FANUC Robot series

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

Optional Function

OPERATOR'S MANUAL

B-83284EN-2/10

© FANUC CORPORATION, 2012

- **Original Instructions**

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.
- All specifications and designs are subject to change without notice.

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

TABLE OF CONTENTS

SAFETY PRECAUTIONS s-1

1 PREFACE 1
1.1 ABOUT MANUAL 1

2 OVERVIEW 3

3 SOFTFLOAT FUNCTION 5
3.1 JOINT / CARTESIAN SOFTFLOAT 5
3.2 PUSHOUT SOFTFLOAT 9
3.3 KNOWHOW OF SOFTFLOAT 12
3.4 KNOWHOW OF CARTESIAN SOFTFLOAT 13
3.5 CAUTIONS / RESTRICTIONS 14

4 CONTINUOUS ROTATION FUNCTION 16
4.1 OVERVIEW 16
4.2 SETUP 17
4.3 USING THE FUNCTION 19
4.3.1 Starting continuous rotation 19
4.3.2 Stopping continuous rotation 19
4.3.3 Notes 19
4.3.4 Example of use 20
4.4 NOTES / LIMITATIONS 20
4.4.1 Functions that cannot be used together 20
4.4.2 Use of Continuous Rotation function with Constant Path function 21
4.4.3 Other notes 21
4.5 RESET ROTATION OF POSITIONER AND INDEPENDENT AXIS BY KAREL PROGRAM 22
4.5.1 Condition to use 22
4.5.2 Usage 22
4.5.3 Note 23
4.5.4 Error message 23

5 OPERATION GROUP DO OUTPUT FUNCTION 25

6 AUTOMATIC ERROR RECOVERY FUNCTION 27
6.1 AUTOMATIC ERROR RECOVERY FUNCTION 27
6.2 RESUME PROGRAM FUNCTION 27
6.3 FAST EXIT/ENTRY FEATURE 28
6.4 RESUME_PROG INSTRUCTION 28
6.5 RETURN_PATH_DSBL INSTRUCTION 31
6.6 MAINT_PROG INSTRUCTION 32
6.7 SETTING OF THE AUTOMATIC ERROR RECOVERY FUNCTION 35
6.8 MANUAL OPERATION SCREEN OF THE RESUME PROGRAM FUNCTION 46

6.9 EXECUTION OF THE RESUME PROGRAM FROM THE TEACH PENDANT AND TEST MODE 48
6.10 CHANGING CONDITIONS FOR EXECUTING THE RESUME PROGRAM 48
6.11 OTHER SPECIFICATIONS AND RESTRICTIONS 49
6.12 WARNINGS (Be sure to read this section for safety.) 50

7 REMOTE TCP FUNCTION 51
7.1 SETUP 53

8 HIGH SENSITIVITY COLLISION DETECTION 56
8.1 SPECIFICATION 56
8.2 SETTINGS 56
8.3 COLLISION GUARD SETUP SCREEN 57
8.4 PROGRAM INSTRUCTIONS 58
8.4.1 COL DETECT ON / COL DETECT OFF 58
8.4.2 COL GUARD ADJUST 59
8.5 CAUTIONS 61

9 LOAD ESTIMATION 62
9.1 OPERATING PROCEDURE 62
9.2 LOAD ESTIMATION PROCEDURE (for 6-Axis Robots) 62
9.2.1 Preparation 62
9.2.2 Estimation Motion Setup 65
9.2.3 Estimation Execution and Applying the Result 69
9.2.4 Estimation Result 70
9.3 CALIBRATION PROCEDURE (for 6-Axis Robots) 70
9.4 OTHER RELATED MATTERS 73

10 PAYLOAD OVER AND PAYLOAD SETTING CONFIRM FUNCTION . 75
10.1 LIMITATIONS 76
10.2 PROCEDURE 76
10.2.1 Procedure to Show Payload Over and Payload Setting Confirm Screen 76
10.2.2 Procedure to Use Payload Setting Confirm (When Payload is Changed) 76
10.2.3 Procedure to Use Payload Over Confirm (When The Program or Payload is Changed) 77

11 GRAVITY COMPENSATION 79
11.1 SYSTEM VARIABLES 79
11.2 MOTION SCREEN 79
11.3 MASTERING 80
11.3.1 “Normal Mastering” and “GC Mastering” 80
11.3.2 How to Choose Mastering Method 81
11.3.3 Mastering Procedure 81
11.3.4 Guidance of GC Mastering 82

12 OPERATION LOG BOOK 83
12.1 RECORDED EVENTS 85
12.2 SETTING UP BOOK 89
12.3 OPERATIONS 89
12.4 EXTENDED ALARM LOG 93
12.4.1 Setup 93

12.4.2 How to Display Alarm Log 94

13 PROGRAM TOOLBOX 96
13.1 SOFT LIMIT SETTING 96

14 ADVANCED CONSTANT PATH 99
14.1 LINEAR DISTANCE 99
14.1.1 How to Use 100
14.1.2 Limitations 101
14.1.3 Procedure to Use 102
14.2 CORNER REGION 102
14.2.1 Limitation of the Specifiable CR Value 103
14.2.2 Limitations 104
14.3 PROCESS SPEED 105
14.3.1 Limitations 107
14.4 MAX SPEED 107
14.4.1 Limitations 108

15 SINGULARITY AVOIDANCE FUNCTION 109
15.1 HOW TO USE SINGULARITY AVOIDANCE 109
15.1.1 How to Use Singularity Avoidance in Jogging 109
15.1.2 How to Use Singularity Avoidance in TPE Program 109
15.2 LIMITATIONS 110
15.3 CAUTIONS 110

16 PATH SWITCHING FUNCTION 112

17 AUXILIARY AXIS SERVO OFF (LOCAL STOP) FUNCTION 118
17.1 SPECIFICATION 118
17.2 CONSTRAINTS 120
17.3 SETTINGS 121
17.4 ATTENTION 124
17.5 PROGRAMMING 125

18 DUAL MOTOR DRIVE 129
18.1 SETUP 130
18.2 SETUP OF DUAL DRIVE PAIR FOR 1-AXIS POSITIONER 132

19 MULTI UOP INTERFACE FUNCTION 136
19.1 PERIPHERAL I/O 137
19.1.1 Setting of the Number of UOP Set 138
19.1.2 Modification and Addition of Signal 139
19.1.3 HOLD#1 to #N 140
19.1.4 CSTOPI#1 to #N 141
19.1.5 START#1 to #N 141
19.1.6 RSR 142
19.1.7 PNSTROBE#1 to #N,PROD_START#1 to #N 142
19.1.8 CMDENBL#1 to #N 142
19.1.9 SYSRDY#1 to #N 142
19.1.10 PROGRUN#1 to #N,PAUSED#1 to #N 143
19.1.11 HELD#1 to #N 143
19.1.12 FAULT#1 to #N 143
19.1.13 ATPERCH 144

19.2 SELECT PROGRAM 144
19.2.1 Setting of Motion Groups Intended by Each UOP Set 145
19.2.2 Select Program at Running or Paused 146
19.2.3 Program Edit Screen Display When Operating Motion Group is Changed 146
19.3 RUN PROGRAM 147
19.3.1 Robot Service Request (RSR) 147
19.3.2 Program Number Select (PNS) 148
19.4 STOP PROGRAM 148
19.4.1 Pause Program by UOP 148
19.4.2 Abort Program by UOP 148
19.4.3 Other 148
19.5 SYSYTEM VARIABLES 148

20 ERROR CODE OUTPUT FUNCTION 151
20.1 SPECIFICATION 151
20.1.1 Types of Alarms 151
20.1.2 Input and Output Signals 151
20.2 MEANING OF ALARM CODE 152
20.2.1 Severity of Alarm 153
20.2.2 Alarm ID 154
20.2.3 Alarm Number 155

21 DATA MONITOR FUNCTION 156
21.1 DATA MONITOR SETUP 158
21.2 DATA MONITOR SCHEDULE 167
21.3 PROGRAMMING 171
21.4 DATA MONITOR CHART 171
21.5 DATA MONITOR OPERATION 173

22 BRAKE CHECK FUNCTION 174
22.1 INITIAL SETTING BEFORE USE 174
22.2 START BRAKE CHECK 175
22.3 RESULTS OF BRAKE CHECK 176
22.4 LIMITATIONS 176
22.5 CAUTION 176

23 PANEL WIZARD 177
23.1 SETTING UP 177
23.1.1 Overview 177
23.1.2 Available *i*Pendant Controls 180
23.1.3 Setting up Fast Label 181
23.1.4 Setting up Fast Lamp 182
23.1.5 Setting up Fast Switch 184
23.1.6 Addition of Button Change Control 185
23.1.7 Addition of Command Button Control 188
23.1.8 Addition of Edit Box Control 189
23.1.9 Addition of Label Control 190
23.1.10 Addition of Toggle Button Control 190
23.1.11 Addition of Toggle Lamp Control 192
23.1.12 Modification of Panel 194
23.1.13 Modification of Control 195
23.1.14 Delete of Control 196

23.1.15 Cut/Copy Paste of Control 197
23.1.16 Modification of Page 198
23.1.17 Re-creation of Panel 199
23.2 RUN KAREL PROGRAM BY PANEL 200
23.2.1 Caution for Creation of KAREL Program 200
23.2.2 Creation of Run Button 200
23.3 USAGE 201
23.3.1 Display of Panel 201
23.3.2 Backup/Restore 202

24 ENHANCED MIRROR IMAGE 203
24.1 PARALLEL MIRROR IMAGE 204
24.2 ROTATIONAL MIRROR IMAGE 206
24.3 MIRROR IMAGE USING EXISTING FRAMES AND MIRROR PLANES, WITH CONTROLLED ORIENTATION 207
24.4 MIRROR IMAGE OF EXTENDED AXES 208

25 CUSTOMIZE SUPPORT FUNCTION 218
25.1 KAREL CONFIG 218
25.1.1 KAREL Config Screen 218
25.1.2 Use KAREL Config Screen 218
25.1.3 Run KAREL Program 219
25.1.4 Abort KAREL Program 220
25.1.5 Start Mode Config of KAREL program 221
25.1.6 Detail Screen of KAREL Config 222
25.1.7 Limitation and Caution of KAREL Config 223
25.1.8 Cycle Power (R-30*i*B Controller) 224
25.2 CUSTOM MENU 224
25.2.1 Overview 224
25.2.2 Starting Custom Menu 224
25.2.3 Set Custom Menu 225
25.2.4 Delete Set 227

26 KAREL PROGRAM EXECUTION HISTORY RECORD 228
26.1 HARDWARE AND SOFTWARE 228
26.1.1 Hardware and Software Requirements 228
26.1.2 Hardware 228
26.1.3 Software 228
26.1.4 Performance 229
26.2 SETUP AND OPERATIONS 229
26.2.1 Setting Up the KAREL Program Execution History Record 229
26.2.2 Dump Selections Screen 229
26.2.3 Task Selection Screen 231
26.2.4 Stop Logging Tasks Screen 232
26.2.5 List Selected Tasks Screen 234
26.2.6 Event Class Selection Screen 235
26.2.7 Event Detail Selection Screen 237
26.2.8 Enable or Disable All Event Logging 237
26.3 LOGGING EVENTS 238
26.3.1 Setting up Events 238
26.3.2 Logging Events to an ASCII File 239
26.3.3 ASCII File General Event Information 239
26.3.4 ASCII File Specific Event Information 240

26.4 APPENDIX....244
26.4.1 Overview....244
26.4.2 KAREL Program Example....244
26.4.3 Teach Pendant Program Example....245
26.4.4 ASCII File Example....245

27 TORQUE LIMIT FUNCTION....247
27.1 TORQUE LIMIT FUNCTION FEATURE....248
27.2 TORQUE LIMIT MULTI-AXIS SETUP FUNCTION....250
27.3 LIMITATIONS....251
27.4 CAUTION....251

28 TCP SPEED OUTPUT....252
28.1 LIMITATIONS....252
28.2 SETTING UP TCP SPEED OUTPUT....253
28.3 TCP SPEED OUTPUT INSTRUCTION....255

29 TP DRAM/FILE STORAGE FUNCTION....257
29.1 STORAGES....257
29.1.1 CMOS Programs....257
29.1.2 SHADOW Programs....257
29.1.3 SHADOW ONDEMAND Programs....258
29.1.4 FILE Programs....258
29.2 STORAGE CONFIGURATION....258
29.3 SAVE / LOAD PROGRAMS....263
29.3.1 Save / Load TP Files....263
29.3.2 Copy Programs....264
29.3.3 Save / Load LS Files....264
29.3.4 Make Backup of Programs....264
29.4 LOADING PROCESS IN PROGRAM EXECUTION....264
29.5 PROGRAM EXCHANGE FUNCTION WITHOUT ENOUGH MEMORY SPACE....265
29.6 PRECAUTION....266
29.6.1 Cause and Remedy for Alarm Occurrence....266

30 CYCLE TIME LOGGING....267
30.1 DISPLAY MODES OF CYCLE TIME....268
30.1.1 Cycle Mode....268
30.1.2 Hourly Mode....269
30.1.3 Line-by-Line Mode....269
30.2 LOGGING CYCLE TIME....270
30.2.1 Change Display Format of Cycle Time Data....270
30.2.2 Save Cycle Time Data....271
30.2.3 Update Cycle Time Data....271
30.2.4 Display Target Cycle Time Line....271
30.2.5 Display Reference Cycle Time Line....272
30.3 EXECUTION EXAMPLE OF SPOT PROGRAM....273

31 MATH FUNCTION INSTRUCTION....275
31.1 TYPE OF MATH FUNCTIONS....275
31.2 INSTRUCTION FORMAT OF MATH FUNCTION....275
31.2.1 Instruction Format of Assignment Statements....276

31.2.2 Instruction Format of Relational Statements 276
31.2.3 Instruction Format of Wait Command Statements 277
31.3 FUNCTION SPECIFICATION OF MATH FUNCTIONS 277
31.3.1 Square Root（SQRT） 277
31.3.2 Trigonometric Function（SIN） 277
31.3.3 Trigonometric Function（COS） 278
31.3.4 Trigonometric Function（TAN） 278
31.3.5 Inverse Trigonometric Function（ASIN） 278
31.3.6 Inverse Trigonometric Function（ACOS） 279
31.3.7 Inverse Trigonometric Function（ATAN2） 279
31.3.8 Inverse Trigonometric Function（ATAN） 280
31.3.9 Exponent 280
31.3.10 Natural Logarithm 280
31.3.11 Absolute（ABS） 281
31.3.12 Truncate（TRUNC） 281
31.3.13 Round Off（ROUND） 281
31.4 BACKGROUND OPERATION OF MATH FUNCTION 282
31.5 TEACH MATH FUNCTION INSTRUCTION 282
31.6 RESTRICTION OF TEACHING MATH FUNCTION 284
31.7 EXCEPTIONS AND RESTRICTION 285

32 SERVO TOOL CHANGE FUNCTION 286
32.1 OUTLINE 286
32.1.1 Feature of Function 286
32.1.2 Basic Specification 286
32.1.3 Restrictions 286
32.1.4 System Configuration 288
32.1.5 Outline of Installation 289
32.2 INITIAL SETUP 289
32.3 PRELIMINARY TOOL ATTACH OPERATION 290
32.4 TOOL CHANGE SETUP 292
32.5 SETTING THE REFERENCE POSITION 297
32.5.1 Battery-less Type Tools 297
32.5.2 Battery-Mounted Type Tools 298
32.6 TOOL CHANGE INSTRUCTION 298
32.6.1 TOOL DETACH Instruction 298
32.6.2 TOOL ATTACH Instruction 299
32.6.3 Sample Program 299
32.6.4 Forward Execution (FWD) 299
32.6.5 Backward Execution (BWD) 300
32.7 TOOL CHANGE SEQUENCE 300
32.8 TOOL CHANGE STATUS 301
32.9 TEACHING 301
32.9.1 Notice for Teaching 301
32.9.2 Sample Program 302
32.10 CONSTRUCTION OF SERVO TOOL CHANGE SCREEN 302
32.11 TOOL CHANGE INITIAL SETUP 303
32.11.1 Setting Motion Parameters for Servo Tool Axes 303
32.11.2 Assigning Tool Numbers to Servo Tool Axes 303
32.11.3 Setting System Variables 305

32.12 TOOL CHANGE REFERENCE POSITION SETUP METHOD (BATTERY-LESS TYPE) 305
32.12.1 Reference Position Setup for Calibration Types 3 and 4 305
32.12.2 Reference Position Setup for Calibration Types 5 and 6 307
32.12.3 Quick Mastering Reference Position Setup 309
32.13 TROUBLESHOOTING 310
32.13.1 The Attach Instruction is Executed when the Tool is not Attached. 310
32.13.2 The Robot Stopped during Calibration. 310
32.13.3 Calibration Motion Failed. 311
32.13.4 A Different Tool from that Specified by the Attach Instruction is Attached. 311
32.13.5 The Attached Tool has been Detached by Mistake (without Using the Detach Instruction). 312
32.13.6 The Tool Axis of a Detached Tool has Moved. 312
32.13.7 The Battery Voltage has Fallen. 312
32.13.8 The Battery Ran Low while the Tool was Detached. 313

33 OPERATION WITHOUT SHIFT FUNCTION 314
33.1 JOG OPERATION WITHOUT SHIFT 314
33.2 TEST EXECUTION OPERATION WITHOUT SHIFT 315

34 HIGH SPEED SHIFT KEY FUNCTION 317
34.1 SETUP FOR HIGH SPEED SHIFT KEY FUNCTION 317
34.2 HIGH SPEED SHIFT JOG OPERATION 317
34.3 HIGH SPEED SHIFT TEST EXECUTION OPERATION 318

35 *i*RDIAGNOSTICS 320
35.1 ROBOT CONDTION DETECTION 320
35.1.1 Setup 320
35.1.2 Execute Program 323
35.1.3 Robot Condition Detection Status 323
35.1.4 After Replacement 323
35.2 SERVO DIAGNOSIS 324
35.3 MOTION PROFILER 325
35.3.1 Setup 326
35.3.2 Results 327
35.3.2.1 Detail 328
35.3.2.2 Details Comparison 331

36 MENU UTILITY FUNCTION 333
36.1 SETUP ABOUT MENU UTILITY 333
36.1.1 Prompt Box Msg 334
36.1.2 Prompt Box Yes/No Menu 336
36.1.3 List Menu 339
36.1.4 Status Menu 341
36.1.5 Operator Entry Menu 344

37 4D GRAPHICS FUNCTION 348
37.1 OVERVIEW 349
37.1.1 Graphic Models 349
37.1.2 Operation Procedure 349
37.2 4D GRAPHICS SCENE 352
37.2.1 4D GRAPHICS Display 352
37.2.1.1 Visual jog 353

37.2.1.2 Jog preview 354
37.2.1.3 Set visibility: 4D GRAPHICS display 354
37.2.2 4D GRAPHICS Node Map 355
37.2.2.1 4D GRAPHICS edit node map 356
37.2.2.2 4D GRAPHICS select node map 357
37.2.2.3 Set Visibility: node map position number 357
37.2.2.4 Data supported by node map 357
37.2.3 4D GRAPHICS Frame Display 358
37.2.4 4D GRAPHICS TCP Trace 358
37.2.5 4D GRAPHICS Position Register 359
37.2.5.1 Set Visibility: position register number 360
37.2.6 4D GRAPHICS DCS 360
37.3 4D EDITOR FUNCTION 361
37.3.1 Overview 361
37.4 FULL SCREEN 4D DISPLAY 362
37.4.1 Overview 362
37.4.2 Setup 362
37.4.2.1 Requirements 362
37.4.2.2 Configuring Microsoft® Internet Explorer 362
37.4.3 Operation 362
37.4.3.1 Accessing the Full Screen 4D Display 362
37.4.3.2 View adjustment mode 364
37.4.3.3 Scene visibility 365

38 DATA TRANSFER BETWEEN ROBOTS FUNCTION 367
38.1 TERMINOLOGY 368
38.2 SETUP 368
38.3 TCP/IP SETUP FOR ROBOGUIDE 369
38.4 STANDARD DATA TRANSFER PROGRAM 370
38.4.1 Program to Get Numeric Register 370
38.4.2 Program to Set Numeric Register 371
38.4.3 Program to Get Position Register 371
38.4.4 Program to Set Position Register 372
38.5 RECOVERY FROM ERROR 374
38.6 KAREL BUILT-IN 375
38.6.1 RGET_PORTCMT Built-in ROUTINE 375
38.6.2 RGET_PORTSIM Built-in ROUTINE 376
38.6.3 RGET_PORTVAL Built-in ROUTINE 377
38.6.4 RGET_PREGCMT Built-in ROUTINE 378
38.6.5 RGET_REG Built-in ROUTINE 378
38.6.6 RGET_REG_CMT Built-in ROUTINE 379
38.6.7 RGET_SREGCMT Built-in ROUTINE 380
38.6.8 RGET_STR_REG Built-in ROUTINE 380
38.6.9 RNUMREG_RECV Built-in ROUTINE 381
38.6.10 RNUMREG_SEND Built-in ROUTINE 382
38.6.11 RPOSREG_RECV Built-in ROUTINE 383
38.6.12 RPOSREG_SEND Built-in ROUTINE 384
38.6.13 RSET_INT_REG Built-in ROUTINE 385
38.6.14 RSET_PORTCMT Built-in ROUTINE 385
38.6.15 RSET_PORTSIM Built-in ROUTINE 386
38.6.16 RSET_PORTVAL Built-in ROUTINE 387
38.6.17 RSET_PREGCMT Built-in ROUTINE 387
38.6.18 RSET_REALREG Built-in ROUTINE 388
38.6.19 RSET_REG_CMT Built-in ROUTINE 389

38.6.20 RSET_SREGCMT Built-in ROUTINE 389
38.6.21 RSET_STR_REG Built-in ROUTINE 390
38.7 TIME OUT AND RETRY 390
38.8 LIMITATIONS 391
38.9 CAUTION 391
38.10 CONFLICT BETWEEN DATA WRITES 391
38.11 TROUBLE SHOOTING 392

39 TOUCH SENSING 394
39.1 ASSIGNMENT OF TOUCH SENSING I/O 396
39.2 SETUP OF TOUCH FRAME 398
39.3 SEARCH PATTERN 401
39.4 TOUCH SCHEDULE 405
39.5 TOUCH SENSING PROGRAMMING 410
39.6 EXECUTION OF TOUCH SENSING PROGRAM 416
39.7 TOUCHING UP OF TOUCH SENSING PROGRAM 419
39.8 MULTIPLE SEARCHES 422
39.9 COORDINATED TOUCH SENSING 423

40 TOUCH SKIP FUNCTION 426
40.1 TOUCH SKIP SCREEN 426
40.2 TOUCH SKIP PROGRAM 426
40.3 CAUTIONS 427

41 MROT INSTRUCTION 428
41.1 HOW TO USE MROT 428
41.2 LIMITATIONS 428

42 ROBOT ISOLATION FUNCTION 430
42.1 SPECIFICATION 430
42.2 METHOD OF OPERATING 430
42.2.1 Operation Panel 430
42.2.2 DCS Safety Signal: RPI 430
42.3 SYSTEM VARIABLE: $ROBOT_ISOLC 431
42.4 TEACHING TO MOTION INSTRUCTION 432

43 ANTI-DEFLECTION FOR EXTERNAL FORCE 433
43.1 PREPARATION TO CREATE COMPENSATION PROGRAM 433
43.2 HOW TO COMPENSATE 434
43.3 SETUP CUSTOMIZATION 435
43.4 RESTRICTIONS 435

44 INTERFACE PANEL FUNCTION 436
44.1 HOW TO DISPLAY PANEL 437
44.2 INTERFACE PANEL 438
44.3 INTERFACE PANEL SETUP SCREEN 440
44.4 BUTTON TYPE SETUP SCREEN 441
44.5 SETTING OF TYPE OF BUTTON 444
44.5.1 Copy and Paste 444

44.6 BUTTON DETAIL SETUP SCREEN....446
44.6.1 Preview of Button....446
44.6.2 Error Display of Button....448
44.7 TYPE OF BUTTON....449
44.8 COMMON PROPERTIES....449
44.9 OPERATION CONDITION....451
44.9.1 Button Types Supported....451
44.9.2 Operation Condition Setup Screen....452
44.10 PUSH BUTTON....454
44.11 PUSH BUTTON LAMP....455
44.12 2 CONTACT POINT SWITCH....456
44.13 LAMP....458
44.14 DIGITAL SWITCH....459
44.15 DIGITAL DISPLAY....460
44.16 MISCELLANEOUS SETTING SCREEN....461
44.17 EXTERNAL I/F PANEL SELECTION SETUP SCREEN....464
44.18 BACKUP AND RESTORE....464
44.19 LIMITATIONS....465

45 SPECIAL JOG SEQUENCE....466
45.1 USAGE....466
45.2 SETUP....466
45.3 APPLICATION....466

46 MOTION INSTRUCTION ENHANCED EDITING....468
46.1 MOTION INSTRUCTION INSERT AND INSTRUCTION DELETE FUNCTION....468
46.1.1 Usage....468
46.1.2 Precautions....469
46.2 DATA OFFSET FUNCTION....470
46.2.1 Feed Rate Conversion....470
46.2.2 Position Data Conversion (direct method)....473
46.2.3 Position Data Conversion (2 point teach method)....476
46.2.4 Precautions....478

47 FINISHING FUNCTION PACKAGE....479

48 JOINT POSITION OUTPUT FUNCTION....480
48.1 HOW TO USE....480
48.1.1 Setting....480
48.1.2 Note....480
48.1.3 Example of a Setting....481

49 EXPANDED REGISTERS FUNCTION....482
49.1 SETTING THE NUMBER OF REGISTERS....482
49.2 SAVEING AND LOADING FILES....483
49.2.1 LOADING NOT EXPANDED xxxREG.VR....483
49.2.2 LOADING EXPANDED xxxREG.VR....483

50 STITCH FUNCTION....485
50.1 SPECIFICATION....485

50.1.1 Instruction....485
50.1.2 Stitch Condition....485
50.1.3 Flow of Stitch Process....487
50.1.4 Other Specifications....488
50.2 ADJUSTMENT....489
50.3 LIMITATIONS....490
50.4 APPENDIX....491

51 VISUAL DIAGNOSTICS....492
51.1 CREATING AND EDITING VISUAL DIAGNOSTIC SCREENS....493
51.2 MANAGING SCREENS....500
51.3 VIEWING VISUAL DIAGNOSTIC SCREENS....501

52 PDF VIEWER FUNCTION....504
52.1 OPENING A PDF DOCUMENT....504
52.2 NAVIGATING WITH THE PDF VIEWER....505

53 HELP AND DIAGNOSTICS DISPLAY....506
53.1 ONLINE HELP FUNCTION....506
53.2 ALARM CAUSE/REMEDY DISPLAY FUNCTION....507
53.3 HELP/DIAGNOSTICS SCREEN....510
53.4 HELP/DIAGNOSTICS MENU....510

54 MAINTENANCE REMINDER....512
54.1 ADDITIONAL OPTION....512
54.1.1 Intelligent Grease Change Reminder(J766)....512
54.1.2 Intelligent Cable Change Reminder (S511)....512
54.2 MAIN MENU....513
54.3 SETUP....513
54.3.1 Common Setting....513
54.3.2 Maintenance Item Setting....515
54.3.3 Ambient Temperature Setting....517
54.4 CHECK MAINTENANCE TIME AND COMPLETE MAINTENANCE....518
54.4.1 Check Maintenance Time....518
54.4.2 Maintenance Remind....518
54.4.3 Upon Completion of Maintenance....519
54.5 ZDT / *i*RConnect....521
54.5.1 Setup for Maintenance Reminder....521
54.6 MAINTENANCE RECORD....522
54.6.1 Display Maintenance Record....522
54.6.2 Maintenance Record File....523
54.7 ROBOT SETTING....524
54.7.1 i-grease Change....524
54.7.2 i-cable Change....524

55 HMI DEVICE COMMUNICATION....525
55.1 CONNECTION OF HMI DEVICE....526
55.1.1 Modbus RS-232-C Connection....526
55.1.2 Modbus/TCP Slave Ethernet Connection....529
55.1.3 OPC UA Server Ethernet Connection....529
55.2 MODBUS DATA MODEL....530
55.2.1 Modbus data model....530

55.2.2 OPC UA Organization of Modbus data model....530
55.2.3 Correspondence of Modbus Address to Robot Data....531
55.2.4 Modbus Function Code....532
55.3 ASSIGNMENT OF HOLDING REGISTERS....532
55.3.1 Data type of Holding Registers....534
55.3.2 Assign Robot Registers....534
55.3.3 Assign Position Registers....536
55.3.4 Assign String Registers....540
55.3.5 Assign Current Position....540
55.3.6 Assign Alarm History....542
55.3.7 Assign Program Execution Status....544
55.3.8 Assign System Variables....545
55.3.9 Assign comment....549
55.3.10 Assign I/O data and simulation status....551
55.3.11 Assign Integrated PMC address data....552
55.3.12 Assign Symbol and Comment of Integrated PMC address....553
55.3.13 Hints....554
55.4 DYNAMIC ASSIGNMENT TO HOLDING REGISTERS....555
55.5 OPC UA DATA MODEL....556
55.5.1 RobotInformation....557
55.5.2 MotionDeviceSystem....558
55.5.2.1 MotionDevices....559
55.5.2.2 Controllers....562
55.5.2.3 SafetyStates....565

56 FAULT & INCIDENT REPORT....566
56.1 ALARMS REPORTING SCREEN....566

57 CORETEC BEAD INSPECTION INTERFACE FUNCTION....568
57.1 CABLE CONNECTION....568
57.2 SETTING (SYSTEM VARIABLES)....569
57.3 PROGRAM EXAMPLE....570

58 RSI INTERFACE....572
58.1 FORMAT OF THE TELEGRAMS SENT BY THE ROBOT....573
58.2 SETTING....574
58.2.1 Configuration of the Data Connection....575
58.3 KAREL PROGRAMS AND KAREL BUILT-IN ROUTINES....575
58.3.1 RSI_CREATE_I( )....576
58.3.2 RSI_DELETE_I( )....576
58.3.3 RSI_RESET_I( )....576
58.3.4 RSI_ON_I ( )....577
58.3.5 RSI_OFF_I ( )....577
58.3.6 RSI_WRXML_I ( )....578
58.4 DETAILS....578

59 SHAPE GENERATION FUNCTION....579
59.1 OVERVIEW....579
59.1.1 Program used for Shape Generation Function....580
59.2 SHAPE GENERATION SETUP....580
59.2.1 Basic Setup....581
59.2.2 Common Setting of All Shapes....584
59.2.3 Specific Setup of Circular Shape....585

59.2.4 Specific Setup of Slot Shape 586
59.2.5 Specific Setup of Hexagon Shape 587
59.2.6 Specific Setup of Rectangle Shape 588
59.2.7 Specific Setup of Keyhole Shape 588
59.2.8 Specific Setup of Pommel Shape 589
59.2.9 Custom 590
59.3 SHAPE GENERATION UTILITY 592
59.3.1 Setting at UTILTIES screen 592
59.3.2 Create and Edit Shape program 592
60 *i*RCalibration SIGNATURE 595
60.1 OVERVIEW 595
60.2 BENEFITS OF SIGNATURE CALIBRATED ROBOT 596
60.3 CUSTOM SIGNATURE CALIBRATION 597
60.3.1 Calibration System 597
60.3.2 Calibration Type 598
60.3.3 Custom Signature Calibration Preparation 599
60.3.4 Position Program 600
60.3.5 Coordinate Systems 602
60.4 BASIC FUNCTION 604
60.4.1 *i*RCalibration Signature Screen 604
60.4.2 Calibration Type 605
60.4.3 Setup Screen 606
60.4.4 Create Calibration Program Screen 607
60.4.5 Run Calibration Program Screen 608
60.4.6 Calculate Parameters Screen 611
60.4.7 View/Apply Calibration Result 612
60.4.8 Create Signature Calibration Parameters Setup File 613
60.5 Advanced Function 614
60.5.1 Calibration Recovery 614
60.5.2 TCP/Uframe Calibration 615
60.5.3 Reference Frame Function 617
60.5.4 Enable/Disable iRCalibration Signature Compensation 618
60.6 iRCalibration Signature Status Screen 619
60.7 Mastering 620
60.7.1 Mastering for Signature Calibrated Robot 620
60.7.2 Quick Mastering 620
60.7.3 Monitoring the Change of Mastering Data 621
60.8 Cautions 622
60.8.1 Accuracy Around Singularity 622
60.8.2 Restrictions 622
61 NC PROGRAM CONVERSION 623
61.1 OVERVIEW 623
61.1.1 NC program syntax 623
61.1.2 Functional Specifications 623
61.2 OPERATION 624
61.2.1 Conversion Operation 624
61.2.1.1 Load NC program in FILE menu 624
61.2.1.2 COPY NC program to MD drive 625
61.3 NC PROGRAM SYNTAX 625
61.3.1 O code – Program Name/Number 625
61.3.2 N code – Line numbers 626

61.3.3 G code – Motion modifiers 626
61.3.4 F code – Feed Rate / Speed 627
61.3.5 Position data 627
61.3.6 K code - Config string 628
61.3.7 M code – Miscellaneous codes 628
61.3.7.1 Pre-defined M-codes 629
61.3.7.2 Special M-codes 629
61.3.7.3 Predefined M-codes 629
61.3.7.4 Other (user defined) M-codes 631
61.4 SYSTEM VARIABLES 631
61.4.1 $NC_SYTX_TBL 631
61.4.2 $NC_GCOD_TBL : To define G-code numbers with their meaning. 632
61.4.3 $NC_MCOD_TBL : To define M-code numbers with their meaning. 632
61.4.4 $NC_DEFMODAL 633
61.4.5 $NC_OFST_CMD 634
61.4.6 $NC_TOFS_CMD 634
61.4.7 $NC_VOFS_CMD : Command string to be used for Vision offset command 634
61.4.8 $NC_DEST_DRV 634
61.5 NC PROGRAM CONVERSION EXAMPLE 635
62 FANUC FIBER LASER INTERFACE 638
62.1 OVERVIEW 638
62.1.1 Necessary Software Option 638
62.1.2 Caution 638
62.1.2.1 Restriction of the number of group 638
62.1.2.2 For the safety 638
62.1.2.3 For multi-task 639
62.2 CONNECTION OF LASER AND ROBOT 639
62.2.1 Ethernet cable connection to robot 640
62.2.2 Ethernet Cable Connection to Laser Oscillator 641
62.2.3 I/O and Data Communication Setting 642
62.2.3.1 Robot ID setting (in case of connecting several robots) 642
62.2.3.2 Communication setting 643
62.2.3.3 Confirm communication setting 645
62.2.3.4 Communication setting on the laser oscillator side 647
62.3 SETTING OF LASER 647
62.3.1 Laser Initial Setup 648
62.3.2 Laser Setup 649
62.3.3 Time Sharing Control Setup 652
62.4 STARTING OF LASER OSCILLATOR 653
62.5 LASER I/O 654
62.5.1 Laser Oscillator I/O 654
62.5.1.1 Laser oscillator output screen 654
62.5.1.2 Laser oscillator input screen 660
62.5.2 Peripheral Equipment I/O 663
62.5.2.1 Peripheral equipment output screen 663
62.5.2.2 Peripheral equipment input screen 664
62.6 LASER INSTRUCTION 665
62.6.1 Laser Instruction 665
62.6.1.1 PRE_LS instruction (preparation instruction for ls instruction) 665
62.6.1.2 LS instruction (laser processing start instruction) 666
62.6.1.3 LE instruction (laser processing end instruction) 670
62.6.1.4 LP instruction (piercing instruction) 672
62.6.1.5 Teaching laser instruction 674

62.6.2 Sub Laser Instruction....675
62.6.2.1 Power control instruction....675
62.6.2.2 Slope up instruction....675
62.6.2.3 Slope down instruction....675
62.6.2.4 Wire control instruction....675
62.6.2.5 Minute output request instruction....675
62.7 SWITCHING CH# FOR LASER CTRL RQ....676
62.7.1 Enable/Disable Ch# Laser Ctrl Rq Switching....676
62.7.2 Enable/Disable Ch# Laser Ctrl Rq....677
62.7.3 Expansion of Peripheral Equipment I/O....677
62.7.4 Ch (Channels) Switching by Pre_Ls Instruction....678
62.8 POWER CONTROL....680
62.8.1 Details of Power Control....680
62.8.2 Power Control Schedule....681
62.8.3 Power Control Instruction....681
62.8.4 Processing Speed Allowable Variation....683
62.8.5 Notifications For Power Control....683
62.9 SLOPE CONTROL....684
62.9.1 Slope up Control....684
62.9.1.1 Details of slope up control....684
62.9.1.2 Slope up schedule....685
62.9.1.3 Slope up instruction....685
62.9.2 Slope Down Control....687
62.9.2.1 Details of slope down control....687
62.9.2.2 Slope down schedule....688
62.9.2.3 Slope down instruction....689
62.9.3 Notifications for Slope Control....691
62.10 WIRE CONTROL....691
62.10.1 Wire Control Instruction....692
62.10.2 Wire Contol Signal....693
62.10.2.1 Wire feed on signal....693
62.10.2.2 Wire feed speed signal....693
62.10.2.3 Wire feed OK signal....694
62.11 MINUTE OUTPUT MODE....694
62.11.1 Minute Output Request Instruction....695
62.12 DIRECT POWER OVERRIDE MODE....696
62.12.1 Direct Power Override Mode in Laser Oscillator....696
62.12.2 Direct Power Override Mode in Robot....697
62.12.3 Enable/Disable Direct Power Override Mode....697
62.12.4 Switching to Direct Power Override Mode by PRE_LS....698
62.12.5 LS_PO Instruction....699
62.13 LS_TA INSTRUCTION WITH TIME ADJUSTMENT....703
62.13.1 LS_TA Instruction....703
62.14 LASER STATUS SCREEN....706
62.15 PROCESSING STATE RECORD FUNCTION....709
62.16 ALARM....710
62.16.1 Laser Alarm....710
62.16.1.1 Display in status window....710
62.16.1.2 Display in the active alarm screen....711
62.16.1.3 Display in laser status screen....711
62.16.1.4 Release and recover of laser alarm....711
62.16.2 Trouble Shooting About This Function'S Alarm....712
62.17 APPENDIX....719
62.17.1 Confirm Communication Setting for Laser Oscillator....719

63 VIA INSTRUCTION 722
63.1 HOW TO USE 722
63.1.1 DEFAULT 722
63.1.2 FINE 722
63.1.3 CNT 722
63.2 LIMITATIONS 723

64 CRW SHAPE GENERATION FUNCTION 724
64.1 OVERVIEW 724
64.1.1 Necessary Software Option 724
64.1.2 Supported Shapes 724
64.1.3 Common Settings 725
64.2 WELDING SHAPE PROGRAM CREATION BY *i*HMI 725
64.2.1 CRW Shape Generation Screen 725
64.2.2 Creation Of Welding Shape Program 725
64.3 EXECUTION OF WELDING SHAPE PROGRAM 727
64.3.1 Execution Of A Point Welding Shape Program 727
64.3.2 Execution Of A Path Welding Shape Program 727

65 SPLINE MOTION INSTRUCTION 729
65.1 NORMAL MOTION 729
65.2 POSITIONING PATH OF THE SPLINE MOTION 732
65.3 SINGLE STEP EXECUTION 732
65.4 BACKWARD EXECUTION 732
65.5 THE EFFECT OF ONE TEACHING POINT ON THE PATH 733
65.6 CASES SPLINE CANNOT BE PLANNED 733
65.6.1 Shortage of the Spline Motion Instruction 733
65.6.2 Teaching of Same Position 734
65.6.3 Case the Path Has High Curvature 734
65.7 RESUME AFTER PAUSE 735
65.7.1 Resume After JOG 735
65.7.2 Pause and Resume from Another Spline Motion Instruction 735
65.7.3 Resume After a Pause During Backward Execution 736
65.7.4 Start Program from Spline Motion Instruction 736
65.8 LOGIC INSTRUCTIONS BETWEEN SPLINE MOTION 736
65.9 AVAILABLE MOTION OPTIONS 738
65.10 CHANGE POSITION DATA DURING S MOTION EXECUTION 739
65.11 RESTART AFTER CHANGING PROGRAM 740
65.12 RESTRICTIONS 741

66 LEARNING PATH CONTROL 742
66.1 OVERVIEW 742
66.1.1 Procedure 742
66.2 HARDWARE AND SOFTWARE 742
66.2.1 Hardware 742
66.2.2 Software 742
66.3 LPC SETUP MENU 743
66.3.1 “Main Program” ITEM 744
66.3.2 “Start Line” ITEM 744
66.3.3 “End Line” ITEM 744
66.3.4 “Geometric Type” ITEM 744

66.3.5 “Geometric Data” ITEM 745
66.3.6 “Process Speed” ITEM 745
66.3.7 “Change Blend-In Speed” ITEM 745
66.3.8 “Path Tolerance” ITEM 746
66.3.9 “Orientation Control” ITEM 746
66.3.10 “Max. Iterations” ITEM 746
66.3.11 “Process Disabled” ITEM 746
66.3.12 “Auto Add TP Points” ITEM 746
66.3.13 “No. CIR Lines” ITEM 746
66.3.14 “LEARN” FUNCTION KEY 746
66.3.15 “RESULT” FUNCTION KEY 747
66.3.16 “APPLY” FUNCTION KEY 747
66.3.17 RELATED VIEW 747
66.4 LPC LEARNING AND PLAYBACK 748
66.4.1 Learning 748
66.4.2 Limitations 749
66.4.3 Production with LPC Playback 749
66.5 LPC BACKUP 749

67 MQTT 750
67.1 OVERVIEW 750
67.2 KAREL BUILT-IN ROUTINES 750
67.2.1 MQTT_CONNECT 750
67.2.2 MQTT_PUBLISH 751
67.2.3 MQTT_DISCON 752
67.3 KAREL PROGRAM SAMPLE 753

68 HIGH SPEED POSITION OUTPUT 756
68.1 DATA FORMATS 756
68.1.1 Binary 756
68.1.1.1 TCP Cartesian Position with Binary 757
68.1.1.2 Joint Angles with Binary 757
68.1.1.3 Variables with Binary 757
68.1.2 XML 758
68.1.2.1 TCP Cartesian Position with XML 758
68.1.2.2 Joint Angles with XML 758
68.1.2.3 Variables with XML 758
68.2 SETUP 758
68.2.1 Destination Information Settings 758
68.2.2 Communication Method 758
68.2.3 Output Data Setup 759
68.2.4 Start Output 760

69 DIGITAL I/O JOG FUNCTION 761
69.1 OVERVIEW 761
69.2 SPECIFICATION 761
69.2.1 Version 761
69.2.2 Setup 761
69.2.3 Limitations 762

70 LASER INTERFACE FUNCTION 763
70.1 OVERVIEW 763
70.1.1 Necessary Software Option 763
70.1.2 Caution 763

70.1.2.1 For the safety 763
70.1.2.2 For multi-task 763
70.2 SETUP 763
70.2.1 Laser Initial Setup 763
70.2.1.1 Switching to FANUC Fiber Laser Interface Function 767
70.2.2 Laser Setup 768
70.2.2.1 YLS Laser setup 768
70.2.2.2 YLR Laser setup 771
70.3 LASER I/O 773
70.3.1 YLS Laser Oscillator I/O 773
70.3.1.1 YLS laser oscillator output screen 773
70.3.1.2 YLS laser oscillator input screen 775
70.3.2 YLR Laser Oscillator I/O 777
70.3.2.1 YLR laser oscillator output screen 777
70.3.2.2 YLR laser oscillator input screen 779
70.3.3 Peripheral Equipment I/O 780
70.3.3.1 Peripheral equipment output screen 780
70.3.3.2 Peripheral equipment input screen 781
70.4 LASER ENABLE/DISABLE 782
70.4.1 Laser Enalbe/Disable Screen 782
70.5 LASER INSTRUCTION 784
70.5.1 PRE_LS Instruction 784
70.5.2 LS Instruction 786
70.5.2.1 When there is LS before PRE_LS (recommended) 787
70.5.2.2 When there is not LS before PRE_LS 789
70.5.3 LS_SCH Instruction 792
70.5.4 LE Instruction 793
70.5.5 Teaching laser instruction 794
70.6 TIMING CHART 795
70.6.1 YLS Timing Chart 796
70.6.1.1 From laser start to end when is Time-sharing is DISABLE 796
70.6.1.2 When laser error or robot E-Stop occurs 797
70.6.1.3 From laser start to end when Time-sharing is ENABLE 798
70.6.2 YLR Timing Chart 799
70.6.2.1 From laser start to end 799
70.6.2.2 When laser error or robot E-Stop occurs 800
70.7 POWER CONTROL 800
70.7.1 Details 800
70.7.2 Power Control Schedules 801
70.7.3 Power Control Instruction 802
70.7.4 Speed Tolerance 804
70.7.5 Notes 804
70.8 POWER CONTROL Ⅱ 805
70.8.1 Details 805
70.8.2 Notes 808
70.9 SLOPE CONTROL 810
70.9.1 Slope Up Control 810
70.9.1.1 Details 810
70.9.1.2 Slope up schedule 811
70.9.1.3 Slope up instruction 812
70.9.2 Slope Down Control 814
70.9.2.1 Details 814
70.9.2.2 Slope down schedules 814
70.9.2.3 Slope down instruction 815
70.9.3 Notes 817

70.10 POWER COMPENSATION 818
70.10.1 Notes 821
70.11 WIRE CONTROL 821
70.11.1 Wire Control Setup 822
70.11.2 Wire Control Instruction 823
70.11.3 Wire Control Signal 825
70.11.3.1 Wire Feed output 825
70.11.3.2 Wire Feed Speed output 825
70.11.3.3 Wire Feed input 825
70.11.4 Notes 825
70.12 LS_TA INSTRUCTION WITH TIME ADJUSTMENT 826
70.12.1 LS_TA Instruction 826
70.12.2 LS_SCH_TA Instruction 828
70.13 PROCESSING STATE RECORD FUNCTION 828
70.14 ALARM 830
70.14.1 Trouble Shooting About This Function's Alarm 830
70.15 LASER INTERFACE UNIT 831
70.15.1 Outline 831
70.15.2 System 832
70.15.3 Hardware Specifications 832
70.15.3.1 Overview 832
70.15.3.2 External view and dimensions 834
70.15.3.3 Connection 835
70.15.3.4 Mounting position in the Robot controller 841
70.15.4 Laser Command Signal Analog Output 841

71 SERVO TOOL FUNCTION 843
71.1 SERVO TOOL FUNCTION 844
71.2 SETUP THE SERVO TOOL FUNCTION 844
71.3 PROGRAM INSTRUCTION 847
71.3.1 SERVOTOOL[schedule number] 847
71.3.2 SERVOTOOL END / SERVOTOOL END[schedule number] 847
71.3.3 TP Program Example 847
71.4 LIMITATIONS 848
71.5 CAUTIONS 849

72 AUTO BACKWARD EXIT 850
72.1 OVERVIEW 850
72.2 SETUP AND OPERATIONS 850
72.2.1 Specification Overview 850
72.2.2 Setup 851
72.2.3 System Variables 852
72.2.4 Programming 853

73 CNC CUSTOM MACRO VARIABLE ACCESS 854
73.1 OVERVIEW 854
73.2 REQUIRED OPTION 854
73.3 ROBOT SETUP 855
73.3.1 IF USE TABLET TP 856
73.3.2 IF USE iPendant 858
73.4 CNC SETUP 862
73.5 ABOUT ALARM 867

1 PREFACE

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

1.1 ABOUT MANUAL

About this manual

FANUC Robot series OPERATOR'S MANUAL (Basic Operation) (B-83284EN). describes how to operate the FANUC Robot. It is controlled by the FANUC R-30*i*B ,R-30*i*B Plus, R-30*i*B Mate, R-30*i*B Mate Plus, R-30*i*B Compact Plus, and R-30*i*B Mini Plus 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-83284EN-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-83284EN**	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-83284EN-1**	Topics: Error code listings, causes, and remedies. Use: Installing and activating the system, connecting the mechanical unit to the peripheral device and maintenance the robot system.
	Arc Welding Function OPERATOR'S MANUAL **B-83284EN-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-83284EN-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-83284EN-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-83195EN (for R-30*i*B/R-30*i*B Plus), B-84035EN (for R-30*i*B Compact Plus), B-83525EN (for R-30*i*B Mate/ R-30*i*B Mate Plus) B-83555EN (for R-30*i*B Mate Open Air) B-84175EN (for R-30*i*B Mini Plus)**	Topics: Installing and activating the system, connecting the mechanical unit to the peripheral device and maintenance the robot 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.

2 OVERVIEW

This manual describes the following software options.

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 PANEL WIZARD
Chapter 24 ENHANCED MIRROR IMAGE
Chapter 25 CUSTOMIZE SUPPORT FUNCTION
Chapter 26 KAREL PROGRAM EXECUTION HISTORY RECORD
Chapter 27 TORQUE LIMIT FUNCTION
Chapter 28 TCP SPEED OUTPUT
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 FUNCTION
Chapter 37 4D GRAPHICS FUNCTION
Chapter 38 DATA TRANSFER BETWEEN ROBOTS FUNCTION
Chapter 39 TOUCH SENSING
Chapter 40 TOUCH SKIP FUNCTION
Chapter 41 MROT INSTRUCTION
Chapter 42 ROBOT ISOLATION FUNCTION
Chapter 43 ANTI-DEFLECTION FOR EXTERNAL FORCE
Chapter 44 INTERFACE PANEL FUNCTION
Chapter 45 SPECIAL JOG SEQUENCE
Chapter 46 MOTION INSTRUCTION ENHANCED EDITING
Chapter 47 FINISHING FUNCTION PACKAGE
Chapter 48 JOINT POSITION OUTPUT FUNCTION
Chapter 49 EXPANDED REGISTERS FUNCTION
Chapter 50 STITCH FUNCTION
Chapter 51 VISUAL DIAGNOSTICS

Chapter 52 PDF VIEWER FUNCTION
Chapter 53 HELP AND DIAGNOSTICS DISPLAY
Chapter 54 MAINTENANCE REMINDER
Chapter 55 HMI DEVICE COMMUNICATION
Chapter 56 FAULT & INCIDENT REPORT
Chapter 57 CORETEC BEAD INSPECTION INTERFACE FUNCTION
Chapter 58 RSI INTERFACE
Chapter 59 SHAPE GENERATION FUNCTION
Chapter 60 *i*RCALIBRATION SIGNATURE
Chapter 61 NC Program Conversion
Chapter 62 FANUC FIBER LASER INTERFACE FUNCTION
Chapter 63 VIA INSTRUCTION
Chapter 64 CRW SHAPE GENERATION FUNCTION
Chapter 65 SPLINE MOTION INSTRUCTION
Chapter 66 LEARNING PATH CONTROL
Chapter 67 MQTT
Chapter 68 HIGH SPEED POSITION OUTPUT
Chapter 69 DIGITAL I/O JOG FUNCTION
Chapter 70 LASER INTERFACE FUNCTION
Chapter 71 SERVO TOOL FUNCTION
Chapter 72 AUTO BACKWARD EXIT
Chapter 73 CNC CUSTOM MACRO VARIABLE ACCESS

3 SOFTFLOAT FUNCTION

Usually, the robot moves accurately toward the goal specified using the teach pendant (taught point).

When the robot is used to mount workpieces on a machine tool, variances in workpiece precision may result in a shift in the workpiece position relative to the tool, thus possibly causing interference between the workpiece and tool.
A softfloat function has been added which is effective in mounting workpieces with variances in precision onto a machine tool.

The softfloat function is also very effective if the synchronization speed is unstable as in the extraction of workpieces in sync with hydraulic extrusion, and if workpieces that the robot cannot grip accurately, such as rough-machined workpieces, are to be handled.

3.1 JOINT / CARTESIAN SOFTFLOAT

Function

The joint / Cartesian softfloat function works as follows:

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

Detail

The detailed descriptions of the softfloat function follow.

- Program instruction

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

- SOFTFLOAT[n]
  The softfloat function is enabled using schedule n.
  * The setting of softfloat schedule is explained in ”Schedule setting menu”.

- SOFTFLOAT END
  The softfloat function is disabled.

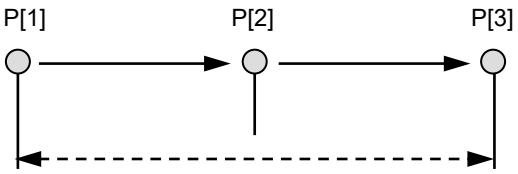
- FOLLOW UP
  When an external force is removed from a robot, it usually tries to go back to the taught point. However, this instruction causes the robot to assume that the current position is the taught point, and prevents it from going back to the taught point.

- Softfloat function effective range

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

- Sole instruction
  The softfloat function is enabled after the end of the motion specified on the line preceding the solely specified SOFTFLOAT[n] instruction.
  In the following example, the softfloat function is enabled after the motion specified on line 1 ends, and disabled by SOFTFLOAT END on line 5.

