection against noise, twisted pair wire should be used for pairs of RD and SG, SD and SG.

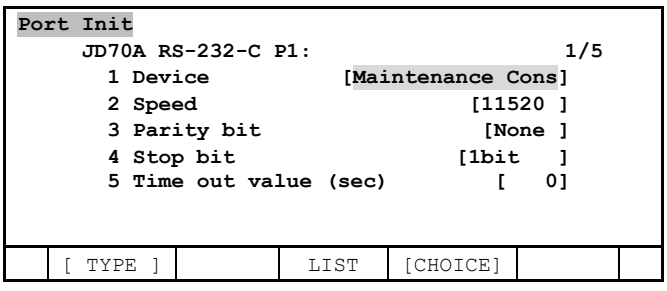
Setting of Serial Port

To use HMI device communication via RS-232-C, it is necessary to specify HMI dev(MODBUS) on the Port Init menu on the robot controller.

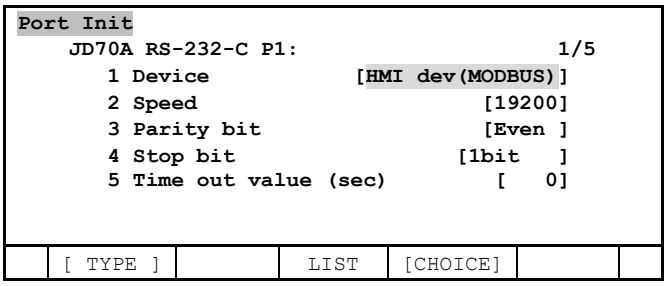
1 Press MENU, and select SETUP.
2 Press F1, [TYPE] and select Port Init.



3 Move the cursor to JD70A RS-232-C P2:, and press F3, DETAIL. The detail screen for JD70A RS-232-C P2: is displayed.



4 Move the cursor to Device, press F4, [CHOICE], and select HMI dev(MODBUS) from the list. HMI device and HMI dev(MODBUS) are on the list. You must set HMI dev(MODBUS) for Modbus communication. You cannot use HMI device and HMI dev(MODBUS) at the same time.



5 Setup Speed, Parity bit, and Stop bit in the same way as with the HMI device.

Set up the slave address

Set up the slave address for this function is set in [SETUP > Network Settings > HMI Device settings > Modbus connection > Slave address]. The default value is 1. The slave address of the Modbus RTU can be changed with the settings screen item. This function responds only to requests for the specified slave address. A request for slave address 0 is processed as a broadcast.

55.1.2 Modbus/TCP Slave Ethernet Connection

Setting up TCP/IP

To use HMI Device Communication via Modbus/TCP Slave Ethernet, you must set up TCP/IP on the robot controller in advance.
Details on the Ethernet interface and TCP/IP configuration can be found in the "SETTING UP TCP/IP" in the *Network and Security Function OPERATOR'S MANUAL* (B-84604EN).

Setting up the Modbus/TCP Slave Ethernet connection

The Modbus/TCP Slave Ethernet connection is disabled by default. To use Modbus/TCP Slave Ethernet connections, set the number of HMI and other devices that are simultaneously connected in [SETUP > Network Settings > HMI Device settings > Modbus connection > Number of connections] (Default is 0). If a Modbus/TCP Slave Ethernet connection is requested from HMI devices etc. that exceeds the set value, communication with the HMI device etc. with the oldest communication will be disconnected automatically.

Unit ID

The robot controller does not use the Unit ID in Modbus TCP communication data. The robot controller responds to all requests from an HMI device etc. regardless of the Unit ID.

Port number

The port number of this function is set in [SETUP > Network Settings > HMI Device settings > Modbus connection > Port number]. The default value is 502. You can change the port number for this function on the Network Settings screen.

Modbus TCP Server (R800)

The Modbus TCP Server (R800) option is independent of this function. However, when this option is loaded, if [SETUP > Network Settings > HMI Device settings > Modbus connection > Number of connections] is set to 1 or greater, the errors "PRIO-090 SNPX communication error", "HRTL-048 Address already in use" will occur and Modbus/TCP Slave Ethernet communication for this will not work. This is because this function and the Modbus TCP Server use the same port number (502).
By changing the port number of this function, both this function and the Modbus TCP Server can be used at the same time.

55.1.3 OPC UA Server Ethernet Connection

OPC UA is a multi-purpose and flexible industrial communication protocol that enables data exchange between multi-vendor products across different operating systems. This function enables a FANUC robot to connect with OPC UA clients and enables various robot data (such as DI, DO, registers, and so on) to be read and written. This function works as an OPC UA server and supports communication with external OPC UA clients over Ethernet.

The information model (data model) used for sending and receiving data is based on the Modbus data model of the HMI DEVICE COMMUNICATION function.

Limitations

The following limitations apply when using the FANUC robot OPC UA Server:

- Up to 10 OPC UA clients can communicate simultaneously.
- This function can be used only with the CD38A and CD38B RJ45 Ethernet port connections on the robot controller. This function cannot be used with the CD38C RJ45 Ethernet port connection.
- Some OPC UA client can set subscription and monitored item. By setting this, the client can get the item value when updated. For robot controller, up to 2 subscriptions can be registered per connection. The Client/Server method is also supported.
- For robot controller, up to 50 items can be monitored per subscription but the minimum data acquiring interval (Minimum notification interval to client) for the monitored item is increased by 100 ms per item: 100ms for the first item, 200ms for the second item, 300ms for the third item. This does NOT mean the 4th item would be updated only every 1 second (= 100ms + 200ms + 300ms + 400ms). 4th item can be updated every 400ms as minimum.
  Normal read/write (Data access(DA)) does not have such limitations.

TCP/IP Settings

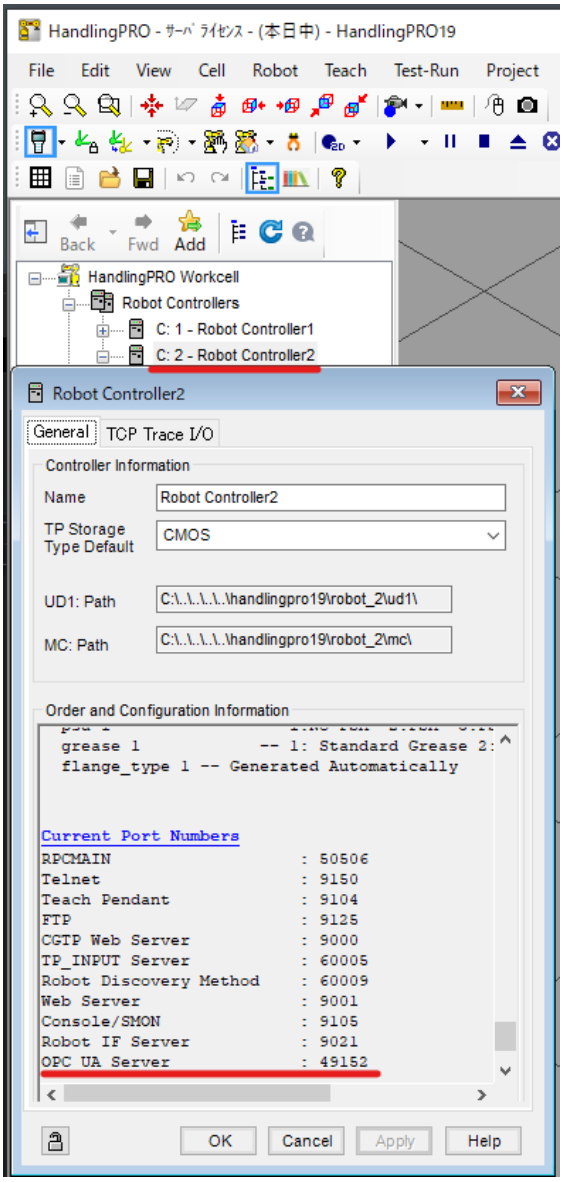
To use HMI Device Communication via the FANUC robot OPC-UA Server, you must set up TCP/IP on the robot controller in advance. See “SETTING UP TCP/IP" in the *Network and Security Function OPERATOR’S MANUAL* (B-84604EN).

The FANUC robot OPC-UA Server can be accessed from an external OPC-UA client by entering the robot’s OPC UA server URL in the following format:
opc.tcp://robot IP address:4880/FANUC/NanoUaServer

Example:
opc.tcp://172.22.194.222:4880/FANUC/NanoUaServer

The OPC UA client can connect with the robot OPC UA server after setting the above URL in the OPC UA client. Refer to Section 55.2 “MODBUS DATA MODEL” for details about assigning and accessing robot data. Authentication, user name and password aren’t supported.

For ROBOGUIDE, set the following format URL to OPC UA client.
opc.tcp://IP address for PC where ROBOGUIDE is running:4880/FANUC/NanoUaServer
4880 in the URL should be changed to the number displayed at “OPC UA Server” at the bottom of the window, which is displayed after double clicking the connecting robot controller, in the following figure.



55.2 MODBUS DATA MODEL

55.2.1 Modbus data model

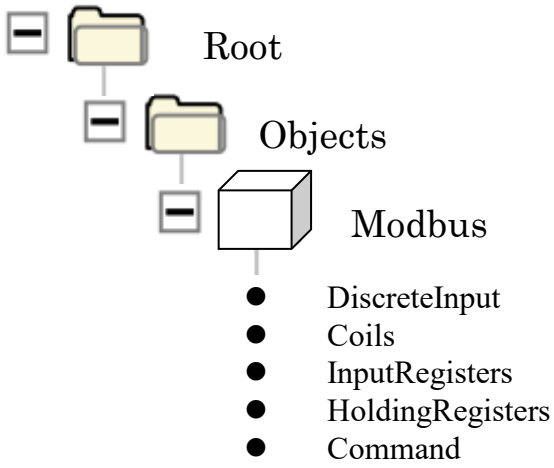
Modbus bases its data model on a series of tables.

Table	Object type	Type of
Discrete input	Single bit	READ only
Coils	Single bit	READ-WRITE
Input Registers	16-bit word	READ only
Holding Registers	16-bit word	READ-WRITE

55.2.2 OPC UA Organization of the Modbus data model

This section applies only when using the OPC UA client for HMI Device Communication with the built-in FANUC robot OPC UA Server.

The Modbus object nodes can be accessed from the Root/Object/Modbus directory as shown below.



Other folders exist just under the root folder but are not used.

The following table describes the node and data types of the five leaf nodes under the Modbus object.

Node	NameSpace	Node ID	Data type	Data size
DiscreteInput	1	301	Boolean	1 byte
Coils		302	Boolean	1 byte
InputRegisters		303	UInt16	2 byte
HoldingRegisters		304	Int16	2 byte
Command		305	String[]	128 byte

- Leaf nodes other than the Command node are arrays. The index range must be supported on the OPC UA client to support partial read or write for the leaf nodes.
- For example, it is not possible to assign a discrete Node ID to each X, Y, Z in a position register.
- The value of the node is the latest value at the read time.

Note that in the following sections, the term *HMI Device etc.* can also refer to an OPC UA client.

55.2.3 Correspondence of Modbus Address to Robot Data

Each Table has an address area of 1 to 65536. This function defines the following correspondence of Modbus address to Robot data. By accessing the Modbus address, the HMI device etc. can read or write the corresponding robot data.
OPC UA clients can read or write the desired robot data by accessing the index that is “Modbus addresses - 1“.

Table	Address range	Corresponded robot data	
Discrete input	1 - 10000	Digital input	DI[a]
	10001 - 20000	Robot input	RI[1-10000]
	20001 - 21000	UOP input	UI[1-10000]
	21001 - 21999	UOP output	UO[1-1000]
	22000 - 22999	SOP input	SI[0-999]
	23000 - 24000	SOP output	SO[0-1000]
	24001 - 25000	Weld interface digital input	WI[1-1000]
	25001 - 26000	Weld interface digital output	WO[1-1000]
	26001 - 27000	Wire soldering detector input	WSI[1-1000]
	27001 - 28000	Wire soldering detector output	WSO[1-1000]
	28001 - 65536	Not used	

Table	Address range	Corresponded robot data	
Coils	1 - 10000	Digital output	DO[1-10000]
	10001 - 20000	Robot output	RO[1-10000]
	20001 - 30000	Flag	F[1-10000]
	30001 - 65536	Not used	
Input Registers	1 - 1000	Group input	GI[1-1000]
	1001 - 2000	Group output	GO[1-1000]
	2001 - 3000	Analog input	AI[1-1000]
	3001 - 4000	Analog output	AO[1-1000]
	4001 - 65536	Not used	
Holding Registers	1 - 16384	Robot data is assigned (assigned to register by default)	
	16385 - 16448	Used in dynamic assignment to holding registers. See Section 55.4 DYNAMIC ASSIGNMENT TO HOLDING REGISTERS.	
	16449 - 65536	Not used	

When reading an address that is not used (or the corresponding robot data is not assigned) from an HMI device etc., the read value is always 0. When the address is written from an HMI device etc., the request is ignored. In these cases, this function does not return an error to the HMI device etc.

The address range of 1 to 16384 Holding Registers can be assigned to various Robot data. (→ Assignment of Holding Registers)

The HMI device etc. cannot write to the signals corresponding to Discrete input and Input Registers. It is possible to write these signals via Holding Registers by assigning WO, WSO, GO, AO in these signals to Holding Registers. (→ Assign I/O data and simulation status)

In this function, the address range is specified as from 1 to 65536. However, some HMI devices etc. specify the address range as from 0000 to FFFF (starting from 0 and specified as a hexadecimal). In this case, specify the value converted to hexadecimal by subtracting 1 from the address of this function in hexadecimal.

The maximum size that can be read or written at one time via Modbus communication is 507 words.
The amount of communication is 2 packets per communication, and 6 packets when connecting and disconnecting. However, 2 packets are added for every 256 bytes of data.

55.2.4 Modbus Function Codes

When using Modbus TCP communication (not OPC UA communication), this function supports the following function codes.

Function code name	Code
Read Coils	01h
Read Discrete Inputs	02h
Read Holding Registers	03h
Read Input Register	04h
Write Single Coil	05h
Write Single Register	06h
Write Multiple Coils	0Fh
Write Multiple Registers	10h

55.3 ASSIGNMENT OF HOLDING REGISTERS

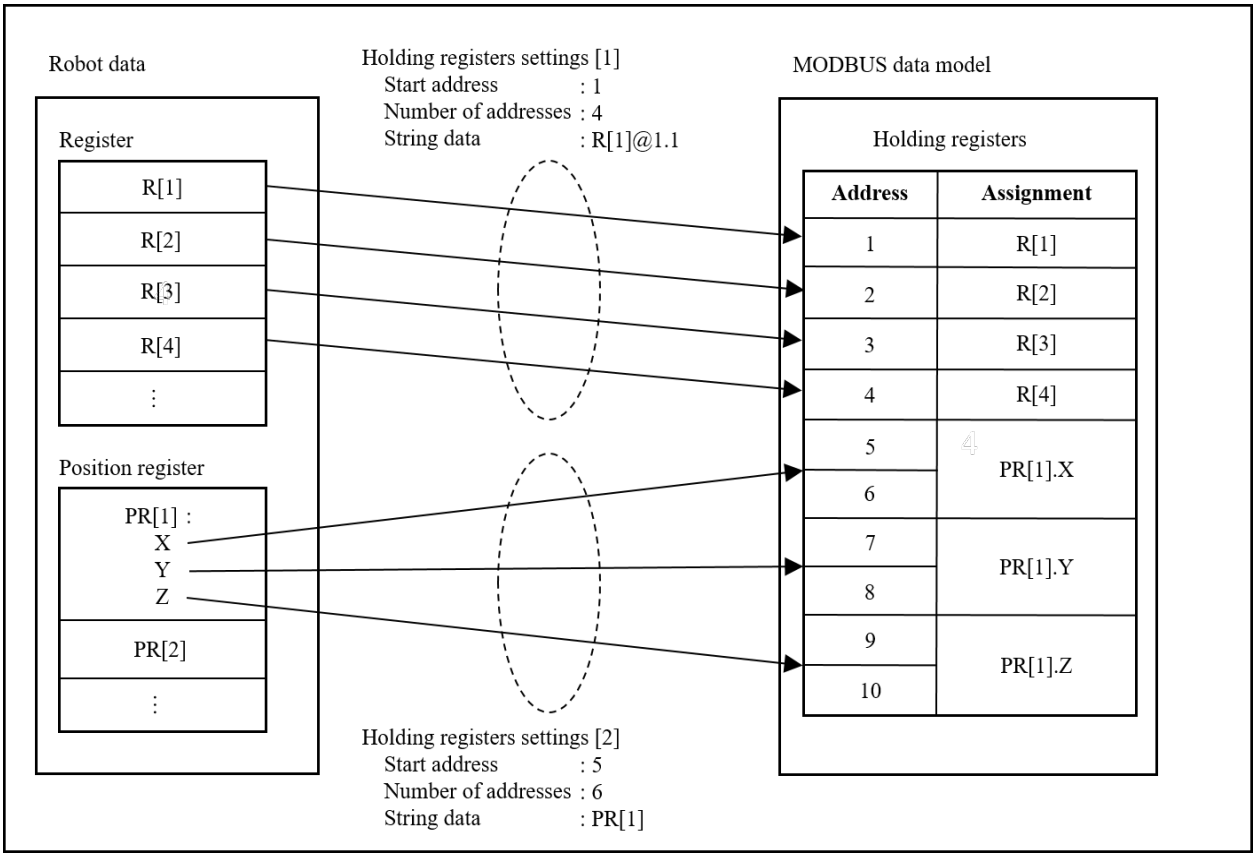
The address range 1 - 16384 (0-16383 when OPC UA) of Holding Registers can be assigned to various robot data. Each address has 2 bytes (16 bits) of memory space. It is possible to assign 32-bit data to two consecutive addresses, and to assign string data to two or more consecutive addresses.

The following robot data can be assigned to the Holding Registers.
- Register data
- Position resister data
- String register data
- Current position
- Alarm history
- Program execution status (running program name and line number)
- Comments of registers, position registers, string registers, and I/O
- I/O data and simulation status
- Integrated PMC address data
- Symbols and comments of an Integrated PMC address.

The assignment between robot data and Holding Registers is defined by setting [SETUP > Network Settings > HMI Device settings > Holding Registers]. The Holding Registers item is 80 arrays of [1]-[80], every element of which has the following items. Various robot data can be assigned to the Holding Registers by setting the Holding Registers item.

Holding Registers items	Description
Start address	This item is the starting address of Holding Registers to assign. Range: 1-16384
Number of addresses	This item is the number of Holding Registers to assign. Range: 1-16384
Data string	This item is the string to specify the robot data. This string specifies the robot data type and index . The contents of this setting vary according to the assigned data type. Refer to the description for each type of robot data. Example: R[1] : Register R[1] PR[1] : Position register PR[1] POS[1] : Current position of group 1 robot
Multiplier	This item is the multiplier for the assigned data. The Multiplier specifies the conversion type from the assigned data to the Holding Registers. The meaning of this setting varies according to the assigned data type. Refer to the description of each type of robot data. Example: If a register is assigned and the register value is 123.45, the value of the Holding Registers is as follows: When Multiplier is 1, the Holding Register is 123. When Multiplier is10, the Holding Register is 1235. When Multiplier is 0.1, the Holding Register is 12.

The following figure shows an example in which robot registers R[1-4] are assigned to Holding Register address 1 - 4 and X,Y,Z of robot position register PR[1] are assigned to Holding Register address 5 - 10.



The default setting of the Holding Registers item is as follows:

Setting items	Value
Allocation setup 1 > Start address	1
Allocation setup 1 > Number of addresses	10000
Allocation setup 1 > Data string	R[1]@1.1
Allocation setup 1 > Multiplier	1

In the default setting, the assignment of Holding Registers is as follows:

Robot data	Address of Holding Registers	Example
Register R[*x*]	*x*	R[1] ⇔ Address 1

Note: The data type is a 16-bit signed integer. The value is rounded off to the nearest whole number. The range of values is from -32768 to 32767. If the value is outside the range, it value cannot be read or written correctly.

55.3.1 Data type of Holding Registers

The data type of a Holding Register is a 16-bit integer. The data range of a Holding Register is -32768 - 32767. The value of the assigned robot data is converted as follows according to the data type of the assigned data.

Assigned data	Description
16-bit signed integer Range: -32768 - 32767	The Holding Register data is the same as the assigned data. (Big endian)
32-bit signed integer Range: -2147483648 - 2147483647	Two consecutive Holding Registers are used to express a 32-bit signed integer. The former address is the lower 16 bits of data, and the latter address is the upper 16 bits of data.

Assigned data	Description
Real	Two consecutive Holding Registers are used to express a single precision binary real format of IEEE 754. The former address is the lower 16 bits of data, and the latter address is the upper 16 bits of data. When real type robot data is assigned to Holding Registers, if Multiplier is 0, the robot data is assigned to Holding Registers as real type data. If Multiplier is not 0, the robot data is assigned to Holding Registers as a 32-bit signed integer. In this case, the Holding Registers value is the robot data multiplied by Multiplier and rounded off to the nearest whole number.
String	Multiple contiguous Holding Registers express string data. Read as UTF-8 when Japanese. One Holding Register expresses two bytes characters data. The byte order of the string data is specified in [SETUP > Network Settings > HMI Device settings > Holding Registers > Allocation setup n > Multiplier] as follows. Multiplier = 1 (0 or more): The lower byte is the first character, the upper byte is the second character. Multiplier = -1 (less than 0): The upper byte is the first character, the lower byte is the second character. The string data must end with 0. When string data is written to Holding Registers, it is necessary to write 0 at the end of the string data.

⚠ WARNING
If integer data is written to Holding Registers of real data type, the assigned robot data may be set to a value that is too large or cannot be processed. This may result in a serious impact on the system possibly causing the robot to move to an unexpected position because the assigned position register has abnormal position data. Normally, set [SETUP > Network Settings > HMI Device settings > Holding Registers > Allocation setup n > Multiplier] to a value other than 0, and assign a 32-bit signed integer data to Holding Registers.

55.3.2 Assigning Robot Registers

Set the [SETUP > Network Settings > HMI Device settings > Holding Registers] item to assign robot registers to Holding Registers, as follows:

Holding Registers items	Description
Start address	This item is the starting address of the Holding Registers to assign. Range: 1-16384
Number of addresses	This item is the number of Holding Registers to assign. One Register uses two Holding Registers. Note that you can change the number of Holding Registers for one robot register by adding “@” in Data string. Range: 1-16384

Holding Registers items	Description
Data string	This item is the string to specify robot data. Set ”R[1]” to assign Register R[1]. The number in the square brackets [ ] is the index of the register. Consecutive registers such as R[2]-R[5] can be assigned to one “Holding Registers” settings element. In this case, set the Number of addresses item to 8, because the number of registers to assign is 4. And set the Data string item to “R[2]”. The index 2 in the case means that consecutive registers are assigned in order starting with R[2]. If “@1.1” is added just after the string, the number of Holding Registers for one register is changed from 2 to 1. In this case, the accessing data size becomes 16 bits. Example: R[1]@1.1 “R[1]” means to assign robot registers from index 1. “@1.1” means one register is accessed as 16-bit data.
Multiplier	This item is the Multiplier The value accessed by an HMI device etc. is multiplied by [SETUP > Network Settings > HMI Device settings > Holding Registers > Allocation setup n > Multiplier]. When Multiplier is 0, an HMI device etc. can access Holding Registers as 32-bit REAL data. When Multiplier is not 0, an HMI device etc. accesses Holding Registers as 32-bit signed integer data, and the value is rounded off to the nearest whole number. Range: 0.0001-10000, 0 Example: Register value is 123.45. When Multiplier is 1, the Holding Register is 123. When Multiplier is 10, the Holding Register is 1235. When Multiplier is 0.1, the Holding Register is 12. When Multiplier is 0, the Holding Register is 123.45 of REAL.

For example, [SETUP > Network Settings > HMI Device settings > Holding Registers] is set as follows.

	Start address	Number of addresses	Data string	Multiplier
Allocation setup 1	1	2	R[1]@1.1	1
Allocation setup 2	3	4	R[1]	100
Allocation setup 3	7	4	R[2]	0.1
Allocation setup 4	11	2	R[1]	0

In this case, the address of Holding Registers corresponds to the robot registers as follows.

Address	Assigned data
1	R[1] as a 16-bit signed integer.
2	R[2] as a 16-bit signed integer.
3-4	R[1] multiplied by 100 as a 32-bit signed integer.
5-6	R[2] multiplied by 100 as a 32-bit signed integer.
7-8	R[2] divided by 10 as a 32-bit signed integer.
9-10	R[3] divided by 10 as a 32-bit signed integer.
11-12	R[1] as a 32-bit REAL

Allocation setup 1 defines that two Holding Registers from addresses 1 to 2 are assigned to robot registers from R[1] multiplied by 1 as a 16-bit signed integer. One Holding Register address is 16-bit data, and one register uses one Holding Register. So, address 1 is assigned to R[1] as a 16-bit signed integer and address 2 is assigned to R[2] as a 16-bit signed integer.

Allocation setup 2 defines that four Holding Registers from addresses 3 to 6 are assigned to robot registers from R[1] multiplied by 100 as a 32-bit signed integer. One robot register uses two Holding Registers. So,

addresses 3-4 are assigned to R[1] multiplied by 100 as a 32-bit signed integer and addresses 5-6 are assigned to R[2] multiplied by 100 as a 32-bit signed integer.

Allocation setup 3 defines that four Holding Registers from 7 to 10 are assigned to robot registers from R[2] divided by 10 as a 32-bit signed integer. One robot register uses two Holding Registers. So, addresses 7-8 are assigned to R[2] divided by 10 as a 32-bit signed integer and addresses 9-10 are assigned to R[3] divided by 10 as a 32-bit signed integer.

Allocation setup 4 defines that two Holding Registers from 11 to 12 are assigned to robot register R[1] as a 32-bit REAL. So, addresses 11 to 12 are assigned to R[1] as a 32-bit REAL. OPC UA clients can read this bit array as Int16, so the OPC UA client should change this value to a REAL number.

55.3.3 Assigning Position Registers

Set the [SETUP > Network Settings > HMI Device settings > Holding Registers] items to assign position registers to Holding Registers, as follows:

Holding Registers items	Description
Start address	This item is the starting address of the Holding Registers to assign. Range: 1-16384
Number of addresses	This item is the number Number of Holding Registers to assign. One position register uses 50 Holding Registers. Note that you can change the number of Holding Registers for one position register by adding “@” in “Data string”. Range: 1-16384
Data string	This item is the string to specify robot data. Set Data string to ”PR[1]” to assign position register PR[1]. The number in the square brackets [ ] is the index of the position register. Consecutive position registers like PR[2]-PR[5] can be assigned by one Holding Registers settings element. In this case, set the Number of addresses item to 200 because the number of position registers to assign is four. And set the Data string to “PR[2]”. The index 2 in this case means that consecutive position registers are assigned in order starting with PR[2]. In a multi-group system, PR[1] represents the group 1 data of PR[1]. Set PR[G2:1] to assign the group 2 data of PR[1]. If “@” is added just after the string, the specified part of the position register is assigned. This is explained later.
Multiplier	This is the Multiplier. Only REAL values like X, Y, Z and J1 are affected by this setting. Refer to the table below to see which elements are affected. The value accessed by an HMI device etc. is multiplied by [SETUP > Network Settings > HMI Device settings > Holding Registers > Allocation setup n > Multiplier]. When Multiplier is 0, it has a special meaning and an HMI device etc. can access Holding Registers as 32-bit REAL data. When Multiplier is not 0, an HMI device etc. accesses Holding Registers as 32-bit signed integers, and the value is rounded off to the nearest whole number. Range: 0.0001-10000, 0 Example: Position register element value is 123.45. When Multiplier is 1, the Holding Register is 123. When Multiplier is 10, the Holding Register is 1235. When Multiplier is 0.1, the Holding Register is 12. When Multiplier is 0, the Holding Register is 123.45 of REAL.

⚠ WARNING

If integer data is written to Holding Registers of real data type, the assigned robot data may be set to a value that is too large or cannot be processed. This may result in a serious impact on the system, possibly causing the robot to move to an unexpected position because the assigned position register has abnormal position data. Normally, set [SETUP > Network Settings > HMI Device settings > Holding Registers > Allocation setup n > Multiplier] to a value other than 0 normally, and assign 32-bit signed integer data to Holding Registers.

NOTE

If the position register is locked by the LOCK PREG instruction, the writes to the position register by this function will be ignored. To confirm whether a write is successful, read back the position data just after the write and confirm that the correct value is set.

One position register uses 50 Holding Registers. The contents of the 50 Holding Registers are as follows:

Address	Description		Effect of "Multiplier"
Cartesian data			
1-2	X	32-bit signed integer or real (mm)	Yes
3-4	Y	32-bit signed integer or real (mm)	Yes
5-6	Z	32-bit signed integer or real (mm)	Yes
7-8	W	32-bit signed integer or real (deg)	Yes
9-10	P	32-bit signed integer or real (deg)	Yes
11-12	R	32-bit signed integer or real (deg)	Yes
13-14	E1	32-bit signed integer or real (mm, deg)	Yes
15-16	E2	32-bit signed integer or real (mm, deg)	Yes
17-18	E3	32-bit signed integer or real (mm, deg)	Yes
19	FLIP	16-bit signed integer (1:Flip, 0:Non flip)	No
20	LEFT	16-bit signed integer (1:Left, 0:Right)	No
21	UP	16-bit signed integer (1:Up, 0:Down)	No
22	FRONT	16-bit signed integer (1:Front, 0:Back)	No
23	TURN4	16-bit signed integer (-128-127)	No
24	TURN5	16-bit signed integer (-128-127)	No
25	TURN6	16-bit signed integer (-128-127)	No
26	VALIDC	16-bit signed integer (→NOTE1)	No
Joint data			
27-28	J1	32-bit signed integer or real (mm, deg)	Yes
29-30	J2	32-bit signed integer or real (mm, deg)	Yes
31-32	J3	32-bit signed integer or real (mm, deg)	Yes
33-34	J4	32-bit signed integer or real (mm, deg)	Yes
35-36	J5	32-bit signed integer or real (mm, deg)	Yes
37-38	J6	32-bit signed integer or real (mm, deg)	Yes
39-40	J7	32-bit signed integer or real (mm, deg)	Yes
41-42	J8	32-bit signed integer or real (mm, deg)	Yes
43-44	J9	32-bit signed integer or real (mm, deg)	Yes
45	VALIDJ	16-bit signed integer (→NOTE2)	No
Frame number			
46	UF	16-bit signed integer (-1-62) (→NOTE3)	No
47	UT	16-bit signed integer (-1-30) (→NOTE4)	No
48-50	Reserve		No

NOTE
1 VALIDC shows whether the position data is valid Cartesian data. VALIDC is 0 in the following cases, otherwise it is set to 1:
   - The position data has uninitialized data (displayed as "******" on the teach pendant).
   - The position data is in Joint representation and cannot be converted to Cartesian.

   If the HMI device etc. writes any data to VALIDC, the position representation is changed to Cartesian representation. In this case, the value to be written can be anything.
2 VALIDJ indicates whether the position data is valid Joint data. VALIDJ is 0 in the following cases, otherwise it is set to 1.
   - The position data has uninitialized data (displayed "******" on the teach pendant).
   - The position data is in Cartesian representation and cannot be converted to Joint.

   If the HMI device etc. writes any data to VALIDJ, the position representation is changed to Joint representation. In this case, the value to be written can be anything.
3 UF is the user frame number.
   If UF is 0, world frame is used.
   If UF is -1, the currently selected user fame is used.
   The UF of a position register is always -1. On the *i*Pendant, it is displayed as F.
   This value cannot be changed.
4 UT is the tool frame number
   If UT is 0, the mechanical interface frame is used.
   If UT is -1, the currently selected tool fame is used.
   The UT of a position register is always -1. On the *i*Pendant, it is displayed as F.
   This value cannot be changed.

- Position registers have two representations: Cartesian and Joint. If the detail of the position register is displayed as X,Y,Z,W,P,R, it is Cartesian representation. If it is displayed as J1-6, it is Joint representation.
- An HMI device etc. can always access any element, regardless of the current representation. Position representation is not changed by reading from an HMI device etc. However, if the representation of the position register and the representation of the element accessed from an HMI device etc. are different, note the following:
- If the position data is outside of the stroke limit, or if there is an uninitialized element, this position data cannot be converted to another representation. In this case, all elements in the part of the data that are not in the current representation will be set to 0.
  For example, if a position register is in Cartesian representation and X is 10000, and it is outside the stroke limit, J1-J9 will be read as all 0s by an HMI device etc. In this case, the elements of the Cartesian part or UF/UT part can still be read correctly.
- If the representation of position register and the representation of the accessed element from an HMI device etc. are different, the communication response time will be increased, because position representation conversion takes so much time. If an HMI device etc. reads many position registers, this conversion time may have a significant effect on response time.

An uninitialized element of position data is displayed as "******" on the position register menu of the teach pendant. These uninitialized elements are read as 0 from an HMI device etc. Read VALIDC or VALIDJ to check whether the value is 0 or uninitialized. If position data has an uninitialized element, VALIDJ and VALIDC are 0.

If you write data to an element in the Cartesian part from an HMI device etc., the position representation becomes Cartesian. If you write data to an element in the Joint part from an HMI device etc., the position representation becomes Joint. If you write data to a UF or UT from an HMI device etc., the position representation is not changed.

Extract a part by adding "@" to a "Data string"

If you add "@" to a "Data string", the specified part of the data structure is extracted. This features has already been used for register assignment (R[1]@1.1).

For example, suppose you need to access only X, Y and Z of PR[1-3]. In this case, you would normally set [SETUP > Network Settings > HMI Device settings > Holding Registers] as follows:

	Start address	Number of addresses	Data string	Multiplier
Allocation setup 1	1	150	PR[1]	1

When this setting is made, the correspondence between the position registers and Holding Registers is as follows:

Address	Assigned data
1-2	X of PR[1] as a 32-bit signed integer
3-4	Y of PR[1] as a 32-bit signed integer
5-6	Z of PR[1] as a 32-bit signed integer
51-52	X of PR[2] as a 32-bit signed integer
53-54	Y of PR[2] as a 32-bit signed integer
55-56	Z of PR[2] as a 32-bit signed integer
101-102	X of PR[3] as a 32-bit signed integer
103-104	Y of PR[3] as a 32-bit signed integer
105-106	Z of PR[3] as a 32-bit signed integer

Although the actual read data is only 18 Holding Registers, this assignment occupies 150 Holding Registers. Also, some HMI devices access addresses 7-50 of Holding Registers that are not necessary to read, because reading one large amount of data is more efficient than reading several small amounts of data from a communications point of view. However, addresses 27-45 include Joint a representation part, and if position data is in Cartesian representation, representation conversion will be required to read this unnecessary data.

To communicate efficiently, set [SETUP > Network Settings > HMI Device settings > Holding Registers] as follows:

	Start address	Number of addresses	Data string	Multiplier
Allocation setup 1	1	18	PR[1]@1.6	1

When this setting is made, the correspondence between position registers and Holding Registers is as follows:

Address	Accessed data
1-2	X of PR[1] as a 32-bit signed integer
3-4	Y of PR[1] as a 32-bit signed integer
5-6	Z of PR[1] as a 32-bit signed integer
7-8	X of PR[2] as a 32-bit signed integer
9-10	Y of PR[2] as a 32-bit signed integer
11-12	Z of PR[2] as a 32-bit signed integer
13-14	X of PR[3] as a 32-bit signed integer
15-16	Y of PR[3] as a 32-bit signed integer
17-18	Z of PR[3] as a 32-bit signed integer

The change in Holding Registers settings is that "@1.6" is added in Data string and Number of addresses is changed from 150 to 18. With this change, the number of Holding Registers for one position register is changed from 50 to 6. The "6" in "@1.6" is the number of Holding Registers for one position register. And the "1" in "@1.6" is the starting address of the extracting part in the position data structure.
By specifying "@" in Data string, you can extract the specified part of a position data structure.

@1.6

The size of the extracting part (the number of Holding Registers for one position)

The starting address of the extracting part in the position data structure.

You can specify "@" not only for position registers, but also for all data that is assigned by [SETUP > Network Settings > HMI Device settings > Holding Registers settings].

The following is another example, which assigns J1-J6 of PR[1-3].

	Start address	Number of addresses	Data string	Multiplier
Allocation setup 1	1	96	PR[3]@27.12	1

55.3.4 Assigning String Registers

Set the [SETUP > Network Settings > HMI Device settings > Holding Registers] items to assign string registers to Holding Registers as follows:

Holding Registers items	Description
Start address	This item is the starting address of Holding Registers to assign. Range: 1-16384
Number of addresses	This item is the number of Holding Registers to assign. One String Register uses 40 Holding Registers. It is possible to deal with the first 80 characters of one string register with the default setting. You can change the number of Holding Registers for one string register by adding "@" in Data string. Range: 1-16384
Data string	This item is the string to specify robot data. Set this to "SR[1]" to assign String Register SR[1]. The number in the square brackets [ ] is the index of the string register. Consecutive string registers such as SR[2]-SR[5] can be assigned by one Holding Registers settings element. In this case, set the Number of addresses item to 160, because the number of string registers to assign is four. Also, set Data string to "SR[2]". The index 2 in this case means that consecutive string registers are assigned in order starting with SR[2]. If "@" is added just after the string, the specified part of the string register is assigned. Example: SR[1]@1.50 "SR[1]" means to assign string registers starting from index 1. "@1.50" means the first 100 characters of string register data are assigned.
Multiplier	This item is the multiplier that specifies the byte order of the string. Range: 1,-1

55.3.5 Assign the Current Position

Set the [SETUP > Network Settings > HMI Device settings > Holding Registers] items to assign the current position to Holding Registers as follows:

Holding Registers items	Description
Start address	This item is the starting address of the Holding Registers to assign. Range: 1-16384
Number of addresses	This item is the number of Holding Registers to assign. The current position uses 50 Holding Registers.

Holding Registers items	Description
Number of addresses	Note that you can change the number of Holding Registers for the current position by adding “@” in Data string. Range: 1-16384
Data string	This item is the string to specify the robot data. Set Data string to ”POS[0]” to assign the current position. The number in the square brackets [ ] is the user frame number. When the user frame number is 0, the current position in World frame is assigned. This is the same as when "WORLD" is selected in the current position screen on the teach pendant. When the user frame number is -1, the current position in the User frame that is currently selected is assigned. This is the same as when "USER" is selected in the current position screen on the teach pendant. When the user frame number is 1-61, the current position of the specified user frame is assigned. The data structure is the same as a position register. In a multi-group system, POS[0] is the current position of the group 1 robot. Set POS[G2:0] to assign the current position of the group 2 robot. If “@” is added just after the string, the specified part of the current position is assigned.
Multiplier	This item is the multiplier. Only REAL values like X, Y, Z and J1 are affected by this setting. Refer to the table below to see which elements are affected. The value of each element of the current position multiplied by the value of [SETUP > Network Settings > HMI Device settings > Holding Registers > Allocation setup n > Multiplier] is read and written. If Multiplier is 0, it means that the data is assigned to Holding Registers as 32-bit REAL data. If Multiplier is not 0, it means that the data is assigned to Holding Registers as a 32-bit signed integer, and the value is rounded off to the nearest whole place. Range: 0.0001-10000, 0 Example: Current position member value is 123.45. When Multiplier is 1, the Holding Register is 123. When Multiplier is 10, the Holding Register is 1235. When Multiplier is 0.1, the Holding Register is 12. When Multiplier is 0, the Holding Register is 123.45 of REAL.

NOTE
The current position is read only. If you write it, nothing will happen.

The current position uses 50 Holding Registers, and the data structure is the same as a position register.
The UT of the current position is always -1.
The UF of the current position is the specified user frame number in Data string.

The current position is not assigned sequentially. For example, the following setting in Holding Registers assigns the current position of user frame 1 to addresses 1-50, but the current position of user frame 2 is not assigned to addresses 51-100. Set POS[2] in another element of Holding Registers settings.

	Start address	Number of addresses	Data string	Multiplier
Allocation setup 1	1	100	POS[1]	1

55.3.6 Assigning Alarm History

Set the [SETUP > Network Settings > HMI Device settings > Holding Registers] items to assign Alarm history to Holding Registers as follows:

Holding Registers items	Description
Start address	This item is the starting address of the Holding Registers to assign. Range: 1-16384
Number of addresses	This item is the number of Holding Registers to assign. One alarm uses 100 Holding Registers. You can change the number of Holding Registers for one alarm by adding “@” in Data string. Range: 1-16384
Data string	This item is the string to specify the robot data. Set Data string to ”ALM[1]” to assign an alarm. The number in the square brackets [ ] is the line number in the alarm menu. The most recent alarm is 1. “ALM[1]” assigns the active alarm. Alarms displayed on the active alarm screen are assigned. “ALM[E1]” assigns the alarm history. Alarms displayed on the alarm history screen are assigned. “ALM[M1]” assigns the motion alarm history. Alarms displayed on the motion alarm screen are assigned. “ALM[S1]” assigns the system alarm history. Alarms displayed on the system alarm menu are assigned. “ALM[A1]” assigns the application alarm history. Alarms displayed on the application alarm menu are assigned. “ALM[P1]” assigns the password log. The password logs displayed on the password log screen are assigned.
Multiplier	This item specifies the byte order of the string. Range: 1,-1

NOTE
Alarm history is read only. If you write it, nothing will happen.

One alarm uses 100 Holding Registers. The contents of the 100 Holding Registers are as follows:

Address	Description
1	Facility code 16 bits signed integer For example, for alarm SRVO-001, the facility code is 11, which means SRVO. Refer to Section 1.2.2 “Facility Name and Code" in the OPERATOR'S MANUAL (Alarm Code List) (B-84464EN-1) for more information on the number of the facility code.
2	Alarm number 16-bit signed integer For alarm SRVO-001, the alarm number is 1.
3	Facility code of the Cause Code 16-bit signed integer When an alarm occurs, the alarm message is displayed on the top line of the teach pendant screen. Sometimes a message is also displayed on the second line, which is the cause code. This displays the facility code of the cause code. If the alarm does not have a cause code, this is 0. For example, for SRVO-001, the facility code is 11, which means SRVO. Refer to Section 1.2.2 “Facility Name and Code" in the OPERATOR'S MANUAL (Alarm Code List)

Address	Description
3	(B-84464EN-1) for more information on the number of the facility code.
4	Alarm number of the cause code 16-bit signed integer If the alarm does not have a cause code, this is 0. For SRVO-001, the alarm number of the cause code is 1.
5	Alarm severity 16-bit signed integer The value displays the alarm severity: NONE 128 WARN 0 PAUSE.L 2 PAUSE.G 34 STOP.L 6 STOP.G 38 SERVO 54 ABORT.L 11 ABORT.G 43 SERVO2 58 SYSTEM 123
6	Occurred Time (year) 16-bit signed integer
7	Occurred Time (month) 16-bit signed integer
8	Occurred Time (day) 16-bit signed integer
9	Occurred Time (hour, 24 hour system) 16-bit signed integer
10	Occurred Time (minutes) 16-bit signed integer
11	Occurred Time (second) 16-bit signed integer
12-51	Alarm message string of up to 80 characters You can read alarm message strings. It shows the same string that is displayed on top line of the teach pendant such as “SRVO-001”.
52-91	Cause code alarm message string of up to 80 characters This is the alarm message of the cause code.
92-100	Alarm severity word 80 characters string This is the string of the alarm severity, such as “WARN”.

The facility code and alarm number of RESET are read as 0. The alarm message is “RESET”.

If there are only two lines on the active alarm screen, all alarm elements after ALM[3] are set to 0.

In string elements, addresses after the string are 0.
One alarm uses 100 Holding Registers, so if you want to read only the facility codes and alarm numbers, most of the 100 Holding Registers area is not used. To communicate efficiently, use “@”.

Example: [SETUP > Network Settings > HMI Device settings > Holding Registers] is set as follows.

	Start address	Number of addresses	Data string	Multiplier
Allocation setup 1	1	12	ALM[E1]@1.4	1

When this setting is made, the correspondence between the address and alarm data is as follows:

Address	Assigned data
1	Alarm ID of alarm1
2	Alarm number of alarm1
3	Alarm ID of cause code of alarm1
4	Alarm number of cause code of alarm1
5	Alarm ID of alarm2
6	Alarm number of alarm2
7	Alarm ID of cause code of alarm2
8	Alarm number of cause code of alarm2
9	Alarm ID of alarm3

Address	Assigned data
10	Alarm number of alarm3
11	Alarm ID of cause code of alarm3
12	Alarm number of cause code of alarm3

55.3.7 Assigning Program Execution Status

Set the [SETUP > Network Settings > HMI Device settings > Holding Registers] items to assign program execution status to Holding Registers as follows:

Holding Registers items	Description
Start address	This item is the starting address of the Holding Registers to assign. Range: 1-16384
Number of addresses	This item is the number of Holding Registers to assign. One task uses 18 Holding Registers. You can change the number of Holding Registers for one task by adding "@" in the Data string. Range: 1-16384
Data string	This item is the string to specify the robot data. Set Data string to "PRG[1]" to assign program execution status. The number in the square brackets [ ] is the task number. In a single task system, program execution status can always be assigned by PRG[1]. In a multi-tasking system, when two tasks are running at the same time, PRG[1] and PRG[2] show the execution status of each task. Which task is PRG[1] is determined by execution timing and communication. However, the task that is read as PRG[1] is always read as PRG[1] until it is aborted.
Multiplier	This item specifies the byte order of the string. Range: 1,-1

NOTE
Program execution status is read only. If you write it, nothing will happen.

One task uses 18 Holding Registers. The contents of the 18 Holding Registers are as follows:

Address	Description
1-8	Program name 16-character string (When the program name is long, truncated to 15-characters + NULL) This is the name of the running program. When sub program is called, it shows sub program name.
9	Line number 16-bit signed integer. This is the execution line number. When a sub-program is called, this shows the line number of the sub-program.
10	Execution status 16-bit signed integer Aborted 0 Paused 1 Running 2
11-18	Parent program name 16-character string This is the name of the started program When a sub-program is not called, this is the same as the program name.

When a program is aborted, all elements are 0.

By assigning each string to a Data string, the type of the program can be selected. Also, the meaning of the line number changes depending on the situation.

Data string	Program name and line number
PRG[1]	Program name This is the name of the running program. Line number This is the execution line number.
PRG[M1]	Program name If the running program is a MACRO program, this shows the name of the parent program that calls the running program. If the parent program is also a MACRO program, this shows the name of the parent program that is not a MACRO program. Line number This is the line number of the CALL instruction that calls the running program. If all parent programs are MACRO programs this is 0.
PRG[K1]	Program name If the running program is a KAREL program, this shows the name of the parent program that calls the running program. If the parent program is also a KAREL program, this shows the name of the parent program that is not a KAREL program. Line number This is the line number of the CALL instruction that calls the running program. If all the parent programs are KAREL program this is 0.
PRG[MK1] or PRG[KM1]	Program name If the running program is a MACRO or KAREL program, this shows the name of the parent program that calls the running program. If the parent program is also a MACRO or KAREL program, this shows the name of the parent program that is not a MACRO nor KAREL program. Line number This is the line number of the CALL instruction that calls the running program. If all of the parent programs are MACRO or KAREL programs this is 0.

By adding "@", a part of this structure is assigned to Holding Registers.
Example: [SETUP > Network Settings > HMI Device settings > Holding Registers] is set as follows.

	Start address	Number of addresses	Data string	Multiplier
Allocation setup 1	1	4	PRG[1]@9.2	1

When this setting is made, the correspondence between the address and program execution is as follows:.

Address	Accessed data
1	Line number of task 1
2	Execution status of task 1
3	Line number of task 2.
4	Execution status of task 2.

55.3.8 Assigning Data ID

Set [SETUP > Network Settings > HMI Device settings > Holding Registers] items to assign Data ID to Holding Registers as follows:

Holding Registers settings items	Description
Start address	This item is the starting address of the Holding Registers to assign. Range: 1-16384

Holding Registers settings items	Description
Number of addresses	This item is the number of Holding Registers to assign. The number of Holding Registers for one Data ID is defined by the data type of the Data ID. Refer the following data type list. You can change the number of Holding Registers for one Data ID by adding "@" in the Data string. Range: 1-16384
Data string	This item is the string to specify robot data. If an array Data ID is set, the array elements will be assigned consecutively, like registers. A KAREL string can be specified in the following format: $[KAREL program name]variable name.
Multiplier	Multiplier is defined according to the data type of the Data ID. Refer the following data type list.

To access to Data IDs, change the data type of the assigned Holding Registers and meaning of Multiplier in Holding Registers settings according to the data type of the Data ID. The following is a list of data types of Data ID that can be assigned to Holding Registers, and it also explains the number of Holding Registers and the meaning of Multiplier.

Data type of Data ID	Number of Holding Registers for one Data ID	Meaning of Multiplier
INTEGER 32-bit signed integer	2	The data is multiplied by [SETUP > Network Settings > HMI Device settings > Holding Registers > [n] > Multiplier]. The data is accessed as a 32-bit signed integer, rounded off to the nearest whole number. If Multiplier is 0, it is same as if Multiplier is 1.
REAL 32-bit real	2	The data is multiplied by Multiplier. If Multiplier is 0, it means that the data is accessed as 32-bit REAL data. If it is not 0, the data is accessed as 32-bit signed integer, rounded off to the nearest whole number.
BOOLEAN TRUE/FALSE	2	Multiplier is not used for this data type. The data is accessed as a 32-bit signed integer. If the value is TRUE, the Holding Register is 1. If it is 0, the Holding Register is 0. If 0 is written, the Data ID is FALSE. If 0 is not written, the Data ID is TRUE.
POSITION JOINTPOS JOINTPOS1-9 XYZWPR XYZWPREXT Position data	50	The data structure is the same as a position register. The meaning of Multiplier is also same as a position register.
STRING String data	40	This item specifies the byte order of a string.

A description and data type of each Data ID can be found in the manual for each function.
In addition, the data type of Data ID can be confirmed from dataid.csv. For the operation to obtain dataid.csv, refer to the "Operator's Manual (Basic Function)"(B84464EN) "B.13.3 Data ID Listing File (DATAID.CSV)".

> **⚠ WARNING**
> If integer data is written to the Holding Registers of real data type, the assigned robot data may be set to a value that is too large or cannot be processed. This may result in a serious impact on the system, possibly causing the robot to move to an unexpected position because the assigned POSITION type variable has abnormal position data. Normally, set [SETUP > Network Settings > HMI Device settings > Holding Registers > [n] > Multiplier] to a value of not 0, and assign 32-bit signed integer data to Holding Registers.

Special Data ID

The following table lists special Data ID that are not actual Data ID but can be assigned to Holding Registers to read robot internal information. The special Data ID are read only. The values cannot be changed when the assigned Holding Registers are written.

Data ID name	Data Type	Description
$RBTIF.MEM_TPP_TOT	INTEGER	This is the total storage capacity in TPP. The units displayed in the holding register are bytes.
$RBTIF.MEM_TPP_FREE	INTEGER	This is the available storage capacity in TPP. The units displayed in the holding register are bytes.
$RBTIF.MEM_TEMP_TOT	INTEGER	This is the total storage capacity in TEMP. The units displayed in the holding register are bytes.
$RBTIF.MEM_TEMP_FREE	INTEGER	This is the available storage capacity in TEMP. The units displayed in the holding register are bytes.
$RBTIF.MEM_PERM_TOT	INTEGER	This is the total storage capacity in PERM. The units displayed in the holding register are bytes.
$RBTIF.MEM_PERM_FREE	INTEGER	This is the available storage capacity in PERM. The units displayed in the holding register are bytes.
$RBTIF.MEM_FROM_TOT	INTEGER	This is the total storage capacity of the F-ROM module. The units displayed in the holding register are bytes.
$RBTIF.MEM_FROM_FREE	INTEGER	This is the available storage capacity of the F-ROM module. The unit displayed in the holding register are bytes.
$RBTIF.SERIAL_NUM	STRING	This is the robot serial number.
$RBTIF.SOFT_VERSION	STRING	This is the robot software version.
$RBTIF.ROBOT_ID_GRP1 - $RBTIF.ROBOT_ID_GRP8	STRING	This is the robot ID. The last number corresponds to the number of groups. If an invalid last number is set, the value will be set to 0.
$RBTIF.OVERRIDE	INTEGER	This is the override. Range is 1 to 100%.
$RBTIF.STIME_PWR_TOT1 - $RBTIF.STIME_PWR_TOT8	INTEGER	This is the total time of controller power on. The last number corresponds to the number of groups. If an invalid last number is set, the value will be set to 0. The units displayed in the holding register are minutes.
$RBTIF.STIME_SRV_TOT1 - $RBTIF.STIME_SRV_TOT8	INTEGER	This is the total servo on time. The last number corresponds to the number of groups. If an invalid last number is set, the value will be set to 0. The units displayed in the holding register are minutes.
$RBTIF.STIME_RUN_TOT1 - $RBTIF.STIME_RUN_TOT8	INTEGER	This is the total program running time. The last number corresponds to the number of groups. If an invalid last number is set, the value will be set to 0. The units displayed in the holding register are minutes.
$RBTIF.STIME_WIT_TOT1 - $RBTIF.STIME_WIT_TOT8	INTEGER	This is the total program waiting time. The last number corresponds to the number of groups. If an invalid last number is set, the value will be set to 0. The units displayed in the holding register are minutes.

Data ID name	Data Type	Description
$RBTIF.STIME_SHM_TOT1 - $RBTIF.STIME_SHM_TOT8	INTEGER	This is the total brake release time. The last number corresponds to the number of groups. If an invalid last number is set, the value will be set to 0. The units displayed in the holding register are minutes.
$RBTIF.STIME_PWR_LAP1 - $RBTIF.STIME_PWR_LAP8	INTEGER	This is the lap time of controller power on. The last number corresponds to the number of groups. If an invalid last number is set, the value will be set to 0. The units displayed in the holding register are seconds.
$RBTIF.STIME_SRV_LAP1 - $RBTIF.STIME_SRV_LAP8	INTEGER	This is the lap time of servo on. The last number corresponds to the number of groups. If an invalid last number is set, the value will be set to 0. The units displayed in the holding register are seconds.
$RBTIF.STIME_RUN_LAP1 - $RBTIF.STIME_RUN_LAP8	INTEGER	This is the lap time of program running. The last number corresponds to the number of groups. If an invalid last number is set, the value will be set to 0. The units displayed in the holding register are seconds.
$RBTIF.STIME_WIT_LAP1 - $RBTIF.STIME_WIT_LAP8	INTEGER	This is the lap time of program waiting. The last number corresponds to the number of groups. If an invalid last number is set, the value will be set to 0. The units displayed in the holding register are seconds.
$RBTIF.STIME_SHM_LAP1 - $RBTIF.STIME_SHM_LAP8	INTEGER	This is the lap time of brake release. The last number corresponds to the number of groups. If an invalid last number is set, the value will be set to 0. The units displayed in the holding register are seconds.
$RBTIF.Q_CURRENT11 - $RBTIF.Q_CURRENT19, $RBTIF.Q_CURRENT21 - $RBTIF.Q_CURRENT29, … $RBTIF.Q_CURRENT81 - $RBTIF.Q_CURRENT89	REAL	This is the Q phase current of each axis. Since the torque constant is not disclosed, it cannot be converted to torque, but the Q-phase current is data proportional to the torque. The last number corresponds to the number of groups and axis number. If an invalid last number is set, the value will be set to 0. The units displayed in the holding register are amperes. Example: Data string: $RBTIF.Q_CURRENT23 Group number is 2, Axis number is 3.
$RBTIF.CURRENT11 - $RBTIF.CURRENT19, $RBTIF.CURRENT21 - $RBTIF.CURRENT29, … $RBTIF.CURRENT81 - $RBTIF.CURRENT89	REAL	This is the current of each axis. The last number corresponds to the number of groups and axis number. If an invalid last number is set, the value will be set to 0. The units displayed in the holding register are amperes. Example: Data string: $RBTIF.CURRENT23 Group number is 2, Axis number is 3.