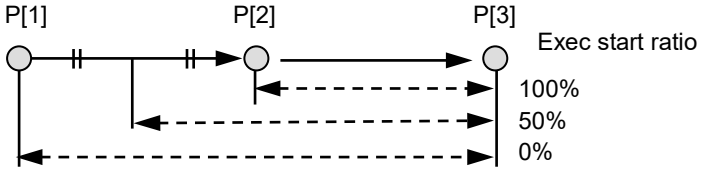
```
1: J P[1] 100% FINE
2:  SOFTFLOAT[1]
3: L P[2] 100mm/sec FINE
4: L P[3] 100mm/sec FINE
5:  SOFTFLOAT END
```



The soft float function is enabled.

- Auxiliary motion instruction
  The softfloat function becomes enabled during execution of a motion statement attached with a SOFTFLOAT [n] instruction.
  The point at which the softfloat function becomes enabled is determined by a softfloat schedule ”Exec Start Ratio”.
  Auxiliary motion instruction is specified as the ratio (from 0% to 100% in 1% steps) of a distance to be traveled before the robot reaches the taught point corresponding to a motion statement attached with a SOFTFLOAT[n].
  In the following example, the softfloat function is effective between P[1] taught using a motion statement on line 1 and P[2] taught using a motion statement on line 2 attached with the SOFTFLOAT[n] instruction.

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



The soft float function is enabled.

NOTE
The auxiliary motion instruction is not supported by Cartesian softfloat.

- Schedule setting menu

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

- List menu

- Detail menu

A function key is used to select either menu.

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

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

List menu

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

 | [ TYPE ] | GROUP | DETAIL |  |  |  |
```

F3, DETAIL ↓ F3, LIST ↑

Joint softfloat screen

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

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

→

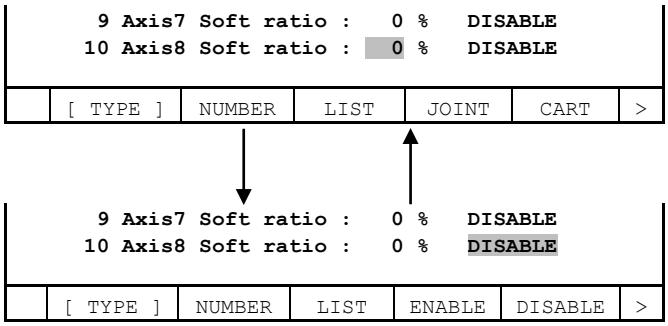
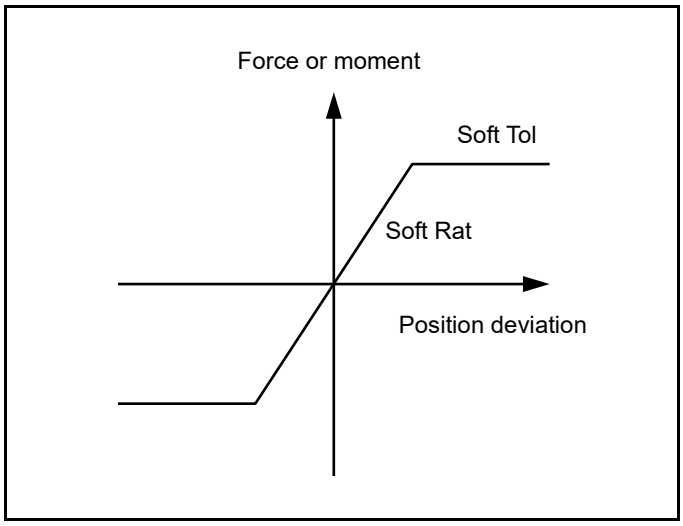
Cartesian softfloat screen

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

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

The following data can be specified on the detail menu.

Table 3.1 Setting items of softfloat detail menu

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

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

- Setting of operation area limitation

Operation area during Cartesian softfloat is able to be limited.
If the position error of TCP between commanded position and actual position in the frame which is set in softfloat schedule exceeds the threshold, alarm occurs and robot stops.
For safety, following system variables have to be set to avoid interference before the program which executes Cartesian softfloat runs.

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

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

3.2 PUSHOUT SOFTFLOAT

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

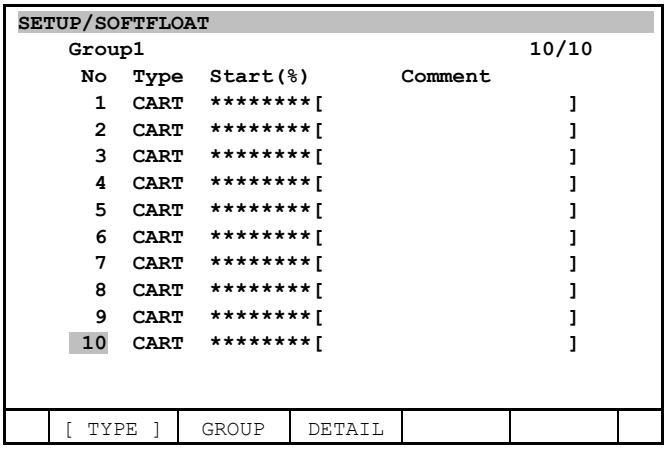
- Joint softfloat
  Softness around robot joint can be specified.
- Cartesian softfloat
  Softness along Cartesian coordinate and around Cartesian coordinate can be specified.
  It is good at contouring and face matching with robot motion.
- Pushout softfloat
  In this softfloat, force and moment necessary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

Procedure 3-1 Procedure to show pushout softfloat screen

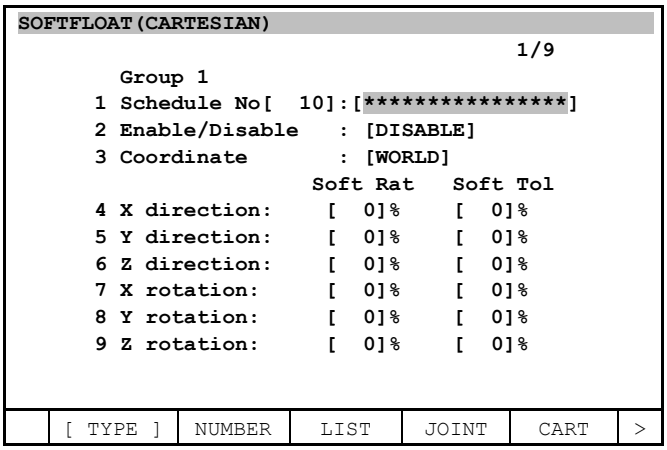
Select “PUSH” in softfloat detail screen.

Step

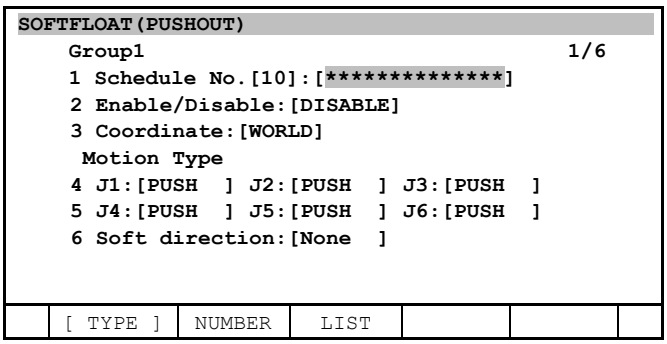
1 Press the [MENU] key to show screen list.
2 Select “6 SETUP”.
3 Push F1, [TYPE] key to show screen list.
4 Select “softfloat”. Softfloat list screen will be displayed.



5 Set cursor on schedule No. to set pushout softfloat, and push F3, DETAIL key.



6 Push NEXT key until “PUSH” are shown on function key line.
7 Push F4, PUSH key. Detail screen of pushout softfloat will be displayed.

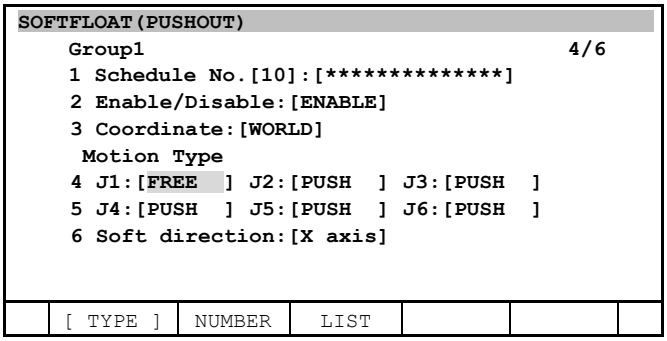


Procedure 3-2 Procedure to set pushout softfloat

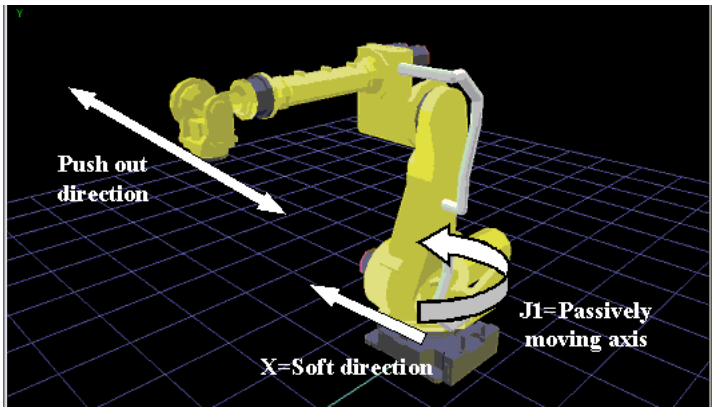
In addition to the setup of Cartesian softfloat, “motion type” and “soft direction” should be set. “Soft rat” and ”soft tol” are removed.

Step

1 Set “enable/disable” and “coordinate” same as Cartesian softfloat.
2 Set “soft direction” to the direction which should be pushout direction in the schedule No.
3 Set motion type of the axis which should move passively following the force from outside to “FREE”. Only one axis can be “FREE” in one group. However, you cannot set “FREE” for the J3 axis or later.



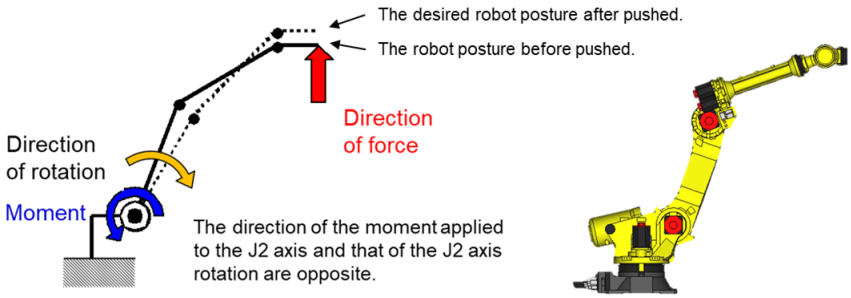
Example)
When J1 is 90 degree, to make X direction of world coordinate soft, motion type of J1 should be “FREE”.



⚠ CAUTION

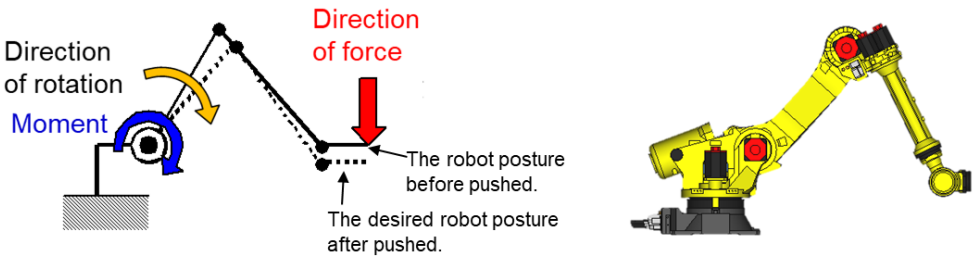
In the pushout softfloat, some axes cannot be used as a FREE axis depending on the robot posture.
The followings are examples when the J2 axis is set to “FREE” and the soft direction is set to “Z axis”.

(Bad case)



In the case shown in this figure, a force is applied in the positive direction of Z axis, and a counterclockwise moment is applied to the J2 axes. However, the direction of this moment will be opposite to the direction the J2 axis should rotate to move along the positive direction on the Z axis. In such a case, it cannot move well passively following the force. Therefore, do not use the J2 axis as “FREE”.

(Good case)



In the case shown in this figure, a force is applied in the negative direction of Z axis, and a clockwise moment is applied to the J2 axes. The direction of this moment will be the same as the direction the J2 axis should rotate to move along the negative direction on the Z axis. In such a case, you can use the J2 axis as “FREE”.

Procedure 3-3 Procedure to use pushout softfloat

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

Step

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

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

3.3 KNOWHOW OF SOFTFLOAT

Timing to start softfloat

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

Bad example)

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

Force from outside will be added if 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.

Know-how to improve softness

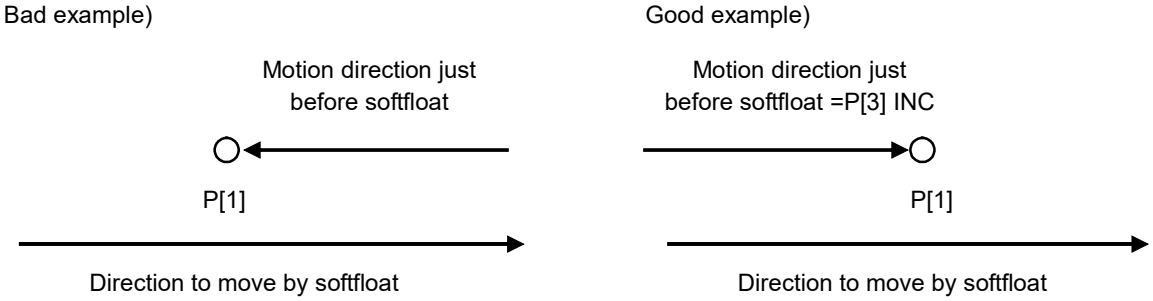
Even with the same softfloat schedule setting, the softness of softfloat differs depending on the direction of the motion just before softfloat starts.
It is caused by the feature of that softfloat is more soft to move in the same direction to the motion just before and less soft to move in the opposite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

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

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

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

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

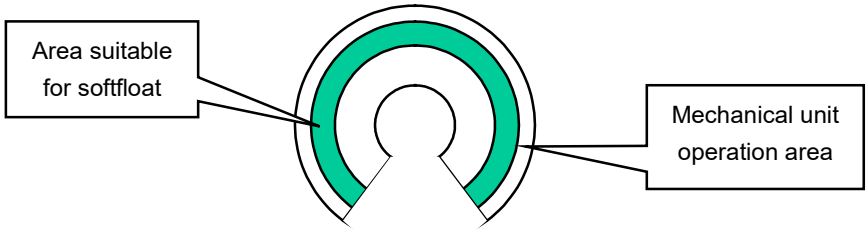


3.4 KNOWHOW OF CARTESIAN SOFTFLOAT

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

All application common know-how

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



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

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

Contouring application know-how

- Softfloat schedule setup
  Example) In case pushing Z direction

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

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

Example)

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

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

TCP set as original tool

Face matching application know-how

- Softfloat schedule setup
  Example) In case pushing Z direction

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

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

3.5 CAUTIONS / RESTRICTIONS

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

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

- Power is applied.

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

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

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

  $SFLT_DISFUP Default value: FALSE
  - If FALSE, when softfloat function is enabled, follow-up is performed at the start of each motion instruction in the program.
  - If TRUE, follow-up is not performed for individual motion instructions in the program.
- This function cannot be used with arc tools.
- When using this function together with gravity compensation function, pay attention to the following.
  Gravity compensation function calculates the compensation amount based on the command position. Even if the command position and the actual position get separated by softfloat, gravity compensation function calculates compensation amount based on the command position. Therefore, for example, even in the case of an application where the robot should be kept stationary by touching against the pressing surface with softfloat enabled, there is a possibility that the gravity compensation amount changes and the robot moves slightly after pressing.
- Force from outside should not be added when softfloat starts, when softfloat ends and when follow up is doing.
  If softfloat starts with not only gravity force but also other force from outside, robot will move above or below, and “Stop/Move error excess” or “Softfloat time out” alarm will occur.
  In such cases, please modify TP program by changing position data or by adding wait instruction as softfloat starts and ends without other force.

NOTE

Follow-up

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

4 CONTINUOUS ROTATION FUNCTION

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

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

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

4.1 OVERVIEW

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

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

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

- 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 rotation axis must satisfy the following mechanical conditions:

- The mechanism must allow continuous operation (must be free of obstacles such as stoppers).
- The gear reduction ratio (value of Gear Ratio (motor) / Gear Ratio (axis) on the setting screen, the 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 deg when the motor rotates 100 deg, Gear Ratio (motor) and Gear Ratio (axis) are as follows:
Gear Ratio (motor) = 100, Gear Ratio (axis) = 1.

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

Angle expression of continuous rotation axis

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

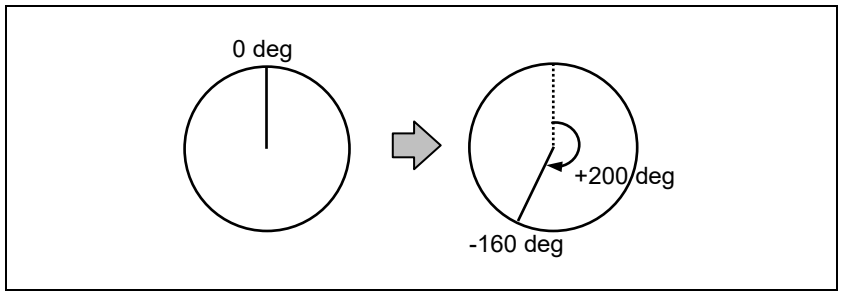


Fig. 4.1 (a) Angle on the Continuous Rotation Axis

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

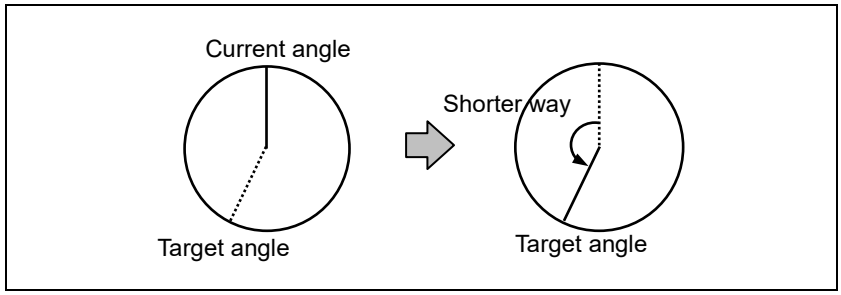


Fig. 4.1 (b) Shorter-Way Operation

4.2 SETUP

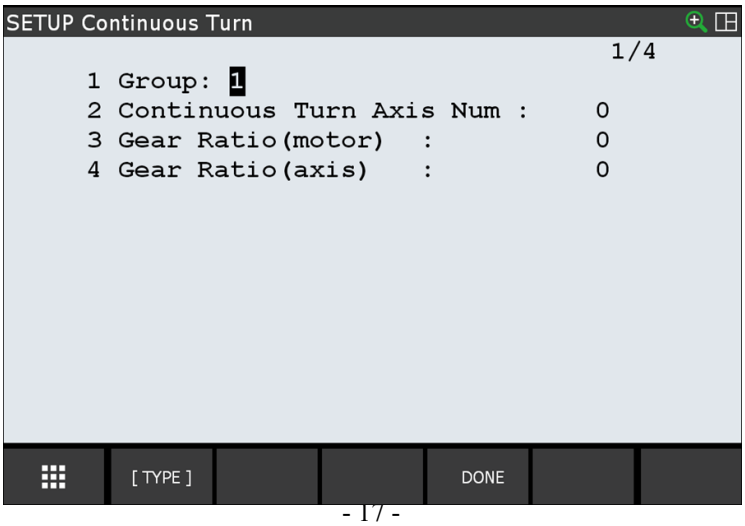
To use the function,

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

Procedure 4-1 Setting up the continuous rotation function

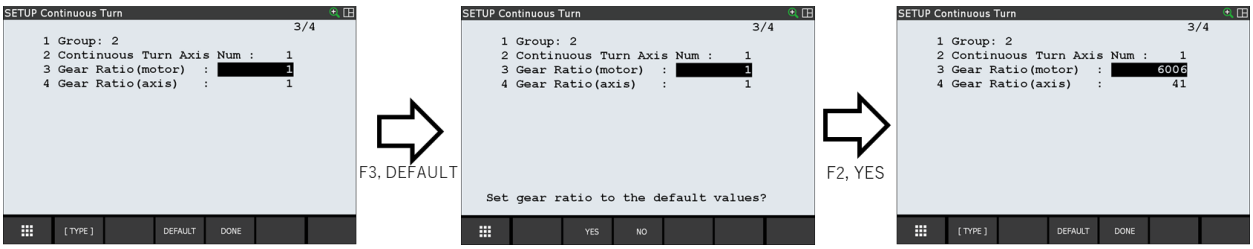
Step

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



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

6 Set the Gear Ratio for the Continuous Turn Axis of the specified Group
- Enter integer values in the range of 1 to 2147483646 in the “Gear Ratio (motor)” and “Gear Ratio (axis)”.
- For the final robot axis, enter the “Group” and “Continuous Turn Axis Num”, and move the cursor to the “Gear Ratio (motor)” or “Gear Ratio (axis)”. Then F3,DEFAULT is displayed. Press F3, DEFAULT, then a message “Set gear ratio to the default values?” is displayed. Press F2, YES to set the Gear Ratio (motor) and Gear Ratio (axis).
  Please note that F3, DEFAULT is not displayed in either of the following condition:
  - Software version is 7DF1/27(V9.10P/27), 7DF3/05(V9.30P/05) or earlier.
  - Integrated additional axis, independent axis or positioner, etc. set by customers.



7 After specifying the items, press F4, DONE.

8 Set up Constant Path function. For detail about simultaneous use of Continuous Rotation and Constant Path function, please refer to next section “Use of Continuous Rotation function with Constant Path function”.
- In case that Constant Path function is needed to be disabled by limitation (Example: Continuous Rotation axis is set in final axis of robot), Constant Path function is disabled and message “ConstPath is disabled.(Press ENTER)” is displayed.
- In case that the user can choose set Constant Path disabled or not (Example: Continuous Rotation axis is set in independent axis), “Disable Constant Path Function?” is displayed.
  - When F2, YES is pressed, Constant Path function is disabled and “ConstPath is disabled.(Press ENTER)” is displayed.
  - When F3, NO is pressed, setup of Constant Path function is NOT changed. If Constant Path function is enabled, “ConstPath is enabled.(Press the [ENTER] key)” is displayed. Else if Constant Path function is disabled, “ConstPath is disabled.(Press [ENTER] key)” is displayed.

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

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

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

Table 4.2 (a) Contents of the Continuous Rotation Setup Screen

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

4.3 USING THE FUNCTION

After setting up the continuous rotation axis, specify the start point of continuous rotation using the operation add instruction, ”continuous rotation speed instruction”.
The following ”continuous rotation speed instruction” is supported. The ”continuous rotation speed instruction” must be specified as an operation add instruction.
* The specification method is the same as that for other operation add instructions, and is therefore omitted. (→ Subsection 5.3.4, ”Teaching an Additional Motion Instruction” in FANUC Robot series OPERATOR’S MANUAL (Basic Operation) (B-83284EN))

- Continuous rotation speed instruction CTV

* where i = –100 to 100, which is the ratio of the rotation axis speed to the maximum axis speed (%)
* You can specify the continuous rotation axis speed indirectly by using a register. To use a register, press the F1 “REGISTER”, then type a register number.

4.3.1 Starting continuous rotation

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

4.3.2 Stopping continuous rotation

Continuous rotation is stopped when the first operation statement with no continuous rotation speed instruction added is started since a continuous rotation speed instruction was started.
When continuous rotation is stopped, the operation on the other axes for the same motion group also terminates. The robot, therefore, decelerates even if the positioning format for the previous operation is CNT.
The robot starts decelerating to stop on the continuous rotation axis after it has completely stopped on the other axes. At this time, the robot is not necessarily at the specified position on the continuous operation axis. Thus, the synchronization of the operation on the continuous rotation axis with the operation on the other axes (including those for other motion groups) is lost.
If an operation statement is specified next, the robot rotates in the same direction as the previous continuous rotation direction to move to the specified position.

4.3.3 Notes

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

4.3.4 Example of use

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

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

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

4.4 NOTES / LIMITATIONS

4.4.1 Functions that cannot be used together

- This function 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 together with this function if you set up the Continuous Rotation axis in other than coordinated motion groups and use programs with separate group mask as described in the section “Use of Continuous Rotation function with Constant Path function” below.)
  - Interference Prevention Area function
- When continuous rotation is to be performed on a robot axis or integrated additional axis, the following functions of Dual Check Safety function are not available.
  - Joint Position Check function for continuous rotation axes.
  - Cartesian Position Check function.
  - Cartesian Speed Check function.
- The following instructions/functions cannot be used for continuous rotation axes.
  - Asynchronous additional axis speed instruction. (The synchronous additional axis speed instruction can be used.)
  - Incremental instruction.
  - Dual Drive function.

- If the software version is 7DF1/16 (V9.10P/16) or earlier, Continuous Rotation function is not available with SR-3*i*A and SR-6*i*A series.

4.4.2 Use of Continuous Rotation function with Constant Path function

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

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

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

4.4.3 Other notes

- When continuous rotation is to be performed on a robot axis or integrated additional axis, The X and Y components of the tool coordinate system must both be 0. (Only the Z-axis component can have a value other than 0.)
  If this condition is not satisfied, the path of linear or circular or circle arc motion cannot be guaranteed in normal motion other than continuous rotation.
- This function automatically updates the mastering data (for the continuous rotation axis only) according to the amount of rotation about the continuous rotation axis. Thus, previously recorded mastering data may not match the current mastering data.
  After this function is disabled, it is not necessary to perform mastering.
- When this function is disabled, the current position on the continuous rotation axis may fall outside the stroke limits. If this occurs, move the position on continuous rotation axis within the stroke limits using jog feed or a program.
- If, on a multi-group system, the settings on the SETUP Continuous Turn screen are changed and the F4, DONE key is pressed, it is necessary to set system variable $PARAM_GROUP[group].$SV_OFF_ENB[i] (where i is an axis number) to FALSE to disable break control for all the axes for all motion groups before turning the power back on with a cold start.
- On a multi-group system, even if there are multiple continuous rotation axes, separate continuous rotation speeds cannot be specified for them.
- At the end of continuous rotation, one or more rotations about the continuous rotation may be performed to ensure smooth deceleration and stop. (The amount of rotation differs depending on the acceleration/deceleration constant.)
- Even during backward execution (single-step execution), shorter-way operation is performed on the continuous rotation axis. If, therefore, forward step execution and backward execution are performed

sequentially in an operation statement with the movement angle being very close to 180° , rotation may be performed about the continuous rotation axis in the same direction during the forward and backward executions.

- Original Path Resume feature is disabled when continuous rotation function is loaded. Even if system variable ($SCR.$ORG_PTH_RSM) set enabled, feature is disabled when cycle power.
- Groups which have continuous rotation axis will disable Cycle Time Priority feature for Cartesian motion. This will be longer cycle time and path change. When setup of Continuous Rotation axis is cleared, Cycle Time Priority feature for Cartesian motion will be restored original state. Please execute program carefully after changing of Continuous Rotation axis setup.
- If the motion speed exceeds the limit of 180[deg] / $SCR.$ITP_TIME[msec] for the continuous rotation axis, warning “CNTR-009 Warn-Cont Vel too high(G:group)” will be posted. If such motion is executed in single-step mode, alarm “CNTR-011 Axis speed exceeds lim(G:group)” will be posted.

4.5 RESET ROTATION OF POSITIONER AND INDEPENDENT AXIS BY KAREL PROGRAM

From software version 7DC3/31(V8.30P31), continuous rotation function provides a KAREL “CALIBNBT.PC” which can reset the number of rotation of positioner and independent axis.
By using CALIBNBT.PC, you can reset the position of a rotational axis to between -180 deg and +180 deg.
Without setting up the continuous rotation axis, you can rotate the axis in one direction continually.
You can also avoid the limitations imposed by the continuous rotation axis mentioned 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 continuous rotation function is not available in this case.

4.5.1 Condition to use

When following conditions are all satisfied, you can use CALIBNBT.PC.

- Target axis does not belong to robot group (i.e. a positioner or an independent axis)
- Mastering is already done before using CALIBNBT.PC
- Absolute values of upper and lower stroke limits of the axis which will be reset is smaller than the value expressed below

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

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].

4.5.2 Usage

1. Set the following two system variables.
   - $PARAM2_GRP[group].$GRATIO_DIV[axis] : Gear Ratio (motor)
   - $PARAM2_GRP[group].$GRATIO_NUM[axis] : Gear Ratio (axis)

   For the robots and FANUC' s positioners, These values are preset and do not need to be changed.
   If the software version is 7DF1/27 (V9.10P/27), 7DF3/05 (V9.30P/05) or earlier, these system variables do not need to be set (the setting has no effect).
2. Call CALIBNBT.PC in a TP program as follows:
   - Set $KAREL_ENB = 1 to display KAREL in program list menu.
   - Use CALL instruction in the TP program to execute CALIBNBT.PC.

- Add two arguments as below to call CALIBNBT.PC
  CALIBNBT (group number, axis number)
  Argument 1 : group number, 1 to 8
  Argument 2 : axis number, 1 to 9
  ex) Reset the number of rotation of group 2 / axis 1.
  CALL CALIBNBT (2, 1)
- If the target axis is dual drive axis, please specify the axis number of master group.

4.5.3 Note

- When all of the following conditions are satisfied, position error after the decimal point is accumulated at every rotation reset. Even though the position error derived from one reset process is very small, execute mastering of the axis periodically in order to clear the accumulated position error.
  - The software version is 7DF1/27 (V9.10P/27), 7DF3/05 (V9.30P/05) or earlier, or the above-mentioned system variables for the Gear Ratio (motor) and Gear Ratio (axis) are not set.
  - Pulse counts corresponding to one rotation of the axis is not an integer number.

$$Pulse\ counts\ of\ one\ axis\ rotation = \frac{Gear\ Ratio\ (motor)}{Gear\ Ratio\ (axis)} \times 2^{19}$$

(Pulse counts of one motor rotation)

Example: Suppose Gear Ratio (motor) / Gear Ratio (axis) = 2763/16
$2763/16*2^{19} = 90537984$
→integer number, no position error

- If the above-mentioned system variables for the Gear Ratio (motor) and Gear Ratio (axis) are set and when you execute a mastering of this axis after using CALIBNBT.PC, set the following system variable to 0.
  $DMR_GRP[group].$REL_SHF_ERR[axis].$OFFSET[axis]
- 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)
- At calling CALIBNBT.PC program, the number of rotation is reset so the updated position of the target axis would 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 DCS Position / Speed Check function is enabled, you cannot use CALIBNBT.PC. If you call CALIBNBT.PC when DCS Position / Speed Check function is enabled, an alarm "SYST-212 Need to apply to DCS param" occurs. If you need to use CALIBNBT.PC, disable Dual Check Safety function by setting $DCS_CFG.$EXCLUDE[Group number] to -1.

4.5.4 Error message

When following message is displayed, please check the remedy of each case.

- 'Argument missing (1: Group number)'
  Argument 1 is not assigned. Please assign group number 1 to 8.
- 'Illegal parameter (1: Group number)'
  Argument 1 is not integer. Please check Argument 1.
- 'Argument missing (2: Axis number)'
  Argument 2 is not assigned. Please assign axis number 1~9.
- 'Illegal parameter (2: Axis number)'
  Argument 2 is not integer. Please 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 illegal number. Please check arguments.
- 'Axis type is not supported'
  Axis type is not rotary axis. Please assign rotary axis to use CALIBNBT.PC.
- 'Robot axis is not supported'
  Robot axis or extend axis in robot group is assigned.
  Please assign independent axis from robot group.
- 'Find slave axis fail'
  Dual drive axes are assigned but slave axis was not found. Please check setting of dual drive.
- 'Current angle is too large'
  Current angle of axis is too large. Confirm that value of upper and lower limit of axis range.
- 'Do not execute during moving '
  KAREL was executed while robot moving.
  Execute KAREL after completion of FINE term-type motion.
- ‘CNTR-28 Cannnot use ContTurn axis(CALIBNBT)’
  Continuous Turn Axis cannot be specified in the argument of 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

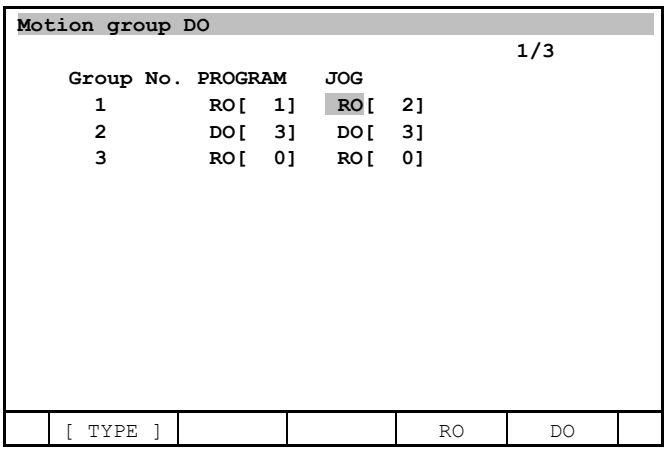
When the teach pendant is disabled, all signals turn off.
When the teach pendant is enabled, the signal for the currently selected operation group on the teach pendant turns on, while the other signals turn off.

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

6.1 AUTOMATIC ERROR RECOVERY FUNCTION

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

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

- Resume program function
- Fast exit/entry feature

6.2 RESUME PROGRAM FUNCTION

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

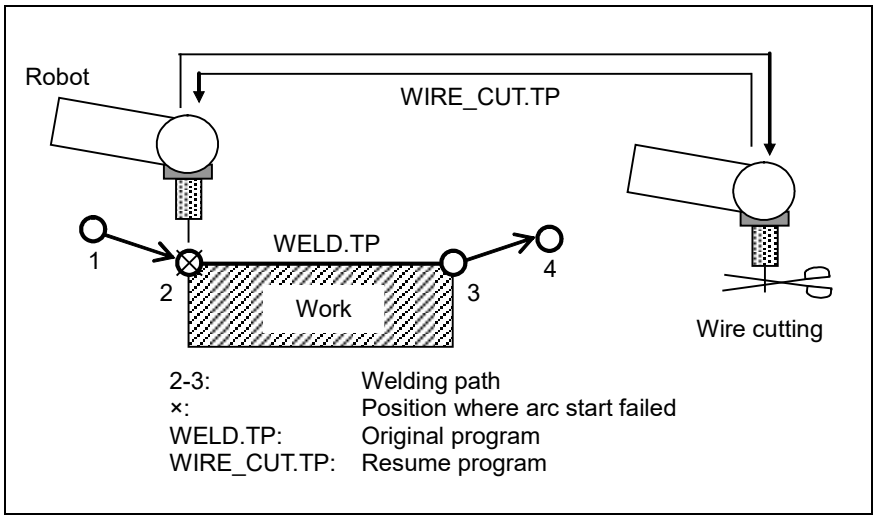


Fig.6.2 Resume program function

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

6.3 FAST EXIT/ENTRY FEATURE

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

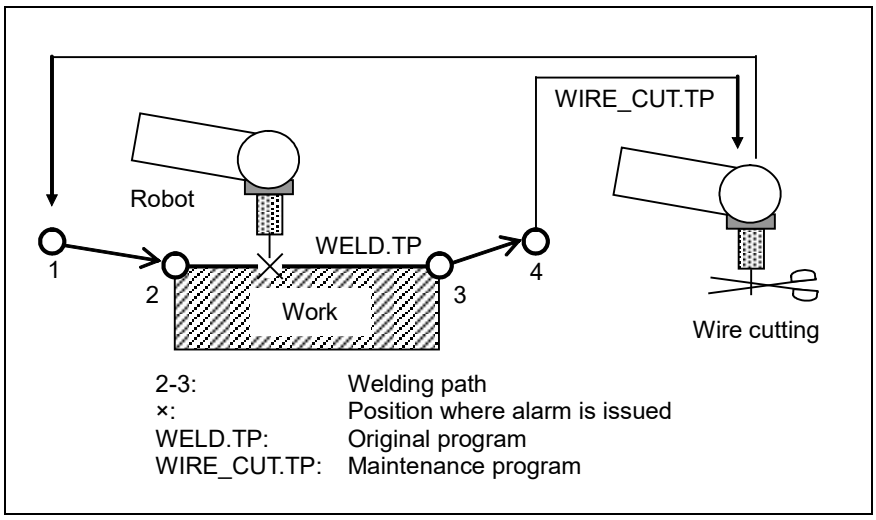


Fig.6.3 Fast exit/entry feature

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

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

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

6.4 RESUME_PROG INSTRUCTION

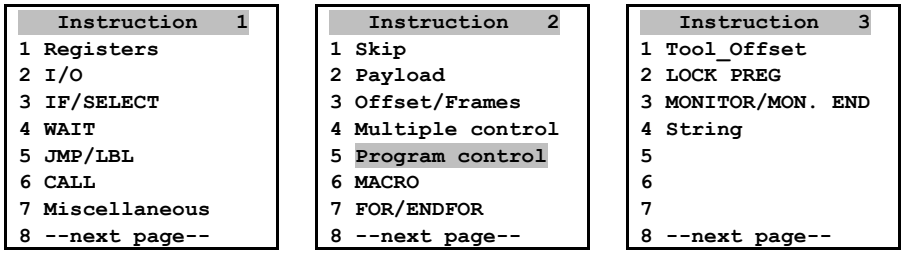
The resume program function executes a resume program defined in an original program, in lieu of the original program. The resume program instructions to define the resume program are the following five instructions. RESUME_PROG[1] can be used in single-task/multi-task, and RESUME_PROG[2-5] can be used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

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

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

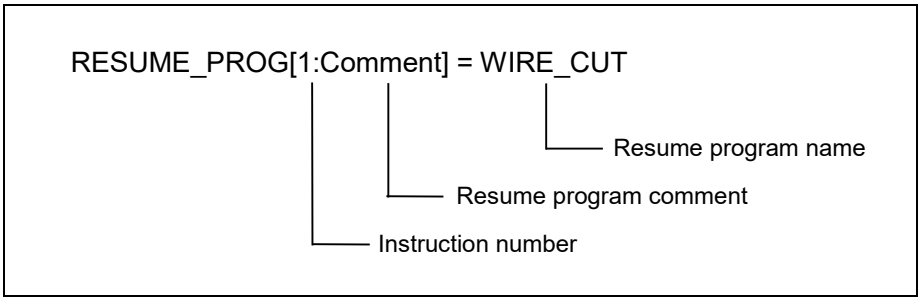
- CLEAR_RESUME_PROG

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



RESUME_PROG[1:Comment]

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



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

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

RESUME_PROG[2]

Dual Arm Control	A05B – 2600 – J605

RESUME_PROG[2-5]

Multi UOP interface function	A05B – 2600 – J964

When specifying the instruction number in [ ] of RESUME_PROG[ ] instruction with multitasking, it is necessary to accord the following conditions.

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

For instance, in multi-task, the RESUME_PROG[3:Comment] can be used only by the program that can be started by START#3/ PROD_STAR#3 of UOP.

The RESUME_PROG[2-5] cannot be used in the single-task because it is for multi-task. Therefore, use the RESUME_PROG[1:Comment] in the single-task.
To define the resume program instructions, there are the following conditions.

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

CLEAR_RESUME_PROG

When the CLEAR_RESUME_PROG instruction is executed, the registration of the resume program registered by the RESUME_PROG [] command is deleted. Therefore, even if an alarm occurs after execution of the 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 defined. For example, in the multitasking system, when the CLEAR_RESUME_PROG instruction is executed in the program in which the RESUME_PROG [1: comment] instruction is defined, only the registration of the resume program registered by the RESUME_PROG[1: comment] instruction is erased, The registration of resume programs registered by other RESUME_PROG[2] - [4] will not be erased.

Apart from the CLEAR_RESUME_PROG instruction, the resume program that was registered also is deleted in the following cases.

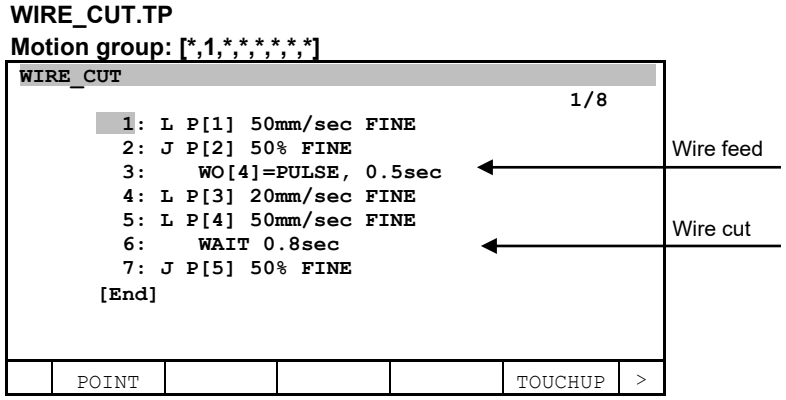
- Backward execution is performed in original program.
- The cursor line is changed manually in original program.
- The original program terminates.

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

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

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   >
```

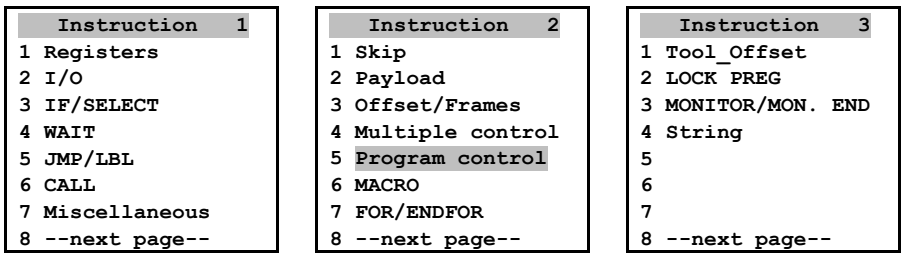


In the above program example, the WIRE_CUT.TP is taught in the second line of the WELD.TP and is erased in the fifth line. Since the WIRE_CUT.TP is defined as the resume program between the third to fourth lines, it is executed as the resume program. In the sixth and subsequent lines, the resume program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

6.5 RETURN_PATH_DSBL INSTRUCTION

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

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



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 multitasking system, if the RETURN_PATH_DSBL instruction is executed in the resume program defined in the RESUME_PROG [1: comment] instruction, only the resume operation function of the original program for which the RESUME_PROG [1: comment] instruction is defined Will be invalid. The resume operation function of the original program for which other RESUME_PROG [2] - [5] instructions are defined remains valid.

The following program is a program example using the RETURN_PATH_DSBL instruction. When defining the RETURN_PATH_DSBL instruction, please define it in the resume program as shown in the following program example. If taught like this, even when the original program is resumed after the

resumption program is terminated, the resume operation function will not work even if the resume operation function itself is set to be effective in 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   <-- Wire feed
      4: L P[3] 20mm/sec FINE
      5: L P[4] 50mm/sec FINE
      6:    WAIT 0.8sec           <-- Wire cut
      7: J P[5] 50% FINE
      8:    RESUME_PATH_DSBL
    [End]
 |  POINT |  |  |  |  TOUCHUP | >
```

6.6 MAINT_PROG INSTRUCTION

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

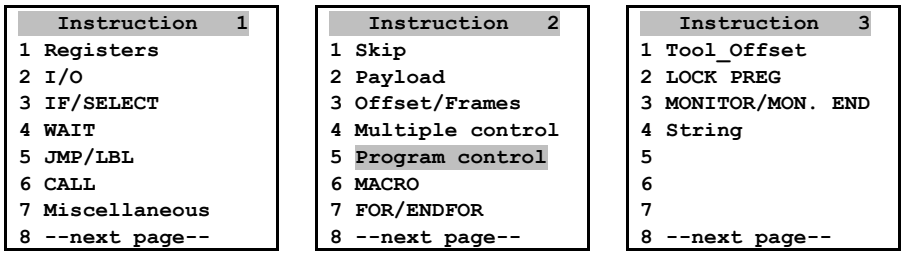
1 From the stopped position, disable arc welding, and execute only the move statements of the original program up to the end.
2 Execute a maintenance program.
3 Disable arc welding, execute the move statements of the original program from the beginning to move the robot to the stopped position.
4 Enable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

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

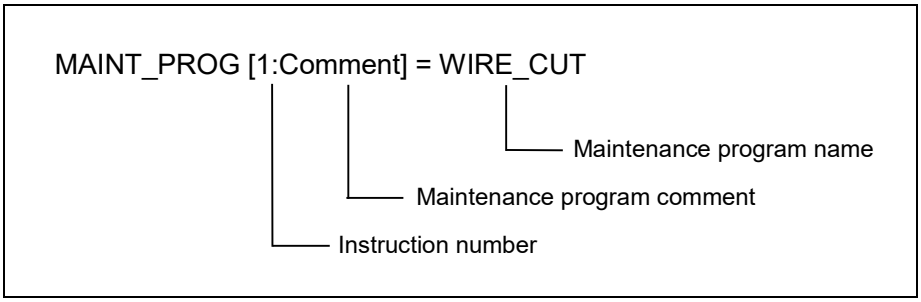
The maintenance program instructions to define the maintenance program are the following five instructions. MAINT_PROG[1] can be used in single-task/multi-task, and MAINT_PROG[2]- [5] can be used in the multi-task.

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

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



MAINT_PROG[1:Comment]



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

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

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

MAINT_PROG[2]

Dual Arm Control	A05B – 2600 – J605

MAINT_PROG[2]-[5]

Multi UOP interface function	A05B – 2600 – J964

When specifying the instruction number in [ ] of MAINT_PROG[ ] instruction with multitasking, it is necessary to accord the following conditions.

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

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

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

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

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

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

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

The maintenance program is erased also when:

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

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

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

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

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

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

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

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

6.7 SETTING OF THE AUTOMATIC ERROR RECOVERY FUNCTION

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

- Selection of instruction number

Error recovery function common setup

- Enable / disable automatic error recovery function
- Registration of recovery switch DI
- Registration of incomplete end DO
- Registration of incomplete end reset DI
- Enable / disable alarm-time automatic start feature
- Enable / disable alarm code monitoring function
- Registration of automatic error recovery alarm conditions

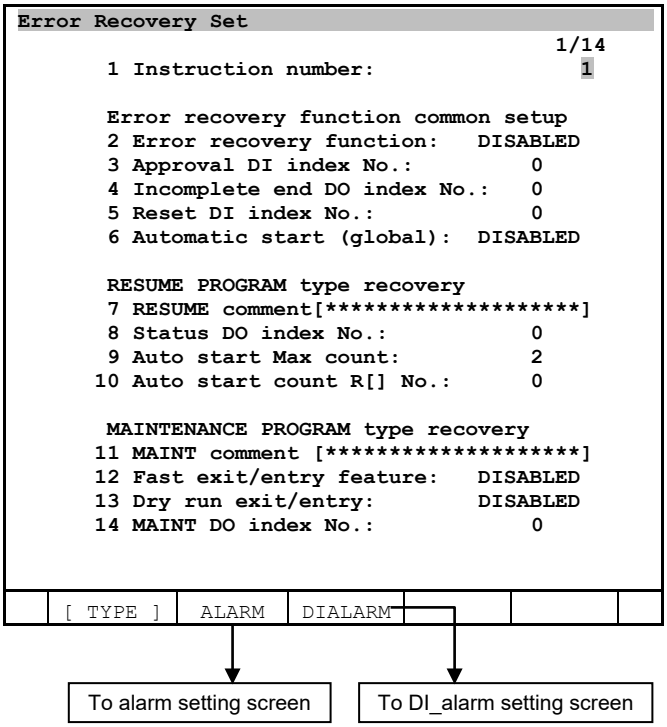
Resume program type recovery

- Registration of resume program comment
- Registration of resume program information DO
- Setting of maximum number of automatic start repetitions
- Registration of automatic start count register

Maintenance program type recovery

- Registration of maintenance program comment
- Enable / disable fast exit/entry feature
- Enable / disable dry run exit/entry operation
- Registration of FFR sequence information DO

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



Selection of instruction number

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

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

Error recovery function common setup

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

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

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

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

Table 6.7 Specifications of the recovery switch DI function

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

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

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

Registration of incomplete end DO
When an incomplete end DO number is defined, the incomplete end DO is output if a certain forced termination alarm is issued during execution of the resume program or the maintenance program. The output incomplete end DO is turned off by the next start signal input.
Before inputting the start signal, the operator must check the incomplete end DO signal status. If this signal is on, the resume program or the maintenance program terminates in the middle, so the robot is not in a specified position.

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

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

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

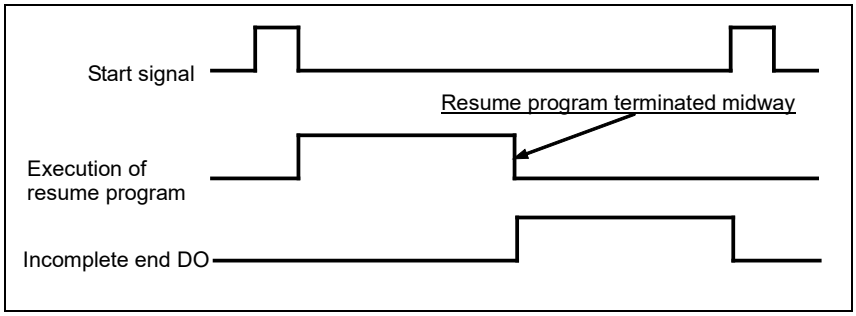


Fig. 6.7 (a) Incomplete end DO output timing chart

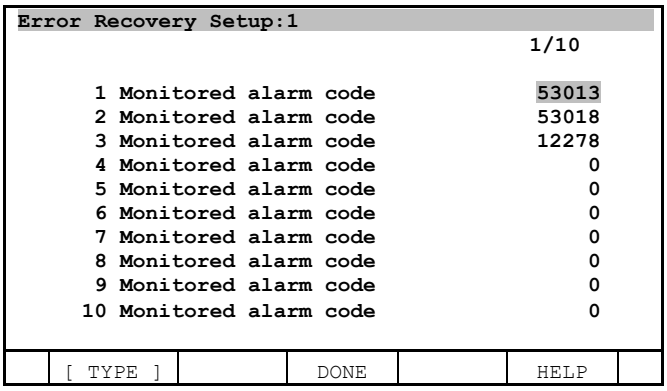
Registration of incomplete end reset DI
When the incomplete end DO is included in the PLC start signal acceptance conditions, the operator requires a means to turn off the incomplete end DO externally. Inputting the incomplete-end reset DI signal turns off the incomplete end DO. When the incomplete end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

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

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

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

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



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

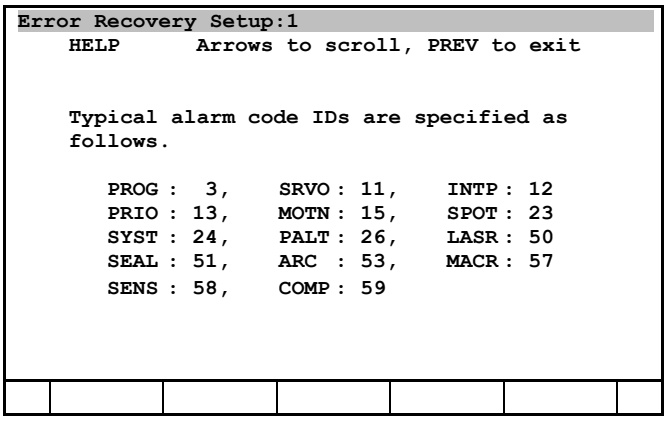
ARC	–	013	Arc Start failed	=	53	013
ID (53)		Number			ID	Number

For alarm numbers, refer to FANUC Robot series OPERATOR'S MANUAL (Alarm Code List) (B-83284EN-1).

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

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

Pressing the F5, HELP key displays the following screen:



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

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

- The resume program or the FFR sequence is executing.

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

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

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

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

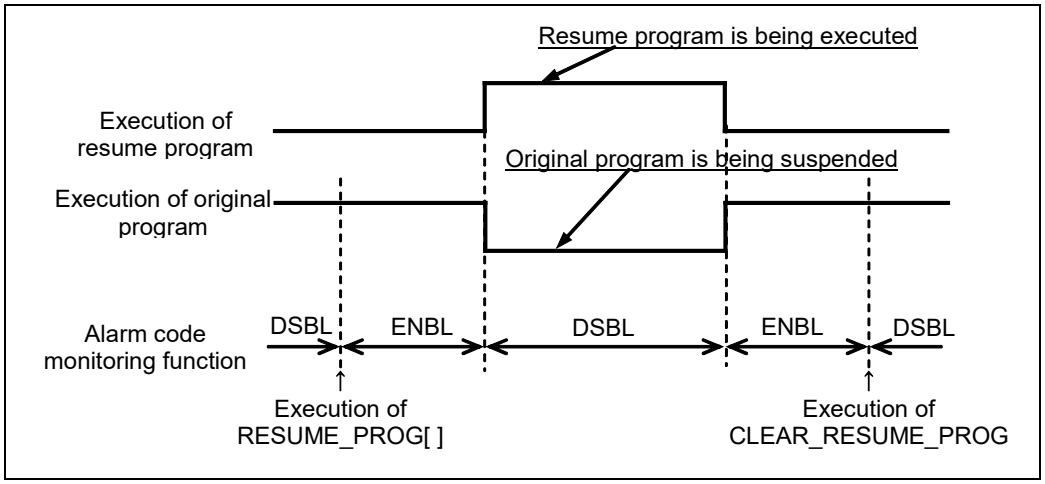


Fig. 6.7 (b) Enabling/disabling the alarm code monitoring function in the resume program function

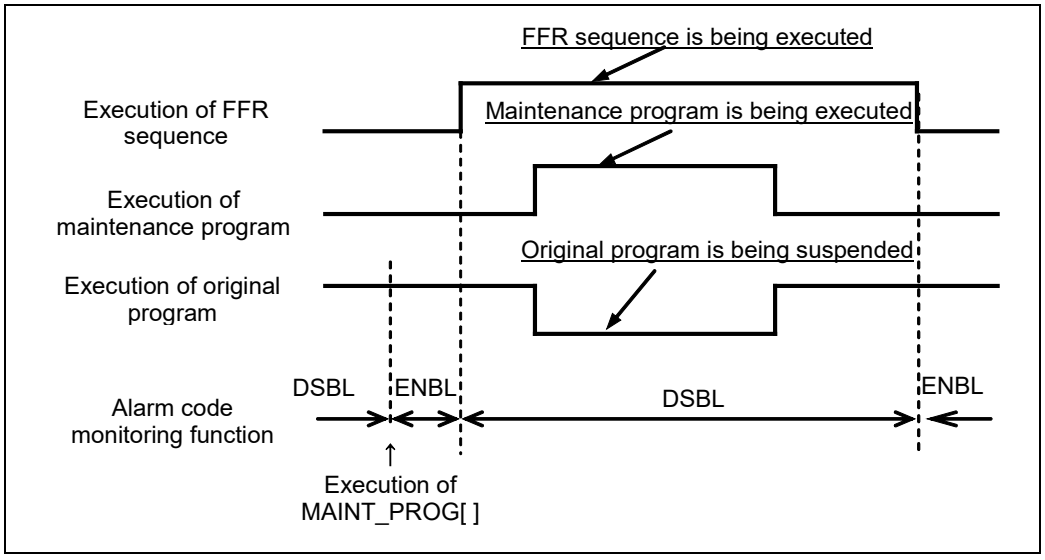


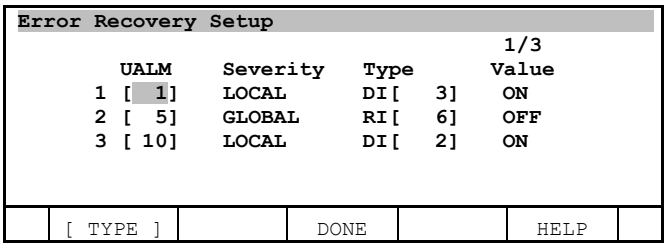
Fig. 6.7 (c) Enabling/disabling the alarm code monitoring function in the fast exit/entry feature

Registration of automatic error recovery alarm conditions

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

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

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



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

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

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

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

Resume program type recovery (Setting items related to restart program function)

Registration of resume program comment

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

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

Registration of resume program information DO

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

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

If the following conditions are met, the resume program information DO goes on:
- The automatic error recovery function is enabled. ("Error recovery function" is "ENABLED" on the automatic error recovery function setting screen.)
- Not the single step mode.
- The resume program is defined in the original program.
- The original program has a motion group.
- The original program is suspended, and the resume program is not yet completed.
- There is no optional function that disables the automatic error recovery function. See “Other specifications and restrictions”.
- The user condition parameter is true. See “Conditions for executing the resume program”.

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

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

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

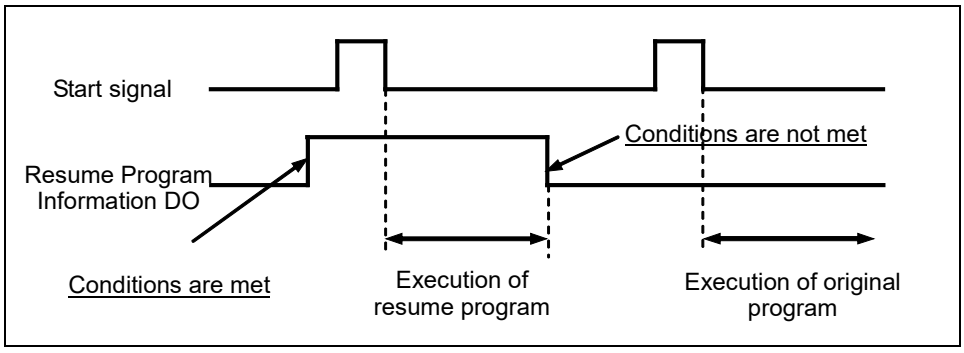


Fig. 6.7 (d) Resume program information DO output timing chart

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

Setting of maximum number of automatic start repetitions

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

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

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

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

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

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

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

Maintenance program type recovery (Setting items related to fast exit/entry function)

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

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

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

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

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

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

Registration of FFR sequence information DO

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

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

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

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

⚠ CAUTION

1 Perform neither backward execution nor the single step operation while the FFR sequence is executed.
2 The update cycle period for the FFR sequence information DO is 300 ms. When the conditions listed above have been changed, wait 300 ms before program execution.
3 When the teach pendant is enabled, the FFR sequence is not executed.

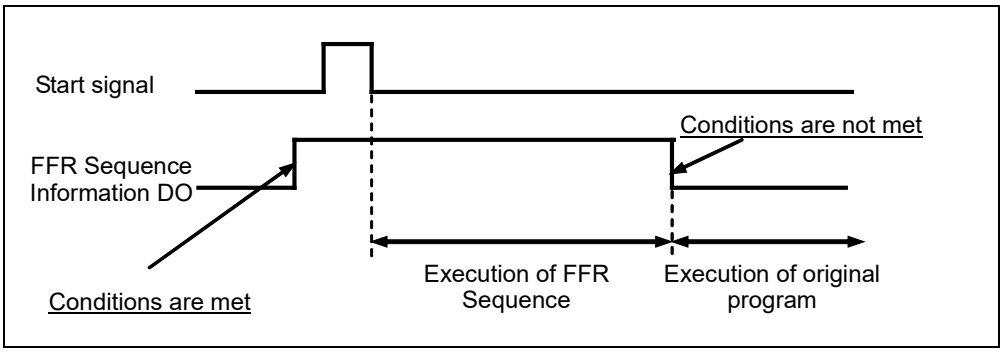


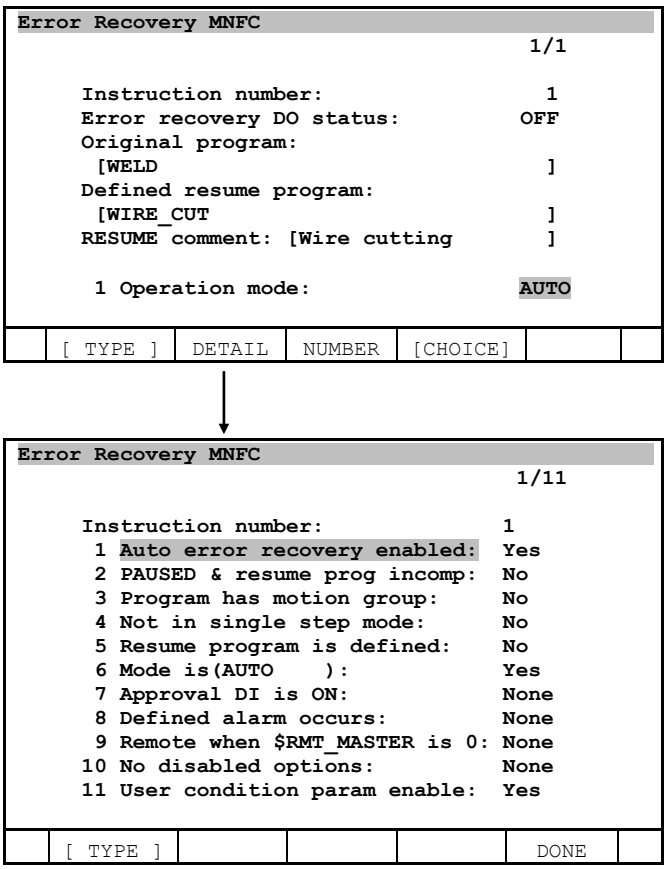
Fig. 6.7 (e) FFR sequence information DO output timing chart

For multi-task

This item can be set by each instruction number.

6.8 MANUAL OPERATION SCREEN OF THE RESUME PROGRAM FUNCTION

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



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

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

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

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

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

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

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

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

Displaying detail conditions of the resume program information DO

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

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

PAUSED & resume prog incomp
This item indicates the following conditions:
- The original program must be in the suspended state.
- A resume program must be defined in the original program, and the execution of the resume program must not have been completed.

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

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

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

Mode is (xxxx)
This item indicates that the operation mode is suitable for the current status.

For example, when the teach pendant is disabled, “AUTO” is indicated in the portion “xxxx.” When the teach pendant is enabled, “TP_TEST” is indicated.

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

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

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

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

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

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

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

6.10 CHANGING CONDITIONS FOR EXECUTING THE RESUME PROGRAM

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

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

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

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

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

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

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

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

The start conditions can be changed by modifying the monitor program. For how to use the status monitoring function, refer to the Status monitoring function section in Utility chapter of operator's manual (Basic Operation) (B-83284EN). In this case, the automatic start function is unavailable.

6.11 OTHER SPECIFICATIONS AND RESTRICTIONS

Automatic error recovery function

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

Resume program function

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

Fast exit/entry feature

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

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

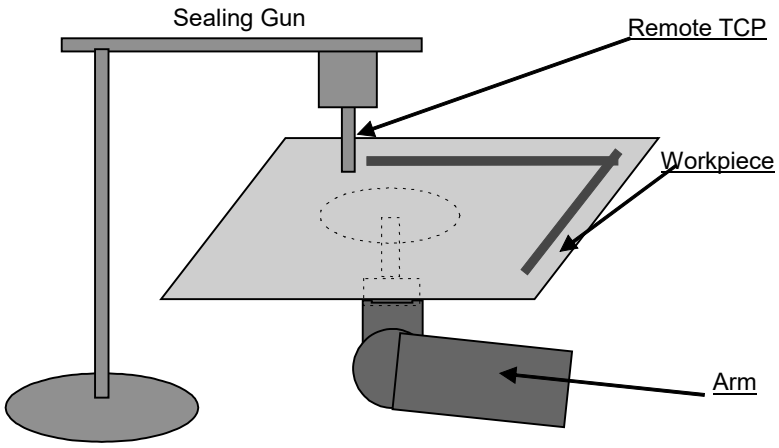
When using the automatic error recovery function, observe the following safety precautions:

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

7 REMOTE TCP FUNCTION

The REMOTE TCP function is used to process to a workpiece that a robot holds and moving against a TCP fixed on the ground (remote TCP).

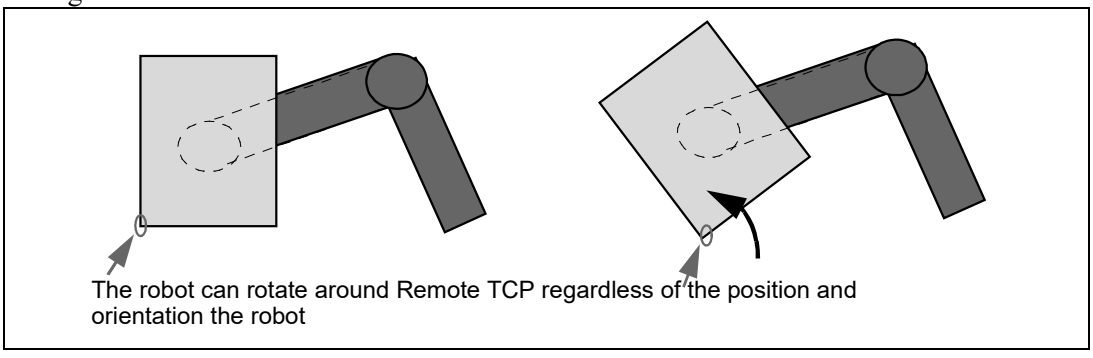
Example of using REMOTE TCP function (by sealing application)



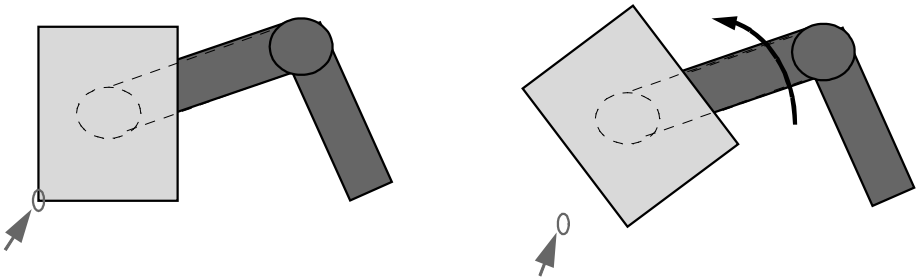
Advantage of Remote TCP function

- Since the gun is fixed on the ground, the cabling of tool is easy.
- It is not necessary that a robot takes the heavy gun.
- Even if you do not use the REMOTE TCP function, you could process to a workpiece that a robot holds and moving against a TCP fixed on the ground, but by using this function, the teaching works become more simply and accurately.

Case of using REMOTE TCP function

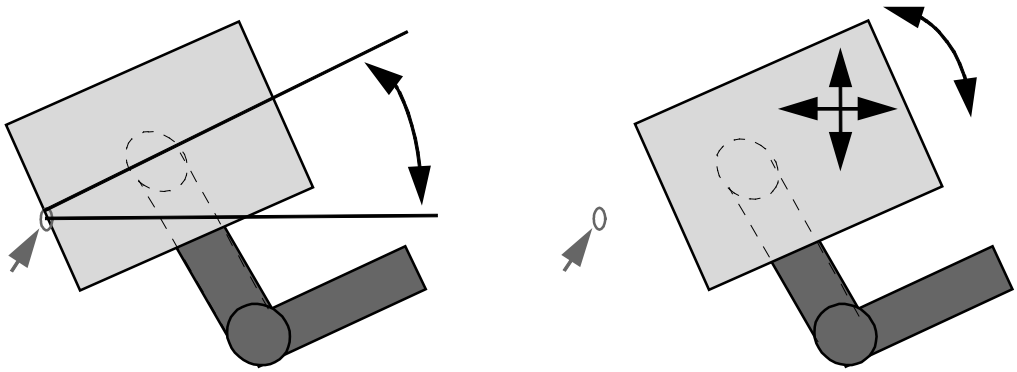


Case of NOT using REMOTE TCP function.



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

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



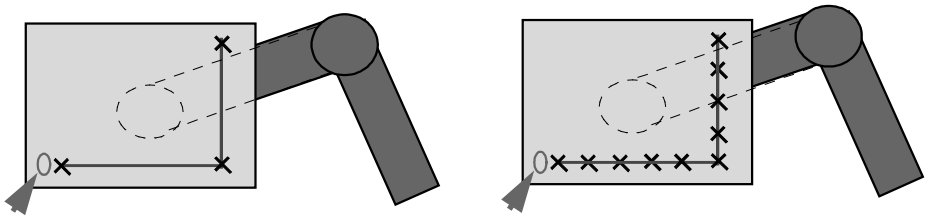
Remote TCP

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

No Remote TCP

Not easy to jog around the Remote TCP.

- You can reduce teaching point drastically.



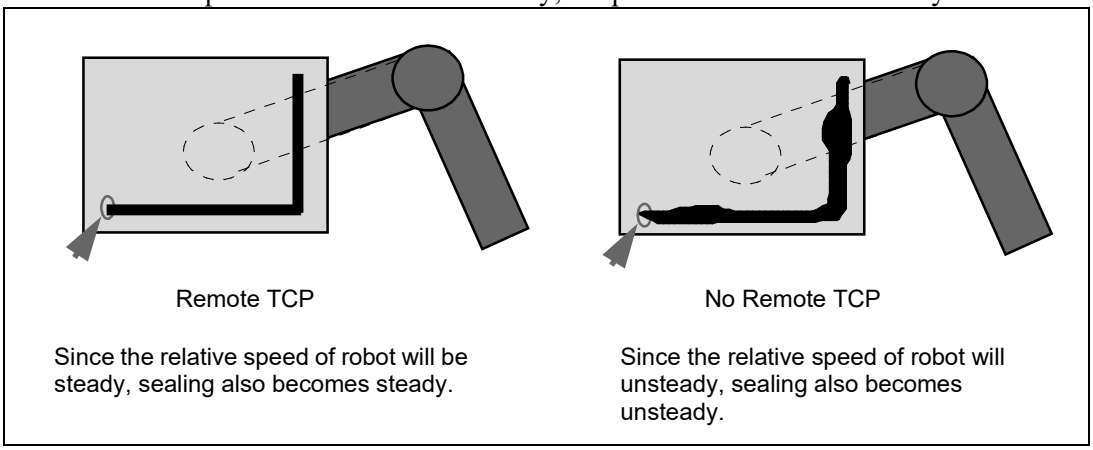
Remote TCP

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.

- Since the relative speed of robot becomes steady, the process also becomes steady as below.



Required option

- Remote TCP (A05B-2600-J624)

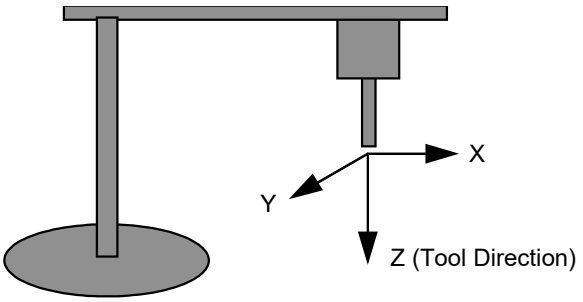
Limitation

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

7.1 SETUP

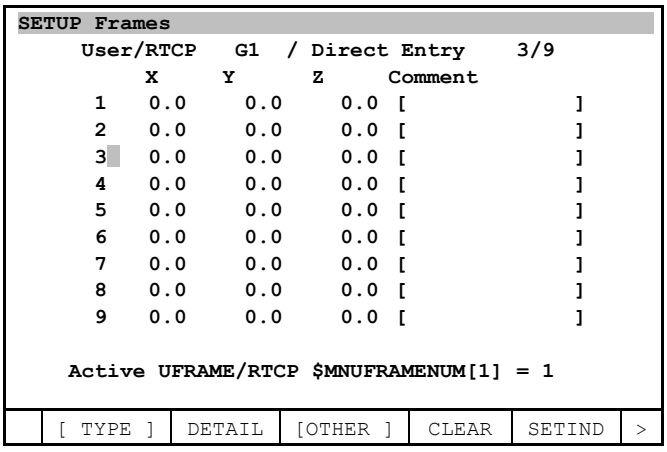
Setup the Remote TCP

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



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

1 Press the [MENU] key 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 the [PREV] key.
4 Move the cursor to the REMOTE TCP frame to use.



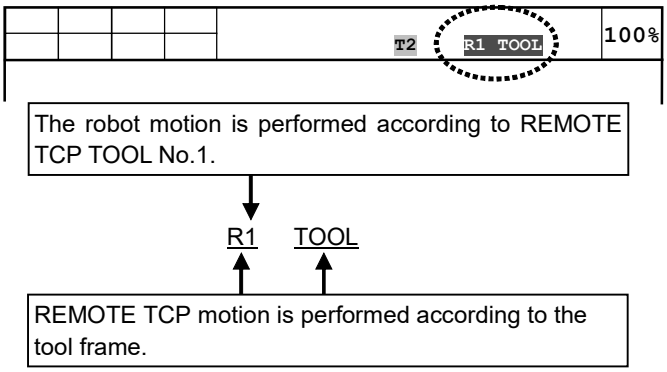
5 Press F2, DETAIL.
6 Select the method from Direct Entry, Three Point or Four Point, and set the frame by using the same method for User frame.

Remote TCP jogging function

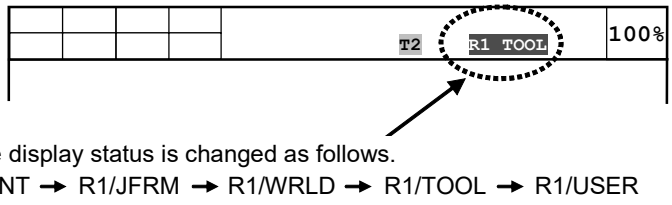
Users jog the robot around remote TCP with REMOTE TCP jogging function.

Users use REMOTE TCP jogging function by the following procedure.

1 Enter to REMOTE TCP jogging mode.
 i. Press the [FCTN] key.
 ii. Select “TOGGLE REMOTE TCP” then the screen is changed as follows.



2 Select the frame from REMOTE TCP USER, REMOTE TCP TOOL, REMOTE TCP JOG by pressed COORD key. Please DO NOT use JOINT.

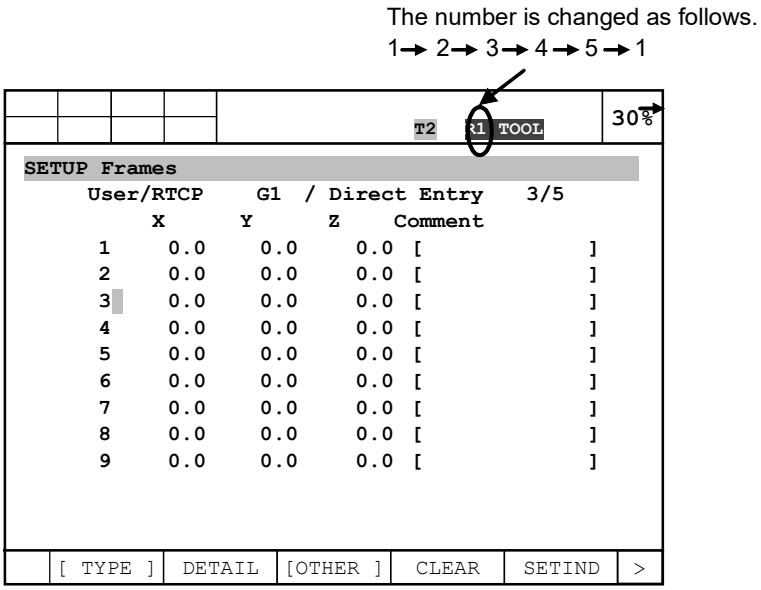


3 Jog the robot.
4 Return from REMOTE TCP jogging mode.
 i.Press the [FCTN] key.
 ii.Select “TOGGLE REMOTE TCP”

Select REMOTE TCP frame

Even if REMOTE TCP mode is selected, you can change the REMOTE TCP frame as below.

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



Programming and running by using REMOTE TCP

If you would like to operate robots on remote TCP instead of robot TCP, you need to specify REMOTE TCP motion option as below.

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

You can not use RTCP with JOINT motion.

<Programming example>

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

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

8 HIGH SENSITIVITY COLLISION DETECTION

High-sensitivity collision detection provides a method to promptly detect that the robot or the end-of-arm tooling has collided with an workpiece etc. and stops the robot.
Detection sensitivity of this function is greatly improved than normal collision detection function and this helps to detect collision more quickly and minimize the potential for damage to the end-of-arm tooling and robot.

8.1 SPECIFICATION

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

8.2 SETTINGS

Load information / Information about Installed Devices (Basic)

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

Collision Guard Setup screen (Optional)

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

Teaching program instructions (Optional)

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

8.3 COLLISION GUARD SETUP SCREEN

Set as below.

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

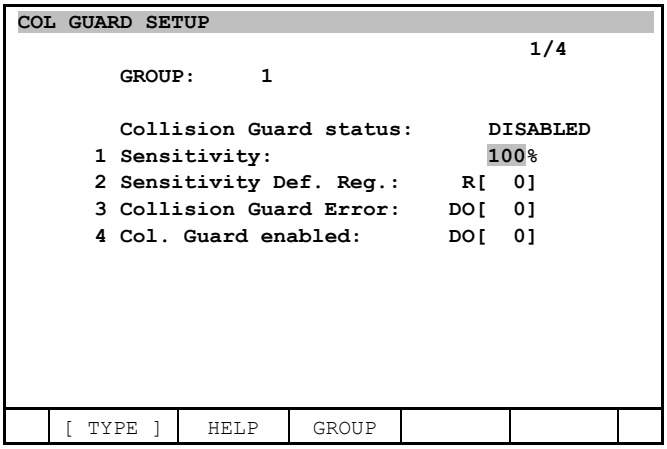


Fig. 8.3 Collision guard setup screen

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

Table 8.3 Setup items of collision guard setup screen

Item	default	Description
Group	1	This item indicates the currently selected group. To 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.
Collision Guard Status	ENABLED	This item specifies whether collision detection is enabled or disabled for the currently selected group. This item cannot be changed in this setup screen.
Sensitivity	100%	This item allows you to set the level of sensitivity for Collision Detection for the currently selected group. Minimum is 1%. Maximum is 200%. The lower the value, the lower the sensitivity. The higher the value, the higher the sensitivity.
Sensitivity Def. Reg.	0	This item allows you to specify the register that can be used with the COL GUARD ADJUST instruction (without arguments) to adjust the sensitivity of Collision Detection within a program. A register number of 0 indicates that COL GUARD ADJUST instruction without arguments is not used.
Collision Guard Error	0	This item allows you to specify a Digital Output that will be turned ON when a Collision Detect Alarm occurs. A value of 0 indicates that no output will be turned on.
Col. Guard enabled	0	This item allows you to specify a Digital Output that will be turned on whenever Collision Detection is enabled via this setup menu, and turned off whenever Collision Detection is disabled via this setup menu. A value of 0 indicates that no output will be used.

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

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

8.4 PROGRAM INSTRUCTIONS

The program instructions below are reserved.
- COL DETECT ON
- COL DETECT OFF
- COL GUARD ADJUST

8.4.1 COL DETECT ON / COL DETECT OFF

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

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

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

Example
```
10: J P[1] 100% FINE
11:  COL DETECT OFF
12: L P[2] 2000mm/sec CNT100
13: L P[3] 2000mm/sec CNT100
14: L P[4] 2000mm/sec CNT100
15:  COL DETECT ON
16: J P[5] 50% FINE
```
This program disables collision detection with lines 12 to 14.

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

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

8.4.2 COL GUARD ADJUST

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

This instruction can be used in three ways depending on how to specify arguments. (See Table 8.4.2.)

Table 8.4.2 COL GUARD ADJUST instruction

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

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

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

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

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

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

- Set the register(s) specified in the COLL GUARD SETUP menu for the appropriate group(s) to the new sensitivity value(s).
- Insert the COL GUARD 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
If the register number in the COL GUARD SETUP menu is zero for one of the groups being adjusted, then an error will be posted as follows.

MOTN-400 No Coll. Guard Reg. Defined

NOTE
If the data in specified register is not an integer between 1 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 COL GUARD SETUP menu.

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

In this program, the collision detection sensitivity becomes 80% in lines 13 to 15, and is 100% in line 18.

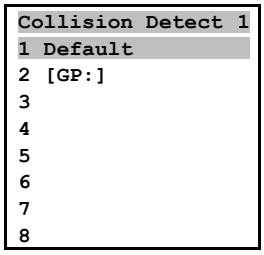
Example 2

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

In this program, the collision detection sensitivity becomes 80% in lines 12 to 14, and is 100% in line 17.
To use the COL GUARD 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 GUARD ADJUST instruction will apply to all groups in the group mask of the TP program that called it. However, if the instruction is used with an argument, the user also has the option to restrict the groups whose sensitivity will be adjusted. This is accomplished by moving the cursor to the

argument filed and pressing the F1, GP_MASK key. Doing so will display the following menu at the top of the Teach Pendant screen:



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 GUARD ADJUST GP1,2: ##

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

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

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

where "g" refers to the group number.

8.5 CAUTIONS

1 Collision Detection might detect a false collision when a collision has not occurred in the following cases:
- 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 using the ACC motion option.
- 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 misdetection occurred for the reasons above, try removing these reasons first. If necessarily, insert COL GUARD ON / OFF or COL GUARD ADJUST instructions at the top and bottom of where misdetection occurs for avoiding alarm stop.

2 Collision Detection is disabled in the following cases:
- Soft Float is enabled.
- The robot brakes are on (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.

9 LOAD ESTIMATION

Load estimation is a function for estimating the weight of the load, such as tool and workpiece, mounted on the hand of the robot.
The function enables the information stated above to be estimated automatically by running the robot.

To use this function, High Sensitive Collision Detection Package option or package option which includes it (ex. Motion Package, Basic Interference Check, Intelligent Interference Check, ARC easy smart quick recovery, Lincoln Asia-Pacific Arc package for Standard EQ, Lincoln Asia-Pacific Arc package) is required. Using the function also requires that your model support the load estimation function. If your model does not support the function, you cannot use it.

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

9.1 OPERATING PROCEDURE

Load is estimated in the following flow:
1 Set the range of motion to be subjected to load estimation.
2 Execute load estimation.

Once a mechanical part such as a motor is replaced, it becomes necessary to make calibration.
If no calibration is made after mechanical part replacement, the precision of load estimation becomes lower.

> ⚠ **CAUTION**
> While in execution, load estimation moves J5 and J6 axis with maximum speed (100% joint motion). Do not enter the operating space of the robot. Also, please make sure that all the devices attached to the concerned parts can withstand such motion.
> Some models support additional type of estimation motion that moves J3, J5 and J6 axis simultaneously.

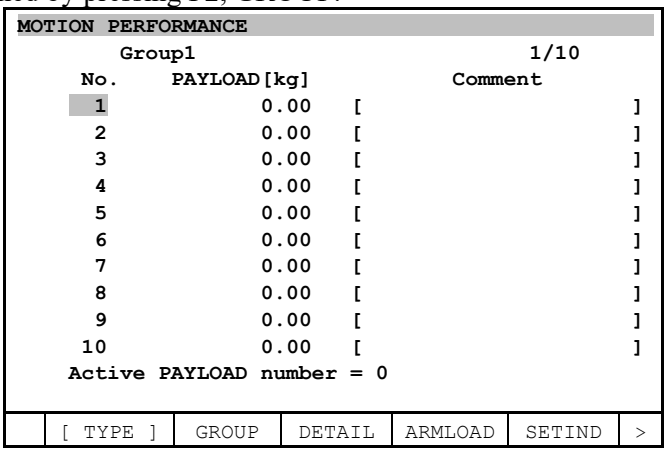
9.2 LOAD ESTIMATION PROCEDURE (for 6-Axis Robots)

There are three types of estimation. The types of estimation supported depend on the robot model.
The first type of estimation, or “2 axes shuttle”, which is the default selection, moves J5 and J6 between two points. It moves in low speed, then repeats in high speed.
The second type of estimation, or “Estimation by 3 motions”, adds J5 only motion to the first type. This may be able to estimate payload with the Center of Gravity near J6 axis of rotation, which cannot be estimated with the first type. It moves the robot to 4 different points. It includes low speed J5 and J6 motion (same as normal estimation), high speed J5 only motion, and high speed J5 and J6 motion (same as normal estimation).
The third type of estimation, or “3 axes waves”, moves J3, J5 and J6 simultaneously. This may be able to improve estimation for payloads of unknown mass.
Though their estimation motion setup is different, they are all executed with the same process.

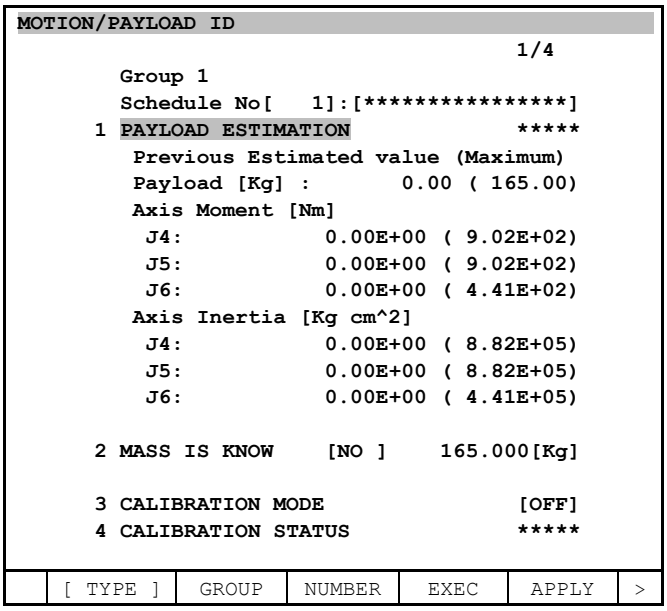
9.2.1 Preparation

All three types of estimation is performed on the load estimation screen. This screen is accessed from the motion performance screen.
1 Press the [MENU] key to display the screen menu.

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



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



6 Place the robot in the position where load estimation is to be performed.
With the default type and “Estimation by 3 motion”, J1 to J4 will stay in this position during the estimation. With “3 axes waves” motion pattern, J1 and J2 will stay in this position.

NOTE
1 Set the J3 arm to be horizontal. The more vertical it is, the lower the precision of the estimation becomes.
2 Put the J5 rotation axis in a horizontal position.
The more vertical posture the J5 rotation axis takes, the lower the precision of estimation becomes.

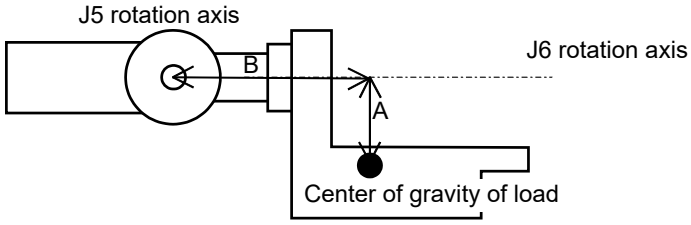
7 Press F3, NUMBER, and select the load setting schedule No. for which a load estimate is to be set up.
8 If the mass of the load for which load estimation is to be performed is known, move the cursor to line 2, select “YES”, and specify (enter) the mass.

NOTE

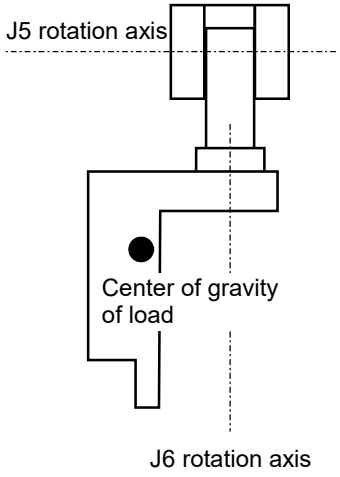
A load with very small mass cannot be estimated. As a rough guide, it requires the load with 5% of maximum load capacity or more.

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

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

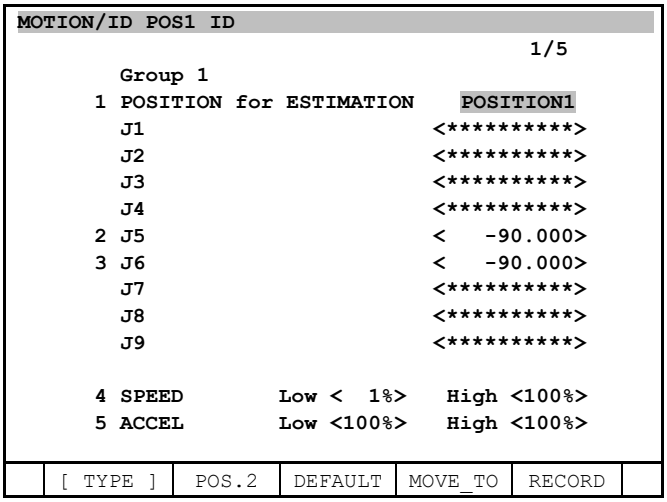


- As for positions set up on estimation position 1 and 2 screens, the center of gravity of the load must be in or near the plane that contains the J5 and J6 rotation axes.



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

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



9.2.2 Estimation Motion Setup

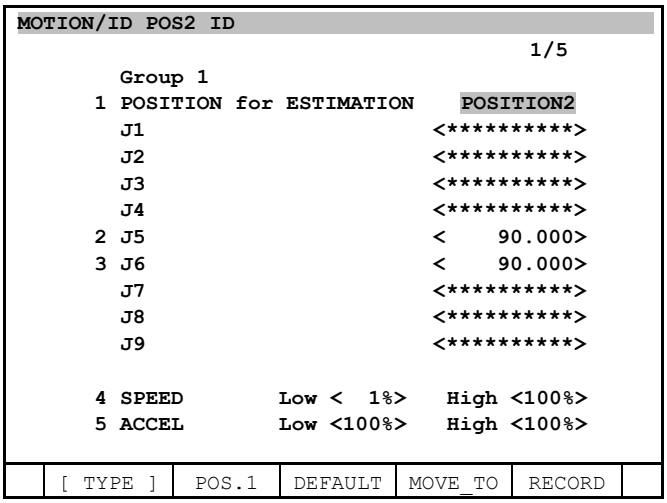
This subsection describes estimation motion setup. This setup differs among three types of estimation. Please read the corresponding part of the subsection to setup.

Estimation Motion Setup (2 axes shuttle)

The following procedures describe estimation motion setup for the default type, 2 axes shuttle estimation.

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

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



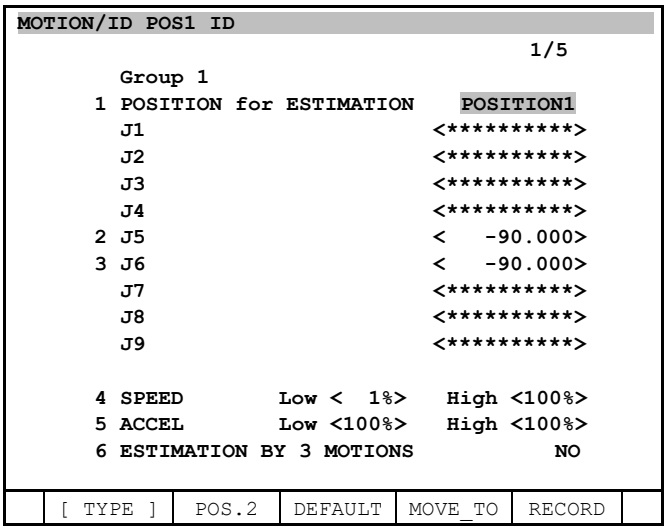
3 Specify estimation position 2. (Alternatively, the initial value can be used.)
Specify the positions of the J5 and J6 axes by entering their values directly. Alternatively, move the robot to the desired position by jogging, then press the [SHIFT] key + F5, RECORD to record the position.

Now pressing the [SHIFT] key + F4, MOVE_TO moves the robot to estimation position 2. Use this procedure to identify the set position.

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

Estimation Motion Setup (Estimation by 3 motions)

The following procedures describe estimation motion setup for "Estimation by 3 motions".
With the model and software versions that support this type of estimation (7DF1/04 and later), there would be a line "ESTIMATION BY 3MOTIONS" added at the end of the position 1 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, the initial value can be used.)
Specify the positions of the J5 and J6 axes by entering their values directly. Alternatively, move the robot to the desired position by jogging, then press 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 estimation position 2 screen.
3 Specify estimation position 2 in the same way as 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 estimation position 1 screen, and press the [SHIFT] key + F4, MOVE_TO to move the robot to position 1.
5 Move the cursor to the line "ESTIMATION BY 3 MOTIONS" and press F4 [YES], and see that the line display changes to YES.
6 Press the [F2] key twice to display position 3 screen. F2 normally switches between [POS.1] and [POS.2], but when "ESTIMATION BY 3 MOTIONS" is "YES", it will switch through [POS.1], [POS.2], [POS.3], and [POS.4]. The position 3 will be the start and end of J5 only motion.
7 Specify the estimation position 3 in the same way as position 1 or 2.
8 Press the [SHIFT] key and F4 [MOVE_TO] to move the robot to the position 3, and confirm that it will not interfere with the environment.
9 Press the F2 [POS. 4] and display position 4 screen.
10 Specify the estimation position 4. The position 4 will be the turn-round of J5 only motion. J6 position will automatically be same as that of position 3 and will not be changeable.
11 Press the [SHIFT] key and F4 [MOVE_TO] to move the robot to the position 4, and confirm that it will not interfere with the environment.
12 Press the [PREV] key to return to the estimation screen.

NOTE

In the “ESTIMATION BY 3 MOTIONS”, the robot will move in the following way. The 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).
Please make sure that it is safe for the robot to move that way using SHIFT key and F4 [MOVE_TO].

The calibration, to be described later, will run the same way as the 2 axes shuttle regardless of the choice at “ESTIMATION BY 3 MOTIONS”.

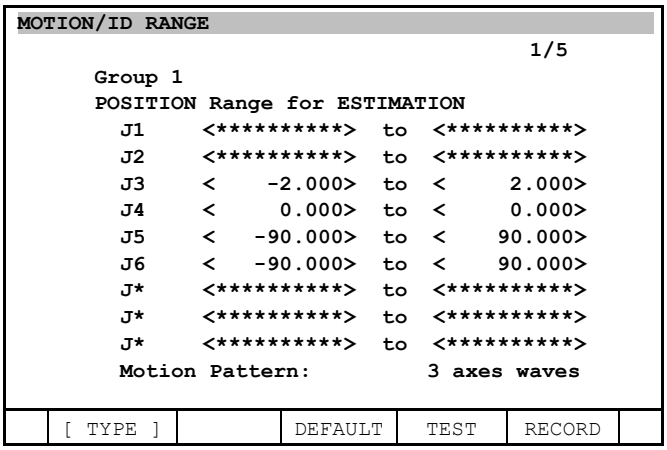
Estimation Motion Setup (3 axes waves)

The following procedures describe estimation motion setup for “3 axes waves” motion pattern. This type of estimation requires that the ARMLOAD setting is correctly done including the Center of Gravity.

1 In the position 1 screen, there is an item for Motion Pattern, which should show 2 axes shuttle. Move the cursor to axes and select 3 axes with F4 [3 axes] key. The Motion/ID RANGE screen will be displayed.
If the item is not shown, the software does not support this type of estimation. If the item is not selectable, the model does not support this type of estimation.

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

      4 SPEED        Low <  1%>   High <100%>
      5 ACCEL        Low <100%>   High <100%>
      6 Motion Pattern:       2 axes shuttle

|  | [ TYPE ] | POS.2 | DEFAULT | 3 axes | 2 axes |  |
```

2 Specify the motion range for estimation. The axes with *s will stay at the position where the estimation is executed. The unselectable axes are fixed to that position. There are limits to the width of the range. Some axes have fixed width, while others can have different widths within certain values. Press F3 [DEFAULT] to set default ranges. Press F5 [RECORD] to set current position of the selected axis to the selected side of the range.

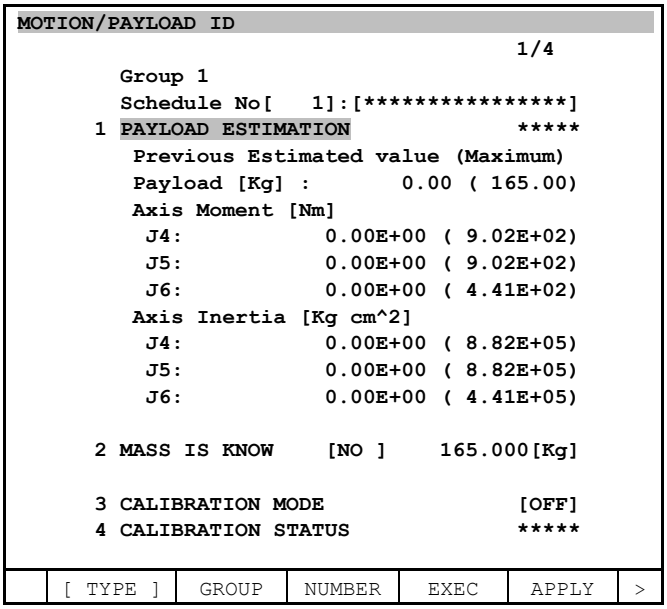


3 Press F4 [TEST] with SHIFT key to test the estimation motion with range set. The robot will move with the current override. Changing override after starting the TEST will not change the speed of the motion.
4 Press the [PREV] key to return to the estimation screen.

NOTE
1 J3, J5 and J6 axes move during the estimation. J4 moves at the beginning of the estimation.
2 This type of estimation requires that ARMLOAD setting, including the Center of Gravity, is done correctly. Please make sure ARMLOAD is set correctly before executing the estimation. (The Center of Gravity setting of ARMLOAD is added to the software versions that support this type of estimation.)
Make sure to cycle power after changing ARMLOAD setting.
3 Set the range so that the center of the range is 0 degree. The farther the center, the lower the accuracy of estimation becomes.
4 There are minimum widths of the ranges; setting one side of the range may change the other to have minimum width.
5 When the range is too wide, it will fail to create estimation motion. Please execute TEST to confirm.

9.2.3 Estimation Execution and Applying the Result

With estimation motion setup and tested, the estimation may be executed from the estimation screen.



1 Set the teach pendant ON/OFF switch (the teach pendant enable switch) to OFF, and press F4, EXEC. The message “Robot moves and estimates. Ready?” will be displayed .
2 Specify whether to execute load estimation. (Selecting “YES” allows the robot to move. Pay sufficient care to avoid danger.)
   - To perform load estimation by running the robot, press F4, YES.
   - To quit execution, press F5, NO.
3 After estimation motions are finished, load information is estimated. When the estimation is complete, “*****” of “1 PAYLOAD ESTIMATION *****” changes to “DONE”. (Operation switches automatically from low speed to high speed. Even when the robot is running at low speed, do not get close to it, because otherwise you may get in a dangerous situation when the robot suddenly starts running at high speed.)
4 Press F5, APPLY to set the estimate at a load setting schedule No. The message “Path and Cycle time will change. Set it?” will be displayed.
5 Specify whether to set the estimate result.
   - To set the estimate, press F4, YES.
   - Not to set 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 Emergency Stop button or HOLD key, or terminated by CSTOPI or ABORT from 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 NOTE at the subsection “Preparation”, to see if the load fulfills the conditions necessary to estimate.

9.2.4 Estimation Result

The estimation result is shown in the estimation screen as Previous Estimated value. With software versions 7DF1/28 (V9.10P/28) or 7DF3/06 (V9.30P/06) and later, there is additional screen to show the estimation result in percentage to the maximum allowed. When the software supports the screen, there would be RESULT key.

1 Press the [NEXT] key, then press F5, RESULT.
The estimation result will be displayed with its percentage to the maximum allowed.

```
MOTION/ID RESULT
                                          1/3
   Estimation Result
      Group  1
      Mass [kg]                0.00 (  0.0%)
      Axis Moment [Nm]
        J4:                0.00E+00 (  0.0%)
        J5:                0.00E+00 (  0.0%)
        J6:                0.00E+00 (  0.0%)
      Axis Inertia [kg cm^2]
        J4:                0.00E+00 (  0.0%)
        J5:                0.00E+00 (  0.0%)
        J6:                0.00E+00 (  0.0%)

[ TYPE ]          GRAPH
```

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

9.3 CALIBRATION PROCEDURE (for 6-Axis Robots)

Once a mechanical part such as a motor is replaced, it becomes necessary to make calibration.
If no calibration is made after mechanical part replacement, the precision of load estimation becomes lower.
Calibration is controlled, using the load estimation screen. Calibration is started by setting the calibration switch to ON and executing load estimation.
2 axes shuttle and Estimation by 3 motions share the calibration data. Regardless of the setting for “Estimation by 3 motions”, the calibration motion would be that of 2 axes shuttle. The calibration for 3 axes waves nedds to be done separately.

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

NOTE
If calibration is performed with anything attached to the robot hand, incorrect calibration data is set up, thus hampering a normal estimation.
In this case, make calibration again, properly this time.

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

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

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

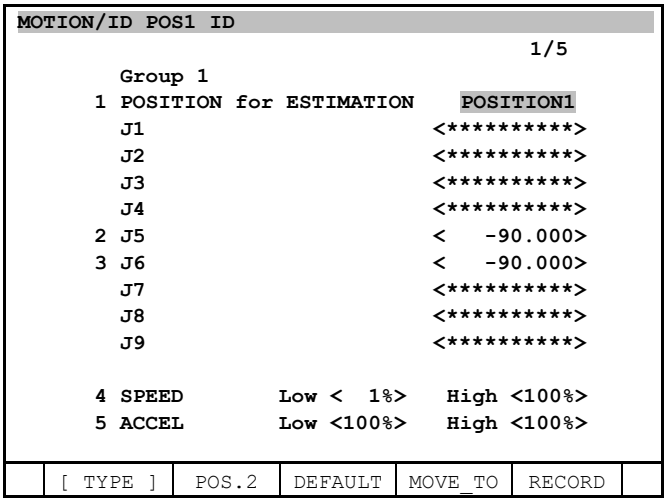
6 Press the [NEXT] key, then F2, IDENT. The load estimation screen will be displayed.

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

    2 MASS IS KNOW    [NO ]    165.000[Kg]

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

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

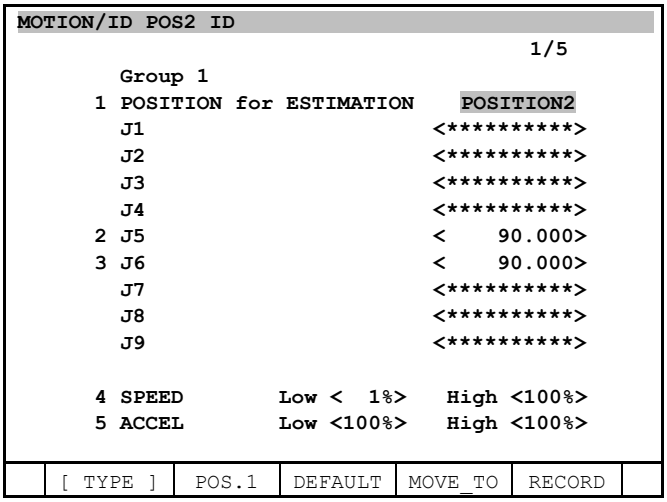
7 Place the robot in the position where load estimation is to be performed.

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

8 Press the [NEXT] key, then F4, DETAIL. The estimation position 1 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 Pressing the [SHIFT] key + the F4, MOVE_TO moves the robot to estimation position 1. Make sure that it is safe to move the robot to estimation position 1.
If it is dangerous to move the robot to estimation position 1, manipulate the J1 to J4 axes by jogging to move the robot to a position where the robot can move safely.
11 Pressing the F2, POS.2 displays the estimation position 2 screen.



12 Pressing the [SHIFT] key + F4, MOVE_TO moves the robot to estimation position 2. Make sure that it is safe to move the robot to estimation position 2.

If it is dangerous to move the robot to estimation position 2, manipulate the J1 to J4 axes by jogging to move the robot to a position where the robot can move safely.
If you moved any of the J1 to J4 axes, press F2, POS.1 to go back to the estimation position 1 screen, and follow this procedure again from step 10.

13 Press the [PREV] key to return to the load estimation screen.
14 Move the cursor to CALIBRATION MODE on line 3 to turn it "on".
With software versions 7DF1/03 and later, or 7DC3/33 and later, the message will confirm whether to "Change to calibration mode?" Select "YES."
Then, it will say "Please execute without any load on the wrist." Select "OK."
Selecting "NO" or "CANCEL" will change CALIBRATION MODE back to "off."

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

15 Move the cursor to line 4 (so that "EXEC" will be displayed at F4), and set the teach pendant ON/OFF switch (the teach pendant enable switch) to OFF, then press "EXEC". The message "Robot moves and estimates. Ready?" will be displayed .
16 Specify whether to perform load estimation. (Selecting "YES" causes the robot to move. Pay sufficient care to avoid danger.)
- To perform load estimation by running the robot, press the F4, YES.
- To quit execution, press the F5, NO.

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

The calibration procedure for 3 axes waves is basically the same as description above. It needs to set the Motion Pattern to "3 axes waves" and set the motion range instead of positions. Make sure that no payload is on the hand or the arm of the robot, armload setting is correctly set to 0, and the controller is power cycled after the setting.

9.4 OTHER RELATED MATTERS

(1) Motion range

If the motion range between estimation positions 1 and 2 becomes narrower, the estimation precision may get lower.
The actual motion range should preferably be as wide as the default motion range.

(2) Acceleration for motion used in load estimation

The estimation precision is low for the load whose moment inertia is relatively low compared with the maximum allowable load of the robot. This is because the influence by the moment inertia to the torque of the robot motor is weak.
The estimation precision for this light load may be able to be increased by increasing the acceleration used during operation for load estimation.
Try to increase the acceleration by specifying a larger value in "ACCEL - High" on the estimation position 1 and 2 screens; however, do not specify so large a value that vibration becomes serious during operation.

(3) Calibration data

The following system variable holds calibration data.

```
SPLCL_GRP[group].$TRQ_MGN[axis] (for default and "Estimation by 3 motions")
                or $TQ_MGN3A[axis] (for "3 axes waves")
```

```
group    :       Group number
axis     :       Axis number
```

If improper calibration data is set up, for example, by making calibration with a load mounted by mistake, reassigning the previous data to the system variable can restore the previous calibration data.
It is recommended to keep note of the previous calibration data so as to enable restoration.

(4) Payload setting for estimating motion

If payload setting data while in estimating motion is far different from the actual payload, the estimating motion may cause oscillation. To stabilize the estimating motion, it is recommended to set up the payload data by rough calculation before executing estimation.
Setting rough payload data in advance may help enhancing the precision of the estimation, especially for the payload which does not have sufficient moment around the J5 and J6 axes.