55.3.9 Assigning comments

Set [SETUP > Network Settings > HMI Device settings > Holding Registers] items to assign comments for registers, position registers, string registers, and I/O to Holding Registers, as follows:

Holding Registers items	Description
Start address	This item is the starting address of the Holding Registers to assign. Range: 1 to 16384
Number of addresses	This item is the number of Holding Registers to assign. One comment uses 40 Holding Registers. Set this according to the number of comments to assign. You can change the number of Holding Registers for one comment by adding "@" in the Data string. Range: 1 to 16384
Data string	This item is the string to specify robot data. Set Data string to "R[C1]" to assign the comment of a register. The number in square brackets [ ] is index of the register. To assign a comment for a position register, string register and I/O, set the following strings: Register R[C1] Position register PR[C1] String register SR[C1] DI DI[C1] DO DO[C1] RI RI[C1] RO RO[C1] UI UI[C1] UO UO[C1] SI SI[C1] SO SO[C1] WI WI[C1] WO WO[C1] WSI WSI[C1] WSO WSO[C1] GI GI[C1] GO GO[C1] AI AI[C1] AO AO[C1] The same Holding Registers settings element can be assigned to multiple comments in consecutive registers such as R[2]-R[5]. In this case, set the Number of addresses item to 160, because the number for one comment to assign is 40. Also set R[C2] to Data string. The index 2 is the starting index of the consecutive assignment. If "@" is added just after the string, the specified part of the comment is assigned. Example: R[C1]@1.5 R[C1] means to assign comment of registers starting from index 1. @1.5 means the first 10 characters of the comment are assigned.
Multiplier	This item specifies the byte order of strings. Range: 1,-1

55.3.10 Assigning I/O data and simulation status

Set [SETUP > Network Settings > HMI Device settings > Holding Registers] items to assign I/O data and simulation status to Holding Registers as described in this section.
Input signals cannot be changed using this function even though they are assigned to Holding Registers. To change an input signal using this function, assign the input signal to a flag in rack 34, slot 1, and change the flag.

Holding Registers items	Description
Start address	This is the starting address of the Holding Registers to assign. Range: 1-16384
Number of addresses	This is the number of Holding Registers to assign. The number of Holding Registers for one I/O value or simulation status depends on the Multiplier setting: When Multiplier is 1, one I/O value or simulation status uses one Holding Register. When Multiplier is 0, one Holding Register includes 16 I/O values or simulation states as bit images. Note that the GI/O and AI/O values use one Holding Register, even if Multiplier is 0. Range: 1-16384
Data string	This is the string to specify robot data. Set Data string to "DI[1]" to assign DI[1] data. The number in the square brackets [ ] is index of the DI. To assign I/O data or simulation status of the other types of I/O, set the following strings: DI: DI[1], DI[S1] DO: DO[1], DO[S1] RI: RI[1], RI[S1] RO: RO[1], RO[S1] UI: UI[1] UO: UO[1] SI: SI[1] SO: SO[1] WI: WI[1], WI[S1] WO: WO[1], WO[S1] WSI: WSI[1], WSI[S1] WSO: WSO[1], WSO[S1] GI: GI[1], GI[S1] GO: GO[1], GO[S1] AI: AI[1], AI[S1] AO: AO[1], AO[S1] Flag: F[1] Marker: M[1] Example: Assign DI[11]-DI[42]: When Multiplier is 1, set the Number of addresses item to 32 because the number of DI is 32, and set Data string to DI[11]. In this case, index 11 means that the starting index of the assignment is 11. When Multiplier is 0, 16 DI data are assigned to one Holding Register, so set the Number of addresses item to 2, and set Data string to DI[11].
Multiplier	One Holding Register has one I/O data or 16 I/O data as a bit image. One I/O data is assigned to one Holding Register 1 16 I/O data are assigned to one Holding Register 0

With the Multiplier setting, you can select whether one I/O data is assigned to one Holding Register or 16 I/O data are assigned to one Holding Register.

When Multiplier is 1, one I/O data or one simulation status is assigned to one Holding Register. When the value is ON, the corresponding Holding Register is set to 1. When the value is OFF, the corresponding Holding Register is set to 0.

When Multiplier is 0, 16 I/O data or 16 simulation statuses are assigned to one Holding Register. When the value is ON, the corresponding bit of the Holding Register is set to 1. When the value is OFF, the corresponding bit of the Holding Register is set to 0. I/O of a lower index is assigned to a lower bit of a Holding Register.
Example: When DI[1-16] is assigned to the Holding Register, if only DI[1] is ON and the others are OFF, the Holding Register is 1. If only DI[16] is ON and the others are OFF, the Holding Register is 32768 (for an unsigned 16-bit integer).

55.3.11 Assigning Integrated PMC address data

Set [SETUP > Network Settings > HMI Device settings > Holding Registers] to assign Integrated PMC address data to Holding Registers as follows:

Holding Registers items	Description
Start address	This is the starting address of the Holding Registers to assign. Range: 1-16384
Number of addresses	This is the number of Holding Registers to assign. The number of Holding Registers corresponding to the PMC address varies according to whether the specified PMC address is a byte address or bit address. When bit address is specified, one bit PMC address is assigned to one Holding Register. When byte address is specified, a two-byte PMC address is assigned to one Holding Register. Range: 1-16384
Data string	This is the string to specify robot data. Set Data string to ”PMC1:X0” to assign data of PMC address X0. The "PMC1" part is the PMC path number. "PMCS" means DCS Safety PMC. The data of a bit address is assigned by specifying a bit address such as ”PMC1:X0.0”. Example: Assign X0.0-X1.7 data of first path PMC When the Number of addresses is set to 1 and Data string is set to PMC1:X0, two bytes address X0-X1 are assigned to one Holding Register address. The lower byte is assigned to X0, and the upper byte is assigned to X1. When the Number of addresses item is set to 16 and Data string is set to PMC1:X0.0, 16 bits X0.0-X1.7 are assigned to 16 Holding Registers. X0.0 is assigned to Address 1, X0.1 is assigned to Address 2, X1.7 is assigned to Address 16.
Multiplier	Not used

NOTE
1 If K900.4 (Change address value) is 0, writes to PMC address data are ignored. To check if the write is ignored, read back the PMC address after the write and confirm that the correct value is set.
2 The address of the DCS safety PMC is read-only. It cannot be written to.

Example: [SETUP > Network Settings > HMI Device settings > Holding Registers] is set as follows.

	Start address	Number of addresses	Data string	Multiplier
Allocation setup 1	1	2	PMC1:R0.0	1
Allocation setup 2	3	2	PMC1:D0	1

The correspondence between Holding Registers and robot data with this setting is as follows:
Since bit addresses are assigned to address 1 - 2, 1 bit of the PMC address is assigned to 1 Holding Register consecutively.
Address 3 – 4 are byte addresses, so 2 bytes of PMC addresses are consecutively assigned to 1 Holding Register.

Address	Assigned robot data
1	R0.0
2	R0.1
3	D0 - D1 (Lower byte:D0, Upper byte:D1)
4	D2 - D3 (Lower byte:D2, Upper byte:D3)

55.3.12 Assigning Symbols and Comments for Integrated PMC addresses

Set [SETUP > Network Settings > HMI Device settings > Holding Registers] to assign symbols and comments for Integrated PMC addresses to Holding Registers as follows:

Holding Registers items	Description
Start address	This is the starting address of Holding Registers to assign. Range: 1-16384
Number of addresses	This is the number of Holding Registers to assign. One symbol or comment uses 40 Holding Registers. Set this according to the number of comments to assign. You can change the number of Holding Registers for one symbol and comment by adding “@” in Data string. Range: 1-16384
Data string	This is the string to specify robot data. Set Data string to ”PMC1:S:X0” to assign the symbol of PMC address X0. Set Data string to “PMC1:C:X0” to assign a comment for the PMC address. The "PMC1" part is the PMC path number. "PMCS" means DCS Safety PMC. Symbols and comments of bit addresses are assigned by specifying bit addresses such as ”PMC1:S:X0.0”. Example: PMC1:S:X0.0@1.5 The “PMC1:S:X0.0” part means that the symbol of the bit address from X0.0 is assigned consecutively. The “@1.5” part means that the top 10 characters are assigned.
Multiplier	This specifies the byte order of the string. Range: 1,-1

NOTE
Symbols and comments for Integrated PMC are read-only. They cannot be written to.

Example: [SETUP > Network Settings > HMI Device settings > Holding Registers] is set as follows.

	Start address	Number of addresses	Data string	Multiplier
Allocation setup 1	1	10	PMC1:S:R0.0@1.5	1
Allocation setup 2	11	10	PMC1:C:D0@1.5	1

The correspondence between Holding Registers and robot data using this setting is as follows:

Address	Assigned robot data
1-5	Symbol of R0.0
6-10	Symbol of R0.1

Address	Assigned robot data
11-15	Comment of D0
16-20	Comment of D1

55.3.13 Hints

Assigned data is not read correctly

If an HMI device etc. reads a Holding Register that is not assigned to any data, this Holding Register is always read as 0. If the Holding Register in which the problem occurs is not 0, this Holding Register is assigned to the other data.
For example, if [SETUP > Network Settings > HMI Device settings > Holding Registers] is set as follows, addresses 101-150 of Holding Registers are assigned by both Allocation setup 1 and Allocation setup 2.

	Start address	Number of addresses	Data string	Multiplier
Allocation setup 1	1	1000	R[1]@1.1	1
Allocation setup 2	101	50	PR[1]	100

In this case, the Holding Registers settings whose index is smaller are used.
Therefore, addresses 101-150 are assigned to R[101]-R[150], and PR[1] cannot be accessed by an HMI device etc.
If the read data of the Holding Register is 0, there may also be a duplicated assignment as above. Check for duplication in [SETUP > Network Settings > HMI Device settings > Holding Registers].
If Holding Registers settings does not have a duplicate assignment, check [SETUP > Network Settings > HMI Device settings > Holding Registers > Allocation setup n > Data string].
The following list shows the correct format of the Data string setting. If the setting does not match this format, robot data will not be assigned.

Correct format	Note
“R[n]”	
“PR[n]” “PR[Gn:n]	If you specify a group number, “:” must be included between the group number and index.
“SR[n]”	
“POS[n]” “POS[Gn:n]	If you specify a group number, “:” must be included between the group number and index.
“ALM[n]” “ALM[En]” “ALM[Pn]	Specify the alarm line number just after E, M, A, S or P. Do not include a ":".
“PRG[n]” “PRG[Mn]”, “PRG[Kn]” “PRG[MKn]”, “PRG[KMn]”	Specify the index just after M, K, MK or KM. Do not include a ":".
$[KAREL program name]variable name	Specify the KAREL program name just after “$[“, then include a “]” and specify KAREL variable name just after the “]”.
“DI[n]”, “DI[Sn]”, “DI[Cn]” “DO[n]”, “DO[Sn]”, “DO[Cn]” “RI[n]”, “RI[Sn]”, “RI[Cn]”, “RO[n]”, “RO[Sn]”, “RO[Cn]” “UI[n]”, “UI[Cn]” “UO[n]”, “UO[Cn]” “SI[n]”, “SI[Cn]” “SO[n]”, “SO[Cn]” “WI[n]”, “WI[Sn]”, “WI[Cn]”, “WO[n]”, “WO[Sn]”, “WO[Cn]” “WSI[n]”, “WSI[Sn]”, “WSI[Cn]” “WSO[n]”, “WSO[Sn]”, “WSO[Cn]” “GI[n]”, “GI[Sn]”, “GI[Cn]”	Specify the index just after S or C. Do not include a ":".

Correct format	Note
“GO[n]”, “GO[Sn]”, “GO[Cn]” “AI[n]”, “AI[Sn]”, “AI[Cn]” “AO[n]”, “AO[Sn]”, “AO[Cn]” “F[n]”, “F[Cn]” “M[n]”, “M[Cn]”	
PMC<path>:address PMC<path>:address.bit PMC<path>:S:address PMC<path>:S:address.bit PMC<path>:C:address PMC<path>:C:address.bit	The <path> is from 1 to 5 or "S". Include a ":" just after the PMC path. Also, specify the PMC address just after the ":". For symbols or comments, include ":S" or ":C" just after the PMC path.

- If there is space character in the “Data string”, data will not be assigned. If you specify “@”, do not add any space before the “@”.
- If you use a sequential array assignment, the number of Holding Registers for one element is defined according to the data type. Check the description of the assigned robot data type.
- The format of “@” is “@n.n”. A “.” must be included between the starting address and the number of Holding Registers. A “@” must be included just after the variable name.

Method to improve response time

- Position representation conversion takes time. When accessing position registers, assign only the elements of position structure actually needed using “@”. In addition, change the position representation of the position registers to the representation that is accessed by the HMI device etc.

55.4 DYNAMIC ASSIGNMENT TO HOLDING REGISTERS

Setting [SETUP > Network Settings > HMI Device settings > Holding Registers] is necessary to read or write data other than I/O signals. Normally, this setting is made on the Network Settings screen. However, dynamic settings, equivalent to “Holding Registers” settings, is available for each connection after a communication connection from a Modbus master or from an OPC UA client is established.
Note that this setting is cleared when the connection is closed.
Settings are enabled only in the same connection so read and write Holding Registers while maintaining the connection or set them for each connection. Many masters / clients disconnect after each write. If dynamic assignment does not work correctly, you must understand and control how the master/client operates for the level of Modbus or TCP/IP protocol.

This setting takes precedence over “Holding Registers” settings.

When 128 characters or less are written to a Holding Registers address of 16385 or later (you must start writing from address 16385) with Modbus from the master or Command node with OPC UA from the client, the robot regards the written characters as a command and processes it. You can change the byte order of the string data for Modbus the setting in [SETUP > Network Settings > HMI Device settings > Modbus connection > Byte swapping]. (This doesn’t affect to the byte order of Holding registers.)
Other than 0 (default) : The lower byte is the first character, the upper byte is the second character.
0 : The upper byte is the first character, the lower byte is the second character.

The following commands can be executed.
Each command must be written once. It cannot be written in installments.

CLRASG Clear and initialize all assignments
Format: CLRASG

Function: Clear and initialize all dynamic assignments in the connection (prepare for using dynamic assignment).

Execute this once before executing the following series of SETASG commands.

SETASG Set assignment
Format: SETASG （Start address） （Number of addresses） （Data string） ［（Multiplier）］
Function: Set the assignment.

The specification of the Start address, Number of addresses, Data string, and Multiplier are same as the specification of the same items in [SETUP > Network Settings > HMI Device settings > Holding Registers].
Multiplier can be abbreviated. If abbreviated, set Multiplier to 1.
Always run CLRASG once before executing the SETASG sequence.

Example: If assignments are as follows:

No.	Start address	Number of addresses	Data string	Multiplier
1	1	2	R[1]@1.1	1
2	3	4	R[1]	100
3	7	4	R[2]	0.1
4	11	2	R[1]	0

Write the following strings in order.

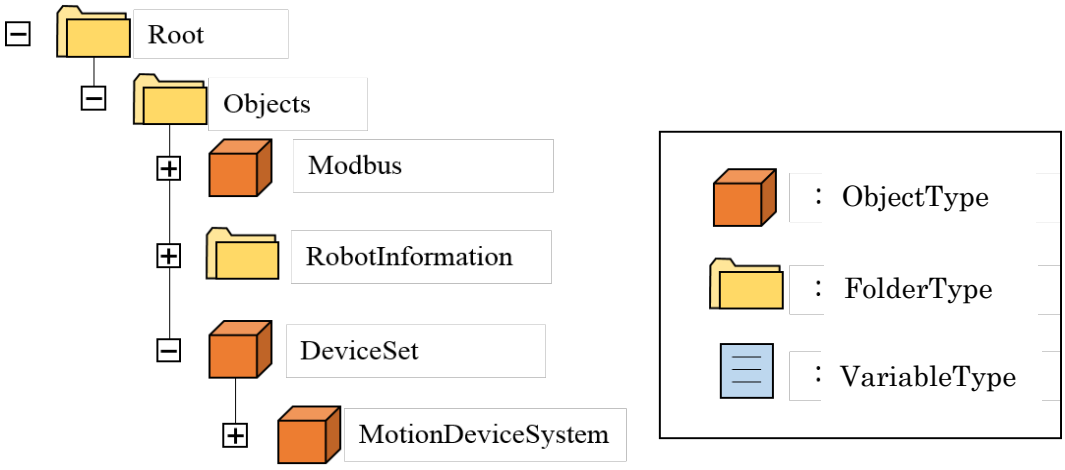
```
CLRASG
SETASG 1 2 R[1]@1.1 1
SETASG 3 4 R[1] 100
SETASG 7 4 R[2] 0.1
SETASG 11 2 R[1] 0
```

Multiple commands can be written at once by separating them with a return control character CR (13) or line feed LF (10). The maximum number of characters in one sequence is 128.

- There is no way to know what the current dynamic allocation is.
- Dynamic assignment allows assignment as long as memory allows.
- Even if allocation is not possible, no error is returned. If an address that cannot be assigned is read, 0 is read, and writing to it is ignored.

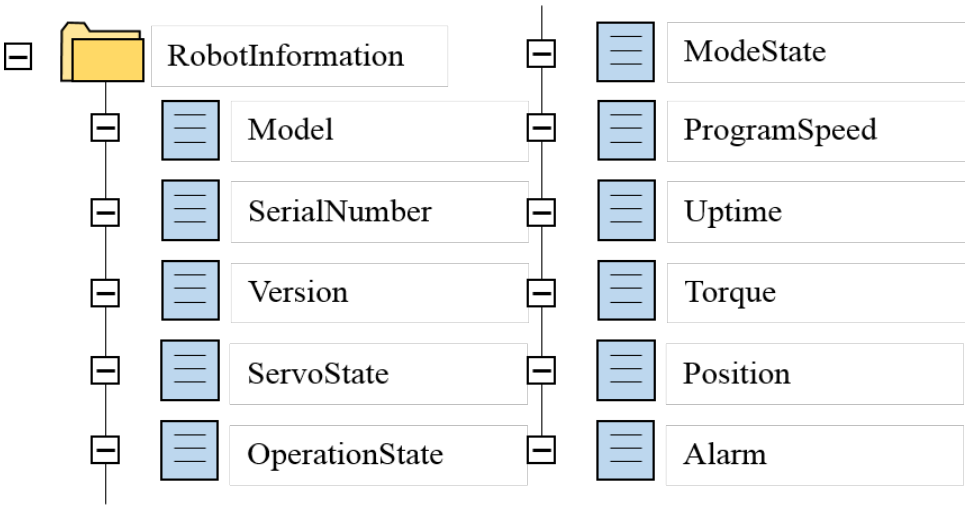
55.5 OPC UA DATA MODEL

OPC UA has ObjectType, FolderType and VariableType nodes.
ObjectType node is the pre-defined object, on the other hand, FolderType node just collects some nodes.
The OPC UA server for FANUC robots contains a Modbus node that provides read/write access to robot information using the Modbus data model (see 55.2 MODBUS DATA MODEL and 55.3 ASSIGNMENT OF HOLDING REGISTERS), a RobotInformation node that provides basic robot information, and a MotionDeviceSystem node of the OPC UA for Robotics specification. These nodes can be accessed by connecting from a OPC UA client.



55.5.1 RobotInformation

The RobotInformation node provides basic information such as the robot model name and the robot controller software version. It has the following nodes.



Model String type

The Model node provides the model name for the group 1 robot.
Example: R-2000/210F-27C-SA

SerialNumber String type

The SerialNumber node provides the robot controller serial number.

Version String type

The Version node provides the software version of the robot controller.
Example: 7DH1/04

ServoState Byte type (1 byte)

The ServoState node indicates servo on/off status.
The node value is set to 1 when the servo is on and 0 when the servo is off.

OperationState Uint16 type (2 byte)

The OperationState node provides status of E-STOP, Abort, PROGRUN, PAUSED, HELD, FAULT, TPENBL, BATALM, and BUSY.

The value of each state is set in bit units, converted to decimal, and expressed as a single node value.
E-STOP is set to 1 when either of the SOP E-STOP, TP E-STOP, and Ext E-STOP is input, and is set to 0 otherwise.
Abort is set to 1 when PROGRUN of the peripheral I / O (UO) is OFF and PAUSED is OFF.
PROGRUN to BUSY is the same as the UO of peripheral I / O.
The OperationState node is set in the following order starting from the lower bits.
E-STOP (1) | Abort (2) | PROGRUN (4) | PAUSED (8) | HELD (16) | FAULT (32) | TPENBL (64) | BATALM (128) | BUSY (256)

ModeState String type

The ModeState node indicates the operation mode of the robot.
The current operation mode is set to the node value from among Auto, T1 (Manual), and T2 (Manual).

ProgramSpeed Byte type (1 byte)

The ProgramSpeed node provides the speed override of the robot.
A value of 0 to 100 (%) is set to the node value.

Uptime Int32 type (4 byte)

The Uptime node indicates the total energization time of the robot controller.
The value of the total energization time (min) is set to the node value.

Torque Float type (4 byte)

The Torque node provides the Q-phase current value for each axis of the robot (up to the sixth axis) of group 1 and independent axis for group 1.
The Q-phase current value (A) of each axis is set to the node value.

Position Float type (4 byte)

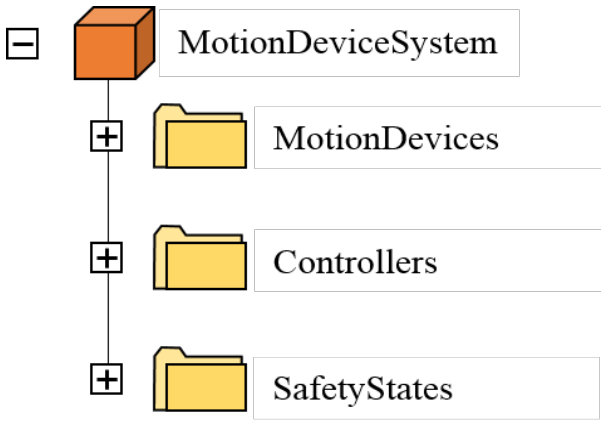
The Position node provides the position of each axis for the robot (up to the sixth axis) of group 1 and independent axis for group 1.

Alarm String type

The Alarm node provides the current alarm.
The current alarm string is set to the node value.
Example: SRVO-001 Operator panel E-stop
Display with UTF-8 when Japanese.

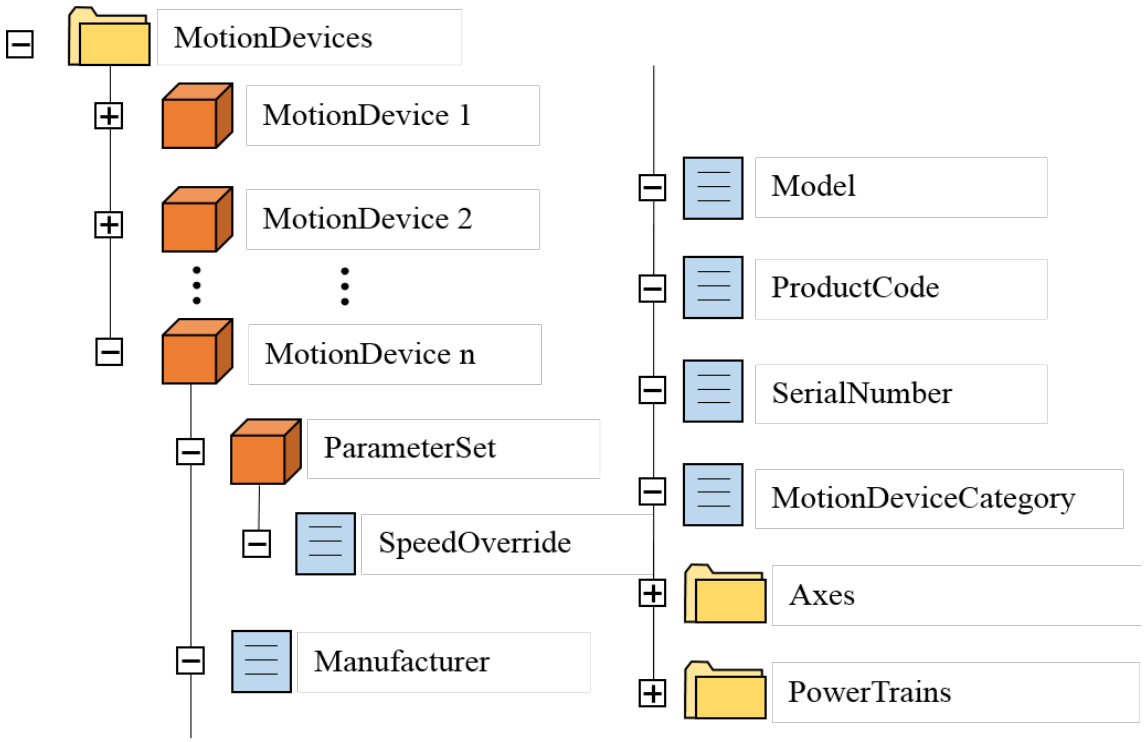
55.5.2 MotionDeviceSystem

The MotionDeviceSystem node is an OPC UA for Robotics compliant node.
It three nodes: MotionDevices, Controllers, and SafetyStatus, as shown below.



55.5.2.1 MotionDevices

The MotionDevices node provides information such as the robot model, overrides, axes, and motors.
The MotionDevices node has as many MotionDevice nodes as the number of robot groups.



SpeedOverride **Double type (8 byte)**

The SpeedOverride node provides the robot speed override.
A value of from 1 to 100 (%) is set to the node value.

Manufacturer **LocalizedText type**

The Manufacturer node provides the name of the robot manufacturer.
"FANUC" is set to the node value.

Model **LocalizedText type**

The Model node provides the robot model name.
Example: R-2000/210F-27C-SA

ProductCode **String type**

The ProductCode node provides the product code of the robot.
Since there is no corresponding data, null is set to the node value.

SerialNumber **String type**

The SerialNumber node provides the serial number of the robot.
Since there is no corresponding data, null is set to the node value.

MotionDeviceCategory **MotionDeviceCategoryEnumeration type**

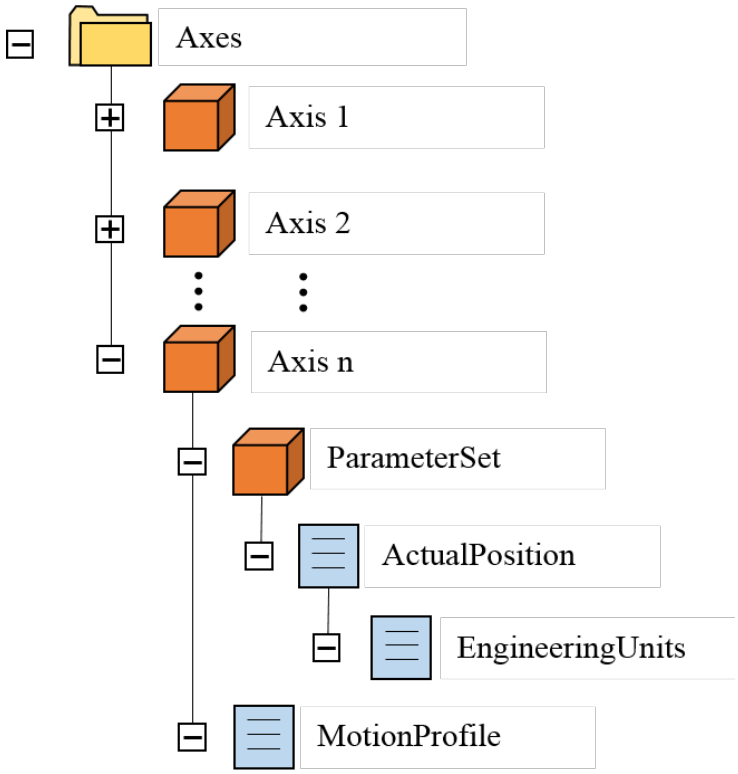
The MotionDeviceCategory node provides the category of the robot.
The category value corresponding to the following table is set to the node value.

Category	Value	Description
OTHER	0	Any motion device that does not fit into any of the other categories.
ARTICULATED_ROBOT	1	This robot design has from 2 to 10 or more rotary joints and can range from simple two joint structures to 10 or more joints. The arm is connected to the base with a twisting joint. The links in the arm are connected by rotary joints.
SCARA_ROBOT	2	The robot has two parallel rotary joints to provide on a selected plane.
CARTESIAN_ROBOT	3	Cartesian robots have three linear joints that use the Cartesian coordinate system (X, Y, and Z). They also may have an attached wrist to allow for rotational movement. The three linear joints move the robot in a linear motion along an axis.
SPHERICAL_ROBOT	4	The arm is connected to the base with a twisting joint and a combination of two rotary joints and one linear joint. The axes form a polar coordinate system and create a spherical-shaped work envelope.
PARALLEL_ROBOT	5	These spider-like robots are built from jointed parallelograms connected to a common base. The parallelograms move a single end-of-arm tooling in a dome-shaped work area.
CYLINDRICAL_ROBOT	6	The robot has at least one rotary joint at the base and at least one linear joint to connect the links. The rotary joint uses rotational motion along the joint axis, while the linear joint moves with linear motion. Cylindrical robots operate within a cylindrical-shaped work envelope.

Axes

The Axes node provides axis information.

The Axes node has as many Axis nodes as the number of robot axes. It supports up to nine axes for extended axes and independent axes.



ActualPosition Double type (8 byte)

The ActualPosition node provides the current position of the axis.

EngineeringUnits EUInformation type

The EngineeringUnits node provides the units for the ActualPosition node.
The units of ActualPosition are “deg.” for a rotary axis and “mm” for a linear axis.

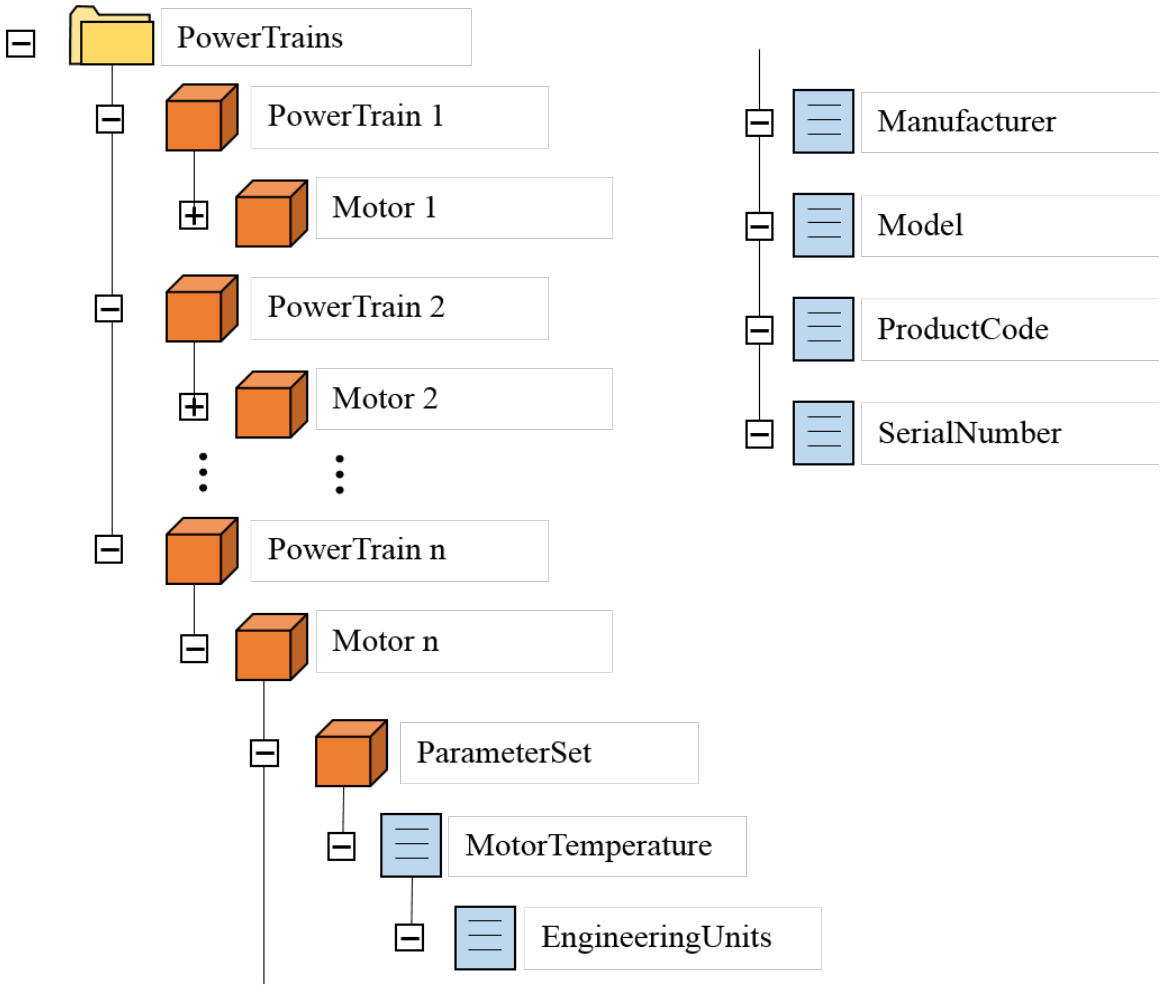
MotionProfile MotionProfileEnumeration type

The MotionProfile node indicates whether the axis is linear or rotary, and whether it rotates continuously.
The value corresponding to the following table is set to the node value.

Category	Value	Description
OTHER	0	Axes that do not fit into any category.
ROTARY	1	The axis is rotary and does not rotate continuously.
ROTARY_ENDLESS	2	The axis is rotary and rotates continuously.
LINEAR	3	The axis is linear and does not rotate continuously.
LINEAR_ENDLESS	4	The axis is linear and rotates continuously.

PowerTrains

The PowerTrains node provides motor information.
The PowerTrains node has as many PowerTrain nodes as the number of robot axes and as many Motor nodes as the number of motors. It supports up to the ninth axes for extended axes and independent axes.



MotorTemperature Double type (8 byte)

The MotorTemperature node provides the motor temperature.
Since there is no corresponding data, null is set to the node value.

EngineeringUnit EUInformation type

The EngineeringUnit node provides the units for the MotorTemperature node.
Since there is no corresponding data, null is set to the node value.

Manufacturer LocalizedText type

The Manufacturer node provides the name of the motor manufacturer.
"FANUC" is set to the node value.

Model LocalizedText type

The Model node provides the motor model name.
Example: aiSR22/4000 80A

ProductCode String type

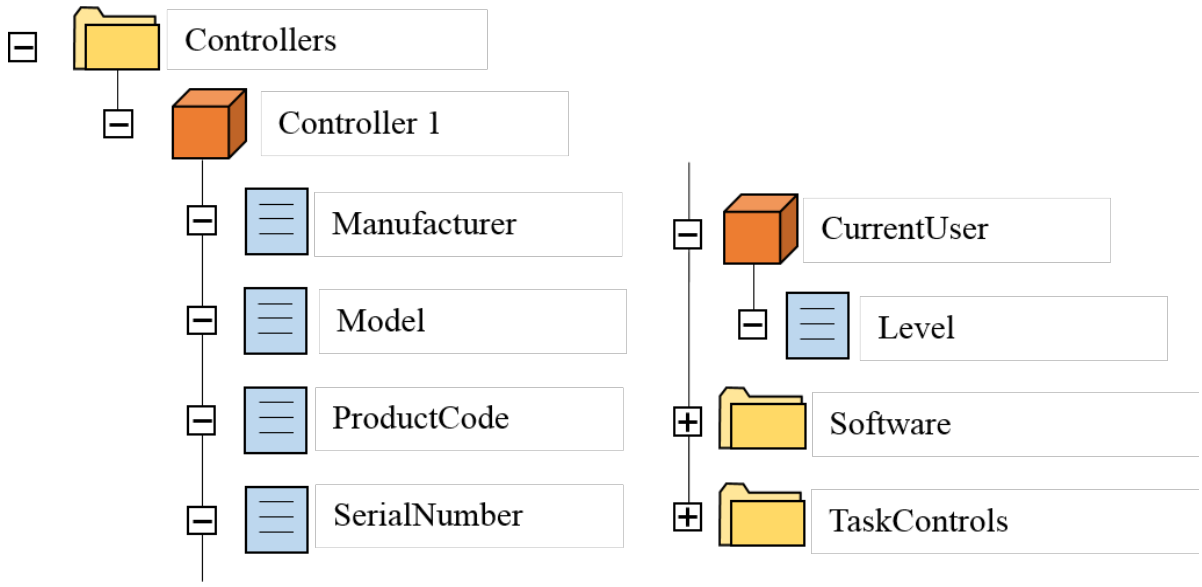
The ProductCode node provides the product code of the motor.
Since there is no corresponding data, null is set to the node value.

SerialNumber String type

The SerialNumber node provides the serial number of the motor.
Since there is no corresponding data, null is set to the node value.

55.5.2.2 Controllers

The Controllers node provides information on the model, software, and tasks of the robot controller.
It has a Controller 1 node. Since the robot controller is an OPCUA server, there is only one Controller node.



Manufacturer LocalizedText type

The Manufacturer node provides the name of the robot controller manufacturer.
"FANUC" is set to the node value.

Model LocalizedText type

The Model node provides the robot controller model name.
Example: R-50*i*A

ProductCode String type

The ProductCode node provides the product code of the robot controller.

SerialNumber String type

The SerialNumber node provides the serial number of the robot controller.

Level String type

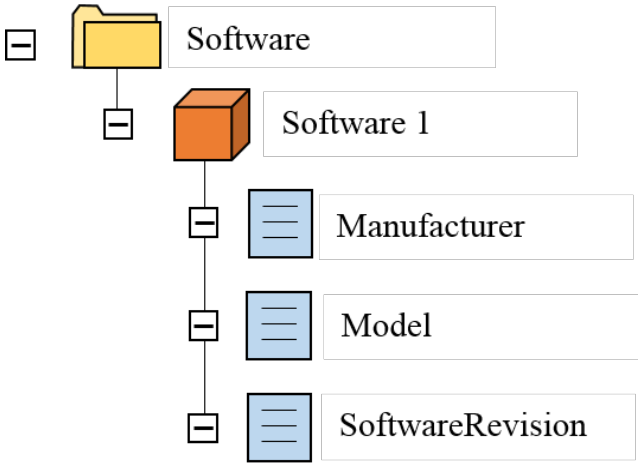
The Level node provides the level of the logged-in user.
The user level corresponding to the following table is set to the node value.

Value	User Level
INSTALL	Install
PWDMGR	PWDMGR
SETUP	Setup
PROGRAM	Program
OPERATOR	Operator

For information on the operation authority of each level, refer to Section 8.11 "Password Function" in the "*Operator's Manual (Basic Function)*" (B-84464EN).

Software

The Software node provides information on the software installed on the robot controller.
The Software node has the Software1 node.



Manufacturer **LocalizedText type**

The Manufacturer node provides the name of the software manufacturer.
"FANUC" is set to the node value.

Model **LocalizedText type**

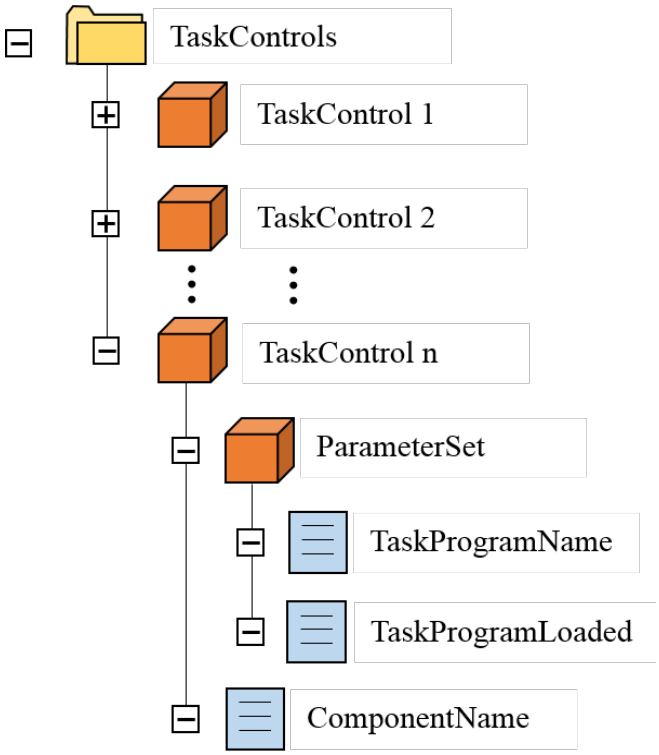
The Model node provides the software model name.
Example: R-50*i*A

SoftwareRevision **String type**

The SoftwareRevision node provides the version of the software.
Example: 7DF1/19 (V9.10P/19)

TaskControls

The TaskControls node provides information on user tasks.
The TaskControls node has as many TaskControl nodes as the number of tasks set in the maximum number setting.



TaskProgramName **String type**

The TaskProgramName node provides the name of the user program that is running or has finished running.
If there is no corresponding program, null is set.

TaskProgramLoaded **Boolean type (1 byte)**

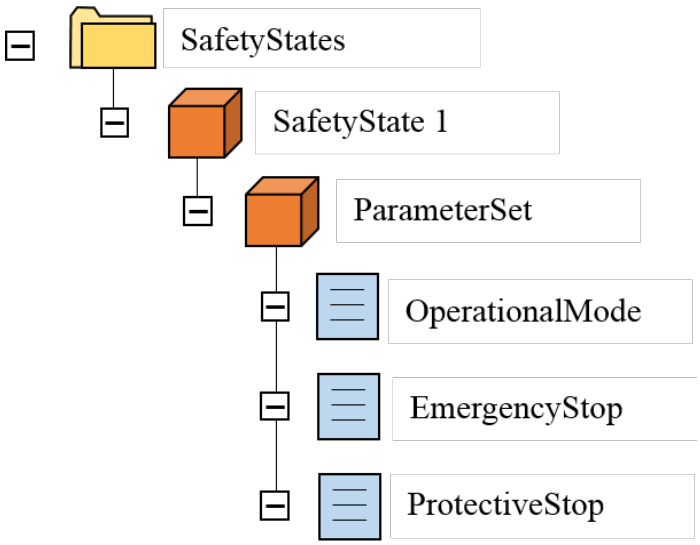
The TaskProgramLoaded node provides the loading status of the user program.
This node is set to 1 if the user program has been loaded and 0 if not.

ComponentName **LocalizedText type**

The ComponentName node provides the name of the user rewritable TaskControl.
The name of the user program that is running or has finished running is set by default.
The set ComponentName is reset when the power is turned off.

55.5.2.3 SafetyStates

The SafetyStates node provides information on operating modes and emergency stop signals.
The SafetyStates node has the SafetyState 1 node.



OperationalMode OperationalMode type

The OperationalMode node provides the operation mode of the robot.
The operation mode corresponding to the following table is set to the node value.

Operation mode	Value
OTHER	0
T1	1
T2	2
AUTO (Local)	3
AUTO (Remote)	4

EmergencyStop Boolean type (1 byte)

The EmergencyStop node provides the status of the emergency stop signal.
This node is set to 1 when either the SOP E-STOP, TP E-STOP, or Ext E-STOP signals is input, and is set to 0 otherwise.

ProtectiveStop Boolean type (1 byte)

The ProtectiveStop node provides the status of the stop signal.
This node is set to 1 when either the Fence Open, TP Deadman, TP Enable, Hand Broken, Overtravel, Low Air Alarm, Belt Broken, SVOFF Input, and Non Teacher Enb.Dev. signals are input, and is set to 0 otherwise.

56 FAULT & INCIDENT REPORT

The Fault & Incident Report function analyzes past alarms. The top five alarms in terms of number of alarms, total duration of alarms, and longest duration of alarms can also be displayed in a graph.

This function is optional. To use this function, the Fault & Incident Report option (R544) is required.

56.1 ALARMS REPORTING SCREEN

The Alarms Reporting screen allows you to check the number of incidents, total time, and longest time for each of the top five alarms that occurred in AUTO mode.
You display the Alarms Reporting screen by selecting MENU ->Alarm->F1, [TYPE]->Reporting.

Alarms with alarm severity of NONE or WARN are not displayed on this screen.

56.1.1 Incident screen

The top five alarms by incident are displayed on this screen. The top five alarms with the highest number of alarms in the past are displayed in order of the number occurred. Each value and bar graph displays the number of alarms that have occurred in the past. This number is counted when the target alarm occurs in AUTO mode. However, alarms “SYST-038 Operation mode T1 Selected” and “SYST-039 Operation mode T2 Selected” are counted in both T1 mode and T2 mode.

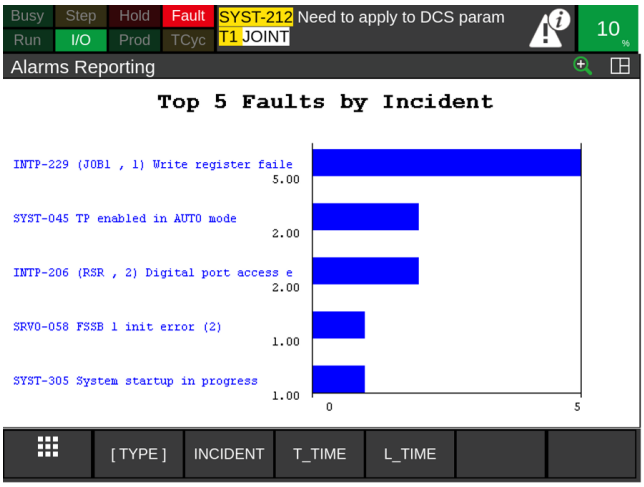


Fig. 56.1(a) Alarms Reporting INCIDENT Screen

56.1.2 Total Time (T_TIME) screen

The top five alarms by total time are displayed on this screen. Alarms are listed in order of the total time they occurred in the past. Each value and bar graph indicate the total time (units are 1/10 minute). The incident time is the time after the target alarm occurs in AUTO mode before you press the fault reset key. The total time is the sum of the incident times. However, alarms “SYST-038 Operation mode T1 Selected” and “SYST-039 Operation mode T2 Selected” are recorded in both T1 mode and T2 mode.

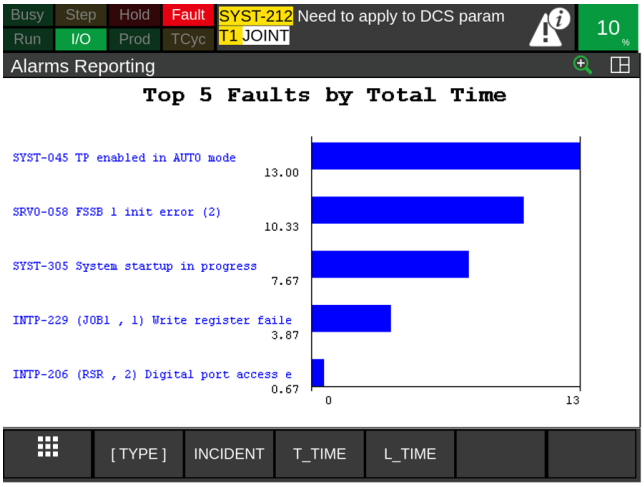


Fig. 56.1(b) Alarms Reporting Total Time Screen

56.1.3 Longest Time (L_TIME) screen

The top five alarms with the greatest total time are displayed on this. Alarms are listed in order of the longest duration. Each value and bar graph represents the longest alarm duration (units are 1/10 minute). The incident time is the time an alarm occurs in AUTO mode before you press the fault reset key. The longest alarm occurrence time is the longest alarm occurrence time among the alarm occurrence times. However, alarms “SYST-038 Operation mode T1 Selected” and “SYST-039 Operation mode T2 Selected” are recorded in both T1 mode and T2 mode.

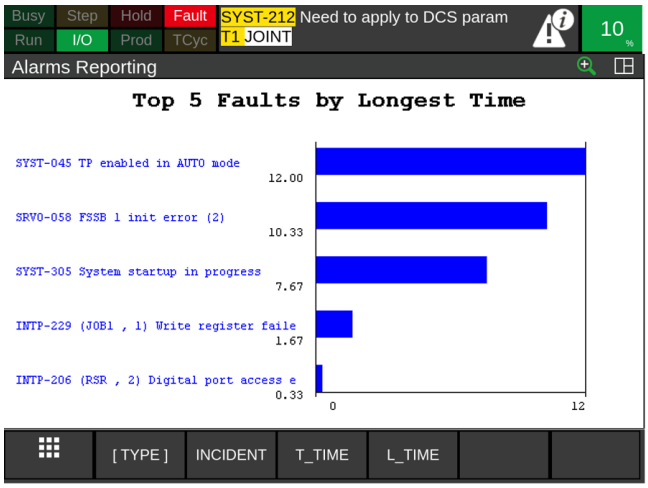


Fig. 56.1(c) Alarms Reporting Longest Time Screen

57 PROCESS AXES

57.1 BASIC PROCESS AXES (R689)

The Basic Process Axes software option (R689) provides the user with the capability of configuring and operating a motor or motors independent of any motion group. A “Process Axis” has the following characteristics:

- The axis accepts only speed commands; its position cannot be commanded directly
- The axis does not belong to any motion group

Process axes are intended to be used to drive devices for which position control is not needed, e.g. a pump or a drill bit.

Process axes are also used by other software options, which integrate the operation of the process axes directly into the application tool itself. Since these options access the process axes directly, they cannot be loaded on the controller with the Basic Process Axes option. That is, Basic Process Axes is mutually exclusive with these options.

The software options that are mutually exclusive with Basic Process Axes are as follows:

- Integral Pump Control (R554)
- FRA Waterborne (R556)
- ISD Gear Meter (R563)
- ISD Dual Meter (R562)
- ISD Single Acting (R564)
- ISD Double Acting (R561)
- Pneumatic Servo (R595)
- Servo Torch (J982)
- Pump Only Motion (J801)

Accordingly, the Basic Process Axes option should be loaded and used in simple, generic cases when the use of a process axis is desired but it does not fit one of the above options. With this option, the process axes can be controlled by TP instructions or by group inputs.

In the TP instructions, the speed command uses a unit called user speed. Please refer to the section 57.1.3 TP Instructions for user speed units.

The Basic Process Axes option provides the following functions:

- The ability to setup and configure process axes, up to a maximum of 8.
- TP instructions to send a speed command to the axes, and to stop the axes.
- A torque monitoring function to post an alarm and stop program execution and robot motion when the process axes encounter excessive torque.
- A setup menu to specify a scale factor for speed commands, torque limit values, group outputs for speed data, and group inputs for speed commands.
- A status screen to display speed and torque data for the process axes.

We also use a name "ISDT" to represent Process Axis. ISDT stands for Integral Servo-Driven Tool.

57.1.1 Setup

Add a process axis

Before using the process axes, you need to add a process axis at the Robot Maintenance screen in the Controlled Start menu. Please refer to the B.12 PROCESS AXES SETUP in chapter B. SPECIAL OPERATION of the "OPERATOR'S MANUAL (Basic Function)" (B-84464EN).

Process axes setup menu

The Basic Process Axes option has a setup menu available to configure its features. The setup menu covers the following items:

- Torque Monitoring function
- Scale factor for the speed value in the TP instruction
- Group Input for speed command
- Group Output for speed display

Refer to Table 57.1.1 (a) and (b) for further details on the items included in the setup menu. Use Procedure 57-1 to setup the Basic Process Axes option.

Table 57.1.1 (a) Process Axes: Axis setup menu items

ITEM	DESCRIPTION
Process Axis Number default: 1	This item indicates which process axis corresponds to the data being displayed in the menu. This field cannot be edited directly, but is changed by pressing F3 [AXIS] and entering the new axis number.
Current Limit default: 0.0	This item defines a maximum value for the threshold of the commanded current of the process axis motor, to be used by the torque limit function. A value of 0.0 indicates that the torque limiting function is disabled. See the section on torque limiting for more details
Cur Lim fault after default: 0	This item is also part of the torque limiting function and defines the number of milliseconds for which the current value indicated in the "Current Limit" item must persist before an alarm is posted.
Scale factor for TP Inst default: 1.0	This item is used in conjunction with the next item (gear ratio) to determine the relationship between the speed indicated in the SET ISDT SPEED instruction and the RPM at the motor shaft.
Current gear ratio is default: (none)	This item is included for display purposes only and must be set from within the Process Axis Setup routine at controlled start.
Speed Command Group Input default: 0	This item defines a group input number that can be used to set the speed command for this axis. Note that cycling power is required for changes in this item to take effect.
Actual Speed Group Output default: 0	This item defines a group output to display the current motor RPM for this process axis. Note that cycling power is required for changes in this item to take effect.

Table 57.1.1 (b) Process Axes: Common setup menu items

ITEM	DESCRIPTION
Change STOP severity to WARN default: YES	If this item is YES, change ISDT alarms with STOP severity to WARN severity. (Data ID: $ISDT_SETUP. CHANGE_STOP_SEVERITY_TO_WARN / BOOLEAN / RW)

Stop all process axes at HOLD default: NO	If this item is YES, all process axes will stop by a HOLD signal input. If this item is NO, process axes will not stop. Note: • A HOLD signal input to any motion group will also work as a HOLD signal input to process axes. • Process axes controlled by group I/O are not affected by this setting. (They do not stop by a HOLD signal input.) • Process axes controlled by the application software may not be affected by this setting. (Data ID: $ISDT_SETUP. STOP_ALL_PROCESS_AXES_AT_HOLD / BOOLEAN / RW)

Procedure 57-1 Setting up Basic Process Axes

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

```
SETUP Process Axes: AXIS
                                        1/5
     Process Axis Number:  1

     Current Limit:             0.0 amps
     Curr Lim fault after         0 msec
     Scale factor for TP Inst:    1.0
     Note:  Scale factor is applied in
     addition to gear ratio!!
     (Current gear ratio is  10:1 )
     Speed Command Group Input: 0
     Actual Speed Group Output: 0

    [ TYPE ]  COMMON  AXIS        HELP
```

5. Move the cursor to the item you wish to change, and enter the new value for that item.
6. **To change the process axis number being setup**, press F3, AXIS, and enter the desired process axis number. If the number entered does not correspond to a valid, properly setup process axis, the following message will be displayed:

```
Invalid Axis Number!
```

7. **To move to the common setup menu**, press F2, COMMON, You will see a screen similar to the following

```
SETUP Process Axes: COMMON
                                        1/1
  Change STOP severity to WARN:   YES
  Stop all process axes at HOLD:  NO

    [ TYPE ]  COMMON  AXIS        HELP
```

8. **To display help information**, press F5, HELP. When you are finished displaying help, press PREV.

Control the process axes with group I/O

If a nonzero value is set for “Speed Command Group Input” in this setup menu, then the resulting behavior of that process axis will be as follows:

- The group input will be monitored continuously, so that the value of the input will be the instantaneous speed command to the process axis motor.
- The value stored in the group input will be treated as a 16–bit integer, converted to RPM at the motor shaft as follows:
  - The most significant bit will be treated as the sign bit, where 0 means positive and 1 means negative. (i.e. for a negative speed value add 32768).
  - The value designated by the remaining 15 bits will be divided by 10 to obtain the RPM to be commanded to the motor.
- The process axis that uses this group input number will be in motion as long as the following conditions are met:
  - The axis' servo amplifier is enabled
  - The axis' brake is released
  - The group input has a nonzero value

For example, if a value of “11” were entered for “Speed Command Group Input” for process axis #1 in the setup menu, then in order to drive the process axis #1 motor at 1000 RPM (at the motor shaft), the group input #11 must be set to a value of 10000. If instead a speed of —1000 RPM were desired, then group input #11 should have the value 42768, which is 10000 + 32768

Note
If a group input is setup to be used as the speed command for any process axes, then the TP instructions are automatically disabled. That is, TP instructions and Group Inputs cannot be simultaneously used to control the process axes.

If a nonzero value is set for “Actual Speed Group Output” in the setup menu, then the actual RPM at the motor shaft of the process axis will be copied in real time to that group output, as a 16–bit integer value, with the most significant bit representing the sign bit. As with the speed command group input, when the most significant bit is 1, then the group output is indicating a negative speed value.

For example, if a value of “11” were entered for “Actual Speed Group Output” for process axis #1 in the setup menu, then if group output 11 displays a value of 1000, it means that process axis #1 has a speed of 1000 RPM at the motor shaft. If group output 11 displays 33768, then the speed of process axis #1 is —1000 at the motor shaft.