(5) Collision Detection for estimating motion

When payload settings are different from the actual payload, it may cause false detection of collision because the motion may be vibratory and the difference from the internal model may be large. When false detection occurs, please set roughly estimated payload data, as in (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

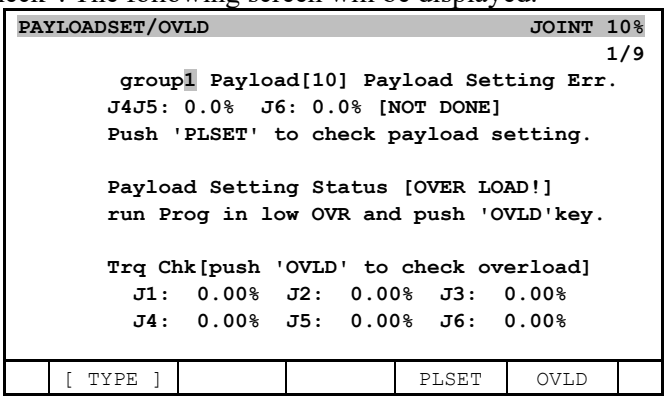
This function supports robot with 6 rotary axes robots.
(In supported type, $PLID_GRP.$ROB_TYPE has to be 1 or 2 and $SCR_GRP.$NUM_ROB_AXS has to be 6.)

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

10.2 PROCEDURE

10.2.1 Procedure to Show Payload Over and Payload Setting Confirm Screen

1 Press 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.



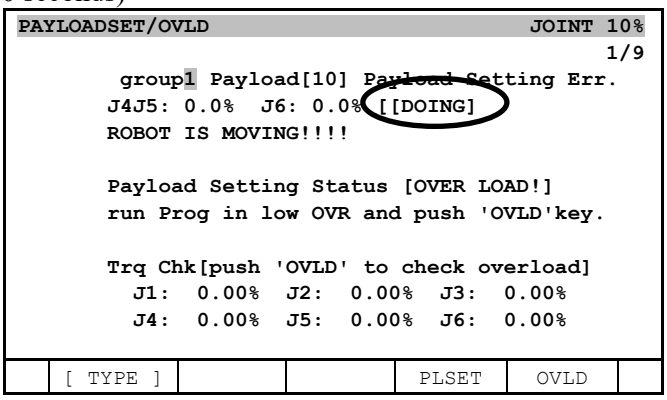
10.2.2 Procedure to Use Payload Setting Confirm (When Payload is Changed)

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.

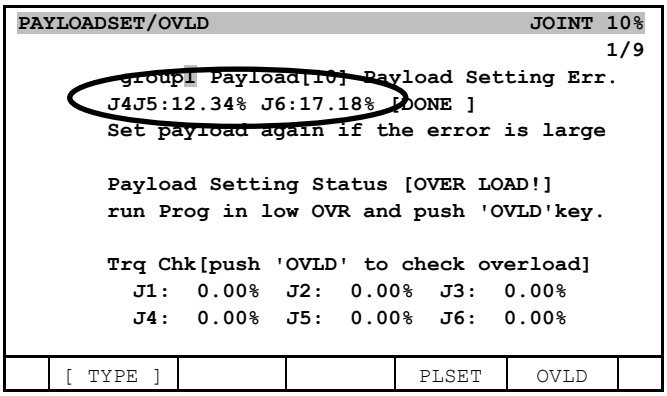
NOTE
Confirm that robot does not interfere if J5 and J6 axes move between current position+1[degree] and current position–1[degree].

3 Turn off the teach pendant enable 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)



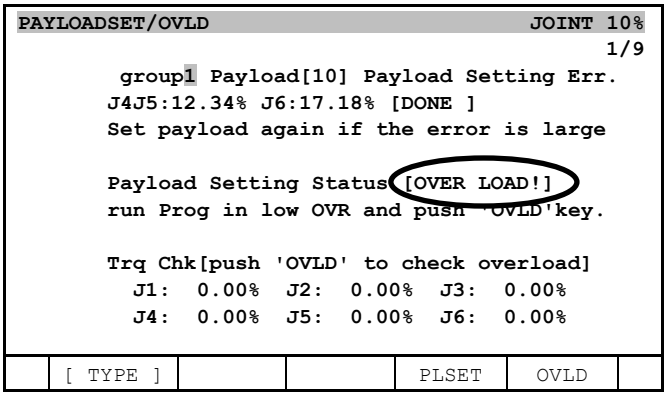
8 From “Payload Setting Err.”, you can confirm whether a payload setting is done correctly. Threshold to decide that a payload setting is done correctly is 10%.



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

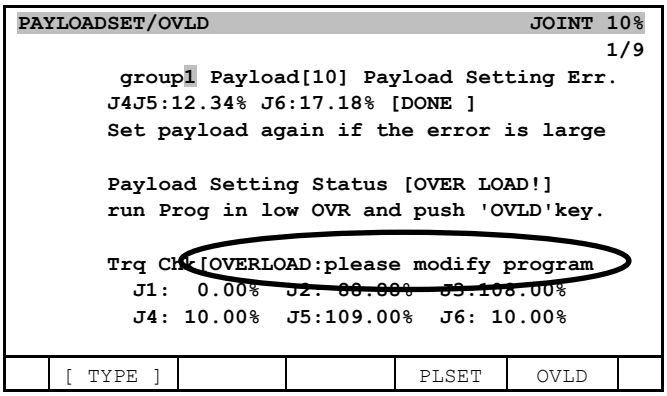
10.2.3 Procedure to Use Payload Over Confirm (When The Program or Payload is Changed)

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

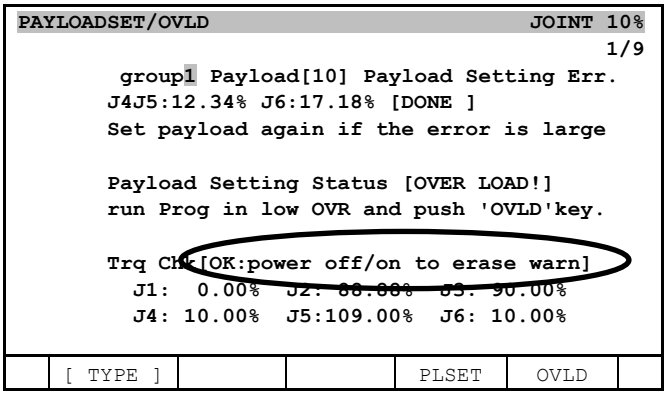


2 The program to be payload over confirmed has to be executed beforehand. Please select the program to be payload over confirmed. (Push [select] key, select program and press the [ENTER] key.)

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 calculates the bending of the robot arm caused by the tool/work on the flange, the equipment on the arm, and the self weight of the arm. Then it compensates the motor position depending on the calculation of the bending, and it improves the absolute position accuracy.

To use this function, Gravity Compensation option (A05B-2600-J649) is required.

11.1 SYSTEM VARIABLES

Gravity Compensation

$PARAM_GROUP[group].$SV_DMY_LNK[8]

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

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

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

$PARAM_GROUP[group].$MOUNT_ANGLE

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

11.2 MOTION SCREEN

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

MOTION Screen (Default screen)

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

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

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

PAYLOAD SET Screen

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

 |  | [ TYPE ] | GROUP | NUMBER | DEFAULT | HELP |  |
```

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

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

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

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

ARMLOAD SET Screen

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

 |  | [ TYPE ] | GROUP |  | DEFAULT | HELP |  |
```

9 Setup the armload on axis J1 and axis J3.
After the value is input, the message "Path and Cycle time will change. Set it ?" is displayed. Please input F4, YES or F5, NO.
After setting up the armload, cycle power.

11.3 MASTERING

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

11.3.1 "Normal Mastering" and "GC Mastering"

You have 2 types of mastering method.

(1) Normal Mastering : Mastering with Gravity Compensation DISABLED

(2) GC Mastering : Mastering with Gravity Compensation ENABLED

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

Mastering method at the shipment of the robot

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

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

11.3.2 How to Choose Mastering Method

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

$DMR_GRP[group].$GRAV_MAST

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

This system variable is updated every time you execute mastering.

11.3.3 Mastering Procedure

Mastering procedure for Normal Mastering

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

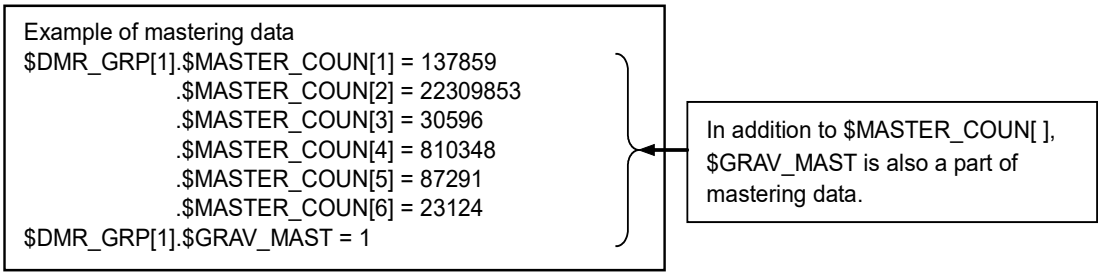
Mastering procedure for GC Mastering

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

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

NOTE

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



11.3.4 Guidance of GC Mastering

Keep the following rules of GC Mastering to get the correct mastering data.

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

3 When you perform Single Axis Master, keep the following rules;
- Just to be safe, write down the Mastering Data ($DMR_GRP[group].$MASTER_COUN[axis]) of all axes before you perform mastering.
- If there are more than one axes which need to be mastered, you must select and master them all at once.
  ex) Single Axis Master after J2, J4 and J5 motor replacement
  You must master J2, J4 and J5 all at once.
  You cannot do master solo J2 when J4 and J5 remain to be mastered.
  * If you have no way to master multiple axes all at once, work around in the following way;
  (NOTE: Master in ascending order. In other words, from robot base to flange)
  (1) First, master J2, J4 and J5 all at once. You need to master J2 correctly. At this time, you don't need to master J4 and J5 correctly, but master them moderately correctly. (In other words, do temporary master for J4 and J5.)
  (2) Next, master J4 correctly.
  (3) Finally, master J5 correctly.
- If you move no-need-master axes to the angle which you have wrote down for a reference position in advance, do it after performing temporary mastering of need-master axes.
  ex) Single Axis Master using reference angle of J1,J2,J4,J5,J6 after J3 motor replacement
  (1) First, master J3 moderately correctly, then do CALIBRATE.
  (2) Next, move no-need-master axes (J1,J2,J4,J5,J6) to the angle which you have wrote down for a reference position.
  (3) Finally, master J3 correctly.
- If you perform "mastering after temporary mastering" as described above, master count of the axis which has an interaction with the target axis of mastering will change. This causes unexpected position change. Therefore, after performing mastering, verify that master count of no-need-master axis keeps the original value. If this value has been changed, enter the original value which you have written down in advance.

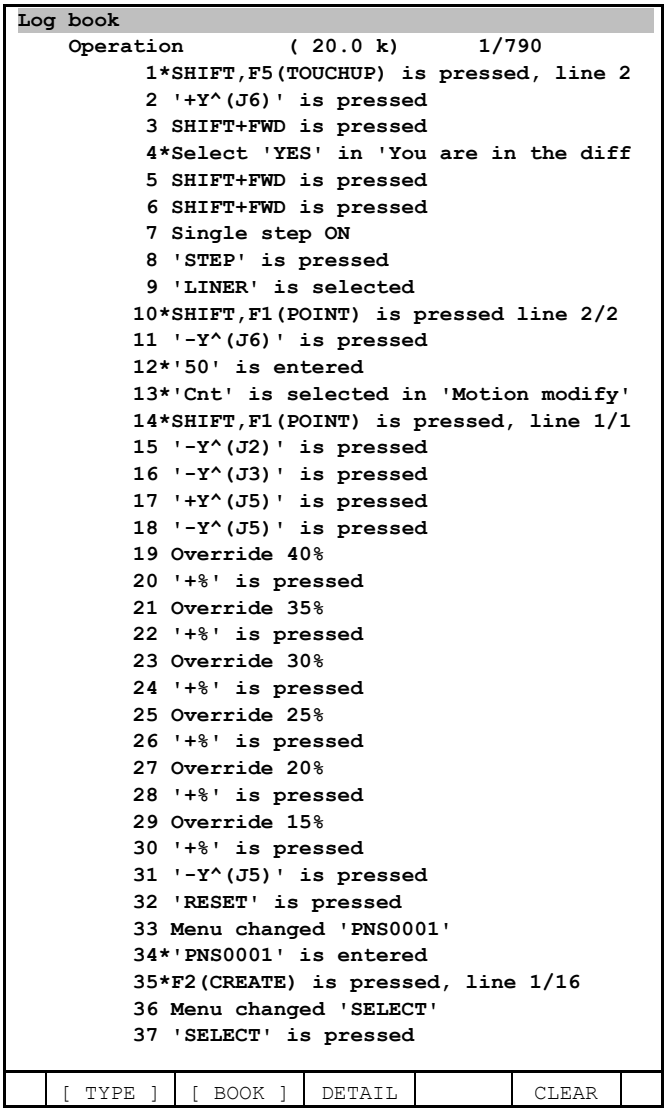
12 OPERATION LOG BOOK

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

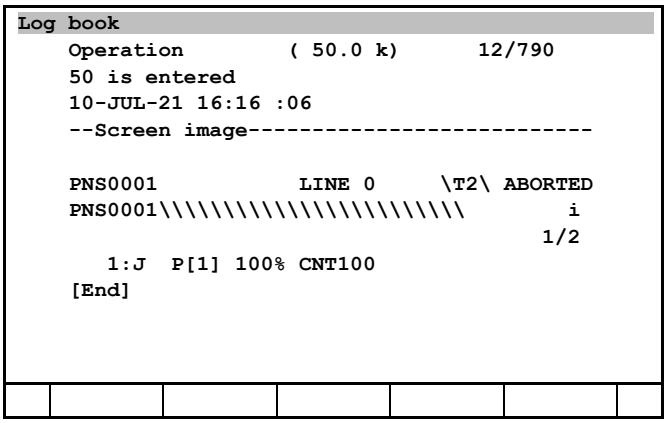
To use function, Operation Log Book option (J695) is required.

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

See the following example of a Log Book.



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.



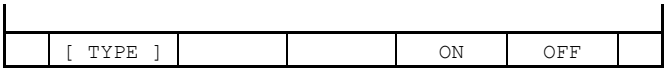
General Limitations

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 $LOGBOOK.$LOG_ENT is a valid book number when you log in, your password input is recorded as “’x’ is entered”. Passwords for other functions are also recorded. To prevent passwords from being recorded, you can use screen filtering to filter out the screens that contain passwords. Refer to Section 12.4 Operations.
- ‘y’ of “’x’ is selected in ‘y’ window” is based on the 1st line of prompt window. If the 1st line is blank, y contains nothing. Please refer to screen image for analysis.
- If you press a function key that has no label, the key number followed by empty parentheses will be recorded.

  Example

  If the function key line is as follows:



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

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

```
0.0 Amps
0.0 Amps
```

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

12.1 RECORDED EVENTS

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

Refer to "12.3 OPERATIONS" later in this section.

Table 12.1 Events recorded by log book

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

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

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

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

12.2 SETTING UP BOOK

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

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

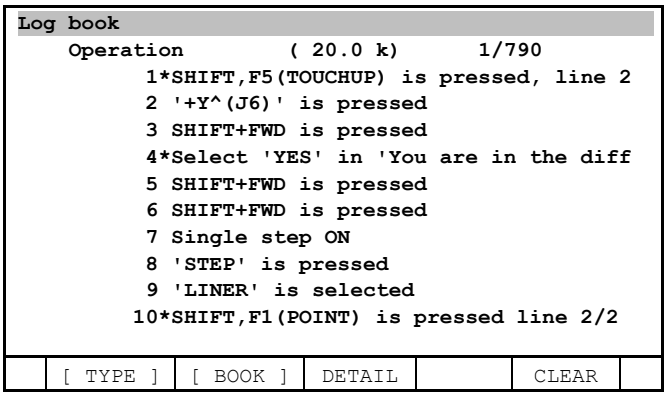
Table 12.2 System variables to setup book

System variable name	Type	Default	Range	Description
$LOG_BUFF[1-16]. $TITLE	String	[1-16] ‘’	Max 13 chars	Title of every buffer. This string is displayed on the top line of LogBook menu. It also be displayed in pop up menu of F2([BOOK]) in LogBook menu. If the value is ‘’, the default title is displayed. Default title of BOOK1 is ‘Operation’. Default title of ‘BOOK2 is ‘I/O’. Default title of BOOK3 is ‘BOOK 3’.
$LOG_BUFF[1-16]. $SIZE	Integer	[1] 50 [2] 100 [3-16] 0	0, 2147483647	Size of the buffer to save log. Unit is K Bytes. One record takes about 300 bytes. If this value is too big, maximum available size of Log Book buffer is allocated automatically. If there is not enough memory to make Log book buffer, SYST – 188 WARN “book(%d) was not created” is posted and the book is not created. Cycle power is needed to enable the change of this variable.
$LOG_BUFF[1-16]. $MEM_TYPE	Integer	[1] 0 [2] 1 [3-16] 0	0, 2147483647	Memory type of every buffer. 0: In SRAM, Log is kept at power down. 1: In DRAM, Log is cleared at power down.
$LOG_BUFF[1-16]. $VISIBLE	Boolean	[1] TRUE [2] FALSE [3-16] TRUE	FALSE, TRUE	If FALSE, the buffer is not displayed in pop up menu of F2([BOOK]) in LogBook menu.

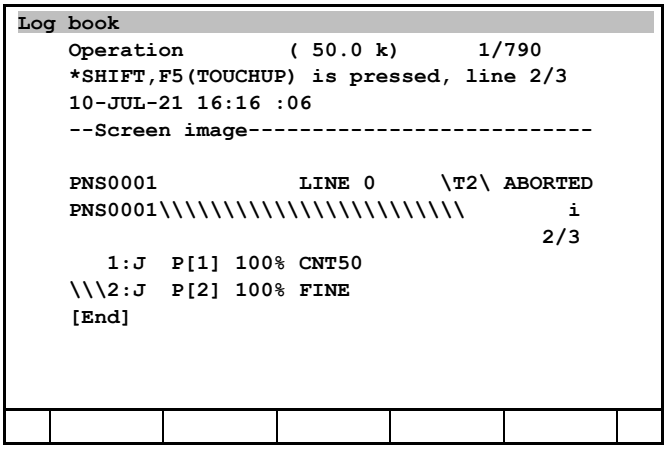
12.3 OPERATIONS

Displaying the Log Book Screen

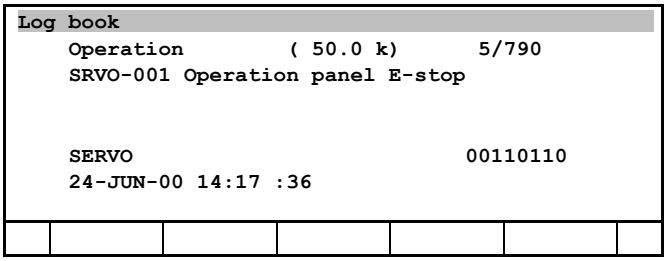
1 Press the [MENU] key.
2 Select 4, ALARM.
3 Press F1, [TYPE].
4 Select Log Book. The following screen will be displayed.



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 system variables to filter specific alarms and screens.

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

Table 12.3 (a) System variables used for filtering

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

Control error filtering by setting the following system variables.
- $LOG_ER_ITM[1-n]
- $LOG_ER_SEV
- $LOG_ER_TYP[1-n]

The priority of the settings is: $LOG_ER_SEV < $LOG_ER_TYP < $LOG_ER_ITM

See the following for an example of error filtering:

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

Table 12.3 (b) Alarm filtering example

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

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

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

NOTE
Supplementary explanation TP'x' ON/OFF is not supported.

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

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

Table 12.3 (c) System variables for screen filtering

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

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

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

12.4 EXTENDED ALARM LOG

12.4.1 Setup

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

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

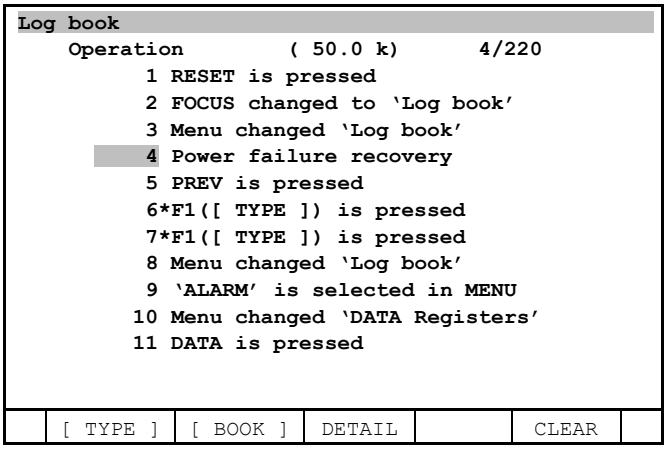
Table 12.4.1 System variables

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

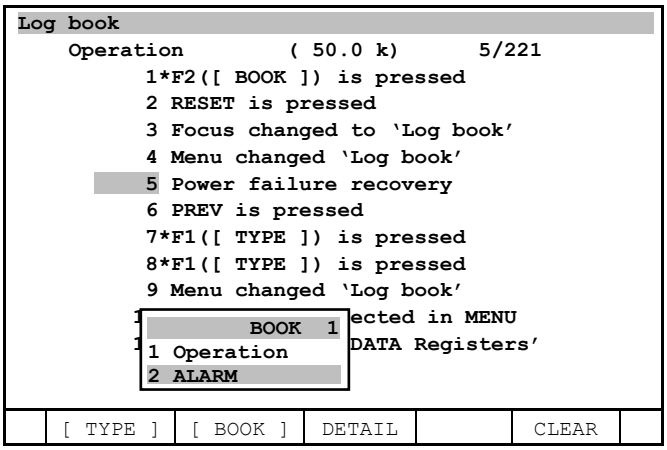
Cycle power is needed to enable these changes.

12.4.2 How to Display Alarm Log

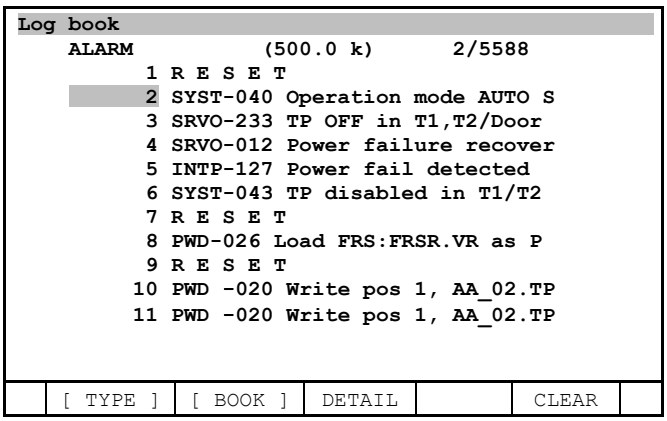
1) Press the [MENU] key.
Select ALARM.
Press the F1 and select "Log book". Logbook screen will be displayed.



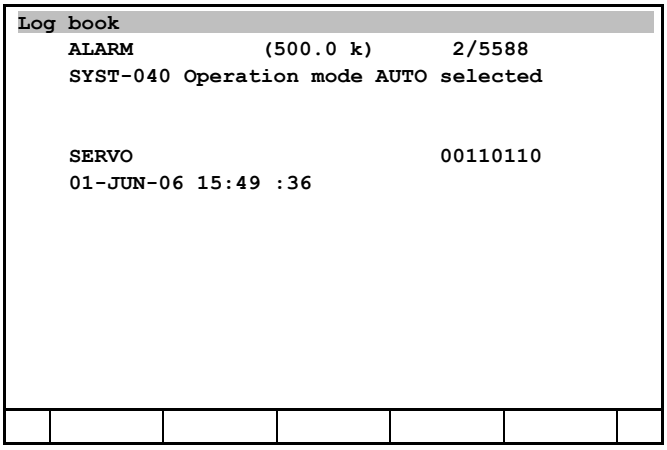
2) Press the F2 and select "ALARM".



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.

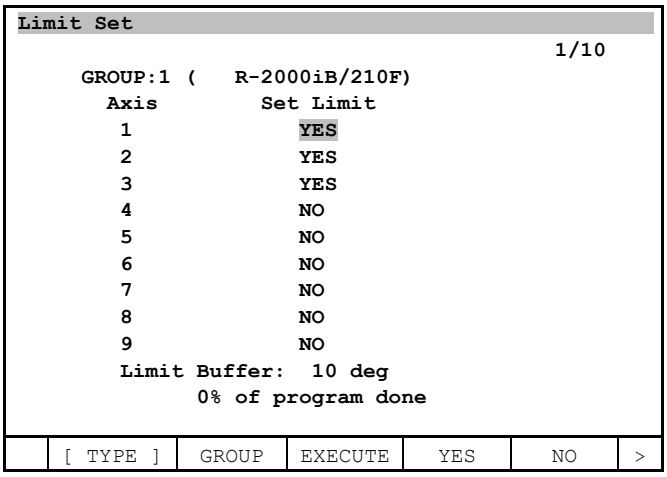
Table 13.1 Toolbox soft limit setup menu items

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.

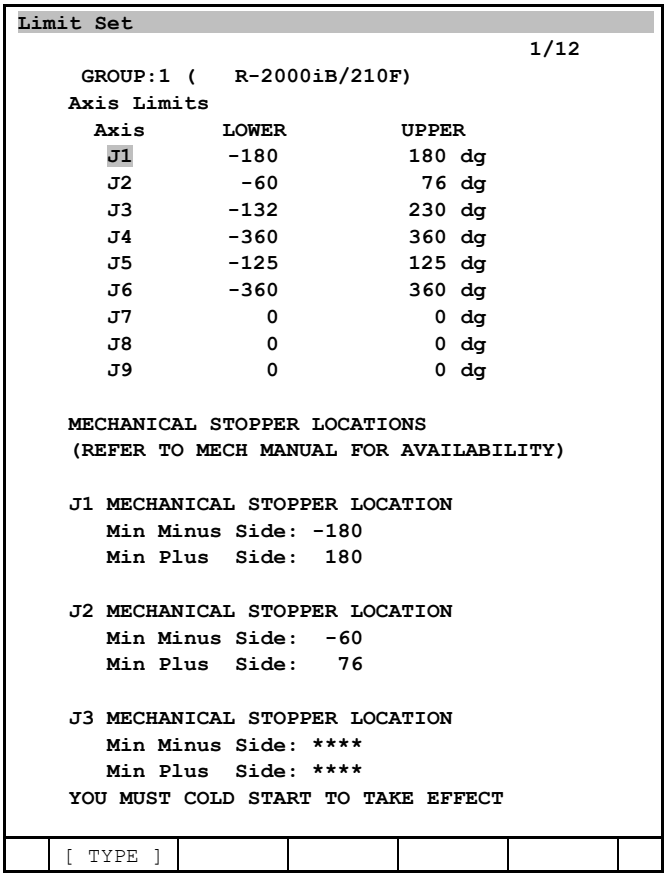
Procedure 13-1 Using the soft limit setting option to set up software axis limits

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

ADV-CP Path CTRL allows you to use Linear Distance and Corner Region for adjusting the path. ADV-Max Speed CTRL allows you to use Process Speed and Max Speed for adjusting the speed of a Robot.

ADV-CP Path CTRL (R806)
- Linear Distance
- Corner Region

ADV-Max Speed CTRL (R805)
- Process Speed
- Max Speed

It requires that the Constant Path function be enabled.

14.1 LINEAR DISTANCE

Linear Distance is a motion feature that is useful for material handling applications such as palletizing. Fig.14.1(a) shows a typical pick and place application. If a robot starts at P1 and goes through P2 and picks up a part at P3 and then goes through P2 to P4 and then to P5 where it places the part and all termination types are FINE or CNT0, then the path would be as shown.

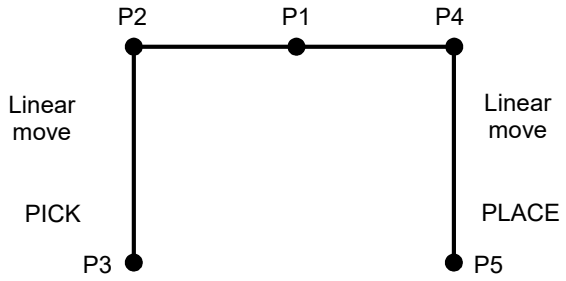
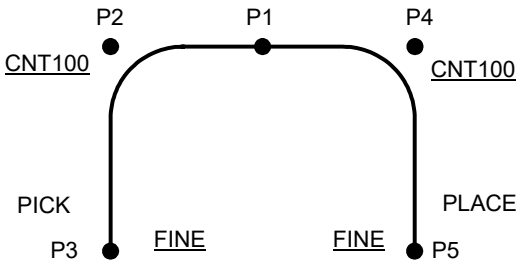


Fig. 14.1 (a) Path with FINE or CNT0

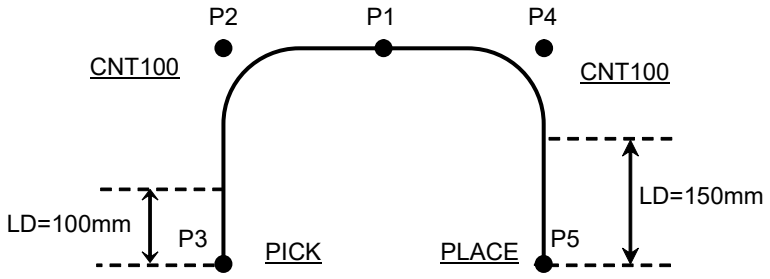
Typically, however, FINE and CNT0 is used only to reach P3 for PICK and to PLACE at P5. For all other motions, high CNT values are used. For instance, if CNT100 is used, the actual path might look like Fig.14.1(b).




```
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 can not specify the linear part of the path from P2 to P3 or from P4 to P5 directly. When you want to adjust the linear part of the path, you need to adjust values of CNT.
With Linear Distance, you do not have to guess and experiment. If you want the last 100mm before pick and the last 150mm above place to always be straight above P3 and P5 respectively, you can use Linear Distance for specifying these amounts. Refer to Fig. 14.1 (c).




```
The program would be (from P1)
J  P[2] 100% CNT100
L  P[3] 2000mm/sec FINE AP_LD100
L  P[2] 2000mm/sec CNT100 RT_LD100
L  P[4] 2000mm/sec CNT100
L  P[5] 2000mm/sec FINE AP_LD150
```

Fig. 14.1 (c) Path with Linear Distance

14.1.1 How to Use

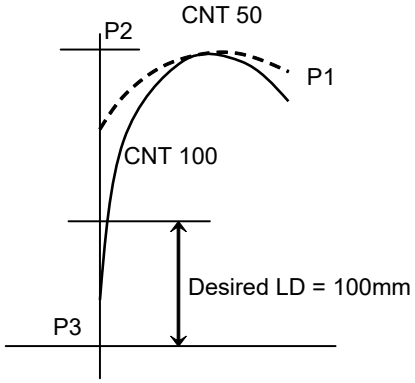
Linear Distance allows you to use two additional motion instructions:

RT_LD: Retract Linear Distance is used for specifying the desired linear distance from the start position to the start of the corner.
AP_LD: Approach Linear Distance is used for specifying the desired linear distance from the end of the corner to the destination position.

WITHOUT USING LINEAR DISTANCE
Robot is at P3

•Dashed line example :
L P[2] 2000mm/sec CNT50
J P[1] 100% CNT100

•Solid line example :
L P[2] 2000mm/sec CNT100
J P[1] 100% CNT100



USE OF LINEAR DISTANCE
Robot is at P3

•Dashed line example :
L P[2] 2000mm/sec CNT50 RT_LD100
J P[1] 100% CNT100

•Solid line example :
L P[2] 2000mm/sec CNT100 RT_LD100
J P[1] 100% CNT100

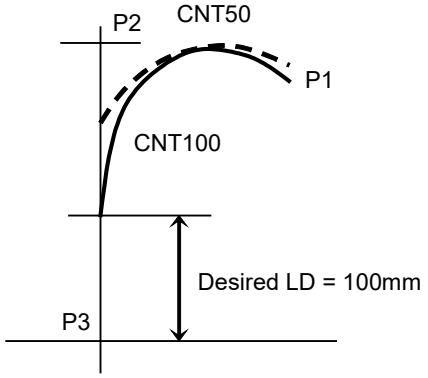


Fig. 14.1.1 (a) Linear Distance (RT_LD)

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.

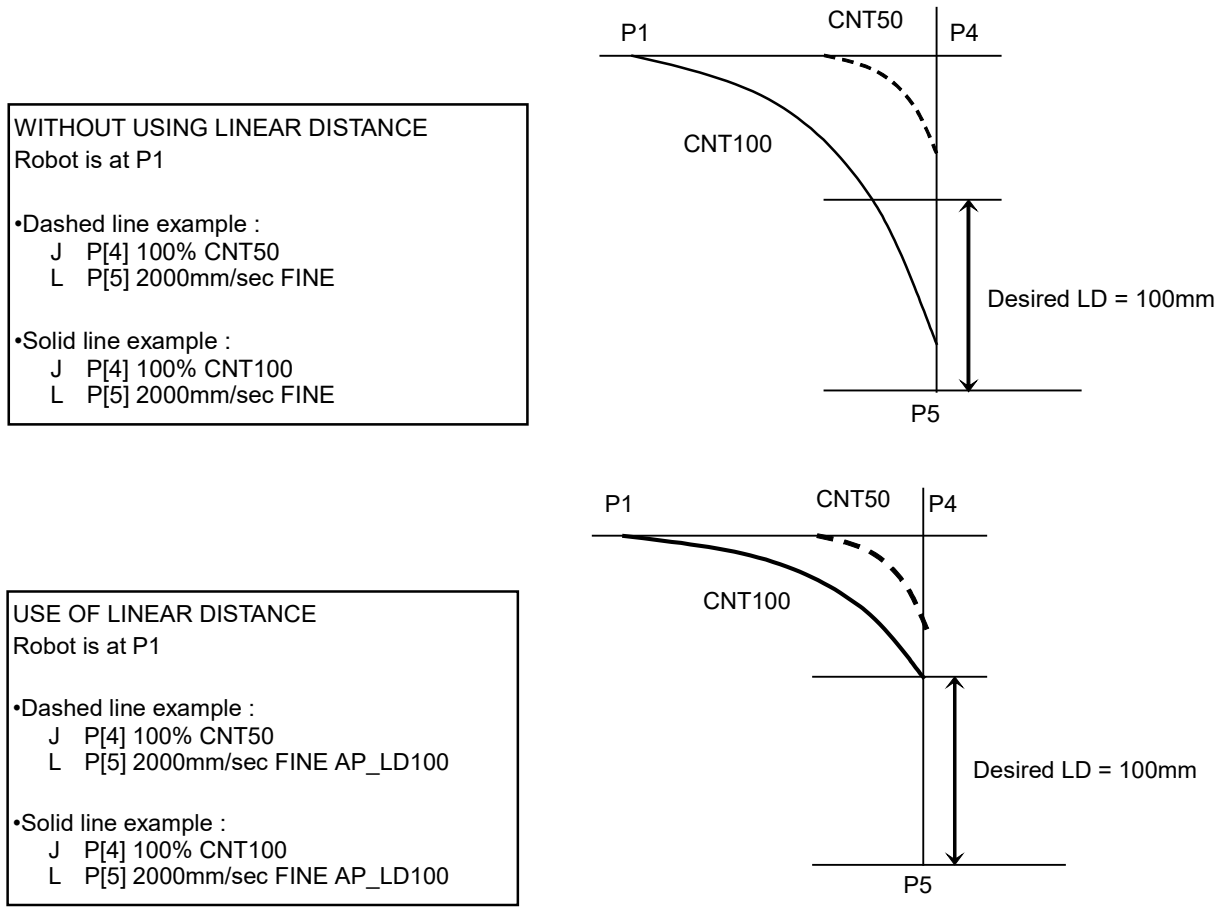


Fig. 14.1.1 (b) Linear Distance (AP_LD)

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 Limitations

- Linear Distance can not match the specified value exactly. However, it will never provide linear distance less than what you specified.
- Linear Distance can be used with only linear motion type.
- Linear Distance can not be used with Independent Axis or Positioner. If you use linear distance instructions, the instructions are disabled.
- When multiple group motion is used, the motion will be synchronized. However, if more than one group has linear distance enabled, all the groups will have linear distance satisfied.
- The local condition trigger time might be different than without Linear Distance. However the timing is repeatable.
- Linear Distance can not be used with the following options.
  - Max Speed
  - Weaving
  - Continuous turn

  - Robot Link
  - TCP speed output function
- Linear Distance can be used with Coordinated Motion.

14.1.3 Procedure to Use

Procedure 14-1 Procedure to use

Step

1 Press 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.

```
Motion Modify  1
1 No option
2 Wrist Joint
3 ACC
4 Skip,LBL[]
5 BREAK
6 Offset/Frames
7 PSPD
8 -next page--
```

```
Motion Modify  2
1 Offset,PR[ ]
2 Incremental
3 Retract_LD
4 Approach_LD
5 Tool_Offset
6 Tool_Offset,PR[
7 TIME BEFORE
8 -next page--
```

```
Motion Modify  3
1 Skip,LBL,PR
2 TIME AFTER
3 DISTANCE BEFORE
4 PTH
5
6
7
8 -next page--
```

4 If you want to specify the linear distance from the start position to the start of the corner, please select Retract_LD.
If you want to specify the linear distance from the end of the corner to the destination position, please select Approach_LD.
5 Type the number of millimeters that you want the tool center point (TCP) to approach or retract using Linear Distance.

NOTE
The default value is “direct” which means that the value is a specific number in millimeters. To use a value stored in a register, press the F3, INDIRECT, and type the register number.

14.2 CORNER REGION

L P[2] 100 mm/sec CRy

Corner Region (CR) is an optional termination type that can be used to adjust the corner rounding for Cartesian motions. When you use the CR termination type, you must specify the corner region value, y, (in millimeters). In addition to that, when you use the CR termination type, the speed at the corner keeps constant as far as possible.
Corner Region value is the distance from the starting of a corner path to the taught position, as shown in the following figure. When CR is specified, the tool center point path will maintain the corner path within the specified region, meaning that the actual distance from the starting/ending of a corner path to the taught position is less than or equal to the specified corner region value.

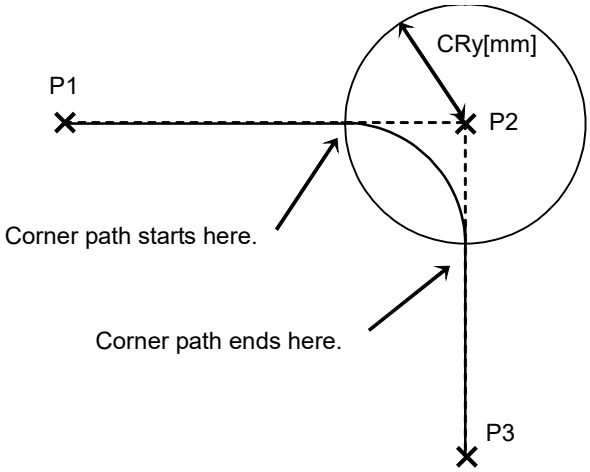


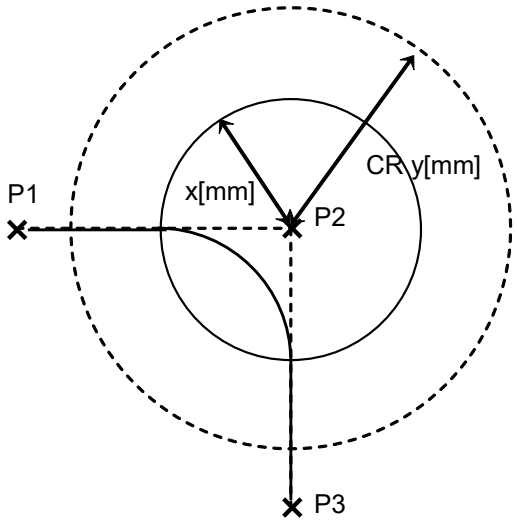
Fig. 14.2 Corner path

When you set corner region, use the following guidelines:
- Specify the corner region in millimeters.
- Corner region value can range in value from 0 mm to 1000 mm.
- The smaller the corner region value, the closer the robot will get to the taught position, and the less the corner rounding.
- With a larger corner region value specified, the robot will not get as close to the position and the more corner rounding.

14.2.1 Limitation of the Specifiable CR Value

If the specified corner region value, y, is greater than half distance between P1 and P2 (P2 and P3), then the actual value used is limited to half distance between P1 and P2 (P2 and P3) as shown below.

CR y[mm] > half distance between P1-P2 (P2-P3)
Actual CR x=half distance between P1-P2 (P2-P3)



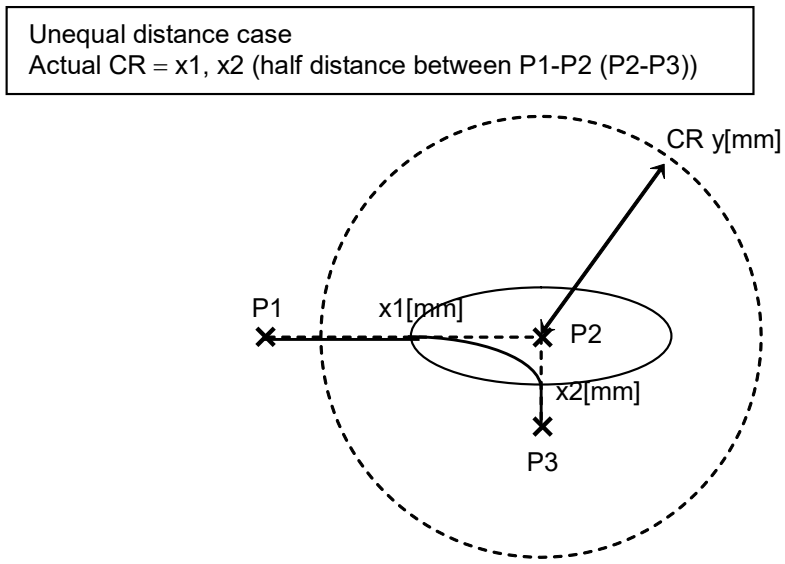


Fig. 14.2.1 Limitation of the specifiable CR value

Please keep in mind that because of the above rule, with large specified corner region value, corner path may still be close to the taught position when the distance between continuous taught points is short. Use the following guidelines when you teach a path:

- Minimize the number of taught positions.
- Reteach positions using the CR termination type to fit the path instead of adding positions.

14.2.2 Limitations

- CR termination type can not be used with JOINT motion type.
- CR termination type can not be used with Independent Axis or Positioner. When Independent Axis or Positioner move with Robot, it is possible to use CR termination type. However, in this case, CR takes effect for only Robot motion. Independent Axis or Positioner synchronizes with Robot.
- When multiple Robots move with multi group, the motion of robots is synchronized. In this case, all robot move so that the corner region is within the specified value.
- If the angle between the motion and the next motion is almost 180 degrees, CR termination type does not work and the path becomes the same as CNT100.
- The speed at the corner keeps constant as far as possible. However, when there is an instruction like the following, the speed decreases because it is impossible to read ahead of the motion.
  - Motion Statements with the target position specified by a position register.
  - Motion Statements with an offset instruction where an offset is given by a position register.
  - Motion Statements with the indirect addressing with the register.
  - Branch instruction (IF or SELECT JMP LBL[*]) is used before the next motion statement.
  - WAIT instruction with TIMEOUT LBL[*] is used before the next motion statement.
  - Frame instructions are used before the next motion statement.
- CR termination type can be used with the following options.
  - Remote TCP
  - Line Tracking
    During the transition (between Remote TCP and non-Remote TCP or between Tracking and non-Tracking), corner can not be defined because a frame is changed. Therefore the corner region can not be specified by CR but the corner path is connected continuously.
  - Linear Distance
    If you specify both Linear Distance and CR termination type, then Linear Distance has preference over CR, as shown following figures.

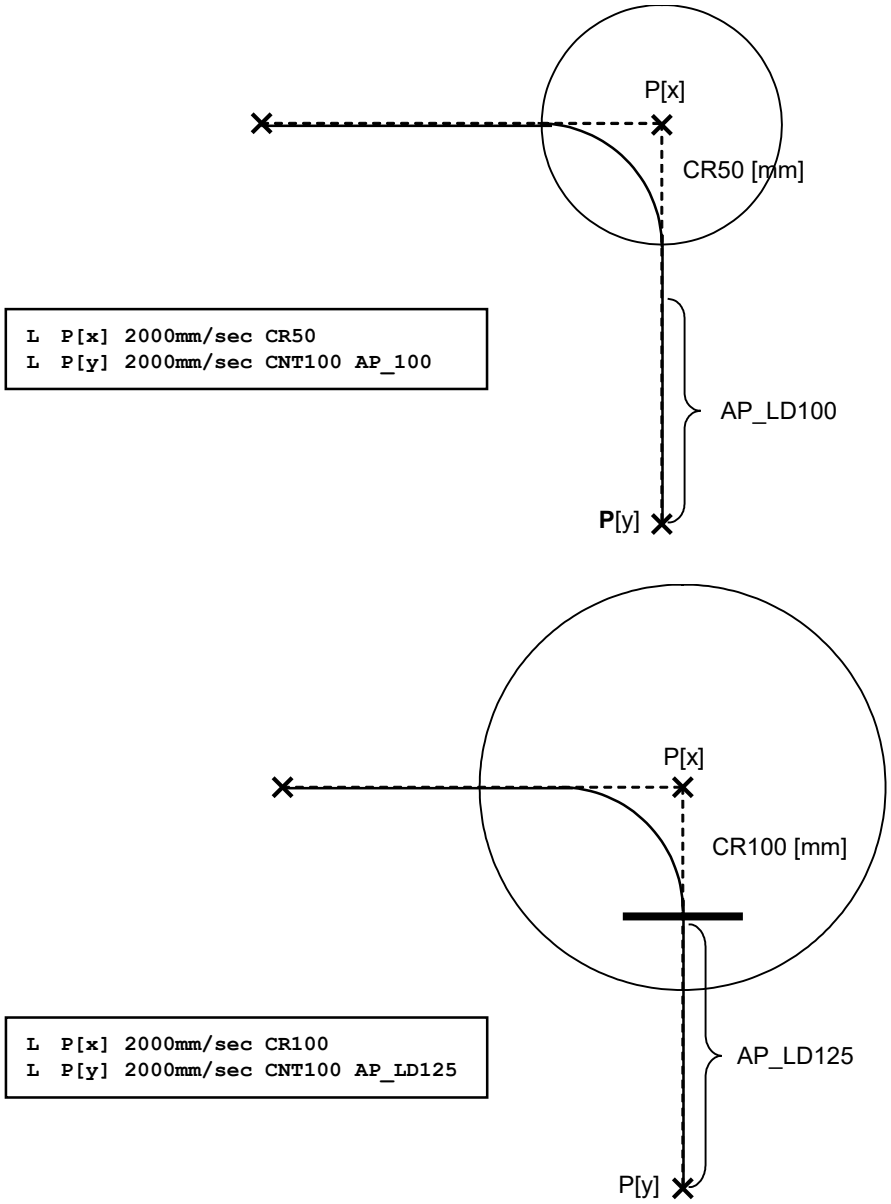


Fig. 14.2.2 Corner Path determined by CRy if Linear Dist is satisfied

14.3 PROCESS SPEED

PSPD xxx

Process speed is a motion option control feature that allows you to adjust robot speed to be faster or slower along a given path (if applicable, the path would be maintained the same regardless of xxx), where xxx is an integer you specify. The larger the value of xxx is, the faster the robot will move along the given path.
Process speed is useful for applications with continuous path motion that don't normally use maximum program speed; for example, sealing and waterjet cutting. Typically, the process controls the program speed: how fast the sealing gun can dispense, and how fast the waterjet can cut.
For these applications, teach the desired path using normal methods, tweaking taught position, speed, and term type.
After the path is taught, if you want to adjust its process speed from the nominal taught value, but do not want to change the path, you can use the Process Speed feature.

Add this motion option to the range of motion lines where adjustment is required.

- PSPD 100 is equivalent to the default cases without PSPD option.

- PSPD greater than 100 means faster process speed, while maintaining the same path.
- PSPD less than 100 means slower process speed, while maintaining the same path.
- You can still change other fields in the motion for further tweaking, but the same original rules apply; that is, the path will change. This allows you to adjust the path easily, even though PSPD is used.
- For PSPD greater than 100, the system internally limits the achievable (but higher) process speed, based on the jerk/acceleration margin available from the default case.

Be careful to use the PSPD option to reduce cycle time while maintaining the same path since jerk/acceleration value will be higher. An example is palletizing, where additional factors such as vibration, duty cycle, reducer life, and so forth, affect cycle time.

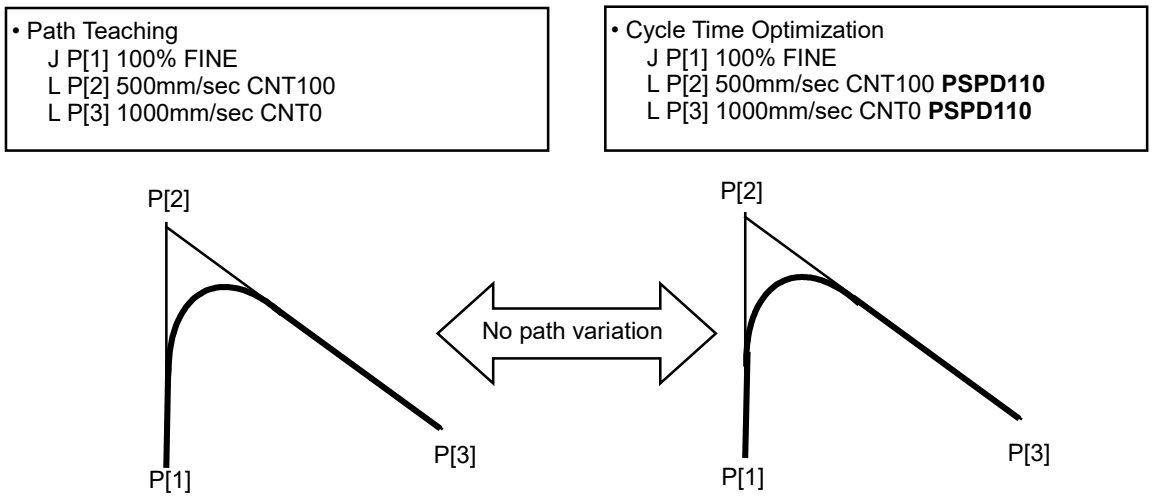


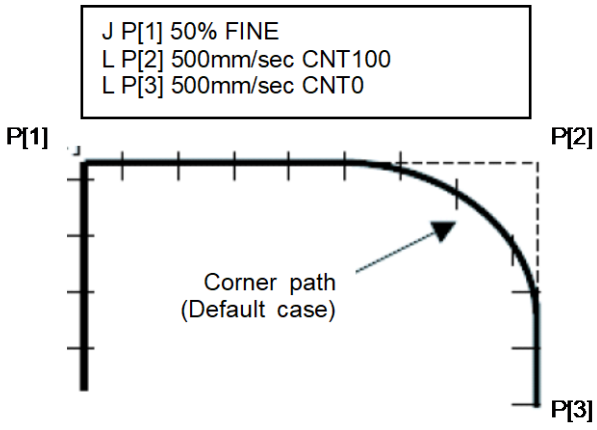
Fig. 14.3 Process speed tweaking

> ⚠ **WARNING**
> Process Speed can cause jerky motion if applied too aggressively. To avoid jerky motion, use a reduced speed.

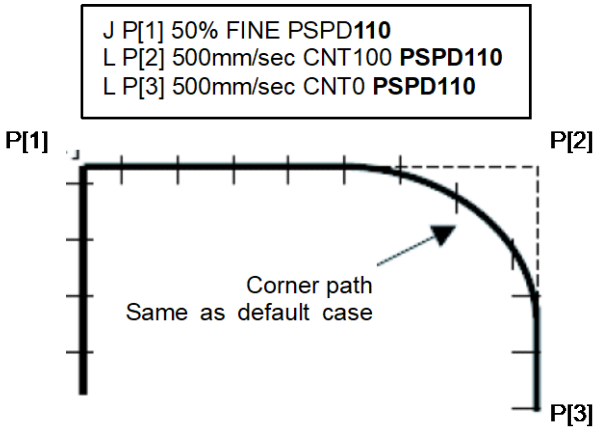
PSPDxxx can be added to any selective motion line in a TP program, and is applicable to all motion types. For examples, see the following:

- Case 1 : *** = 100, motion behavior is exactly the same as 100% speed override in the default case, as though there were no PSPD100.
- Case 2 : *** > 100, the speed will be faster than 100% speed override in the default case.
- Case 3 : *** < 100, the speed will be slower than 100% speed override in the default case.

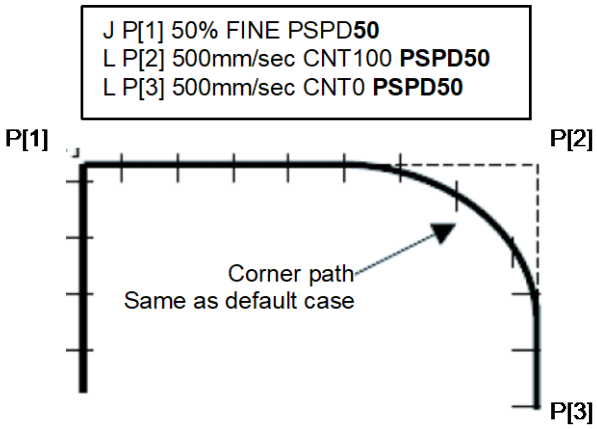
- Default case :



- Faster motion case (path is the same, cycle time is shorter):



- Slower motion case (path is the same, cycle time is longer):



> ⚠ **CAUTION**
> The system will internally limit the speed override such that the resulting motion performance is within mechanical capabilities. As a result, a large value of xxx may not take effect in some cases and the actual speed override may be smaller than the specified value.

14.3.1 Limitations

- Under T1 mode, PSPDxxx (with xxx>100) will not take effect.
- PSPDxxx (with xxx>100) might not take effect for the motion line that has max_speed as programmed speed in a TP program.
- The PSPD option does NOT support TCP speed prediction function (TCPP). That is, for the motions with PSPD option, TCPP might not result in correct results.
- With large PSPD value or very short segments, the actual corner path might deviate from the one without PSPD option.

14.4 MAX SPEED

L P[1] max_speed CNT100

In some applications, the desired speed is the maximum speed that the robot can deliver. For joint motion moves, the system delivers the maximum capability of the robot; that is, one of the axes reaches its

maximum speed. For linear moves, the system delivers the speed that is specified in the teach pendant instruction. However, the maximum linear speed of 2000mm/sec imposes a limit on the capability of the motor to reach higher speeds. The robot can move faster than the speed specified in the motion instruction.

The max speed option allows you to specify a linear motion that will use the maximum speed capability of the robot. It improves cycle times in Load/Unload applications by speeding up long linear motions. When this option is loaded, the choice of max_speed will be displayed in the speed field of the teach pendant motion instruction for a linear motion. The max_speed option affects only the motions for which the speed is specified as max_speed.

- If you change the motion type from Linear to Joint, the speed field will change to 100%.
- When the speed field changes from max_speed to another choice, the speed value will return to the original speed value.

> ⚠ **WARNING**
> When you specify max_speed, the robot will run at high speed. Be sure any loose parts are firmly attached and that the workpiece is secured. Otherwise, you could injure personnel or damage equipment.

14.4.1 Limitations

- If unsupported options are used, max_speed will be disabled automatically. No warning or error message will be displayed. This option does not support the following:
  - Any tracking option, such as line tracking, TAST, Coordinated motion, and so forth.
  - Multiple group motion
  - RTCP function
- If you run a program with an override speed different than 100%, the system will drive the robot such that one of its axes will reach the override value of its maximum joint speed.
- The local condition trigger time might have some variation.
- If the path becomes too aggressive, you might need to use ACC to smooth it.
- If you are using Dry Run, max speed will be disabled and the speed specified in dry run will be used.
- If you are using Org path resume, max speed will be disabled for the motion line that is resumed.
- If T1 is selected, the T1 speed will be used.
- In single step mode (FWD/BWD) max speed will be disabled and the maximum speed value will be used.
- Max speed will be disabled automatically for a circular motion.
- The max speed option will still apply when the Miscellaneous teach pendant instruction LINEAR_MAX_SPEED is used.
- The robot will try to attain the maximum speed capability of at least one of its axes. It determines the maximum speed for the current move by comparing the teach pendant instruction LINEAR_MAX_SPEED with the maximum linear speed of 2000 mm/sec. The ratio of these two speeds is the percentage of the maximum axis speed that the axis will reach.

  For example: the maximum linear speed is 2000 mm/sec.

```
1. LINEAR_MAX_SPEED = 1200
2. L P[1] max_speed CNT100
```

  The ratio of 1200 to 2000 is 60%. The system will drive the robot such that one of its axes will reach 60% of its maximum joint speed for line 2 of the program above.
- Max speed does not work with TCP speed prediction function (TCPP). That is, with Max speed, TCPP results may not be accurate.

15 SINGULARITY AVOIDANCE FUNCTION

For most of six-axis robots in FANUC, the singularity that occurs at elbow (at the boundary between "up" and "down" configurations) or at the boundary between "front" and "back" configurations causes regional structure degeneracy. This kind of singularity can easily be avoided by restricting the workplace of a robot. However, the wrist singularity may happen virtually any place inside the workplace. When a FANUC robot travels through/near a wrist singular position, motion performance becomes undesirable because

- Joint 4 and 6 will change a lot within a short time period;
- The TCP speed of the robot will slowdown;
- Path may deviate from the commanded one if a motor speed exceeds its limit.

Singularity avoidance function will provide a real-time solution to avoid the wrist singularity. With this function, the followings are achieved.

- The rotation of joint 4 and 6 is minimized, and robot can travel through/near a wrist singular position smoothly, and TCP speed can be maintained;
- This function works for both LINEAR TPE program motion and LINEAR jogging.

To use this function, singularity avoidance function option (A05B-2600-R792) or Motion package (A05B-2600-R809) is required. Some robot models do not support this function.

15.1 HOW TO USE SINGULARITY AVOIDANCE

How to use singularity avoidance in jogging and TPE program will be explained below.

15.1.1 How to Use Singularity Avoidance in Jogging

Enabled/disabled status of singularity avoidance function in jogging can be confirmed by checking whether “S/” is added before the manual-feed coordinate system shown in the teach pendant.
Example: “S/WORLD”
If “S/” is added, the singularity avoidance in jogging is enabled. If “S/” is not added, the singularity avoidance in jogging is disabled.
Enabled/disabled of singularity avoidance in jogging is automatically switched when the program is selected.
If you select a program for which the singularity avoidance is enabled, it will be enabled, and if you select a program for which it is disabled, it will be disabled.
In order to enable/disable singularity avoidance without changing the selected program, refer to the following procedure.

Procedure 15-1 Select enabled/disabled of singularity avoidance in jogging

Step

1 Press the [FCTN] key, and select “T/Singularity JOG” to toggle enabled/disabled of singularity avoidance in jogging.

15.1.2 How to Use Singularity Avoidance in TPE Program

Enabled/disabled of singularity avoidance in TPE program can be selected for each program. When a new TPE program is created, the enabled/disabled of the program is automatically determined depending on the enabled/disabled status of singularity avoidance in jogging. Please note that in the case that the user loads a TPE program which is saved before the singularity avoidance option is added or is saved in the system

which does not have the singularity avoidance option, the singularity avoidance is disabled for the TPE program.

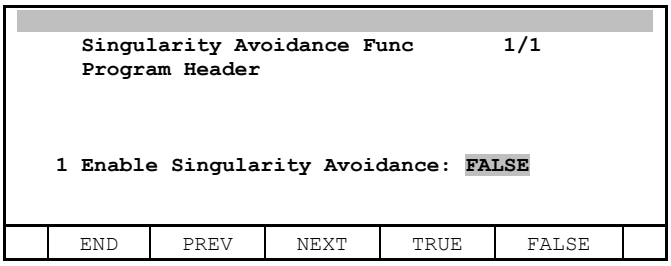
In order to select enabled/disabled for each program, refer to the following procedure.

Procedure 15-2 Select enabled/disabled of singularity avoidance in program

Step

1 In the program select screen, move the cursor to the program and press the F2, DETAIL key to enter the detail screen.

2 Press the F3, NEXT key, and the following screen will be displayed. (This screen shows the singularity avoidance is disabled.)



3 In order to enable the singularity avoidance, press the F4, TRUE. In order to disable the singularity avoidance, press the F5, FALSE.

15.2 LIMITATIONS

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 switching to Handling in Spot Tool +.)
- Some robot models does not support this function.
- This function cannot be ordered with Coordinated motion option.
- This function cannot be used with Line tracking option.
- This function cannot be used with Continuous turn option.
- This function cannot be ordered with Shape GenerationII option.
- This function can be used only for linear motion. This function cannot be used for Circular/Arc motion.

15.3 CAUTIONS

Keep the following important information in mind when you set up and use singularity avoidance.

- With the singularity avoidance function, actual wrist configuration (flip/nonflip) might be different from the taught destination positions. The function might change configuration internally not only for the destination position in the motion line where singularity is detected but also for the subsequent destination positions in the following motions in the TP program.
- When the taught point is near singularity, reached joint position may be different from taught joint
- Since the function will change configuration internally, single step forward and backward might produce different behavior. To prevent the above from happening, during single step forward motion, the system will post a warning message "MOTN-208 Config Not Reached" at the point where the actual configuration is different from the taught one. By observing the message, the user should re-touch up the taught point at the specific line shown in the warning message this will update the

taught configuration to the actual configuration. As a result, when stepping backward, the motion will behave the same way as stepping forward.

- Jogging and program motion might be different when the robot moves near singularity.
- The function might not help for the singularity at the corner path.
- If the position reached in single-step mode is near singularity, “MONT-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

Path switching function (A05B-2600-J693) is the function to execute the assignment statements at the several points in a motion path.
When Path Switching (hereinafter called PS) instruction is specified in the DB calling program as follows, PS statements define the trigger point and the assignment statement to be executed.

DB CALL statement

```
1: L P[1] 200mm/sec FINE DB 100mm, CALL A
```

Contents of sub-program A

```
1:    PS -100mm +0.2sec,DO[1]=(ON)
2:    PS -150mm -0.2sec,IF(DI[1]),DO[2]=(DI[2] AND !DI[3])
```

The distance specified just after DB statement does not affect to the each trigger timing of the PS commands in the sub-program. The PS statements in the sub program start their execution when the motion starts. User can mix another normal logical statement with the PS statement in the sub program.

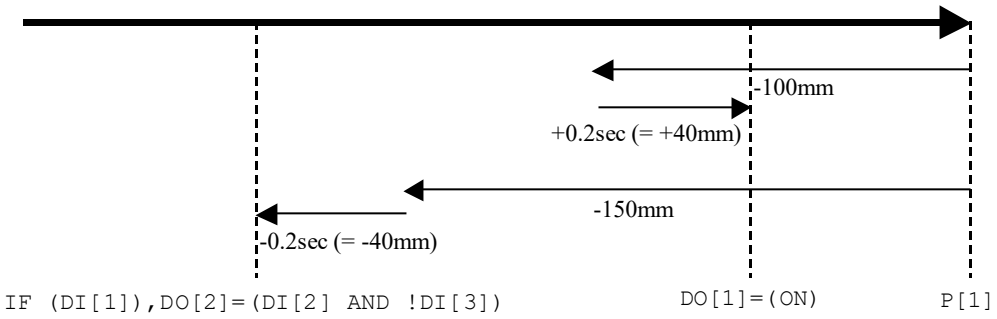


Fig. 16 (a) Path switching application

PS instruction is composed of two parts, trigger point and assignment statement.

```
PS -100mm +0.2sec,DO[1]=(ON)
     Trigger point   Assignment statement
```

Trigger point part specifies the trigger distance (mm) from the destination position and the offset time (sec). When you enter a negative value, you can specify the trigger point before the motion is completed (distance-before type).
When you enter a positive value, you can specify the trigger point after the motion is completed. (distance-after type)
The offset time is converted to the distance according to the specified speed, in the above case, 0.2sec is converted to 40mm (= 200mm/sec * 0.2sec). This conversion is not accurate in the acceleration and deceleration period.
The adjustment of trigger point by the offset time is affected by override value.

For R-30*i*B/R-30*i*B Mate, according to the sign of the trigger distance value, its behavior is treated as distance-before or distance-after type. When the trigger distance value is minus value, PS instruction is regarded as distance-before type. Therefore, even if plus value is set as offset time and the trigger point is calculated as plus value, actual trigger point will be clamped to -0.0mm. And when the trigger distance value is plus value, PS instruction is regarded as distance-after type. Therefore, even if minus value is set as offset time and the trigger point is calculated as minus value, actual trigger point will be clamped to +0.0mm.
For R-30*i*B Plus/R-30*i*B Mate Plus, the trigger point is not clamped by plus/minus of the trigger distance.

Assignment statement part specifies the statement to be executed at the trigger point. The assignment statement can have IF condition.

```
PS -150mm -0.2sec,IF (DI[1]),DO[1]=(ON)
```

In this case, when DI[1] is ON, DO[1] is turned on, when DI[1] is OFF, DO[1] is not changed.
The assignment statement can use the following items and operators.

Items for left side	Operator for right side or IF condition	Items for right side or IF condition
DO[ ]	( )	ON OFF
RO[ ]	+	Constant value
WO[ ]	-	DI[] DO[ ]
AO[ ]	*	RI[] RO[ ]
GO[ ]	/	WI[] WO[ ]
SO[ ]	MOD	AI[] AO[ ]
UO[ ]	DIV	GI[] GO[ ]
F[ ]	AND	SI[] SO[ ]
R[ ]	OR	UI[] UO[ ]
PR[i,j]	!	F[ ]
System variables	= (comparison)	M[ ]
M[ ]	<	R[ ]
TC_ONLINE	>	PR[i,j]
	<=	TIMER[ ]
	>=	System variables
	<>	TCP_SPD[ ]

Marker assignment and TC_ONLINE can be specified in PS statement.
AR[ ] (calling parameter) can not be specified in PS statement.
Up to 20 operations can be specified in one motion line as total of right side formula and IF condition.
One motion statement can have up to 20 PS statements.

Case that the trigger point is negative and it is farther than the start position (distance-before type)

If the trigger point is negative and it is farther than the start position, this is triggered immediately.

```
1: L P[1] 200mm/sec FINE
2: L P[2] 200mm/sec FINE DB 100mm, CALL A
```

Contents of sub-program A

```
1:   PS -1000mm +0sec,DO[1]=(ON)
```

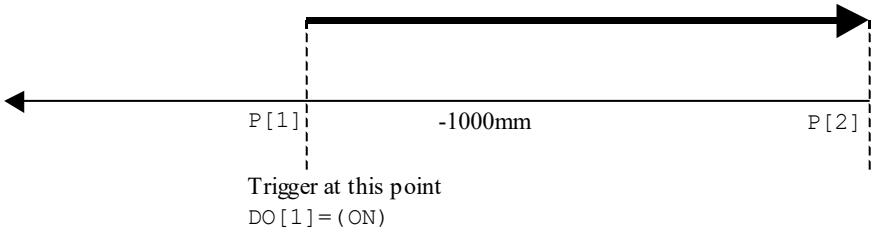


Fig. 16 (b) Path switch application

You can choose to cause error at this situation by following parameter setting.

$PS_CONFIG.$DB_IMMTRIG (distance-before type)
$PS_CONFIG.$DA_IMMTRIG (distance-after type)

Value	Description
0	Trigger PS condition immediately, but post neither alarm nor warning.
1 (default)	Trigger PS condition immediately, and post “INTP-539 PS(program name, line number, distance[mm]) Already in area” (warning).
other than (0, 1)	Not triggered, and post “INTP-540 PS(program name, line number, distance[mm]) Already in area” (PAUSE).

Case that the trigger point is passed by jog while the program is paused

When original path resume is disabled ($SCR.$ORG_PTH_RSM=FALSE), if the program is paused during the motion with PS, and jogged by TP, and the trigger point is passed by jog, then the PS condition is triggered immediately when the program is resumed.
When original path resume is enabled ($SCR.$ORG_PTH_RSM=TRUE), once back to the paused position on the original motion path and re-start the program. In this case, the PS condition is triggered when the trigger point is passed after resume and re-start.

You can choose to cause error at this situation by above parameter setting.
($PS_CONFIG. $DB_IMMTRIG, $DA_IMMTRIG)

```
1: L P[1] 200mm/sec FINE
2: L P[2] 200mm/sec FINE DB 100mm, CALL A
```

Contents of sub-program A

```
1:    PS -500mm +0sec,DO[1]=(ON)
```

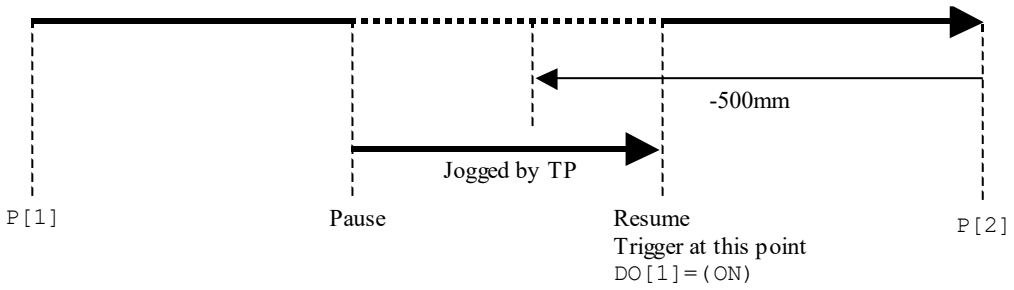


Fig. 16 (c) Path switch application (original path resume is disabled)

Case that TCP path moves away without entering the trigger area on CNT motion

In case of CNT motion, teaching point does not on the actual robot path. If the robot path is too far from the trigger area, reminding untriggered distance-before type PS conditions are triggered when the TCP path starts to move away from the teaching point.

```
1: L P[1] 200mm/sec FINE
2: L P[2] 200mm/sec CNT100 DB 100mm, CALL A
3: L P[3] 200mm/sec FINE
```

Contents of sub-program A

```
1:   PS -10mm +0sec,DO[1]=(ON) (distance-before type)
2:   PS +10mm +0sec,DO[2]=(ON) (distance-after type)
```

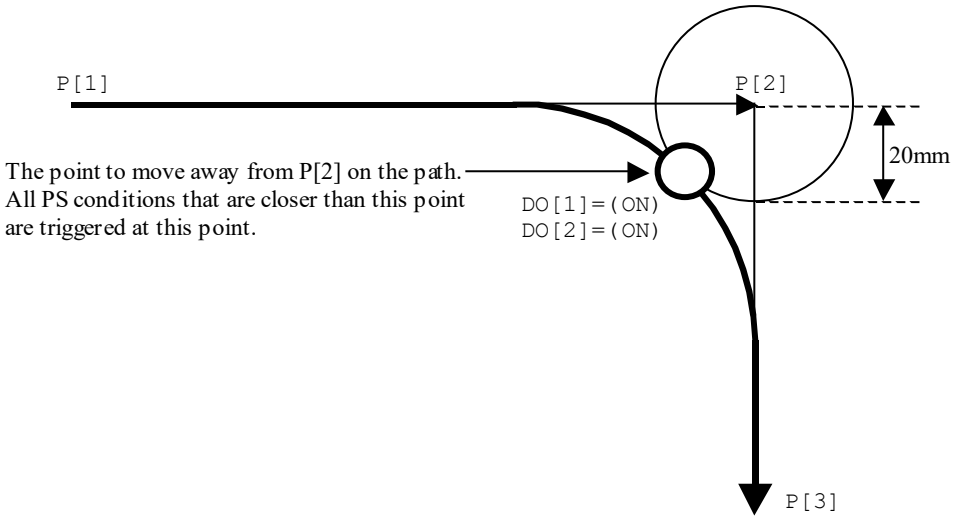


Fig. 16 (d) Path switch application

You can choose to cause error at this situation by following parameter setting.

$PS_CONFIG.$DB_NOTRIG (distance-before type)
$PS_CONFIG.$DA_NOTRIG (distance-after type)

Value	Description
0	Trigger PS condition at the closest point, but post neither alarm nor warning.
1 (default)	Trigger PS condition at the closest point, and post “INTP-541 PS(program name, line number, distance[mm]) Forced trigger” (warning).
other than (0, 1)	Not triggered, and post “INTP-542 PS(program name, line number, distance[mm]) No trigger” (PAUSE).

Case that the trigger point is positive and it is farther than the destination position (distance-after type)

If the trigger point is positive and it is farther than the destination position of the next motion, the PS condition is triggered when the next motion is completed.
You can choose to cause error at this situation by above parameter setting.
($PS_CONFIG. $DB_NOTRIG, $DA_NOTRIG)

```
1: L P[1] 200mm/sec FINE
2: L P[2] 200mm/sec FINE DB 100mm, CALL A
3: L P[3] 200mm/sec FINE
```

Contents of sub-program A

```
1:    PS +1500mm +0sec,DO[1]=(ON)
```

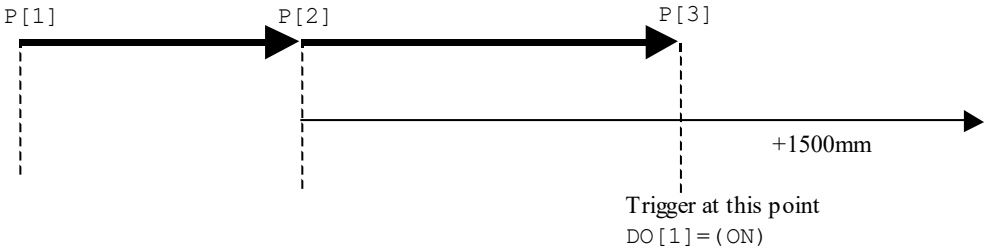


Fig. 16 (e) Path switch application

About multi group configuration

In case of multi group configuration, the PS instruction is implicitly executed for the first motion group in the group mask of the TP program. For example, if the motion mask is [*,1,1,*,*,*,*,*], then the PS instruction is executed for group 2.
If the software version is 7DF3/03(V9.30P/03) or earlier and you want to use PS instruction in a group other than group 1, set $PS_CP_CFG[group].$ENB=1 and restart the controller.
If you do not set $PS_CP_CFG[group].$ENB=1, "INTP-537 PS Failed to get cur pos(G:%d)" alarm will occur.

Limitations

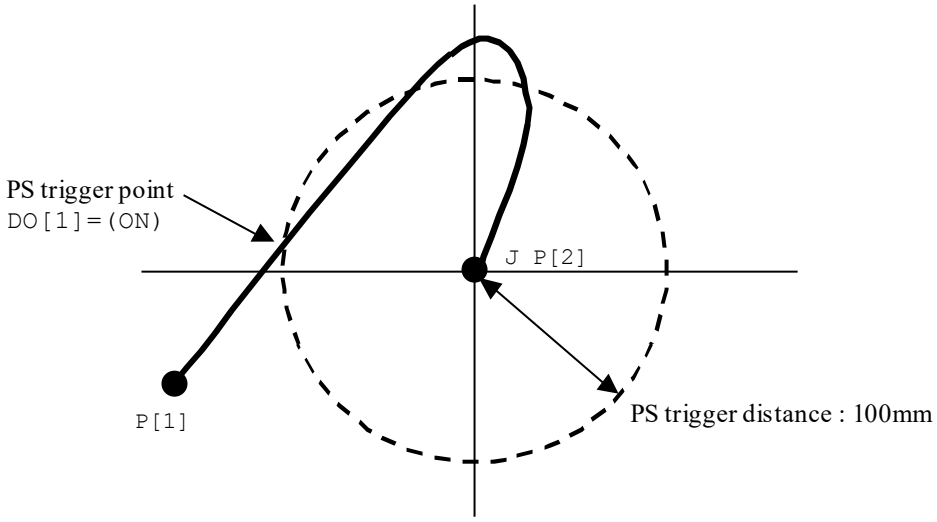
PS statement is not available for the following motion type.

- Weaving motion
- Tracking motion which can not define actual destination point beforehand, as line-tracking, MIG-EYE, Robot-Link.

Caution

- PS statement is interpreted in a flash only at the start of the new motion line. When the motion including PS statement is paused, if you edit and change the PS statement in the sub program during this pause status, modified PS commands do not take effect until the motion line is newly executed. After pausing program, to take effect of the modified PS statement immediately, you can BWD the motion line and re-start. Then the modified PS statements will take effect.
- PS command for joint motion is not prohibited. But it must be used carefully. Regardless of the motion type Linear, Circular, or Joint, PS command is checking the linear distance between TCP and destination point as trigger threshold.

> ⚠ **CAUTION**
> In cases of PS command's Distance-Before type usage, If TCP once goes into the defined sphere around the destination point, the PS command will be triggered regardless how long the rest of path. Please note the distance which is being checked in PS motion is not the total remaining path length to the destination point. Please see the following figure for example.



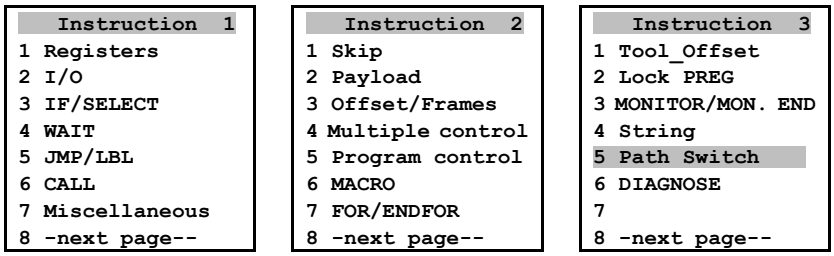

```
1: J P[1] 100% FINE
2: J P[2] 100% FINE DB 100mm, CALL PROG1
```

Contents of sub-program PROG1

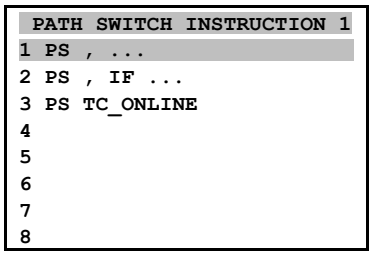
```
1: PS -100mm +0.0sec, DO[1] = ON
```

TP editor operation

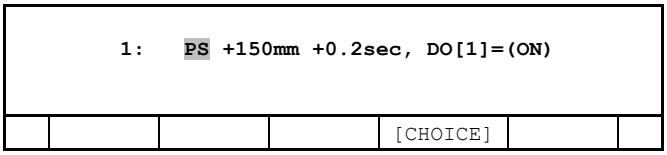
1 "Path Switching" item is displayed in the [INST] menu.



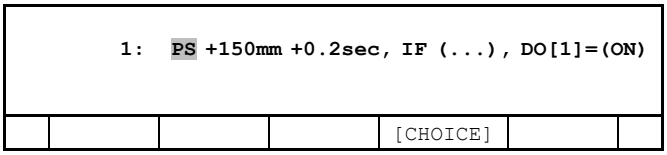
2 Select "Path Switching" item, the following menu will be displayed.



3 The distance and offset time can be set, and the assignment statement can be edited as the same way as mixed logic instructions.



4 When the cursor is on 'PS ' item, you can switch "PS ..." to "PS ... IF ..." and vice versa from menu.



17 AUXILIARY AXIS SERVO OFF (LOCAL STOP) FUNCTION

In the robot system that auxiliary axis controls a jig, there is a requirement to cut off the power of auxiliary axis for safety when an operator does a job with that jig. Auxiliary axis servo off (Local stop) is a function for that purpose. This function can turn off motor power supply for each auxiliary axis respectively and prevent axis from unexpected movements.

This function has following merits.

- Keep safety for operator who works at jig (auxiliary axis).
- Improve the cycle time of process because it is not necessary to stop the operation of the robot when the operator enters in Load station to replace the work piece.
- Output servo ON/OFF signal by just using macro program or press button.

To use this function, auxiliary axis servo off option (A05B-2600-J806) and dedicated hardware and setup at DCS Local stop menu are required. In regard to the dedicated hardware or DCS Local stop menu, please refer to the following manuals.

- R-30*i*B Local Stop function with STO Maintenance and Order Manual (A-95028E)
- Chapter "AUXILIARY AXIS SERVO OFF (LOCAL STOP) FUNCTION" in R-30*i*B/R-30*i*B Mate/R-30*i*B Plus CONTROLLER Dual Check Safety Function OPERATOR'S MANUAL (B-83184EN)

17.1 SPECIFICATION

- Order number for software option
  A05B-2600-J806

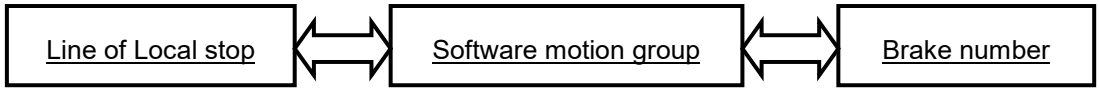
- Way to cut off motor power supply
  This function cut off power from servo amplifier to motor. Even in the Local stop mode (power supply for motor is cut off), power is supplied to the amplifier.

- Motor for the auxiliary axis
  Please be sure to use motor with brake for this function and set brake number independent of other motion group when you set up auxiliary axis.

- Local stop Hardware
  A dedicated hardware is required. In regard to the dedicated hardware, please refer to R-30*i*B Local Stop function with STO Maintenance and Order Manual (A-95028E).

- Setup DCS Local stop menu
  To use this function, please be sure to setup at DCS Local stop menu. In regard to DCS Local stop menu, please refer to R-30*i*B Local Stop function with STO Maintenance and Order Manual (A-95028E) and chapter "AUXILIARY AXIS SERVO OFF (LOCAL STOP) FUNCTION" in Dual Check Safety Function OPERATOR'S MANUAL (B-83184EN).

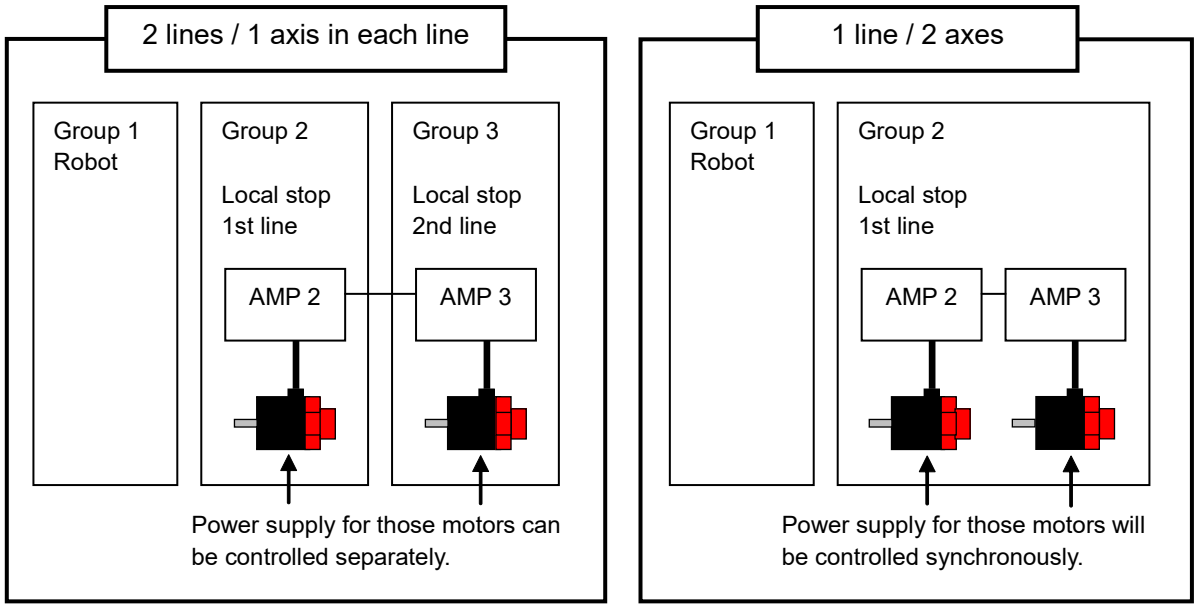
- Line of Local stop
  The unit (group) in which power supply of motors is controlled synchronously is called as Line of Local stop. Line concept corresponds to the software motion group. Motors in the same line must have the same brake number and motors in the different line must have the different brake number. Software supports 8 Local stop lines at maximum.



- Number of axes in the Local stop line
  If there are multiple axes in the one Local stop line, power supply for those motors will be controlled synchronously. In this case, there are multiple axes in one motion group. Power supply for those motors cannot be controlled separately. If you want to control separately, please separate a Local stop line.

> Configuration of Local stop function is determined by the number of lines and the number of axes in one line.

- Example of system configuration
  When there are two axes that use Local stop function, possible configurations are as follows.



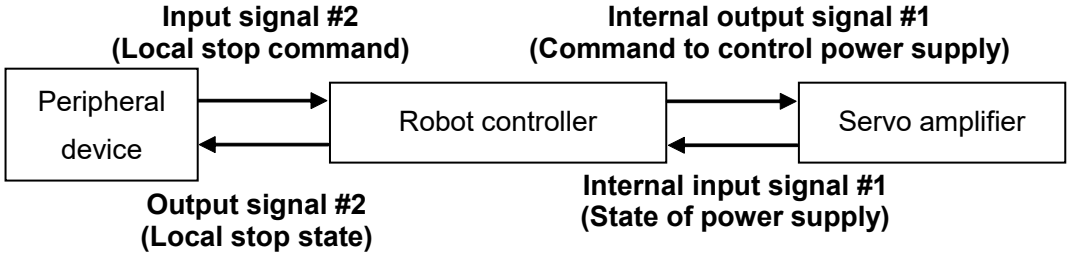
- Necessary signals for Local stop function
  One input signal and one output signal are required for each Local stop line.

  1 Input signal #2
  Local stop command signal by the user.
  Local stop software controls power supply for the motor in response to this signal.
  - When this signal is ON, supply power to the motor and the axis can move.
  - When this signal is OFF, cut off power to the motor and the axis cannot move.

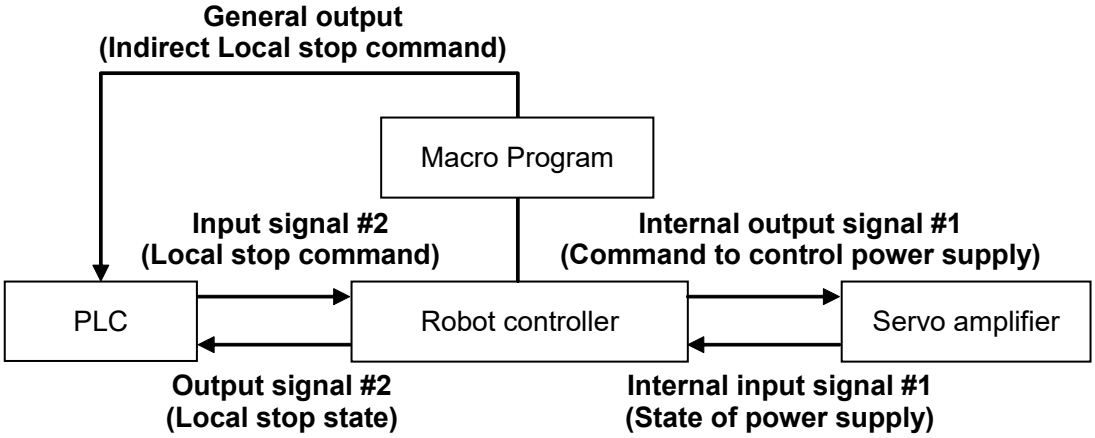
  2 Output signal #2
  Local stop status feedback signal from a robot controller to a peripheral device such as PLC.
  This signal shows the state of Local stop software controls power supply for the motor. Please use this signal as muting indicator and WAIT condition of wait for transit servo off/on to complete in TP program. Please do not operate this signal.
  - When this signal is OFF, power is supplied to the motor and the axis can move.
  - When this signal is ON, power is not supplied to the motor and the axis cannot move.

  Please assign general I/O or robot I/O for input signal #2 and output signal #2.

- Basic flow of process
  When the user inputs the input signal #2, software outputs the internal output signal #1 to the amplifier and power supply for motor will be controlled. Software also outputs the output signal #2 to a peripheral device after checking the state of power supply in amplifier by the internal input signal #1. Please connect a peripheral device such as push button to the input signal #2.



In case a macro program will input that Local stop command, please output general output signal in that program to the peripheral device such as PLC and control input signal #2 according to that output.



17.2 CONSTRAINTS

- "LSTP-001 Motion grp $_{xx}$ is active" alarm occurs when the user command cutting off the power supply (input signal #2 = OFF) while auxiliary axis is moving. Then running program will be paused and robot stops when this alarm occurs. Please command power off while axis is not moving.
- "LSTP-011 Motion grp $_{xx}$ is in LSTOP" alarm occurs when the user tried to move axis while the power supply is cut off. Then axis does not move. Please turn on power supply for auxiliary axis before you move axis.
- Please be sure to specify FINE positioning path for last motion instruction right before command power off (input signal #2 = OFF). If specify CNT positioning path for last motion instruction right before command power off, "LSTP-001 Motion grp $_{xx}$ is active" alarm might occur.
- Please be sure to keep time more than 0.2 sec before command power off (input signal #2 = OFF). If command power off right after last motion instruction without waiting 0.2 sec, "LSTP-001 Motion grp $_{xx}$ is active" alarm might occur.

17.3 SETTINGS

1 Firstly, please connect a dedicated hardware. In regard to the dedicated hardware, please refer to R-30*i*B Local Stop function with STO Maintenance and Order Manual (A-95028E).
2 Secondly, please setup DCS Local stop menu. In regard to DCS Local stop menu, please refer to R-30*i*B Local Stop function with STO Maintenance and Order Manual (A-95028E) and chapter "AUXILIARY AXIS SERVO OFF (LOCAL STOP) FUNCTION" in Dual Check Safety Function OPERATOR'S MANUAL (B-83184EN).
3 Thirdly, please set system variables. For each Local stop line, setting of motion group number, input signal #2, output signal #2 and internal output signal #1 are necessary. For input signal #2 and output signal #2, please specify port type and port number. For internal output signal #1, please specify the SPO index number. In regard to the SPO index number, please refer to R-30*i*B Local Stop function with STO Maintenance and Order Manual (A-95028E) and chapter "AUXILIARY AXIS SERVO OFF (LOCAL STOP) FUNCTION" in Dual Check Safety Function OPERATOR'S MANUAL (B-83184EN).

Procedure to set up:

1 Display the SYSTEM Variables screen.
  a) Press the [MENU] key.
  b) Select 0 NEXT and then 6 SYSTEM.
  c) Press the F1, [TYPE] key.
  d) Select Variables, and then SYSTEM variables screen will be displayed.
2 Move the cursor to $LS_IOPORT and press the [ENTER] key. The following screen will be displayed.

```
SYSTEM Variables
    $LS_IOPORT                                  1/8
      1   [1]           LS_IOPORT_T
      2   [2]           LS_IOPORT_T
      3   [3]           LS_IOPORT_T
      4   [4]           LS_IOPORT_T
      5   [5]           LS_IOPORT_T
      6   [6]           LS_IOPORT_T
      7   [7]           LS_IOPORT_T
      8   [8]           LS_IOPORT_T

    [ TYPE ]  DETAIL
```

3 Select the Local stop line (1-8) that needs to be configured and press the [ENTER] key. The following screen will be displayed.

```
SYSTEM Variables
    $LS_IOPORT[1]                               1/13
      1 $MO_GRP_NUM     0
      2 $SDI1_P_TYPE    0
      3 $SDI1_P_NUM     0
      4 $SDI1_P_STAT    FALSE
      5 $SDI2_P_TYPE    0
      6 $SDI2_P_NUM     0
      7 $SDI2_P_STAT    FALSE
      8 $SDO1_P_TYPE    -1
      9 $SDO1_P_NUM     0
     10 $SDO1_P_STAT    FALSE
     11 $SDO2_P_TYPE    0
     12 $SDO2_P_NUM     0
     13 $SDO2_P_STAT    FALSE