57.1.2 TP Instructions

The Basic Process Axes option includes two teach pendant instructions for setting process axes speeds:

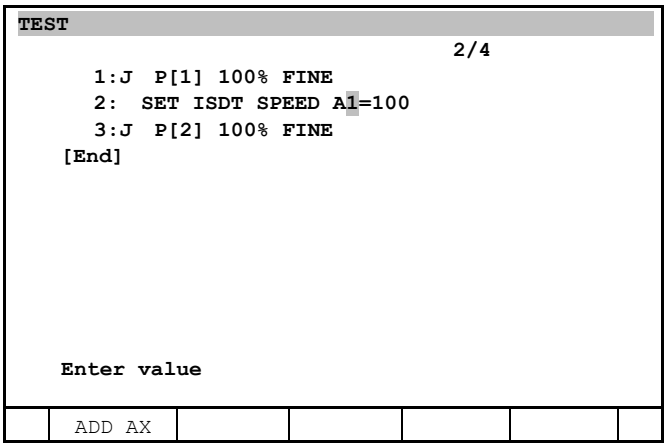
- SET ISDT SPEED
- STOP ALL ISDT

SET ISDT SPEED Instruction

The SET ISDT SPEED instruction is used to send speed commands directly to the process axes and can include speed data for up to 8 axes. The instruction allows two fields for each axis, one for axis number and one for the speed value.
The relationship between speed (user speed units) and motor RPM is below

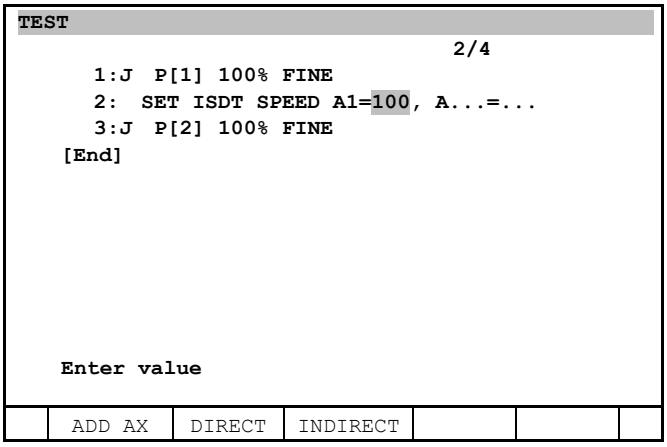
$$speed(user\ units) = \frac{motor\ RPM}{gear\ ratio} \times scale\ factor$$

The following screen is an example of the usage of this instruction:



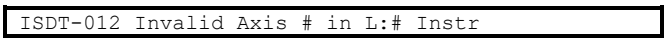
To the right of this field, following the "=" is the speed value that will be sent to process axis #1, which in this example is 100 user speed units.

If this instruction needs to set speeds for additional process axis, pressing F1 (labeled "ADD AX") will add an additional set of fields to the instruction as shown in the following screen:



The cursor can then be moved to the "..." fields to enter a new axis number and speed value. Note also in the above screen that the cursor is on the speed field, which means that function keys F2 and F3 are available to allow the speed value to be specified as a direct integer value or indirectly via numeric register.

There is no error checking performed on the fields of the SET ISDT SPEED instruction at the time the instruction is added to, or edited in, a TP program. However, error checking is performed at the time the instruction is actually executed. For each field in the instruction, first the axis number will be checked to verify that it is a properly setup process axis. If that process axis has not been setup, then the following alarm will be posted:



The error text will indicate the process axis number in the instruction that is invalid, as well as the line number in the TP program containing the offending instruction. If the same process axis number is used in multiple fields in the instruction, then the following alarm will be posted:

```
ISDT-013 Duplicate Ax:# in L:# Instr
```

The alarm text will indicate which axis number was repeated, and also the line number from the TP program. If the speed value specified exceeds the speed limit of that process axis (as specified in the process axis setup routine at controlled start), then the following alarm will be posted:

```
ISDT-001 Joint speed limit (G:0 A:#)
```

If the speed value specified results in a motor RPM that exceeds the maximum RPM for this motor, then the following alarm is posted:

```
ISDT-002 Motor speed limit (G:0 A:#)
```

Finally, if there is a process axis that has been properly setup at controlled start but that does not have a corresponding field in the SET ISDT SPEED instruction, then when the instruction is executed that process axis will remain unaffected: it will continue moving at its current speed, or remain stopped if it was stopped.

STOP ALL ISDT Instruction

The STOP ALL ISDT instruction is used to immediately send a zero speed to all properly setup process axis. It does not have any arguments. An example of the usage of this instruction appears in the screen shot below:

```
TEST
                                        10/11
      1:J  P[1] 100% FINE
      2:  SET ISDT SPEED A1=100, A2=200
      3:J  P[2] 100% FINE
      4:  SET ISDT SPEED A2=0
      5:J  P[3] 100% FINE
      6:  SET ISDT SPEED A1=50
      7:J  P[4] 100% FINE
      8:  SET ISDT SPEED A1=100, A2=50
      9:J  P[5] 100% FINE
     10:  STOP ALL ISDT
   [End]
```

In this example program, once the robot has moved to P[1], process axis #1 will begin moving at speed 100 while process axis #2 will begin moving at speed 200. Then the robot will move to P[2], after which process axis #2 will stop (i.e. speed will change to 0), but process axis #1 will continue moving at speed 100. Next, after the robot moves to P[3], process axis #1 will change its speed from 100 to 50, while process axis #2 will remain stopped. Then the robot will move to P[4], after which both process axes will change speed, process axis #1 from 50 to 100 and process axis #2 from 0 to 50. Then the robot will move to P[5], and finally on the last line of the program both process axis #1 and #2 will stop.

57.1.3 PROCESS AXES STATUS

The Process Axes Status screen displays speed and torque data for all process axes defined on the controller. This screen is only available when the Basic Process Axes option is loaded on the controller.

See Fig. 57.1.3 (a) for an example screen and Table 57.1.3 (a) for a description of the items on this screen.

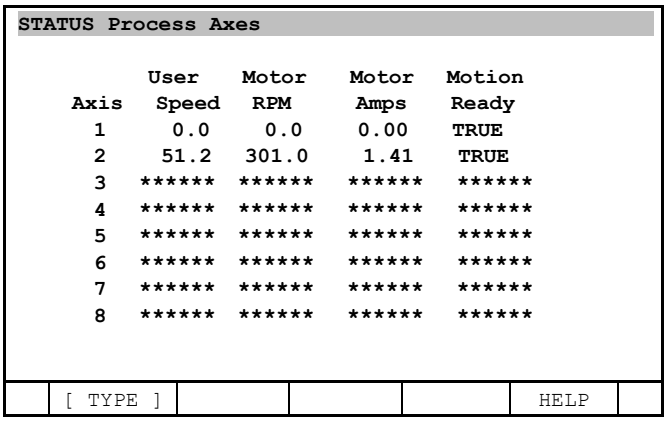


Fig. 57.1.3 (a) Process Axes Status Screen

Table 57.1.3 (a) Process Axes Status Items

ITEM	DESCRIPTION
Axis	This item is the axis number whose data is displayed in this row.
User Speed	This item is the current speed of the axis in user units (i.e. units of the TP instruction).
Motor RPM	This item is the current speed of the axis in RPM at the motor shaft.
Motor Amps	This item is the commanded current in amps from the servo amplifier for this axis.
Motion Ready	This item is the motion ready status of the axis. (Data ID: $ISDT_STATUS.MOTION_READY[a] / BOOLEAN / RO / a: axis number)

* "******" indicates that the axis is not defined.

57.1.4 Torque Monitor Function

The Basic Process Axes option includes a torque monitor function, which allows the user to specify a level of torque (more specifically, of commanded motor current) and a corresponding length of time beyond which a stop error will be posted. This is useful for protecting equipment connected to the process axis in such a case as the torque capacity of the motor exceeds that of the hardware being driven.

The torque monitor function is setup and enabled by the first two items in the process axis setup screen (see section for further details). These items are the current limit and the fault time. The current limit is specified in amps and represents an absolute value of the commanded current from the servo amplifier. The fault time is specified in milliseconds and represents the amount of time for which the absolute value of the commanded motor current must exceed the specified current limit for the alarm to be posted.

When the commanded motor current has exceed the "Current limit" value from the Process Axis Setup menu for at least the number of milliseconds defined by "Curr Lim fault after" in the setup menu, then the following alarm will be posted:

```
ISDT-014 Torque Limit on Ax:#
```

This alarm will stop program execution and robot motion, in addition to stopping the process axes.

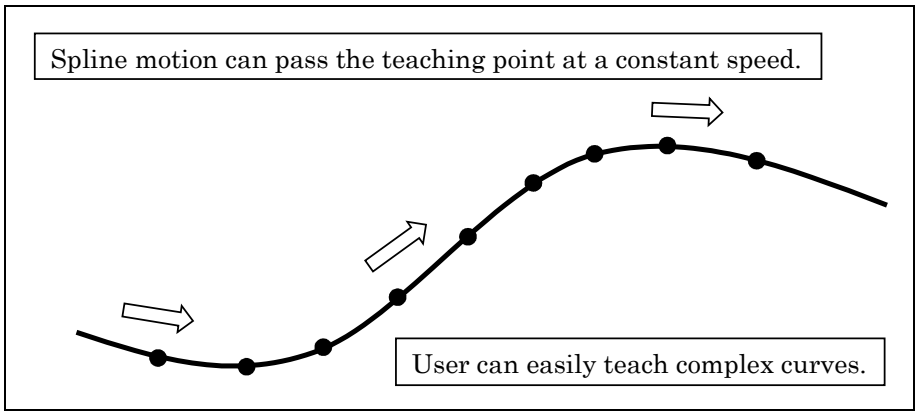
If the commanded motor current exceeds the current limit for only a brief period of time, specifically a period of time that is less than “Curr Lim fault after” in the setup menu, and then returns to a value below the limit, then no alarm will occur.

If a value of zero is specified under “Curr Lim fault after” in the setup menu, then the alarm will be posted immediately when the commanded current exceeds the limit.

If a value of zero is specified for the current limit in the setup menu, then the torque monitor function is disabled and motor current will not be checked.

58 SPLINE MOTION INSTRUCTION

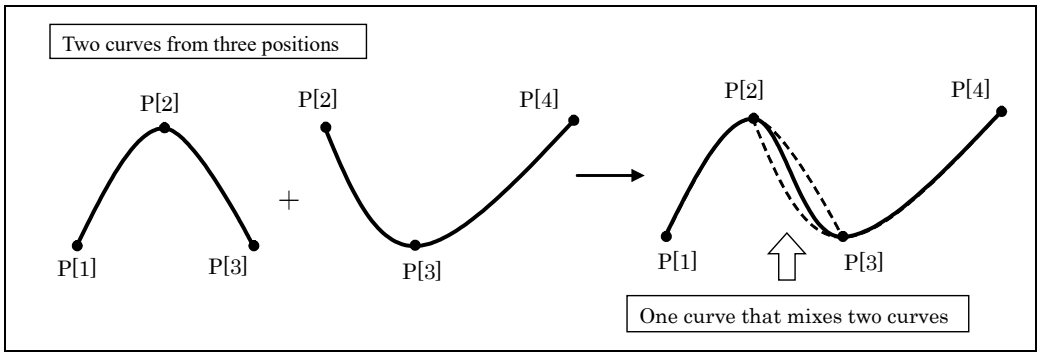
Spline (motion type S) motion moves on a curve that passes through many positions. Only one position is taught per program instruction, and the controller calculates a curve through all positions from successive spline motion instructions. If the teaching speed is slow, spline motion moves at a constant speed.
The Spline Motion (R904) is required to use the Spline Motion Instruction.



NOTE
If the distance between taught points is extremely short, or if the robot path curves sharply, the robot will automatically decelerate because it cannot pass through the path at the taught speed. The alarm MOTN-632 SPL:Speed Reduced (G: , L:) will be posted if the robot cannot pass through the path at the taught speed. Also, if the robot slows down but cannot pass through the path, MOTN-633 SPL:VAJ Limit (G: , L:) will be displayed.

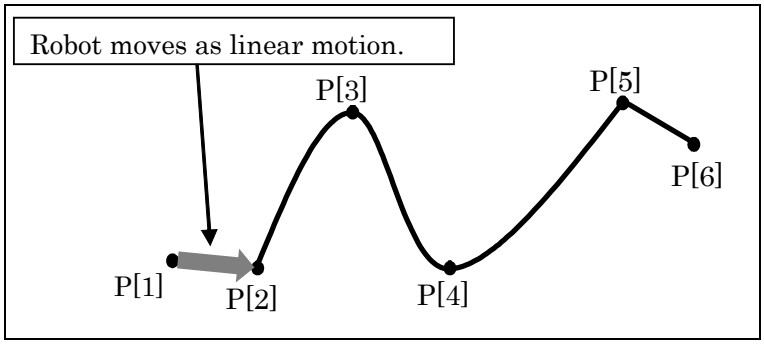
58.1 NORMAL MOTION

Spline motion with three lines draws a curve that smoothly passes through the three positions. Spline motion is more than four lines draws a curve that by combining the two curves between the two positions.



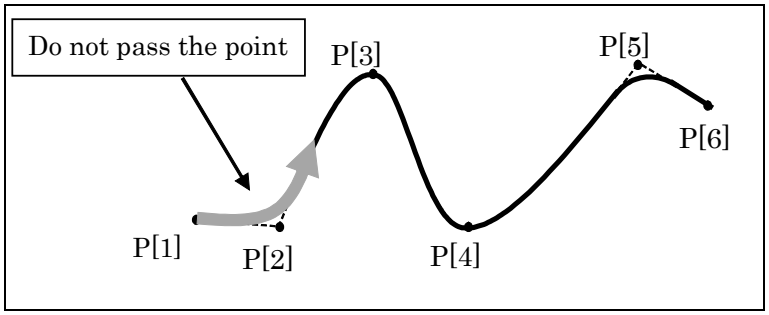
Teach the starting point of the curve using spline motion. The first spline motion instruction moves in the same way as linear (L) motion. If L motion to the starting position is not required, teach the first instruction of the spline motion at the same position as the previous motion instruction.

First spline motion instruction



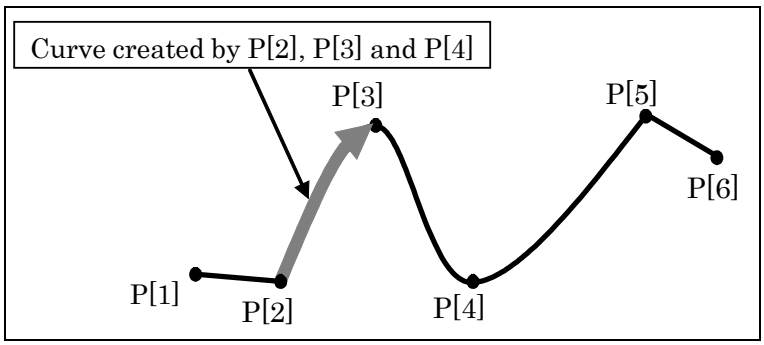
If you specify CNT in the first S motion instruction, the connection will be as smooth as L motion. At this time, near the start point, L motion and spline motion are blended by CNT, so they do not pass through the taught point.

CNT from linear motion to spline motion



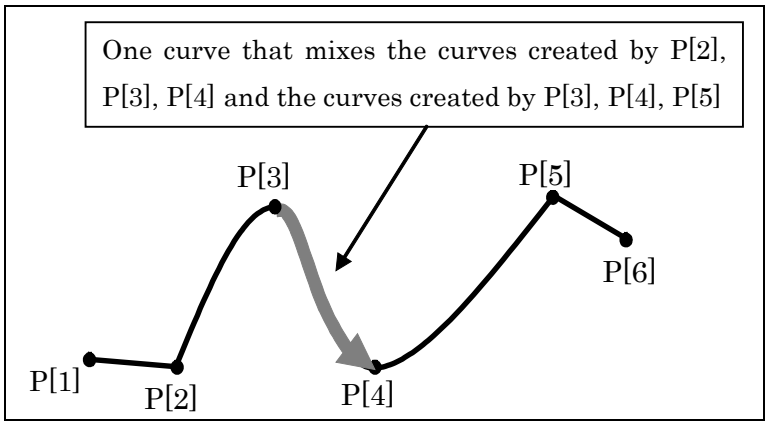
Execution of the second spline motion instruction moves on the curve created by the three points: current position, destination position of this spline motion instruction, the destination position of the next spline motion instruction.

Second spline motion instruction



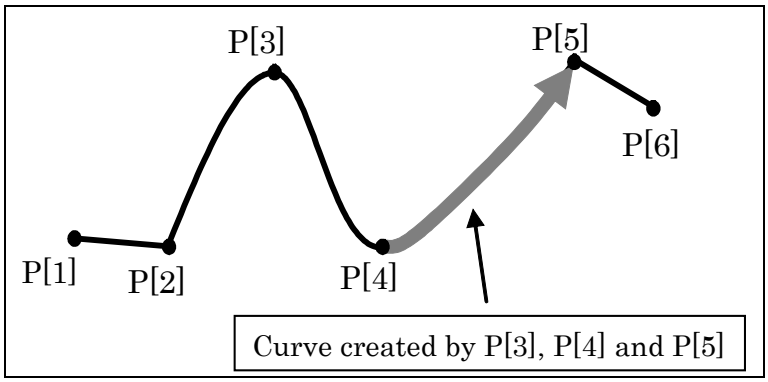
The third and subsequent spline motion instructions move on a curve calculated by the four positions.

Third and subsequent spline motion instructions



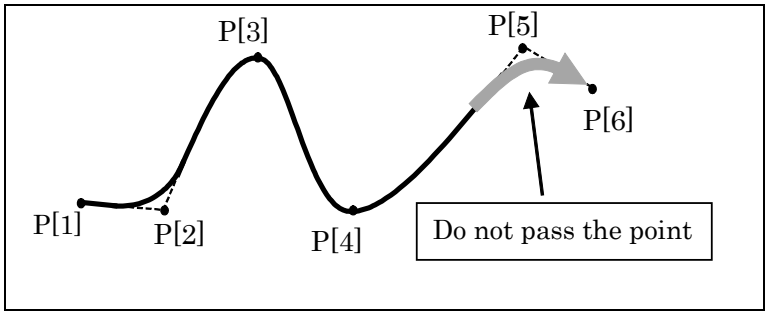
If the next motion instruction is not a spline motion instruction, the last spline motion instruction is considered to be the end position of the curve. In this case, the robot moves on a curve that is calculated by the three positions: start position of the previous motion instruction, current position, and destination position of this motion instruction.

Last spline motion instruction



If CNT is specified in the last spline motion instruction, it will connect smoothly to other motion types. At this time, the taught point does not pass through the end point of the spline motion.

CNT from spline motion to other types of motion

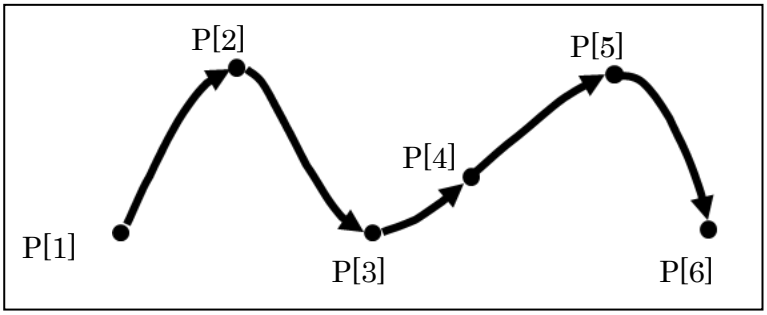


58.2 TERMINATION TYPE OF SPLINE MOTION

Unlike L motion or J motion, S motion passes through taught points even if the termination type is set to CNT 1-100. If CNT100 is specified for by connecting two S-motions with the same speed, the robot will move at a constant speed. If you use FINE or CNT0-99 termination types, the robot will slow down near the point. If you use FINE or CNT0, the robot will pause at the point. At this time, the path does not change. When passing the taught point, the robot passes at an orientation close to the taught posture. If CNT1-100 is specified for connection with other motion types, the connection will be as smooth as the connection between other motion instructions.

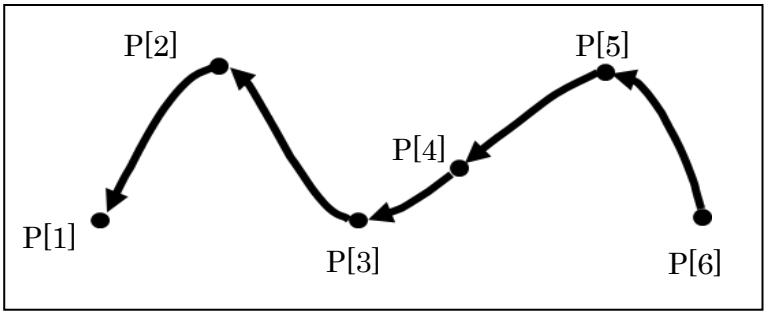
58.3 SINGLE STEP EXECUTION

If a spline instruction is executed in single steps, the program will pause at each taught point as with other motion instruction.



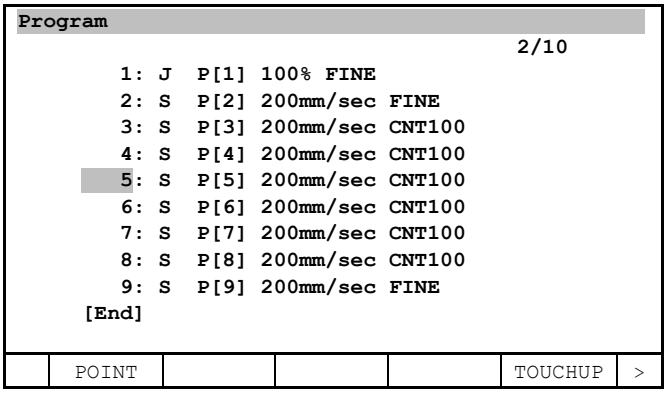
58.4 BACKWARD EXECUTION

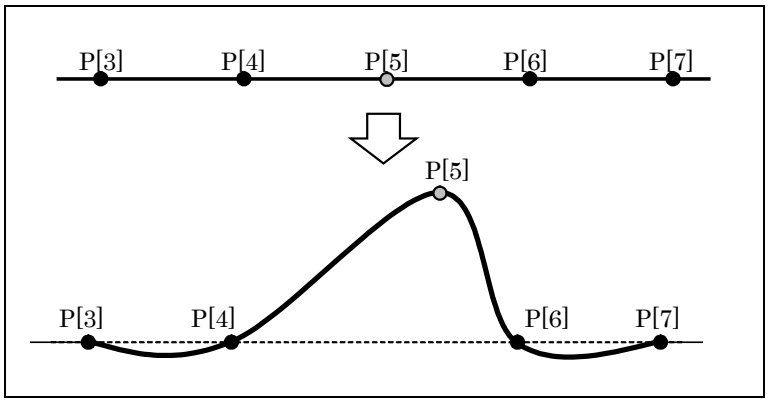
If the spline motion instruction is executed backward, the robot moves backward on the same path as forward execution, and the program is paused at each taught point.



58.5 EFFECT OF A SINGLE TAUGHT POINT ON THE PATH

In the following example, changing the position of P [5] will affect the path from P [3] to P [7]. Changing the position of a single point affects the four spline motion instructions. Note that one point affects many segments in spline motion compared to other motion types.



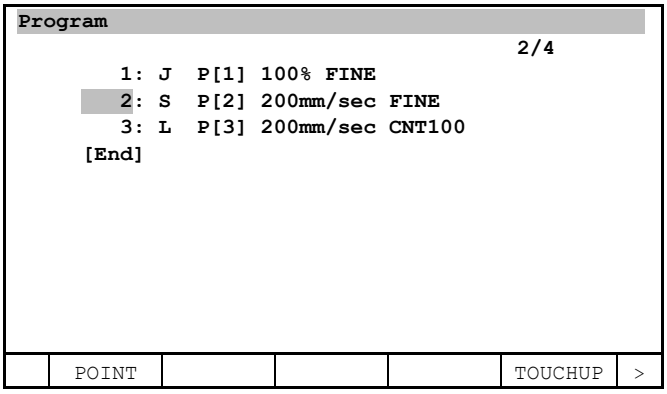


58.6 WHEN SPLINE MOTION IS NOT POSSIBLE

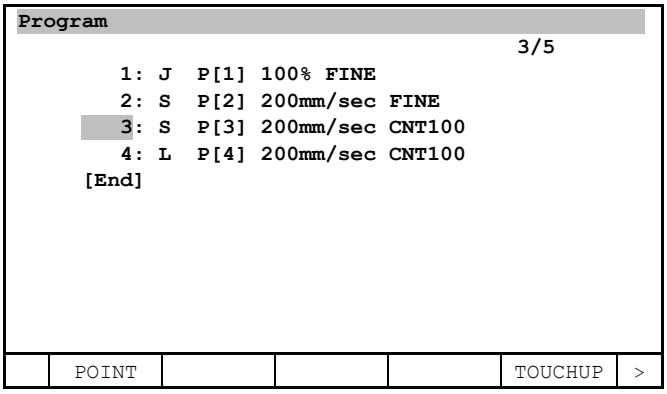
58.6.1 Insufficient Number of Spline Motion Instructions

You must teach three or more consecutive spline motion instructions. If you teach fewer than tree spline motion instructions, the message “INTP-737 SPLIN:S needs 3 points” will be displayed.

Example.1: Only one spline motion instruction has been taught. “INTP-737 SPLIN:S needs 3 points (1)” will be displayed.

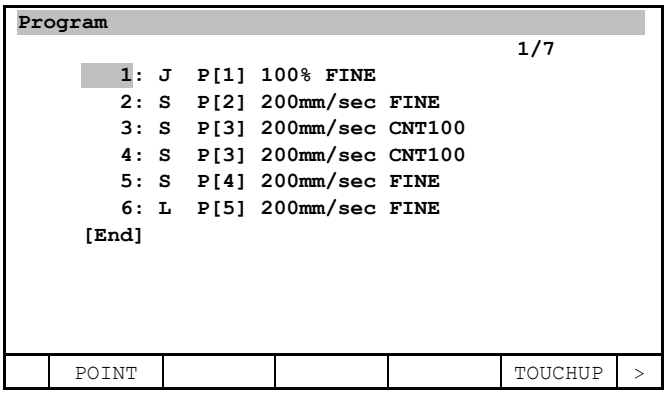


Example.2: There are only two spline motion instructions. “INTP-737 SPLIN:S needs 3 points (2)” will be displayed.



58.6.2 Teaching the Same Position

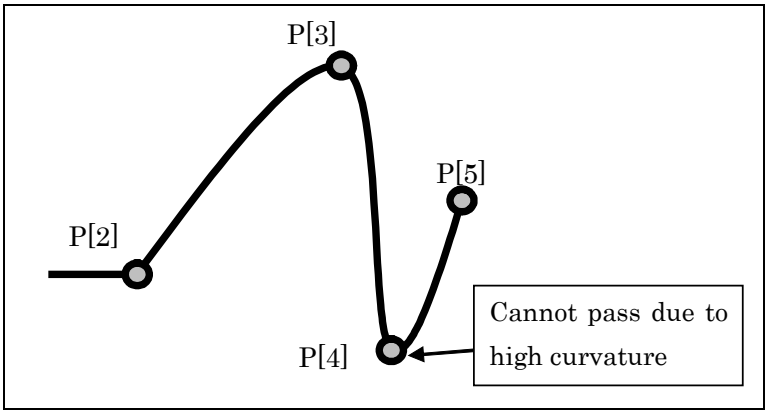
If two consecutive points are taught at the same position, “MOTN-635 SPL: Zero move” will be displayed.



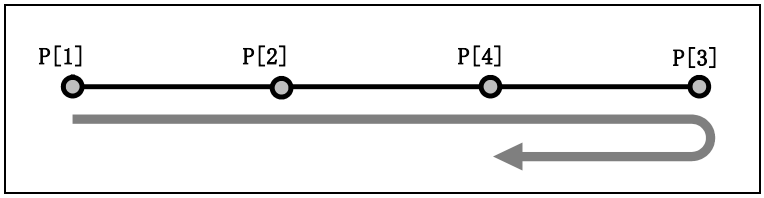
58.6.3 Path has a High Curvature

If the robot cannot move on a curve at the taught speed, it will automatically decelerate. If the automatic deceleration insufficient, an alarm will be issued. If this happened, you must change the position of the point, reduce the taught speed, or change the termination type.

Example.1: If automatic deceleration cannot be completed in time, “MOTN-633 SPL:VAJ Limit” will be displayed.

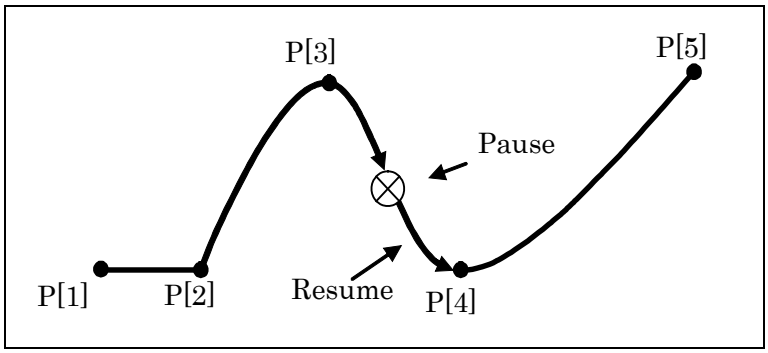


Example.2: When moving in the reverse direction, “MOTN-637 SPL:Invalid Set of Points” will be displayed.



58.7 RESUMING AFTER PAUSING

If robot motion is paused and resumed without changes, the path will be resumed path as usual.

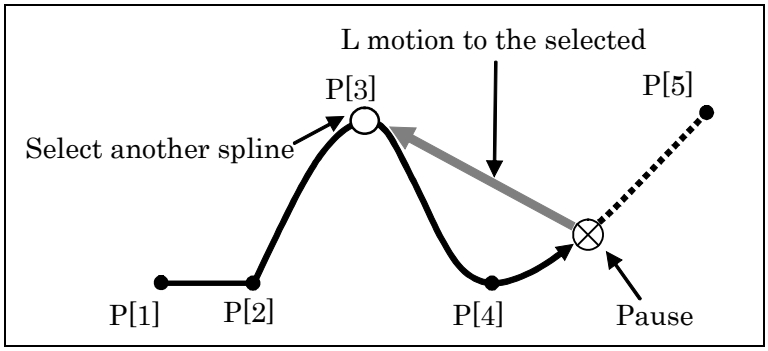


58.7.1 Resuming after Jogging

Suppose that a program is paused during spline motion. Then the robot is jogged and the program is resumed. In this case, if the Original Path Resume function is enabled, the robot returns to the paused position, and then continues the motion. If Original Path Resume is disabled, the robot does not resume. After jogging the robot to a safe position if you start executing from a different line than the paused line, the robot can return to the original path after moving to that point with L motion.

58.7.2 Pausing and Resuming from Another Spline Motion Instruction

The robot moves with L motion to the target position at the current cursor position. When the robot reaches the target position, it draws a curve from that point. If single step or backward execution is used, only the L motion to the target position is executed.



58.7.3 Resuming After a Pause During Backward Execution

If the robot is paused during backward execution, it cannot be resumed and "INTP-731 SPLIN: path change" will be issued. This alarm will not be issued when resuming from another motion instruction.

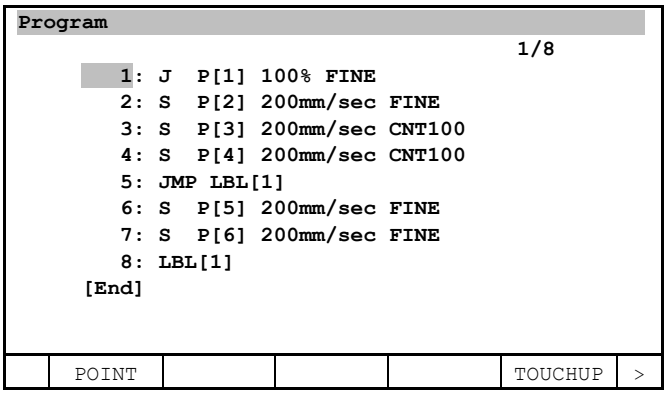
58.7.4 Starting a Program from a Spline Motion Instruction

This case is same as "Pausing and resuming from Another Spline Motion Instruction"

58.8 LOGIC INSTRUCTIONS BETWEEN SPLINE MOTION INSTRUCTIONS

Logic instructions can be taught between spline motion instructions. Spline motion instructions use position of successive data from multiple consecutive spline motion instructions to calculate the curve of a path. For this reason, logic instructions that change the order of execution such as JMP, LBL cannot be executed between spline motions. Logic instructions that affect position data, such as position register instructions, also connot be executed between spline motion instructions.

For example, when the following program is executed, an error occurs when starting motion on line 4.



Logic instructions that can be executed between spline motion instructions

Instructions	Examples	Notes
Register instruction	R[1]=1	
I/O instruction	DO[1]=ON DO[1]=PULSE	DO, RO, UO, SO, WO, GO, AO and F are available.
Wait	WAIT 2.0sec WAIT DI[1]=ON	WAIT /TIMEOUT/ LBL causes an error. When used between S motion instructions connected with CNT1 to CNT100, stop near the taught point on the path. If you want to wait on the exact taught point, set the termination type to FINE or CNT0.
Call	CALL (Prg)	If the called program contains a motion instruction or a logic instruction that cannot be executed between spline motion instructions, an error occurs.
Macro	HAND1 OPEN	If the macro program contains a motion instruction or a logic instruction that cannot be executed between spline motion instructions, an error occurs.
If / Call	IF DI[1]=ON CALL (Prg)	IF / JMP / LBL causes error.

Instructions	Examples	Notes
Select / Call	SELECT R[1]=1 CALL (Prg1) =1 CALL (Prg2) OTHERS CALL(Prg3)	SELECT / JMP / LBL causes error.
RSR	RSR[1]	
User alarm	UARM[1]	
Override	OVERRIDE=50%	
Timer	TIMER[1]=START	
Remark Multi language remark	! (Remark) -- (Remark)	
TP line remark	// J P[1] 100% FINE	
Message	Message[message statement]	
Positional register lock / unlock	LOCK PREG UNLOCK PREG	
Maximum speed	JOINT_MAX_SPEED[1]=50	
Pause	PAUSE	
Program execution instruction	RUN (program name)	
Torque limit	TORQ_LIMIT 50.0%	
Monitor instruction	MONITER (Prg) MONITER END (Prg)	The contents of the program specified by monitor instruction are not checked.
Condition	CONDITION R[1]=1 CALL(Prg)	The condition instruction is not checked, because it can only be taught in the program specified by the monitor instruction.
InterBus-S	IBS ATTACH	
InterBus PCI	IB attach	
Path switch	PS -10mm,1.0sec DO[1]=ON	
Collision detect	COL DETECT ON	
Approach inst	Approach_STOP[1] = TMP_DISABLE	
Arc	Arc Start[1]	
ERROR_PROG	ERROR_PROG=Prg	
RESUME_PROG	RESUME_PROG[1:Comment]=Prg	
CLEAR_RESUME_PROG	CLEAR_RESUME_PROG	
RETURN_PATH_DSBL	RETURN_PATH_DSBL	
MAINT_PROG	MAINT_PROG[1:Comment]=Prg	
TC_ONLINE	TC_ONLINE(!DI[1])	
Data monitor	MONITOR[1] MONITOR END	

Instructions unavailable between spline motions but available in called programs

Execution of the following instructions between spline motion instructions causes an error. However, these instructions can be executed in programs called between the spline motion instructions.

Instructions	Examples	Notes
End	END	
Jump	JMP LBL[1]	
Label	LBL[1]	

Instructions unavailable between spline motion instructions

Instructions other than those listed above cannot be executed between spline motion instructions. The main unavailable instructions are as follows.

Instructions	Examples	Notes
Position register	PREG[1]=LPOS PREG[1,1]=100	
Abort	ABORT	
Offset	OFFSET_CONDITION PR[1]	
Tool offset	TOOL_OFFSET_CONDITION PR[1]	
Vision offset	VOFFSET VR[1]	
User frame User frame number Tool frame Tool frame number	UFRAME[1]=PR[1] UFRAME_NUM=1 UTOOL[1]=PR[1] UTOOL_NUM=1	
Spot tool	SPOT[1]	
Palletizing	PALLETIZING-B_1	
Simultaneous group instruction Independent group instruction	Simultaneous GP Independent GP	

58.9 AVAILABLE MOTION MODIFIERS

Some motion modifiers cannot be used with spline motion instructions. Unavailable motion modifiers will not be displayed in the Motion Modify submenu.

Available Motion Modifiers

Instructions	Example	Comment
Wrist joint	WJNT	
Acceleration override	ACC 80	
Path	PTH	
Offset	Offset PR[1]	
Tool Offset	Tool_Offset PR[1]	
Remote TCP	RTCP	
Time Before Time After Distance Before	TB 2.00sec CALL PRG TA 0.30sec CALL PRG DB10.0mm CALL PRG	If the called program contains instructions that cannot be executed between spline motion instructions in called program, an error is posted when they are executed.
Arc Welding	Arc Start[1]	
Dispense start/end	SS[1]/SE	
Coordinated motion	COORD	
Process speed	PSPD 50	
Vision offset	VOFFSET	
Orient Base	ORNT_BASE	

Main Unavailable Motion Modifiers

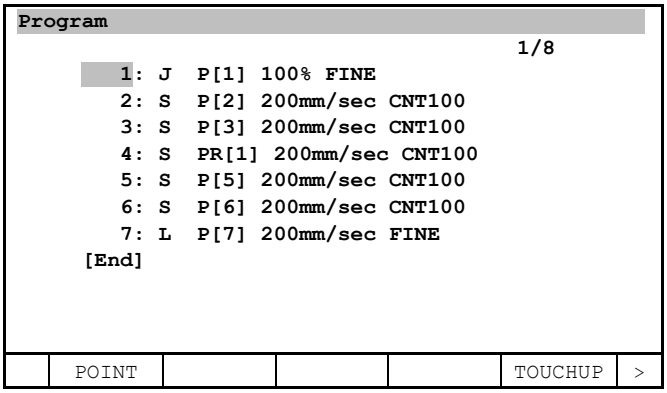
Motion modifiers not listed in the above table cannot be added to the spline motion instruction. The main unavailable motion modifiers are as follows:

Instructions	Example	Comment
Incremental	INC	
Skip	Skip LBL[1]	
High Speed Skip	Skip PR[1] LBL[1]	
Spot Welding	SPOT[1]	
Weave	Weave Sine[1]	
Touch Sensing	Search [X]	
Corner Region	CR50	
Continuous Rotation	CTV100	
Minimal rotation	WJNT MROT	

58.10 CHANGING POSITION DATA DURING S MOTION EXECUTION

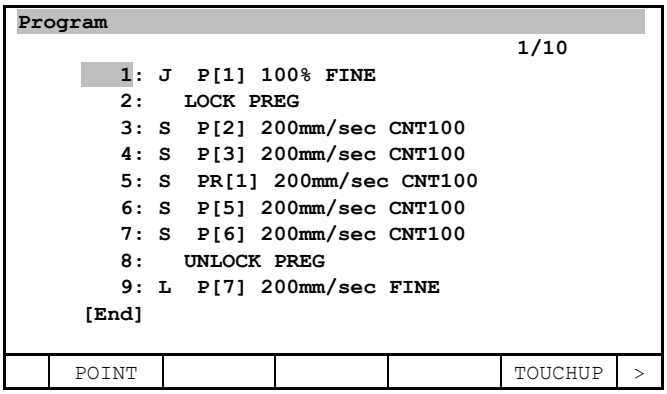
In a program with spline motion instructions, the same position data is accessed by multiple motion instructions. Therefore, if the position data is changed while the data is being used to calculate the curve path, there is a risk of inconsistency in the calculation of the curve path, causing the robot to move along an unexpected path. To avoid this, the spline motion instruction can detect an inconsistent position during execution, and issue an alarm if it is detected.

Example.1: If the value of PR[1] is changed on the position register screen during the execution of line 5, “INTP-755 SPLIN: Inconsistent position” is posted when line 6 is executed.



To avoid stopping production by changing a position register, use the “LOCK PREG”/“UNLOCK PREG” instructions. Teach “LOCK PREG”/“UNLOCK PREG” before/after spline motion instructions that access position registers.

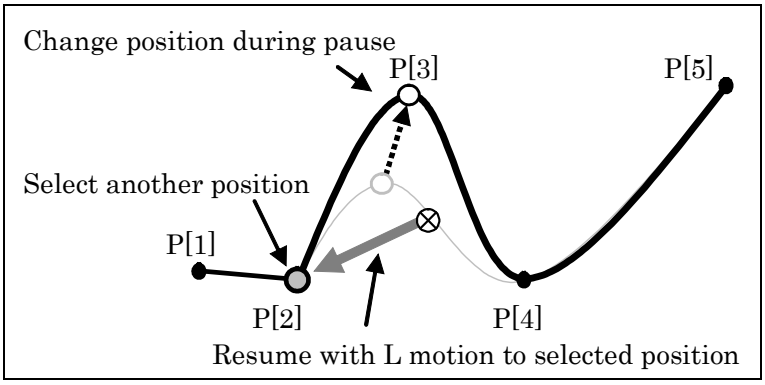
Example.2: When the instructions between the “LOCK PREG” and “UNLOCK PREG” are executed, the value of the position register cannot be changed on the position register screen.



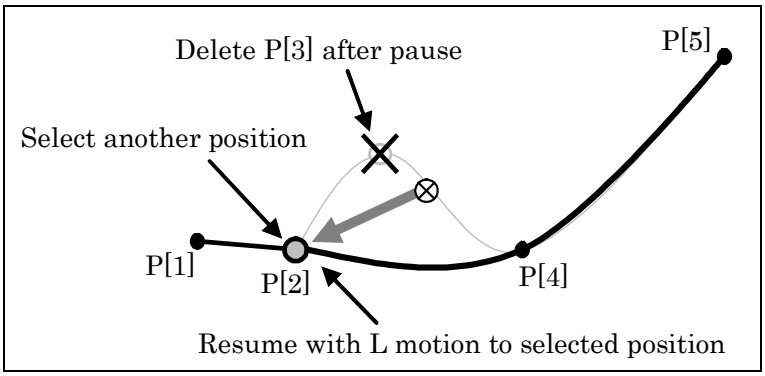
58.11 RESUMING AFTER MODIFYING THE PROGRAM

If the path of the current instruction is modified while a running program is paused during spline motion, the current position will be not on the path. At this time, the program cannot be resumed and an alarm will be issued. After jogging to ensure safety, resume from a different spline motion instruction than the one where you paused. In this case, the robot will move to the target positon with L motion.

Example.1: After pausing an S motion instruction, if you change the position of a point used in the current path, “INTP-731 SPLIN:path change” will be displayed.



Example.2: After pausing an S motion instruction, if you delete the position of a point used in the current path, “INTP-731 SPLIN:path change” will be displayed.



58.12 RESTRICTIONS

- Backward execution from an interrupted program is not supported
- If PAL_*[ ] is used as the position of a type in a spline motion instruction, an error is posted when it is executed.
- If multiple robots are set in the motion group of the TP program, an error is posted when it is executed.
- If a position register is used as the position type in a spline motion instruction and its index is specified indirectly with a register, an error is posted when it is executed.
- If an Offset or Tool_Offset motion modifier is attached to a spline motion instruction and the index of the PR is indirectly specified, an error is posted when it is executed.
- Singularity Avoidance is not supported.
- The LVC function is not supported.
- Line Tracking is not supported.
- Spline motion at the beginning of a program executed by a CALL is changed to L-motion even if the motion type executed just before is S.
- Spline motion just after a CALL for a program containing a motion instruction is changed to L-motion even if the motion type executed just before is S.

59 HIGH SPEED POSITION OUTPUT

This function enables the robot to stream its current Tool Center Point (TCP) or joint angle position at high speed through Ethernet. The data can be output in binary or XML, and Karel variables can also be output. To use this function, R891 must be ordered. This function uses one of the controller's regular Ethernet communication ports.

NOTE
The software that runs on the receiving device used to receive the packets must be provided by the system designer.

NOTE
If it is a system configuration such as multi-group, it may be necessary to set a slower data rate.

59.1 DATA FORMATS

Multiple data types can be selected, and the positions of multiple motion groups can be output. The output data format can be selected from binary or XML. The unit of the output position data is millimeter for the X-Y-Z and degree for the W-P-R and each Joint angles.

⚠ CAUTION
For a fast data rate, please select one data type only to minimize CPU loading. For a slower data rate, more than one data type can be output.

59.1.1 Binary

The following information is added to the front part of the packet output in binary format.

Data in the front part of the packet

Data	Description
Version (long - 4 bytes)	The version number of the data format. It may differ depending on the software version.
Index (long - 4 bytes)	The number of the packet sent. The value increases in the order of output.
Clock (long - 4 bytes)	The internal time of the controller when the packet was sent. The unit is μsec
Type (short - 2 bytes)	The data type of this packet. 1 : TCP Cartesian position 4 : Joint angles 16: Variables

59.1.1.1 TCP Cartesian Position with Binary

When using this data format, the following data is output.

- Version (long - 4 bytes)
- Index (long - 4 bytes)
- Clock (long - 4 bytes)
- Type (short - 2 bytes)
- Motion Group (short - 2 bytes)
- Position [x, y, z, w, p, r] (six floats - 24 bytes total)
- Status (long - 4 bytes)
- I/O (long - 4 bytes)]

59.1.1.2 Joint Angles with Binary

When using this data format, the following data is output.

- Version (long - 4 bytes)
- Index (long - 4 bytes)
- Clock (long - 4 bytes)
- Type (short - 2 bytes)
- Motion Group (short - 2 bytes)
- Position [j1, j2, j3, j4, j5, j6, j7, j8, j9] (nine floats - 36 bytes total)
- Status (long - 4 bytes)
- I/O (long - 4 bytes)]

59.1.1.3 Variables with Binary

When using this data format, the following data is output.

- Version (long - 4 bytes)
- Index (long - 4 bytes)
- Clock (long - 4 bytes)
- Type (short - 2 bytes)
- Data [1 – 10] (ten long/float - 40 bytes total)

59.1.2 XML

59.1.2.1 TCP Cartesian Position with XML

If the TCP Cartesian position is selected to be output with XML, the following packet will be output. "*" are the time, TCP cartesian position, program execution status, and I/O value.

```
<POS><usec>*</usec><G1><X>*</X><Y>*</Y><Z>*</Z><W>*</W><P>*</P><R>*</R></G1></
POS><CYCLE>*</CYCLE><IO>*</IO>
```

59.1.2.2 Joint Angles with XML

If the joint angles is selected to be output with XML, the following packet will be output. "*" are the time, joint angles, program execution status, and I/O value.

```
<ANG><usec>*</usec><G1><J1>*</J1><J2>*</J2><J3>*</J3><J4>*</J4><J5>*</J5><J6>*</J6><J
7>*</J7><J8>*</J8><J9>*</J9></G1></ANG><CYCLE>*</CYCLE><IO>*</IO>
```

59.1.2.3 Variables with XML

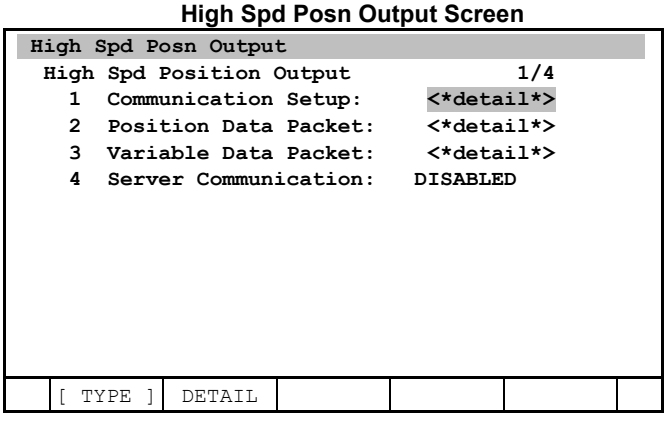
If the Variables is selected to be output with XML, the following packet will be output.
* are the time and selected variables.

```
<VAR><usec>*</usec><V1>*</V1><V2>*</V2><V3>*</V3><V4>*</V4><V5>*</V5><V6>*</V6
><V7>*</V7><V8>*</V8><V9>*</V9><V10>*</V10></VAR>
```

59.2 SETUP

High Speed Position Output is set up by adjusting the settings in the Setup/High Spd Posn Output menu. After setting the settings as described below, change "Server Communication" from DISABLED to ENABLED to start communication.

High Spd Posn Output Screen



NOTE
"Server Communication" must be DISABLED in order to change the settings.

59.2.1 Communication Setup

Change the following settings to adjust the communication method with the server.

Communication Setup Menu Items

Menu Item	Description
Server Name/IP	External process IP address. Changes require controller restart.
Server Port	External process port number. Changes require controller restart.
Protocol	"UDP" or "TCP"
Data Send Rate (ms)	Data sending rate.
Mode	"Binary" or "XML"
Binary Data Type	"Big Endian" or "Little Endian"

59.2.2 Position Data Packet Settings

Change the following settings to adjust the position data packet.

Position Data Packet Menu Items

Menu Item	Description
Cartesian Position Output	Enable Cartesian position output data to be sent.
Joint Angles Output	Enable Joint Angles output data to be sent.
Motion Groups	Set the motion group of the output data. Multiple groups can be selected
Cartesian Pos User Frame	For TCP Cartesian Position, a User Frame must be selected. (default = 0, World Frame)
I/O Type 1	The I/O port type available for sending the program status. By default, this is set to UO signals. This is the port type for the I/O point sent as STATUS or CYCLE in the packet definitions.
I/O Port 1	The I/O port number available for sending the program status. By default, this is 3. STATUS or CYCLE outputs the 3rd I/O port selected by In Cycle I/O Type.
I/O Type 2	This is the I/O port type for the I/O point sent as I/O in the packet definitions. If the system does not use I/O value, you do not need to set this. Select one from DI, DO, AI, AO, RI, RO, GI, GO, UI, UO, F.
I/O Port 2	This is the I/O port number for the I/O point sent as I/O in the packet definitions. If the system does not use I/O value, you do not need to set this.

59.2.3 Variable Data Packet Settings

When Variables Data is output, changing the following settings will select the Karel variables to output.

Variable Data Packet Menu Items

Menu Item	Description
Variable Data Output	Enable Variable output data to be sent.
Variable Data [1-10]	Changing the following menu items will select the variables to output. 10 variables are available to setup. - Program Name : The name of the KAREL program to which the variable belongs. - Variable Name : Variable name - Variable Status : Whether the variable is defined. (read-only)

59.2.4 Start Output

When “Server Communication” is changed from DISABLED to ENABLED after all the settings are completed, it connects to the destination and starts data output. To end the communication, change “Server Communication” to DISABLED.

60 SOFTWARE PLC

The Software PLC function is a PLC (Programmable Logic Controller) integrated in the robot controller, which executes sequence control of the robot system (such as executing robot programs and controlling peripheral devices).
"Sequence control" refers to the type of control where each stage of the process is advanced sequentially according to a predetermined order or logic.
To use Software PLC function, Software PLC option (A05B-2700-S621) is required.

To create a sequence program for Software PLC, you need the CODESYS® Development System and the package file for the FANUC robot, named "CODESYS Control for FANUC Robot" which is installed in the CODESYS Development System.
For information on how to obtain and install these, please refer to "60.2 Preparation".

CODESYS is a registered trademark or a trademark of CODESYS Development GmbH, Germany.

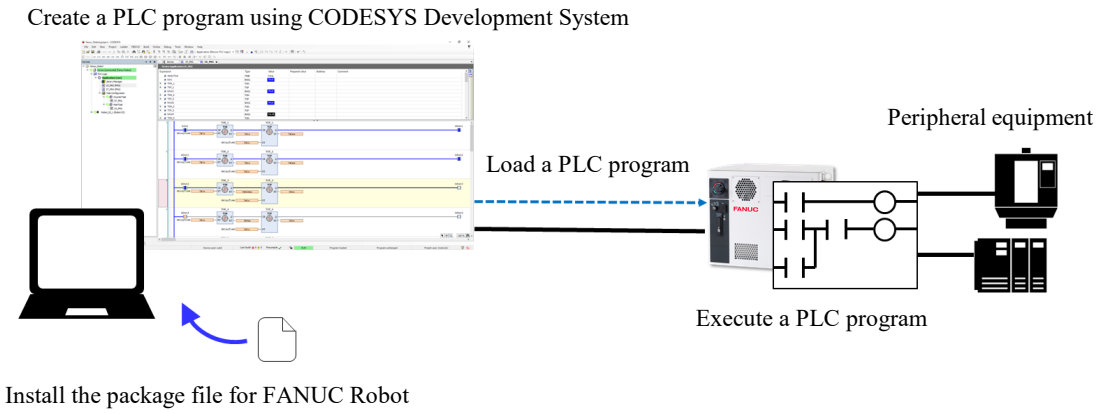


Fig. 60 (a) Overview of Software PLC

This function cannot be used simultaneously with the integrated PMC function (A05B-2700-J760).
The software PLC program is not compatible with the integrated PMC program. If you want to use ladder programs from previous models such as the R-30*i*B Plus, please use the integrated PMC function.

60.1 Overview of Software PLC

The Software PLC function allows for PLC programming as defined by the international standard IEC 61131-3.
This function supports the following six programming languages.