    [ TYPE ]
```

4 Please refer to the below tables and setup the system variables according to the actual I/O connection. It is required to setup system variables which "O" (Need to setup) is written in rightmost line in the following Table 17.3. For system variables which "X" (No need to setup) is written, please do not change.

Table 17.3 Description of system variable (Following settings are necessary for each Local stop line)

Variable name	Description	O: Need to setup X: No need to setup
$MO_GRP_NUM	Motion group number that uses a Local stop function. Valid range is 2-8. Setting this variable to 0 disables Local stop function for the Local stop line.	O
$SDI1_P_TYPE	Not in use.	X
$SDI1_P_NUM	Not in use.	X
$SDI1_P_STAT	Status of internal input signal #1. Please do not change this variable.	X
$SDI2_P_TYPE	Port type for input signal #2 (Local stop command signal by a user).	O
$SDI2_P_NUM	Port number for input signal #2.	O
$SDI2_P_STAT	Status of input signal #2. Please do not change this variable.	X
$SDO1_P_TYPE	Signal type for internal output signal #1 (Command to control power supply from a robot controller to servo amplifier). Please do not change from the default value (-1).	X
$SDO1_P_NUM	Specify the SPO index number. In regard to the SPO index number, please refer R-30*i*B Local Stop function with STO Maintenance and Order Manual (A-95028E).	O
$SDO1_P_STAT	Status of internal output signal #1. Please do not change this variable.	X
$SDO2_P_TYPE	Port type for output signal #2 (Local stop status feedback signal from a robot controller).	O
$SDO2_P_NUM	Port number for output signal #2.	O
$SDO2_P_STAT	Status of output signal #2. Please do not change this variable.	X

Values for port type ($LS_IOPORT[n].$SDI2_P_TYPE, $SDO2_P_TYPE) are as follows.
Please assign general I/O or robot I/O for the input signal #2 and output signal #2.

Value for port type	Type of signal
1	DI (General input signal)
2	DO (General output signal)
8	RI (Robot input signal)
9	RO (Robot output signal)

Example of configuration
In case that assign DI[10] for the input signal #2 and DO[11] for the output signal #2 for Local stop line 1, set the above system variables as follows.
$LS_IOPORT[1].$SDI2_P_TYPE = 1
$LS_IOPORT[1].$SDI2_P_NUM = 10
$LS_IOPORT[1].$SDO2_P_TYPE = 2
$LS_IOPORT[1].$SDO2_P_NUM = 11

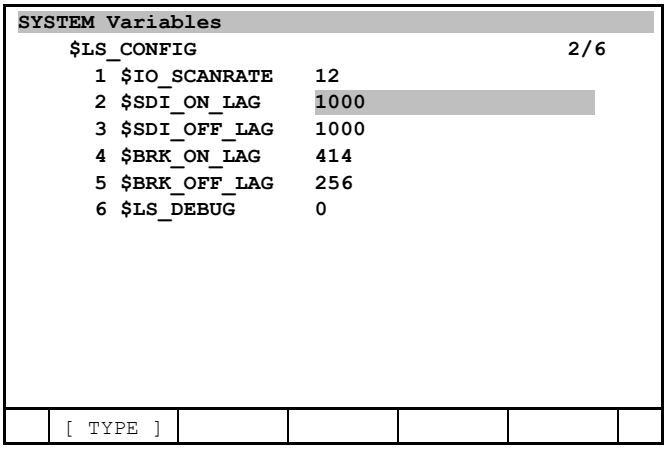
5 When you use more than 2 Local stop lines, please set system variables for 2nd line or later too.
6 Please do not change $LS_CONFIG and $LS_SYSTEM.
7 That is all for software setup. In order to make settings available, please cycle power once. After cycling power, setting will be available.
8 To confirm settings are proper, please do check by following procedure for each Local stop lines.

Confirmation 1) Exiting from Local stop mode
When you set the input signal #2 to ON, the output signal #2 becomes OFF.

Confirmation 2) Entering into Local stop mode
When you set the input signal #2 to OFF, the output signal #2 becomes ON.

⚠CAUTION
Please perform these checks when the system is NOT in servo off status.
If any alarm factor such as Operator panel E-stop, Teach pendant E-stop, deadman switch release, External emergency stop, etc. exists, please clear these alarms first. Then perform a test when the system is in servo on status.

9 If no alarm occurs and signals change like above, Local stop function works correctly.
10 Following alarms could occur.
"LSTP-006 DI1 ON timer expired (G: xx)" or "LSTP-007 DI1 OFF timer expired (G: xx)"
In case that
"LSTP-006 DI1 ON timer expired (G: xx)" occurs but the input signal #1 becomes ON or
"LSTP-007 DI1 OFF timer expired (G: xx)" occurs but the input signal #1 becomes OFF,
please increase following system variables
$LS_CONFIG.$SDI_ON_LAG and $LS_CONFIG.$SDI_OFF_LAG 100 by 100 until alarm does not occur. The default value for these variables is 1000. If you change the value, please set the same value to both variables.

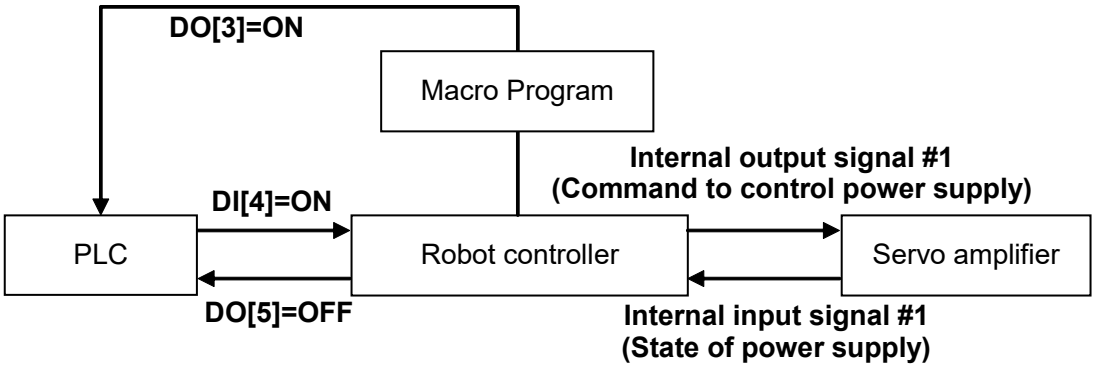


When "LSTP-006 DI1 ON timer expired (G: xx)" or "LSTP-007 DI1 OFF timer expired (G: xx)" occurs, it is necessary to cycle power once to clear alarm status.
11 If alarms occur except for the above situation, please refer to FANUC Robot series OPERATOR'S MANUAL (Alarm Code List) (B-83284EN-1) and check the settings of system variable, setting of brake number, and hardware connections.

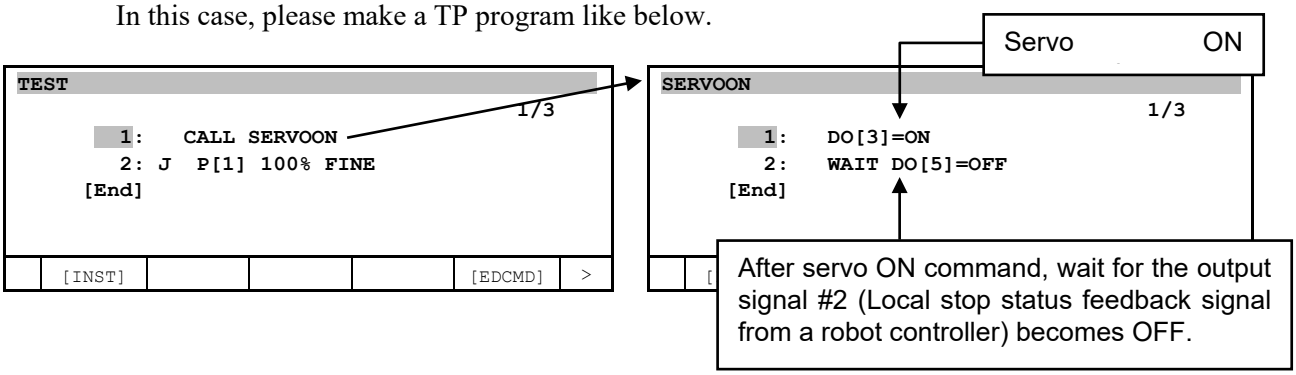
17.4 ATTENTION

- Note for moving auxiliary axis right after exiting from Local stop mode (servo ON command)
  Please do not move auxiliary axis until output signal #2 becomes OFF.

  Example) Assuming the system in which I/O has been set up like below.



  In this case, please make a TP program like below.



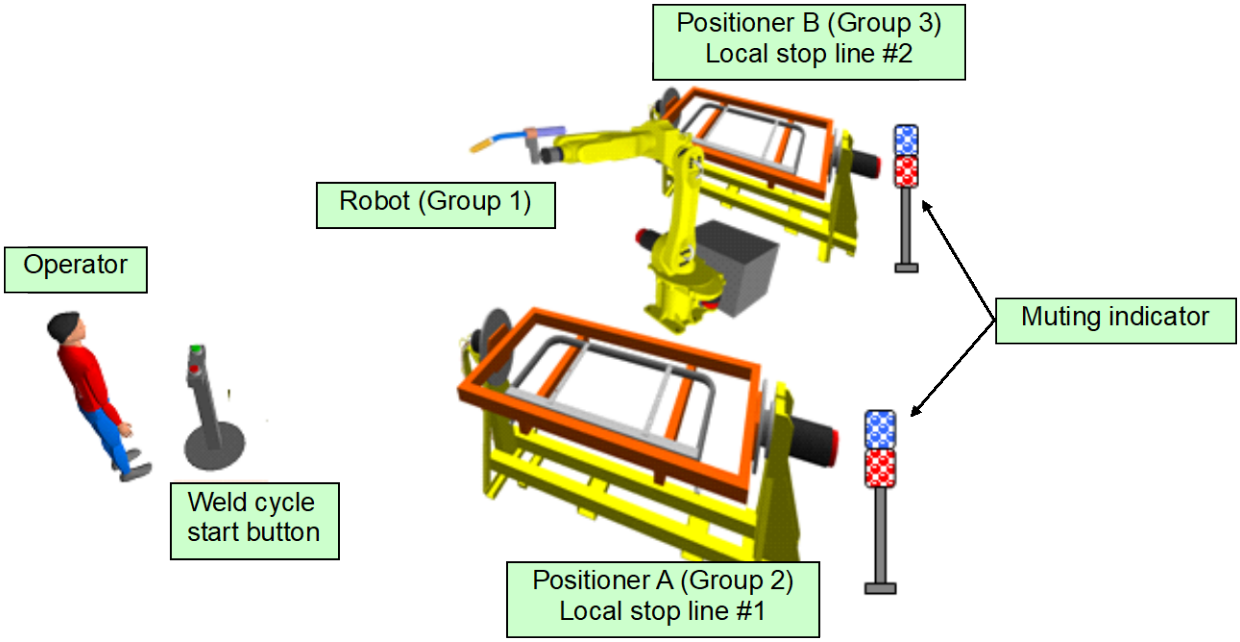
  If you teach motion instruction right after servo ON command without confirming the Local stop status (the output signal #2),
  "LSTP-011 Motion grp $_{xx}$ is in LSTOP" alarm might occur.

- Motion group mask of a TP program in which servo OFF will be commanded.
  Please set a motion group mask to (*,*,*,*,*,*,*,*) for a TP program in which servo OFF will be commanded to a certain auxiliary axis group by RO or DO. (NOTE: Please set a motion group mask at creating a program.)
- When turning on a robot controller, servo ON command will be output once.
  This is due to the initialization of software internal status. If the input signal #2 is ON, the status will be kept (servo ON). If the input signal #2 is OFF, servo OFF command will be output and auxiliary axis will enter into the Local stop mode (servo OFF).

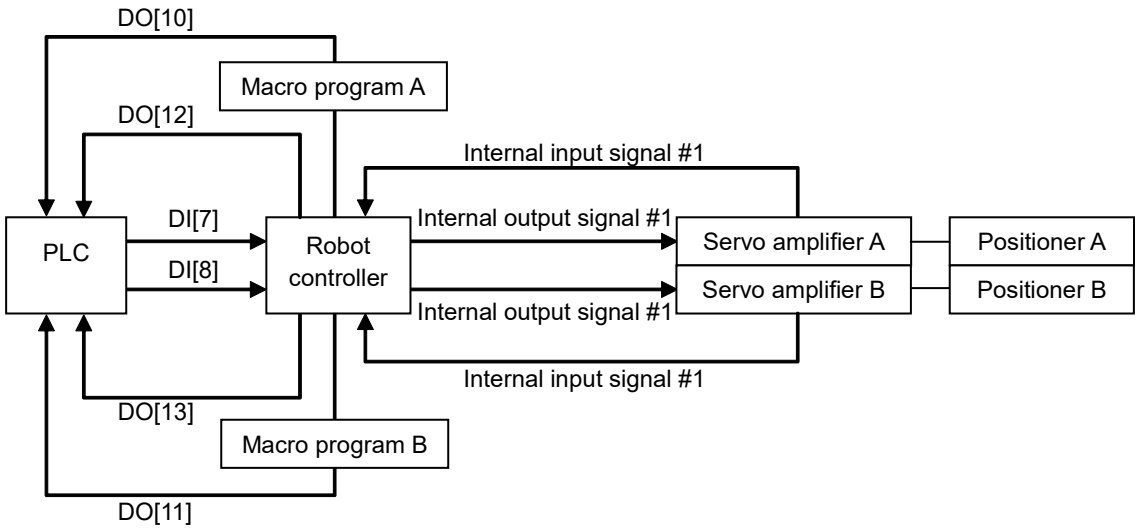
17.5 PROGRAMMING

Suppose a system like below.

- 3 groups welding system that consists of a robot and two positioners (A and B).
- Operator loads/unloads a workpiece at one side of the positioner while robot welds at another positioner.
- 2 Local stop lines are required.



- Assume that I/O is connected like below.



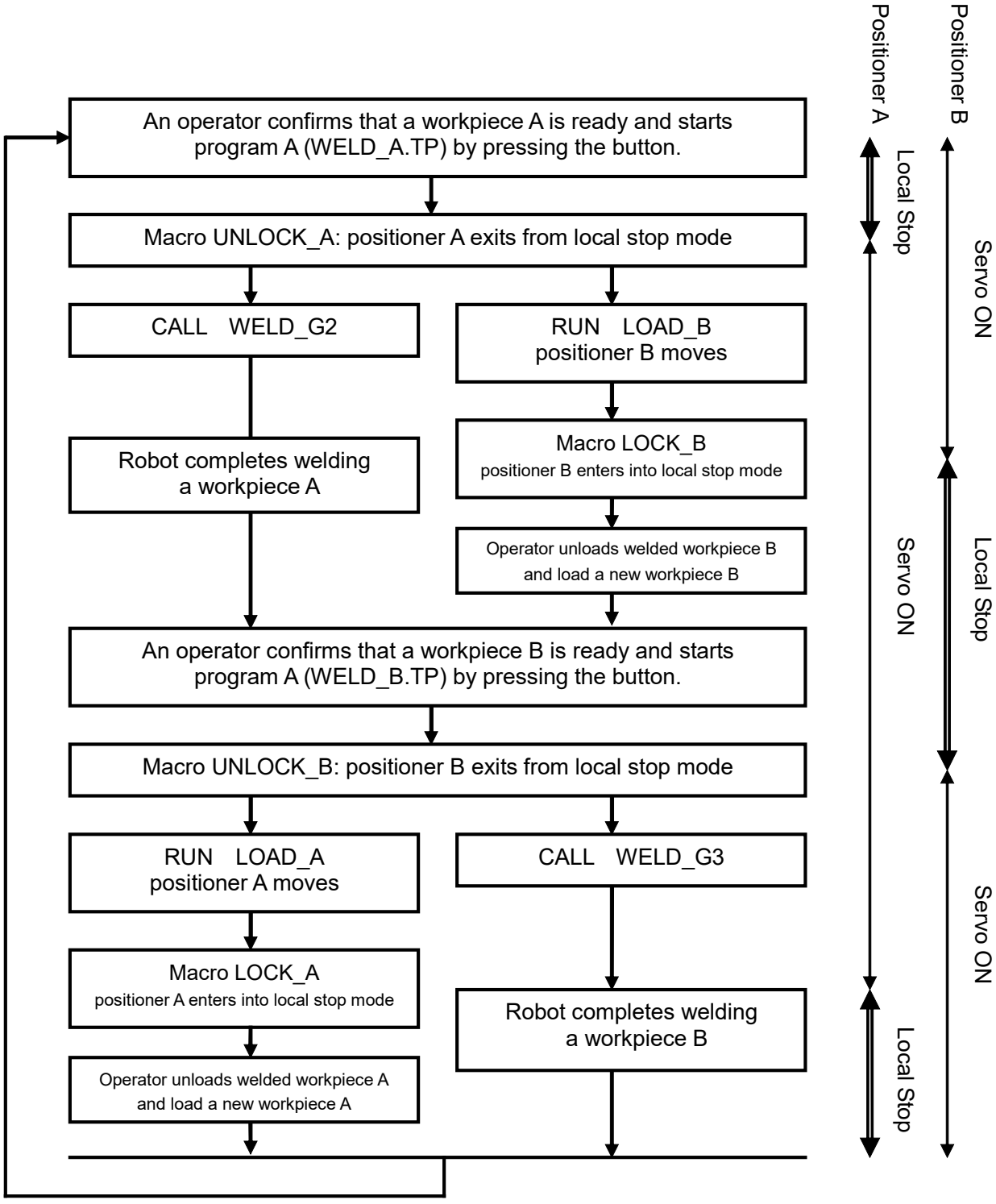
Suppose a welding sequence as follows.

A) Robot completes welding of a workpiece B at positioner B (Group 3) and returns to a home position.
B) An operator confirms that workpiece A of positioner A (Group 2) is ready and starts program A by pressing the cycle start button.
C) WELD_A.TP program starts.
   C.1 In macro UNLOCK_A, positioner A exits from Local stop mode.

C.2 In LOAD_B.TP, positioner B moves to the position for loading/unloading workpiece B.
C.3 In macro LOCK_B, positioner B enters into Local stop mode.
C.4 In WELD_G2.TP, robot starts to weld a workpiece A at positioner A.
C.5 An operator confirms that a muting indicator (This indicator is connected to the output signal #2) indicates positioner B is in Local stop mode. Then an operator unloads welded workpiece B and loads a new workpiece B.

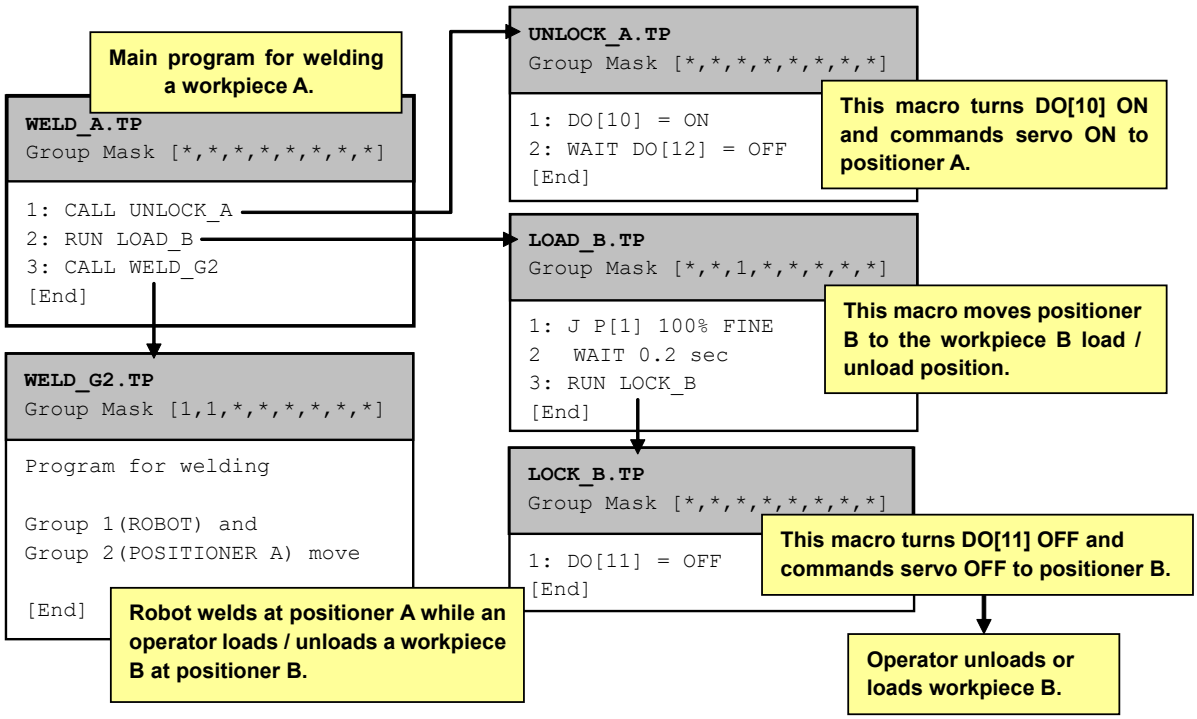
D) Robot completes welding of a workpiece A at positioner A (Group 2) and returns to a home position.
E) An operator confirms that a workpiece B is ready and starts next program B by pressing the button.
F) Repeat C)~D) and then go back to program A.

Cycle flow of the above-mentioned example.

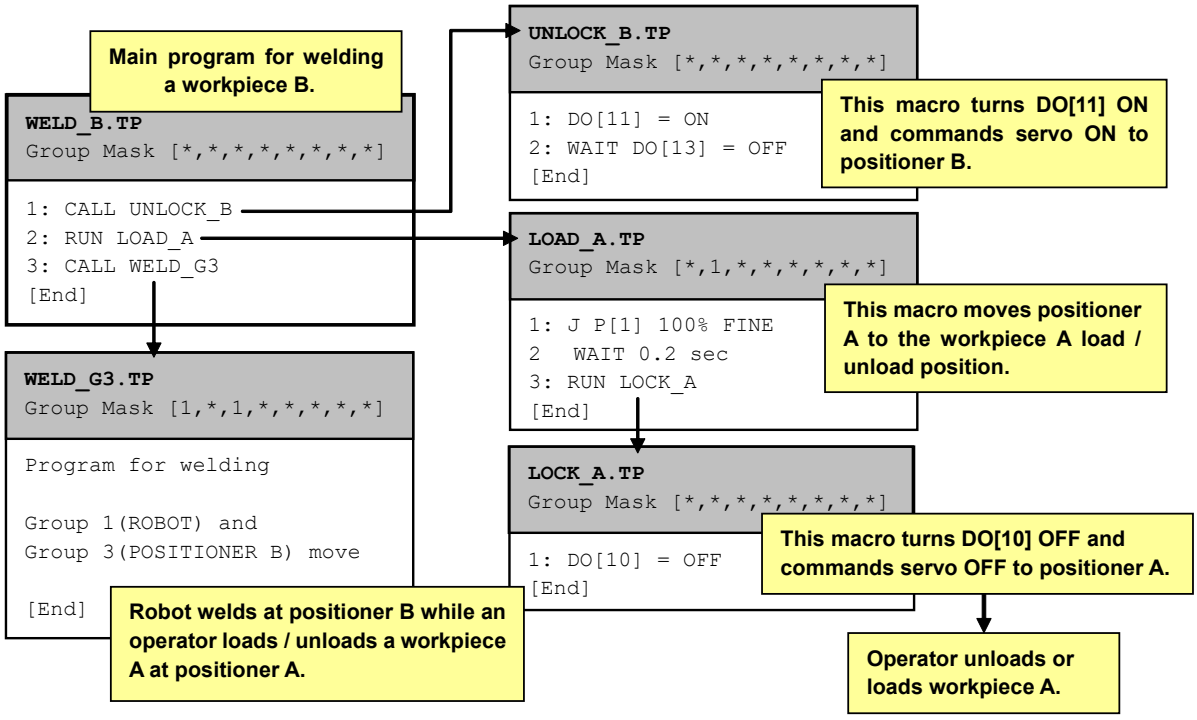


For example, following programs accomplish this sequence.

Program A for welding of a workpiece A : WELD_A

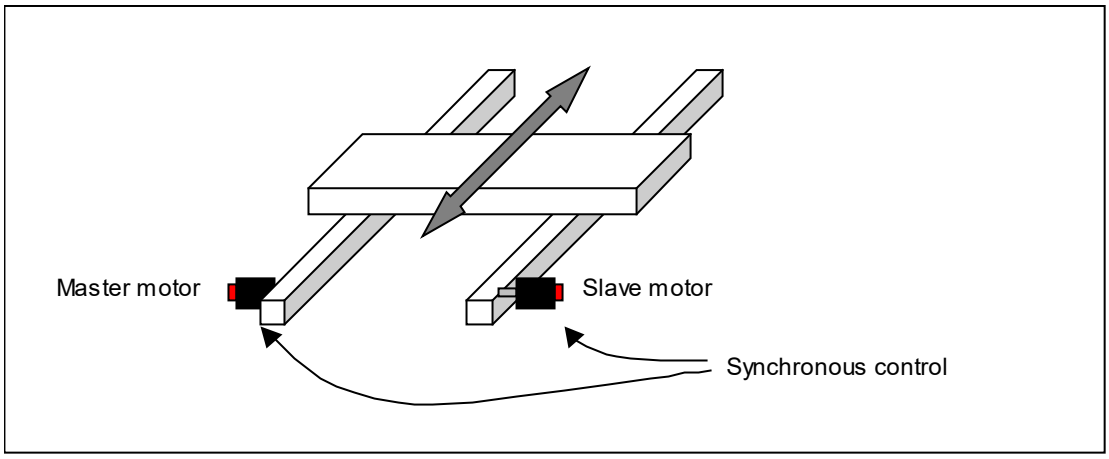


Program B for welding of a workpiece B : WELD_B



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-2600-J836) is required. Extended axis (J518), Independent axis (H895) and Positioner (H874, H896) can be used as Dual Drive.
If the software version is 7DF3/22, 7DF5/24, or later, you can 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.

NOTE
- In the case of Extended axes, you can add 1 pair of Dual Drive motor in one motion group.
- In the case of Independent axis or Positioner axes, you can add up to 2 pairs of Dual Drive motor in one motion group.
- The maximum number of axes per motion group is 9 that includes slave axes of Dual Drives.
- Extended axis is supported at 7DC3 series and later.

⚠ CAUTION
- If the software version is 7DC3/20 or earlier and you use Extended axis as Dual Drive, the motor direction of a slave motor must be same with the master.
- If the software version is 7DC3/21 or later, there is not such a restriction.

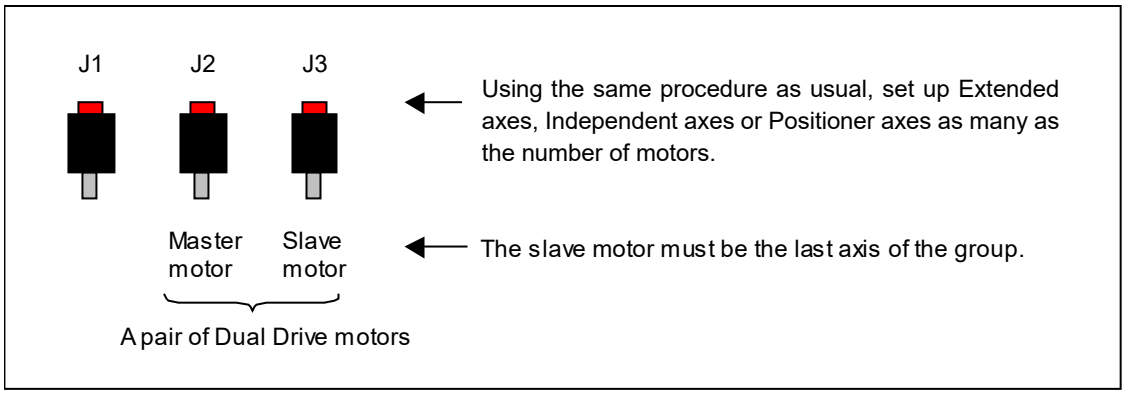
⚠ CAUTION
This function synchronizes based on positions. This function doesn't synchronize based on torques. So This function doesn't support several synchronous motion similar to transmitting power with two motors to one shared gear.

18.1 SETUP

Preparation

Set up Extended axes or Independent axes or Positioner axes which you plan to assign as a pair of Dual Drive motors.

- If the system has multiple operation groups, Dual Drive can be set up for each group.
- The pair of Dual Drive motors must belong to the same motion group.
- In this step, set up both a master motor and a slave motor individually.
- The slave motor must be the last axis in the group. If there exist multiple Dual Drive motor pairs in one group, the slave axes must be the last axes in the group though they can be set up in an arbitrary order.
- Limitation: If the software version is 7DC3/20 or earlier and you use Extended axis as dual drive, the motor direction of a slave motor must be same with the master.



Dual Drive motor axes setup

Set up Dual Drive motor axes at the Controlled Start.
Set the following variables, then select Cold Start.

System Variable	Description
$SCR_GRP[g].$NUM_AXES	This system variable means the total number of axes in the motion group. Here, count a pair of Dual Drive motors as one axis. In other words, change this variable to the original number minus the number of Dual Drive motor pairs. For example, when one group has three motors and one pair of dual drive, change the number from three to two.
$SCR_GRP[g].$NUM_ROB_AXS	Change this variable in the same manner as $NUM_AXES.
$SCR_GRP[g].$NUM_DUAL	Enter the number of Dual Drive motor pairs. The maximum value of this system variable is 3. One pair consists of two motors.
$DUAL_DRIVE[g].$M_AXIS_NUM[n]	Enter the axis number of the master motor. “n” is an index of the Dual Drive motor pair in the motion group. For the first pair, n = 1.
$DUAL_DRIVE[g].$S_AXIS_NUM[n]	Enter the axis number of the slave motor.

[Example 1]

System configuration:
- Group 2 has three motors as Independent Additional Axes.
- J2-J3 of Group 2 is a pair of Dual Drive motors.
- J2 is the master motor and J3 is the slave motor.

Setting:
- $SCR_GRP[2].$NUM_AXES = (3→) 2
- $SCR_GRP[2].$NUM_ROB_AXS = (3→) 2
- $SCR_GRP[2].$NUM_DUAL = 1
- $DUAL_DRIVE[2].$M_AXIS_NUM[1] = 2

- $DUAL_DRIVE[2].$S_AXIS_NUM[1] = 3

When jogging or teaching, the Dual Drive pair is treated as “J2”.

[Example 2]

System configuration:
- Group 2 has four motors as Independent Additional Axes.
- J1-J3 and J2-J4 of Group 2 are pairs of Dual Drive motors.
- J1 and J2 are the master motors and J3 and J4 are the slave motors.

Setting:
- $SCR_GRP[2].$NUM_AXES = (4→) 2
- $SCR_GRP[2].$NUM_ROB_AXS = (4→) 2
- $SCR_GRP[2].$NUM_DUAL = 2
- $DUAL_DRIVE[2].$M_AXIS_NUM[1] = 1
- $DUAL_DRIVE[2].$M_AXIS_NUM[2] = 2
- $DUAL_DRIVE[2].$S_AXIS_NUM[1] = 3
- $DUAL_DRIVE[2].$S_AXIS_NUM[2] = 4

When jogging or teaching, the Dual Drive pairs are treated as “J1” and “J2”.

Dual Drive motor axes reset

Dual Drive motor axis setting can be reset by setting the variables back to the original values.
At the Controlled Start, set the variables back to the original values, then select Cold Start.

[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 the master motor and J3 is the slave motor.

Setting:
- $SCR_GRP[2].$NUM_AXES = (2→) 3
- $SCR_GRP[2].$NUM_ROB_AXS = (2→) 3
- $SCR_GRP[2].$NUM_DUAL = 0
- $DUAL_DRIVE[2].$M_AXIS_NUM[1] = 0
- $DUAL_DRIVE[2].$S_AXIS_NUM[1] = 0

[Example 2]

System configuration:
- Group 2 has four motors as Independent Additional Axes.
- J1-J3 and J2-J4 of Group 2 are pairs of Dual Drive motors.
- J1 and J2 are the master motors and J3 and J4 are the slave motors.

Setting:
- $SCR_GRP[2].$NUM_AXES = (2→) 4
- $SCR_GRP[2].$NUM_ROB_AXS = (2→) 4
- $SCR_GRP[2].$NUM_DUAL = 0
- $DUAL_DRIVE[2].$M_AXIS_NUM[1] = 0
- $DUAL_DRIVE[2].$M_AXIS_NUM[2] = 0
- $DUAL_DRIVE[2].$S_AXIS_NUM[1] = 0
- $DUAL_DRIVE[2].$S_AXIS_NUM[2] = 0

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 software version is 7DF3/22, 7DF5/24 or later, with the Dual Drive option (A06B-2600-J836) 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.

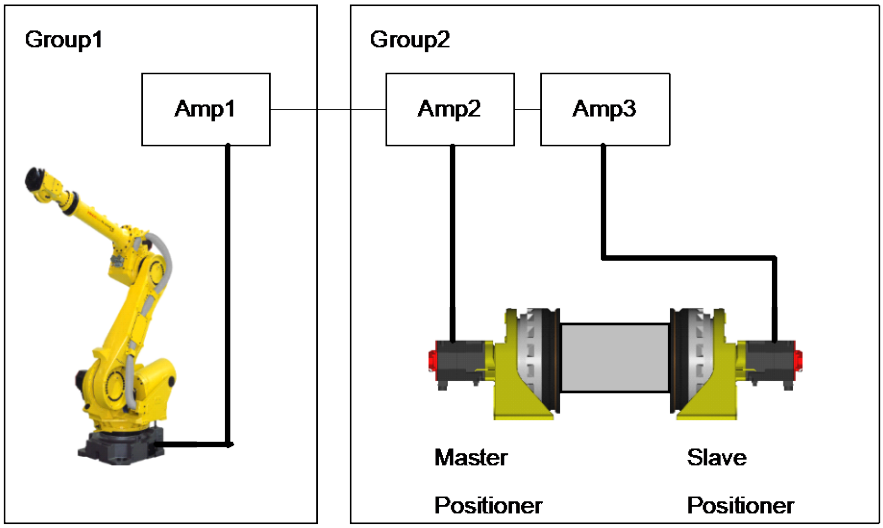
Dual Drive supported 1-axis positioner	Software specification No. (A05B-2600-****)	Mechanical unit specification No.
300kg/500kg payload (Hollow type)	H879 / H875	A05B-1220-J103
1000kg/1500kg payload (Hollow type)	H880 / H876	A05B-1220-J102
1000kg/1500kg payload (Compact type)	H877 / H878	A05B-1220-J104
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

NOTE
- The following abbreviations are used herein.
  300kg/500kg payload (Hollow type)
  : 300kg/500kg payload 1-axis servo positioner (Hollow type)
  1000kg/1500kg payload (Hollow type)
  : 1000kg/1500kg payload 1-axis servo positioner (Hollow type)
  1000kg/1500kg payload (Compact type)
  : 1000kg/1500kg payload 1-axis servo positioner (Compact type)

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 without system variable setting.

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.

```
ROBOT MAINTENANCE                                    2/10
    Setup Robot System Variables

      Group  Robot Library/Option    Ext Aces
        1    R-2000iC/165F              0
        2    1AxisPositioner 1500       0

    [ TYPE ]  ORD NO  AUTO  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-2600-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”.

```
******* Group: 2 Initialization *******
* 1Axis Positioner Hollow Type(1500kg)*

  -- DRIVE TYPE SELECT --

  1: Single Motor Drive
  2: Dual Drive (Use 2 Positioners)

  Select ==>
```

- 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.

```
******* Group: 2 Initialization *******
* 1Axis Positioner Hollow Type(1500kg)*

  -- Brake number setting --
  enter Brake number[ 1] =
  enter Brake number[ 2] =
```

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.

```
******* Group: 2 Initialization *******
* 1Axis Positioner Hollow Type(1500kg)*

  -- Amplifier Number (1~84) setting --
  enter Amp number[ 1] =
  enter Amp number[ 2] =
```

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-6400 series 6 axes amplifier
  2. A06B-6240 series Alpha i amp. or
     A06B-6160 series Beta i 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.

19 MULTI UOP INTERFACE FUNCTION

Multi UOP Interface function enable us to use up to 5 UOP sets that have a connection with start of program. Therefore, It is possible to control (select, execute, pause and so on) multiple program independently with a single robot controller.

Specification

The number of usable UOP set
Up to 5 UOP sets are usable.

UOP Interface
The number of UOP signals depend on how many UOP sets is used.

Number of set	1	2	3	4	5
Input	18	23	32	41	50
Output	20	26	36	46	56

Software Options

The following software option is needed for Multi UOP Interface function.

Multi UOP Interface	A05B-2600-J964

If 2 or more motion groups (multiple robot, independent axis and so on) is used, the following software option is needed.

Multi-Group Motion	A05B-2600-J601

Multi UOP Interface for Dual Arm Control (J605)

Multi UOP interface for Dual Arm Control (J605) is the function that is Multi UOP Interface function but it is limited to use UOP sets up to 2. If more than 2 UOP sets are not needed, we can choose not Multi UOP Interface (J964) but Multi UOP interface for Dual Arm Control (J605).

Multi UOP interface for Dual Arm Control is different from Multi Robot Control Function that had been used since R-30*i*A in the following point:

- One CSTOPI signal was prepared for 2 UOP sets by Multi Robot Control Function, but two CSTOPI signals are prepared for 2 UOP sets by Multi UOP interface for Dual Arm Control. (For this difference, Multi UOP interface for Dual Arm Control uses more one signal than Multi Robot Control Function by CSTOPI signal.)
- In case of Multi Robot Control function, the motion groups intended by each UOP depend on coupling structure of Isolation switch. But in case of Multi UOP Interface function, the user can configure the motion groups intended by each UOP freely.

And it is possible to use UOP as the specification of Multi Robot Control function. Please refer to “19.6 System variables”.

19.1 PERIPHERAL I/O

The following signals are provided in response to the number of UOP set.

INPUT SIGNAL	OUTPUT SIGNAL
<Number of set : 1>	
*IMSTP	CMDENBL#1
*HOLD#1	SYSRDY#1
*SFSPD	PROGRUN#1
CSTOPI#1	PAUSED#1
FAULT_RESET	HELD#1
START#1	FAULT#1
HOME	ATPERCH
ENBL	TPENBL
RSR1/PNS1/STYLE1	BATALM
RSR2/PNS2/STYLE2	BUSY
RSR3/PNS3/STYLE3	ACK1/SNO1
RSR4/PNS4/STYLE4	ACK2/SNO2
RSR5/PNS5/STYLE5	ACK1/SNO1
RSR6/PNS6/STYLE6	ACK1/SNO1
RSR7/PNS7/STYLE7	ACK1/SNO1
RSR8/PNS8/STYLE8	ACK1/SNO1
PNSTROBE#1	ACK1/SNO1
PROD_START#1	ACK1/SNO1
	SNACK
	RESERVE
Total 18 points	Total 20 points
<Number of set : 2>	
*HOLD#2	CMDENBL#2
START#2	SYSRDY#2
PNSTROBE#2	HELD#2
PROD_START#2	FAULT#2
CSTOPI#2	PROGRUN#2
	PAUSED#2
Total 23 points	Total 26 points
<Number of set : 3>	
RSR9	ACK9
RSR10	ACK10
RSR11	ACK11
RSR12	ACK12
*HOLD#3	CMDENBL#3
START#3	SYSRDY#3
PNSTROBE#3	HELD#3
PROD_START#3	FAULT#3
CSTOPI#3	PROGRUN#3
	PAUSED#3
Total 32 points	Total 36 points

<Number of set : 4>	
RSR13	ACK13
RSR14	ACK14
RSR15	ACK15
RSR16	ACK16
*HOLD#4	CMDENBL#4
START#4	SYSRDY#4
PNSTROBE#4	HELD#4
PROD_START#4	FAULT#4
CSTOPI#4	PROGRUN#4
	PAUSED#4
Total 41 points	Total 46 points
<Number of set : 5>	
RSR17	ACK17
RSR18	ACK18
RSR19	ACK19
RSR20	ACK20
*HOLD#5	CMDENBL#5
START#5	SYSRDY#5
PNSTROBE#5	HELD#5
PROD_START#5	FAULT#5
CSTOPI#5	PROGRUN#5
	PAUSED#5
Total 50 points	Total 56 points

I/O Configuration:

In case that DI/DO are 40 points

Number of set	1	2	3	4	5
Specialized input（UI）	18	23	32	-	-
General-purpose input（DI）	22	17	8	-	-
Specialized output（UO）	20	26	36	-	-
General-purpose output（DO）	20	14	4	-	-

In case that DI/DO are 96 points

Number of set	1	2	3	4	5
Specialized input（UI）	18	23	32	41	50
General-purpose input（DI）	78	73	64	55	46
Specialized output（UO）	20	26	36	46	56
General-purpose output（DO）	76	70	60	50	40

19.1.1 Setting of the Number of UOP Set

The number of UOP set is 1 by default. Please change $MULTI_ROBO.$NUM_PROG from 1 to the number you want to use when you want to change the number of UOP set. You can change the value up to 5. And the number of UOP set is changed after you cycle power.

Then the number of UI/UO signal is changed, but signal assignment isn't changed. If you clear assignment, new assignment in response to new number of UI/UO signal is given automatically (You can clear assignment by selecting F5, CLR_ASG in I/O Link Device screen. Power cycling is needed after that. And all assignments are cleared. If this is inconvenient for you, please change assignment manually).

Here from, 'N' means the number of UOP set, in other words, the value of $MULTI_ROBO.$NUM_PROG ('N' is used for explanation after "19.1.3 HOLD#1 to #N").

19.1.2 Modification and Addition of Signal

In Multi UOP Interface Function, the meaning of the following signals are changed.

Input:	HOLD#1:	UI[2] HOLD is changed to HOLD#1.
	CSTOPI#1:	UI[4] CSTOPI is changed to CSTOPI#1.
	START#1:	UI[6] START is changed to START#1.
	PNSTROBE#1:	UI[17] PNSTROBE is changed to PNSTROBE#1.
	PROD_START#1:	UI[18] PROD_START is changed to PROD_START#1.
Output:	CMDENBL#1:	UO[1] CMDENBL is changed to CMDENBL#1.
	SYSRDY#1:	UO[2] SYSRDY is changed to SYSRDY#1.
	PROGRUN#1:	UO[3] PROGRUN is changed to PROGRUN#1.
	PAUSED#1:	UO[4] PAUSED is changed to PAUSED#1.
	HELD#1:	UO[5] HELD is changed to HELD#1.
	FAULT#1:	UO[6] FAULT is changed to FAULT#1.

In Multi UOP Interface Function, the following signals are added.

<Number of set : 2>

	Specialized I/O signal	UI/UO[*]
Input	HOLD#2	UI[19]
	START#2	UI[20]
	PNSTROBE#2	UI[21]
	PROD_START#2	UI[22]
	CSTOPI#2	UI[23]
Output	CMDENBL#2	UO[21]
	SYSRDY#2	UO[22]
	HELD#2	UO[23]
	FAULT#2	UO[24]
	PROGRUN#2	UO[25]
	PAUSED#2	UO[26]

<Number of set : 3>

	Specialized I/O signal	UI/UO[*]
Input	RSR9	UI[24]
	RSR10	UI[25]
	RSR11	UI[26]
	RSR12	UI[27]
	HOLD#3	UI[28]
	START#3	UI[29]
	PNSTROBE#3	UI[30]
	PROD_START#3	UI[31]
	CSTOPI#3	UI[32]
Output	ACK9	UO[27]
	ACK10	UO[28]
	ACK11	UO[29]
	ACK12	UO[30]
	CMDENBL#3	UO[31]
	SYSRDY#3	UO[32]
	HELD#3	UO[33]
	FAULT#3	UO[34]
	PROGRUN#3	UO[35]
	PAUSED#3	UO[36]

<Number of set : 4>

	Specialized I/O signal	UI/UO[*]
Input	RSR13	UI[33]
	RSR14	UI[34]
	RSR15	UI[35]
	RSR16	UI[36]
	HOLD#4	UI[37]
	START#4	UI[38]
	PNSTROBE#4	UI[39]
	PROD_START#4	UI[40]
	CSTOPI#4	UI[41]
Output	ACK13	UO[37]
	ACK14	UO[38]
	ACK15	UO[39]
	ACK16	UO[40]
	CMDENBL#4	UO[41]
	SYSRDY#4	UO[42]
	HELD#4	UO[43]
	FAULT#4	UO[44]
	PROGRUN#4	UO[45]
	PAUSED#4	UO[46]

<Number of set : 5>

	Specialized I/O signal	UI/UO[*]
Input	RSR17	UI[42]
	RSR18	UI[43]
	RSR19	UI[44]
	RSR20	UI[45]
	HOLD#5	UI[46]
	START#5	UI[47]
	PNSTROBE#5	UI[48]
	PROD_START#5	UI[49]
	CSTOPI#5	UI[50]
Output	ACK17	UO[47]
	ACK18	UO[48]
	ACK19	UO[49]
	ACK20	UO[50]
	CMDENBL#5	UO[51]
	SYSRDY#5	UO[52]
	HELD#5	UO[53]
	FAULT#5	UO[54]
	PROGRUN#5	UO[55]
	PAUSED#5	UO[56]

19.1.3 HOLD#1 to #N

Here from, the word “Program#1” appears. “Program#1” means the program selected by UOP set 1. “Program#2” to “Program#N” are same as “Program#1”. Please refer to “19.2 SELECT PROGRAM”.

Multi UOP Interface has one HOLD signal per UOP set (#1 to #N). The function of HOLD signal is changed by setting system variable “$MULTI_ROBO.$HOLD_TYPE.

- In case of $MULTI_ROBO.$HOLD_TYPE = 0 （Default）
  If whichever of HOLD signals (HOLD#1 to #N) is turned OFF, all the executing programs are paused.

- In case of $MULTI_ROBO.$HOLD_TYPE = 1
  If HOLD#X is turned OFF, “Program#X' is paused (X = 1 to N).
  The running program which is not selected as “Program#X”, can NOT be held. Please turn off ENBL to pause the program.
  (However, if “Program#X” is not selected, when HOLD#X is OFF, all the programs in execution are paused.)

But, when a program is tried to execute, regardless of $MULTI_ROBO.$HOLD_TYPE value, if any HOLD signal is OFF, it is impossible to execute program. Then if program execution is needed, please turn ON all HOLD signals.

19.1.4 CSTOPI#1 to #N

Multi UOP Interface has one CSTOPI signal per UOP set (#1 to #N). The function of CSTOPI signal is changed by setting "CSTOPI for ABORT" or "Abort all programs by CSTOPI" in System Config screen.

- When "Abort all programs by CSTOPI" is FALSE (Default)
  If CSOTPI#X is turned ON, "Program#X" is aborted (X = 1 to N).
  It also release (clear) programs in UOP set X queue from the wait state by RSR.
  Please refer to "19.2.6 RSR".

- When "Abort all programs by CSTOPI" is TRUE
  If whichever of CSTOPI signals (HOLD#1 to #N) is turned ON, all the executing programs are aborted. It also release (clear) programs in all UOP set queue from the wait state by RSR.

TRUE for "Abort all programs by CSTOPI" is recommended for Multi UOP Interface function.

The way of terminating program is changed by setting "CSTOPI for ABORT" in System Config screen.

- When "CSTOPI for ABORT" is FALSE (Default)
  If CSOTPI signal is turned ON, this signal aborts the program in execution as soon as execution of the program completes.

- When "CSTOPI for ABORT" is TRUE
  If CSOTPI signal is turned ON, this signal immediately aborts the program in execution.

19.1.5 START#1 to #N

NOTE
In the system with Multi UOP Interface, multiple programs are controlled at the same time, and it is more complex to control program than control single program. So it is recommended that the Multi UOP Interface is in condition that "START for CONTINUE only:" in system config screen is "TRUE". Only paused program can be started by START signal under this setting. In case of starting the ended program, use RSR or PROD_START signal, please.

Multi UOP Interface has one START signal per UOP set (#1 to #N).

"Program#X" is started when START#X is turned off (X = 1 to N).

The program is continued to execute from paused line in case that START signal is input with paused program.

When "START for CONTINUE only:" in system config screen is "TRUE", the program, which is aborted (neither running nor paused), can NOT be started and the message "PROG-023 Task is not paused" is displayed in TP screen in case that START signal is inputted with the ended program (neither running nor paused).

19.1.6 RSR

Multi UOP Interface has four RSR signal per UOP set (#1 to #N).
(However, if the number of UOP set is 1, eight RSR is usable.)

Number of set	RSR1 to 4	RSR5 to 8	RSR9 to 12	RSR13 to 16	RSR17 to 20
1	#1		-	-	-
2	#1	#2	-	-	-
3	#1	#2	#3	-	-
4	#1	#2	#3	#4	-
5	#1	#2	#3	#4	#5

And each UOP set has RSR queue, so the number of RSR queue is N.
Please refer to “19.4.1 Robot Service Request (RSR)”.

19.1.7 PNSTROBE#1 to #N,PROD_START#1 to #N

Multi UOP Interface has one PNSTROBE and one PROD_START signal per UOP set (#1 to #N).
Please refer to “19.3.2 Program Number Select (PNS)”.

19.1.8 CMDENBL#1 to #N

Multi UOP Interface has one CMDENBL signal per UOP set (#1 to #N).

CMDENBL#1 is ON if all the following conditions are satisfied.

- Remote condition is satisfied.
- No alarm. (Every FAULT#X are OFF. (X = 1 to N))
- No single step.
- If Isolation switch is used, motion groups that is intended by UOP set 1 are connected.
  (“Remote condition” and “motion groups that is intended by UOP set 1”are mentioned after.)

In case of CMDENBLE#2 to #N, only last condition is different.

- If Isolation switch is used, motion groups that is intended by UOP set 2 (to N) are connected.

1. Remote Condition:
   - The Teach Pendant enable switch is set OFF.
   - Setting of Remote/Local is set to Remote.
   - The *SFSPD input is ON.
   - The ENBL input is ON.
   - The system variable $RMT_MASTER is 0 (UOP).

2. Motion groups that is intended by UOP set
   Motion groups that is intended by UOP set is changed by setting system variable
   “$MULTI_ROBO.$MULTI_PROGC”.
   Please refer to “19.2.1 Setting of Motion Group Intended by Each UOP Set”.

19.1.9 SYSRDY#1 to #N

Multi UOP Interface has one SYSRDY signal per UOP set (#1 to #N).

SYSRDY#1 is ON if all the following conditions are satisfied.

- All the axes are ready. (If system has Isolation switch, all the axes except for isolated are ready.)

- If Isolation switch is used, motion groups that is intended by UOP set 1 are connected.

In case of SYSRDY#2 to #N, only last condition is different.
- If Isolation switch is used, motion groups that is intended by UOP set 2 (to N) are connected.

19.1.10 PROGRUN#1 to #N,PAUSED#1 to #N

Multi UOP Interface has one PROGRUN and one PAUSED signal per UOP set (#1 to #N).
PROGRUN and PAUSED signals show the program execution status as the following.

- When program is running:
  PROGRUN#X (X = 1 to N)
  ・ In case that "Program#X" is in execution.

  In case that the program, which is not "Program#X", is in execution, every PROGRUN#X is output. (Default)

  If the value of the system variable $MULTI_ROBO.$UNSELOUTPUT is 0, the behavior of signals is changed as follows.
  In case that the program, which is not "Program#X", is in execution, PROGRUN#X is not output. But BUSY is output.

- When program is paused:
  PAUSED#X (X = 1 to N)
  ・ In case that "Program#X" is paused.

  In case that the program, which is not "Program#X", is paused, every PAUSED#X is output. (Default)

  If the value of the system variable $MULTI_ROBO.$UNSELOUTPUT is 0, the behavior of signals is changed as follows.
  In case that the program, which is not "Program#X", is paused, PAUSED#X is not output.

19.1.11 HELD#1 to #N

Multi UOP Interface has one HELD signal per UOP set (#1 to #N).

HELD#X is ON if one of the following conditions is satisfied. (X = 1 to N)
- HOLD#X is OFF.
- The HOLD switch on the Teach Pendant is pushed.
- The HOLD switch on the Operator Panel is pushed.

19.1.12 FAULT#1 to #N

Multi UOP Interface has one FAULT signal per UOP set (#1 to #N).

FAULT#X turns ON when the alarm occurs by "Program#X" execution or other than program execution. (X = 1 to N)

When system is in alarm status, at least one of FAULT#X must be ON.

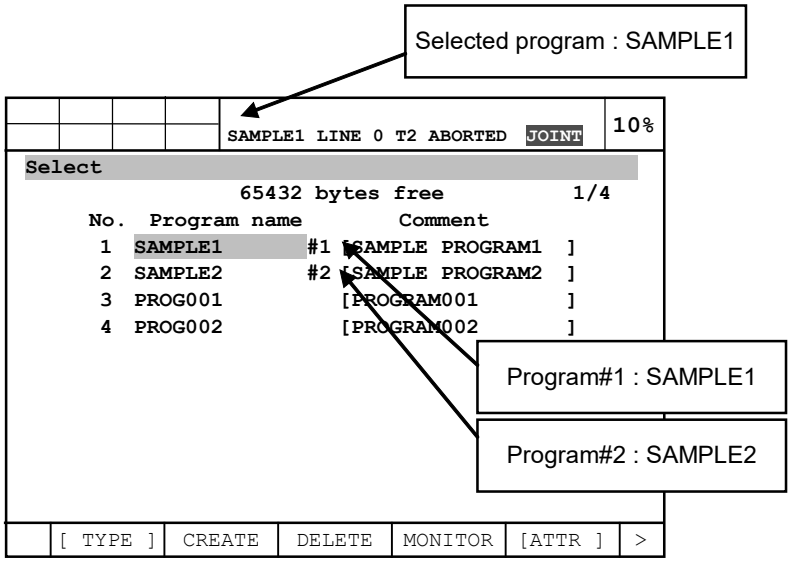
19.1.13 ATPERCH

The specified signal ATPERCH is set to ON when the robot of motion group 1 is at the reference position 1.
Please assign reference position output except motion group 1.

19.2 SELECT PROGRAM

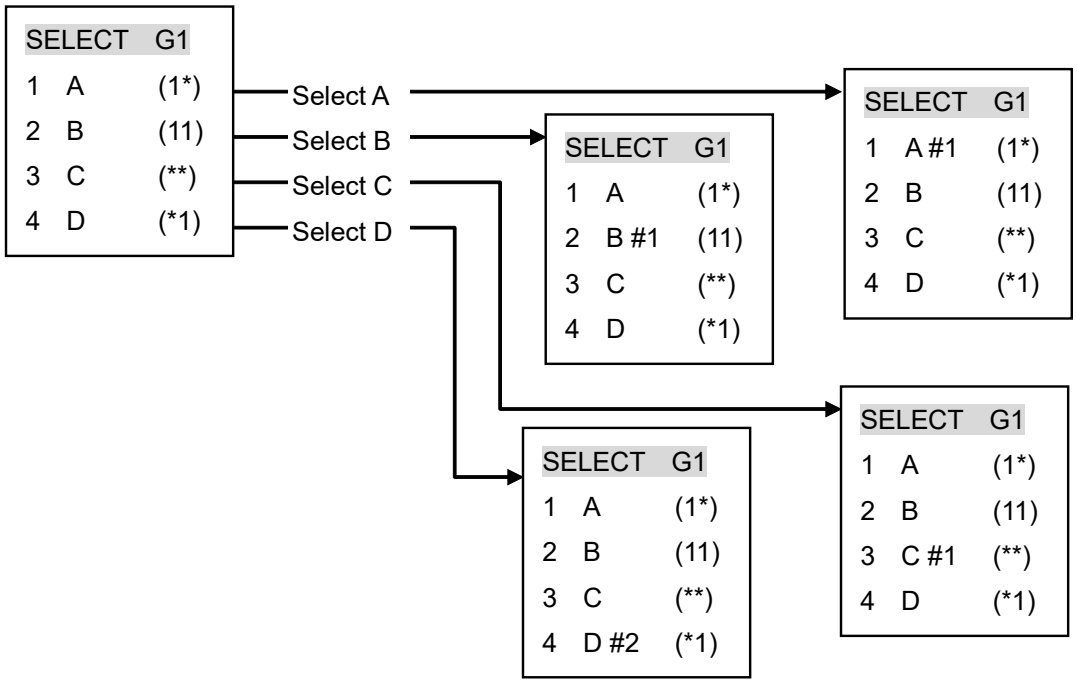
In Multi UOP Interface function, it is possible for each UOP set to select program individually.
One UOP set is given one program # for selected program.

Here from, “Program#1” means the program selected by UOP set 1.
“Program#1” is run by UOP START#1, PROD_STAR#1.
On program selection screen, “#1” is displayed to the right of program name that is used as “Program #1”.
“Program#2” to “Program#N” are same as “Program#1”.
On the contrary, a program that is displayed on program edit screen is referred as “selected program”.
[FWD]/[BWD] keys on teach pendant or start button on Operator Panel run “selected program”.
“Selected program” is displayed on left side of second line of TP screen.



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 variable "$MULTI_ROBO.$MULTI_PROGC.

$MULTI_PROGC has 5 alignments, and each one corresponds one UOP set. If the number of UOP set is smaller than 5, $MULTI_PROGC alignments are available as many as the number of UOP set, and the other alignments is ineffective.

UOP set 1: $MULTI_PROGC[1]
UOP set 2: $MULTI_PROGC[2]
UOP set 3: $MULTI_PROGC[3]
UOP set 4: $MULTI_PROGC[4]
UOP set 5: $MULTI_PROGC[5]

The each bit of $MULTI_PROGC value is equivalent to each motion group.

Bit31	・・・	Bit1	Bit0
Motion Group32	・・・	Motion Group2	Motion Group1

For example, the default values of $MULTI_PROGC are set as the following, it is assumed that one UOP set is equivalent to one motion group.

Example1 (Default)

```
UOP set 1: $MULTI_PROGC[1] = 1
UOP set 2: $MULTI_PROGC[2] = 2
UOP set 3: $MULTI_PROGC[3] = 4
UOP set 4: $MULTI_PROGC[4] = 8
UOP set 5: $MULTI_PROGC[5] = 16
```

In case that PROGRAM1 is equivalent to motion group 1,4,
PROGRAM2 is equivalent to motion group 2,5,
and PROGRAM3 is equivalent to motion group 3,
$MULTI_PROGC value is the following.

Example 2

```
UOP set 1: $MULTI_PROGC[1] = 9
UOP set 2: $MULTI_PROGC[2] = 18
UOP set 3: $MULTI_PROGC[3] = 4
UOP set 4: $MULTI_PROGC[4] = 0
UOP set 5: $MULTI_PROGC[5] = 0
```

Every UOP set can select a program having no motion group. However, when a program having no motion group is selected by TP, if $MULTI_PROGC[#x] is 0, the program become “Program#X” by priority.

Detailed explanation is mentioned in “19.6 SYSTEM VARIABLES”. Please refer it.

19.2.2 Select Program at Running or Paused

When “Program#1” is paused, if you try to select another program as “Program#1”:

- When program is selected by UOP, you can not select the program, and the message “TPIF-013 Other program is running” is displayed.
- When program is selected by Teach Pendant and Teach Pendant Enable switch is OFF, you can not select the program, and the message “TPIF-013 Other program is running” is displayed.
- When program is selected by Teach Pendant and Teach Pendant Enable switch is ON, the original “Program#1” is aborted, and the selected program becomes “Program#1”.(This is the case that “Multi Program Selection” is disabled. If “Multi Program Selection” is enabled, “Program#1” doesn’t change.)

When “Program#1” is running, if you select another program as “Program#1”, you can not select the program regardless of selection way, and the message “TPIF-013 Other program is running” is displayed.

“Program#2” to “Program#N” are same as “Program#1”.

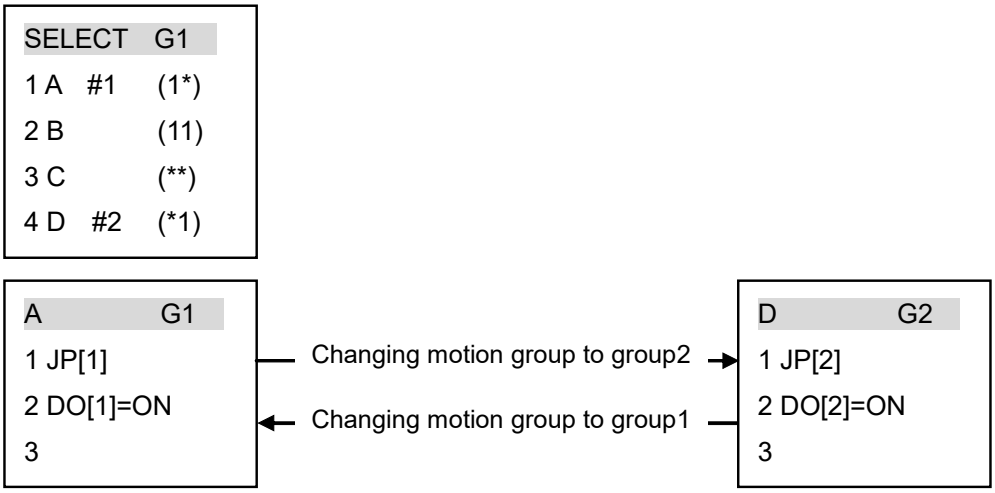
19.2.3 Program Edit Screen Display When Operating Motion Group is Changed

When the user teach motions to each motion group, the user needs to operate to select motion group that can be operated by teach pendant.
It is possible to select motion group to select “CHANGE GROUP” in function menu.

Currently selected motion group is displayed on teach pendant like “G1”,”G2”.

By changing motion group, “selected program” is also changed according to selected motion group. Changed ”selected program” is displayed on program edit screen.

For example, UOP set 1 intends Motion group 1, and UOP set 2 intends Motion group 2.
Suppose “Program #1 “ has only group1 and “Program #2 has” only group2.
By Selecting group1, “Program #1”
By Selecting group2, “Program #2”
automatically becomes “selected program” and the program is displayed on program edit screen.



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 VARIABLES

Name	$MULTI_ROBO.$MLT_ENABLE
Type	BOOLEAN
Protection	Protected
Range	TRUE/FALSE
Meaning	Show whether Multi UOP Interface function is enabled or not. TRUE: Multi UOP Interface function is enabled. FALSE: Multi UOP Interface function is disabled.

Name	$MULTI_ROBO.$HOLD_TYPE
Default	0
Type	INTEGER
Protection	Not protected
Range	0/1
Meaning	Select the function of HOLD#1 to #N. 0: All programs are paused when one of the HOLD#1 to #N signals turns OFF. 1: "Program#X" is paused when HOLD#X turns OFF.

Name	$MULTI_ROBO.$NUM_PROG
Default	1
Type	INTEGER
Protection	Not protected
Range	1 to 5
Meaning	The number of UOP set to use. Set this value to 1 if you want to disable Multi UOP Interface function.

Name	$MULTI_ROBO.$MULTI_PROGC[5]
Default	[1]1, [2]2, [3]4, [4]8, [5]16
Type	INTEGER
Protection	Not protected
Range	0 to 0xFFFF
Meaning	Bit mask for intended motion group of each UOP set. In the default, UOP set 1 intends Motion group 1, UOP set 2 intends Motion group 2, and so on.

[Supplement] Basis of selection for program #

We can select program by UOP or TP, and program number is set on the basis of $MULTI_PROGC. Program number setting rule is the following:

< Common rule for UOP and TP >

Rule 1) If motion group of $MULTI_PROGC[#n] and motion group of program match at least one group, the program can be set #n.

Rule 2) No motion group program can be set ALL program number.

Rule 3) If motion group of every $MULTI_PROGC[#n] and motion group of program match none, the program can be set ALL program number.

Example $MULTI_PROGC[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.

$MULTI_PROGC[1] = 1 >> match at least one group to PROGRAM A, C, D
$MULTI_PROGC[2] = 2 >> match at least one group to PROGRAM B, C, F
$MULTI_PROGC[3] = 3 >> match at least one group to PROGRAM A, B, C, D, F
$MULTI_PROGC[4] = 4 >> match at least one group to PROGRAM D
$MULTI_PROGC[5] = 0 >> match at least one group to none
(by Rule 1)

$MULTI_PROGC[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 $MULTI_PROGC[#x] is equal to motion group of program. (If PROGRAM C is selected by TP, set #3 not #1 or #2. If PROGRAM E is selected by TP, set #5 not #1, #2, #3 or #4.)

[Priority 2] Motion group of $MULTI_PROGC[#x] inhere motion group of program. (If PROGRAM B is selected by TP, set #2 not #3.)

[Priority 3] Motion group of $MULTI_PROGC[#x] and motion group of program match at least one group.

[Priority 4] If program isn't along the above priority from 1 to 3, the program is set to smallest number X that "Program #X" is aborted or is not selected. (A program along Rule 2 or 3 is treated here unless along priority 1.)

[Priority 5] #X is the number that "Program#X" is paused. On that basis, the smallest number that complies with priority 3. (Priority 5 is applied when TP is enabled and "Multi Program Selection" is disabled. Then paused program is aborted.)

When none of the above Priorities is corresponded, it means that all "Program#X" that #X complies with Priorities from 1 to 4 are in execution. Then warning message is displayed and # isn't set to selected program.

Name Default Type Protection Range	$MULTI_ROBO.$TRAD_MODE FALSE BOOLEAN Not protected TRUE/FALSE
Meaning	Compatible flag for Multi Robot Control Function used in the old robot controller. TRUE: Compatible mode - There is single CSTOPI signal. - $MULTI_ROBO.$NUM_PROG is 2. - $ROBOT_ISOLC plays role of $MULTI_ROBO.$MULTI_PROGC. FALSE: Normal Multi UOP Interface mode

Name Default Type Protection Range	$MULTI_ROBO.$UNSELOUTPUT 1 INTEGER Not protected 0/1
Meaning	Output form of PROGRUN#1 to #N signals at the execution of an unselected program and PAUSE#1 to #N signals at the pause of an unselected program. 0: When an unselected program is executed, PROGRUN#1 to #N signals are not output, but BUSY is output. When an unselected program is paused, PAUSE #1 to #N signals are not output. 1: When an unselected program is executed, all PROGRUN#1 to #N signals are output. When an unselected program is paused, all PAUSE #1 to #N signals are output.

20 ERROR CODE OUTPUT FUNCTION

To use this function, Error Code Output Function Option (J527) is required.
This function outputs the contents of the alarm that occurred in the robot as a digital output signal code with strobe.

20.1 SPECIFICATION

20.1.1 Types of Alarms

All alarms except ones whose severity WARN are the targets of external output.

20.1.2 Input and Output Signals

33 points of DO are used as output signals of external error code, and 1 point of DI is used as input signal.

Set the signals used in this function to the following system variables.

$ER_OUT_PUT.$OUT_NUM : **Output signal number**
$ER_OUT_PUT.$IN_NUM : **Input signal number**

- Output signals
  $ER_OUT_PUT.$OUT_NUM determines which DO is used.
  33 points of DO from setup number are used.
  When setup number is 0, this function is disabled.

- Types of output signal
  Output of both code signals and strobe are kept on until reset signal is inputted.

Table 20.1.2 Meanings of DO (in case of $ER_OUT_PUT.$OUT_NUM=1)

DO	Types of alarms
1 to 32	Output alarms as binary (32 bit).
33	Used as strobe.