- Structured Text (ST)
- Function Block Diagram (FBD)
- Ladder Diagram (LD)
- Sequential Function Chart (SFC)
- Continuous Function Chart (CFC)
- Instruction List (IL)

Table 60.1 (a) Basic Specifications

Function	Specification
Supported Programming languages	Structured Text (ST) Function Block Diagram (FBD) Ladder Diagram (LD) Sequential Function Chart (SFC) Continuous Function Chart (CFC) Instruction List (IL)

Function	Specification
Memory Capacity for PLC Programs	3.0MB (Approximately 390,000 steps)
Maximum number of tasks	5
Task type	Cyclic
Task Interval	2ms ~ 100000ms
Input/Output area ・Input ・Output	 128 WORD (256 Byte) 128 WORD (256 Byte)

The following operators are available in the Software PLC.
For more details, please refer to the link below
https://content.helpme-codesys.com/en/CODESYS%20Development%20System/_cds_struct_reference_operators.html

Table 60.1 (b) Supported operators

Classification	Operator	Description
Arithmetic operators	ADD	Add
	SUB	Subtract
	MUL	Multiply
	DIV	Divide
	MOD	Modulo division
	MOVE	Assigning a variable to another variable of a corresponding type
	XSIZEOF	Determines the number of bytes which are needed in the passed variable or data type
Bitstring operators	AND	Bitwise AND of bit operands
	OR	Bitwise OR of bit operands
	XOR	Bitwise XOR of bit operands
	NOT	Bitwise NOT of a bit operand
Bitshift operators	SHL	Bitwise shift of an operand to the left
	SHR	Bitwise shift of an operand to the right
	ROL	Bitwise rotation of an operand to the left
	ROR	Bitwise rotation of an operand to the right
Selection Operators	SEL	Bitwise selection
	MAX	Yields the greatest value of all inputs
	MIN	Yields the least value of all inputs
	LIMIT	Limiting
	MUX	Multiplexer
Comparison Operators	GT	Return TRUE if OP1 > OP2
	LT	Return TRUE if OP1 < OP2
	GE	Return TRUE if OP1 ≧ OP2
	LE	Return TRUE if OP1 ≦ OP2
	EQ	Return TRUE if OP1 = OP2
	NE	Return TRUE if OP1 ≠ OP2
Address operators	ADR	Yields the 32-bit address (or the 64-bit address, if possible) of its argument
	<pointer>^	Dereferencing a pointer
	BITADR	Yields the bit offset within a segment in a DWORD
Call operators	CAL	Calling function blocks

Classification	Operator	Description
Type conversion operators	<type1>_TO_<type2>	Type conversion from <type1> to <type2> Example: BYTE_TO_WORD: Conversion from BYTE type to WORD type Data types selectable as <type>: BIT, BOOL, BYTE, DATE, DATE_AND_TIME, DINT, DT, DWORD, INT, LDATE, LDATE_AND_TIME, LDT, LINT, LREAL, LTIME, LTOD, LWORD, REAL, SINT, TIME, TOD, UDINT, UINT, ULINT, USINT, WORD
	TRUNC	Converting the REAL data type into the DINT data type (Takes only the integer part of the number)
	TRUNC_INT	Conversion from REAL to INT (Takes only the integer part of the number)
Numeric Operators	ABS	Yields the absolute value of a number
	SQRT	Yields the square root of a number
	LN	Yields the natural logarithm of a number
	LOG	Yields the base-10 logarithm of a number
	EXP	Yields the exponential function
	EXPT	Returns the power of the base raised to the exponent
	SIN	Yields the sine value of a number
	ASIN	Yields the arcsine value of a number
	COS	Yields the cosine value of a number
	ACOS	Yields the arccosine value of a number
	TAN	Yields the tangent value of a number
	ATAN	Yields the arctangent value of a number
Namespace operators	.<varname>	Global Namespace
	<global variable list name>.<variable>	Namespace for Global Variable Lists
	<enumeration name>.<constant name>	Enumeration Namespace
	<library namespace>.<library POU identifier>	Library Namespace

The following literals (constants directly written in the program code, such as numbers and strings) can be used in the Software PLC.
For more details, please refer to the link below.
https://content.helpme-codesys.com/en/CODESYS%20Development%20System/_cds_struct_reference_operands.html

Table 60.1 (c) Supported literals

Type	Example
BOOL	TRUE : 1 FALSE : 0
Decimal number	14
Binary number	2#1001_0011
Octal number	8#67
Hexadecimal number	16#A
Decimal number(REAL)	7.4
Decimal fraction	1/3.0 : 0.333333343
Exponential notation	1.64e+009 1.64E+009
String	'Hello world!'
UTF8# String	UTF8#'aäoöuü'
Unicode character	UCHAR# '๓'
TIME, LTIME constants to operate	T#14ms

Type	Example
the standard timer modules	T#100s12ms T#12h34m15s T#1d23h45m6s
Date (32 bits)	DATE#2018-8-8 D#2018-8-31 date#1996-05-06 Range : D#1970-1-1 ~ D#2106-2-7
Date (64 bits)	LDATE#2018-8-8 LD#1996-05-06 ldate#2018-8-31 Range : LD#1970-1-1 ~ LD#2554-7-21
Time (32 bits)	TIME_OF_DAY#0:0:0.000 TOD#23:59:59.999
Time (64 bits)	TIME_OF_DAY#0:0:0 TOD#23:59:59.999999999
Date and Time (32 bits)	DATE_AND_TIME#1996-05-06-15:36:30 DT#1972-03-29-00:00:00 Range: DT#1970-1-1-00:00:00 ~ DT#2106-2-7-6:28:15
Date and Time (64 bits)	LDATE_AND_TIME#1996-05-06-15:36:30 LDT#1972-03-29-00:00:00 Range: LDT#1970-1-1-00:00:00 ~ LDT#2554-7-21-23:59:59
Typed Literal	DINT#34 If the type is not explicitly specified, the smallest type is used.
Access to array elements	aiCounter : ARRAY[0..9] OF INT … iLocalVariable := aiCounter[2];

Access to structure components	TYPE S_POLYGONLINE :
	STRUCT
	aiStart : ARRAY[1..2] OF INT := [-99, -99];
	aiPoint1 : ARRAY[1..2] OF INT;
	aiPoint2 : ARRAY[1..2] OF INT;
	aiPoint3 : ARRAY[1..2] OF INT;
	aiPoint4 : ARRAY[1..2] OF INT;
	aiEnd : ARRAY[1..2] OF INT := [99, 99];
	END_STRUCT
	END_TYPE
	…
	iPoint := sPolygon.aiPoint1[1];

Access to variables in POUs	xStart defined within the POU called fbController … fbController.xStart := TRUE;
Bit Access in Variables	wA.2 : The 2nd bit of wA
Addresses	%QB7 : The address of the 7th byte in the output memory map %IW215 : The address of the 215th word in the input memory map

The following data types can be used in the Software PLC.
For more details, please refer to the link below.
https://content.helpme-codesys.com/en/CODESYS%20Development%20System/_cds_reference_datatypes.html

Table 60.1 (d) Supported data types

Type name	Memory size	Range
BOOL	8 bits	TRUE(1), FALSE(0)
BYTE	8 bits	0 ~ 255
WORD	16 bits	0 ~ 65535
DWORD	32 bits	0 ~ 4294967295

Type name	Memory size	Range
LWORD	64 bits	0 ~ 2^{64}-1
SINT	8 bits	-128 ~ 127
USINT	8 bits	0 ~ 255
INT	16 bits	-32768 ~ 32767
UINT	16 bits	0 ~ 65535
DINT	32 bits	-2147483648 ~ 2147483647
UDINT	64 bits	0 ~ 4294967295
LINT	64 bits	-2^{63} ~ 2^{63}-1
ULINT	64 bits	0 ~ 2^{64}-1
REAL	32 bits	1.0E-44 ~ 3.402823E+38
LREAL	64 bits	4.94065645841247E-324 ~ 1.7976931348623157E+308
STRING	Latin-1 encoding: 8 bits / 1 文字 UTF-8 encoding: 32 bits / 1 文字	~ 255 characters (If the number of characters is not specified, it defaults to ~8 characters.)
WSTRING	16 bits (UCS-2 encoding)	~ 255 characters (If the number of characters is not specified, it defaults to ~8 characters.)
TIME	32 bits	T#0d0h0m0s0ms ~ T#49d17h2m47s295ms
LTIME	64 bitts	LTIME#0NS ~ LTIME#213503D23H34M33S709MS551US615 NS
DATE	32 bits	DATE#1970-01-01 ~ DATE#2106-02-07 (D#1970-01-01 ~ D#2106-02-07)
DATE_AND_TIME (DT)	32 bits	DATE_AND_TIME#1970-1-1-0:0:0 ~ DATE_AND_TIME#2106-02-07-06:28:15 (DT#1970-1-1-0:0:0 ~ DT#2106-02-07-06:28:15)
TIME_OF_DAY (TOD)	32 bits	TIME_OF_DAY#0:0:0 ~ TIME_OF_DAY#23:59:59.999 (TOD#0:0:0 ~ TOD#23:59:59.999)
LDATE	64 bits	LDATE#1970-1-1 ~ LDATE#2554-7-21 (LD#1970-1-1 ~ LD#2554-7-21)
LDATE_AND_TIME (LDT)	64 bits	LDATE_AND_TIME#1970-1-1-0:0:0 ~ LDATE_AND_TIME#2554-7-21-23:34:33.709551615 (LDT#1970-1-1-0:0:0 ~ LDT#2554-7-21-23:34:33.709551615)
LTIME_OF_DAY (LTOD)	64 bits	LTIME_OF_DAY#0:0:0 ~ LTIME_OF_DAY#23:59:59.999999999 (LTOD#0:0:0 ~ LTOD#23:59:59.999999999)
BIT	1 bit Can only be used within a structure member or a function block	TRUE(1), FALSE(0)

In the IEC 61131-3 standard library (Standard), the following function blocks and functions can be used.
For more details, please refer to the link below.
https://content.helpme-codesys.com/en/libs/Standard/3.5.18.0/index.html

Table 60.1 (e) Supported function blocks and functions in the IEC 61131-3 standard library (Standard)

Classification	Function block name / Function name	Description
Bistable	SR	A bistable reset-dominat latch
	RS	A bistable set-dominat latch
Counter	CTU	Increments a given value
	CTD	Decrements a given value
	CTUD	Increments and decrements a given value

Classification	Function block name / Function name	Description
Timer	TON	A timer with a turn-on delay
	TOF	A timer with a turn-off delay
	TP	A pulse timer
Trigger	F_TRIG	Detects a falling edge of a boolean signal
	R_TRIG	Detects a rising edge of a boolean signal
String Functions	CONCAT	Concatenates two strings
	DELETE	Deletes a number of characters from a string
	FIND	Searches for the position of a partial string within a string
	INSERT	Inserts a string into another string at a specific position
	LEFT	Returns a specific number of characters of a string, starting from left
	LEN	Returns the number of characters of a string
	MID	Returns a specific number of characters of a string, starting from a specific position
	REPLACE	Replaces a specific number of characters of a string by another string
	RIGHT	Returns a specific number of characters of a string, starting from right

The Software PLC supports the following libraries.
- 3SLicense (Default)
- Analyzation (SFC)
- BreakpointLogging (Default)
- CAA Device Diagnosis (Default)
- IecSfc (SFC)
- IoStandard (Default)
- Standard (Default)

60.2 Preparation

Download the CODESYS Development System from the following website and install it on your PC.

LINX Corporation (Japan)
https://linx.jp/download/
CODESYS Store North America (North America)
https://us.store.codesys.com/
CODESYS Store International (Other regions)
https://store.codesys.com/en/

Check the installation requirements and precautions on each download page.
The Software PLC supports the following versions of the CODESYS Development System. Please use the versions listed below.
- V3.5 SP19

To enable communication between the CODESYS Development System and the robot control device, and to add I/O devices for the FANUC robot, you need to install CODESYS Control for FANUC Robot in the CODESYS Development System.
CODESYS Control for FANUC Robot can be downloaded from FANUC membership website.

FANUC membership website (Japan)
https://member.fanuc.co.jp/portal
For the regions other than the above, contact the service base of the neighborhood.

Double-click the downloaded "CODESYS_Control_for_FANUC_Robot_x_x_x_x.package" (where x_x_x_x represents the version of Control for FANUC Robot) to install CODESYS Control for FANUC Robot.

- The software license agreement is displayed. Please make sure to review all the contents carefully.
  If you agree to all the terms, check the box at the bottom of the window labeled "I accept the license agreement(s)"
- CODESYS Control for FANUC Robot can be uninstalled via the CODESYS Installer. If any issues occur during installation and it does not install correctly, try uninstalling it and then reinstalling.

60.3 I/O assignment for The Software PLC

The Software PLC has the following two types of I/O assignments.

- **Software PLC External I/O Assignment**
  Assign the input and output signals of external I/O devices to the Software PLC Input/Output area.
- **Software PLC Internal I/O Assignment**
  Assign the input and output signals, such as the robot's DI/DO, to the Software PLC Input/Output area.

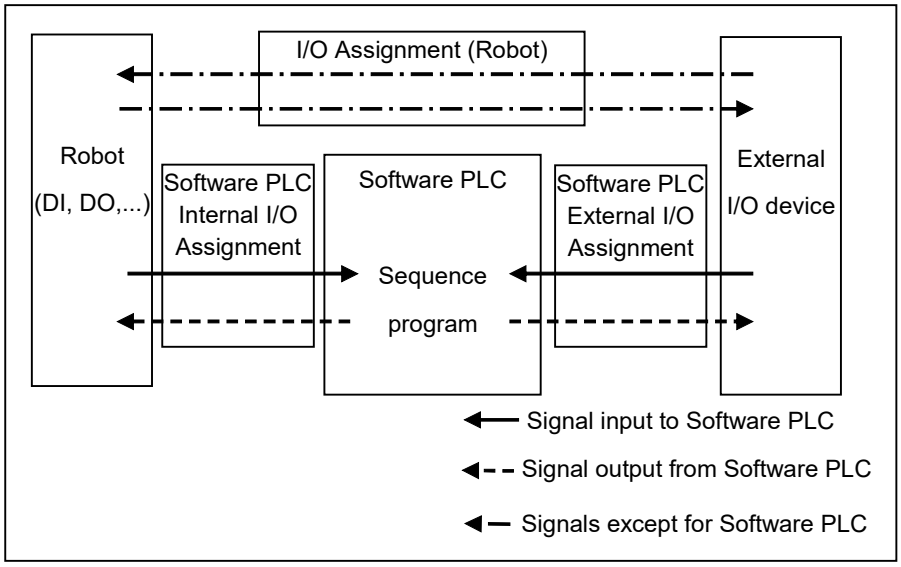


Fig. 60.3 (a) I/O assignment for the robot controller

60.3.1 Software PLC Input/Output area

The Software PLC Input/Output area is the region where the Software PLC executes input and output operations with external I/O devices or signals such as the robot's DI/DO.

When you install CODESYS Control for FANUC Robot in the CODESYS Development System, the following I/O devices for the FANUC robot are added.

- FANUC Robot IO
  - In 64 Word
  - Out 64 Word

By configuring these devices in the CODESYS Development System, the Software PLC Input/Output area becomes available for use.

- FANUC Robot IO is automatically added when creating a project for the Software PLC. For creating a project for the Software PLC, refer to "60.4.1 Creating a Project for the Software PLC"
- In 64 Word and Out 64 Word are devices of the Software PLC Input/Output area. They can be added under FANUC Robot IO in the device tree of the CODESYS Development System.
- In 64 Word is the input area of the Software PLC. It is the area where the output values of external I/O devices or output signals such as the robot's DO are input into the Software PLC.
- Out 64 Word is the output area of the Software PLC. It is the area where the Software PLC outputs values to the input signals of external I/O devices or input signals such as the robot's DI.

- In 64 Word and Out 64 Word are 64-WORD areas of the Software PLC Input/Output area. Each WORD consists of 16 bits.
- In the CODESYS Development System, you can map PLC program variables to the Software PLC Input/Output area. Only variables of type WORD (2 bytes) or BOOL (1 bit) can be mapped.
  WORD variables can be mapped to a 1-WORD channel in the Software PLC Input/Output area.
  BOOL variables can be mapped to a 1-Bit channel in the Software PLC Input/Output area.
- In 64 Word and Out 64 Word can each be added up to two times under FANUC Robot IO. Add them if you need an Input/Output area larger than 64 Word (equivalent to 128 bytes). In this case, set the Offset to 128 in the "Parameters" tab of the second device.
- In the "I/O Mapping" tab of the device, you can map each variable to the Software PLC Input/Output area.
  For specific configuration methods, refer to "60.4.3 Addition of I/O devices for the FANUC robot and the I/O mapping"

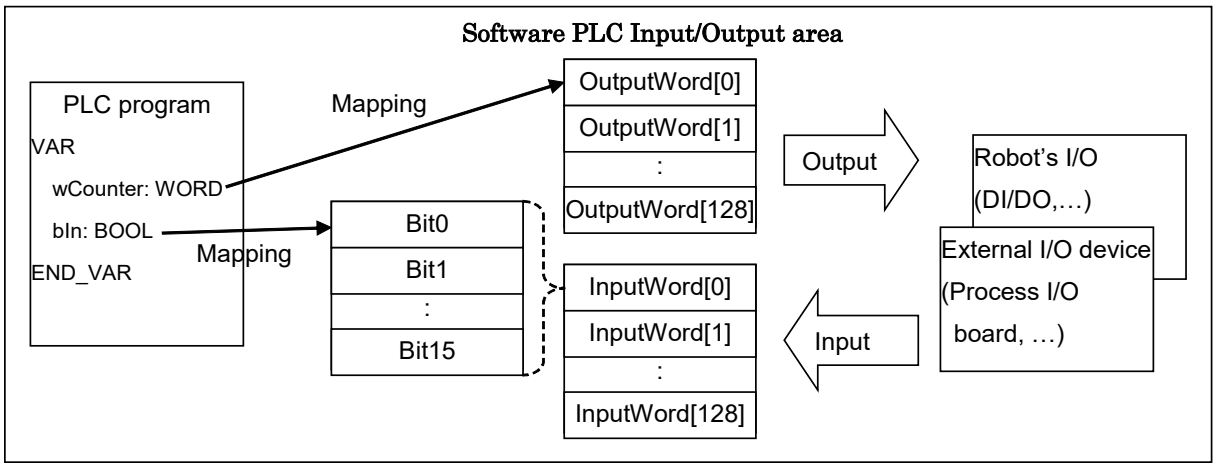


Fig. 60.3.1 (a) Software PLC Input/Output Area and Variable I/O Mapping

60.3.2 Software PLC External I/O Assignment

By configuring the Software PLC External I/O Assignment, the Software PLC can input and output signals to and from external I/O devices.
The Software PLC External I/O Assignment assigns the input and output signals of external I/O devices (specified by Signal type, Rack, and Slot) to the Software PLC Input/Output area.

The Software PLC External I/O Assignment is configured on the Software PLC External I/O Assignment screen (MENU → "I/O" → F1" [TYPE]" → "Software PLC" → F2"[DATA] " → "Ext. I/O asg").

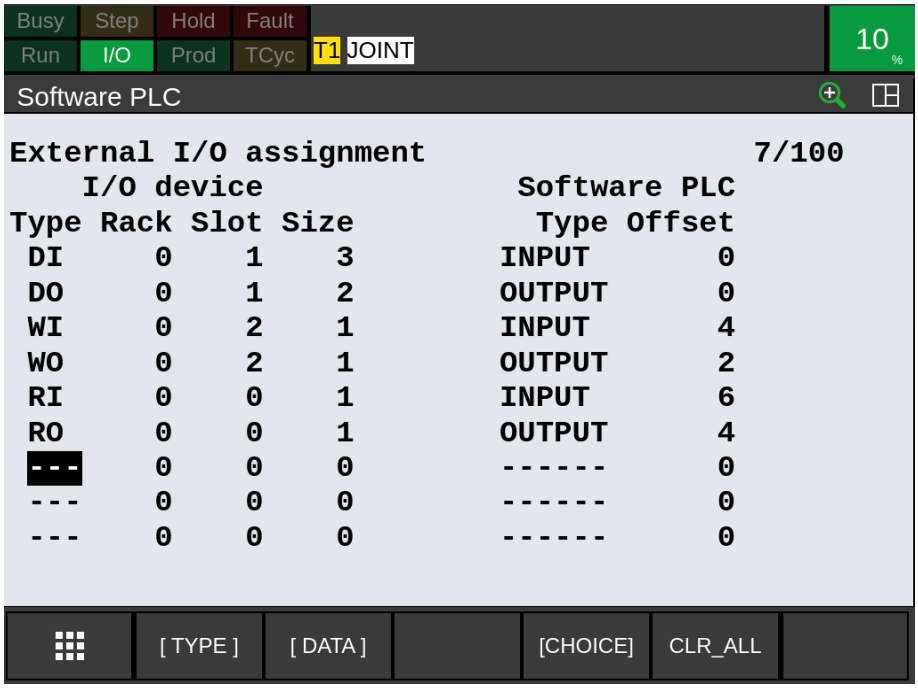


Fig. 60.3.2 (a) Software PLC External I/O Assignment screen

- In the "Type" column of the "I/O device", set the signal type. For details on signal types, refer to " Table 60.3.2 (a) Signal types for Software PLC External I/O Assignment". The "Type" of the "Software PLC" is automatically set according to the signal type specified in the "Type" of the "I/O device".
- Specify the signals of the external I/O device using the "Type", "Rack" and "Slot" columns of the "I/O device". Then, set the offset (Unit: Byte) of the Software PLC Input/Output area in the "Offset" column of the "Software PLC" to configure the Software PLC External I/O Assignment. If configured correctly, the device name of the external I/O device is displayed in the prompt line.
- The "Size" column is automatically determined by the signal type and number of the configured external I/O device and cannot be changed. If the corresponding external I/O device is not connected, "Size" is displayed as 0.
- The maximum number of setting lines is 100.
- Pressing F5 [CLR_ALL] removes all settings on the Software PLC External I/O Assignment screen.
- The Software PLC External I/O Assignment settings will take effect after cycle power.

For example, if a process I/O board MA is connected to rack 0, slot 1, by configuring as follows, 20 digital input signals (Size: 3 bytes) will be assigned from the 0th byte of the Software PLC Input area.



Fig. 60.3.2 (b) Example of the setting the Software PLC External I/O Assignment

Table 60.3.2 (a) Signal types for the Software PLC External I/O Assignment

IO device Type	Software PLC Type	Description
DI (Digital Input)	INPUT	Digital input signals from external I/O devices. Specify the external I/O device by rack and slot. Each digital input signal corresponds to 1 bit in the Software PLC Input area.
DO (Digital Output)	OUTPUT	Digital output signals from external I/O devices. Specify the external I/O device by rack and slot. Each digital output signal corresponds to 1 bit in the Software PLC Output area.
AI (Analog Input)	INPUT	Analog input signals from external I/O devices. Specify the external I/O device by rack and slot. Each analog input signal corresponds to 2 bytes in the Software PLC Input area.

IO device Type	Software PLC Type	Description
AO (Analog Output)	OUTPUT	Analog output signals from external I/O devices. Specify the external I/O device by rack and slot. Each analog output signal corresponds to 2 bytes in the Software PLC Output area.
RI (Robot Input)	INPUT	Robot input signals. Specify 0 for both rack and slot. Each input signal corresponds to 1 bit in the Software PLC Input area.
RO (Robot Output)	OUTPUT	Robot output signals. Specify 0 for both rack and slot. Each output signal corresponds to 1 bit in the Software PLC Output area.
SI (SOP Input)	INPUT	SOP input signals. Specify 0 for both rack and slot. Each input signal corresponds to 1 bit in the Software PLC Input area.
WI (Welding interface digital Input)	INPUT	Welding interface digital input signals from the process I/O board. Specify the process I/O board by rack and slot. Each input signal corresponds to 1 bit in the Software PLC Input area.
WO (Welding interface digital output)	OUTPUT	Welding interface digital output signals from the process I/O board. Specify the process I/O board by rack and slot. Each output signal corresponds to 1 bit in the Software PLC Output area.
WSI (Wire Soldering detector Input)	INPUT	Wire Soldering detector input signals from the process I/O board. Specify the process I/O board by rack and slot. Each input signal corresponds to 1 bit in the Software PLC Input area.
WSO (Wire Soldering detector Output)	OUTPUT	Wire Soldering detector output signals from the process I/O board. Specify the process I/O board by rack and slot. Each output signal corresponds to 1 bit in the Software PLC Output area.

The specifications for the Software PLC External I/O Assignment are as follows.

- In the Software PLC External I/O Assignment, the entire signal of the specified external I/O device's signal type (e.g., DI, DO) is assigned to the Software PLC Input/Output area. For example, when assigning an external I/O device with 20 digital input signals, all 3 bytes (24 bits, with the last 4 bits unused) are assigned. It is not possible to assign only a part of the signal (e.g., only 1 byte out of 3 bytes). However, for external I/O devices that have both input signals (DI) and output signals (DO), it is possible to assign only the input signals or only the output signals.
- It is not possible to specify the external I/O device that is already configured in another line of the Software PLC External I/O Assignment screen.
- 110 cannot be specified for "Rack".
- RI (Robot Input), RO (Robot Output), and SI (SOP Input) cannot specify anything other than 0 for the "Rack" and "Slot".
- The "Offset" of the "Software PLC" must be a multiple of 2.
- For lines where the "Offset" of the Software PLC is less than 128 bytes, the sum of the "Size" and the "Offset" of the "Software PLC" within that line must be within 128 bytes.
- The sum of the "Size" and the "Offset" of the "Software PLC" within one line must be within 256 bytes.
- You cannot specify the Software PLC Input/Output area that is already set in another line on the Software PLC External I/O Assignment screen or the Software PLC Internal I/O Assignment screen.
- Since the entire 2 bytes of the AI (Analog Input) are copied to the Software PLC Input area, the obtained data must be masked according to the specifications of the external I/O device used. For the mask range, refer to the manual of the external I/O device (such as the Process I/O Board or I/O Unit-MODEL A) being used.

60.3.2.1 Relationship with I/O Assignment (Robot)

The Software PLC External I/O Assignment and the I/O Assignment (Robot) are independent settings. Changing the Software PLC External I/O Assignment does not affect the correspondence between the robot's DI/DO and the external I/O devices. Similarly, changing the I/O Assignment (Robot) does not affect the correspondence between the Software PLC Input/Output area and the external I/O devices.

It is possible to set an external I/O device to either the Software PLC External I/O Assignment or the I/O Assignment (Robot), or to both.

The figure below is an example of assigning input signals of an external I/O device to both the Software PLC Input area and the robot's DI. When Input signal 1 turns ON, the first bit of the assigned Software PLC Input area (InputWord[0].0) becomes 1, and at the same time, DI[1] also turns ON.

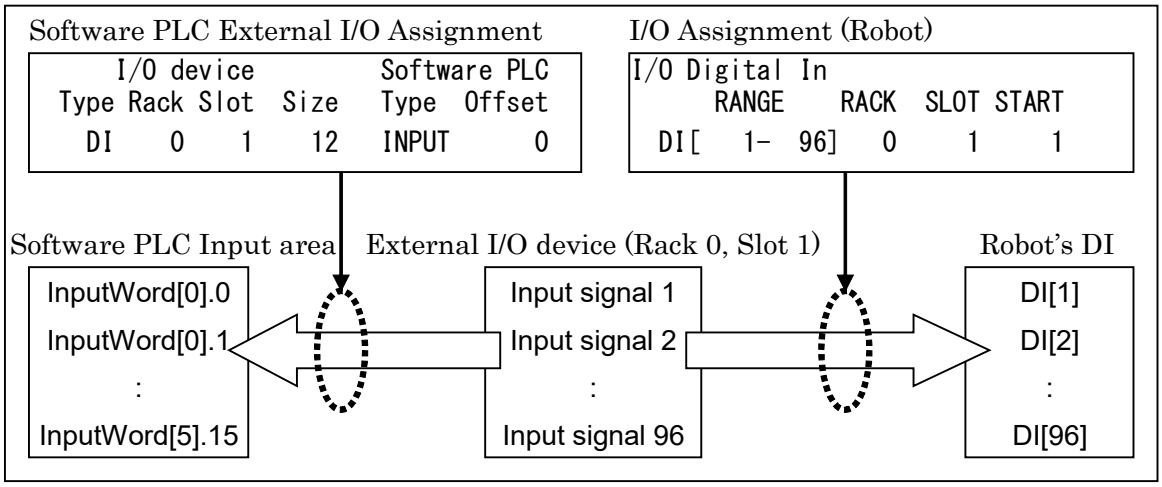


Fig. 60.3.2.1 (a) Software PLC External I/O Assignment for Input signals

The figure below is an example of assigning output signals of an external I/O device to both the Software PLC Output area and the robot's DO.
When the first bit of the assigned Software PLC Output area (OutputWord[0].0) is set to 1(0), Output signal 1 turns ON(OFF). At this time, DO[1] also turns ON(OFF).
Additionally, when DO[1] is turned ON(OFF), Output signal 1 also turns ON(OFF). At this time, it corresponds to OutputWord[0].0. If signal output is executed from both OutputWord[0].0 and DO[1], Output signal 1 will take the value of the most recent output.

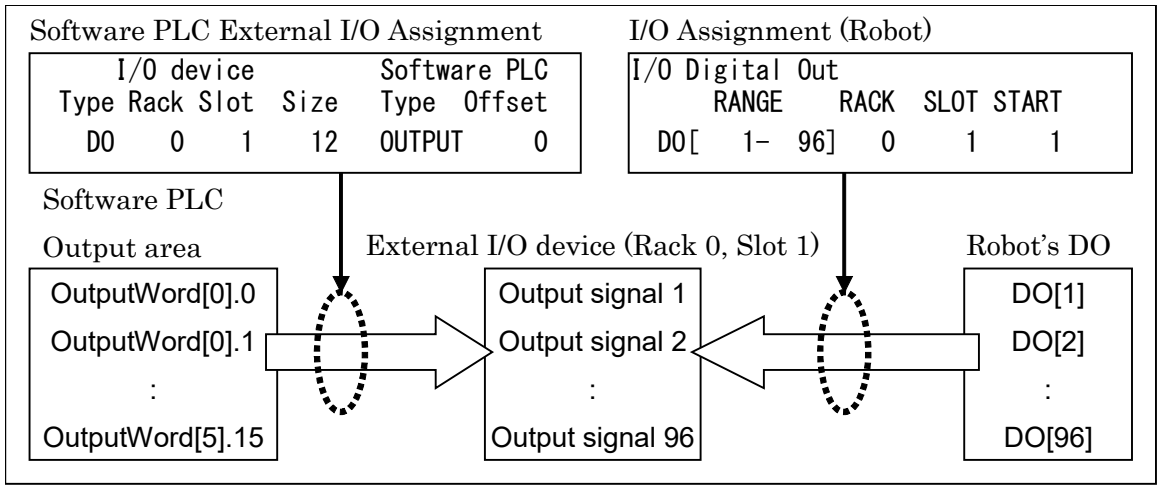


Fig. 60.3.2.1 (b) Software PLC External I/O Assignment for Output signals

⚠ CAUTION

- Since the Software PLC External I/O Assignment is independent of the I/O Assignment (Robot), the ports of the external I/O devices specified in the Software PLC External I/O Assignment can be assigned to the robot's I/O. Therefore, the I/O values written by the Software PLC may be overwritten by the robot's program or other means. If the robot's I/O auto assignment setup is enabled, external I/O devices are automatically assigned to the robot's I/O.
- The following functions of the robot's I/O do not affect the corresponding Software PLC Input/Output area.
  1 Even if the robot's I/O is set to Simulated, the Software PLC Input/Output area executes input and output operations based on the actual signals from the external I/O devices.
  2 Even if the Polarity of the robot's I/O is set to INVERSE, the polarity of the bits within the Software PLC Input/Output area are not reversed.
  3 Even if the robot's I/O is set to Complementary, the output from the Software PLC is not complementary.

60.3.2.2 Example of Setting the Software PLC External I/O Assignment and Mapping

Here is an example of setting the Software PLC External I/O Assignment and Mapping for using the Software PLC with the following configuration.

- A process I/O board with 20 digital inputs and 16 digital outputs is connected.
  (The process I/O board is recognized as being connected at Rack 0, Slot 1)
- Assign 16 digital inputs to the beginning of the Software PLC Input area (InputWord[0]).
- Assign 20 digital outputs to the beginning of the Software PLC Output area (OutputWord[0]).
- Map variables to the beginning of the Software PLC Input area (Bit 0~15 of InputWord[0]) to receive digital inputs from the process I/O board in the PLC program.
- Map variables to the beginning of the Software PLC Output area (Bit 0~15 of OutputWord[0]) to output to the digital outputs of the process I/O board from the PLC program.

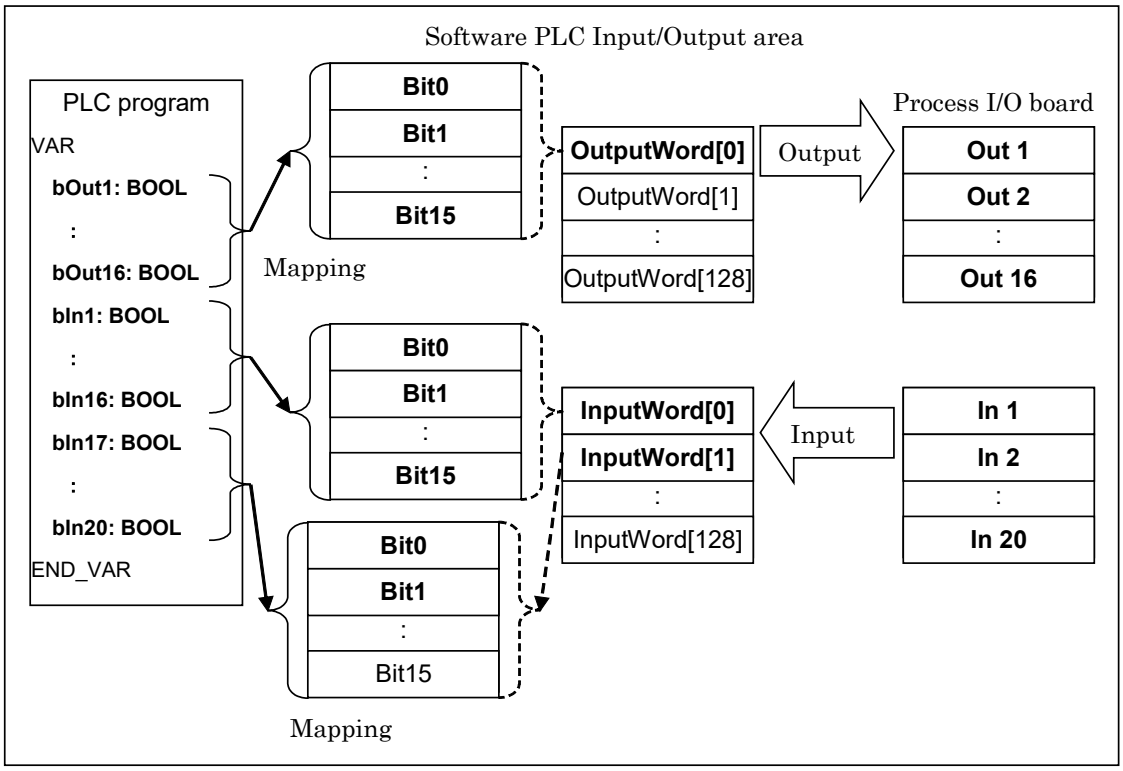


Fig. 60.3.2.2 (a) Example of setting the Software PLC External I/O Assignment and Mapping

Configure as follows.

Software PLC External I/O Assignment screen

```
    I/O device                    Software PLC
Type Rack Slot Size                 Type Offset
 DI     0    1    3               INPUT      0
 DO     0    1    2               OUTPUT     0
```

Fig. 60.3.2.2 (b) Example of setting the Software PLC External I/O Assignment

I/O Mapping in the CODESYS Development System

Variable	Mapping	Channel	Address	Type
		InputWord	%IW0	ARRAY [0..63] OF WORD
		InputWord[0]	%IW0	WORD
Application.PLC_PRG.bIn1		Bit0	%IX0.0	BOOL
Application.PLC_PRG.bIn2		Bit1	%IX0.1	BOOL
Application.PLC_PRG.bIn3		Bit2	%IX0.2	BOOL
Application.PLC_PRG.bIn4		Bit3	%IX0.3	BOOL
Application.PLC_PRG.bIn5		Bit4	%IX0.4	BOOL
Application.PLC_PRG.bIn6		Bit5	%IX0.5	BOOL
Application.PLC_PRG.bIn7		Bit6	%IX0.6	BOOL
Application.PLC_PRG.bIn8		Bit7	%IX0.7	BOOL
Application.PLC_PRG.bIn9		Bit8	%IX1.0	BOOL
Application.PLC_PRG.bIn10		Bit9	%IX1.1	BOOL
Application.PLC_PRG.bIn11		Bit10	%IX1.2	BOOL
Application.PLC_PRG.bIn12		Bit11	%IX1.3	BOOL
Application.PLC_PRG.bIn13		Bit12	%IX1.4	BOOL
Application.PLC_PRG.bIn14		Bit13	%IX1.5	BOOL
Application.PLC_PRG.bIn15		Bit14	%IX1.6	BOOL
Application.PLC_PRG.bIn16		Bit15	%IX1.7	BOOL
		InputWord[1]	%IW1	WORD
Application.PLC_PRG.bIn17		Bit0	%IX2.0	BOOL
Application.PLC_PRG.bIn18		Bit1	%IX2.1	BOOL
Application.PLC_PRG.bIn19		Bit2	%IX2.2	BOOL
Application.PLC_PRG.bIn20		Bit3	%IX2.3	BOOL
		Bit4	%IX2.4	BOOL

Fig. 60.3.2.2 (c) Example of Input Mapping

Variable	Mapping	Channel	Address	Type
		OutputWord	%QW0	ARRAY [0..63] OF WORD
		OutputWord[0]	%QW0	WORD
Application.PLC_PRG.bOut1		Bit0	%QX0.0	BOOL
Application.PLC_PRG.bOut2		Bit1	%QX0.1	BOOL
Application.PLC_PRG.bOut3		Bit2	%QX0.2	BOOL
Application.PLC_PRG.bOut4		Bit3	%QX0.3	BOOL
Application.PLC_PRG.bOut5		Bit4	%QX0.4	BOOL
Application.PLC_PRG.bOut6		Bit5	%QX0.5	BOOL
Application.PLC_PRG.bOut7		Bit6	%QX0.6	BOOL
Application.PLC_PRG.bOut8		Bit7	%QX0.7	BOOL
Application.PLC_PRG.bOut9		Bit8	%QX1.0	BOOL
Application.PLC_PRG.bOut10		Bit9	%QX1.1	BOOL
Application.PLC_PRG.bOut11		Bit10	%QX1.2	BOOL
Application.PLC_PRG.bOut12		Bit11	%QX1.3	BOOL
Application.PLC_PRG.bOut13		Bit12	%QX1.4	BOOL
Application.PLC_PRG.bOut14		Bit13	%QX1.5	BOOL
Application.PLC_PRG.bOut15		Bit14	%QX1.6	BOOL
Application.PLC_PRG.bOut16		Bit15	%QX1.7	BOOL
		OutputWord[1]	%QW1	WORD

Fig. 60.3.2.2 (d) Example of Output Mapping

For I/O mapping in the CODESYS Development System, refer to "60.4.3 Addition of I/O devices for the FANUC robot and the I/O mapping".

60.3.3 Software PLC internal I/O Assignment

By configuring the Software PLC Internal I/O Assignment, the Software PLC can execute input and output operations with the robot's programs and other systems. The Software PLC Internal I/O Assignment configures the correspondence between the robot's input and output signals (DI/DO, GI/GO, UI/UO) and the Software PLC Input/Output area.

The Software PLC Internal I/O Assignment is configured on the Software PLC Internal I/O Assignment screen (MENU → "I/O" → F1" [TYPE]" → "Software PLC" → F2"[DATA] " → "Ext. I/O asg").

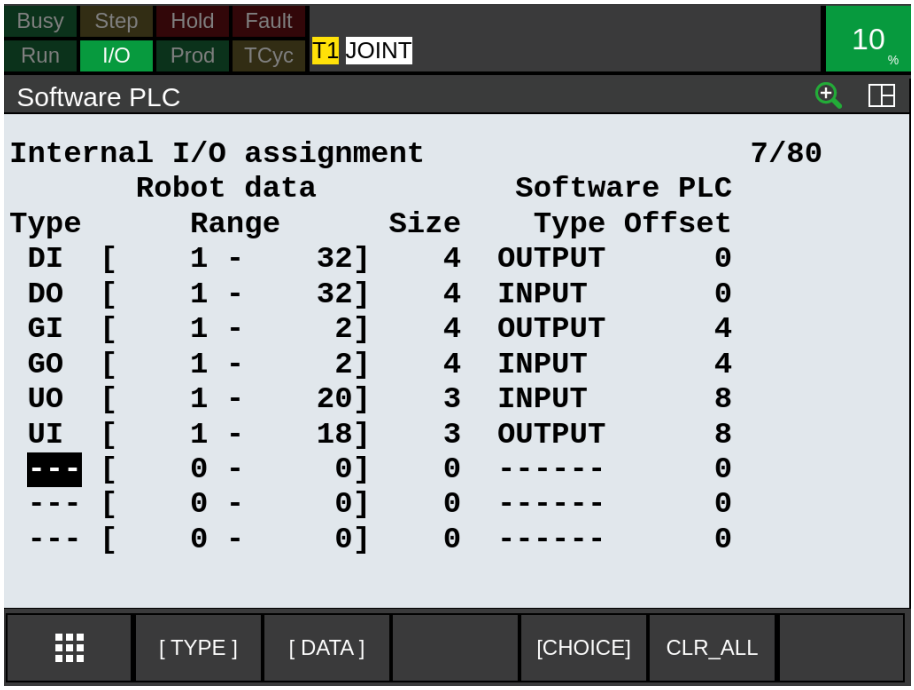


Fig. 60.3.3 (a) Software PLC Internal I/O Assignment screen

- In the "Type" column of the "Robot data" set the data type. For details on data types, refer to " Table 60.3.3 (a) Data types for Software PLC Internal I/O Assignment". The "Type" for the Software PLC is automatically set according to the signal type specified in the "Type" of "Robot data".
- Specify the start number in the "Range" column of the "Robot data" and the "Size" column. Then, set the offset (Unit: Byte) of the Software PLC Input/Output area in the "Offset" column of the "Software PLC" to configure the Software PLC Internal I/O Assignment.
- The end number of the "Range" column of the "Robot data" is automatically determined by the specified the "Size" and cannot be changed. If the "Size" is 0, the end number of the "Range" is displayed as 0.
- The maximum number of setting lines is 80.
- Pressing F5 [CLR_ALL] removes all settings on the Software PLC Internal I/O Assignment screen.
- The Software PLC Internal I/O Assignment settings will take effect after cycle power.

For example, if configured as follows, 32 signals of the robot's DI[1-32] will be assigned starting from the 0th byte of the Software PLC Output area.

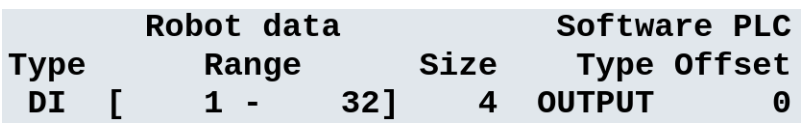


Fig. 60.3.3 (b) Example of the setting the Software PLC Internal I/O Assignment

The types of robot data available for use in the Software PLC Internal I/O Assignment are as follows.

Table 60.3.3 (a) Data types for Software PLC Internal I/O Assignment

Robot data	Software PLC Type	Description
DO	INPUT	Assign the robot's DO to the Software PLC Input area. You can transfer bit values from the robot to the Software PLC.
DI	OUTPUT	Assign the robot's DI to the Software PLC Output area. You can transfer bit values from the Software PLC to the robot.
GO	INPUT	Assign the robot's GO to the Software PLC Input area. You can transfer 2 bytes values from the robot to the Software PLC.
GI	OUTPUT	Assign the robot's GI to the Software PLC Output area. You can transfer 2 bytes values from the Software PLC to the robot.
UO	INPUT	Assign the robot's UO to the Software PLC Input area. You can transfer UOP outputs such as CMDENBL from the robot to the Software PLC.
UI	OUTPUT	Assign the robot's UI to the Software PLC Output area. You can transfer UOP inputs such as START from the Software PLC to the robot.

The specifications for the Software PLC Internal I/O Assignment are as follows.

- You cannot specify data outside the range of the robot's input and output signals (DI/DO, GI/GO, UI/UO) in "Robot data".
- You cannot specify robot data that is already set in another line of the Software PLC Internal I/O Assignment.
- Robot's input and output signals (DI/DO, GI/GO, UI/UO) assigned in I/O Assignment (Robot) cannot be specified in the "Robot data" of the Software PLC Internal I/O Assignment.
- The start number of the "Range" of the "Robot data" and "Size" must be set to values other than 0.
- GI (Group Input) and GO (Group Output) must have a "Size" that is a multiple of 2.
- The "Offset" of the "Software PLC" must be a multiple of 2.
- For lines where the "Offset" of the Software PLC is less than 128 bytes, the sum of the "Size" and the "Offset" of the "Software PLC" within that line must be within 128 bytes.
- The sum of the "Size" and the "Offset" of the "Software PLC" within one line must be within 256 bytes.
- You cannot specify the Software PLC Input/Output area that is already set in another line on the Software PLC External I/O Assignment screen or the Software PLC Internal I/O Assignment screen.

60.3.3.1 Example of Software PLC Internal I/O Assignment Configuration

Here is an example of setting the Software PLC Internal I/O Assignment for using the Software PLC with the following configuration.

- Assign DO[1-32] from the second byte (InputWord[2~3]) of the Software PLC Input area.
- Assign GI[1-2] from the second byte (OutputWord[2~3]) of the Software PLC Output area.
- Map variables to the second byte (Bit0~15 of InputWord[2], Bit0~15 of InputWord[3]) of the Software PLC Input area to receive the value of DO in the PLC program.
- Map variables from the beginning (OutputWord[2~3]) of the Software PLC Output area to output from the PLC program to GI.

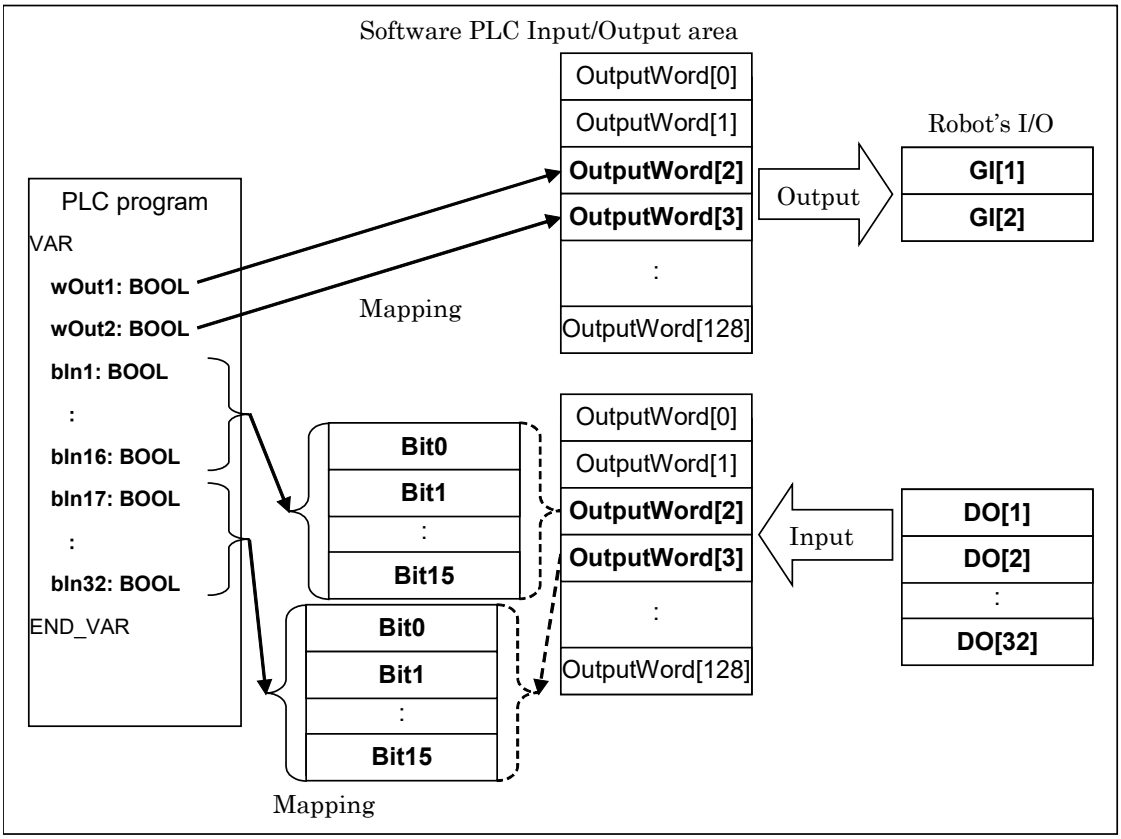


Fig. 60.3.3.1 (a) Example of setting the Software PLC Internal I/O Assignment

Configure as follows.

Software PLC Internal I/O Assignment screen

```
        Robot data                Software PLC
Type        Range          Size     Type Offset
 DO   [     1 -    32]     4   INPUT        2
 GI   [     1 -     2]     4   OUTPUT       2
```

Fig. 60.3.3.1 (b) Example of setting the Software PLC Internal I/O Assignment

I/O Mapping in the CODESYS Development System

Variable	Mapping	Channel	Address	Type
		InputWord	%IW0	ARRAY [0..63] OF WORD
		InputWord[0]	%IW0	WORD
		InputWord[1]	%IW1	WORD
		InputWord[2]	%IW2	WORD
Application.PLC_PRG.bIn1		Bit0	%IX4.0	BOOL
Application.PLC_PRG.bIn2		Bit1	%IX4.1	BOOL
Application.PLC_PRG.bIn3		Bit2	%IX4.2	BOOL
Application.PLC_PRG.bIn4		Bit3	%IX4.3	BOOL
Application.PLC_PRG.bIn5		Bit4	%IX4.4	BOOL
Application.PLC_PRG.bIn6		Bit5	%IX4.5	BOOL
Application.PLC_PRG.bIn7		Bit6	%IX4.6	BOOL
Application.PLC_PRG.bIn8		Bit7	%IX4.7	BOOL
Application.PLC_PRG.bIn9		Bit8	%IX5.0	BOOL
Application.PLC_PRG.bIn10		Bit9	%IX5.1	BOOL
Application.PLC_PRG.bIn11		Bit10	%IX5.2	BOOL
Application.PLC_PRG.bIn12		Bit11	%IX5.3	BOOL
Application.PLC_PRG.bIn13		Bit12	%IX5.4	BOOL
Application.PLC_PRG.bIn14		Bit13	%IX5.5	BOOL
Application.PLC_PRG.bIn15		Bit14	%IX5.6	BOOL
Application.PLC_PRG.bIn16		Bit15	%IX5.7	BOOL
		InputWord[3]	%IW3	WORD
Application.PLC_PRG.bIn17		Bit0	%IX6.0	BOOL
Application.PLC_PRG.bIn18		Bit1	%IX6.1	BOOL
Application.PLC_PRG.bIn19		Bit2	%IX6.2	BOOL
Application.PLC_PRG.bIn20		Bit3	%IX6.3	BOOL
Application.PLC_PRG.bIn21		Bit4	%IX6.4	BOOL
Application.PLC_PRG.bIn22		Bit5	%IX6.5	BOOL
Application.PLC_PRG.bIn23		Bit6	%IX6.6	BOOL
Application.PLC_PRG.bIn24		Bit7	%IX6.7	BOOL
Application.PLC_PRG.bIn25		Bit8	%IX7.0	BOOL
Application.PLC_PRG.bIn26		Bit9	%IX7.1	BOOL
Application.PLC_PRG.bIn27		Bit10	%IX7.2	BOOL
Application.PLC_PRG.bIn28		Bit11	%IX7.3	BOOL
Application.PLC_PRG.bIn29		Bit12	%IX7.4	BOOL
Application.PLC_PRG.bIn30		Bit13	%IX7.5	BOOL
Application.PLC_PRG.bIn31		Bit14	%IX7.6	BOOL
Application.PLC_PRG.bIn32		Bit15	%IX7.7	BOOL
		InputWord[4]	%IW4	WORD

Fig. 60.3.3.1 (c) Example of Input Mapping

Variable	Mapping	Channel	Address	Type
		OutputWord	%QW0	ARRAY [0..63] OF WORD
		OutputWord[0]	%QW0	WORD
		OutputWord[1]	%QW1	WORD
Application.PLC_PRG.wOut1		OutputWord[2]	%QW2	WORD
Application.PLC_PRG.wOut2		OutputWord[3]	%QW3	WORD
		OutputWord[4]	%QW4	WORD

Fig. 60.3.3.1 (d) Example of Output Mapping

For I/O mapping in the CODESYS Development System, refer to "60.4.3 Addition of I/O devices for the FANUC robot and the I/O mapping".

60.4 How to use the CODESYS Development System

Use the CODESYS Development System to create PLC programs.

Refer to the following for how to use the CODESYS Development System.
https://www.helpme-codesys.com/codesys-engineering/codesys-development-system.html

The Software PLC supports the following functions.
- Connecting to the CODESYS Development System (Login/Logout), downloading, and executing PLC programs.
- Monitoring variables and task cycle time.
- Program verification functions: writing arbitrary values to variables, breakpoints and sampling variable values.

The Software PLC does not support PLC initialization (Menu bar → "Online" → "Reset Origin"). If you have done "Login" or "Create Boot Application" on the robot controller even once, the most recently updated boot application will automatically start after cycle power, even if you did "Reset Origin".

From here on, the specific development procedures using the CODESYS Development System are explained based on an example.

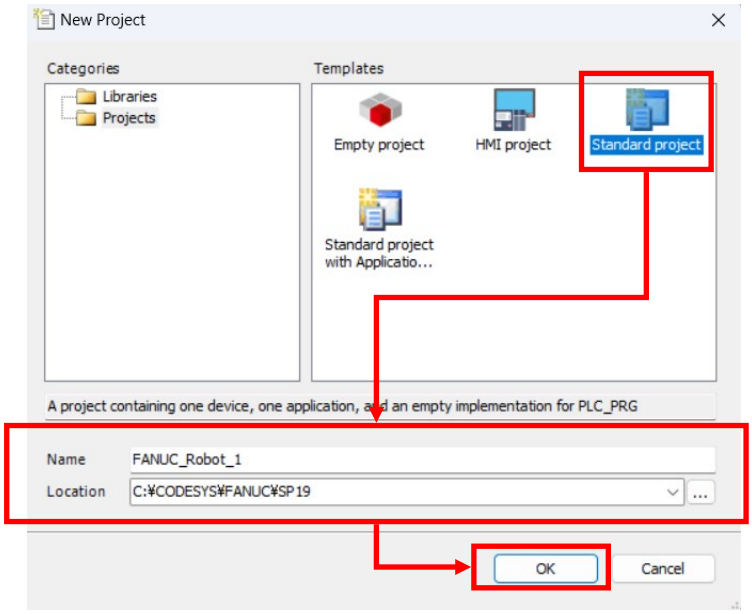
60.4.1 Creating a Project for the Software PLC

The steps for creating a project for the Software PLC are as follows.

1. Launch the CODESYS Integrated Development Environment.

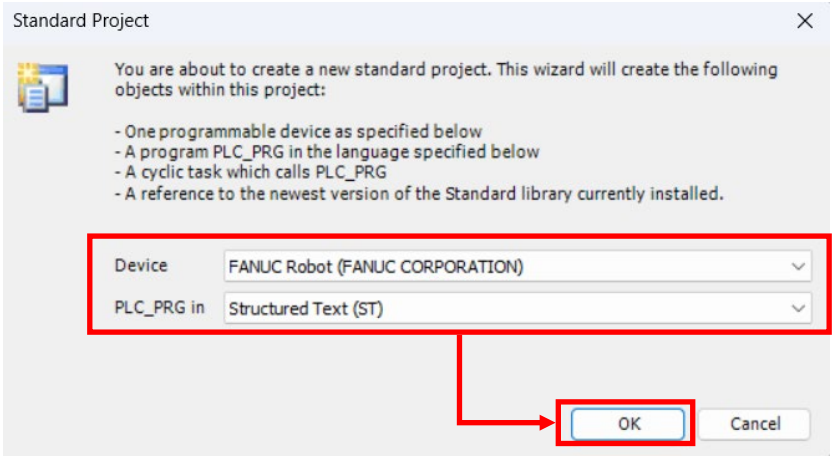
2. Select "File" → "New Project" from the menu bar.

3. Select "Standard project" and enter any project name. Also, specify any project save location.
   In this example, specify as follows.
   Name: FANUC_Robot_1
   Location: C:\CODESYS\FANUC\SP19



4. Select "FANUC Robot (FANUC CORPORATION) " for the device.
   PLC_PRG is a PLC program automatically created when the project is created. Select the programming language to be used in PLC_PRG. After creating the project, you can create a new PLC program in a different language from the one specified here.
   In this example, specify as follows.

PLC_PLG in: Structured Text (ST)



This completes the creation of the Software PLC project.

60.4.2 PLC programming and Task assignment

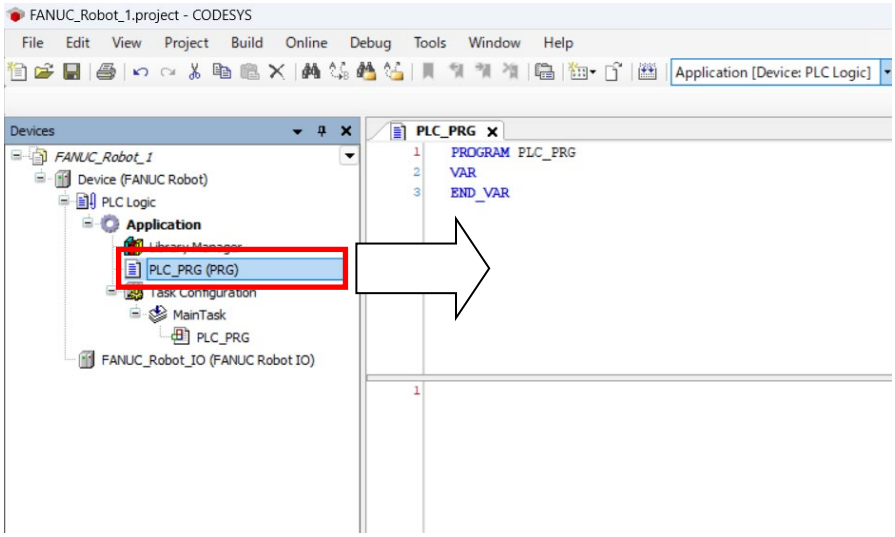
To execute sequence control with a robot control device, PLC programming and the assignment of the created PLC program to tasks are necessary. In tasks, Priority and Interval settings are configured.

PLC programs are written in units called " POU (Program Organization Unit) ". Only one language can be used in a single POU.

There are three types of POUs: "Program", "Function", and "Function block". Programs can be executed by assigning them to tasks. Functions and Function blocks serve as subroutines. Function blocks have internal retention and multiple output capabilities.

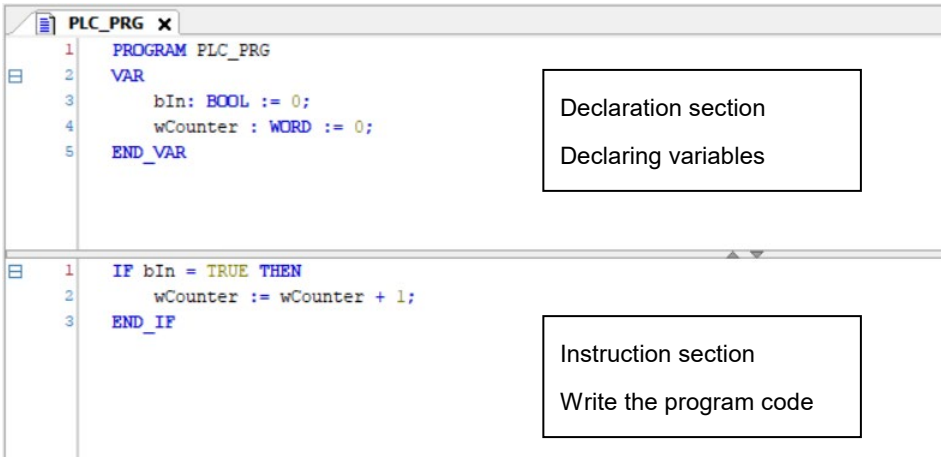
The procedure for PLC programming and Task assignment is as follows.

1. Double-click the POU in the device tree to open the program language editor.
   The figure below shows the program language editor for Structured Text (ST).



2. Program using the programming language set in the opened POU.
   The program language editor consists of a "Declaration" section and an "Instruction" section.

In the Declaration section, variables are declared, and programming is done in the Instruction section.

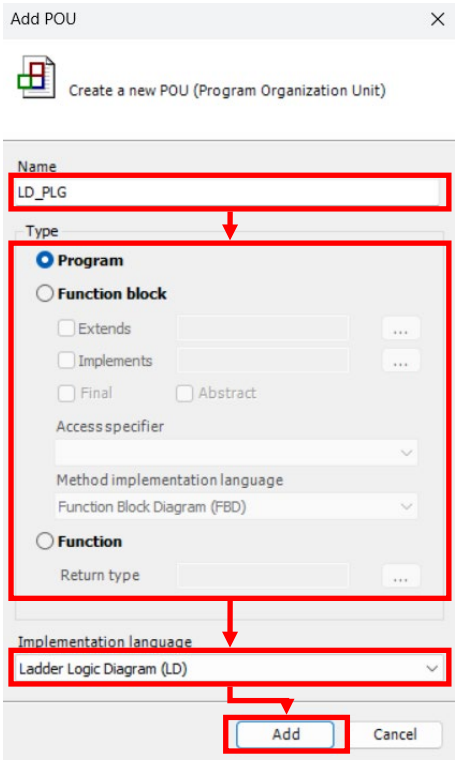


3. Once completed, select "Build" → "Generate Code" from the menu bar to build.
   The build results will be displayed in the message window as shown below.
   If build errors occur, check the messages and take appropriate action.

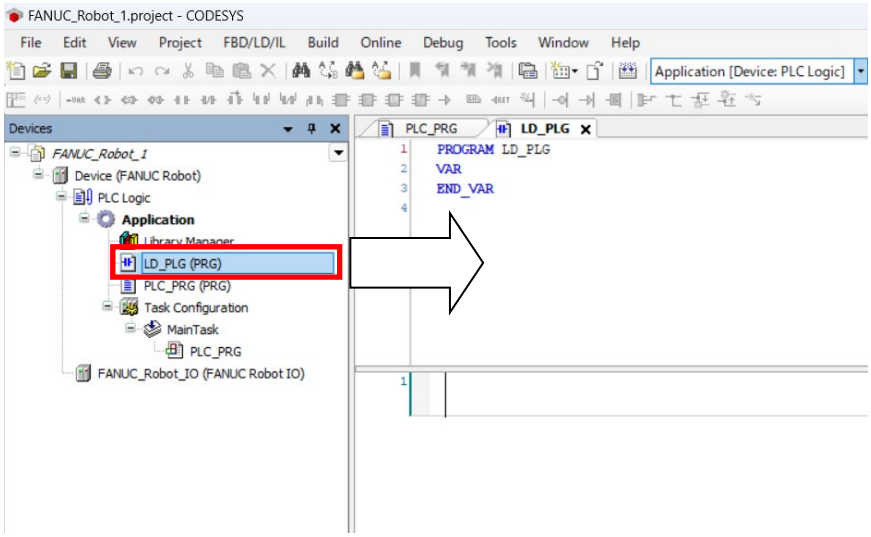
Multiple programs can be created within a single project.

4. If multiple programs are needed, add a new POU.
   For example, you can add a POU for Ladder Logic Diagram (LD), which is different from the Structured Text (ST) mentioned above.
   Right-click on "Application" in the device tree, then select "Add Object" → "POU".

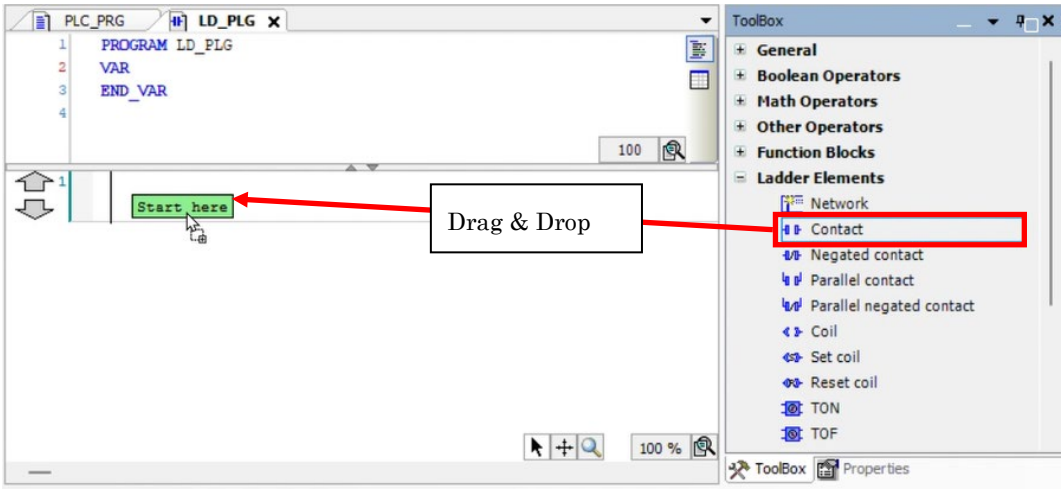
5. In the "Add POU" window, specify the POU Name, Type, and Implementation language.
   In this example, specify as follows.
   Name: LD_PRG
   Type: Program
   Implementation language: Ladder Logic Diagram (LD)



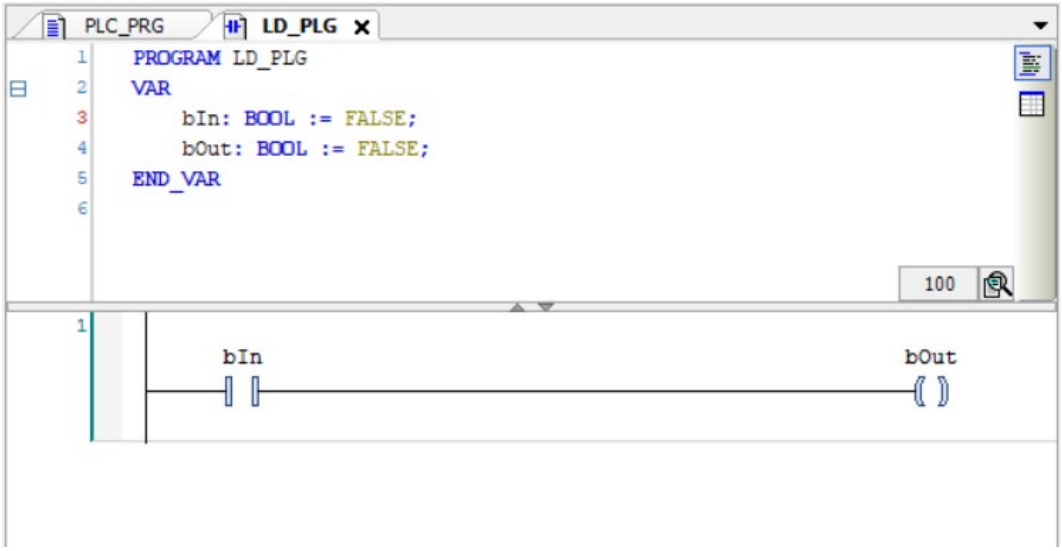
6. "LD_PRG" is added to the device tree. Double-click to open the LD editor.



7. In Ladder Logic Diagram (LD), you create programs by dragging and dropping components from the ToolBox into the Instruction section and assigning variables or values to the components. Similar programming can be done with FBD, SFC, and CFC.



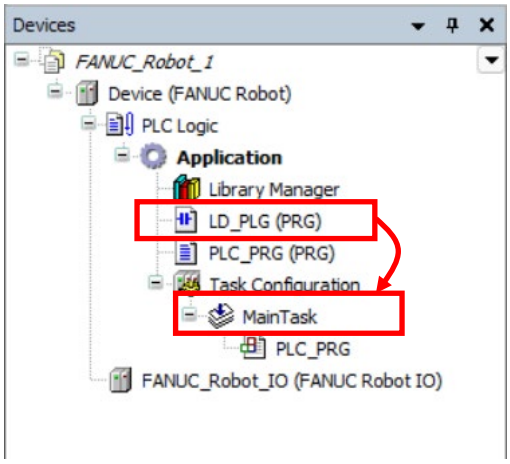
プログラム完成後:



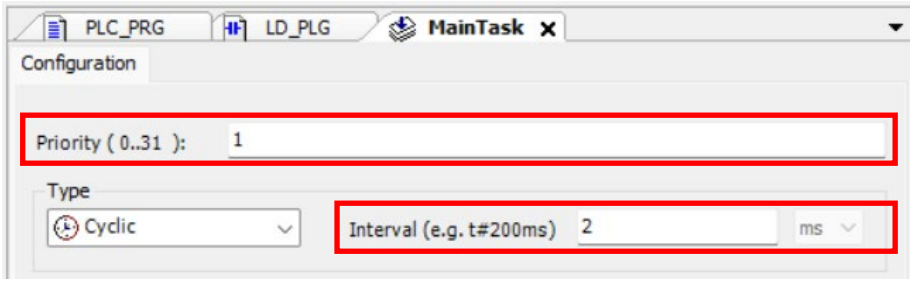
8. After completing the program, select "Build" → "Generate Code" from the menu bar to build.
   Ensure there are no build errors.

9. Assign each POU to a task.
   Newly added POUs can be assigned to tasks by dragging and dropping them into the task in the device tree.
   The automatically created PLC_PRG is assigned to MainTask from the beginning when the project is created.

If you want to add a new task, right-click on "Task Configuration" in the device tree, then select "Add Object" → "Task".



10. Configure the task settings.
    Double-click the task to be configured in the device tree to display the task settings editor.
    Set the task's Priority and Interval.
    The Priority can be specified in the range of 1 to 32.
    The Interval can be specified in the range of 2ms to 100000ms (100s). The resolution is 1ms.



This completes the PLC programming and Task assignment.

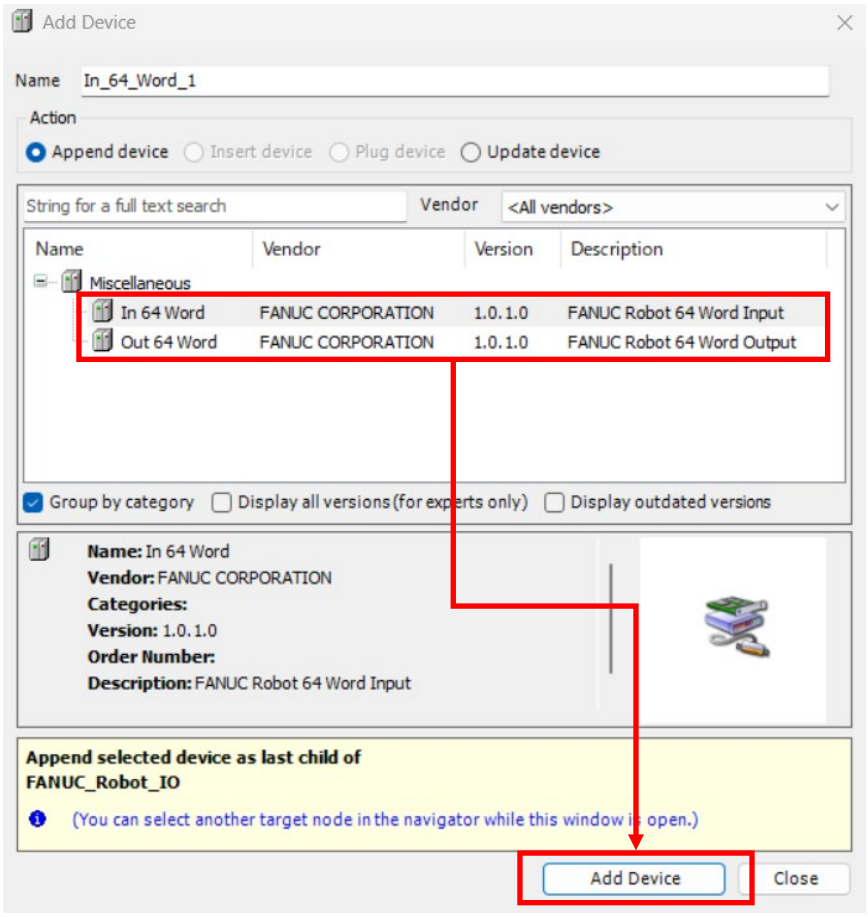
60.4.3 Addition of I/O devices for the FANUC robot and the I/O mapping

To enable the Software PLC to input and output signals with external I/O devices or the robot's DI/DO, add the Software PLC Input/Output area. For details on the Software PLC Input/Output area, refer to "60.3.1 Software PLC Input/Output area". Specifically, add the device that serves as the Software PLC Input/Output area under FANUC Robot IO.

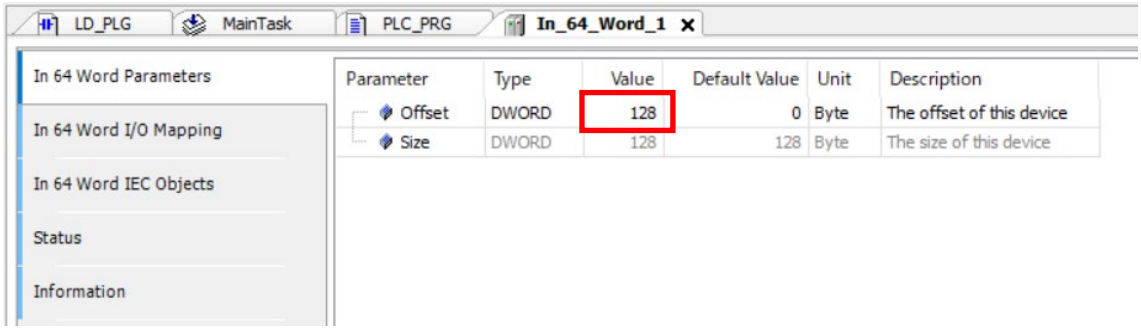
The procedure for adding I/O devices for the FANUC robot and configuring I/O mapping is as follows.

1. Right-click on FANUC_Robot_IO (FANUC Robot IO) in the device tree and select "Add Device".

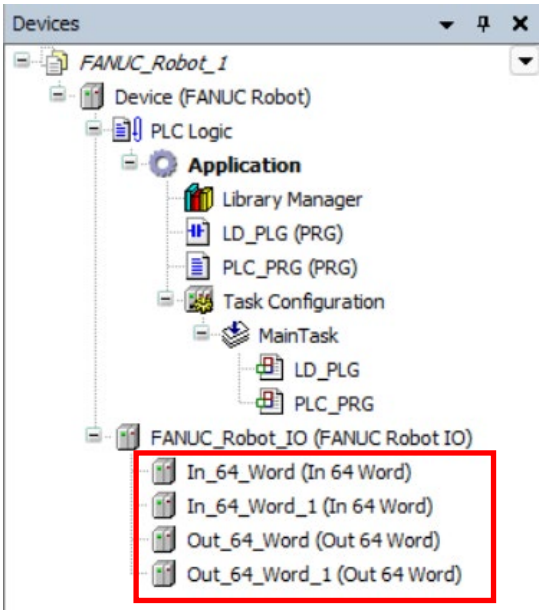
2. In the "Add Device" window, add "In 64 Word" and "Out 64 Word".



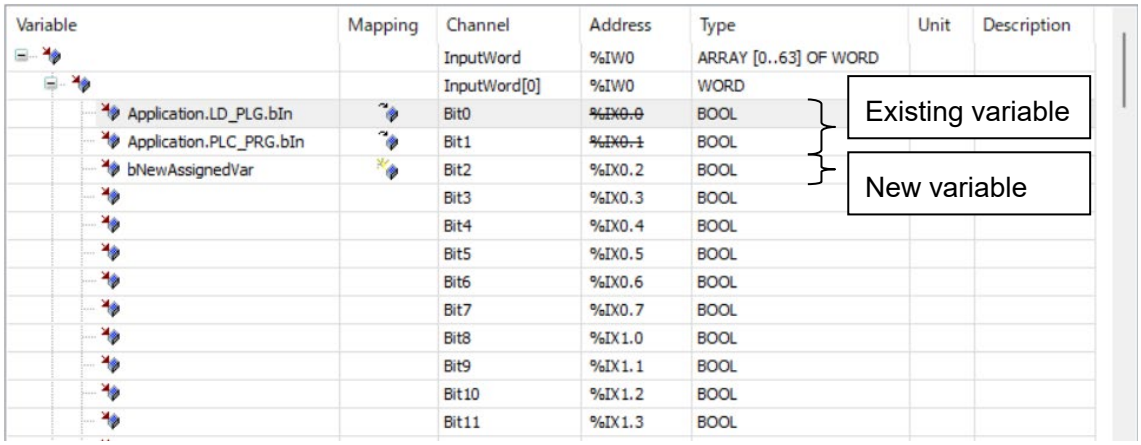
3. If you need to use more than 64 Word (128 bytes) for the Software PLC Input/Output area, add another In 64 Word and Out 64 Word device. After adding, double-click the device (second device) assigned to the Software PLC Input/Output area beyond the 128th byte to open the device editor. Set the value of "Offset", which is a "Parameter" in the editor, to 128.
   This means that this device will be assigned from a position offset by 128 bytes from the beginning of the Software PLC input (or output) area.



4. The addition of I/O devices for FANUC robots is complete.
   If both the Software PLC input area and the Software PLC output area use the maximum of 256 bytes, it looks like the figure below.



5. Double-click the In 64 Word (Robot → Software PLC) or Out 64 Word (Software PLC → Robot) and set up the I/O mapping. Enter the variable names to be mapped to each channel.
   You can map either "existing variable" defined in the already created PLC program or "new variable" defined directly within the I/O mapping.
   For existing variables, double-click the channel you want to map, then click the … button to display the "Input Assistant". From there, you can select the variable you want to map. When mapping existing variables, the type of the variable to be mapped must match the "Type" of the target channel.
   For new variables, directly enter the variable name in the channel. New variables are defined as global variables based on the channel type. Global variables have a scope that spans the entire project.



This completes the addition of I/O devices for the FANUC robot and the I/O mapping.

However, to actually execute input or output between the Software PLC and external I/O devices or robot's signals such as DI/DO, it is necessary to configure the Software PLC External I/O Assignment or the Software PLC Internal I/O Assignment on the teach pendant according to "60.3 I/O assignment for The Software PLC".

60.4.4 Login/Logout to the robot control device, download and start/stop the PLC program.

After completing "60.4.2 PLC programming and Task assignment" and "60.4.3 Addition of I/O devices for the FANUC robot and the I/O mapping", download the completed PLC program to the robot controller and execute it.

To download the PLC program, it is necessary to establish communication between the CODESYS Development System and the robot controller (Login).
For the CODESYS Development System to communicate with the robot controller, the PC with the CODESYS Development System and the robot controller must be on the same network. This requires proper Ethernet cable connection between both nodes and pre-configuring the IP address and other settings on the robot controller. For Ethernet connection settings, refer to "Setting up TCP/IP" in the "Network and Security Function OPERATOR'S MANUAL" (B-84604EN).

The PLC program is downloaded to the robot controller as an "Application".
The application is compiled to generate code, including information such as POU and task settings, and I/O mapping.

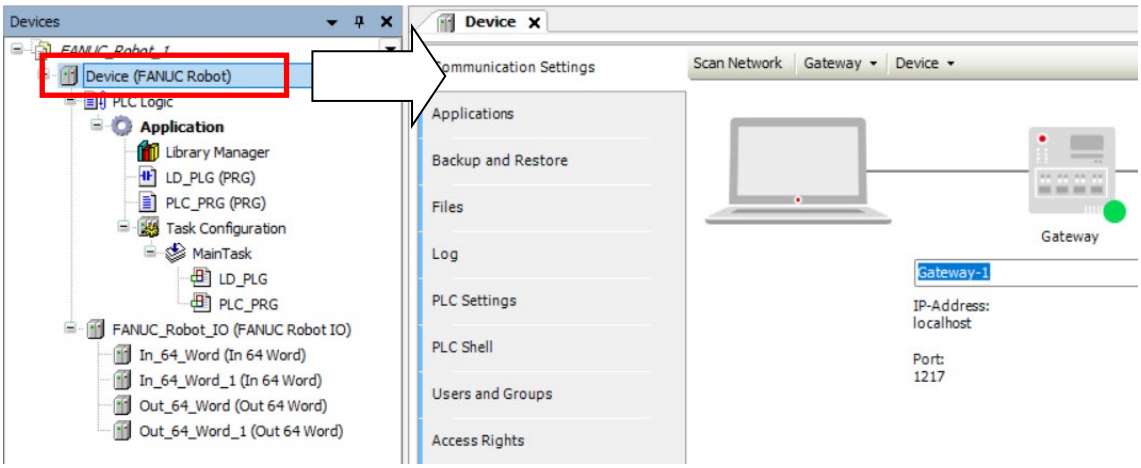
After downloading the application, you can start or stop the PLC program from the CODESYS Development System. Additionally, even if the CODESYS Development System is not communicating with the robot controller, the boot application will automatically start when the robot controller is powered on.

The applications used in production need to automatically start the program when the robot controller is powered on. If you want the program to start automatically when the robot controller is powered on, you need to set up the boot application.
When you download an application for the first time to the robot controller that does not have an existing application, the boot application is automatically generated.
Update the boot application using the following steps.

- When you log in to the robot controller after modifying the application, check "Update boot application" in the displayed dialog and download the application.
- From the menu bar, select "Online" → "Create Boot Application".

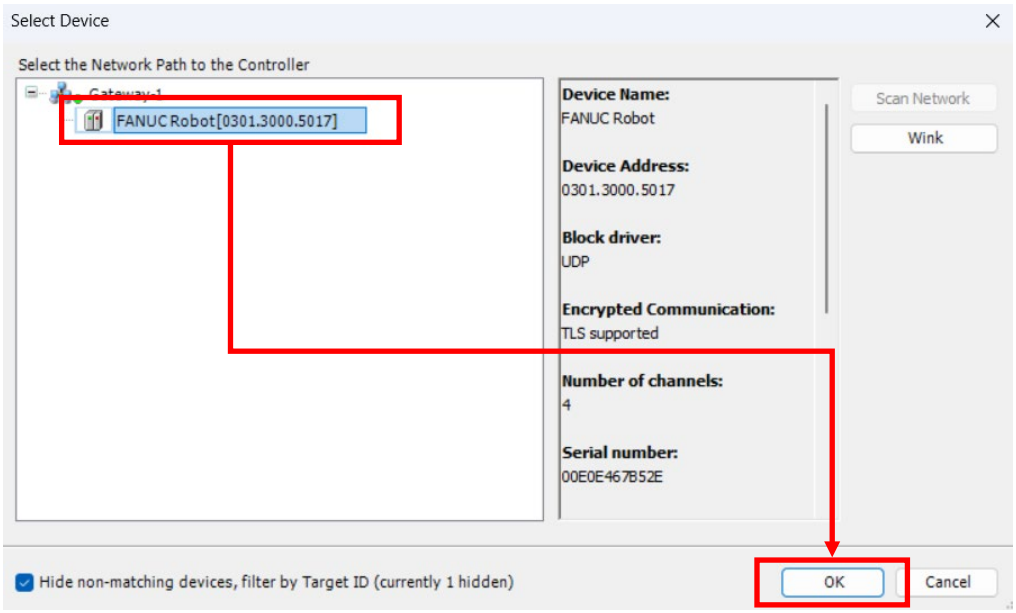
The steps for log in to the robot controller, downloading the PLC program, starting/stopping the PLC program, and logout from the robot controller are as follows.

1. Double-click the Device (FANUC Robot) in the device tree to open the generic device editor.



2. Select "Scan Network" to display the "Select Device" window, where "FANUC Robot" will appear. Choose "FANUC Robot". If "FANUC Robot" does not appear, it is possible that the PC with the CODESYS Development System and the robot controller are not on the same network, or the IP address settings of the robot controller are incorrect. Please check the

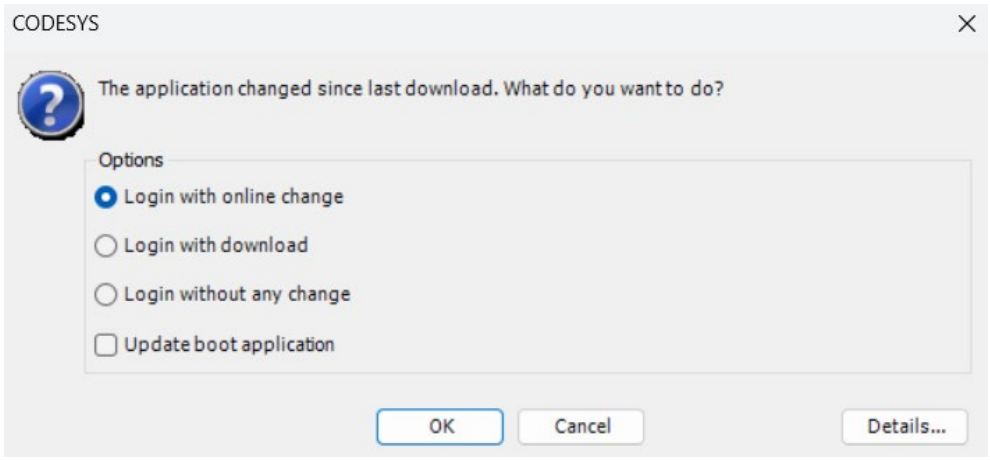
network configuration again to ensure it is correct.



3. Press the "Login" button on the menu bar to log in to the robot controller. The application is automatically downloaded to the robot controller upon the login.

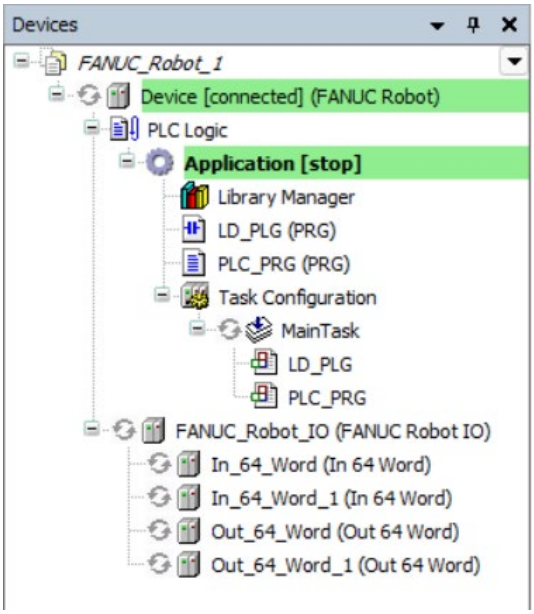
   If the application does not exist on the robot controller, such as at the first login or after doing "Reset Origin", a message is displayed. Select "Yes (Y) " to proceed with the download.

   If the application has already been downloaded to the robot controller, it checks whether an online change is possible.
   If the changes are minor and an online change is possible, the following message will be displayed.
   If you select "Login with online change", the online change will be executed. This allows you to modify the program code and variables without interrupting the running PLC program.
   If you select "Login with download", the PLC program starts from its initial state after the download.

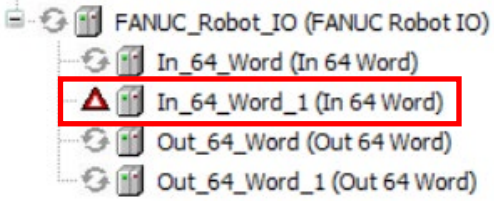


   If the changes are significant and an online change is not possible, the program starts from its initial state after the download.

4. The application is downloaded to the robot controller.



If a warning (red triangle) appears on a device displayed in the device tree, the "Offset" value in the device parameter may be incorrect.
Ensure that the "Offset" is either 0 or 128, and that the "Offset" values are not the same for two identical devices.



5. If the PLC program is stopped, [stop] is displayed next to the "Application" in the device tree.
   You can start the program by pressing the "Start" button on the menu bar.

6. If the PLC program is running, [run] is displayed next to the "Application" in the device tree. You can stop the program by pressing the "Stop" button on the menu bar.

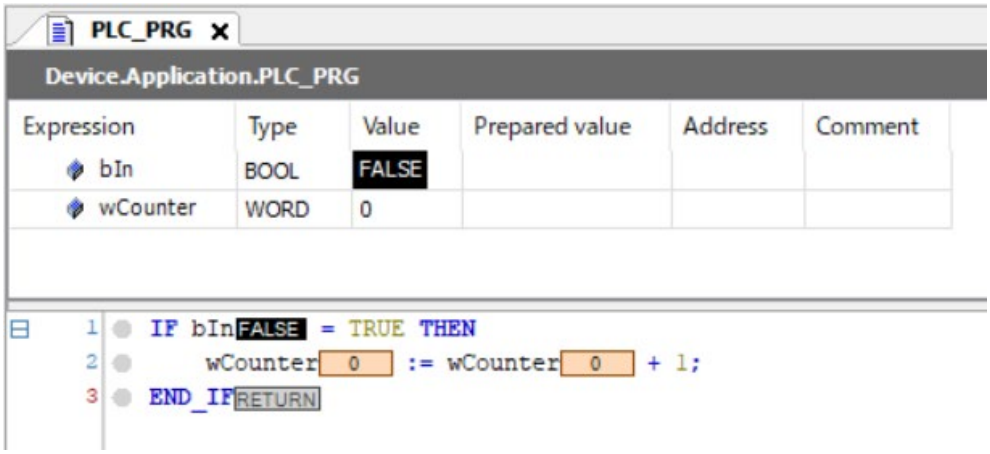
7. To end communication between the CODESYS Development System and the robot controller, press the "Logout" button on the menu bar.

60.4.5 Monitoring the PLC Program

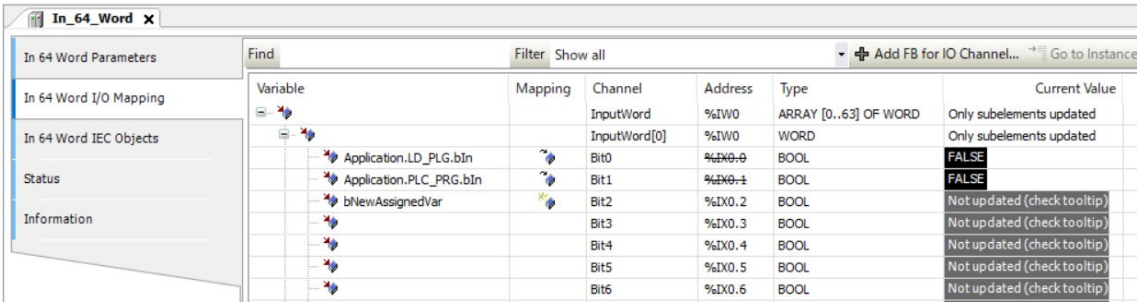
When the CODESYS Development System is logged into the robot controller, the following can be monitored in real-time.

Values of variables used in the PLC program

Displayed in the program language editor of the running PLC program.

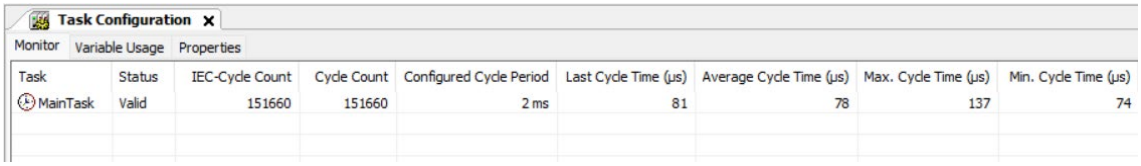


You can also refer to the values of variables mapped to I/O map from the I/O Mapping tab of the I/O device.



Actual cycle time of the task

It is displayed in the task configuration window, which appears by double-clicking "Task Configuration" in the device tree.



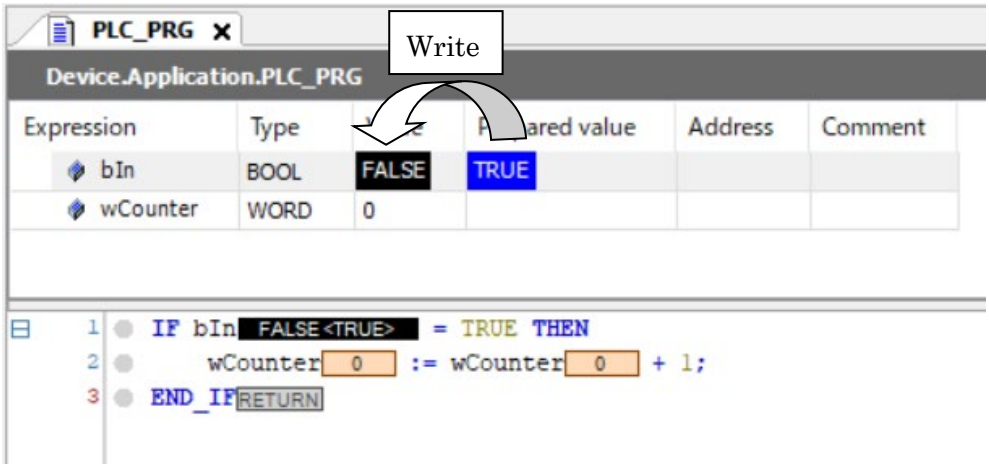
60.4.6 Verification of the PLC program

When the CODESYS Development System is logged into the robot controller, the following verification functions are available.

Write an arbitrary value to a variable used in the PLC program.

You can write an arbitrary value to a variable used in the running PLC program from the program language editor. Enter the value you want to write in the "Prepared value" field, then select "Debug" → "Write Values" from the menu bar to write the set

value.

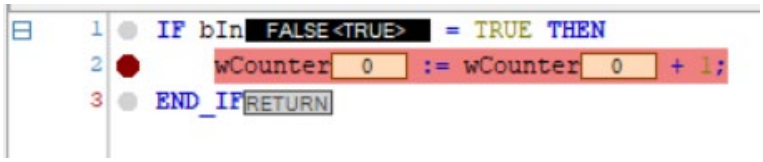


In the above method, the written value may be overwritten by the actual input or output signal values since the write operation is executed only once. If you want to fix the value to an arbitrary value, select "Debug" → "Force Values" from the menu bar.

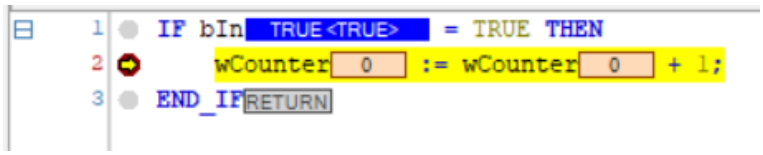
Using Breakpoints

You can set breakpoints at specific locations in the program to forcibly stop execution and monitor variable values.
It is also possible to set data breakpoints to stop program execution when the value of a specific variable changes.
Breakpoints can be used in all IEC language editors.

Right-click on the line where you want to place a breakpoint and select "Toggle Breakpoint".



When the actual processing reaches a breakpoint, execution stops. The line where execution is halted will be highlighted in yellow.



While at a breakpoint, you can do step execution using commands such as "Step Over", "Step Into", and "Step Out" from the "Debug" in the menu bar.

Sampling variable values and Displaying graphs

It is possible to sample the current value of a specified variable over time and plot it on a graph for display.
Please refer to the link below for usage instructions.
https://content.helpme-codesys.com/en/CODESYS%20Trace/_trace_start_page.html

60.5 Saving and Loading Software PLC related data

The PLC application and the settings in the Software PLC screen in the robot controller can be saved as files to an external file device.

By loading these files, you can restore the PLC application and the settings in the Software PLC screen to the state they were in when the files were saved.
By loading these files, you can also copy the PLC application and the settings in the Software PLC screen to other robot controllers.

Files to be saved

File name	Contents
SOFTPLC.DAT	PLC application PLC programs Task settings I/O mapping *1 The boot application is backed up. If online changes are made without updating the boot application, those changes are reflected in the backup. To reflect online changes in the backup, log in to the controller and update the boot application by selecting "Online" → "Create Boot Application" from the menu bar in the CODESYS Development System. *2 This file is included in the "System files" section of the file screen. *3 Cycle power is required to start the restored PLC application.
PLCCFG.SV	The settings of the Software PLC screen. Software PLC external I/O assignment Software PLC internal I/O assignment

Save files

These files can be saved from the file screen on the teaching pendant.
For instructions on how to save files, refer to the "SAVING FILES" section in the "OPERATOR'S MANUAL (Basic Function)" (B-84464EN).

Load files

These files can be loaded from the file screen on the teaching pendant.
For instructions on how to load files, refer to the "LODING FILES" section in the "OPERATOR'S MANUAL (Basic Function)" (B-84464EN).

61 DIGITAL I/O JOG FUNCTION

61.1 OVERVIEW

Digital I/O Jog Function (J543) is the function that can do jog operation according to external digital signal input.

61.2 SPECIFICATION

61.2.1 Setup

Settings for this function can be made using the Digital I/O Jog Function on the System Setting screen.

Table 61.2.2 (a) System config screen

item	explaination
Digital I/O Jog Function	Set enable/disable of function.
Group/Ext setting	0: Specify to use axis 7 to axis 9 as J7 to J9 of Group1. 1: Specify to use axis 7 to axis 9 as J1 to J3 of Group2. (For multi group system.) 2: Select the group to jog using Group signal Input (GI). (For multi group system.)
DI index Setting	Specify the index of Digital signal Input (DI). (For actual usage, please refer to Table 61.2.2(b).)
GI index	Specify the index of GI which is used to specify group number in case of Group/Ext setting is 2.

The meaning of each number in the DI index Setting is as shown in the table below.
The setup in below table shows the example in case that specify DI[41] to DI[66] for DI jog.

Table 61.2.2(b) DI index Setting

Number of DI index Setting	The role of DI signals	DI index to use DI jog (Example)	Value (Example)
1	Enable	DI[41]	41
2	Specify JOINT Frame	DI[42]	42
3	Specify JOG Frame	DI[43]	43
4	Specify JOG Frame 1	DI[44]	44
5	Specify JOG Frame 2	DI[45]	45
6	Specify JOG Frame 3	DI[46]	46
7	Specify JOG Frame 4	DI[47]	47
8	Specify JOG Frame 5	DI[48]	48
9	J1+/X+	DI[49]	49
10	J1-/X-	DI[50]	50
11	J2+/Y+	DI[51]	51
12	J2-/Y-	DI[52]	52
13	J3+/Z+	DI[53]	53
14	J3-/Z-	DI[54]	54
15	J4+/W+	DI[55]	55
16	J4-/W-	DI[56]	56
17	J5+/P+	DI[57]	57
18	J5-/P-	DI[58]	58
19	J6+/R+	DI[59]	59
20	J6-/R-	DI[60]	60
21	Axis 7+	DI[61]	61

Number of DI index Setting	The role of DI signals	DI index to use DI jog (Example)	Value (Example)
22	Axis 7-	DI[62]	62
23	Axis 8+	DI[63]	63
24	Axis 8-	DI[64]	64
25	Axis 9+	DI[65]	65
26	Axis 9-	DI[66]	66

61.2.2 Limitations

- Digital I/O jog enables jog when below conditions are all satisfied.
  - Enable signal ON
  - The remote conditions are satisfied
  - Digital I/O jog function is enabled on the System Config screen

- Jog will stop if both "Specify JOINT Frame DI" and "Specify JOG Frame DI" are turned ON or turned OFF at the same time.

Specify JOINT Frame DI	Specify JOG Frame DI	Selected Frame
OFF	OFF	Jog is stopped
ON	OFF	JOINT Frame
OFF	ON	JOG Frame
ON	ON	Jog is stopped

- When turning ON the "Specify JOG Frame" signal, jogging will be stopped in any combination other than the following. For example, if both "JOG Frame 1" and "JOG Frame 2" are turned ON at the same time, jogging will stop.

Specify JOG Frame 1	Specify JOG Frame 2	Specify JOG Frame 3	Specify JOG Frame 4	Selected Frame
OFF	OFF	OFF	OFF	Jog is stopped
ON	OFF	OFF	OFF	JOG Frame 1
OFF	ON	OFF	OFF	JOG Frame 2
OFF	OFF	ON	OFF	JOG Frame 3
OFF	OFF	OFF	ON	JOG Frame 4

- When external start was input during DI jogging, jog motion will be stopped. In this case, external start will be ignored.

- If jog command more than two directions are input at the same time, robot will move to all specified directions.
  Example: If +X direction and +Y direction were input at the same time, robot will move to +X and +Y at the same time.

- If JOG key was pressed during DI jogging, jog will be stopped.

- When DI jog was ON during TP is disabled and OVR100%, robot moving speed will be limited less than 250mm/sec.

62 PALLETIZING MODE

62.1 Overview

The Palletizing Mode is a function that makes it possible to enables Cartesian motion that passes through the wrist singularity (J5 = 0 [deg]), which is normally impossible for 6-axis robots.

This function is available during Cartesian motion. When using it, the wrist axis must be downwards in WORLD coordinate system (*1), in other words, the flange surface must be strictly parallel to the X-Y plane of WORLD coordinate system, and the J4 axis must be 0 [deg] (*2). Joint motion can be used without any limitations.

This function can be used for TP program execution, jogging, and Manual Guided teaching. Enable/disable in TP program execution and in jogging/Manual Guided teaching must be set independently.

The Palletizing Mode option (S575) is required to use this function. It is installed as standard on the CRX series.

*1) Can also be used with the wrist axis be upwards (ceiling mount).
*2) Can also be used with J4 = +180 / -180 [deg].

62.2 TP program execution

Enable/disable of Palletizing Mode in TP program execution is set for each program. If you create a new TP program, load a TP program that was backed up before adding the Palletizing Mode option, or on a system that does not have the Palletizing Mode option, the Palletizing Mode will be disabled for those programs by default.

62.2.1 Enable/Disable setting

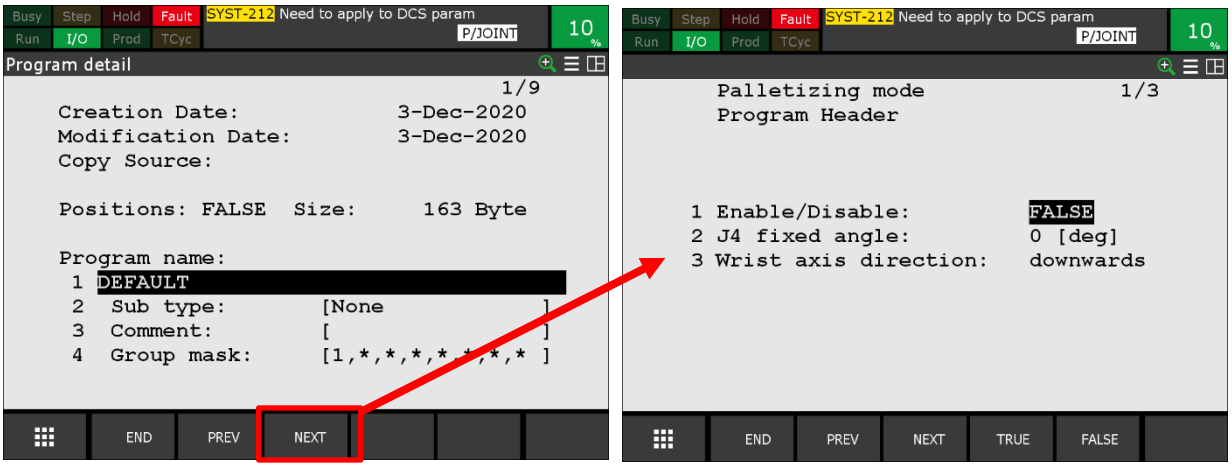
Enable/disable settings are made in the application header settings on the TP program details screen. Set the following items in the application header for Palletizing mode.

ITEM	CHOICE	EXPLANATION
Enable/Disable	Disable	The robot cannot move with Cartesian motion near the wrist singularity. There are no restrictions on wrist posture during Cartesian motion.
	Enable	The robot can move with Cartesian motion even near the wrist singularity. All Cartesian motions have restricted wrist posture.
J4 fixed angle (J4 TURN)	0 [deg]	
	+180 [deg]	Mainly useful for models that have an offset on the wrist axis.
	-180 [deg]	Mainly useful for models that have an offset on the wrist axis.
Wrist axis direction (ORIENT)	Downwards	
	Upwards	Mainly useful for ceiling mount. By specifying “upwards”, the wrist axis will face the floor.

NOTE
The Palletizing Mode cannot be enabled in a TP program where the Singularity Avoidance is enabled. If you want to enable Palletizing Mode, please disable the Singularity Avoidance first.

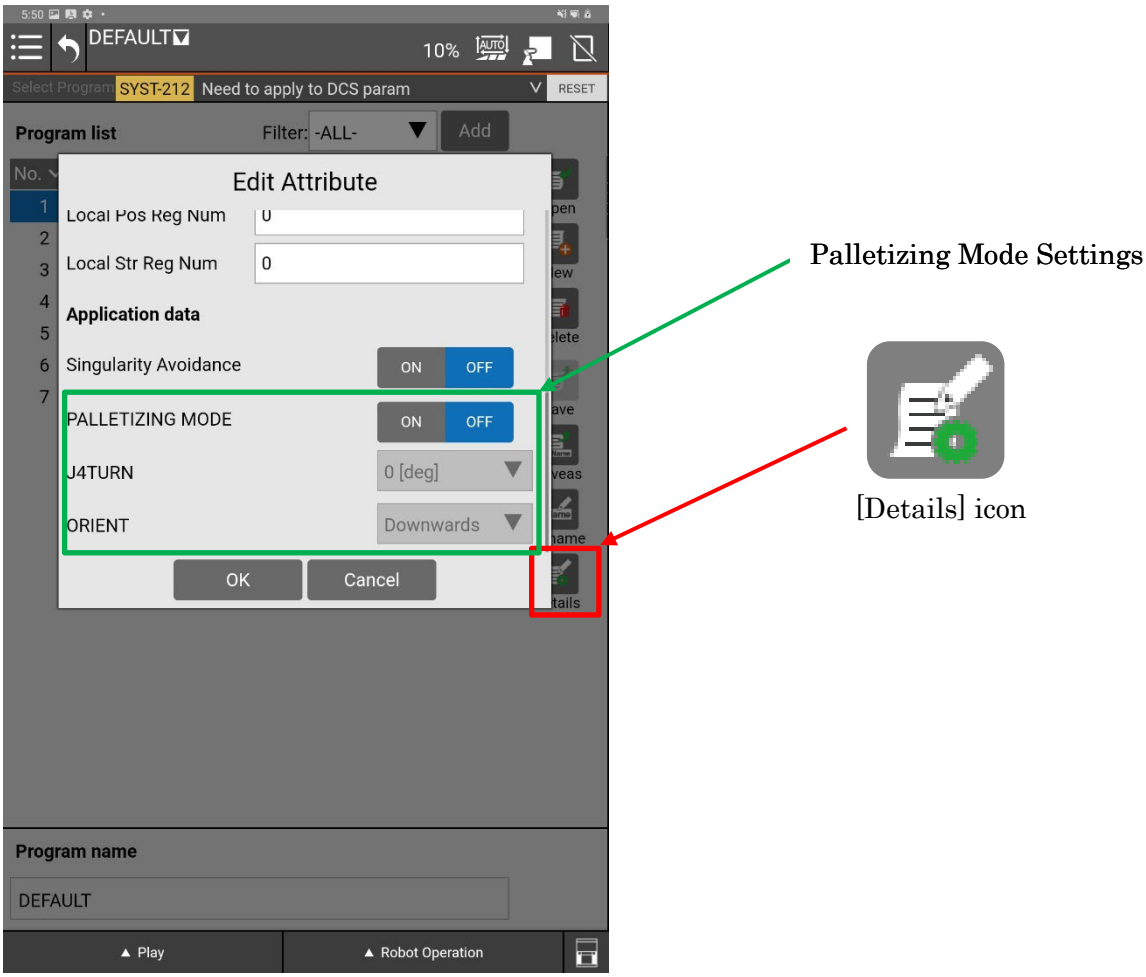
For Teach Pendant

Set from [Details] on the program list screen.



For Tablet UI

Set from the [Details] icon on the program list screen.



NOTE
The Palletizing Mode enable / disable setting in the TP program and in the Jogging / Manual Guided teaching are independent. Even if the Palletizing Mode enable / disable is switched on the TP program attribute edit screen, or TP program with different settings is selected, the Palletizing Mode enable / disable in Jogging / Manual Guided teaching does not change.

62.2.2 Cartesian motion (Linear, Circle, Arc, and Spline)

- An alarm will occur and the robot will not be able to move (*) if:
  - The wrist axis is not downward in the WORLD coordinate system at the start position of the motion.
  - J4 is not 0 degree at the start position of the motion.
  - The wrist axis is not downward in the WORLD coordinate system at the destination position of the motion.
  - J4 is not 0 degree at the destination position of the motion.

(*) Depending on the setting, read as "upward in the WORLD coordinate system", "+180 / -180 degree".

62.2.3 Joint motion

- No restrictions. It can move as a normal 6-axis robot.

62.2.4 Restrictions for TP program execution

- Cannot be used together with the Singularity Avoidance function. An alarm will occur if a TP program with both the Singularity Avoidance and Palletizing Mode enabled is executed.
- When calling a sub-program from a main-program using the CALL instruction, if the Palletizing Mode settings are different, CNT blending is not available. When calling a sub-program with different settings, set the motion statement of the main-program immediately before calling and the last motion statement of the sub-program to be called FINE.
- In a program in which the Palletizing Mode is enabled, "Wjnt when config changes" is invalid. However, if there is a WJNT instruction, Wrist Joint motion will be performed. At this time, the wrist axis may not keep downward (or upward).
- Line Tracking, Coordinated motion, and remote TCP (RTCP) cannot be used simultaneously. Also, offset that change the wrist axis posture (ex. position/tool offset, vision offset, and Dynamic Path Modification (DPM), etc..) cannot be applied.

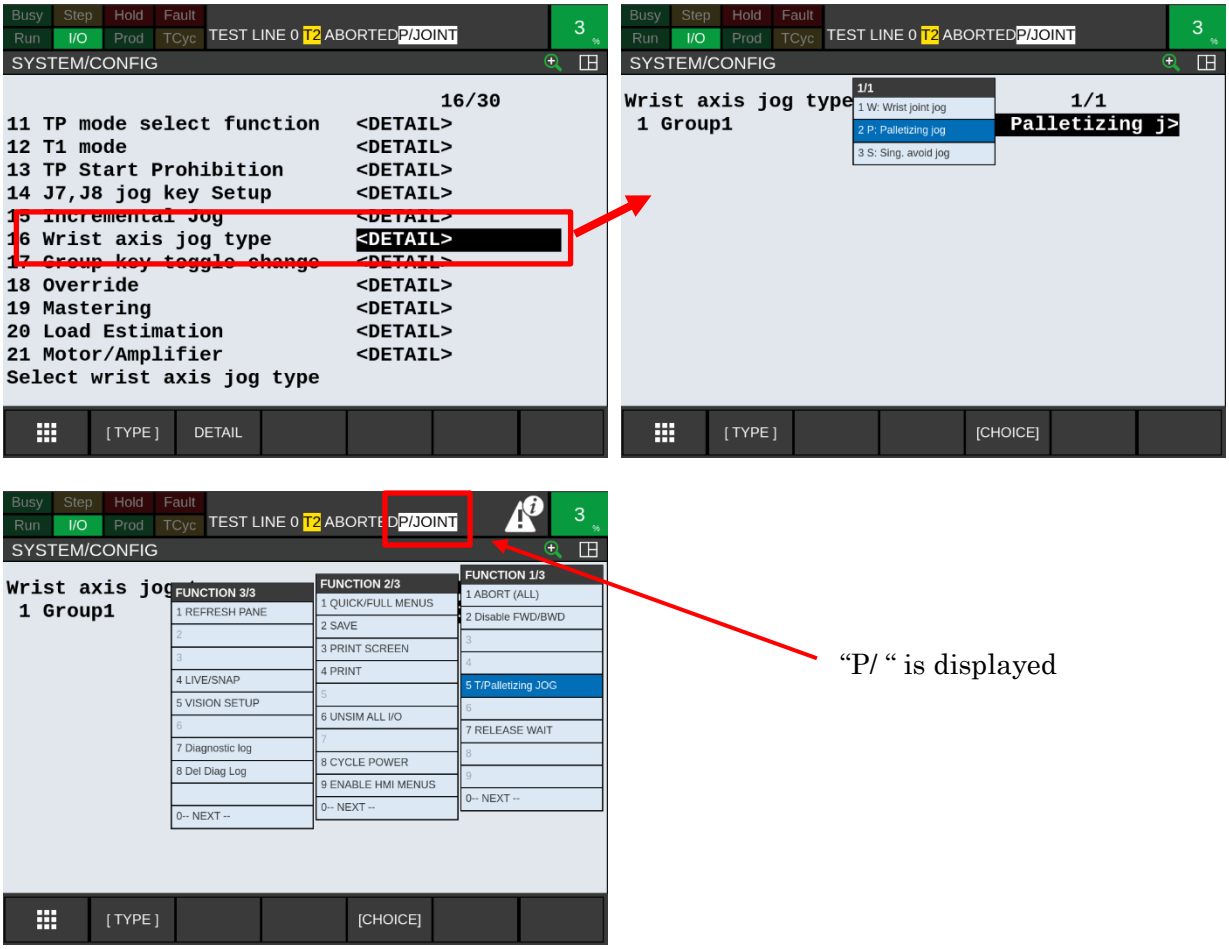
62.3 Jogging Operation

Palletizing Mode can also be used in Jogging operation. In jogging operation, J4TURN and ORIENT are automatically determined according the current position.

62.3.1 Enable/Disable setting

For Teach Pendant

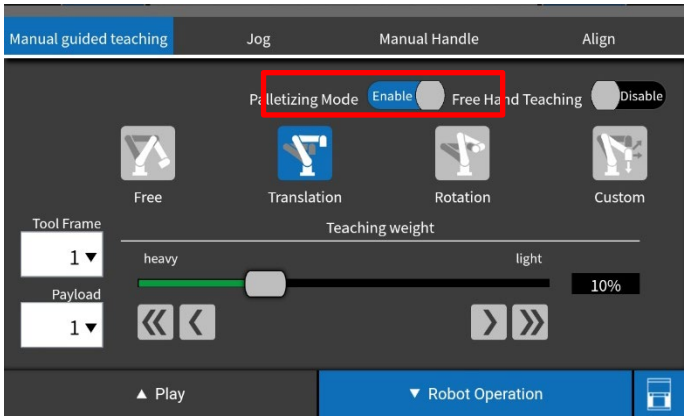
When "P: Palletizing" is selected for "Wrist axis jog type" on the System/Config screen, "T/Palletizing Jog" is displayed in the FCTN menu for that group. Selecting this item switches between enable/disable the Palletizing Mode during Jogging operation.



When the Palletizing Mode during Jogging operation is enabled, "P/" is displayed on the screen. (e.x. P/JOINT, P/WOLRD, P/TOOL, …)

For Tablet UI

The palletizing mode enable / disable setting for Jogging operation is shared with Manual Guided Teaching. A toggle switch is displayed when Translation, Rotation, or Custom is selected on the Manual Guided Teaching tab of the robot operation panel.



You can also confirm this setting by the following Data ID.
(Data ID: $JOG_MODE.IS_PALLETIZING_JOG[g] / BOOLEAN / RO / g: group number)

NOTE
The Palletizing Mode enable / disable setting in the Jogging / Manual Guided teaching and in the TP program are independent. Enable / disable the Palletizing Mode in Jogging / Manual Guided teaching does not change the enable / disable of the Palletizing Mode in the selected TP program.