- Input signal (1bit)
  When there are multiple errors, input signal is used as search signal.
  $ER_OUT_PUT.$IN_NUM determines which DI is used.

- Time chart of signals
  When an alarm occurs, the time chart is as follows.

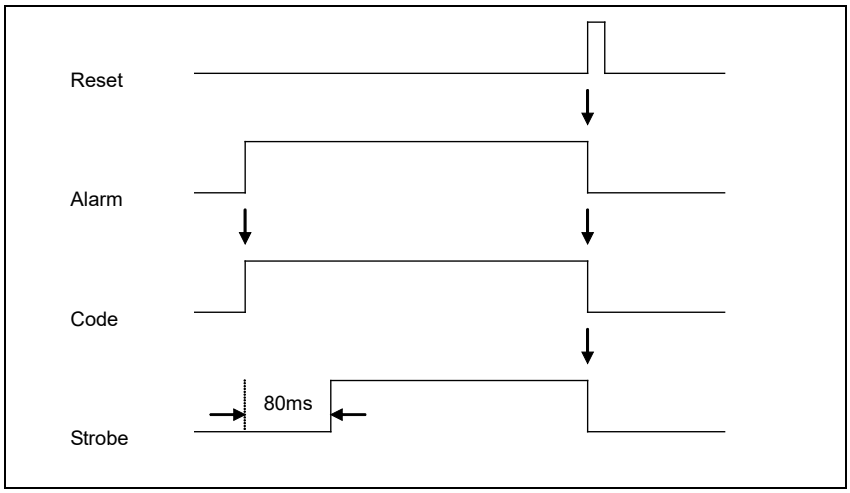


Fig. 20.1.2 (a) Time chart of an alarm

When multiple alarms occur at a time, the initial alarm is outputted at first. Then, the next alarm is output every time search signal is input. These are cyclically output and after that reset signal with strobe is output. Up to 100 alarms can be stored.

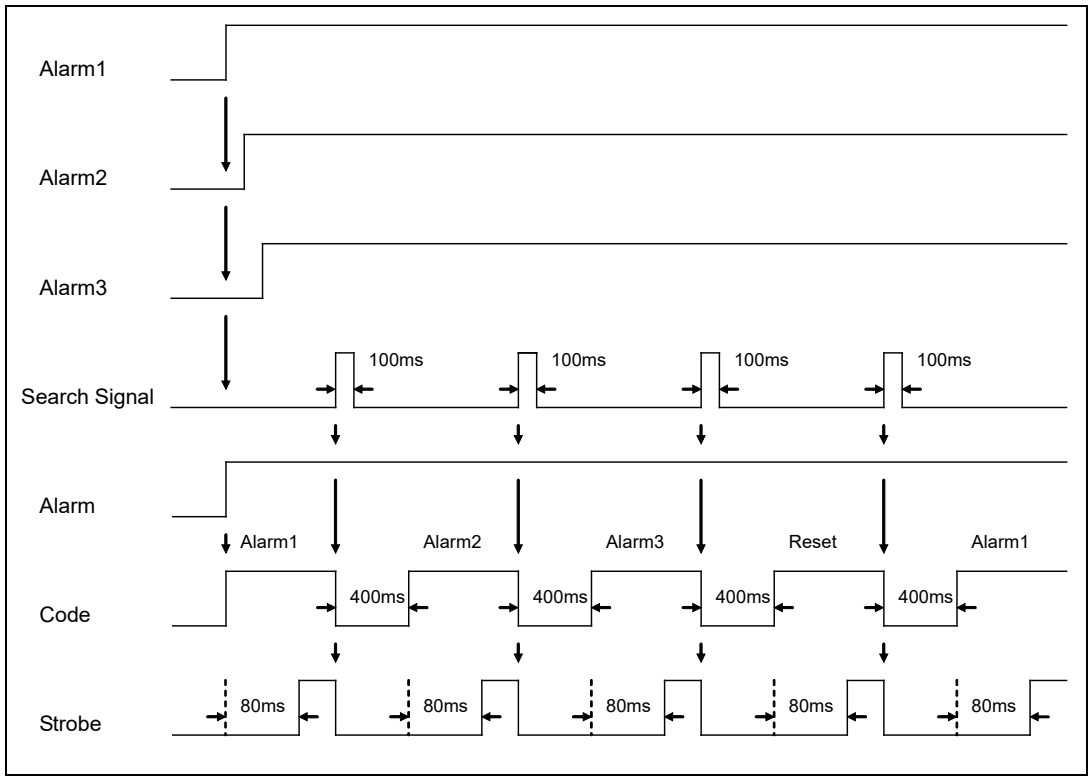


Fig. 20.1.2 (b) Time chart of multiple alarms

20.2 MEANING OF ALARM CODE

In case of $ER_OUT_PUT.$OUT_NUM = 1

DO	Types of alarms
1 to 16	Output an alarm number as binary (16bit).
17 to 24	Output an alarm ID as binary (8bit).
25 to 32	Output an alarm Severity as binary (8bit).
33	Used as strobe.

20.2.1 Severity of Alarm

DO / Severity	25	26	27	28	29	30	31	32
NONE	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON
WARN	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF

Errors that have the severity above will not be outputted.

DO / Severity	25	26	27	28	29	30	31	32
PAUSE.L	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF
PAUSE.G	OFF	ON	OFF	OFF	OFF	ON	OFF	OFF
STOP.L	OFF	ON	ON	OFF	OFF	OFF	OFF	OFF
STOP.G	OFF	ON	ON	OFF	OFF	ON	OFF	OFF
SERVO	OFF	ON	ON	OFF	ON	ON	OFF	OFF
ABORT.L	ON	ON	OFF	ON	OFF	OFF	OFF	OFF
ABORT.G	ON	ON	OFF	ON	OFF	ON	OFF	OFF
SERVO2	ON	ON	OFF	ON	ON	ON	OFF	OFF
SYSTEM	ON	ON	OFF	ON	ON	ON	ON	OFF

DO[25] and DO[26] means the severity of running a program.

DO［25］	DO［26］	
OFF	OFF	The program continues running.
OFF	ON	The program will be paused.
ON	ON	The program will be aborted.

DO[27] and DO[28] means the severity of motion.

DO［27］	DO［28］	
OFF	OFF	The motion continues running.
ON	OFF	The motion will be stopped.
OFF	ON	The motion will be canceled.

DO[29] means the severity of a servomotor.

OFF	The servomotor continues running.
ON	The servomotor will be powered down.

DO[30] means the severity of the target tasks.

OFF	Effective only for the task.
ON	Effective for all the tasks.

DO[31] means the request for recovery.

OFF	No need to power down.
ON	Cold start is necessary.

DO[32] means the request for the display on TP.

OFF	No need to display.
ON	Need to display.

20.2.2 Alarm ID

Alarm ID will be outputted as the number in the following.

Typical alarm ID:

Number	Alarm ID
0	OS
3	PROG
7	MEMO
9	TPIF
11	SRVO
12	INTP
15	MOTN
19	JOG
20	APPL
23	SPOT
24	SYST
26	PALT
53	ARC
57	MACR
58	SENS
59	COMP

Other alarm ID:

Number	Alarm ID
1	SRIO
2	FILE
4	COND
5	ELOG
6	MCTL
8	GUID
10	FLPY
13	PRIO
14	TPAX
16	VARS
17	ROUT
18	WNDW
21	LANG
25	SCIO
27	UAPL
33	DICT
34	KCLI
35	TRAN

Number	Alarm ID
36	TKSP
37	COPT
38	APSH
42	CMND
43	RPM
44	LNTK
45	WEAV
46	TCPP
47	TAST
48	MUPS
49	MIGE
50	LSR
51	SEAL
52	PANE
54	TRAK
55	CALB
56	SP
60	THSR

20.2.3 Alarm Number

Alarm number will be directly output as binary data.

Example

In case of “SRVO 002 (severity SERVO)”
Alarm number is 2.
Alarm ID “SRVO” is 11.
The severity is SERVO.

DO［1］	OFF	Alarm number
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］	OFF	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 Data monitor items

ITEM	DESCRIPTION
INTEGER	System variables or KAREL program variables, whose data type is integer, will be monitored or recorded.
REAL	System variables 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.(Available from 7DC3/44, 7DF1/17)
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. (Available from 7DC3/44, 7DF1/17)
Q-CURRENT	A value of Q phase current about assigned Axis will be monitored or recorded. (Available from 7DC3/44)
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. (Available in 7DF1 series or later)
Bundle I/O	Continuous physical port corresponding to assigned logical port will be monitored or recorded. (Available in 7DF1 series or later)
SquareCurrent	Square current value about assigned Axis will be monitored or recorded. Unit is [Ap^2].(Available from 7DC3/44, 7DF1/17)
CART CAL POS	Cartesian position calculated by feedback. Assigned position element(X,Y,..,R) will be monitored or recorded. (Available from 7DC3/44, 7DF1/17)
JOINT COMM POS	Joint command position. Assigned Axis will be monitored or recorded. Data will be updated by about 100 ms cycle. (Available from 7DC3/44, 7DF1/17)
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. (Available from 7DC3/44, 7DF1/17)

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.

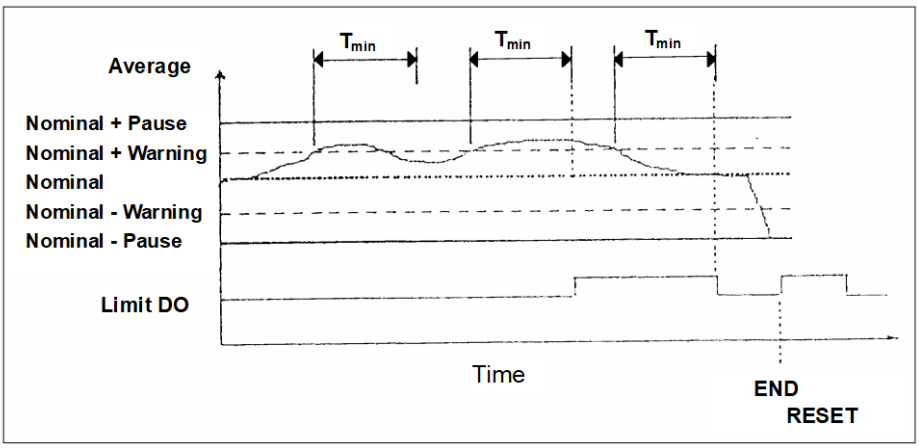


Fig.21 (b) Process Limit DO

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.

ITEM	DESCRIPTION
Schedule Default: Disabled	This item enables or disables printing of Schedule information in the report header.
Triggers Default: Disabled	This item enables or disables printing of Trigger information in the report header.
Program Name Default: Enabled	This item enables or disables printing the program name column in the Data Monitor report. See Fig. 21.1.
Line Number Default: Enabled	This item enables or disables printing the line number column in the Data Monitor report. See Fig. 21.1.
Date Default: Disabled	This item enables or disables printing the date and time of day column in the Data Monitor report. See Fig. 21.1.
Tick + time Default: Disabled	This item enables or disables printing the Tick and Time column in the Data Monitor report. See Fig. 21.1.
Event Default: Disabled	This item enables or disables printing the event column in the Data Monitor report. See Fig. 21.1. NOTE At this time, there is only one event defined. An event value of 1 indicates the data was recorded as a result of the recording frequency.
Distance Default: Enabled	This item enables or disables printing the distance column in the Data Monitor report. See Fig. 21.1.

Number	Date	Tick	Time	Event	Program	Line	Distance	Voltage(Command)[Volts]
1	18-AUG-98 10:41:46	0	0	1	TEST	2	0	0
2	18-AUG-98 10:41:46	62	0.248	1	TEST	2	0	0
3	18-AUG-98 10:41:46	124	0.496	1	TEST	2	0	0
4	18-AUG-98 10:41:46	186	0.744	1	TEST	2	0	0
5	18-AUG-98 10:41:46	248	0.992	1	TEST	3	0	0
6	18-AUG-98 10:41:46	310	1.24	1	TEST	3	0	0
7	18-AUG-98 10:41:46	372	1.488	1	TEST	3	0	0
8	18-AUG-98 10:41:48	434	1.736	1	TEST	3	0	0
9	18-AUG-98 10:41:48	496	1.984	1	TEST	3	0	0
10	18-AUG-98 10:41:48	558	2.232	1	TEST	3	0	0
11	18-AUG-98 10:41:50	620	2.48	1	TEST	3	0	0

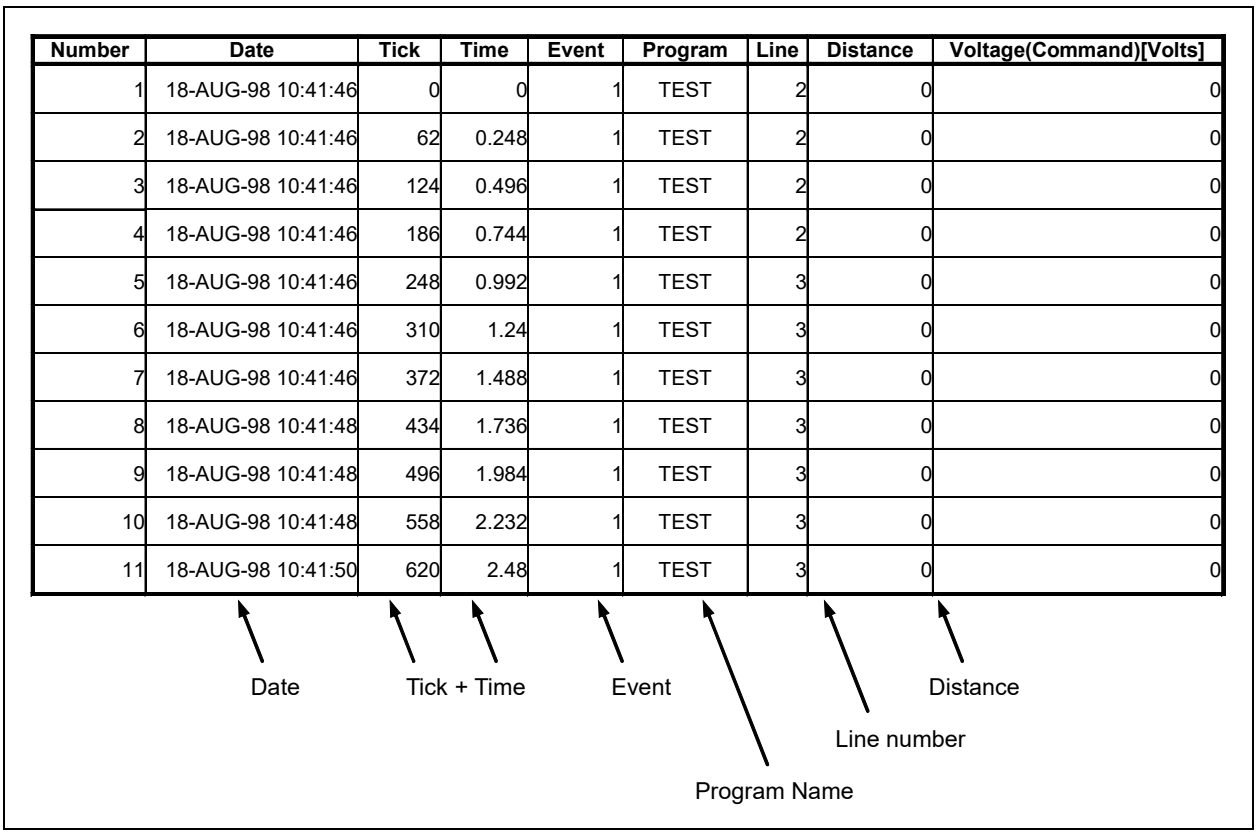


Fig. 21.1 Sample report(when each setup item is enabled)

Procedure 21-1 Setting up data monitor

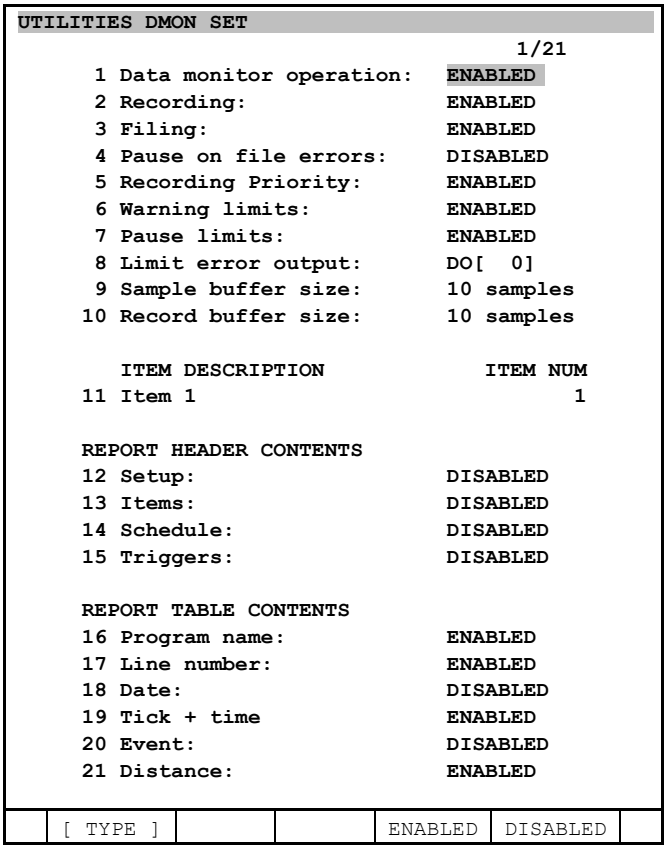
Condition

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.

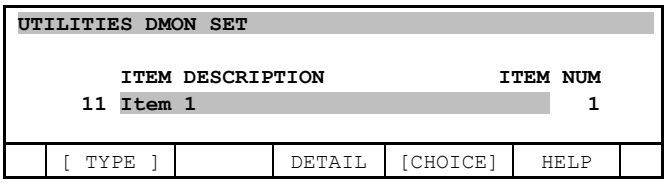


5 Select each item on the menu and set it as desired.

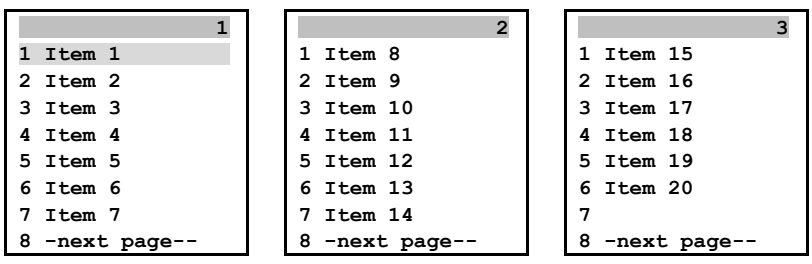
Edit Data Monitor Items

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.

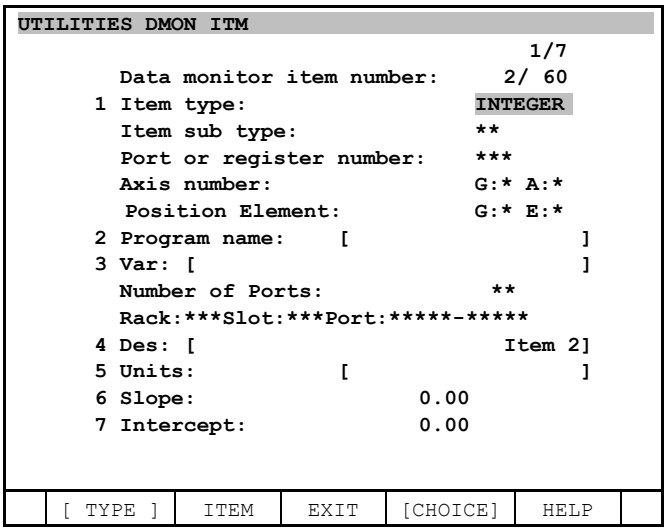
Item 11 has two columns. The right column contains an item number from 1 to 60. The left column contains the corresponding item description.



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.



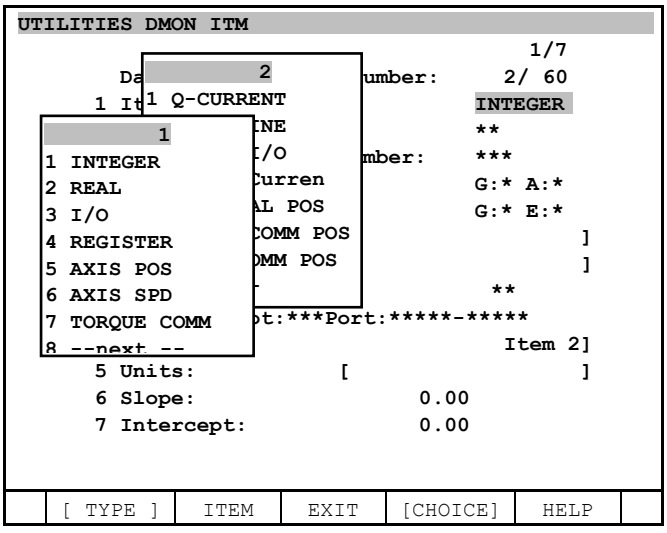
You can specify comment about the item in “4 Des” and “5 Unit”.
“6 Slope”, “7 Intercept” is for linear transformation. When Non-zero value is specified as slope, Data is transformed as follows.

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 system variable or KAREL variable.

Specify “2 Program name”.
Specify [*system*] for system variable or [name of KAREL program] for KAREL variable.

Specify “3 Var”.
Specify variable name in “3 Var”.

Example for setting system variable $MCR.$GENOVERRIDE

```
UTILITIES DMON ITM
                                          1/7
      Data monitor item number:     2/ 60
    1 Item type:                    INTEGER
      Item sub type:                **
      Port or register number:      ***
      Axis number:                  G:* A:*
      Position Element:             G:* E:*
    2 Program name:     [           *SYSTEM*]
    3 Var: [              $MCR.$GENOVERRIDE]
      Number of Ports:               **
      Rack:***Slot:***Port:*****-*****
    4 Des: [                          Item 3]
    5 Units:            [                   ]
    6 Slope:                  0.00
    7 Intercept:              0.00

|  | [ TYPE ] | ITEM | EXIT | [CHOICE] | HELP |  |
```

Case that I/O is selected:
Specify “2 Item sub type” and “3 Port or register number”.

Example for setting DO[3]

```
UTILITIES DMON ITM
                                          1/7
      Data monitor item number:     2/ 60
    1 Item type:                    I/O
    2 Item sub type:                DO
    3 Port or register number:      3
      Axis number:                  G:* A:*
      Position Element:             G:* E:*
      Program name:     [                 ]
      Var: [                              ]
      Number of Ports:               **
      Rack:***Slot:***Port:*****-*****
    4 Des: [                          Item 2]
    5 Units:            [                   ]
    6 Slope:                  0.00
    7 Intercept:              0.00

|  | [ TYPE ] | ITEM | EXIT | [CHOICE] | HELP |  |
```

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 Bundle I/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:                   Bundle I/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.
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 |
```

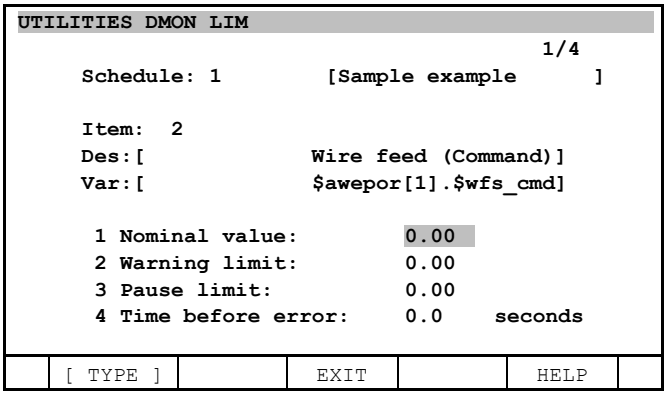
5 There are 5 Data Monitor schedules. The top line of the Schedule screen displays the current schedule number and its comment. To select a different schedule, press the F2, SCHEDULE, and enter the number of the schedule you want to modify after the prompt.
6 To enter or modify the schedule comment, 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.

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 system variable $DMONCFG.$NUM_DM_SCH to set up the number of the Data Monitor schedules. This system variable can be changed by system variable screen under controlled start mode.

NOTE
This system variable can be changed from 1 to 10 on software version 7DF5/20 or older. This system variable can be changed from 1 to 40 on software version 7DF5/21 or later.

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.

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.

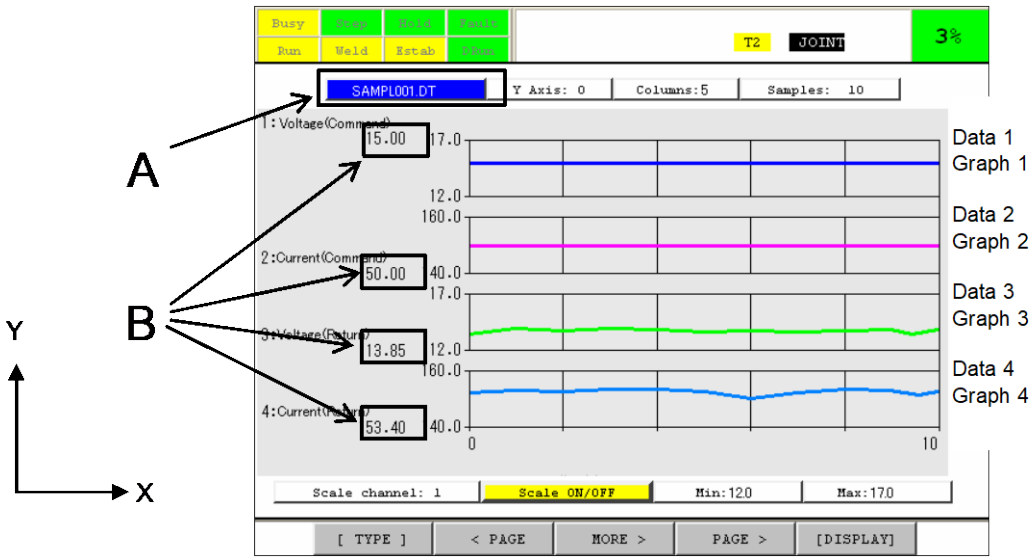


Fig. 21.4 Data monitor chart screen

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.

NOTE
This screen has been supported since 7DF1/01.
This screen does not support on Tablet Teach Pendant.

Procedure 21-4 Data monitor operation

Condition

Data monitor function is ordered.
The software version is 7DF1/01 or later.

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.

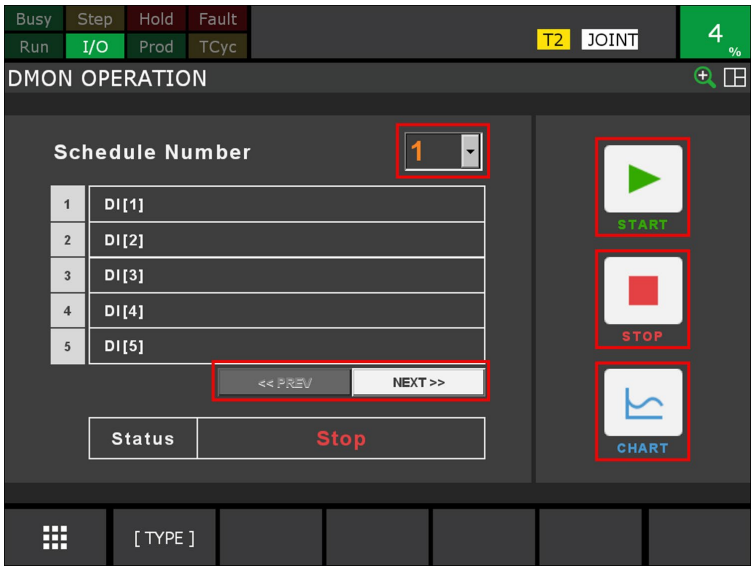


Fig. 21.5 Data monitor operation screen

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 FUNCTION

Brake check function (A05B-2600-J951) is a preventive maintenance function. It is used to check that the motor does not move when motor torque is applied while brake is engaged. This allows user to perform early detection of potential brake failure. User is informed of the checking result by warning message or system variables.

22.1 INITIAL SETTING BEFORE USE

Enable/Disable switch for brake check function

The following system variable is the switch that enable/disable brake check function.

`$BCCFG.$BCK_ENABLE` TRUE: Enable (default) / FALSE: Disable

In order to check or modify the system variables, press the [MENU] key and select "SYSTEM". Then, press the F1, [TYPE] key and select "Variables". The system variables screen will be displayed.

Please confirm this variable is TRUE (Enable) before use. The default value of this variable is TRUE (Enable). If this variable is FALSE (Disable), brake check never starts.

Enable/Disable switch for each axis

On this function, the user can customize enable/disable brake check for each axis. Please set the following system variable to TRUE (Enable) for the axis where you want to execute brake check.

`$BCK_GRP[group].$CHK_AXIS[axis]` TRUE: Enable / FALSE: Disable

On robot axis, the default value is TRUE (Enable). On except robot axis (aux axis, servo gun, or etc.), the default value is FALSE (Disable).

In addition, if you execute brake check on dual drive axis, please set this variable to TRUE (Enable) for both master and slave axis.

Setting brake torque

This function needs brake torque for each axis in order to determine torque command during brake check. Please set the following system variable with the brake torque [kgf*cm] for each axis.

`$BCK_GRP[group].$BRK_TORQUE[axis]` brake torque [kgf*cm]

On robot axis, this variable is set automatically, however, on except robot axis (aux axis, servo gun, or etc.), this variable is not set automatically, so please manually set this variable with brake torque if you execute brake check on except robot axis. For specification value of brake torque, please refer to "BRAKE SPECIFICATIONS" on AC SERVO MOTOR DESCRIPTIONS corresponding to your motor. Please note that the unit of this variable is [kgf*cm].

AC SERVO MOTOR DESCRIPTIONS	Specification number
AC SERVO MOTOR α series DESCRIPTIONS	B-65142EN
AC SERVO MOTOR β series DESCRIPTIONS	B-65232EN
AC SERVO MOTOR αi series DESCRIPTIONS	B-65262EN
AC SERVO MOTOR βi series DESCRIPTIONS	B-65302EN

In addition, if you execute brake check on dual drive axis (except robot axis), please set this variable as follows. (For robot axis, it is no need to set even if it's dual drive axis.)

Case-1: Either master or slave motor has one brake.
Please set this variable with the brake torque [kgf*cm] for only the axis that has brake. For the axis that has no brake, please set 0.0 (default).

Case-2: Both master and slave motors have brake respectively.
Please set this variable with the brake torque [kgf*cm] for both master and slave axes. And then the brake torque value of master and slave axis must be same. If these values are different in the Case-2, it is not supported.

22.2 START BRAKE CHECK

To start brake check, please set the following system variable to TRUE.

$BCCFG.$BCK_START TRUE: Brake check is running. / FALSE: Brake check is finished.

User can start brake check by parameter instruction of TP program or manually setting on system variable screen. This variable automatically returns to FALSE when brake check is finished. Please not to move robot till this variable automatically returns to FALSE. In addition, if brake check function is disabled (**$BCCFG.$BCK_ENABLE=FALSE**), this variable immediately returns to FALSE even if this variable is set to TRUE.

Follows is a sample of TP program that execute brake check.

```
1: J P[1] 100% FINE
2: L P[2] 2000mm/sec FINE
3:  WAIT 1.00 (sec)
4:  $BCCFG.$BCK_START=1
5:  WAIT $BCCFG.$BCK_START=0
6: L P[3] 2000mm/sec FINE
```

Brake check cannot be started while the robot is moving as specification, so this program waits for robot to stop completely by WAIT instruction at Line 3. And this program starts brake check at Line 4 and waits to finish brake check at Line 5. As just described on this sample program, brake check must be executed under the robot stops completely.

> ⚠ **CAUTION**
> In case of brake failure, the motor can move during brake check. So please be sure to execute brake check in sufficiently spaced layout. Total motion command per single check is vary by robot or axis, however, it is up to 200000[pulse]. This is about 0.4[rev] in motor revolution and about 1.4[deg] in degree of the angle assuming the gear ratio as 1/100.

Additionally, in case where the power is shut OFF during brake check for some reason, the following alarm will be posted at next power ON. This alarm indicates abnormal termination at last check.

```
BRCH-003  Last check was failed
```

The following system variable will be also set to TRUE at next power ON.

$BCCFG.$LAST_FAILED TRUE: Last check was failed. / FALSE: Last check was normally finished.

In this case, the results of brake check will not be generated, so please execute brake check again at next power ON. If next brake check would be normally finished, this variable will return to FALSE automatically.

22.3 RESULTS OF BRAKE CHECK

$BCCFG.$BCK_START automatically returns to FALSE when brake check finishes. And then the results are also posted as warning message.
If all brake passes as "Good" brake, the following message will be posted.

```
BRCH-001  All brake are normal
```

By contrast, if brake failure is detected as "Bad" brake, the following message will be posted.

Ex) Brake failure is detected on group1/J2.