62.3.2 Cartesian jogging (WORLD, TOOL, USER, and JGFRM)

- In the following cases, an alarm will occur and the robot will not be able to move:
  - The wrist axis is not downward or upward in WORLD coordinate system.
  - J4 is not 0 / -180 / -180 degree at the start position of the move.
  - Rotational jog in the W / P direction (around X axis / Y axis rotation) (*)

(*) Rotational jog in the R direction (around Z axis) is allowed only in the following cases:
- When WORLD coordinate system is selected
- When TOOL coordinate system is selected and the Z axis of the TOOL coordinate system is parallel to the Z axis of the Mechanical interface coordinate system.

62.3.3 Joint jogging

- Jogging is paused and a warning is displayed under the following conditions so that the robot's wrist axis can be turned downward / upwards easily.
  - When J4 = 0 or +180 or -180 degree by jogging the J4 axis
  - When the wrist axis to go downwards / upwards in WOLRD coordinate system by jogging J3 or J5 axis with J4 = 0 or +180 or -180 degree

62.4 Manual Guided Teach

With CRX series, the Palletizing Mode can be used even during Manual Guided Teaching.

62.4.1 Enable / disable setting

Same as Jogging operation. Please refer section 62.3.1, **Enable/Disable setting**.

62.4.2 Translation

- In the following cases, an alarm will occur and the robot will not be able to move:
  - The wrist axis is not downward or upward in WORLD coordinate system.
  - J4 is not 0 / -180 / -180 degree at the start position of the move.

62.4.3 Rotation

- In the following cases, an alarm will occur and the robot will not be able to move:
  - Rotation in the W / P direction (around X axis / Y axis rotation) (*)

(*) Rotation in the R direction (around Z axis) is allowed only in the following cases:
- When WORLD coordinate system is selected
- When TOOL coordinate system is selected and the Z axis of the TOOL coordinate system is parallel to the Z axis of the Mechanical interface coordinate system.

62.4.4 Free

- You can operate freely regardless of whether the Palletizing Mode is enabled or disabled.
- In Free mode, the palletizing mode enable / disable switch toggle is not displayed.

62.5 Remarks

62.5.1 Align function

When teaching with the Tablet UI, the Align function can be used. You can easily move to the posture of the Palletizing Mode by using Align function with reference “world Frame Z axis”.

NOTE

1 The Align function moves toward a posture in which the referencing axis and the wrist axis are parallel, with the movement that minimizes rotation of each axis. So even if you use the Align function expecting to move to a posture which the wrist axis facing downward and J4 = 0 degree, you may find that the wrist axis is facing upward or J4 = +180 / -180 degree. In that cases, use the Align function after moving the wrist axis almost downward and near J4 = 0 degree using Joint Jogging or Free mode Manual Guided Teach.

2 The Align function cannot be used near J5 = 0 degree. In that case, use the Align function after moving J5 axis slightly using Joint Jogging or Free mode Manual Guided Teach.

63 CUSTOMIZE SUPPORT FUNCTION

This chapter describes Customize Support Function (J742).

This function provides the necessary function for the one who develops the system specific function. It supports followings.

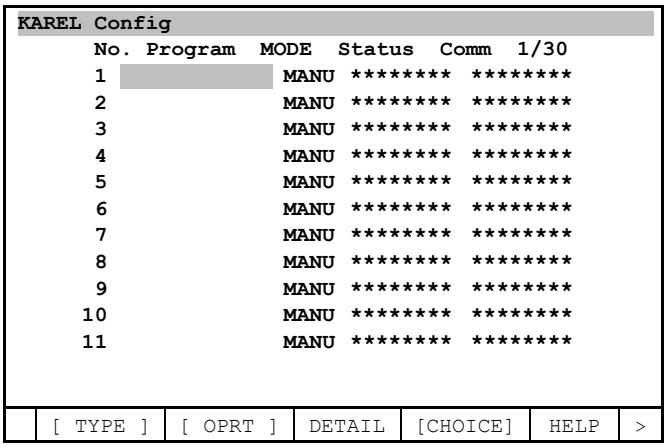
- Executing or aborting KAREL programs
- Auto running the KAREL programs at the time of power on
- Periodically monitor and output the specified KAREL program status
- Set the configuration of custom menu setting (cf. Section 63.2)

63.1 KAREL CONFIG

63.1.1 KAREL Config Screen

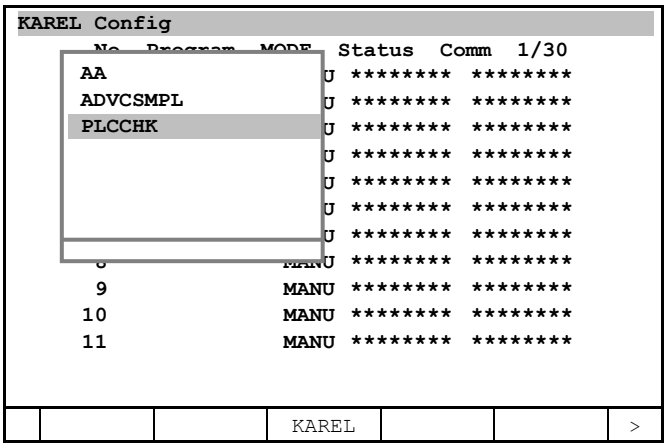
STEP

1 Select MENU – 6. SETUP – KAREL Config.
2 You will see the KAREL Config screen like the following.

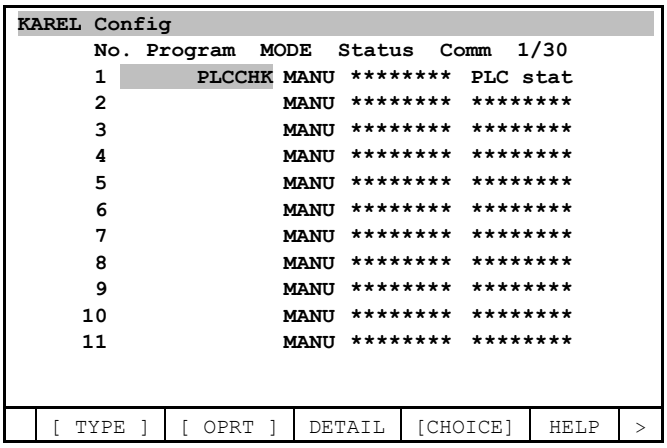


63.1.2 Use KAREL Config Screen

Select F4, [CHOICE], then you will see the list of KAREL program to be registered to run.



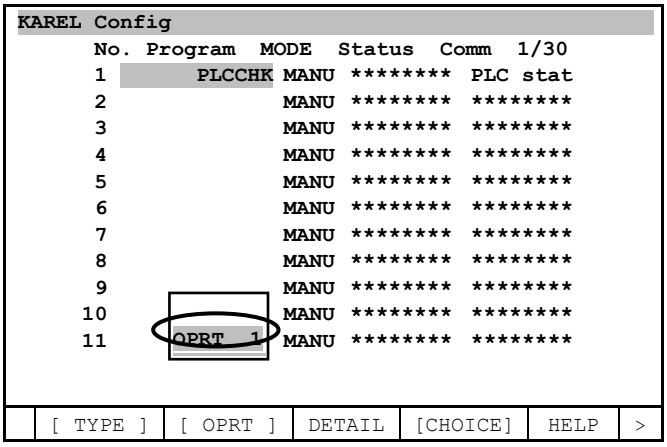
In this documentation, PLCCHK is used as the sample to explain. Set the cursor onto the PLCCHK, and press the [ENTER] key.



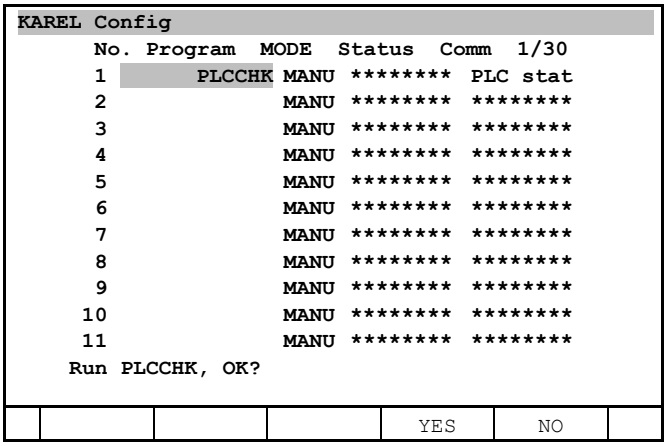
There are four status, running, paused, aborted and ********, where ******** means that none of these 3 status applies.
Comm is the comment that is described in the KAREL source file as %COMMENT.

63.1.3 Run KAREL Program

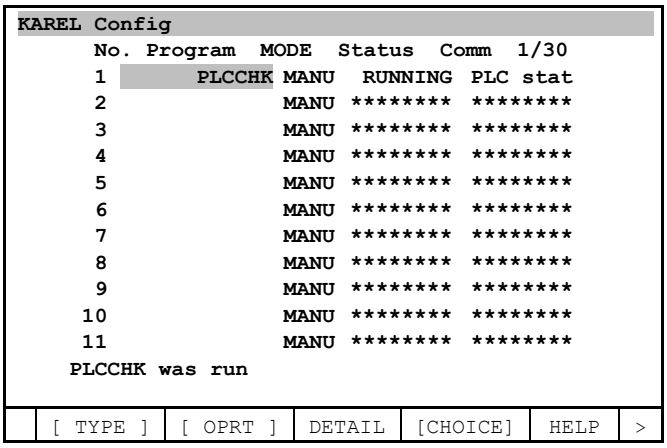
Set the cursor onto PLCCHK in Program row then press the F2, [OPRT].



Select 1. RUN, then you will see confirmation message.

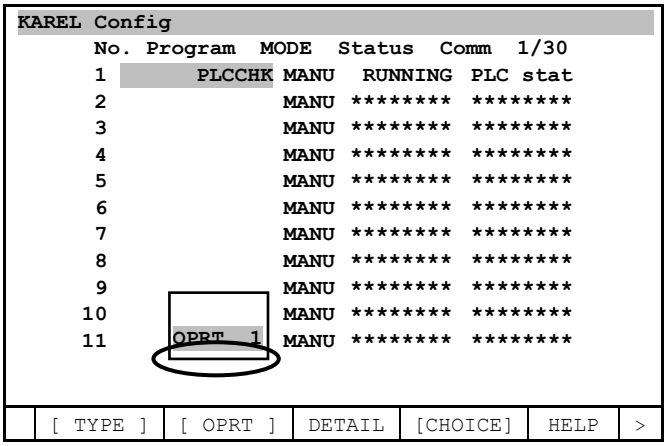


Press F4, YES to run the program.

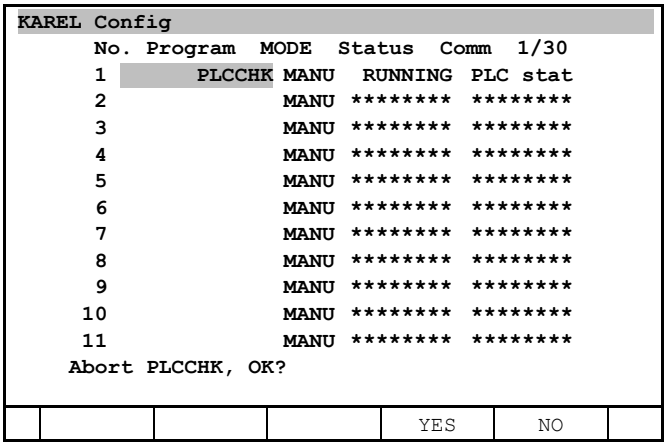


63.1.4 Abort KAREL Program

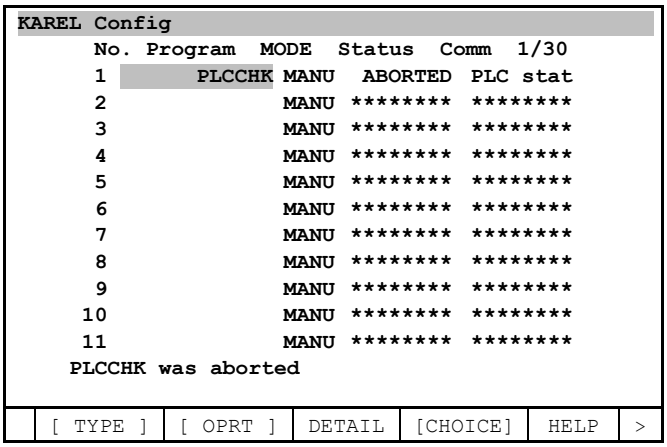
Set the cursor onto PLCCHK in Program row then press the F2, [OPRT].



Set the cursor onto 2. ABORT then press the [ENTER] key.
You will see a confirmation message.

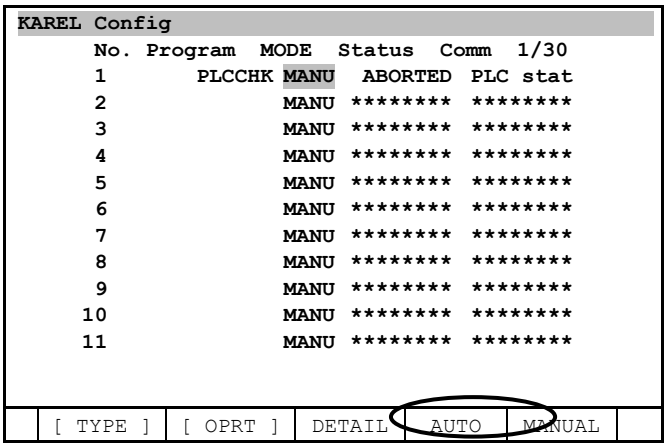


Select F4, YES, then KAREL program will be aborted.

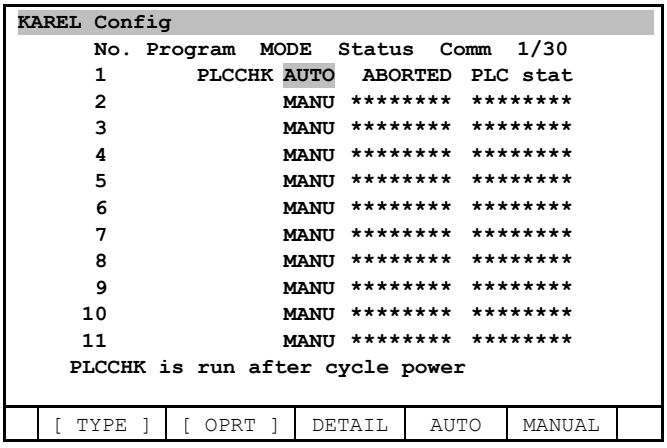


63.1.5 Start Mode Config of KAREL program

Set the cursor onto the MODE row, then the F4 and F5 keys turns to AUTO and MANUAL respectively.



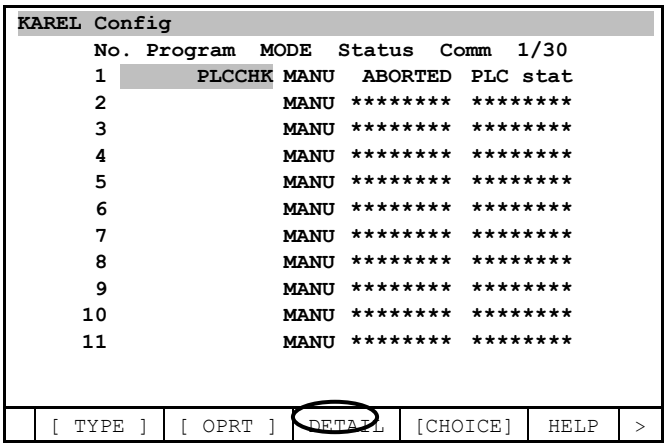
Default is MANUAL. If you press F4 [AUTO], the KAREL program runs after cycling power of the controller.



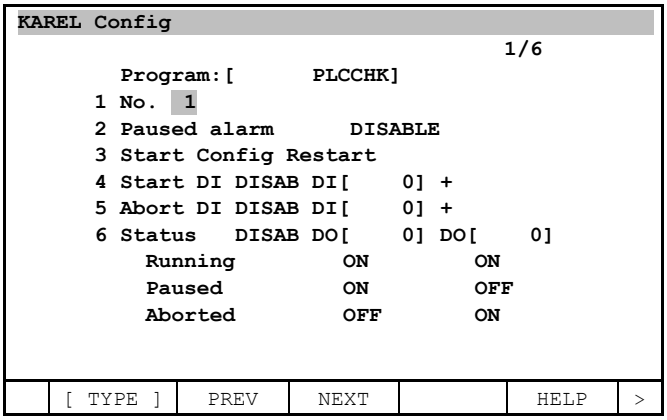
There is a limit that the number of running the KAREL simultaneously. It depends on the configuration of the robot system. “PROG-014 Max task number exceed” will be posted when the number of running KAREL is going to be over max number.

63.1.6 Detail Screen of KAREL Config

Press the F3, DETAIL.

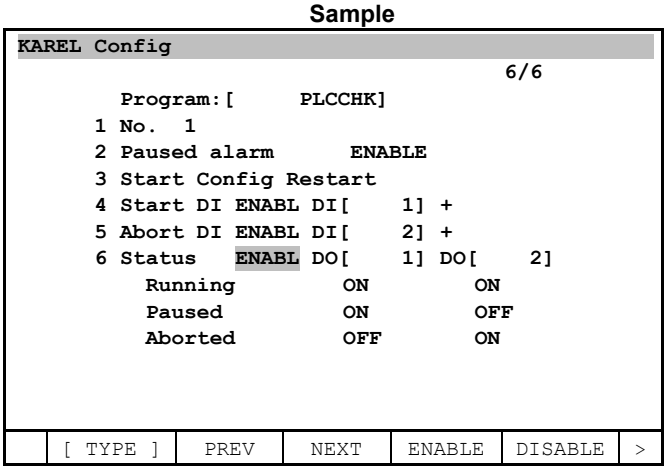


You will see the following screen.



Item	Comment	
Paused alarm	When this item is set ENABLE, a warning appears when specified program, PLCCHK in the previous example, is paused or aborted.	
	Paused	"SYST 294 Paused (program name)" "SYST 296 Specified KAREL was paused"
	Aborted	"SYST 295 Aborted (program name)" "SYST 297 Specified KAREL was aborted"
Start Config	Select whether to start from the beginning or start from the middle when execute KAREL.	
	Restart	Once aborting the specified program and start from the top of the program.
	Resume	Specified program is restarted from the paused point.
Start DI	Sets the start operation. Detecting the rising edge (+) or falling edge (-) of specified DI starts the specified program.	
	When "Start DI" is set enable, DI port number and Up or Down cannot be changed. If you would like to set them, please set "Start DI" disable once. After setting them, please set it enable. Then this configuration is enabled after cycle power.	
Abort DI	Sets the start operation. Detecting the rising edge (+) or falling edge (-) of specified DI starts the specified program. The setting of Abort DI is the same as start DI	
Status	Is to monitor the specified program and output its status with two DOs. As the default, program is monitored every 100msec. Its config is enabled after cycle power.	

Sample



PLCCHK status	DO[1]	DO[2]
Running	ON	ON
Paused	ON	OFF
Aborted	OFF	ON

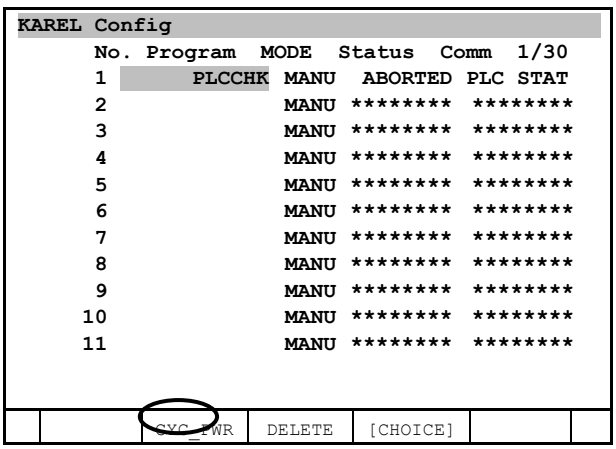
To switch previous screen, press the F2, PREV. In this case, screen is switched to No.30.
To switch next detail screen, press the F3, NEXT. In this case, screen is switched to No.2.

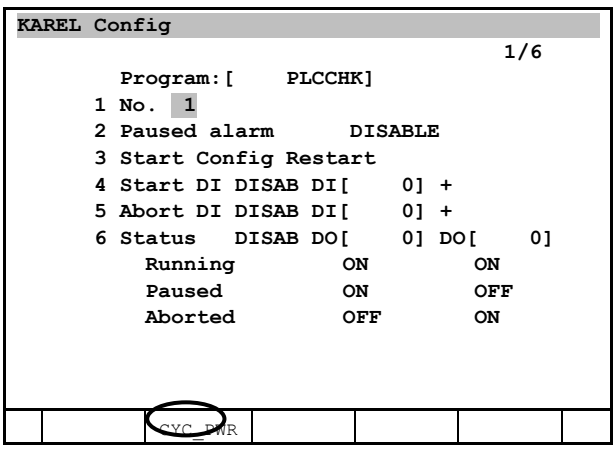
63.1.7 Limitation and Caution of KAREL Config

- Status is monitored at a constant frequency. Thus status is not changed at a real time. For example, program status is changed from running to paused, and aborted in one monitoring cycle, paused status is not detected at next monitoring cycle. Therefore status is changed from running to aborted in this case.
- In the case of above sample, after you set the detail of PLCCHK, return to KAREL Config screen with [PREV] key and change the registered No.1 program from PLCCHK to other program, such as C2, PLCCHK does not start if DI[1] is turned on. C2 is run instead. For prevention of miss operation, please do not change program after set the detail.

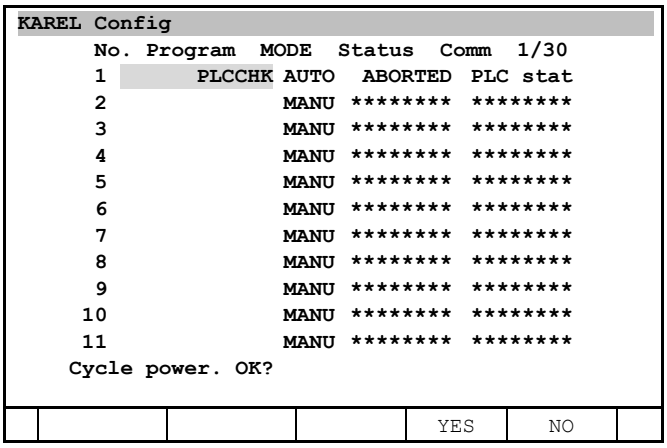
63.1.8 Cycle Power

Both KAREL config and detail screen enable to cycle power.
In R-50*i*A Mate controller, do cycle power manually.





F2, CYC_PWR is displayed when you press the [NEXT] key. Set the TP to enable and press the F2, CYC_PWR.



Selecting F4, YES do cycle power.

63.2 CUSTOM MENU

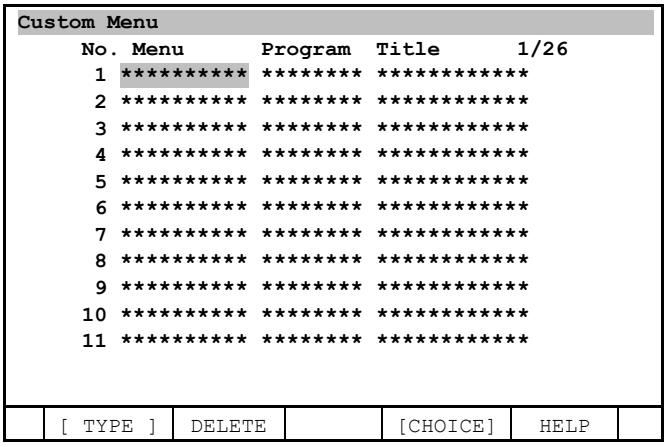
63.2.1 Overview

This function allows to set the TP or KAREL program to the menu. It is easy to run the specified program to select the menu.

For example, user-created screen by KAREL program is allowed to display quickly by selecting the menu.

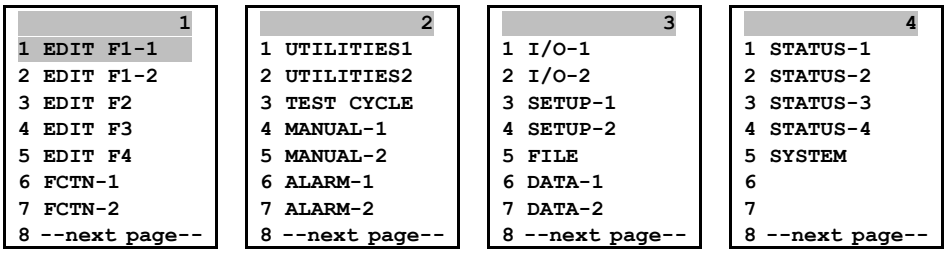
63.2.2 Starting Custom Menu

Select MENU – setup – Custom Menu. You will see the following screen.

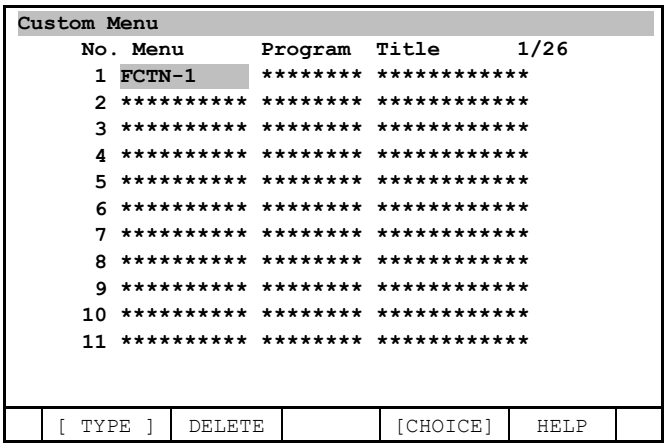


63.2.3 Set Custom Menu

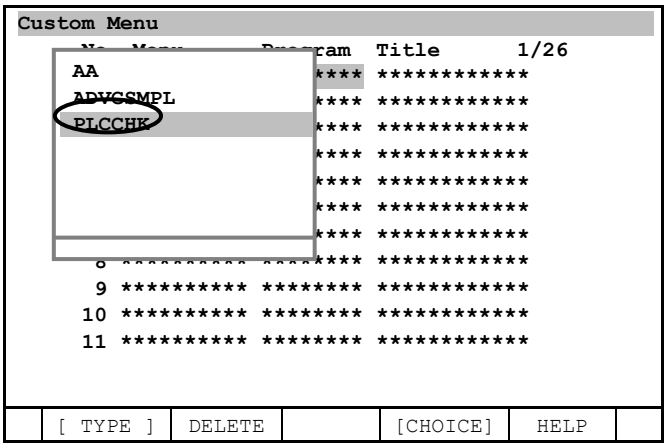
Press the F4, [CHOICE].



Select item that you would like to add. Select FCTN –1 as an example.

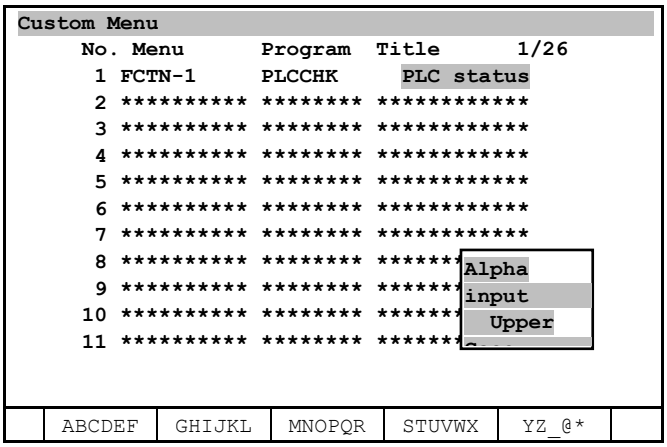


Set the cursor onto program row and press F4, [CHOICE] key. As an example PLCCHK is selected.

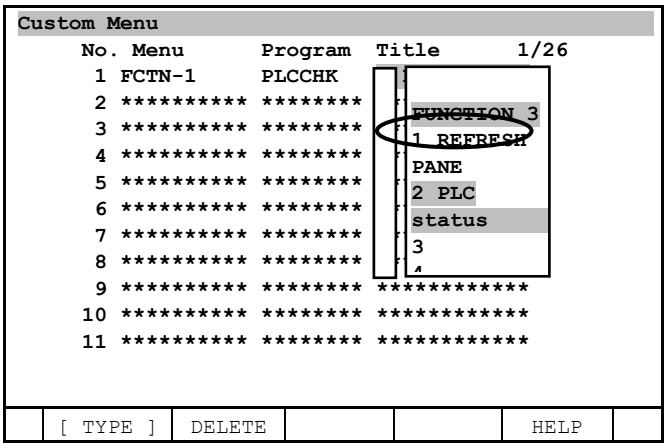


Next, set the cursor onto title row and press the [ENTER] key. PLCCHK is input for instance.

NOTE
Backslashes and commas cannot be used in screen titles.

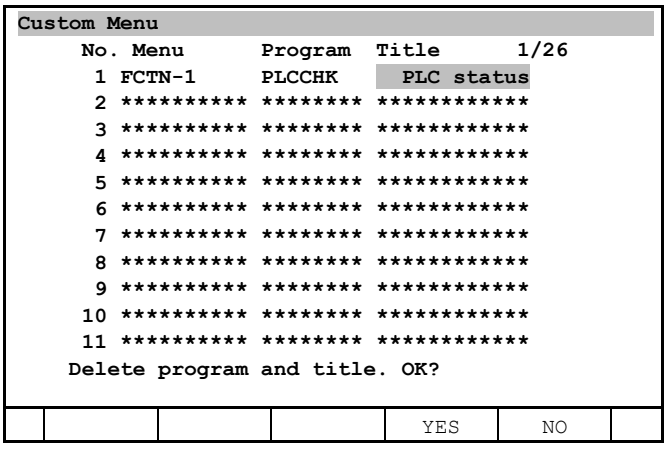


Select FCTN → 0 -- NEXT -- → 0 --NEXT --. Verify PLCCHK is added to FCTN menu.

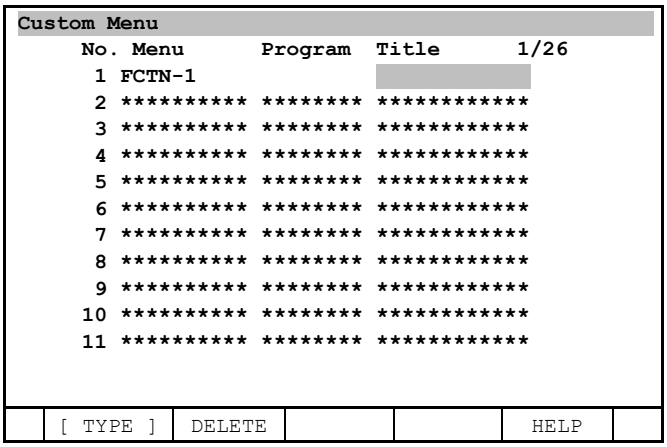


63.2.4 Delete Set

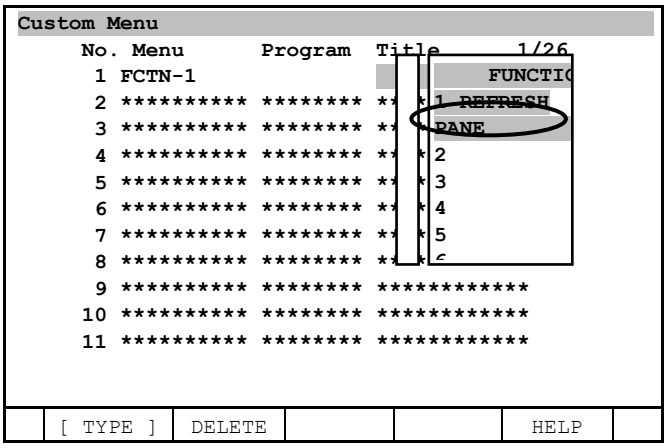
In the case of deleting set, press the F2, DELETE. You will see a confirmation message like the following.



Select the F4, YES.
Specified program and title will be deleted.



Select FCTN → 0 -- NEXT -- → 0 -- NEXT --. Verify PLCCHK will be deleted from FCTN menu.



64 PALLETIZING FUNCTION

This chapter explains the palletizing function. To use this function, Palletizing option (J500) is required.

Contents of this chapter

64.1 PALLETIZING FUNCTION
64.2 PALLETIZING INSTRUCTIONS
64.3 TEACHING THE PALLETIZING FUNCTION
64.4 EXECUTING THE PALLETIZING FUNCTION
64.5 MODIFYING THE PALLETIZING FUNCTION
64.6 PALLETIZING FUNCTION WITH EXTENDED AXES
64.7 PALLETIZING ALL-POINT TEACHING

64.1 PALLETIZING FUNCTION

Palletizing function

Palletizing is a function for the orderly stacking of workpieces by only teaching several representative points.

- A stacking pattern can be created easily by teaching representative stack points.
- A path pattern can be created by teaching path points (approach points and retraction points).
- Multiple path patterns can be set to perform palletizing in a wide variety of patterns.

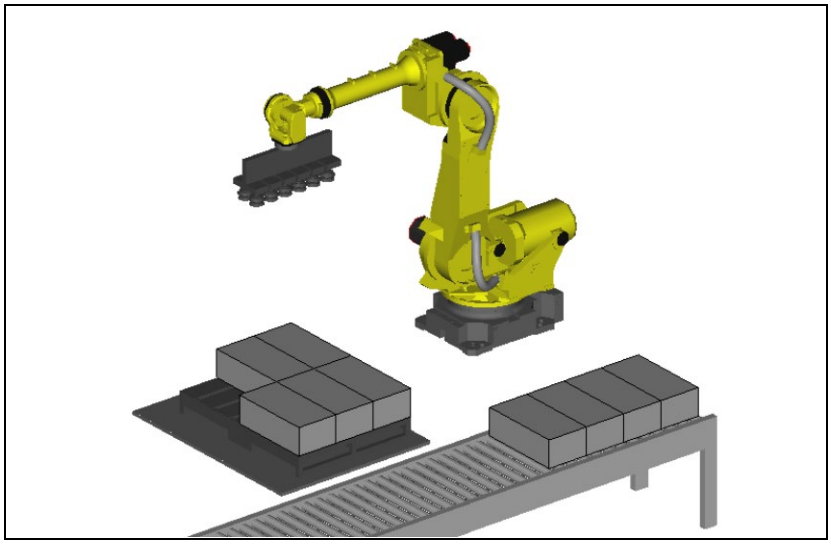


Fig. 64.1 (a) Palletizing

Structure of the palletizing function

The palletizing function consists of the following two patterns:

- Stacking pattern : Determines the method of stacking workpieces.
- Path pattern : Determines the path along which the robot hand moves to stack workpieces.

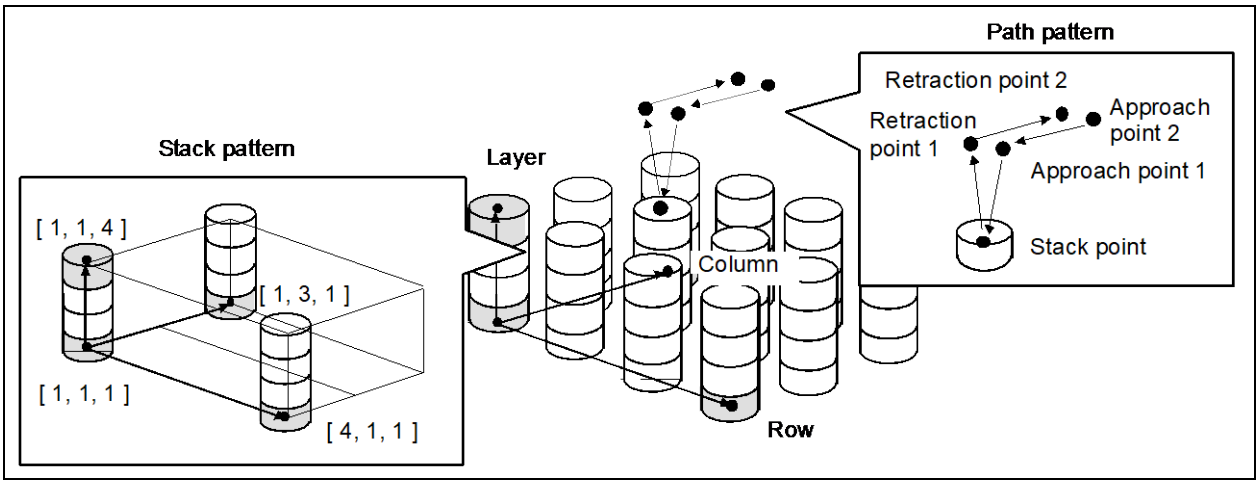


Fig. 64.1 (b) Palletizing pattern

Types of palletizing

There are the following four types of palletizing according to the methods for setting stack and path patterns (See Section 64.3).

- Palletizing B and palletizing BX
- Palletizing E and palletizing EX

- Palletizing B

Palletizing B can achieve a comparatively easy stacking pattern by one kind of path pattern.

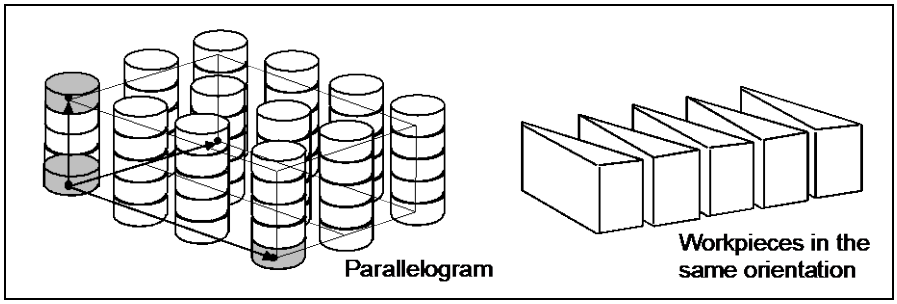


Fig. 64.1 (c) Palletizing-B function

- Palletizing E

Palletizing E can be used for more complex stack patterns (such as when the postures of workpieces are to be changed or when the shape made by the stacked workpieces, as viewed from below, is not a rectangle).

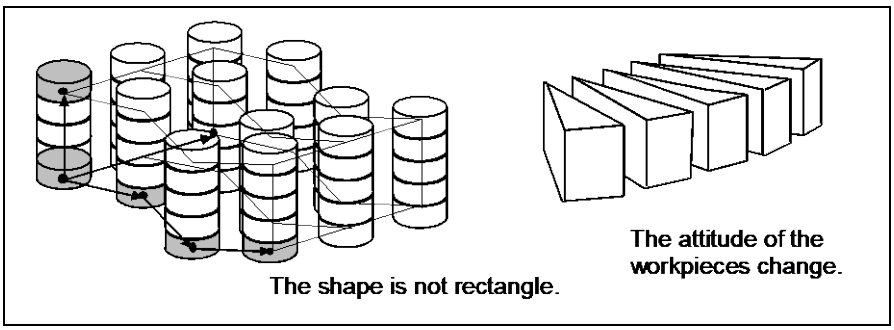


Fig. 64.1 (d) Palletizing E

- **Palletizing BX and EX**

  For palletizing BX and EX, multiple path patterns can be set. For palletizing B and E, only one path pattern can be set.

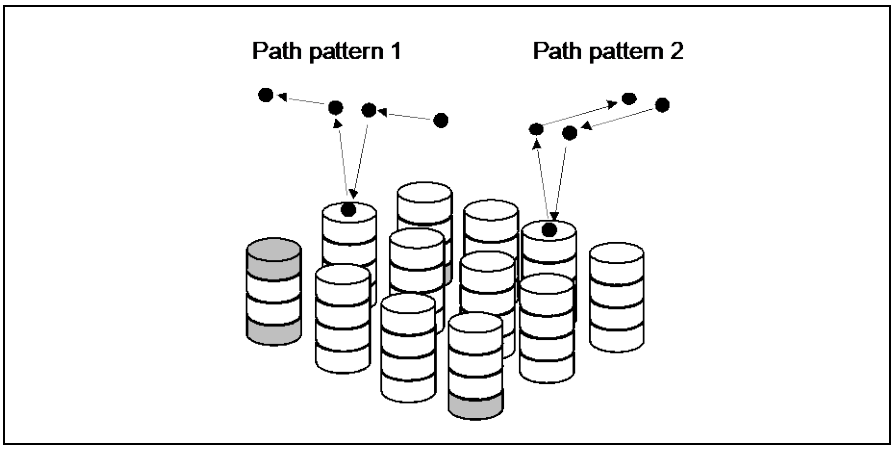


Fig. 64.1 (e) Palletizing BX and EX

64.2 PALLETIZING INSTRUCTIONS

The following palletizing instructions are available:

Table 64.2 Palletizing instructions

Instruction	Function
Palletizing instruction	Calculates the current path based on a stacking pattern, path pattern, and the value held in the palletizing register, and rewrites the position data of a palletizing motion instruction.
Palletizing motion instruction	A motion instruction dedicated to palletizing. It has position data of an approach point, stack point, or retraction point.
Palletizing end instruction	Increments (or decrements) the value of a palletizing register.

Palletizing instruction

Based on the value held in the palletizing register, the palletizing instruction calculates the position of the current stack point from a stack pattern, and also calculates the current path from a path pattern. It then rewrites the position data of a palletizing motion instruction.

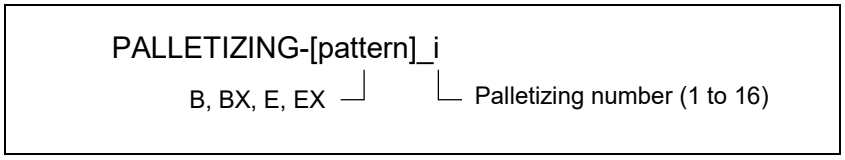
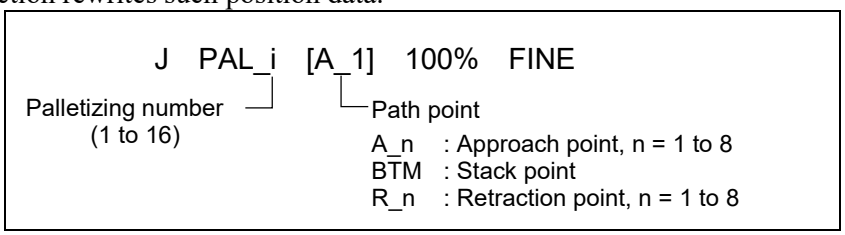


Fig. 64.2 (a) Format of the palletizing instruction

Palletizing motion instruction

The palletizing motion instruction is a motion instruction that uses three path points ― an approach point, stack point, and retraction point ― as position data. This instruction is dedicated to palletizing. Each palletizing instruction rewrites such position data.



fFig. 64.2 (b) Format of the palletizing motion instruction

Palletizing end instruction

The palletizing end instruction calculates the next stack point and increments (or decrements) the palletizing register value.

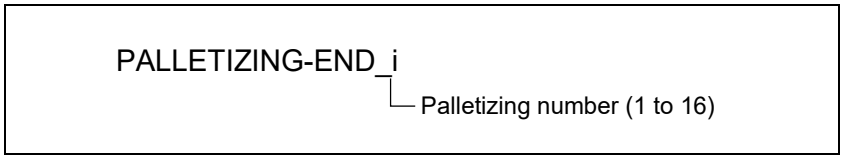


Fig. 64.2 (c) Format of the palletizing end instruction

Example
```
1:   PALLETIZING-B_3
2: J PAL_3[ A_2 ] 50% CNT50
3: L PAL_3[ A_1 ] 100mm/sec CNT10
4: L PAL_3[ BTM ] 50mm/sec FINE
5:   hand1 open
6: L PAL_3[ R_1 ] 100mm/sec CNT10
7: J PAL_3[ R_2 ] 100mm/sec CNT50
8:   PALLETIZING-END_3
```

- Palletizing number

Upon completion of the teaching of palletizing data, palletizing numbers are written automatically together with the instructions (palletizing instruction, palletizing motion instructions, and palletizing end instruction).
When a new palletizing operation is taught, a palletizing number is assigned automatically.

Palletizing register instruction

The palletizing register instruction is used to control palletizing. It performs stack point specification, comparison, and branch. (See Subsection 64.4.1.)

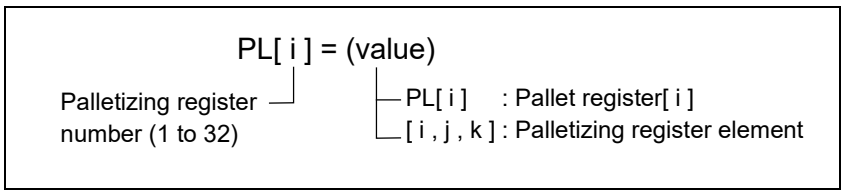


Fig. 64.2 (d) Palletizing register

64.3 TEACHING THE PALLETIZING FUNCTION

The palletizing function is taught using the following procedure:

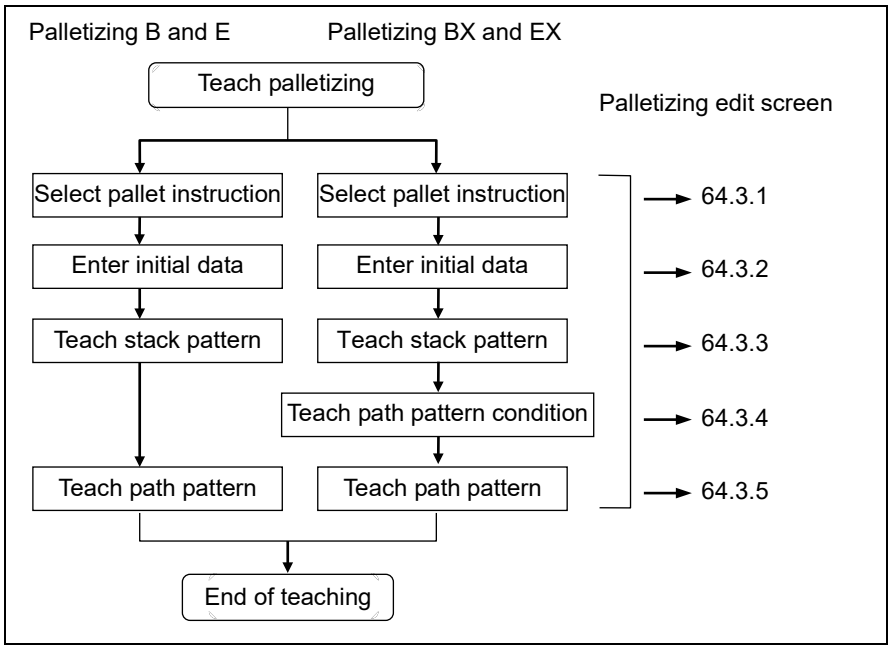


Fig. 64.3 Procedure for teaching the palletizing function

The palletizing function is taught on the palletizing edit screens. One of the palletizing edit screens will be displayed automatically when a palletizing instruction is selected.
When the palletizing function is taught, necessary palletizing instructions such as a palletizing instruction, palletizing motion instruction, and palletizing end instruction are inserted automatically.
The following sections explain the teaching of palletizing EX. For palletizing B, BX, or E, assume that some functions of palletizing EX are restricted.

NOTE
To improve the motion accuracy of palletizing, Tool Center Point should be accurately set. (Please see chapter of ”2.10.1 Setting a Tool Coordinate System” in R-50iA/R-50iA Mate CONTROLLER OPERATOR’S MANUAL (Basic Operation) (B-84464EN))

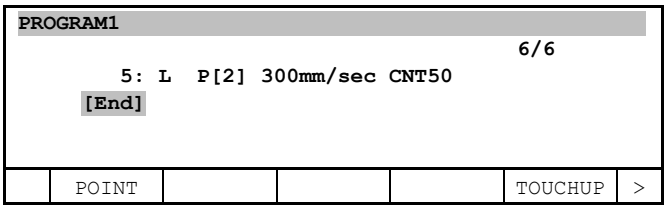
64.3.1 Selecting a Palletizing Instruction

To select a pallet instruction, select the type of palletizing to be taught (palletizing B, BX, E, or EX).

Procedure 64-1 Selecting a palletizing instruction

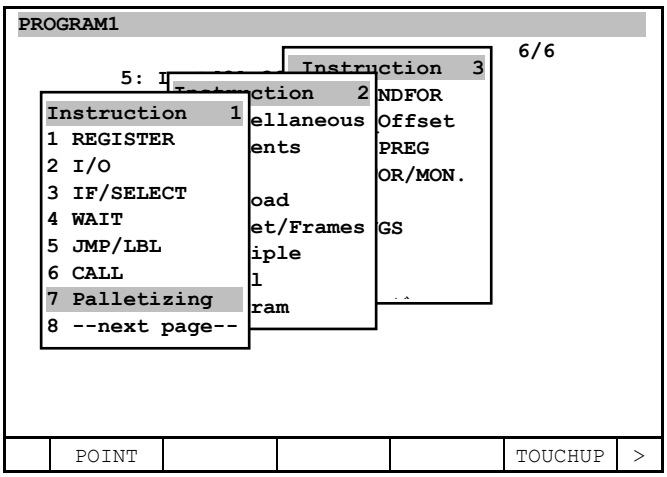
Condition

- Make sure that the teach pendant is enabled.
- Make sure that the palletizing instruction is selected on the program edit screen.

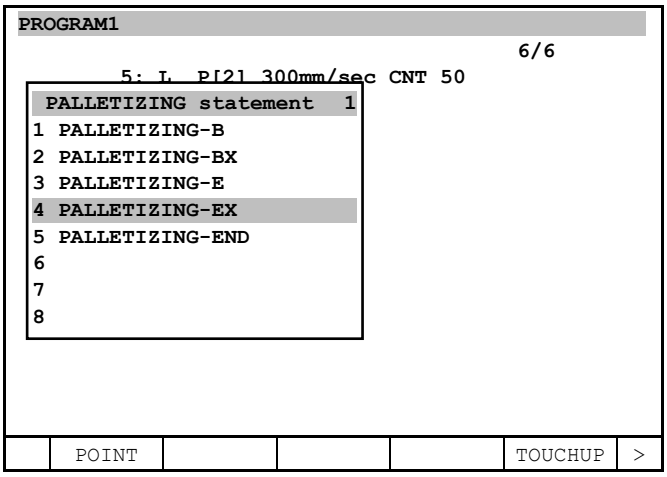


Step

1 Press NEXT ">" to display the next page. Press F1, [INST] to display a menu.



2 Select "Palletizing".



3 Select "PALLETIZING-EX." The screen then changes to the initial data input screen automatically, one of the palletizing edit screen.

```
PROGRAM1
     PALLETIZING Configuration

       PALETIZING_1  [                    ]
       TYPE = [PALLET     ]   INCR = [   1 ]
       PAL REG   = [  1] ORDER = [RCL]
         ROWS    = [  1   LINE   FIX ]
         COLUMNS = [  1   LINE   FIX ]
         LAYERS  = [  1   LINE   FIX    1 ]
         AUXILIARY POS = [ NO         ]
       APPR =[ 1] RTRT =[ 1] PATTERN=[ 1]


     Press ENTER

   | PROG |  |  |  | DONE |
```

64.3.2 Inputting Initial Data

On the initial data input screen, specify how palletizing is performed. The data set on the initial data input screen is used for subsequent teach screens. The initial data screen contains the following items:

For palletizing B

```
PROGRAM1
     PALLETIZING Configuration

       PALETIZING_4  [WORK PALLET       ]
       TYPE = [PALLET     ]   INCR = [   1 ]
       PAL REG   = [  1] ORDER = [RCL]
         ROWS    = [  5]
         COLUMNS = [  4]
         LAYERS  = [  3]
         AUXILIARY POS = [NO   ]
       APPR = [ 2] RTRT = [ 2]


     Press ENTER

   | PROG |  |  |  | DONE |
```

For palletizing BX

```
PROGRAM1
     PALLETIZING Configuration

       PALETIZING_4  [WORK PALLET       ]
       TYPE = [PALLET     ]   INCR = [   1 ]
       PAL REG   = [  1] ORDER = [RCL]
         ROWS    = [  5]
         COLUMNS = [  4]
         LAYERS  = [  3]
         AUXILIARY POS = [NO   ]
       APPR = [ 2] RTRT = [ 2] PATTERN= [ 2]


     Press ENTER

   | PROG |  |  |  | DONE |
```

For palletizing E

```
PROGRAM1
     PALLETIZING Configuration

       PALETIZING_4  [WORK PALLET       ]
       TYPE = [PALLET     ]   INCR = [   1 ]
       PAL REG   = [  1] ORDER = [RCL]
         ROWS    = [  5   LINE   FIX ]
         COLUMNS = [  4   LINE   FIX ]
         LAYERS  = [  3   LINE   FIX    1 ]
         AUXILIARY POS = [ NO         ]
       APPR = [ 2] RTRT = [ 2]


     Press ENTER

   | PROG |  |  |  | DONE |
```

For palletizing EX

```
PROGRAM1
     PALLETIZING Configuration

       PALETIZING_4  [WORK PALLET       ]
       TYPE = [PALLET     ]   INCR = [   1 ]
       PAL REG   = [  1] ORDER = [RCL]
         ROWS    = [  5   LINE   FIX ]
         COLUMNS = [  4   LINE   FIX ]
         LAYERS  = [  3   LINE   FIX    1 ]
         AUXILIARY POS = [ NO         ]
       APPR = [ 2] RTRT = [ 2] PATTERN= [ 2]


     Press ENTER

   | PROG |  |  |  | DONE |
```

Table 64.3.2 (a) Types of palletizing

	Arrangement mode	Layer pattern	Posture control	Path pattern count
B	2-point teaching only	Not set	Always fixed	1
BX	2-point teaching only	Not set	Always fixed	1 to 16

	Arrangement mode	Layer pattern	Posture control	Path pattern count
E	2-point teaching, all-point teaching, or interval specification	Set	Fixed or split	1
EX	2-point teaching, all-point teaching, or interval specification	Set	Fixed or split	1 to 16

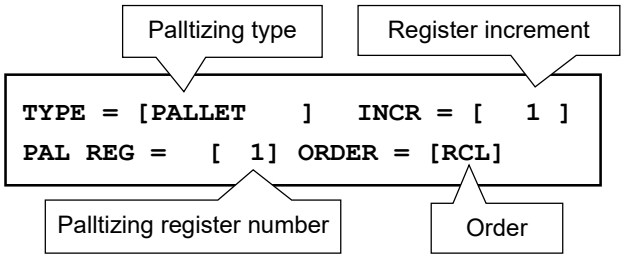
When a pallet instruction is selected, the initial data input screen corresponding to the selected type of palletizing will be displayed. For palletizing EX, all palletizing functions can be specified. For palletizing B, BX, and E, restrictions are imposed on the specification of the functions.
This section explains how to enter initial data for palletizing EX. For palletizing B, BX, or E, assume that some functions of palletizing EX are restricted.

Table 64.3.2 (b) Initial palletizing data

Palletizing number	A number is assigned automatically when a palletizing statement is taught. PALLETIZING_N: 1 to 16
Palletizing type	Specify whether the palletizing register is to be incremented or decremented by the palletizing end instruction. (See Subsection 64.4.1.) Select stacking (PALLET) or unstacking (DEPALLET).
Register increment	Specify the value by which the value held in the palletizing register is to be incremented or decremented by the palletizing end instruction. (See Subsection 64.4.1.)
Palletizing register	Specify the palletizing register to be used by the palletizing instruction and palletizing end instruction.
Order	Specify the stacking (unstacking) order of row, column, and layer. R: Row, C: Column, L: Layer
Numbers of rows, columns, and layers	Numbers of rows, columns, and layers for a stacking pattern. (See Subsection 64.3.3.) 1 to 127
Arrangement mode	How rows, columns, and layers are arranged for a stack pattern. The 2-point or all-point teaching, or interval specification can be specified (only for palletizing E or EX).
Posture control	Control the posture at rows, columns, and layers for a stacking pattern. Select E or EX.
Layer pattern count	How workpieces are stacked can be specified for each layer (only for palletizing E or EX). 1 to 16
Number of approach points	Number of approach points in a path pattern. (See Subsection 64.3.5.) 0 to 8
Number of retraction points	Number of retraction points in a path pattern. (See Subsection 64.3.5.) 0 to 8
Path pattern count	Number of path patterns (Subsection 64.3.4) (only for palletizing BX or EX). 1 to 16

Initial data related to the stacking method

In the palletizing function, the stack point is controlled using a palletizing register. (See Subsection 64.4.1, “Palletizing Register”.) How the palletizing register is controlled can be specified as initial data. According to this data, the way of stacking is determined.



- For the pallet type (TYPE), specify either PALLET or DEPALLET (standard setting: PALLET). (See Subsection 64.4.1, “Palletizing Register”.)
- For the register increment (INCR), specify by which amount the stack (unstack) position advances or retracts. That is, specify a value by which the palletizing register is incremented or decremented by

the palletizing end instruction. The standard setting is 1. (See Subsection 64.4.1, “Palletizing Register”.)

- As the palletizing register, specify the register number of a palletizing register used for stack control.

> **⚠ CAUTION**
> Make sure that the specified palletizing register number is not used by another palletizing function.

- For the order (ORDER), specify the stacking/unstacking order of row, column, and layer.

Stacking workpieces in the order: Row, Column, Layer

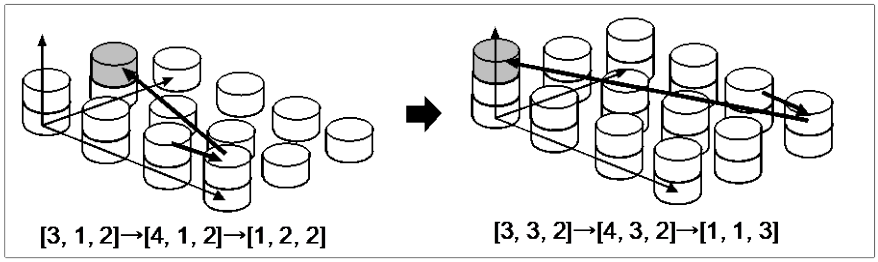
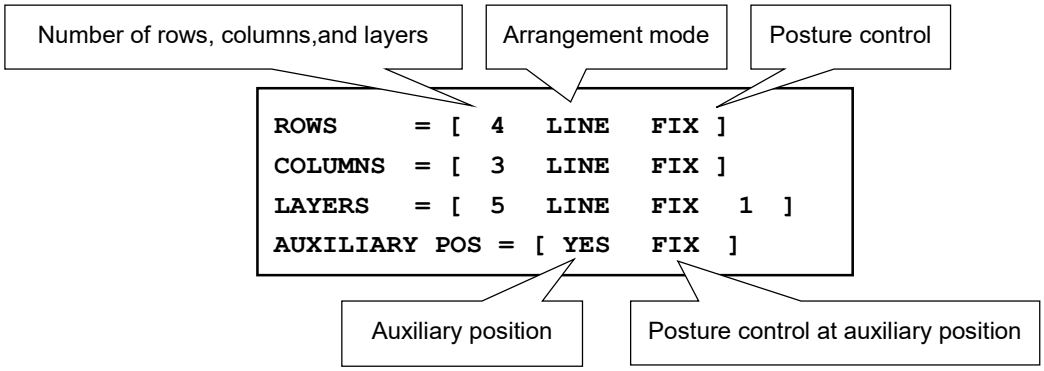


Fig. 64.3.2 Palletizing order

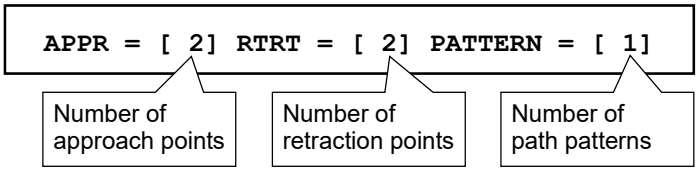
Initial data related to a stacking pattern

As the stacking pattern data, specify the numbers of rows, columns, and layers, arrangement mode, posture control type, the number of layer patterns and also specify whether to provide an auxiliary position. (See Subsection 64.3.3, “Teaching a Stacking Pattern”.)



Initial data related to a path pattern

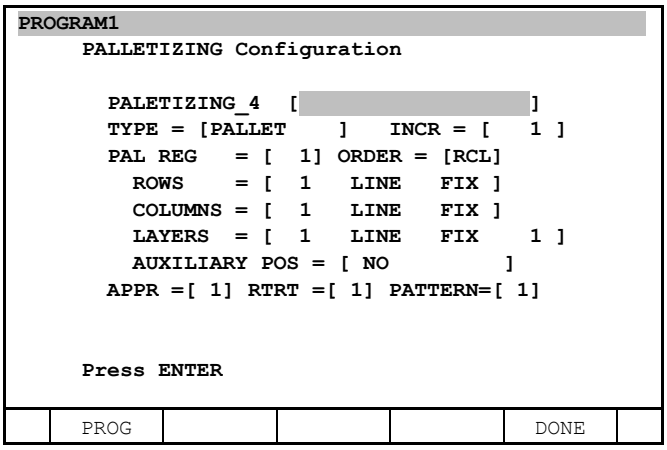
As the initial path pattern data, specify the number of approach points, the number of retraction points and the number of path patterns. (See Subsection 64.3.5, “Teaching a Path Pattern”.)



Procedure 64-2 Inputting initial palletizing data

Step

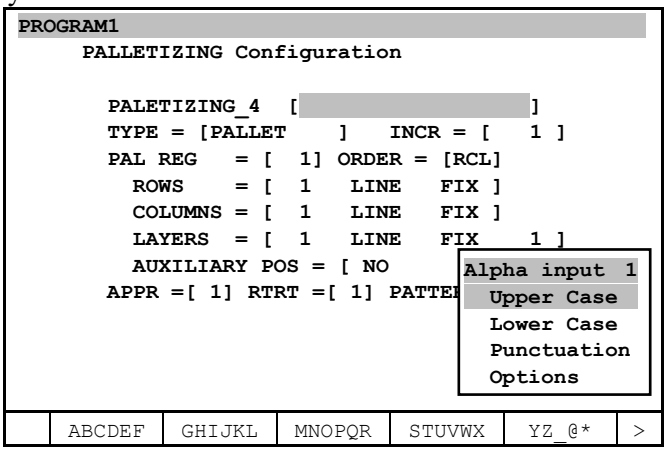
1 As the palletizing instruction, select palletizing-EX. The initial data input screen will be displayed. (See Subsection 64.3.1, “Selecting a Palletizing Instruction”.)



NOTE
PALLETIZING_4 displayed on the initial data input screen indicates the 4th palletizing instruction in the program.

2 To enter a comment, follow the procedure below.

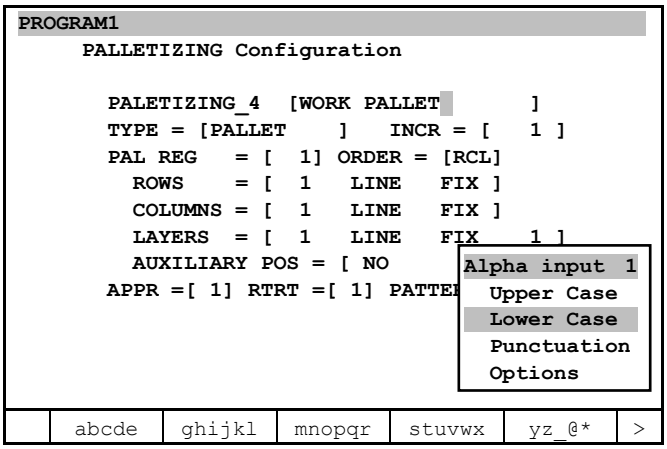
a Place the cursor on the comment line, then press the [ENTER] key. The character input menu will be displayed.



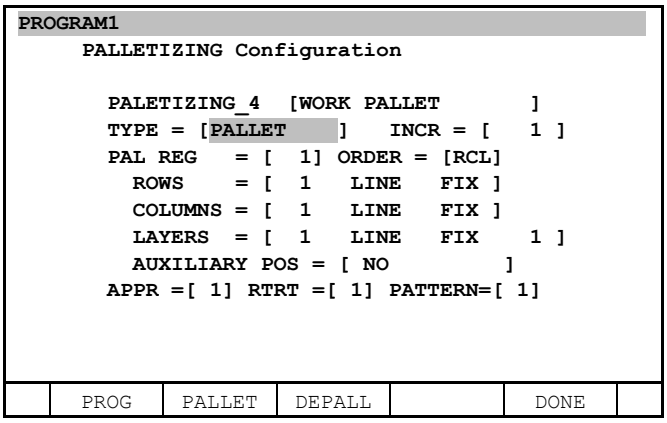
b Select the type of character input to be used ― upper case, lower case, punctuation or options ― with the ↑ and ↓ keys.

c Press an appropriate function key, then enter characters.

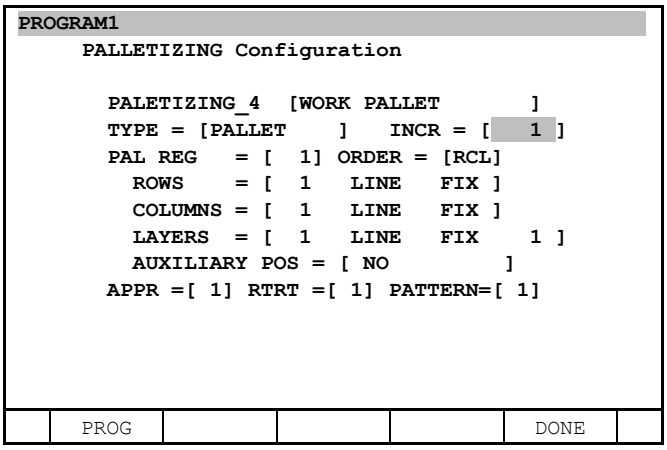
d After a comment has been completely entered, press the [ENTER] key.



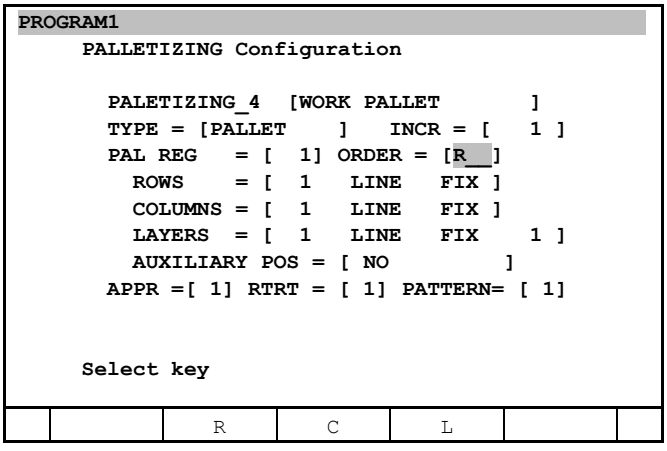
3 To select a palletizing type, move the cursor to the TYPE field, then select a function key.



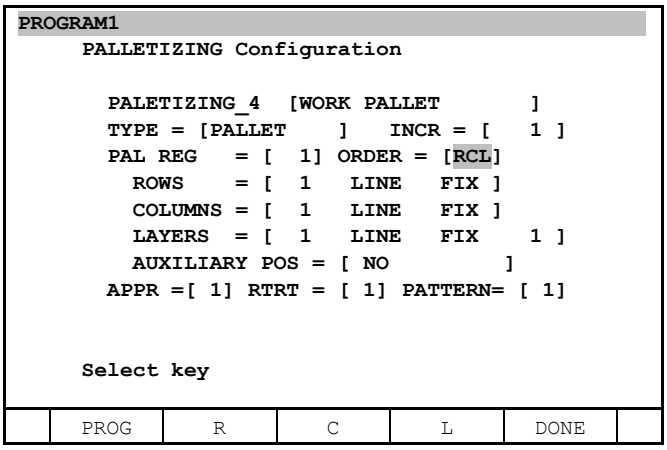
4 To enter a numeric value to the item INCR and PAL_REG, press a numeric key, then press the [ENTER] key.



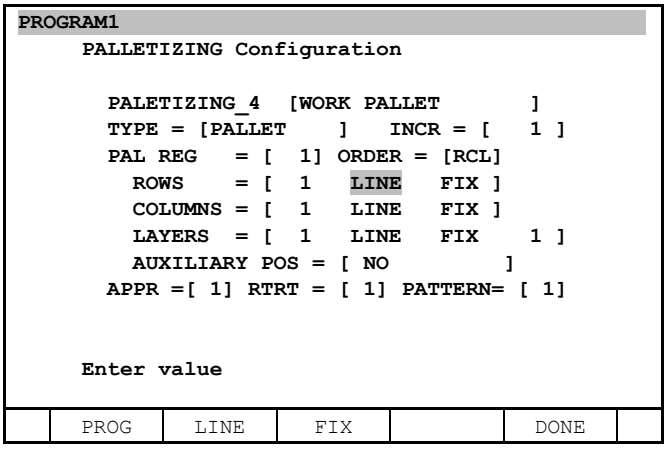
5 Specify a palletizing order by selecting the function keys in the target order.



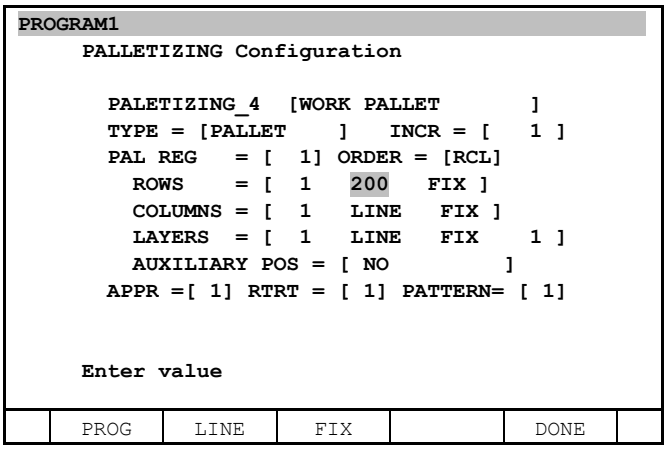
After the second item has been selected, the third item is determined automatically.



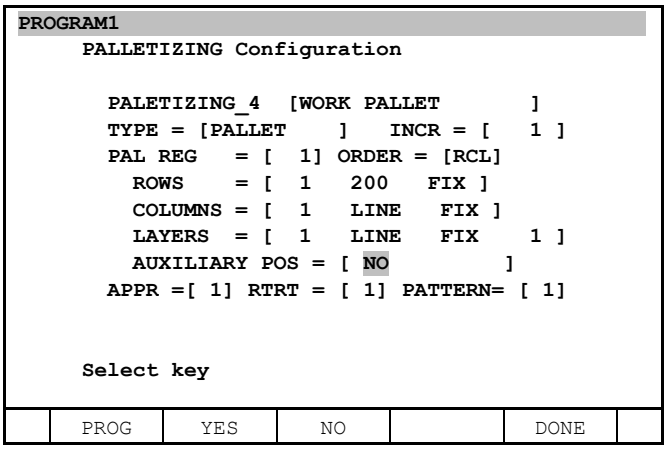
6 To specify the number of rows, columns and layers, press a numeric key, then press the [ENTER] key. To specify the type of posture control, move the cursor to the setting field and select the function key.



7 To specify the arrangement mode by the interval specification, position the cursor to arrangement mode field and input interval value. (The unit of interval is mm.)



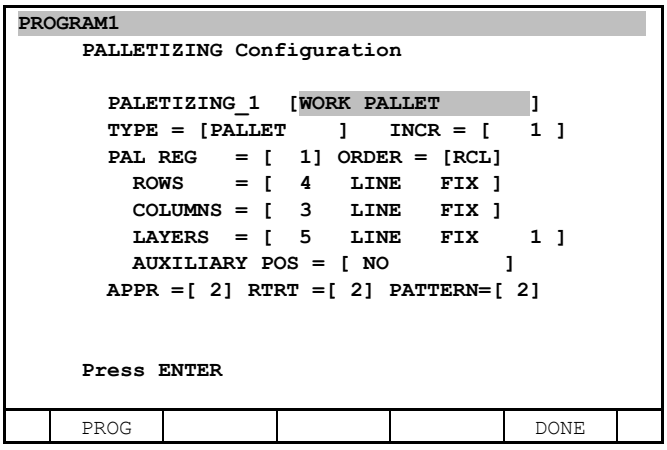
8 To specify whether to set auxiliary points, position the cursor at the auxiliary point field and select the desired function key menu.



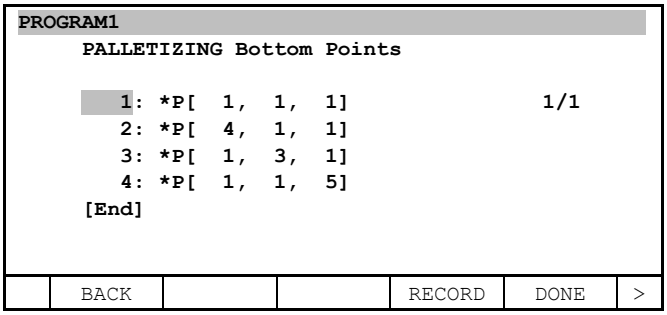
NOTE
When specifying the setting of auxiliary points, also select either of FIX/INTER.

9 Enter the approach point count and retraction point count.
10 To stop the initial data setting, press F1, PROG.

⚠ CAUTION
When the initial data setting is stopped before it is completed, the values set up to that time are invalidated.



11 When all data items have been entered, press F5, DONE. The initial data input screen then disappears and the palletizing stacking pattern teach screen will be displayed.



When the palletizing stacking pattern teach screen is displayed after the setting or changing of initial palletizing data is completed with F5, DONE, the palletizing register is initialized automatically. (See Subsection 64.4.1, “Palletizing Register”.)

64.3.3 Teaching a Stacking Pattern

On the palletizing stacking pattern teach screen, teach representative stack points of a stacking pattern. From these representative points, a target stack point is calculated automatically at the time of palletizing.

Stacking pattern teach screen

```
PROGRAM1
     PALLETIZING Bottom Points

      1:--P[  1,  1,  1]--              1/1
        2:--P[  4,  1,  1]--
        3:--P[  1,  3,  1]--
        4:--P[  1,  1,  5]--
     [End]

 |  | BACK |  |  | RECORD | DONE | > |
```

A list of the positions to be taught is displayed based on the initial palletizing data. Following this list, teach the positions of the representative stack points.

With or without an auxiliary position

For the stacking pattern without an auxiliary position, individually teach four tops of the stacking pattern.

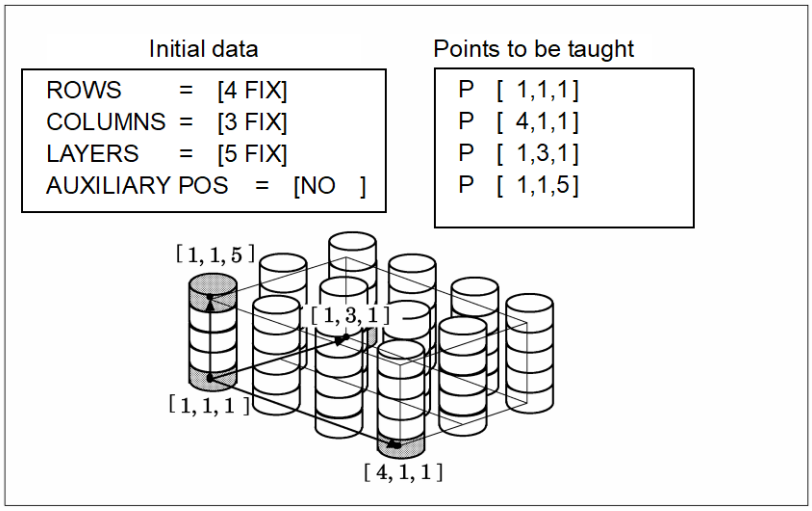


Fig. 64.3.3 (a) Stacking pattern with no auxiliary position

For a stacking pattern with an auxiliary position, when the shape of the first layer is a trapezoid, also teach the fifth position using the function provided.

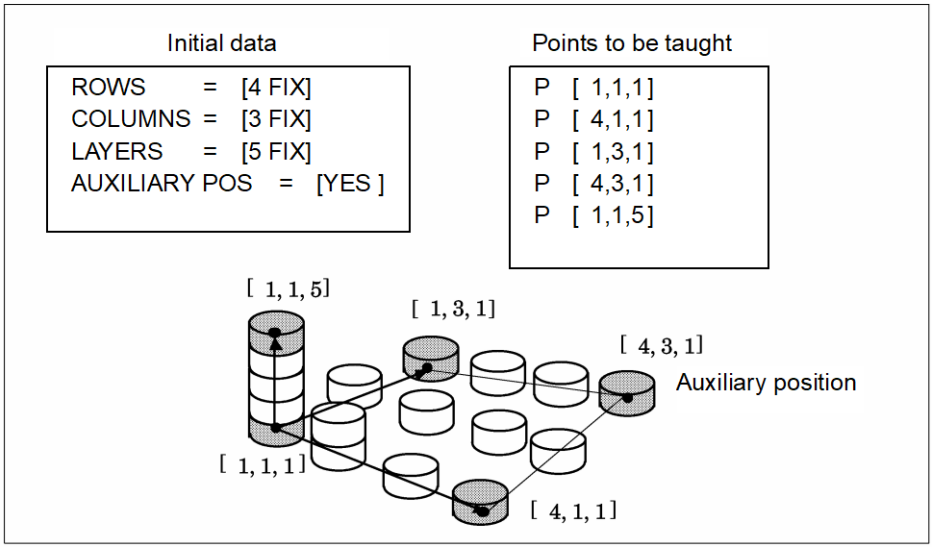


Fig. 64.3.3 (b) Stacking pattern with an auxiliary position

When selecting with auxiliary point, specify posture control type (fixed, split) at the auxiliary point. (E, EX only)

Types of arrangement modes/2-point teaching

When 2-point teaching is selected, teach the representative two points at both ends to set all the points in the row, column, and layer directions (standard).

NOTE
The following explanation is not relevant to palletizing B and BX. See Subsection 64.3.4.

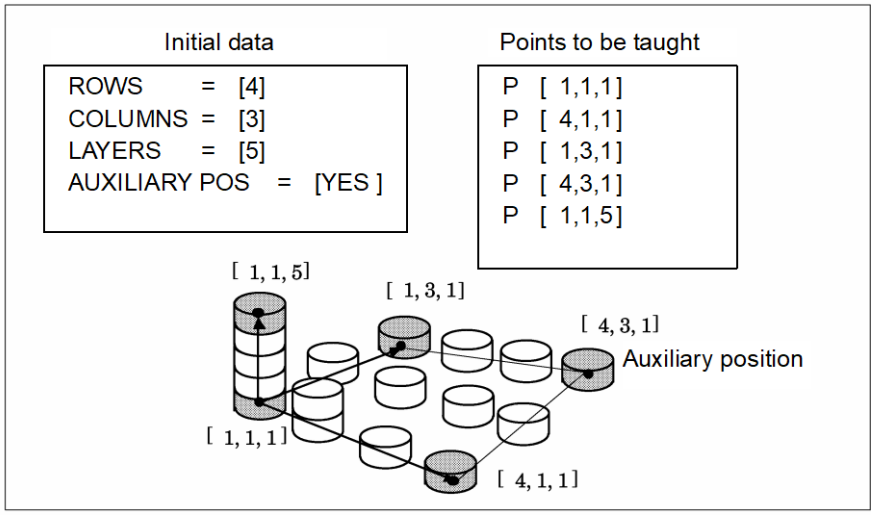


Fig. 64.3.3 (c) Teaching method by 2-point teaching

All-point teaching

When selecting all-point teaching, directly teach all the points in the row, column, and layer directions.

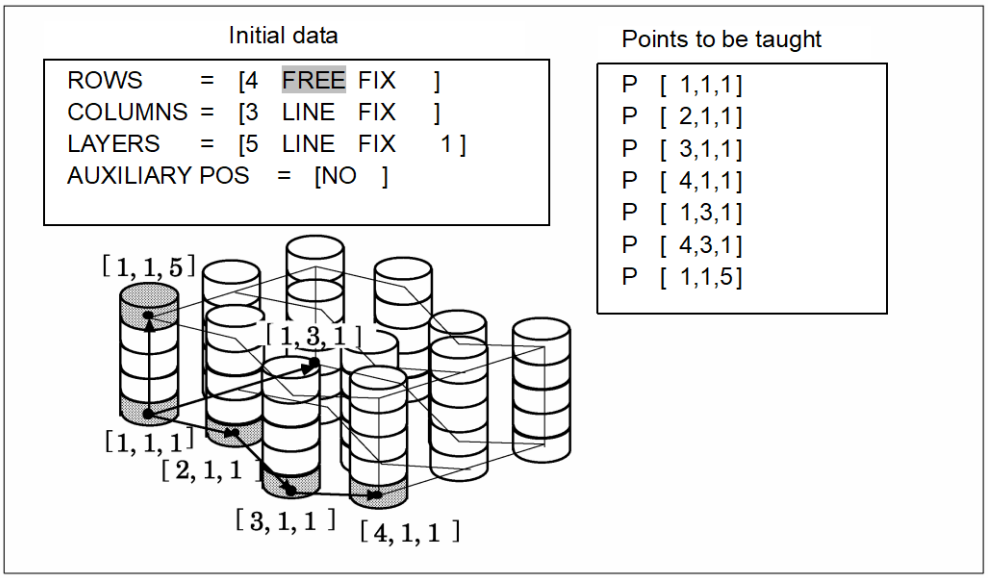


Fig. 64.3.3 (d) Teaching method by all-point teaching

Interval specification

When selecting the interval specification, specify the two points at both ends in each of the row, column, and layer directions, as well as the distance between workpieces, to set all points.

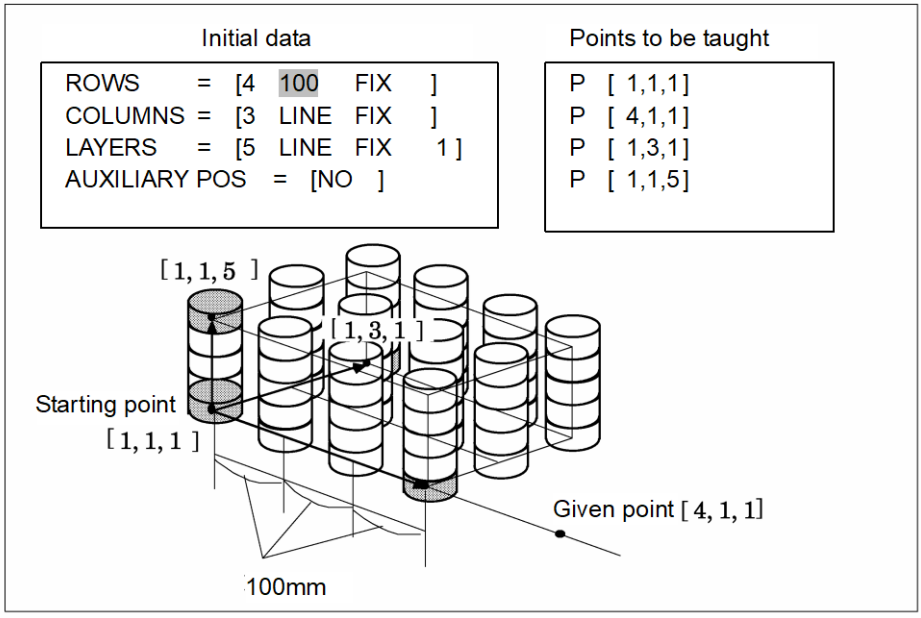


Fig. 64.3.3 (e) Teaching method by interval specification

Types of posture control

When the fixed posture is specified, workpieces at all the stack points always take the posture taught at point [1,1,1] (standard).

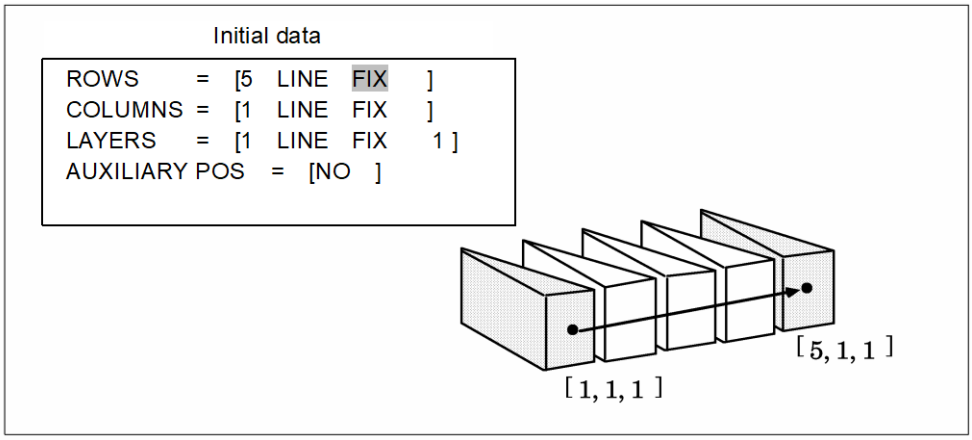


Fig. 64.3.3 (f) Postures of workpieces at stack points when the fixed posture is specified

For the split posture, when 2-point teaching is specified, workpieces take the postures obtained by splitting the postures taught at the two end points. When all-point teaching is specified, workpieces take the postures at the taught points.

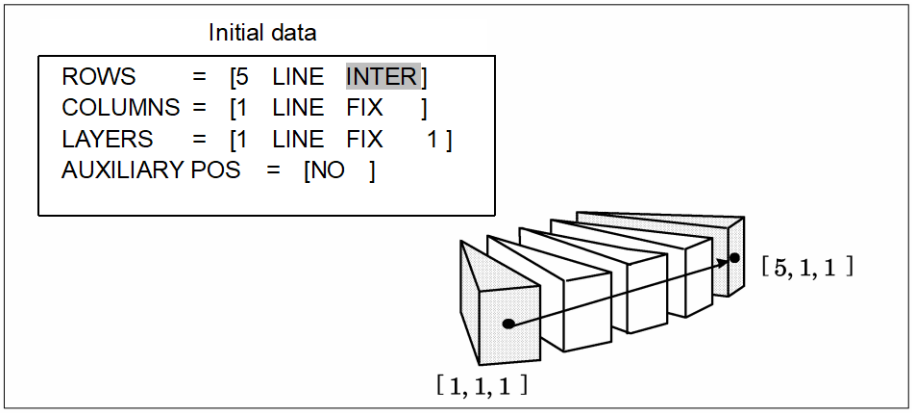


Fig. 64.3.3 (g) Postures of workpieces at stack points when the split posture is specified

Layer pattern count

To change the stack pattern every few layers, enter the number of layer patterns.
The layer pattern count is valid only when 2-point teaching is specified for the layer arrangement (for other cases, the layer pattern count is always 1).
For the first layer, the workpieces are always stacked at the stack points in layer pattern 1.
When the layer pattern count is N, the numbers of layers and layer patterns are the same until layer N. For layer (N+1) and beyond, layer patterns starting from layer pattern 1 are repeated.
Teach the position for the layer direction only in the teaching for layer pattern 1.
The position for the layer direction in each layer is calculated by the taught position for layer pattern1.

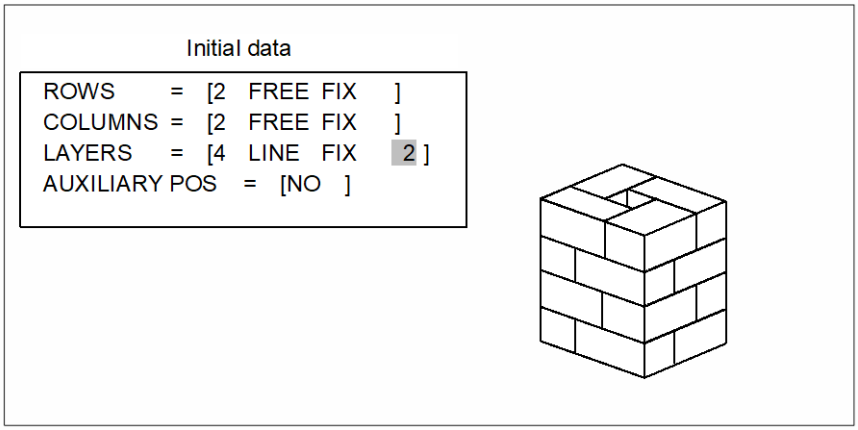


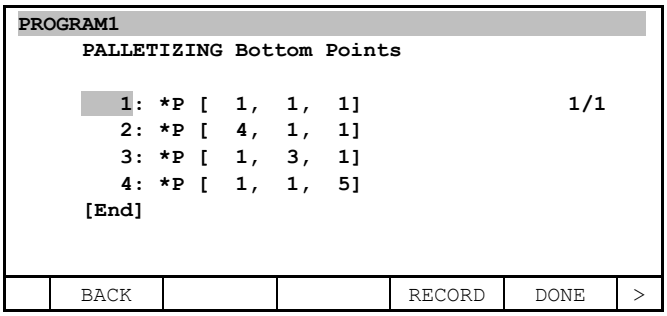
Fig. 64.3.3 (h) Posture at the stacking point in division posture

When the total number of layers is less than 16, a number not greater than the total number can be set for the layer pattern count. If a number less than the layer pattern count is subsequently specified for the number of layers, the layer pattern count is automatically changed to the number of layers.

Procedure 64-3 Teaching a palletizing stacking pattern

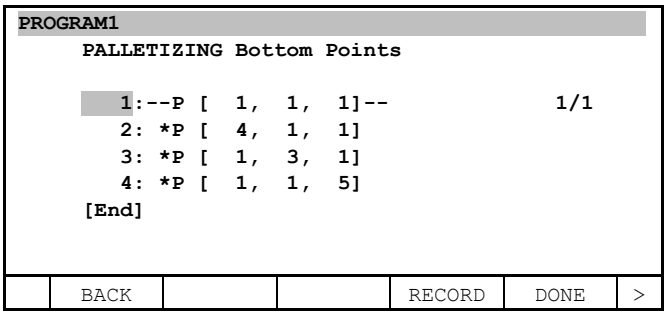
Step

1 Based on the initial data setting, a list of stack points to be taught is displayed.



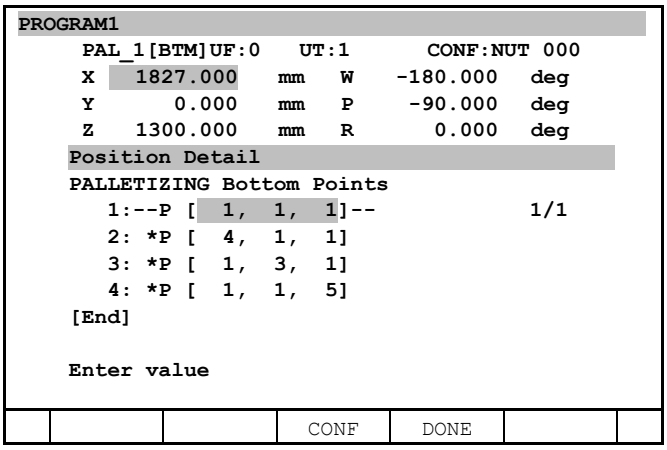
NOTE
The number of representative stack points to be recorded depends on the numbers of rows, columns, and of layers set on the initial data input screen. In the above sample screen, four rows, three columns, and five layers are set. For each point, row, column, and then layer numbers are specified.

2 Move the robot by jog feed to a position which is to be taught as a representative stack point.
3 Place the cursor on the appropriate line, and press F4, RECORD while holding down the [SHIFT] key. The current robot position is then recorded.

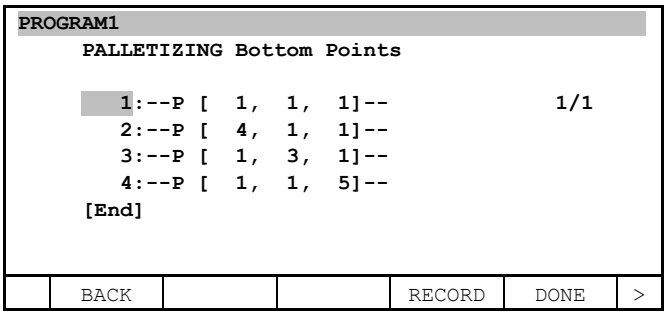


The positions not yet taught are marked with an asterisk. The positions already taught are marked with "-".

4 To display detailed position data for a taught representative stack point, move the cursor to the target stack point number, then press F5, POSITION. The detailed position data is then indicated.



Numeric values can also be entered directly to specify position data. To return to the previous stacking pattern teach screen, press F4, DONE.

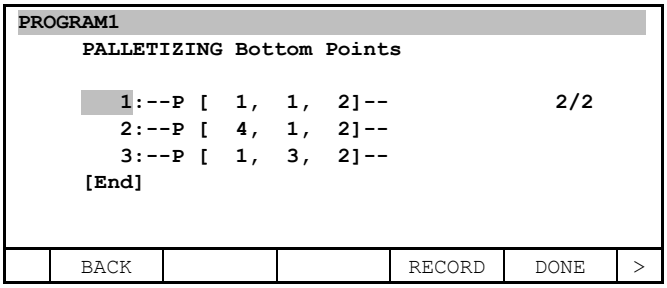


5 Pressing [FWD] key while holding down the [SHIFT] key causes the robot to move to the representative stack point indicated by the cursor. This operation can be performed to confirm the taught point.

6 To return to the previous initial data teach screen, press F1, BACK.

7 Press F5, DONE to display the path pattern condition setting screen (BX or EX) or path pattern teaching screen (B or E) (Subsection 64.3.4 or 64.3.5).

NOTE
When layer patterns are used (E or EX) and F5, DONE is pressed, the screen for specifying the stack pattern for the next layer will be displayed.



64.3.4 Setting Path Pattern Conditions

The screen explained in this section is not displayed for palletizing B or E. (See Subsection 64.3.5).
The palletizing path pattern condition setting screen is used to set conditions indicating which path pattern is to be used for each stack point in advance when multiple path patterns are to be set on the path pattern teaching screen (See Subsection 64.3.5).
For palletizing BX and EX, multiple path patterns can be set independently for stack points. For palletizing B and E, this screen is not displayed because only one path pattern can be set.

Path pattern condition setting screen

```
PROGRAM1
     PALLETIZING Route Patterns

      PTN [ 1] = [     * ,  3- 1 ,     * ]
      PTN [ 2] = [     * ,  3- 2 ,     * ]
      PTN [ 3] = [     * ,  3- 0 ,     * ]
     [End]


     Enter value
```

	BACK		DIRECT	MODULO	DONE	>

To specify a path for each stack point, as many path patterns as required must be specified when initial data is set. For each path pattern, set a path pattern condition.

In this example, pattern 1 is used for the stack points in column 1, pattern 2 is used for the stack points on column 2, and pattern 3 is used for the stack points in column 3.

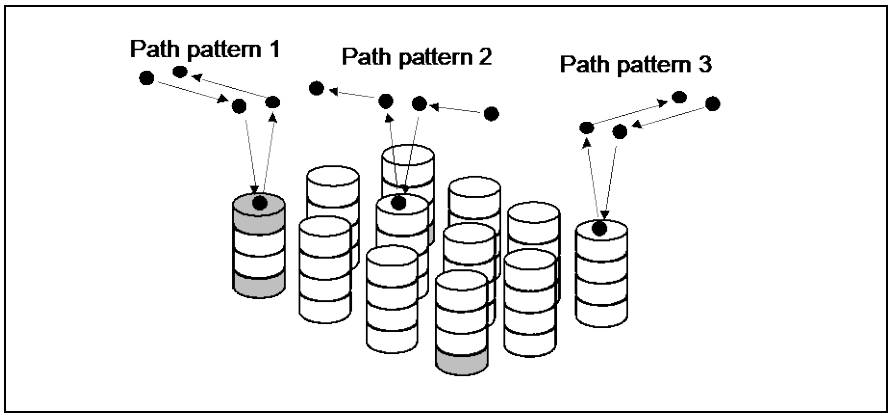


Fig. 64.3.4 (a) Palletizing using three path patterns

How to use path pattern conditions

- The execution of palletizing uses the path pattern with the condition number for which the row, column, and layer numbers at the stack point match the row, column, and layer (element) values of a path pattern condition.
- In direct specification mode, specify numbers from 1 to 127 for a stack point. An asterisk indicates an arbitrary stack point.
- In the remainder specification mode, specify path pattern condition element ”m-n” using a remainder system for a stack point.
  Layer element ”3-1” indicates a layer corresponding to a stack point value for which a remainder of 1 is obtained by dividing the value by 3.
- If the current stack point corresponds to no path pattern condition, an alarm occurs. If the current stack point corresponds to two or more path pattern conditions, a path pattern condition is used according to the following conditions:

a. A path pattern condition specified in direct specification mode is used.
b. When two or more path pattern conditions are specified in direct specification mode, a path pattern condition specified in the remainder specification mode is used. When two or more path pattern conditions are specified in remainder specification mode, a path pattern condition in which the greatest value is specified for m is used.
c. When two or more path pattern conditions satisfy conditions a and b above, the path pattern condition having the smallest path pattern condition number is used.
The following shows the priority among the sample path pattern conditions:

```
Example  PTN    [1] = [ *   ,  1    ,  2    ]
         PTN    [2] = [ *   ,  *    ,  2    ]
         PTN    [3] = [ *   ,  3–2  ,  4–1  ]
         PTN    [4] = [ *   ,  *    ,  4–1  ]
         PTN    [5] = [ *   ,  *    ,  2–1  ]
         PTN    [6] = [ *   ,  *    ,  *    ]
```

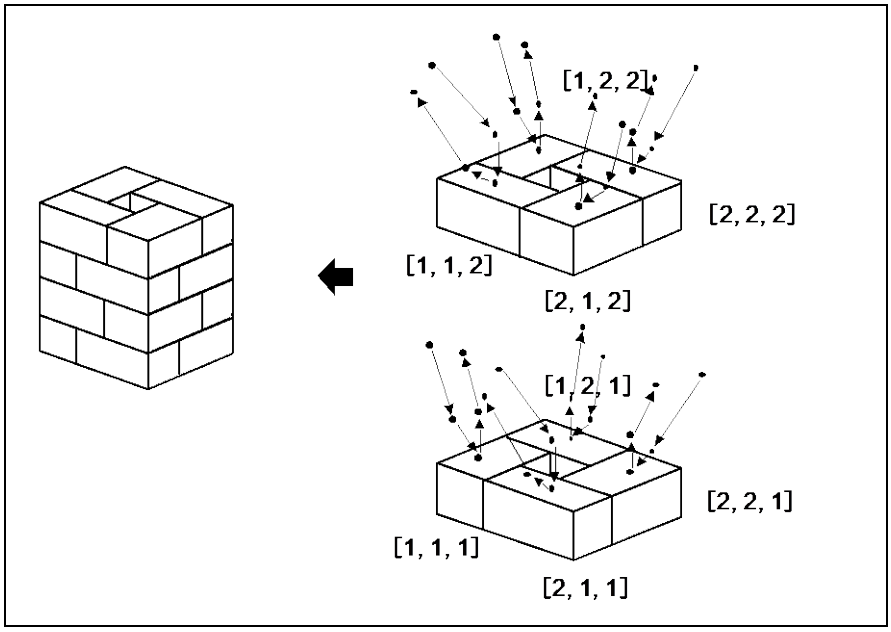


Fig. 64.3.4 (b) Box palletizing using eight path patterns

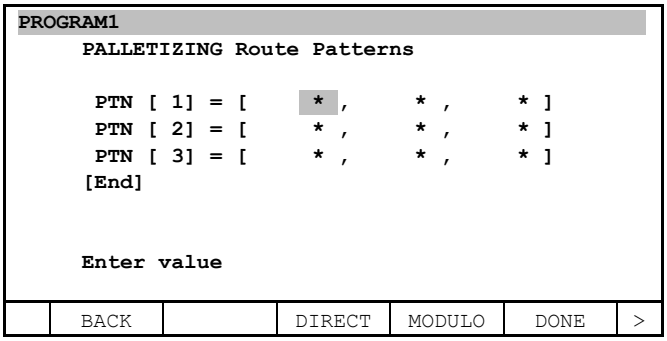
```
Example  PTN    [1] = [ 1   ,  1    ,  2–1  ]
         PTN    [2] = [ 2   ,  1    ,  2–1  ]
         PTN    [3] = [ 1   ,  2    ,  2–1  ]
         PTN    [4] = [ 2   ,  2    ,  2–1  ]
         PTN    [5] = [ 1   ,  1    ,  2–0  ]
         PTN    [6] = [ 2   ,  1    ,  2–0  ]
         PTN    [7] = [ 1   ,  2    ,  2–0  ]
         PTN    [8] = [ 2   ,  2    ,  2–0  ]
```

In this example, eight path patterns are defined and repeated for every two layers because different paths must be set according to the box position.

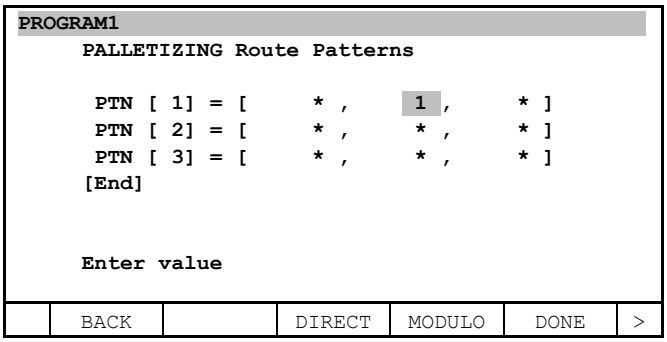
Procedure 64-4 Setting palletizing path pattern conditions

Step

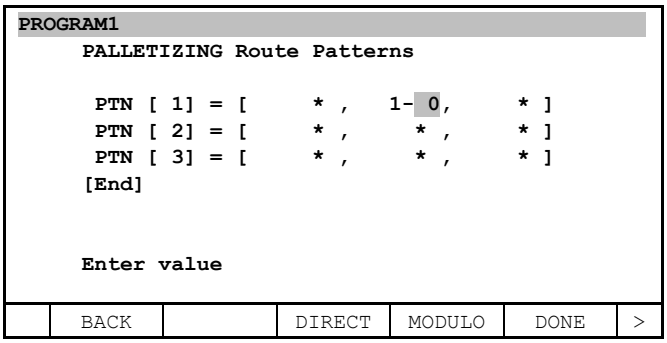
1 Condition items to be entered are displayed according to the value set for the pattern count as initial data.



2 In direct specification mode, position the cursor to the point to be changed and enter a numeric value. To specify an asterisk (*), enter zero.



3 In remainder specification mode, press F4, MODULO. The target item is divided into two sub-items. Enter a value for each sub-item.



4 To specify values in direct specification mode, press F3, DIRECT.
5 Press F1, BACK to display the previous stack point teaching screen again.
6 Press F5, DONE to display the next path pattern teaching screen.

```
PROGRAM1
    PALLETIZING Route Points

          IF PL[ 1]=[*,*,*]              1/1
      1:Joint *P [A_2 ] 30%        FINE
        2:Joint *P [A_1 ] 30%        FINE
        3:Joint *P [BTM ] 30%        FINE
        4:Joint *P [R_1 ] 30%        FINE
        5:Joint *P [R_2 ] 30%        FINE
    [End]


    Teach Route Points

 | BACK | POINT |  | RECORD | DONE | >
```

64.3.5 Teaching a Path Pattern

On the palletizing path pattern teach screen, set several path points which are passed before and after a workpiece is stacked (or unstacked) at a stack point. The path points change depending on the position of the stack point.

Path pattern teach screen

```
PROGRAM1
    PALLETIZING Route Points

          IF PL[ 1]=[*,*,*]              1/3
      1:Joint *P [A_2 ] 30%        FINE
        2:Joint *P [A_1 ] 30%        FINE
        3:Joint *P [BTM ] 30%        FINE
        4:Joint *P [R_1 ] 30%        FINE
        5:Joint *P [R_2 ] 30%        FINE
    [End]


    Teach Route Points

 | BACK | POINT |  | RECORD | DONE | >
```

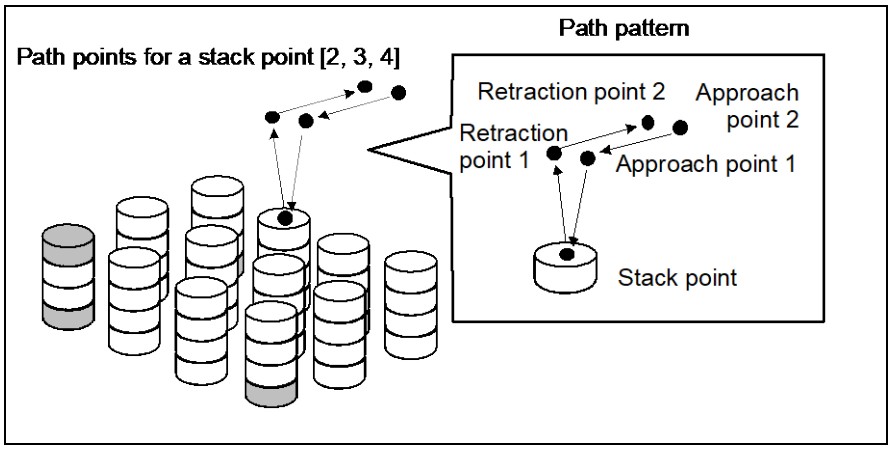
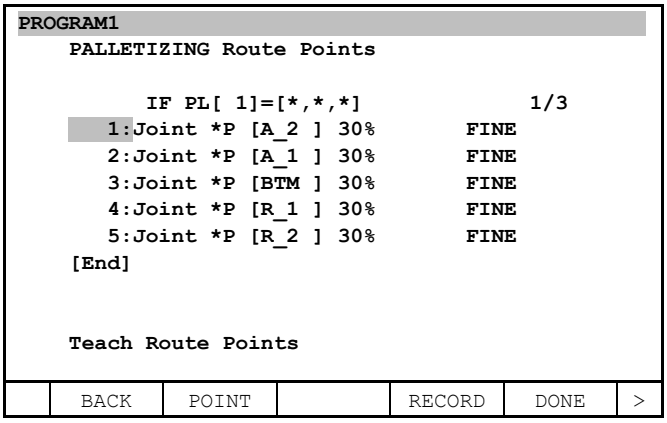


Fig. 64.3.5 Palletizing path

Procedure 64-5 Teaching a palletizing path pattern

Step

1 Based on the initial data setting, a list of the path points to be taught will be displayed.



NOTE

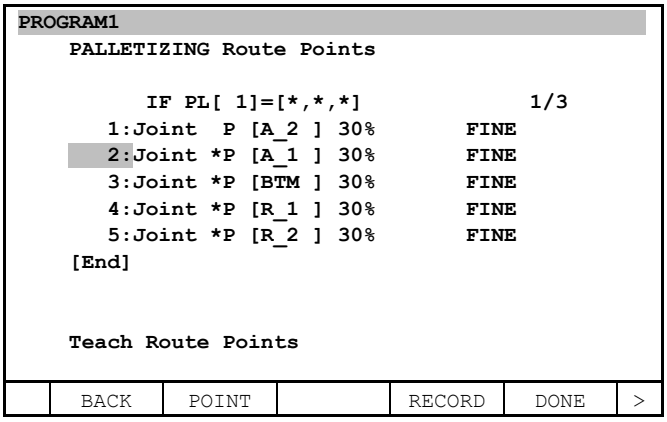
The number of path points to be recorded depends on the number of approach points and the number of retraction points set on the initial data input screen. In this sample screen, the number of approach points is 2 and the number of retraction points is 2.

2 Move the robot by jog feed to a position which is to be taught as a path point.

3 Move the cursor to the setting field to be taught and teach the position using one of the following operations.

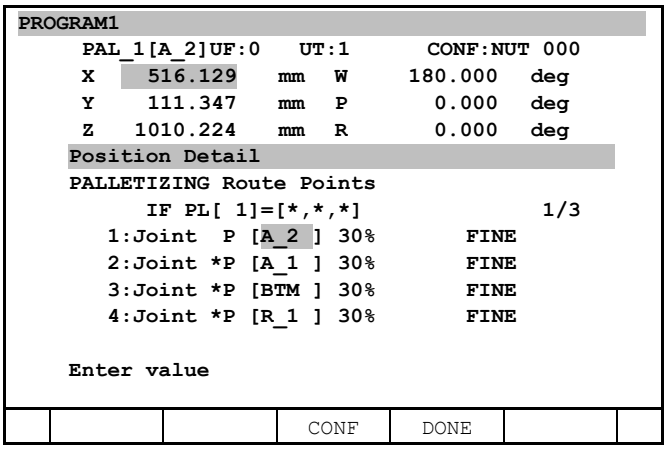
a Press F2, POINT while holding down the [SHIFT] key. When pressing F2, POINT, without pressing and holding the [SHIFT] key, default logical motion menu will be displayed and then you can set the motion type or feed rate, etc. (This key is displayed only at teaching route pattern 1.)

b Press and hold the [SHIFT] key and press F4, RECORD.

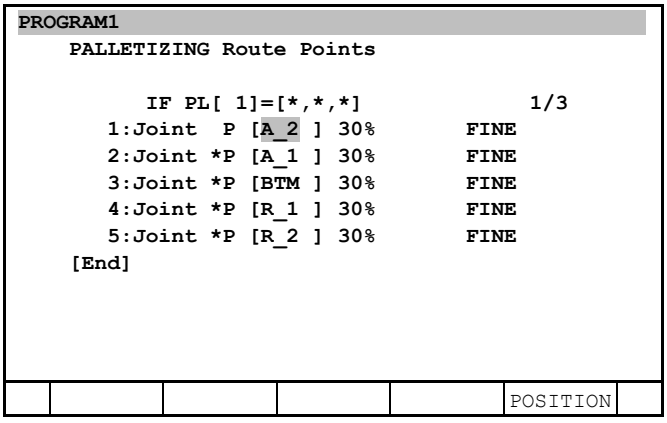


The positions not yet taught are marked with an asterisk *.

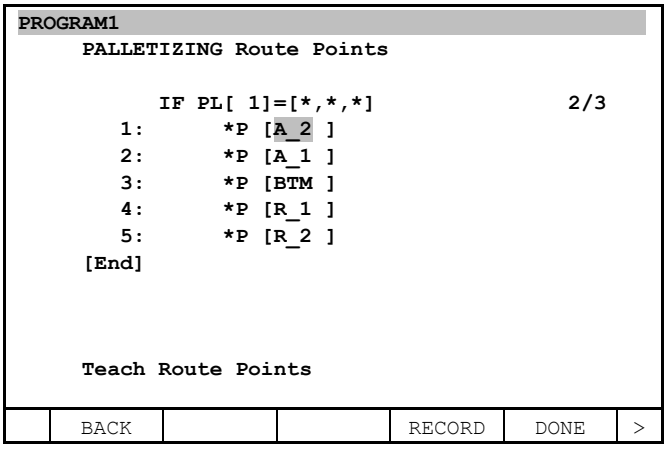
4 To display detailed position data for a taught path point, move the cursor to the target path point number, and press F5, POSITION. The detailed position data is then indicated.



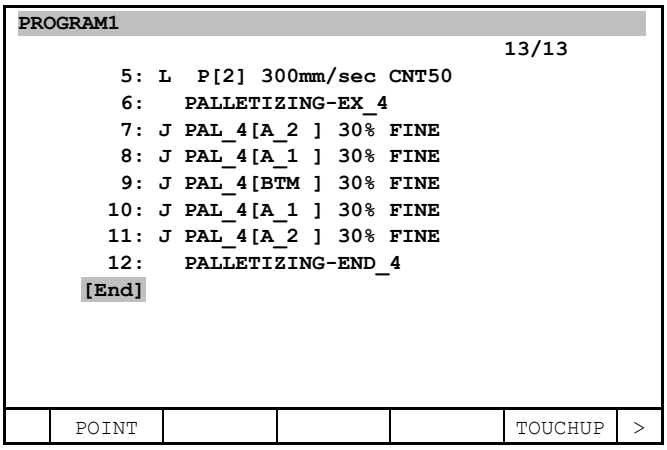
Numeric values can also be entered directly to specify position data. To return to the previous path pattern teach screen, press F4, DONE.



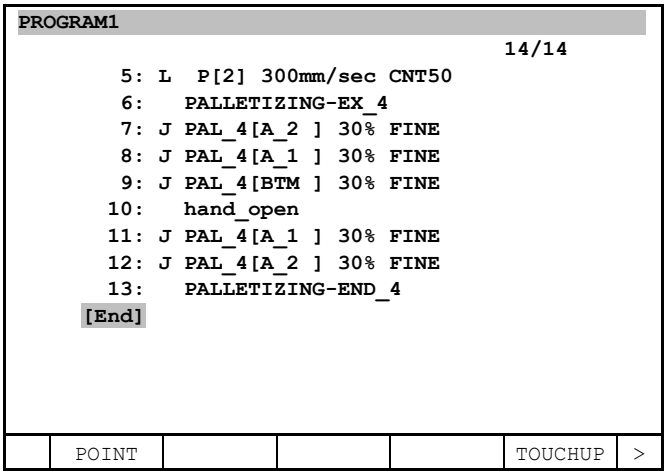
5 Pressing [FWD] key while holding down the [SHIFT] key causes the robot to move to the path point indicated by the cursor. This operation can be performed to confirm the taught point.
6 To return to the stacking pattern teach screen, press F1, BACK.
7 Press F5, DONE to display the next path pattern teaching screen. If the path pattern is only one pattern, go to the step 9.



8 Press F1, BACK to specify the previous path pattern. Press F5, DONE to specify the next path pattern.
9 After teaching of all the path patterns is complete, press F5, DONE to exit from the palletizing edit screen and display the program screen again. The palletizing instruction is automatically written in the program.



10 Editing, such as modifying a hand instruction executed at a stack position or the motion format at a path point can be performed on this program screen, in the same way as for normal programs.



For details of palletizing programs, see Section 64.4, “Executing the Palletizing Function”.

64.3.6 Notes on Teaching the Palletizing Function

- The palletizing function is enabled only when a program contains these three instructions: A palletizing instruction, palletizing motion instruction, and palletizing end instruction. When just one of the three instructions is taught into a subprogram by another operation such as copying, normal operation cannot be performed. When changing the TP program, re-teach the palletizing instruction after deleting the instructions.
- When all palletizing data has been taught, palletizing numbers are automatically written together with the instructions (a palletizing instruction, palletizing motion instruction, and palletizing end instruction). The user need not be concerned about the duplication of these numbers in other programs. (Each program has its own data for palletizing numbers.)
- In the palletizing motion instruction, C (circular motion), A (circle arc motion), and S (spline motion) cannot be specified as the motion format.
- When palletizing, with a system with extended axes, there are some special conditions. For a system with extended axes, refer to Section 64.6 ”Palletizing Function with extended axes”.
- When the position data in bottom points and route points are taught, user frame is not used. World frame is always used in teaching these points, and user frame number is the position data of these points is always 0.
- Do not add the time before function to the palletizing motion instruction. Palletizing motion instruction may not function correctly.

- Programs including palletizing instructions cannot use the coordinate system change shift function.

64.4 EXECUTING THE PALLETIZING FUNCTION

Palletizing program

The execution of the palletizing function is shown below.

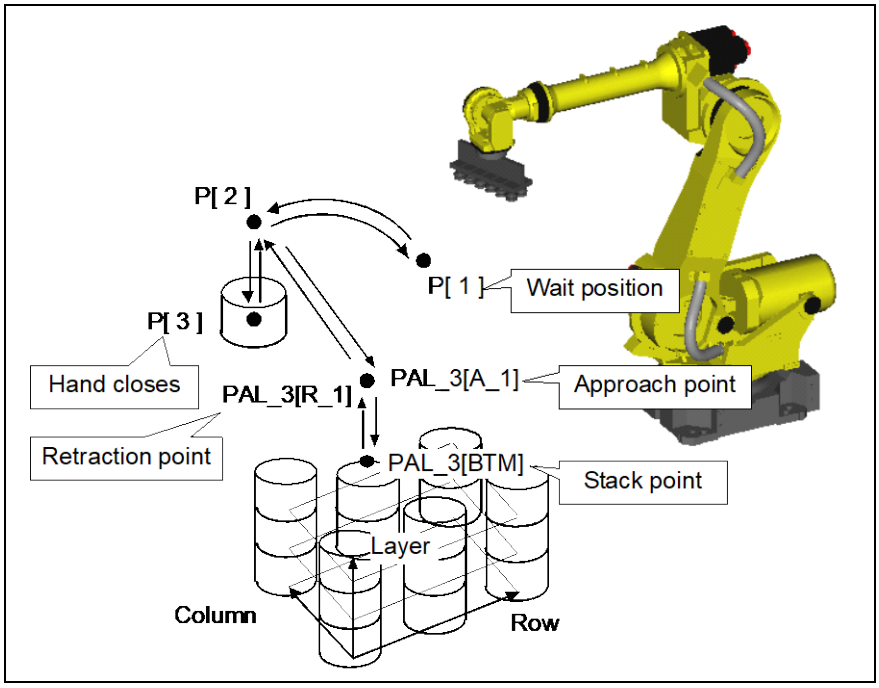


Fig. 64.4 (a) Example of palletizing

Example

```
 5: J  P[1] 100% FINE
 6: J  P[2] 70% CNT50
 7: L  P[3] 50mm/sec FINE
 8:    hand close
 9: L  P[2] 100mm/sec CNT50
10:   PALLETIZING-B_3
11: L PAL_3[ A_1 ] 100mm/sec CNT10
12: L PAL_3[ BTM ] 50mm/sec FINE
13:   hand open
14: L PAL_3[ R_1 ] 100mm/sec CNT10
15:   PALLETIZING-END_3
16: J  P[2] 70% CNT50
17: J  P[1] 100% FINE
```

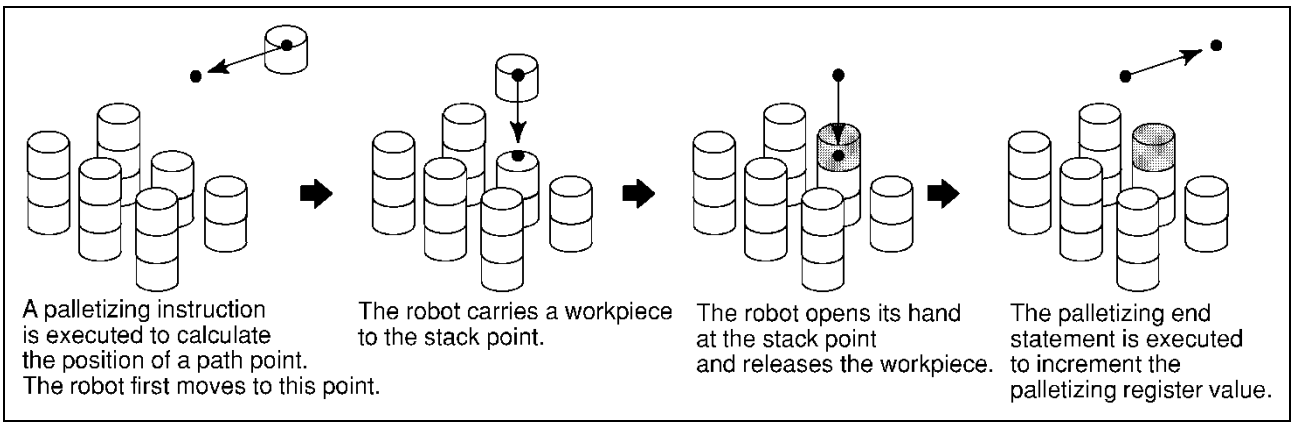


Fig. 64.4 (b) Workpiece stacking process

64.4.1 Palletizing Register

The palletizing register manages the position of the current stack point. When a palletizing instruction is executed, the value held in the palletizing register is referenced, and the actual stack point and path points are calculated. (Please see chapter of "6.16 PALLETIZING REGISTER" in R-50iA/R-50iA Mate CONTROLLER OPERATOR'S MANUAL (Basic Operation) (B-84464EN))

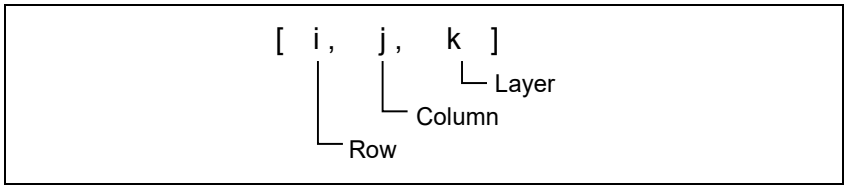


Fig. 64.4.1 (a) Palletizing register

The palletizing register indicates the row, column, and layer which are used for calculating the position of the stack point when the palletizing instruction is executed.

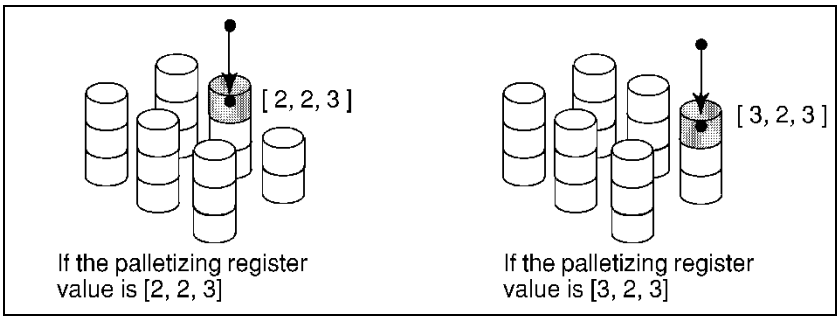


Fig. 64.4.1 (b) Relationship between the palletizing register and stack point

Updating the Palletizing Register

The palletizing register is incremented (or decremented) by executing the palletizing end instruction. The increment (decrement) method is determined depending on the initial data setting.
For 2-row, 2-column, and 2-layer palletizing with ORDER = [RCL] specified, executing the palletizing end instruction changes the palletizing register as follows:

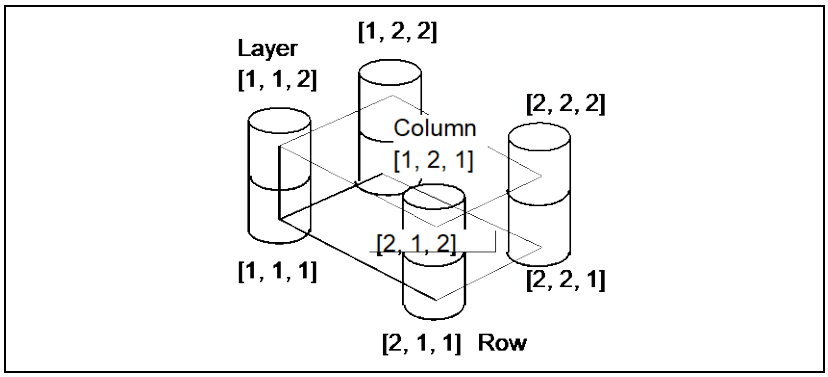


Fig. 64.4.1 (c) Example of 2-row, 2-column, and 2-layer palletizing

Table 64.4.1 (a) Incrementing (Decrementing) order of the palletizing register

	TYPE = [PALLET]		TYPE = [DEPALLET]	
	INCR = [1]	INCR = [-1]	INCR = [1]	INCR = [-1]
Initial value	[1, 1, 1]	[2, 2, 1]	[2, 2, 2]	[1, 1, 2]
↓	[2, 1, 1]	[1, 2, 1]	[1, 2, 2]	[2, 1, 2]
↓	[1, 2, 1]	[2, 1, 1]	[2, 1, 2]	[1, 2, 2]
↓	[2, 2, 1]	[1, 1, 1]	[1, 1, 2]	[2, 2, 2]
↓	[1, 1, 2]	[2, 2, 2]	[2, 2, 1]	[1, 1, 1]
↓	[2, 1, 2]	[1, 2, 2]	[1, 2, 1]	[2, 1, 1]
↓	[1, 2, 2]	[2, 1, 2]	[2, 1, 1]	[1, 2, 1]
↓	[2, 2, 2]	[1, 1, 2]	[1, 1, 1]	[2, 2, 1]
↓	[1, 1, 1]	[2, 2, 1]	[2, 2, 2]	[1, 1, 2]

Initializing the palletizing register

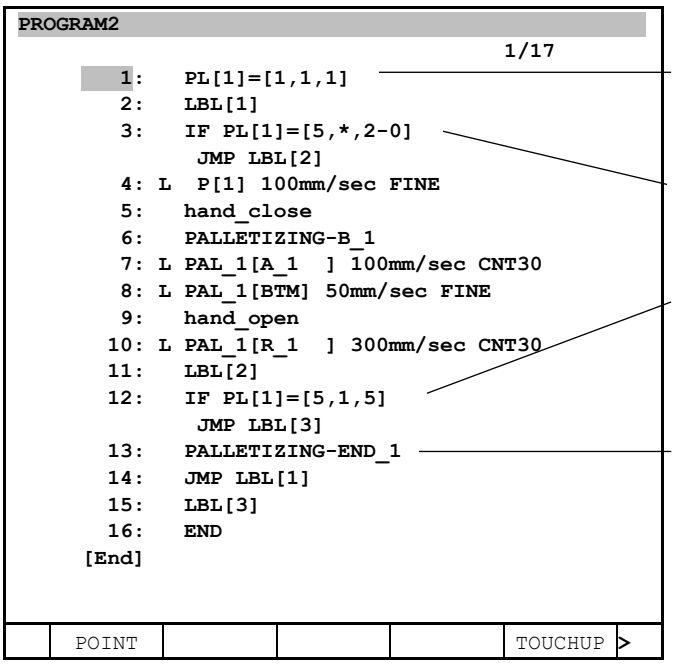
When F5, DONE is pressed upon completion of the setting or changing of initial palletizing data, the palletizing stacking pattern teach mode is set. At this time, the palletizing register is initialized automatically. (See Subsection 64.3.2, “Inputting Initial Data”.)

Table 64.4.1 (b) Initial value of the palletizing register

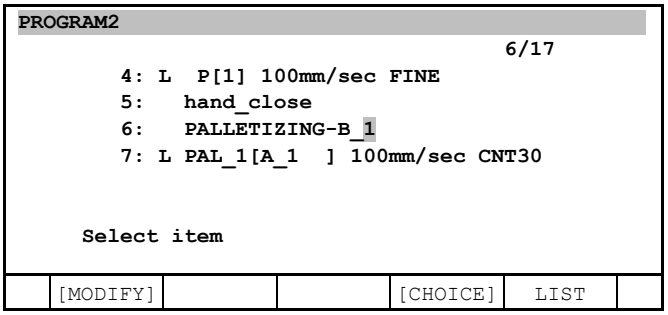
Initial data		Initial value		
TYPE	INCR	ROWS	COLUMNS	LAYERS
PALLET	Positive value	1	1	1
	Negative value	Total number of rows	Total number of columns	1
DEPALLET	Positive value	Total number of rows	Total number of columns	Total number of layers
	Negative value	1	1	Total number of layers

64.4.2 Controlling the Palletizing Function by a Palletizing Register

In 5-row, 1-column, and 5-layer palletizing, suppress stack operation for the fifth workpiece in each even-numbered layer. (Stack five workpieces in odd-numbered layers, and stack four workpieces in even-numbered layers.)

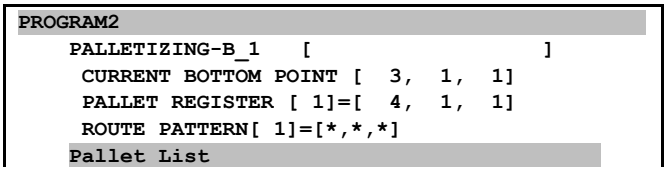


Procedure 64-6 Displaying the palletizing status



Step

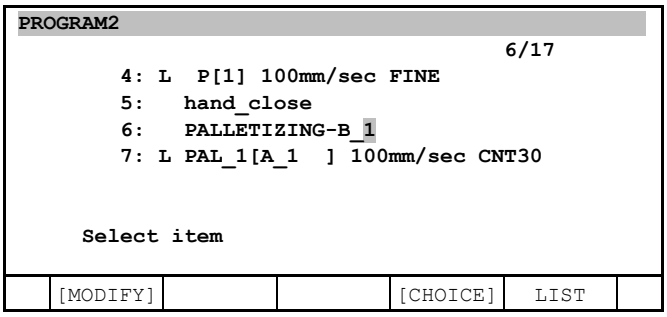
1 To display the palletizing status, place the cursor on the palletizing instruction, then press F5, LIST. The current stack point and the value of the palletizing register are displayed.



64.5 MODIFYING THE PALLETIZING FUNCTION

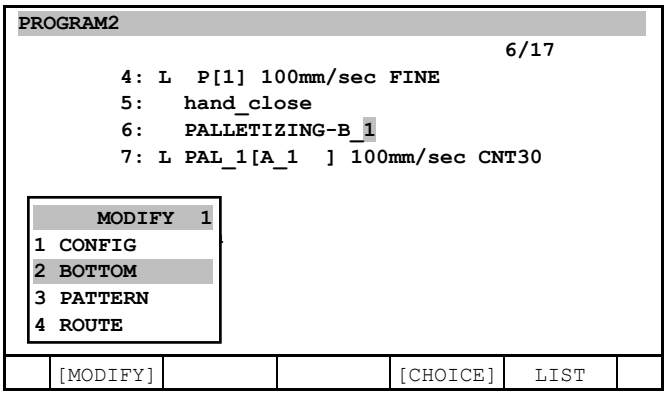
The palletizing data and palletizing instructions which were taught can be modified later.

Procedure 64-7 Modifying palletizing data

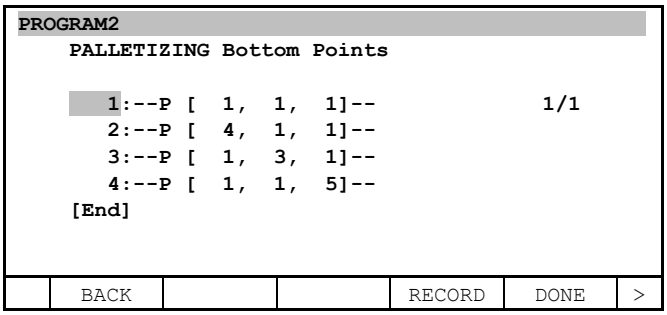


Step

1 Place the cursor on the palletizing instruction which is to be modified, then press F1, [MODIFY] to display the modification menu.



2 Select a target palletizing edit screen in the modification menu.

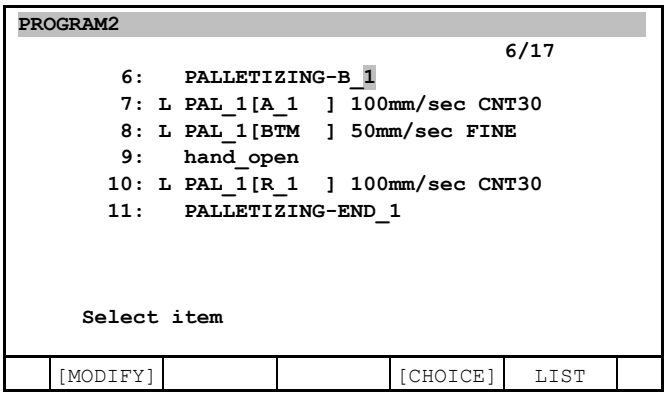


To return to the previous palletizing edit screen, press F1, BACK.
To proceed to the next palletizing edit screen, press F5, DONE.

NOTE
After palletizing data has been modified, the normal edit screen can be called from any palletizing screen. In this case, the new data after modification is kept valid.

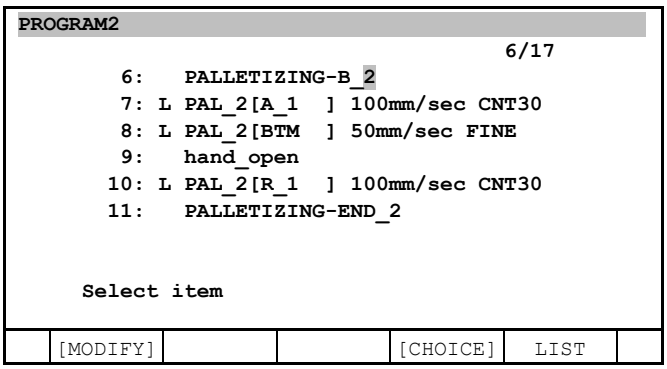
3 When the modification is completed, press NEXT “>” to display the next page. Then press F1, PROG.

Procedure 64-8 Changing the palletizing number



Step

1 Place the cursor on the palletizing instruction having the palletizing number which is to be changed, then enter a new number.



At the same time the palletizing number of the palletizing instruction is changed, the palletizing numbers of the palletizing motion and palletizing end instructions are also changed.

> ⚠ **CAUTION**
> When changing palletizing numbers, make sure that the new numbers are not used by other palletizing instructions.

64.6 PALLETIZING FUNCTION WITH EXTENDED AXES

This section explains the special items when using palletizing on a system with extended axes.
When teaching the bottom point or route points for palletizing, which are different from the usual teaching of motion instruction, the position which is removed the position of extended axes is recorded.

Execution

When palletizing is executed, palletizing will be done at the position of the extended axes at that time. (The robot does not automatically return to the position of the extended axes at teaching bottom points or route points.) For example, on a system with a linear extended axis, when the palletizing instruction is executed at a point 1000 mm away from the position at teaching palletizing, the robot will perform the same motion as taught at the position which is 1000 mm away from the teaching position.

In case that the skip instruction is taught in the motion instruction immediately before the palletizing instruction, when the skip condition is satisfied during the execution of that motion instruction, the palletizing is executed at the position of the extended axis when the skip condition is satisfied.

Position teaching/modifying

When teaching the bottom/route points for palletizing on a system with extended axes or modifying the position, the following attention is necessary.

- When teaching bottom/route points for palletizing, it must be done at the fixed point.
  - When the position of extended axes shifts while teaching the bottom/route points for the same palletizing, the robot will follow a different path then the taught path when the program is executed.
- When modifying the position, move the robot to the position of the extended axes at teaching bottom/route points before modifying.

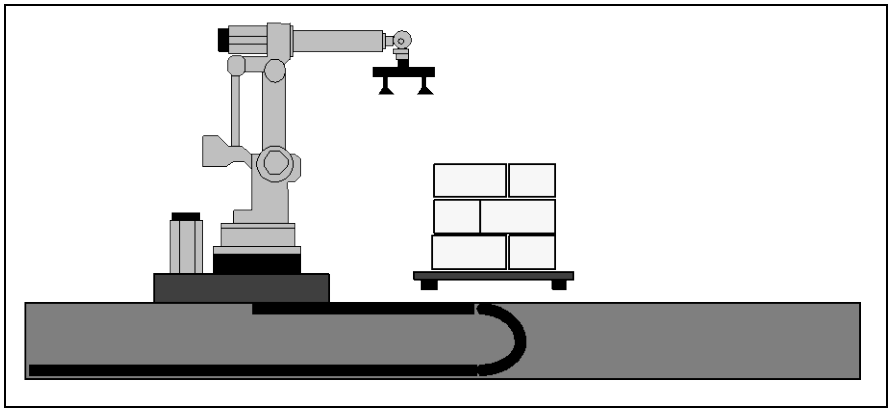


Fig. 64.6 Palletizing function with extended axes

64.7 PALLETIZING ALL-POINT TEACHING

Palletizing all-point teaching allows the stacking (or unstacking) of workpieces without changing the taught configuration.

Operation

Set the following values to stack (or unstack) workpieces without changing the taught posture and configuration:

1 On the palletizing initial data screen, set INTER for posture control in the row, column, or layer direction for which FREE is specified as the arrangement mode.

For each taught workpiece in the specified direction, all workpieces corresponding to the taught workpiece are stacked (or unloaded) with the same posture and configuration as that for the taught workpiece.

The following shows an example of palletizing for an irregular arrangement of four rows, two columns, and five layers.

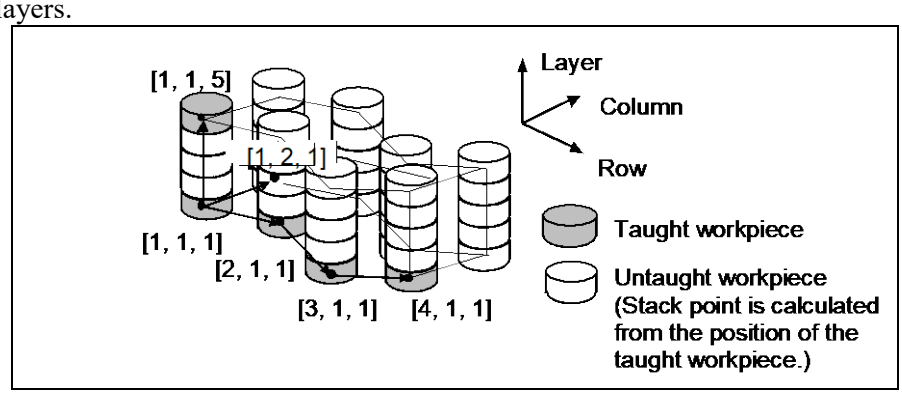


Fig. 64.7 Palletizing all-point teaching

The following palletizing initial data is set:

```
ROWS=        [4    FREE   INTER]
COLUMNS  =   [2    LINE   FIX   ]
LAYERS   =   [5    LINE   FIX   1]
```

In this example, FREE and INTER are set in the row direction. The following configurations are used:

- Configuration for stacking (unloading) workpieces on row 1: Configuration in P[1,1,1]
- Configuration for stacking (unloading) workpieces on row 2: Configuration in P[2,1,1]
- Configuration for stacking (unloading) workpieces on row 3: Configuration in P[3,1,1]
- Configuration for stacking (unloading) workpieces on row 4: Configuration in P[4,1,1]

Notes

Note the following points when using this function:

1 FREE and INTER can be set at the same time in only one of the row, column, and layer directions.

This is because if FREE and INTER are set at the same time in two or more of the row, column, and layer directions, two or more configurations to be taken at the position of an untaught workpiece (workpiece for which the stack point is calculated from the position of a taught workpiece) are made.

If such a setting is made in a program, the program causes the error indicated by PALT-024 Calculation error occurred and cannot be executed.

2 Carefully teach a program so that the program is not stopped due to a configuration mismatch alarm.

If the configuration in the current position differs from the configuration data for the destination position, the robot cannot move in Linear operation mode (a configuration mismatch alarm occurs and execution of the program is stopped).

The configuration at a stack point is used for the configuration at an approach or retraction point during palletizing. Therefore, if the pallet operation instruction to be executed first is in Linear operation mode, a configuration mismatch may occur depending on the configuration of the robot when an attempt is made to execute the line.

To avoid such a problem, specify Joint for the operation mode of the first pallet operation instruction.

For example, to avoid a configuration mismatch alarm, the following programming can be used for palletizing with three approach points and two retraction points.

```
  :
10:   PALLETIZING-EX_1
11: J PAL_1[A_3 ] 100% FINE
12: L PAL_1[A_2 ] 500mm/sec CNT50
13: L PAL_1[A_1 ] 300mm/sec CNT10
14: L PAL_1[BTM ] 100mm/sec FINE
15:   Open hand 1
16: L PAL_1[R_1 ] 300mm/sec CNT10
17: L PAL_1[R_2 ] 500mm/sec CNT50
18:   PALLETIZING-END_1
  :
```

INDEX

<Number>

4D EDITOR FUNCTION 405
4D GRAPHICS DCS 404
4D GRAPHICS Display Scene 396
4D GRAPHICS edit node map.....400
4D GRAPHICS Frame Display.....402
4D GRAPHICS FUNCTION 392
4D GRAPHICS Node Map 399
4D GRAPHICS Position Register.....403
4D GRAPHICS SCENE.....396
4D GRAPHICS select node map.....401
4D GRAPHICS TCP Trace.....402

<A>

Abort KAREL Program.....711
Abort Program by UOP.....193
ABOUT MULTI-GROUP CONFIGURATIONS 161
Absolute（ABS） 305
Accessing the Full Screen 4D Display 406
Accuracy Around Singularity.....32
Add Dual Drive axes 174
Addition of I/O devices for the FANUC robot and the I/O mapping.....691
ADJUSTMENTS 562
ADVANCED CONSTANT PATH 143
Advanced Function 24
After Replacement.....348
ALARM CAUSE/REMEDY DISPLAY FUNCTION .581
ALARM CODE DEFINITIONS 197
Alarm ID 199
Alarm Number.....200
ALARM RECOVERY 163
Alarm Recovery Faults [CHOICE] Screens 164
Alarm Severity 198
Alarms.....529
ALARMS REPORTING SCREEN.....641
Align function 708
Ambient Temperature Setting 591
APPLICATION.....530
“APPLY” FUNCTION KEY 257
Assign the Current Position 615
Assigning Alarm History.....617
Assigning comments 624
Assigning Data ID.....620
Assigning I/O data and simulation status 625
Assigning Integrated PMC address data.....626
Assigning Position Registers.....611
Assigning Program Execution Status 619
Assigning Robot Registers 609
Assigning String Registers 615
Assigning Symbols and Comments for Integrated PMC addresses.....627
Assigning Tool Numbers to Servo Tool Axes.....327
ASSIGNMENT OF HOLDING REGISTERS 607
ATPERCH 189
Attach Instruction is Executed when the Tool is not Attached. 334
Attached Tool has been Detached by Mistake (without Using the Detach Instruction).....336
AUTO BACKWARD EXIT.....544
“Auto Change CIR to L” ITEM 256
AUTOMATIC ERROR RECOVERY FUNCTION .66,88
Automatic Error recovery function common setup 75
AUTOMATIC ERROR RECOVERY FUNCTION OVERVIEW 66
AVAILABLE MOTION MODIFIERS.....660

<B>

BACKGROUND OPERATION OF MATH FUNCTION.....306
BACKWARD EXECUTION.....654
Backward Execution (BWD).....324
BASIC PROCESS AXES (R689).....643
Basic Setup 480
Basic Specification.....310
Battery Ran Low while the Tool was Detached. 336
Battery Voltage has Dropped 336
Battery-Containing Tools.....322
Battery-less Tools.....322
Benefits of Signature Calibrated Robot.....4
Binary.....664
BRAKE CHECK.....220

<C>

Calculate Parameters Screen 21
CALIBNBT.PC Conditions 60
Calibration data.....112
Calibration Motion Failed.....335
CALIBRATION PROCEDURE (for 6-Axis Robots) ..109
Calibration Recovery 24
Calibration System.....5
Calibration Type 6
Cartesian jogging (WORLD, TOOL, USER, and JGFRM).....706
Cartesian motion (Linear, Circle, Arc, and Spline).....704
CARTESIAN SOFTFLOAT GUIDELINES.....50
Cause and Remedy for Alarm Occurrence 289
CAUTION.....224
Cautions 32,100,155,161,269,434
CAUTIONS AND LIMITATIONS.....52
“Change Blend-In Speed” ITEM 256
Change Display Format of Cycle Time Data.....294
CHANGING POSITION DATA DURING S MOTION EXECUTION.....661
CHECK MAINTENANCE TIME AND COMPLETE MAINTENANCE.....591
Check No Part.....522
Checking and Choosing the Mastering Method.....119
Checking the Maintenance Time.....592

Circle Shape Setup 484
CIRCLE WITH FLAT SPOT 476
CIRCLE WITH INWARD BUMP 477
Clear to Proceed 523
CLEAR_RESUME_PROG 69
CMDENBL#1 to #N 187
CMOS Programs 280
CNT 251
COL GUARD ADJUST 97
COLLISION DETECT SETUP SCREEN 96
Collision Detection for estimating motion 113
COMBINED USE OF PERMANENT TORQUE LIMIT AND DYNAMIC TORQUE LIMIT 273
Common Settings 587
Common Settings for All Shapes 483
Communication Setup 667
COMPENSATION VALUE SETTING SCREEN 474
Completing Maintenance 593
Condition 173
Configuring Tooling Valve I/O 507
CONFLICT BETWEEN DATA WRITES 435
CONNECTION AN HMI DEVICE 600
CONTINUOUS TURN FUNCTION 54
Continuous Turn Speed instruction example 58
Controllers 637
CONTROLLING A TOOLING VALVE IN MANUAL MODE 515
Controlling the Palletizing Function by a Palletizing Register 748
Controlling Valve I/O 509
Conversion Operation 226
Coordinate Systems 10
COORDINATED TOUCH SENSING 466
Copy Programs 287
Copying an NC program to the MD drive 227
CORNER REGION AND CORNER DISTANCE 147
Corner Region and Corner Distance Limitations 149
Correspondence of Modbus Address to Robot Data 605
Create Calibration Program Screen 17
Create Signature Calibration Parameters Setup File 23
Creating a Project for the Software PLC 686
Creating and Editing a Shape program 491
CREATING AND EDITING VISUAL DIAGNOSTIC SCREENS 567
CSTOPI#1 to #N 186
Custom 489
CUSTOM MENU 715
Custom Signature Calibration 5
Custom Signature Calibration Preparation 7,12
Custom Signature Calibration Procedure 12
CUSTOMIZE SUPPORT FUNCTION 709
Cycle Mode 292
Cycle Power 714
CYCLE TIME LOGGING 291

<D>

DATA FORMATS 664
DATA MONITOR CHART 217
DATA MONITOR FUNCTION 201
DATA MONITOR OPERATION 218
DATA MONITOR SCHEDULE 213
DATA MONITOR SETUP 203
DATA OFFSET FUNCTION 534
DATA TRANSFER BETWEEN ROBOTS FUNCTION 411
Data type of Holding Registers 608
DEFAULT 251
Delete Set 717
Design of jig for mounting reflector 37
Detail Screen of KAREL Config 713
Detailed Data Comparison 355
Details 351
Different Tool from that Specified by the Attach Instruction is Attached. 335
DIGITAL I/O JOG FUNCTION 700
Disable for 20 Cycles 529
Display / Modify / Clear Dual Drive settings 176
DISPLAY / MODIFY PERMANENT TORQUE LIMIT 271
DISPLAY MODES OF CYCLE TIME 292
Display Reference Cycle Time Line 296
Display Signature Screen 14
Display Target Cycle Time Line 295
Displaying detail conditions of the Error recovery status DO 86
Displaying Maintenance Record 595
DUAL MOTOR DRIVE 172
DYNAMIC ASSIGNMENT TO HOLDING REGISTERS 629
DYNAMIC TORQUE LIMIT 261

<E>

EFFECT OF A SINGLE TAUGHT POINT ON THE PATH 655
Enable / disable setting 707
Enable/Disable iRCalibration Signature Compensation .29
Enable/Disable setting 702,705
“End Line” ITEM 255
ENHANCED MIRROR IMAGE 236
ERROR CODE OUTPUT FUNCTION 196
Error messages 62
Estimation Motion Setup 104
Estimation Result 109
Example mhgripcfg.xml file 513
Example of Setting the Software PLC External I/O Assignment and Mapping 680
Example of Software PLC Internal I/O Assignment Configuration 683
Example Program 323,326,528
Examples 516
EXCEPTIONS AND RESTRICTION 309
EXECUTING THE PALLETIZING FUNCTION 745
EXECUTION EXAMPLE OF SPOT PROGRAM 297
Execution Notes 57,62
EXECUTION OF THE RESUME PROGRAM FROM THE TEACH PENDANT AND TEST MODE 87

EXECUTION OF TOUCH SENSING PROGRAM 460
EXPANDED REGISTERS FUNCTION 554
Exponent 304
EXTENDED ALARM LOG 137

<F>

F code – Feed Rate / Speed 229
FAST EXIT/ENTRY FEATURE 67,88
FAULT & INCIDENT REPORT 641
FAULT#1 to #N 188
Feature of Function 310
Feed Rate Conversion 534
FILE Programs 281
FINE 251
FINISHING FUNCTION PACKAGE 543
Forward Execution (FWD) 324
Free 707
FULL SCREEN 4D DISPLAY 406
FUNCTION SPECIFICATION OF MATH FUNCTIONS 301
Functions that cannot be used together 58

<G>

G code – Motion modifiers 228
GC Mastering Guidance 121
"Geometric Data" ITEM 255
"Geometric Type" ITEM 255
Graphic Models 393
GRAVITY COMPENSATION 118
Grip Part 517

<H>

Hardware 253
HARDWARE AND SOFTWARE 253
HELD#1 to #N 188
HELP AND DIAGNOSTICS DISPLAY 580
HELP/DIAGNOSTICS MENU 584
HELP/DIAGNOSTICS SCREEN 583
Hexagon Shape Setup 486
HIGH SPEED POSITION OUTPUT 664
HIGH SPEED SHIFT JOG OPERATION 342
HIGH SPEED SHIFT KEY FUNCTION 342
HIGH SPEED SHIFT TEST EXECUTION OPERATION 343
HIGH-SENSITIVITY COLLISION DETECTION 95
Hints 628
HMI DEVICE COMMUNICATION 599
HOLD#1 to #N 185
Hourly Mode 293
How Linear Distance Works 144
How to Display Alarm Log 138
HOW TO USE 251
HOW TO USE MROT 471
HOW TO USE SINGULARITY AVOIDANCE 153
How to Use Singularity Avoidance in Jogging 153
How to use the CODESYS Development System 686

<I>

I/O assignment for The Software PLC 675
i-cable Change 598
i-grease Change 597
Importing and Exporting Material Handling Gripper xml 511
Incident screen 641
INITIAL SETTING BEFORE USE 220
INITIAL SETUP 314
Input and Output Signals 196
Inputting Initial Data 725
Instruction Format of Assignment Statements 300
INSTRUCTION FORMAT OF MATH FUNCTION 299
Instruction Format of Relational Statements 300
Instruction Format of Wait Command Statements 301
Instructions 558
Insufficient Number of Spline Motion Instructions 655
Intelligent Cable Change Reminder (option: S511) 586
INTELLIGENT CHANGE REMINDER 586
Intelligent Grease Change Reminder 586
Inverse Trigonometric Function（ACOS） 303
Inverse Trigonometric Function（ASIN） 302
Inverse Trigonometric Function（ATAN） 304
Inverse Trigonometric Function（ATAN2） 303
*i*RCalibration Signature 3
iRCalibration Signature Status Screen 29
*i*RDIAGNOSTICS 345

<J>

JOG OPERATION WITHOUT SHIFT 338
Jog preview 397
Jogging Operation 705
JOINT / CARTESIAN SOFTFLOAT 40
Joint Angles with Binary 665
Joint Angles with XML 666
Joint jogging 706
Joint motion 704

<K>

K code - Config string 230
KAREL BUILT-INS 418
KAREL CONFIG 709
KAREL Config Screen 709
Keyhole Shape Setup 487

<L>

"LEARN" FUNCTION KEY 257
Learning 259
LEARNING PATH CONTROL 253
LEMON SHAPE CIRCLE 476
LEMON SHAPE CIRCLE WITH RIPPLES 476
Limitation and Caution of KAREL Config 714
LIMITATIONS115,155,161,224,252,259,267,275,434,471,564,701
Limitations of Corner Size by Distance between Taught Points 148
LINEAR DISTANCE 143
Linear Distance Limitations 146

Line-by-Line Mode 293
LIST MENU.....368
LOAD ESTIMATION.....101
LOAD ESTIMATION PROCEDURE (for 6-Axis Robots) 101
Load NC program in FILE menu 226
LOADING A NOT EXPANDED xxxREG.VR FILE..555
LOADING AN EXPANDED xxxREG.VR FILE 556
LOADING PROCESS IN PROGRAM EXECUTION 287
LOGGING CYCLE TIME 294
LOGIC INSTRUCTIONS BETWEEN SPLINE MOTION INSTRUCTIONS.....658
Login/Logout to the robot control device, download and start/stop the PLC program.....694
Longest Time (L_TIME) screen.....642
LPC BACKUP 260
LPC LEARNING AND PLAYBACK.....259
LPC SETUP MENU 254

<M>

M code – Miscellaneous codes.....230
MACRO PROGRAM : SV_ADJST.TP 474
MAIN MENU 586
“Main Program” ITEM 254
MAINT_PROG [1:Comment].....72
MAINT_PROG INSTRUCTION.....71
Maintenance Item Settings 589
MAINTENANCE PROGRAM type recovery (Setting items related to the Fast exit/entry function) 83
Maintenance Record File 596
MAINTENANCE RECORDS 595
MAINTENANCE REMINDER 586
Maintenance Reminder Notification 592
MAINTENANCE REMINDER SETUP.....587,594
Make Backup of Programs 287
MANAGING SCREENS 574
MANUAL FUNCTIONS SCREEN FOR THE RESUME PROGRAM FUNCTION 85
Manual Guided Teach 707
Mastering 31,118
Mastering for Signature Calibrated Robot 31
Mastering of Dual Drive axes.....177
Mastering Procedure 120
MATERIAL HANDLING GRIPPER OPTION.....494
MATERIAL HANDLING VALVE MACRO PROGRAMS 516
MATH FUNCTION INSTRUCTION.....299
MAX SPEED 151
Max Speed Limitations 152
“Max. Iterations” ITEM 256
Measuring the robot installation position.....33
MENU UTILITY 357
MH Tryout Mode 529
MIRROR IMAGE OF EXTENDED AXES.....241
MIRROR IMAGE USING EXISTING FRAMES AND MIRROR PLANES, WITH CONTROLLED ORIENTATION 240
MODBUS DATA MODEL.....604
Modbus Function Codes 606
Modbus RS-232-C Connection.....600
Modbus/TCP Slave Ethernet Connection 602
Modification and Addition of Signal 184
MODIFYING THE PALLETIZING FUNCTION.....749
Monitoring the Change of Mastering Data.....32
Monitoring the PLC Program 697
MOTION INSTRUCTION ENHANCED EDITING...532
MOTION INSTRUCTION INSERT AND INSTRUCTION DELETE FUNCTION.....532
MOTION PROFILER.....348
Motion range.....112
MotionDevices.....633
MotionDeviceSystem.....632
MROT INSTRUCTION.....471
MULTI UOP INTERFACE FUNCTION 181
MULTIPLE SEARCHES.....465

<N>

N code – Line numbers 228
Natural Logarithm.....304
NAVIGATING WITH THE PDF VIEWER.....579
NC PROGRAM CONVERSION.....225
NC PROGRAM CONVERSION EXAMPLE 233
NC program syntax 225,227
Node Map supported data 401
Normal Mastering and GC Mastering 118
NORMAL MOTION.....651
NOTES / LIMITATIONS 470
NOTES AND LIMITATIONS.....58
Notes on Teaching the Palletizing Function 744

<O>

O code – Program Name/Number 227
ONLINE HELP FUNCTION.....580
OPC UA DATA MODEL 630
OPC UA Organization of the Modbus data model.....604
OPC UA Server Ethernet Connection.....603
OPENING A PDF DOCUMENT.....578
OPERATING PROCEDURE 101
OPERATION 226,393,406
OPERATION GROUP DO OUTPUT FUNCTION 64
OPERATION LOG BOOK.....123
OPERATION WITHOUT SHIFT FUNCTION.....338
OPERATIONS.....133
OPERATOR ENTRY MENU.....383
“Orientation Control” ITEM.....256
Other 193
Other (user-defined) M-codes 233
Other notes.....59
OTHER RELATED MATTERS 112
Other Specifications.....561
OTHER SPECIFICATIONS AND RESTRICTIONS....88
OUTLINE 310
Outline of Servo Tool Change Installation.....313
OVERVIEW1,3,54,225,253,393,405,406,478,700,702,715
Overview of Software PLC 669

<P>

PALLETIZING ALL-POINT TEACHING.....751
PALLETIZING FUNCTION 719
PALLETIZING FUNCTION WITH EXTENDED AXES 751
PALLETIZING INSTRUCTIONS.....721
PALLETIZING MODE.....702
Palletizing Register 746
PARALLEL MIRROR IMAGE 237
PARAMETER SETTINGS 473
Part Present 521
Path has a High Curvature.....656
PATH SWITCHING FUNCTION 156
“Path Tolerance” ITEM 256
Pause Program by UOP.....193
Pausing and Resuming from Another Spline Motion Instruction 657
PAYLOAD OVER AND PAYLOAD SETTING CONFIRM FUNCTION.....114
Payload setting for estimating motion 112
PDF VIEWER FUNCTION.....578
Performing Estimation and Applying the Result.....107
PERIPHERAL I/O 182
PERMANENT TORQUE LIMIT FOR ADDITIONAL MOTOR 270
PLC programming and Task assignment.....687
PNSTROBE#1 to #N,PROD_START#1 to #N 187
Points to keep in mind when teaching.....325
Pommel Shape Setup.....488
Position data.....229
Position Data Conversion (2 point teach method).....540
Position Data Conversion (direct method) 537
Position Data Packet Settings.....667
Position Program.....8
PRECAUTION 289
Precautions.....533,542
Predefined M-codes 231
PREFACE..... p-1
PRELIMINARY TOOL ATTACHING OPERATION 314
Preparation 102,674
Prepare to Pick 522
PROCEDURE.....115,253
Procedure to Show Payload Over and Payload Setting Confirm Screen.....115
Procedure to Use Payload Over Confirm (When The Program or Payload is Changed) 116
Procedure to Use Payload Setting Confirm (When Payload is Changed).....115
PROCESS AXES.....643
PROCESS AXES STATUS 648
“Process Disabled” ITEM 257
PROCESS SPEED.....150
Process Speed Limitations 151
“Process Speed” ITEM.....256
PRODUCTION.....529
Production with LPC Playback 259
Program Edit Screen Display When Operating Motion Group is Changed.....191
PROGRAM EXCHANGE FUNCTION WITHOUT ENOUGH MEMORY SPACE.....288
Program Execution.....347
PROGRAM INSTRUCTION.....265
PROGRAM INSTRUCTIONS 97
Program Number Select (PNS) 192
Program to Get a Numeric Register 414
Program to Get a Position Register 415
Program to Set a Numeric Register 415
Program to Set a Position Register.....416
PROGRAM TOOLBOX 140
PROGRAMMING 216,475,548
PROGRUN#1 to #N,PAUSED#1 to #N 188
PROMPT BOX MESSAGE MENU 358
PROMPT BOX YES/NO MENU 363
PS INSTRUCTION.....156
PUSHOUT SOFTFLOAT.....45

<Q>

Quick Mastering.....31
Quick Mastering Reference Position Setup.....333

<R>

RECORDED EVENTS 125
RECOVERY FROM ERROR.....417
Rectangle Shape Setup.....487
Reference 33
Reference Frame Function 28
Reference Position Setup for Calibration Types 3 and 4329
Reference Position Setup for Calibration Types 5 and 6331
RELATED VIEW 257
Relationship with I/O Assignment (Robot).....678
Release Part.....519
Remarks 708
REMOTE TCP FUNCTION.....90
Requirements 406
Reset to a specified range.....61
RESETTING THE ROTATION OF POSITIONER AND INDEPENDENT AXES USING A KAREL PROGRAM 60
RESTRICTION OF TEACHING MATH FUNCTION308
Restrictions 33,663
Restrictions for TP program execution 704
Restrictions, prohibitions, and precautions 310
“RESULT” FUNCTION KEY 257
Results.....349
RESULTS OF BRAKE CHECK 223
RESUME PROGRAM FUNCTION.....66,88
RESUME PROGRAM type recovery (Setup items related to the Resume program function) 81
RESUME_PROG [1: Comment] 68
RESUME_PROG INSTRUCTION 67
Resuming After a Pause During Backward Execution .658
Resuming after Jogging.....657
RESUMING AFTER MODIFYING THE PROGRAM662
RESUMING AFTER PAUSING 657
RETURN_PATH_DSBL 70
RETURN_PATH_DSBL INSTRUCTION 70

RGET_PORTCMT Built-in ROUTINE.....419
RGET_PORTSIM Built-in ROUTINE 420
RGET_PORTVAL Built-in ROUTINE 420
RGET_PREGCMT Built-in ROUTINE.....421
RGET_REG Built-in ROUTINE.....422
RGET_REG_CMT Built-in ROUTINE 422
RGET_SREGCMT Built-in ROUTINE.....423
RGET_STR_REG Built-in ROUTINE.....424
RNUMREG_RECV Built-in ROUTINE.....424
RNUMREG_SEND Built-in ROUTINE.....425
ROBOT CONDITION DETECTION 345
Robot Condition Detection Status.....347
Robot Service Request (RSR).....192
ROBOT SETTINGS.....597
Robot Stopped during Calibration. 334
RobotInformation.....631
Rotation.....707
ROTATIONAL MIRROR IMAGE.....239
Round Off（ROUND） 305
RPOSREG_RECV Built-in ROUTINE 426
RPOSREG_SEND Built-in ROUTINE.....427
RSET_INT_REG Built-in ROUTINE.....428
RSET_PORTCMT Built-in ROUTINE.....428
RSET_PORTSIM Built-in ROUTINE.....429
RSET_PORTVAL Built-in ROUTINE.....430
RSET_PREGCMT Built-in ROUTINE.....430
RSET_REALREG Built-in ROUTINE 431
RSET_REG_CMT Built-in ROUTINE.....432
RSET_SREGCMT Built-in ROUTINE.....432
RSET_STR_REG Built-in ROUTINE 433
RSR.....187
Run Calibration Program Screen.....18
Run KAREL Program 710
RUN PROGRAM.....192

<S>

SAFETY PRECAUTIONS s-1
SafetyStates.....640
Save / Load LS Files 287
SAVE / LOAD PROGRAMS 286
Save / Load TP Files 286
Save Cycle Time Data.....295
SAVING AND LOADING FILES.....555
Saving and Loading Software PLC related data.....699
Scene visibility.....409
SEARCH PATTERN 445
Select Calibration Type.....15
SELECT PROGRAM.....189
Select Program at Running or Paused 191
Selecting a Palletizing Instruction.....723
Selecting the Handling Config (SpotTool+ only).....495
SERVO TOOL CHANGE FUNCTION.....310
SERVO TOOL CHANGE SCREENS 326
Set CurrentValve.....527
Set Custom Menu.....715
Set MH Tool.....528
Set visibility: 4D GRAPHICS display.....398
Set Visibility: node map position numbers 401
Set Visibility: position register numbers.....404
Setting Motion Parameters for Servo Tool Axes.....327
Setting of Motion Groups Intended by Each UOP Set.190
Setting of the Number of UOP Set.....183
Setting Path Pattern Conditions 738
SETTING THE NUMBER OF REGISTERS 554
SETTING THE REFERENCE POSITION.....322
SETTING UP A MATERIAL HANDLING TOOL.....494
SETTING UP BOOK.....132
SETTING UP PERMANENT TORQUE LIMIT.....270
SETTING UP TCP SPEED OUTPUT.....276
SETTING UP THE AUTOMATIC ERROR RECOVERY FUNCTION 74
SETTING UP THE DYNAMIC TORQUE LIMIT.....262
Setting Up Tooling Valve Signals.....505
Setting Up Tooling Valves.....497
SETTINGS.....95,473
SETUP55,92,118,137,345,349,406,412,530,545,644,666,700
Setup Calibration settings 16
SETUP FOR HIGH SPEED SHIFT KEY FUNCTION342
SETUP OF DUAL DRIVE PAIR FOR 1-AXIS POSITIONER 178
SETUP OF DUAL DRIVE PAIR FOR EXTENDED AXIS (J518), INDEPENDENT AXIS (H895) AND POSITIONER (H874, H896) 173
SHADOW ONDEMAND Programs.....281
SHADOW Programs.....280
SHAPE GENERATION FUNCTION.....478
Shape Generation Programs.....479
SHAPE GENERATION SETUP 479
SHAPE GENERATION UTILITY 491
SINGLE STEP EXECUTION 654
SINGULARITY AVOIDANCE FUNCTION.....153
Slot Shape Setup 485
SMALL CIRCLE ACCURACY IMPROVEMENT....473
SOFT LIMIT SETTING 140
SOFTFLOAT FUNCTION 40
SOFTFLOAT GUIDELINES.....48
Software 253
SOFTWARE PLC.....669
Software PLC External I/O Assignment 676
Software PLC Input/Output area.....675
Software PLC internal I/O Assignment.....682
SPECIAL JOG SEQUENCE 530
Special M-codes.....231
SPECIFICATION 700
Specification Overview.....545
SPECIFICATIONS.....95,196,225,473,558
SPLINE MOTION INSTRUCTION.....651
Square Root（SQRT） 301
STANDARD DATA TRANSFER PROGRAMS 414
START BRAKE CHECK 222
“Start Line” ITEM 254
Start Mode Config of KAREL program.....712
Start Output.....668
START#1 to #N.....186
Starting a Program from a Spline Motion Instruction..658
Starting Continuous Turn.....57

Starting Custom Menu 715
STATUS MENU 374
STITCH FUNCTION 558
STITCH FUNCTION SCREEN STRUCTURE 565
Stitch Process Flow 560
Stitch Schedules 558
STOP PROGRAM 193
Stopping Continuous Turn 57
STORAGE CONFIGURATION 281
STORAGES 280
SYSRDY#1 to #N 187
SYSTEM CONFIG MENU 154,162,563
System Configuration 312
SYSYTEM CONFIG SCREEN ITEMS 193

<T>

TCP Cartesian Position with Binary 665
TCP Cartesian Position with XML 666
TCP SPEED OUTPUT 275
TCP SPEED OUTPUT INSTRUCTION 278
TCP/IP SETUP FOR ROBOGUIDE 413
TCP/Uframe Calibration 26
TEACH MATH FUNCTION INSTRUCTION 306
TEACHING 325
Teaching a Path Pattern 741
Teaching a Stacking Pattern 732
TEACHING THE PALLETIZING FUNCTION 723
Teaching the Same Position 656
TERMINATION TYPE OF SPLINE MOTION 654
TERMINOLOGY 411
TEST EXECUTION OPERATION WITHOUT SHIFT339
TIMEOUT AND RETRY 434
TOOL ATTACH Instruction 323
Tool Axis of a Detached Tool has Moved. 336
TOOL CHANGE INITIAL SETUP 327
TOOL CHANGE INSTRUCTION 323
TOOL CHANGE REFERENCE POSITION SETUP METHOD (BATTERY-LESS SERVO TOOL TYPE) 329
TOOL CHANGE SETUP 316
TOOL CHANGE STATUS 324
TOOL DETACH Instruction 323
TORQUE CONTROL 261
Torque Monitor Function 649
Total Time (T_TIME) screen 641
TOUCH FRAME SETUP 442
TOUCH SCHEDULES 449
TOUCH SENSING 438
TOUCH SENSING I/O ASSIGNMENT 440
TOUCH SENSING PROGRAMMING 453
TOUCH SKIP FUNCTION 469
TOUCH SKIP PROGRAM 469
TOUCH SKIP SCREEN 469
TOUCHING UP OF TOUCH SENSING PROGRAM 462
TP DRAM/FILE STORAGE FUNCTION 280
TP Instructions 646
TP program execution 702
Translation 707
TRIGGER POINT OF PS INSTRUCTION 158
Trigonometric Function（COS） 302
Trigonometric Function（SIN） 301
Trigonometric Function（TAN） 302
TROUBLE SHOOTING (TUNING GUIDELINES) ...475
TROUBLESHOOTING 334,435
Truncate（TRUNC） 305
Turn OFF Blowoff 526
Turn OFF Vacuum 525
Turn ON Vacuum 524
TYPE OF MATH FUNCTIONS 299
Types of Alarms 196

<U>

Update Cycle Time Data 295
USAGE 530,532
Use KAREL Config Screen 709
Useing Continuous Turn with Constant Path 59
User Alarm Recovery 166
Using CALIBNBT.PC 61
USING CONTINUOUS TURN 57
Using Linear Distance 146
Using Singularity Avoidance in a TPE program 154
USING YOUR TEACH PENDANT MENUS 390
UTILITIES Shape Settings 491

<V>

Variable Data Packet Settings 668
Variables with Binary 665
Variables with XML 666
Verification of the PLC program 697
VIA INSTRUCTION 251
View adjustment mode 408
View/Apply Calibration Result 22
VIEWING VISUAL DIAGNOSTIC SCREENS 575
VISUAL DIAGNOSTICS 566
Visual jog 396

<W>

WARNINGS (Be sure to read this section for safety.) ...89
WHEN SPLINE MOTION IS NOT POSSIBLE 655

<X>

XML 666

<Z>

ZDT 594

REVISION RECORD

Edition	Date	Contents
03	Apr., 2025	• Following new function has been added. Software PLC, DIGITAL I/O JOG FUNCTION, Palletizing Mode, CUSTOMIZE SUPPORT FUNCTION, PALLETIZING FUNCTION • New features are added in SOFTFLOAT FUNCTION, CONTINUOUS ROTATION FUNCTION, AUTOMATIC ERROR RECOVERY FUNCTION, REMOTE TCP FUNCTION, GRAVITY COMPENSATION, MULTI UOP INTERFACE FUNCTION, TCP SPEED OUTPUT, HMI DEVICE COMMUNICATION, PROCESS AXES. • Explanations are changed in *i*RCALIBRATION SIGNATURE, HIGH SENSITIVITY COLLISION DETECTION, DATA MONITOR FUNCTION, NC PROGRAM CONVERSION, LEARNING PATH CONTROL, TORQUE CONTROL.
02	Feb., 2024	
01		

B-84464EN-2/03