```
BRCH-002  Brake abnormal (G:1,A:2)
```

The results are also recorded on the following system variable at the same time. The value of this variable is kept till next brake check starts.

$BCK_GRP[group].$CHK_RESULT[axis] TRUE: Brake failure/FALSE: Normal brake (default)

In the case of Dual Drive, or if 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 measured.

22.4 LIMITATIONS

Brake check function cannot be executed under the following conditions, or alarm will be posted.

- Calibration has not been done.
- Robot is moving.
- Servo is OFF.
- Servo parameter is under updating.
- Dual motor is under the adjusting mode.
- Even though the axis has brake and brake check is enabled (**$BCK_GRP[group].$CHK_AXIS[axis]** is TRUE), brake torque (**$BCK_GRP[group].$BRK_TORQUE[axis]**) is not set.
- Teach Pendant is enabled.
- Any axis is near by stroke limit when brake check starts.

22.5 CAUTION

- It takes about 10[sec] for single brake check execution.
- When the motor is heating up, the motor torque can decrease, so it may not be able to detect brake failure. To detect it with high accuracy, please execute brake check preferably when the motor is in low-temperature, e.g. first thing in the morning.

23 PANEL WIZARD

The Panel Wizard option allows you to use the controller to create up to four operation panels to be displayed on the teach pendant. You don't need PC to create panel.

To use this function, Panel wizard option (A05B-2600-R594) is required.

NOTE
Panel wizard is included in KAREL use support (A05B-2600-J971) in 7DC1 and 7DD0 series. However, KAREL use support is divided to Customize support (A05B-2600-J742) and Panel wizard from 7DC2 series. If the same function of KAREL use support is needed, please order both Customize support and Panel wizard.

NOTE
Please use the teach pendant which equips the touch panel for this function.

23.1 SETTING UP

23.1.1 Overview

Use procedure 23-1 to setup the Panel Wizard.

Procedure 23-1 Setting up the Panel Wizard

Step

1 Press the [MENU] key.
2 Select BROWSER.
3 Press the F1, [TYPE].
4 Select Panel Wizard. The following screen will be displayed.

Fig. 23.1.1 (a) Panel wizard start screen

5 Select Start Wizard. The following screen will be displayed.

NOTE
Double Pane mode will be displayed automatically.

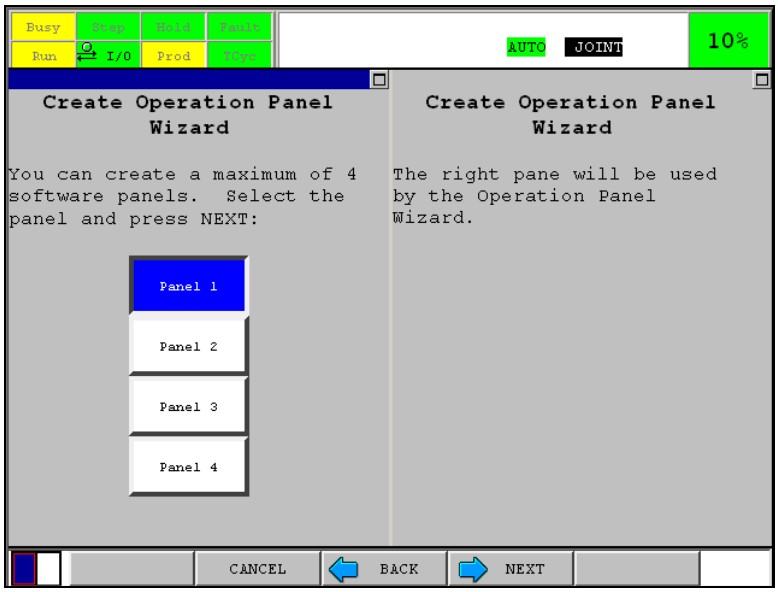


Fig. 23.1.1 (b) Panel selection screen

NOTE
After you make a selection on any screen, press the [NEXT] key. To cancel the Panel Wizard at any time, press the F2, CANCEL. To display the previous screen at any time, press the F3, BACK.

6 Select the panel that you want to create. Refer to Fig. 23.1.1 (c) for a flowchart overview of the steps required to create an operator panel.

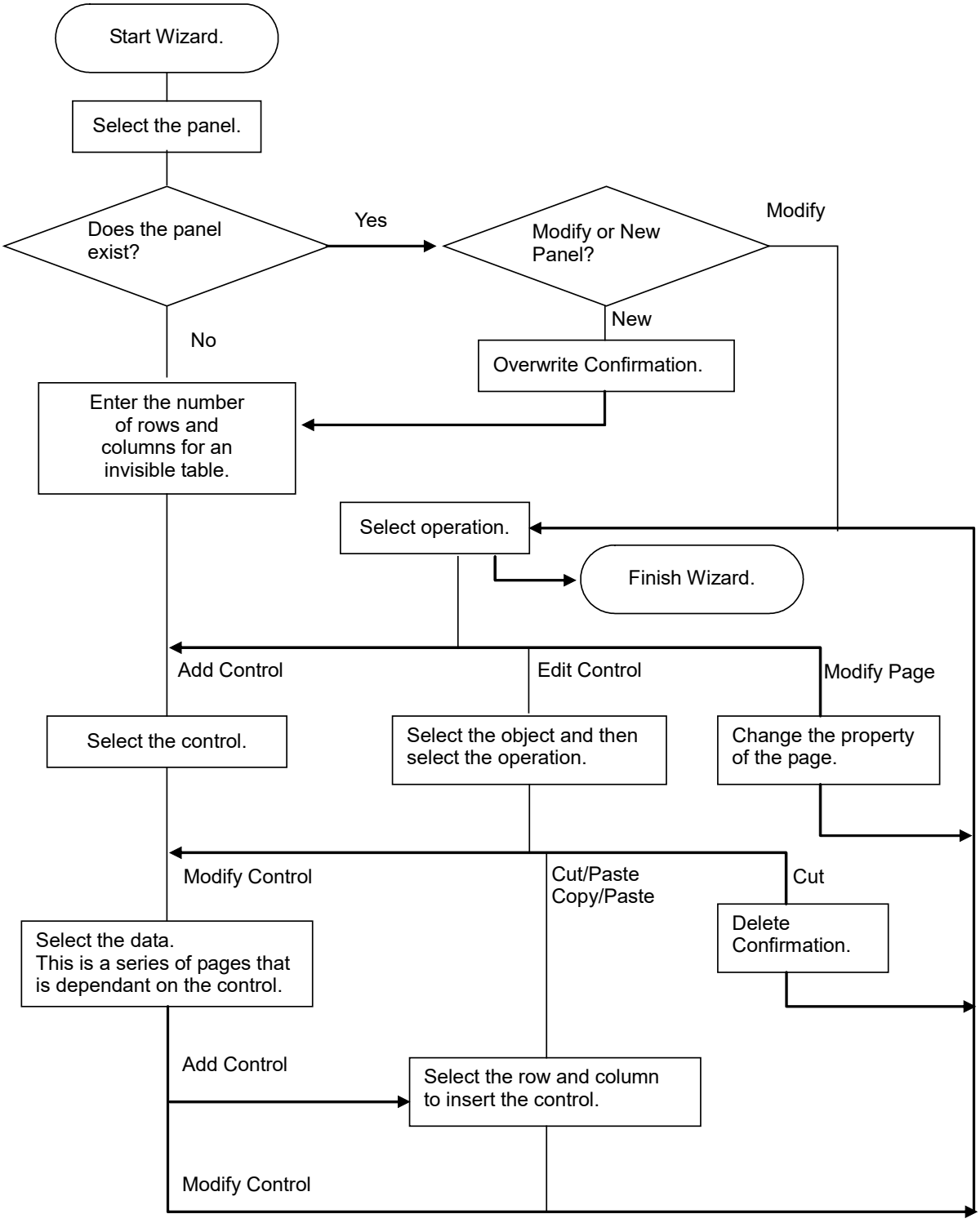


Fig. 23.1.1 (c) Create operation panel flowchart

NOTE
Control in the chart is *i*Pendant control. It corresponds to parts like buttons displayed by Panel Wizard.

23.1.2 Available *i*Pendant Controls

Panel wizard create operation panel by placing *i*Pendant control on invisible table. *i*Pendant Control to add is selected in a screen similar to following. Please refer to Fig. 23.1.1 (c) for how to display the screen.

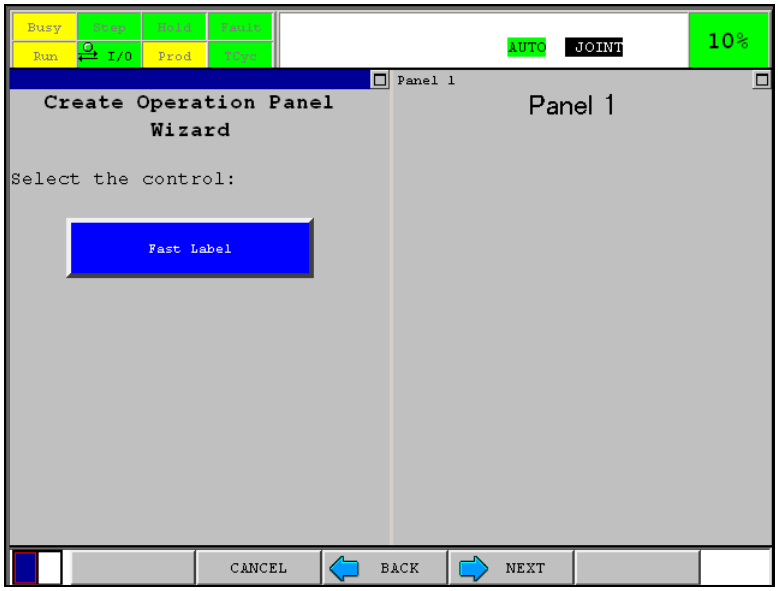


Fig. 23.1.2 *i*Pendant control selection screen

Available *i*Pendant Controls are displayed after button on the center of the screen is pressed. The following table is list of the choices.

Table 23.1.2 Available *i*Pendant controls

Choice	Description
Fast Label	This is simply shorted version of another choice “Label”. It is described later. Refer to the description.
Fast lamp	This is simply shorted version of another choice “Toggle Lamp”. It is described later. Refer to the description.
Fast Switch	This is simply shorted version of another choice “Toggle Button”. It is described later. Refer to the description.
Button Change	This control is used to display the specified web page. It can also be used to run KAREL program.
Command Button	This control is used to write the specified value to a Register, System Variable and so on when the button is pushed.
EditBox	This control is used to change the value of a Register, System Variable and so on.
Label	The value of a Register, System or KAREL Variable or I/O is displayed.
Toggle Button	This control is used to change the value of a Register, System Variable and so on depending on the ON (Pushed)/OFF (Popped) status of the button.
Toggle Lamp	This control is used to change the color of the control if the value of a Register, System Variable and so on fulfills the specified condition.

Setup of each control is described following sections.

NOTE
Wizard is always display in double pane mode. Setup is done in the left pane. Right pane is used to display rough image of result of setup. Because display on the right pane depends on content of setup, figures in this manual usually do not match what you see during setup.

23.1.3 Setting up Fast Label

Fast lamp is shorted version of Label control. Label control displays value of a Register, System Variable and so on. Procedure to add Fast Label is following.

Procedure 23-2 Addition of Label control by Fast Label

Step

1 Select Fast Label in *i*Pendant Control selection screen (Fig. 23.1.2).
2 Press the F4. The following screen will be displayed.

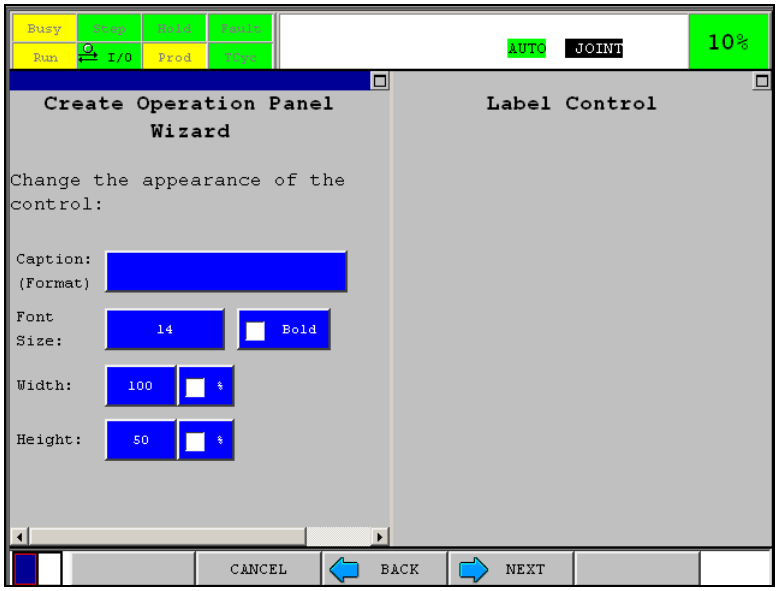


Fig. 23.1.3 (a) Size setup screen

Table 23.1.3 (a) Items in size setup screen

Item	Description
Caption (format)	If data type is static (this is always true for Fast Label), input string is text displayed. If the other data type is used, caption is format string.
Font Size	Size of font. Choices are 14,16,18 and 24.
Bold	If this is checked, value is displayed in bold.
Width and %	This is width of control. If % is not checked, width is specified in pixels. If % is checked, width is specified in percentage of screen size.
Height and %	This is height of control. Unit is specified in the same way as width.

NOTE
In setup of Fast Label, data type is default to static. Input string to be displayed to Caption.

3 Setup items as required.
4 Press the F4, NEXT. The following screen will be displayed.

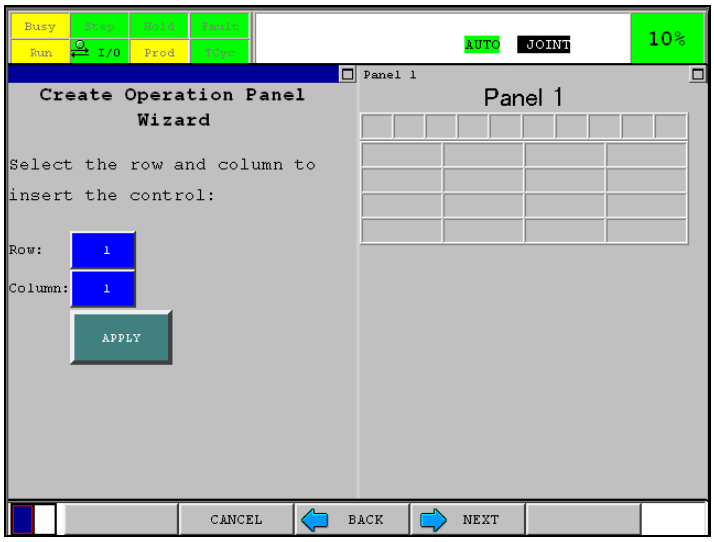


Fig. 23.1.3 (b) Control insertion screen

Table 23.1.3 (b) Items in control insertion screen

Item	Description
Row and Column	Row and column to insert *i*Pendant Control
APPLY	This is not setup item. Pressing the button displays rough image of screen after insertion is displayed on right pane. Please note that screen image in right pane is just a rough guide.

5 Press the F4, NEXT. This is end of addition of Label control by Fast Label.

23.1.4 Setting up Fast Lamp

Fast lamp is shorted version of Toggle lamp. Toggle lamp is used to change the color of the control if the value of a Register, System Variable and so on fulfills the specified condition. Procedure to add Fast Lamp is following.

Procedure 23-3 Addition of toggle lamp by Fast Lamp

Step

1 Select Fast Lamp in *i*Pendant Control selection screen (Fig. 23.1.2).
2 Press the [F4] key. The following screen will be displayed.

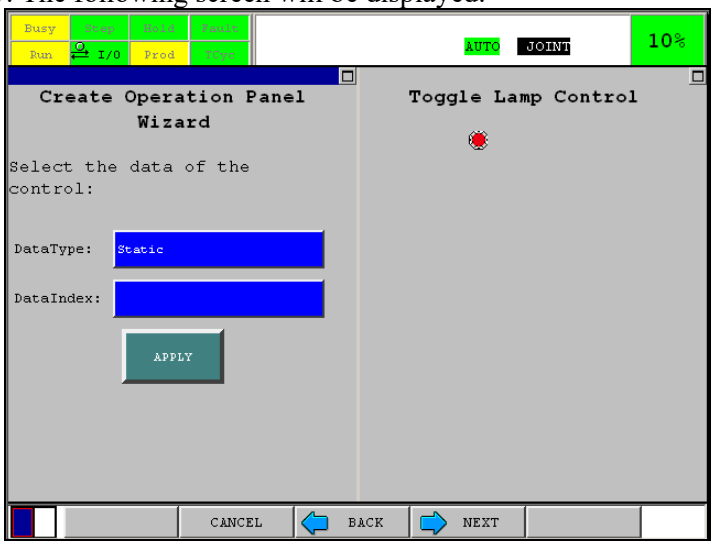


Fig. 23.1.4 (a) Date type setup screen

DataType specifies the type of the data to be monitored or modified depending on the type of control. DataIndex specifies the number or the variable name associated with the DataType.

Table 23.1.4 (a) Data type and data index

Data type	Description
Static	Specified string is used for caption.
Numeric Register	The value of numeric register specified in Date index is used.
System variable	The value of system variable specified in Data index is used. For System Variables, the type must be Integer, Real, Boolean, Short, Byte, or String. For example, the value $MNUFRAMENUM[1] is can be specified in Data index because this system variable is Byte type, so following is specified in Data index in this case. $MNUFRAMENUM[1]
KAREL variable	The value of KAREL variable specified in Data index is used. For KAREL Variables, enclose the program name inside […]. The type must be Integer, Real, Boolean, Short, Byte, or String. Following is example of variable STR_VAR of KAREL program USEREXT. [USEREXT]STR_VAR
Dictionary Element	Dictionary element specified in Data index is used. For Dictionary Elements, specify the dictionary name and enclose the dictionary element inside […]. For example, TPAR[5] Only Label control uses dictionary element.
DI	The value of DI specified in Data index is used.
DO	The value of DO specified in Data index is used.
…….. (the other I/O type)	The value of I/O type specified in Data type and Data index is used. Data index is port number.

Toggle lamp uses specified data to decide ON/OFF of lamp.

3 Setup items as required.

4 Press F4, NEXT. The following screen will be displayed.

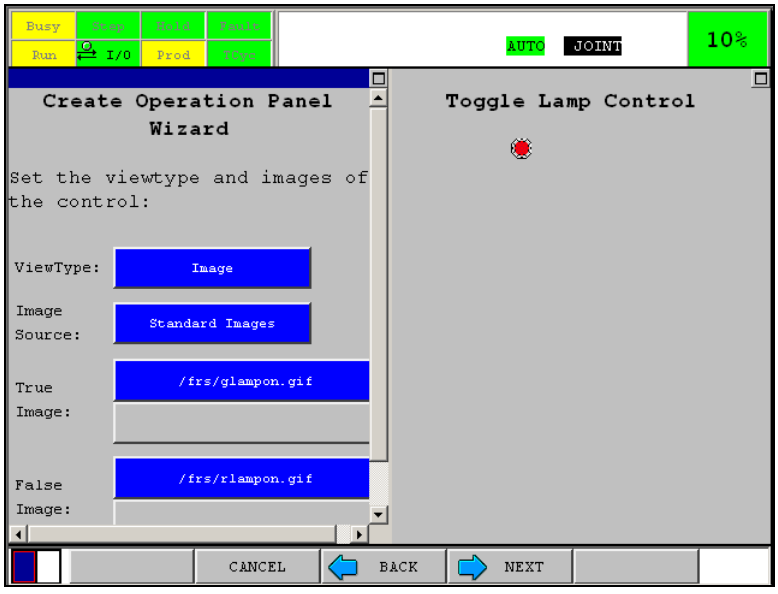


Fig. 23.1.4 (b) Image setup screen

Table 23.1.4 (b) Items in image setup screen

Item	Description
View type	Type of button to display Panel: specify panel type. Circle: specify circle type. Fixed circle: specify circle with fixed size. Image: specify image type.
Image Source	This is active when view type is image. Standard Images: arranged image files can be used. fr:*.gif : GIF file in FR: can be used. FR is device on F-ROM of controller. fr:*.jpg : JPG file in FR: can be used. Keyboard Entry: Input path of image file.
True Image	This specifies the image to be displayed when the read value fulfill the condition expression. Used only in case that ViewType is Image type. When image source is Keyboard Entry, button of lower side is active. Input path of image file by the button. When image source is not Keyboard Entry, button of upper side is active. The menu is displayed by the button. Select image fie from the sub menu.
False Image	This specifies the image to be displayed when the read value does not fulfill the condition expression. This is used only in case that View Type is Image type. Refer to description of True Image for how to specify image file.

5 Setup items as required.
6 Press of F4, NEXT leads you to control insertion screen. Please refer to Subsection 23.1.3.
7 Decide position to insert and press the F4, NEXT. This is end of addition of Toggle Lamp by Fast Lamp.

23.1.5 Setting up Fast Switch

Fast switch is shorted version of Toggle Button. Toggle button is used to change the value of a Register, System Variable and so on depending on the ON (Pushed)/OFF (Popped) status of the button. Procedure to add Toggle button is following.

Procedure 23-4 Addition of Toggle button by Fast Switch

Step

1 Select Fast Switch in *i*Pendant Control selection screen (Fig. 23.1.2).
2 Press of F4, NEXT leads you to data type setup screen. Please refer to Subsection 23.1.4.
3 Press of F4, NEXT leads you to image setup screen (Fig. 23.1.5). Screen is same as Subsection 23.1.4. But choices of view type are different.

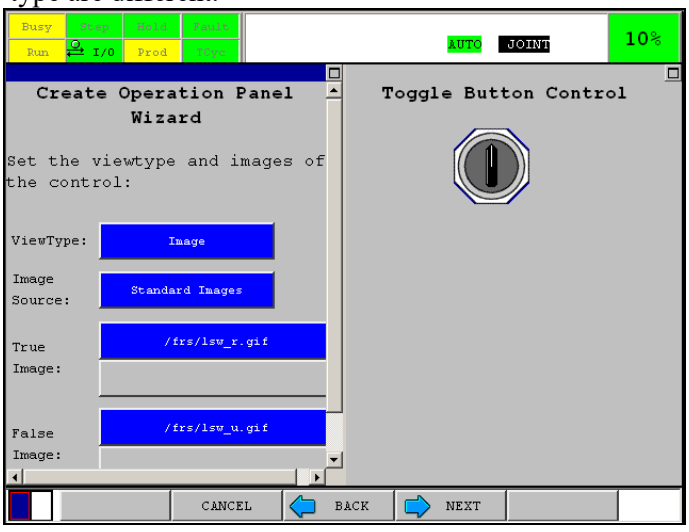


Fig. 23.1.5 Toggle button image setup screen

Table 23.1.5 Items in toggle button image setup screen

Item	Description
View type	Type of button to display Panel: specify panel type. Check box: specify check box type. Image: specify image type. F2-F5, F7-F10: specify function key type. Circle: specify circle. (Size of circle is fixed. This is just like fixed circle.)
Image Source	This is same as toggle lamp. Please refer to table 23.1.4 (b).
True Image	This specifies the image to be displayed when the button status is ON. This is used only in case that ViewType is image type. How to setup is just same as toggle lamp. Please refer to table 23.1.4 (b).
False Image	This specifies the image to be displayed when the button status is OFF. This is used only in case that ViewType is image type. How to setup is just same as toggle lamp. Please refer to table 23.1.4 (b).

4 Press of F4, NEXT leads you to control insertion screen. Please refer to Section 23.1.3.
5 Decide position to insert and press the F4, NEXT. This is end of addition of Toggle button by FAST Switch.

23.1.6 Addition of Button Change Control

Button Change control is used to display the specified web page. It can also be used to run KAREL program. Method to run KAREL program is described later. Use Procedure 23-5 to add Button change control.

Procedure 23-5 Addition of Button Change Control

Step

1 Select Button Change in *i*Pendant Control selection screen (Fig. 23.1.2).
2 Press the [F4] key. The following screen will be displayed.

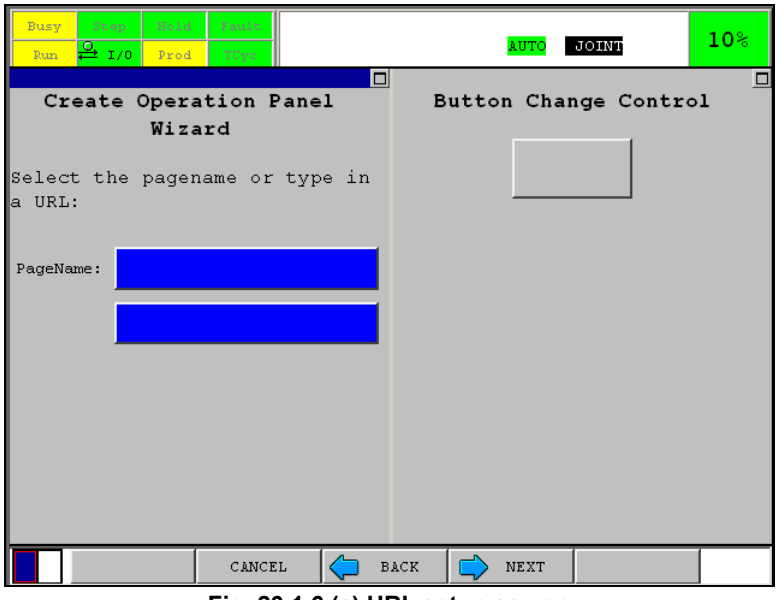


Fig. 23.1.6 (a) URL setup screen

Table 23.1.6 (a) URL setup screen

Item	Description
PageName	Press of upper side button displays STM/HTM files in FR:. Select file to display. If you want to display the other file, press lower side button and input URL.

3 Press of F4, NEXT, leads you to Button Change image setup screen. Screen is same as image setup screen of section 23.1.4. But choices of view type are different.

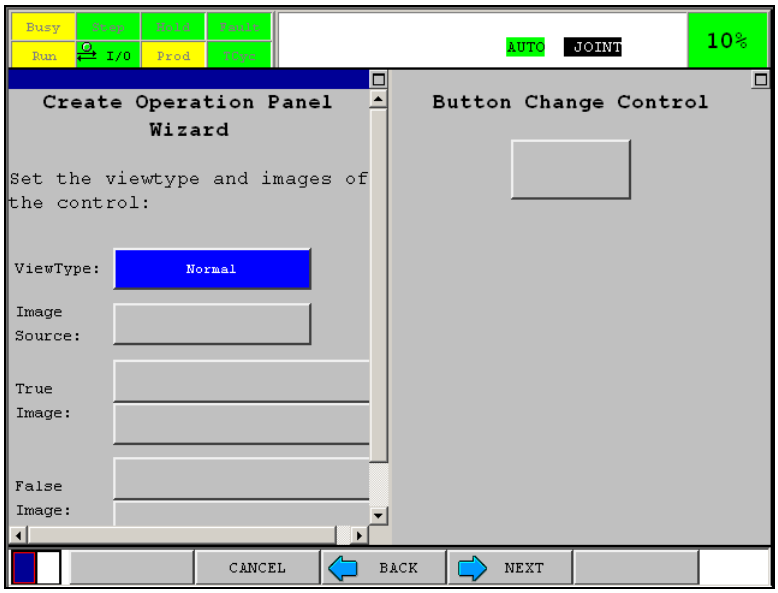


Fig. 23.1.6 (b) Button change image setup screen

Table 23.1.6 (b) Items in Button change image setup screen

Item	Description
View type	Type of button to display Normal: Specify the normal button. Image: specify image type. F2-F5, F7-F10: specify function key type.
Image Source	This is same as toggle lamp. Please refer to table 23.1.4 (b).
True Image	This specifies the image to be displayed when the button is pushed. Used only in case that ViewType is Image type. How to setup is just same as toggle lamp. Please refer to table 23.1.4 (b).
False Image	Specify the image to be displayed when the button is not pushed. Used only in case that ViewType is Image type. How to setup is just same as toggle lamp. Please refer to table 23.1.4 (b).

4 Press the F4, NEXT. The following screen will be displayed.

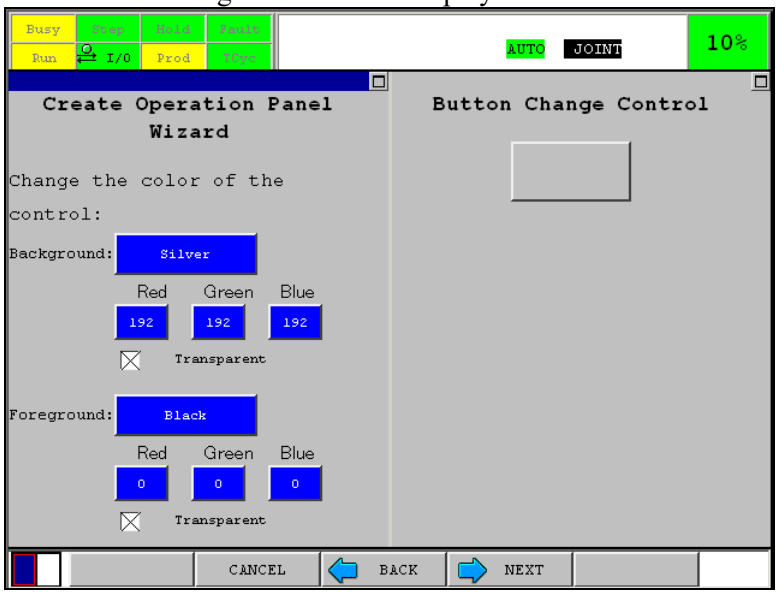


Fig. 23.1.6 (c) Color setup screen

Table 23.1.6 (c) Items in color setup screen

Item	Description
BackGround	This specifies background color. Pressing button displays list of standard colors in the menu.
(BackGround) Red, Green, Blue	Specify background color by RGB. This enables more detailed setup than selection from menu of BackGround button.
(BackGround) Transparent	With check of this item, background color of control becomes background color of web page.
ForeGround	This specifies color of character. How to setup is same as background color.
(ForeGround) Red, Green, Blue	Please refer to the same item for background color.
(ForeGround) Transparent	With check of this item, foreground color of the control becomes foreground color of web page.

5 Press of F4, NEXT leads you to size setup screen (Fig. 23.1.3 (a)). Please refer to Section 23.1.3. Caption is used as fixed string. It is not used as format.
6 Press F4, NEXT. The following screen will be displayed.

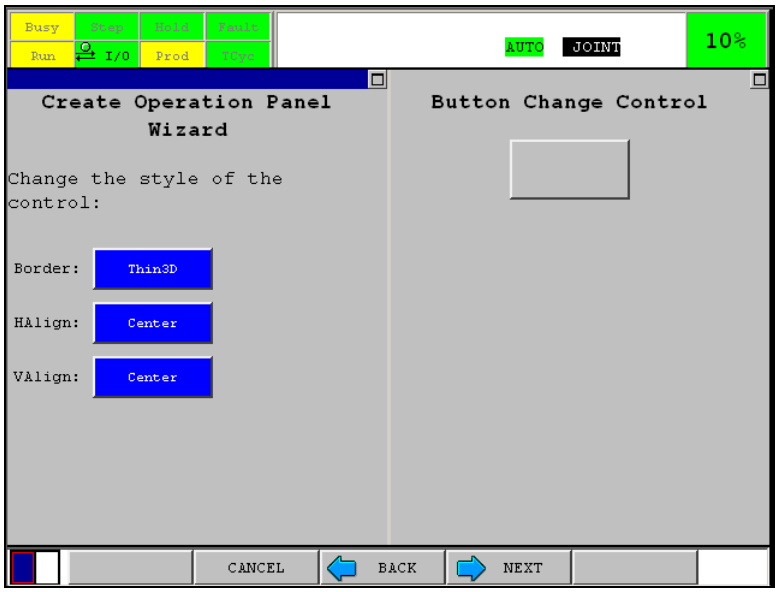


Fig. 23.1.6 (d) Style setup screen

Table 23.1.6 (d) Items in style setup screen

Item	Description
Border	This specifies type of borderline of button. Choices are Thin 3D, None, Black, ForeColor and Bold 3D.
HAlign	This specifies alignment of string in horizontal direction. Choices are Left, Center and Right.
VAlign	This specifies alignment of string in vertical direction. Top, Center and Bottom.

Following figure shows difference between types of borderline. The right pane shows buttons of each 5 border types. This figure is just a sample that has nothing to do with flow to add button change control.

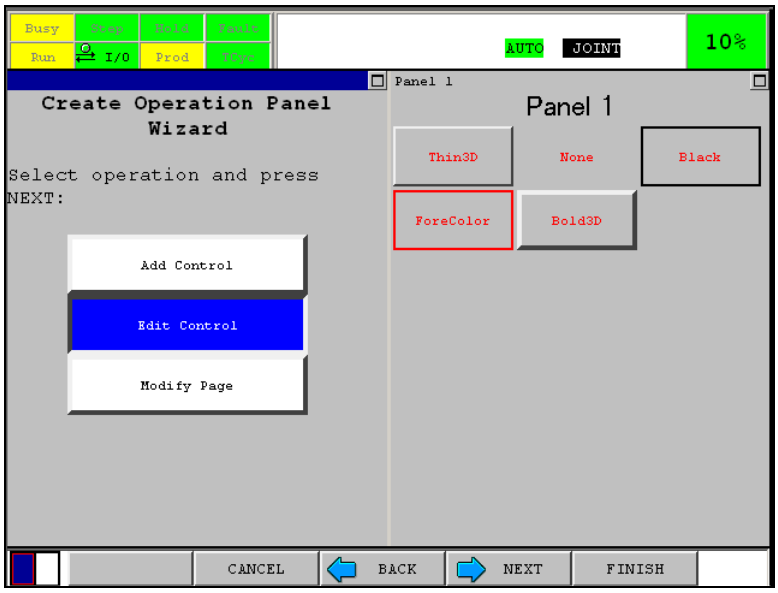


Fig. 23.1.6 (e) Type of borderlines

7 Press of F4, NEXT leads you to control insertion screen. Please refer to Subsection 23.1.3.
8 Decide position to insert and press the F4, NEXT. This is end of addition of Button Change control.

23.1.7 Addition of Command Button Control

Command Button Control is used to write the specified value to a Register, System Variable when the button is pushed. Procedure to add Command Button Control is following.

Procedure 23-6 Addition of Command Button Control

Step

1 Select Command Button in *i*Pendant Control selection screen (Fig. 23.1.2).
2 Press of F4, NEXT leads you to data type setup screen. Please refer to section 23.1.4.
3 Press the F4, NEXT. The following screen will be displayed.

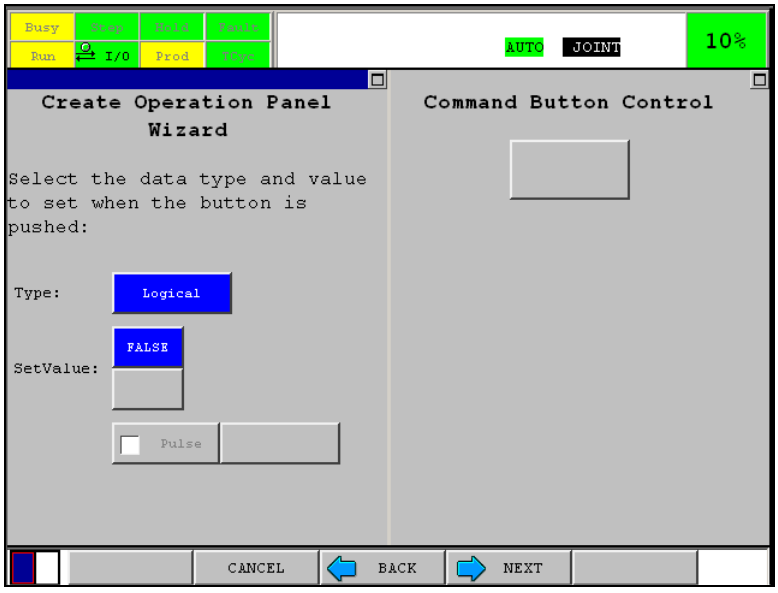


Fig. 23.1.7 Command button setup screen

Table 23.1.7 Items in command button setup screen

Item	Description
Type	Logical: Data is transacted as bool type. (SetValue is TRUE/FALSE.) Numerical: Data is transacted as numerical type. The value of SetValue is applied.
SetValue	This specifies the value written when the button is pushed.
Pulse	This is active only when data type is DO. If this is checked, signal output becomes pulsed output.
Pulse width (right item of pulse check box)	This specifies width of pulse in msec.

4 Press of F4, NEXT leads you to image setup screen. This is same as Button change image setup screen (Fig. 23.1.6 (b)). Please refer to section 23.1.6.
5 Press of F4, NEXT leads you to color setup screen. Please refer to section 23.1.6.
6 Press of F4, NEXT leads you to size setup screen. Please refer to section 23.1.3. Caption is used as fixed string. It is not used as format.
7 Press of F4, NEXT leads you to style setup screen. Please refer to section 23.1.6.
8 Press of F4, NEXT lead you to control insertion screen. Please refer to section 23.1.3.
9 Decide position to insert and press F4, NEXT. This is end of addition of Command Button Control.

23.1.8 Addition of Edit Box Control

Edit Box Control is used to enter and change the value of a Register, System Variable and so on. Procedure to add Edit Box Control is following.

Procedure 23-7 Addition of Edit Box Control

Step

1 Select Edit Box in *i*Pendant Control selection screen (Fig. 23.1.2).
2 Press of F4, NEXT leads you to data type setup screen. Please refer to section 23.1.4.
3 Press the F4, NEXT. The following screen will be displayed.

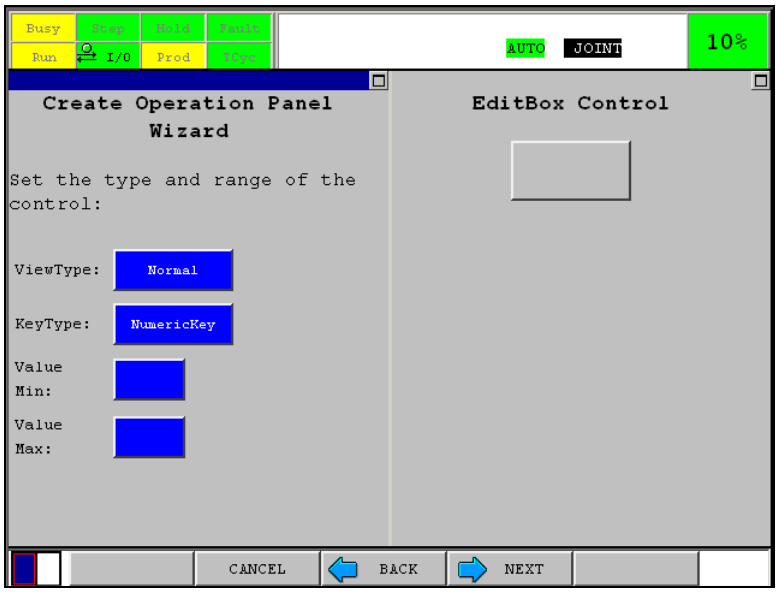


Fig. 23.1.8 Edit box setup screen

Table 23.1.8 Items in edit box setup screen

Item	Description
View type	Normal: specify simple button. F2-F5, F7-F10: specify function key type.
KeyType	Select the type of virtual keyboard. NumericKey: virtual keyboard for number input FullKey: virtual keyboard for string input
Value Min, Value Max	Minimum and maximum value of numerical value.

4 Press of F4, NEXT leads you to color setup screen. Please refer to section 23.1.6.
5 Press of F4, NEXT leads you to size setup screen. Please refer to section 23.1.3.
6 Press of F4, NEXT leads you to style setup screen. Please refer to section 23.1.6.
7 Press of F4, NEXT lead you to control insertion screen. Please refer to section 23.1.3.
8 Decide position to insert and press the F4, NEXT. This is end of addition of Edit Box Control.

23.1.9 Addition of Label Control

Label control displays value of a Register, System Variable and so on. Procedure to add Fast Label is following.

Procedure 23-8 Addition of Label Control

Step

1 Select Label in *i*Pendant Control selection screen (Fig. 23.1.2).
2 Press of F4, NEXT leads you to data type setup screen. Please refer to section 23.1.4.
3 Press of F4, NEXT leads you to size setup screen. Please refer to section 23.1.3.
4 Press of F4, NEXT leads you to style setup screen. Please refer to section 23.1.6.
5 Press of F4, NEXT leads you to color setup screen. Please refer to section 23.1.6.
6 Press of F4, NEXT lead you to control insertion screen. Please refer to section 23.1.3.
7 Decide position to insert and press F4, NEXT. This is end of addition of Label Control.

Setup of Fast Label included step 3 and 6 only. Default value is used for items in the other steps. You can add Label Control by Fast Label and then modify it.

23.1.10 Addition of Toggle Button Control

Toggle Button Control is used to change the value of a Register, System Variable and so on depending on the ON (Pushed)/OFF (Popped) status of the button. Procedure to add Toggle Button Control is following.

Procedure 23-9 Addition of Toggle Button Control

Step

1 Select Toggle Button in *i*Pendant Control selection screen (Fig. 23.1.2).
2 Press of F4, NEXT leads you to data type setup screen. Please refer to section 23.1.4.
3 Press the F4, NEXT. The following screen will be displayed.

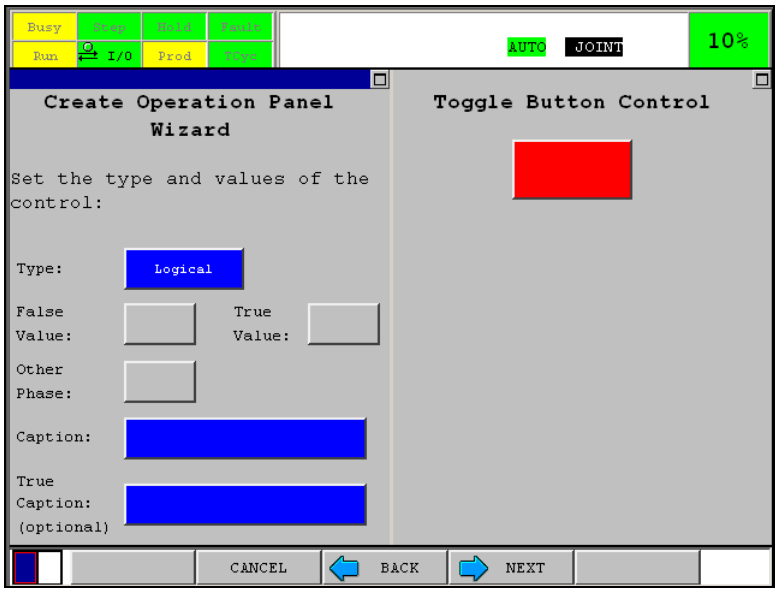


Fig. 23.1.10 (a) Toggle button setup screen

Table 23.1.10 (a) Items in toggle button setup screen

Items	Description
Type	This specifies data type to process. Logical: Data is transacted as boolean type. False value and True Value are not used. Numerical: Data is transacted as numerical type. False Value and True Value are used.
False Value	This is active when Type is Numerical. This specifies the value written when the button status is changed to ON (TRUE).
True Value	This is active when Type is Numerical. This specifies the value written when the button status is changed to OFF (FALSE).
Other Phase	This is active when Type is Numerical. This specifies the status (TRUE/FALSE) in case that the value is equal to neither TrueValue nor FalseValue.
Caption	This specifies the fixed String.
True Caption (optional)	This specifies the fixed String when the value is TRUE. If not specified, Caption is used.

4 Press of F4, NEXT leads you to Toggle Button Image setup screen. Please refer to section 23.1.5.
5 Press the F4, NEXT. The following screen will be displayed.

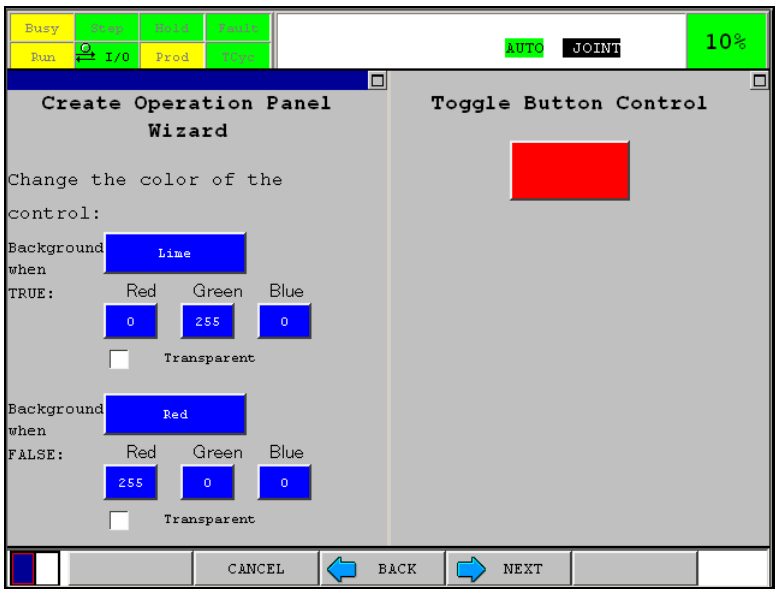


Fig. 23.1.10 (b) True/False color setup screen

Table 23.1.10 (b) Items in True/False color setup screen

Item	Description
Background when TRUE	This specifies the color displayed when the read value is equal to TrueValue or not equal to FalseValue. How to setup is same as color setup screen. Please refer to section 23.1.6.
Red, Green, Blue	Role is same as color setup screen. Please refer to section 23.1.6.
Transparent	Role is same as color setup screen. Please refer to section 23.1.6.
Background when FALSE	This specifies the color displayed when the read value is equal to FalseValue or not equal to TrueValue. How to setup is same as color setup screen. Please refer to section 23.1.6.

6 Press of F4, NEXT leads you to color setup screen. Please refer to Subsection 23.1.6.
7 Press of F4, NEXT leads you to size setup screen. Please refer to Subsection 23.1.3. Caption is used as fixed string. It is not used as format.
8 Press of F4, NEXT leads you to style setup screen. Please refer to Subsection 23.1.6.
9 Press of F4, NEXT lead you to control insertion screen. Please refer to Subsection 23.1.3.
10 Decide position to insert and press the F4, NEXT. This is end of addition of Toggle Button Control.

Setup of Fast Switch includes setup of data type (step2), image (step 4) and control insertion (step 9) only. Default value is used for items in the other steps. You can add Toggle Button Control by Fast Switch and then modify it.

23.1.11 Addition of Toggle Lamp Control

Toggle Lamp Control is used to change the color of the control if the value of a Register, System Variable and so on fulfills the specified condition. Procedure to add Toggle Lamp Control is following.

Procedure 23-10 Addition of Toggle Lampe Control

Step

1 Select Toggle Lamp in *i*Pendant Control selection screen (Fig. 23.1.2).
2 Press of F4, NEXT leads you to data type setup screen. Please refer to Section 23.1.4.
3 Press the F4, NEXT. The following screen will be displayed.

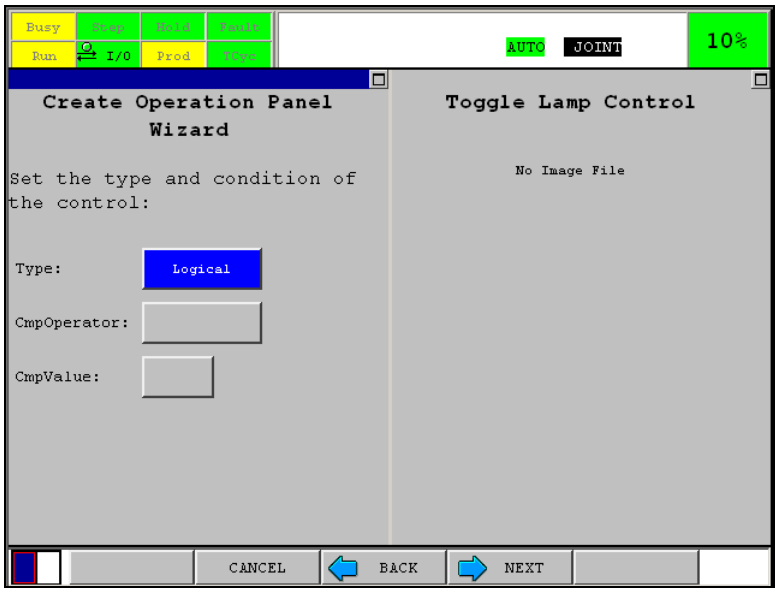


Fig. 23.1.11 Toggle lamp setup screen

Table 23.1.11 (a) Items in toggle lamp setup screen

Item	Description
Type	This specifies data type to process. Logical: Data is transacted as bool type. Numerical: Data is transacted as numerical type. The setting of CmpOperator and the value of CmpValue is applied.
CmpOperator	This is active when Type is Numerical. This selects the condition expression evaluated as TRUE. EQ: Specify the equal case (=). NE: Specify the not equal case (<>). LT: Specify the less than case (<). LE: Specify the less than or equal case (<=). GT: Specify the greater than case (>). GT: Specify the greater than or equal case (>=).
CmpValue	This specifies the standard value for comparison. Result of comparison between read value and this value decides status of TRUE or FALSE. It is reflected to display of lamp.

4 Press of F4, NEXT leads you to image setup screen. Please refer to Section 23.1.4.
5 Press of F4, NEXT leads you to Fig. 23.1.10 (b): True/False color setup screen (Fig. 23.1.10 (b)). Please refer to Section 23.1.10.
6 Press of F4, NEXT leads you to color setup screen. Please refer to Section 23.1.6.
7 Press of F4, NEXT leads you to size setup screen. Please refer to Section 23.1.3. Caption is used as fixed string. It is not used as format.
8 Press of F4, NEXT leads you to style setup screen. Please refer to Section 23.1.6.
9 Press of F4, NEXT lead you to control insertion screen. Please refer to Section 23.1.3.
10 Decide position to insert and press the F4, NEXT. This is end of addition of Toggle Lamp Control.

Setup of Fast Lamp includes setup of data type (step2), image (step4) and control insertion (step9) only. Default value is used for items in the other steps. You can add Toggle Lamp Control by Fast Lamp and then modify it.

Background color used by Toggle Lamp depends on its View type.

Table 23.1.11 (b) Background color of Toggle Button

View type	Status	Background color
Panel	True	Background when TRUE
	False	Background when FALSE
Circle, Fixed Circle	True	Circle part: Background when TRUE (If transparent is checked, black) The other part: Background
	False	Circle part: Background when TRUE (If transparent is checked, black) The other part: Background
Image	Both true and false	Image uses background color if it uses transparent color. 1 Background when TRUE (False) if transparent is not checked 2 Background if transparent is not checked 3 Otherwise background color of web page

23.1.12 Modification of Panel

Use procedure 23-11 to modify created panel.

Procedure 23-11 Modification of panel

1 Start panel wizard by procedure 23-1 and select panel.
2 If selected panel already exists, the following screen will be displayed.

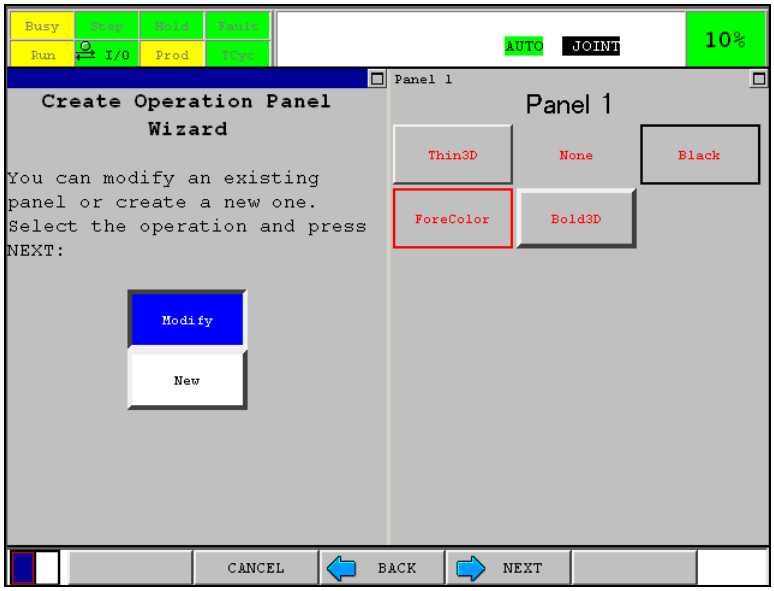


Fig. 23.1.12 (a) Selection of modification or creation

3 Select modify and press the F4, NEXT. The following screen will be displayed.

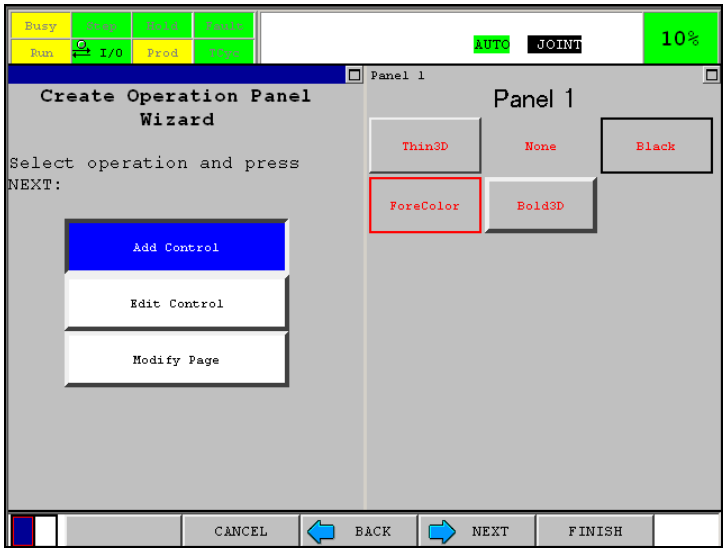


Fig. 23.1.12 (b) Selection of operation screen

Table 23.1.12 Items in selection of operation screen

Item	Description
Add Control	You can add control to panel.
Edit Control	You can modify control already added to panel. Please refer to section 23.1.13.
Modify Page	You can modify other property of panel. For example, title of panel, background color. Please refer to section 23.1.16.

4 Select operation and press the F4, NEXT.

For detail of each operation, please refer to section written in table above.

23.1.13 Modification of Control

Use procedure 23-12 to modify control that is already added to panel.

Procedure 23-12 Modification of control

1 Display selection of operation screen by procedure 23-11.
2 Select Edit Control.
3 Press the F4, NEXT. The following screen will be displayed.

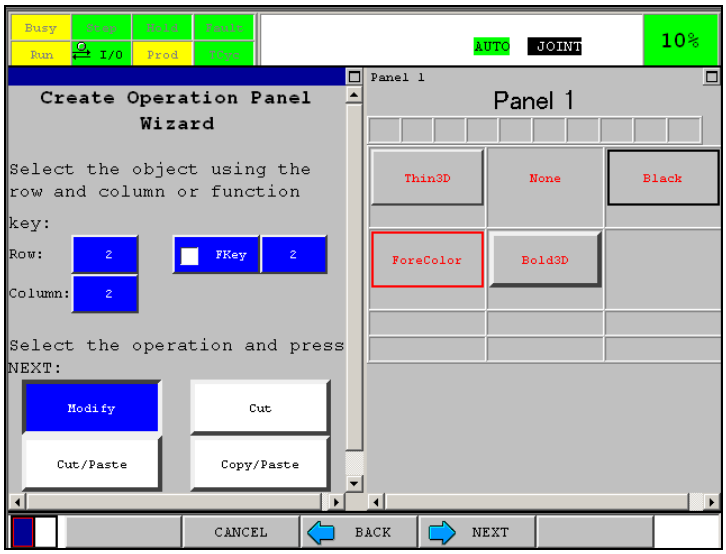


Fig. 23.1.13 Edit control screen

Table 23.1.13 Items in edit control screen

Item	Description
Row, Column	These specify control to modify, cut, and copy.
Fkey	If this is checked, control assigned to function key is selected.
The number next to Fkey	This number specifies function key when Fkey is checked.
Modify	Select this to modify control already added to panel.
Cut	Select this to delete control.
Cut/Paste	Select this to cut selected control and paste it elsewhere.
Copy/Paste	Select this to copy selected control and paste it elsewhere.

4 Select Modify and press the F4, NEXT.
5 Setup screen appropriate for the selected control is displayed. If you added the control by Fast label, Fast switch or Fast lamp, setup screen for corresponding control is displayed. For example, setup screen for Label control is displayed for control added by Fast label.
6 Setup control just same as addition of control. Control insertion screen at the last step is not displayed.

23.1.14 Delete of Control

Use procedure 23-13 to delete control.

Procedure 23-13 Delete of control

1 Display Edit control screen by procedure 23-12.
2 Select control to delete.
3 Select Cut and press the F4, NEXT.
4 The following screen will be displayed.

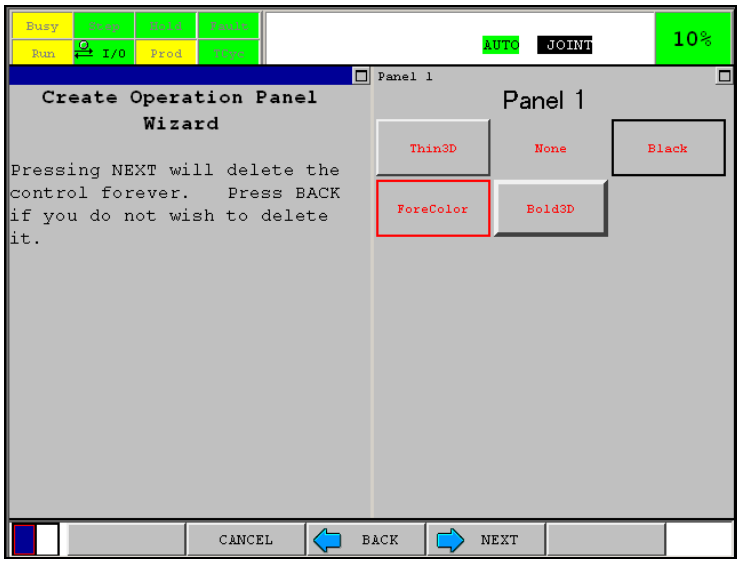


Fig. 23.1.14 Confirmation of delete of control

Press the [F4] key to delete. Otherwise, press the [F3] key.

NOTE
F2 CANCEL cancels panel wizard. Press of the [F2] key in this screen causes delete of control.

23.1.15 Cut/Copy Paste of Control

You can move added control by cut and paste. You can copy added control by copy and paste. Use procedure 23-14.

Procedure 23-14 Cut/Copy paste of control

1 Display Edit control screen by procedure 23-12.
2 Select control.
3 If you want to move the control, select Cut/Paste. If you want to copy it, select Copy/Paste. Press the F4, NEXT.
4 Next screen depends on control you selected.
4-1 If selected control is NOT assigned to function key, the following screen will be displayed.

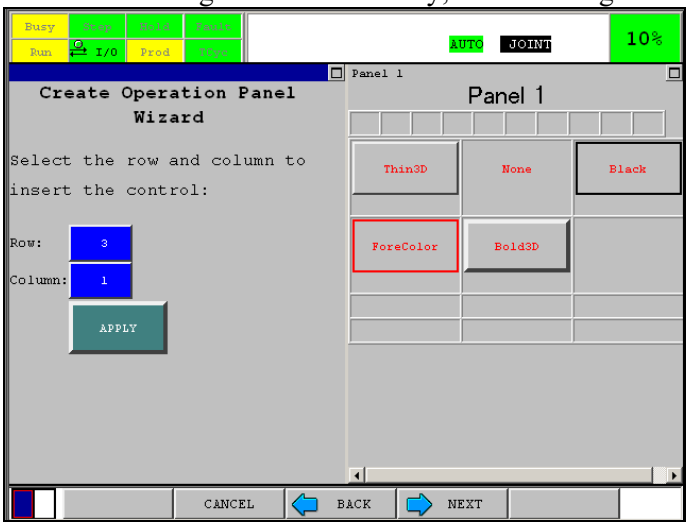


Fig. 23.1.15 (a) Control insertion screen

4-2 If selected control is assigned to function key, the following screen will be displayed. This screen is to select function key to paste control.

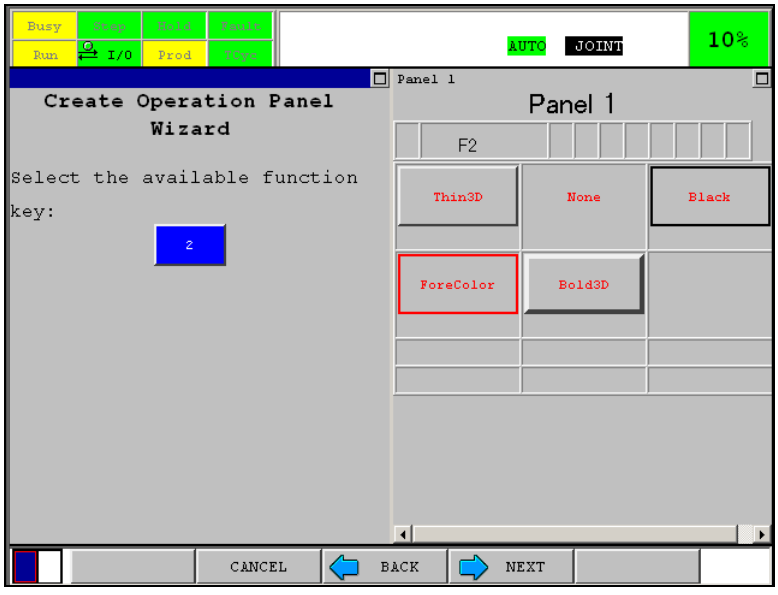


Fig. 23.1.15 (b) Control insertion screen (function key)

5 Decide position to insert control and Press the F4, NEXT.

23.1.16 Modification of Page

Background color and title can be modified.

Procedure 23-15 Modification of page

1 Display selection of operation screen by procedure 23-11.
2 Select Modify page.
3 Press the F4, NEXT. The following screen will be displayed.

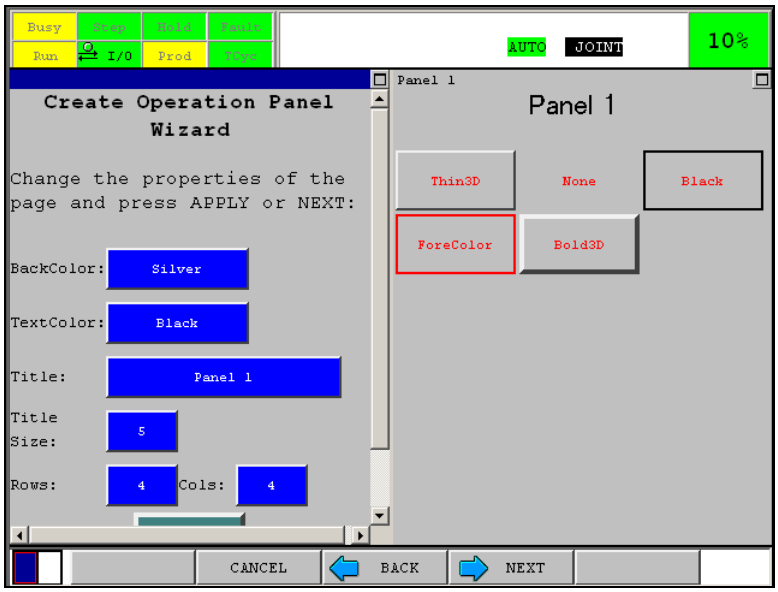


Fig. 23.1.16 Page edit screen

Table 23.1.16 Item in page edit screen

Item	Description
BackColor	Background color of page
Text Color	Color of text. Title of color uses this.
Title	This is title displayed at top of panel. If panel1, it defaulted to Panel 1.
Title Size	Size of title. Range is from 1 to 5.
Rows	Control is placed in invisible table. This item is the number of rows of the table. You cannot decrease it.
Cols	The number of columns of invisible table. You cannot decrease it.
APPLY	This is not item to setup. Press of the button applies property edited and you can see rough image in right pane.

4 Setup as required and Press the F4, NEXT. This is the end of modification of page.

23.1.17 Re-creation of Panel

Use procedure 23-16 to re-create panel with disposing existing panel.

Procedure 23-16 Re-creation of panel

1 Display panel selection screen by procedure 23-1.
2 Select panel to re-create and press the F4, NEXT.
3 Screen to select modification or creation is displayed.
4 Select New.

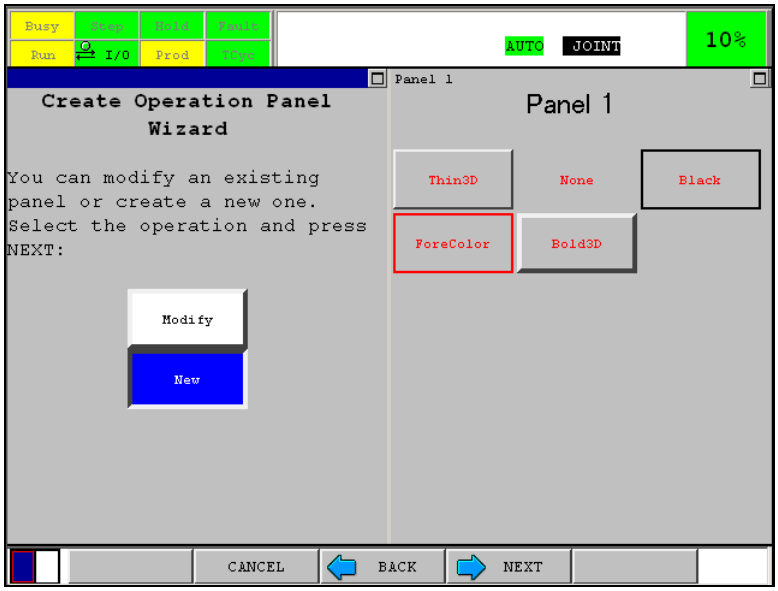


Fig. 23.1.17 (a) Selection of modification or creation

5 Press the F4, NEXT. The following screen will be displayed.

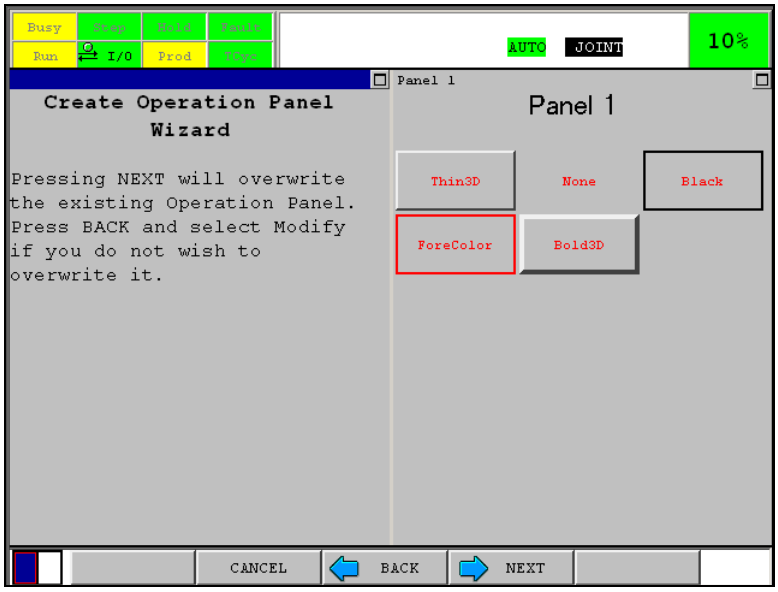


Fig. 23.1.17 (b) Confirmation of re-creation of panel

6 Press of F4 delete existing panel. Re-creation starts from setup of the number of rows and columns.

23.2 RUN KAREL PROGRAM BY PANEL

You can create button to run KAREL program. Required steps are following.
(A) Creation of KAREL program that can be run by panel
(B) Addition and setup of Button Change Control

NOTE
You cannot run arbitrary KAREL program. There is caution of creation of KAREL program.

23.2.1 Caution for Creation of KAREL Program

(A) Create static (program level) INTEGER variable, return_code. Set it to 204.
(B) Program must complete in a short time.
(C) If it takes long time to complete, run program by RUN_TASK built-in by another program that was run by button change control.
(D) Specify %NOLOCKGROUP.

23.2.2 Creation of Run Button

Use procedure 23-17 to create button to run program.

Procedure 23-17 Creation of run button

1 Start addition of Button Change Control and display URL setup screen.
Please refer to Subsection 23.1.6 for procedure to add Button Change control.

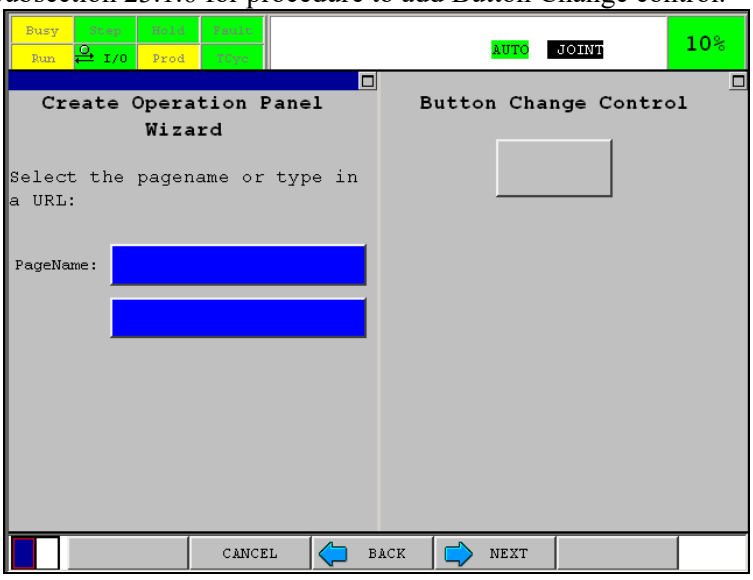


Fig. 23.2.2 (a) URL setup screen

2 Press the lower side button and input URL in format of "/KAREL/program name".
For example, /KAREL/TESTPROG to run KAREL program TESTPROG.

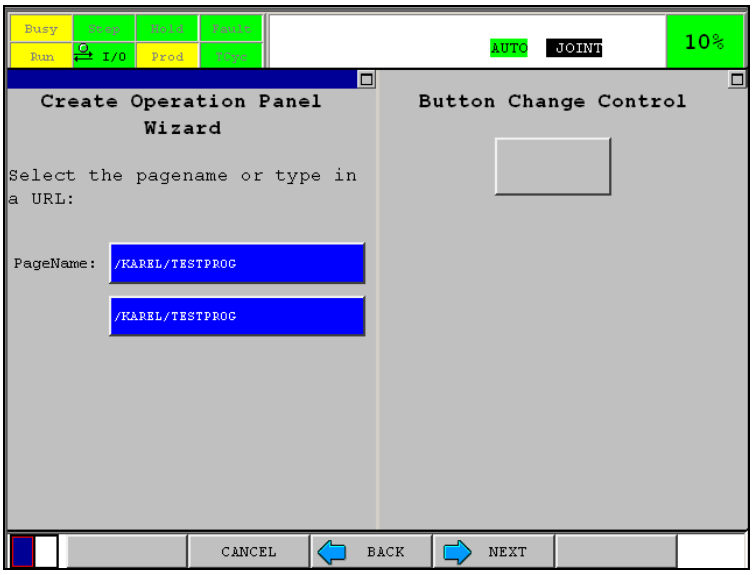


Fig. 23.2.2 (b) Setup to run TESTPROG

23.3 USAGE

23.3.1 Display of Panel

MENU→BROWSER→[TYPE] displays panels created.

> **NOTE**
> Created panels are used in all languages. If you display the panel in more than one language, create the panel in English. Japanese panel is not displayed correctly in English.

If there are too many entries in [TYPE] menu of BROWSER to add panel to the menu, the following screen will be displayed at finishing addition of panel .

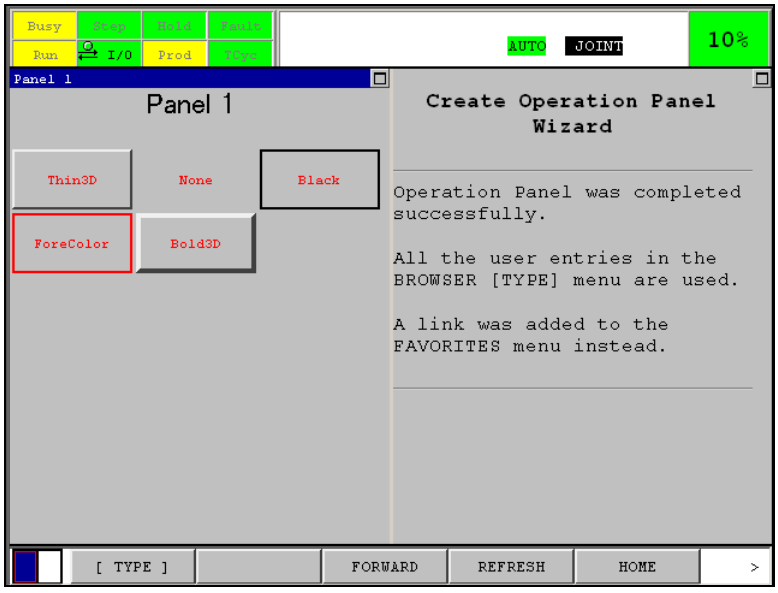


Fig. 23.3.1 (a) Notification of addition to favorite

In this case, panel is added to link in FAVORITE. Display panel by selecting link.

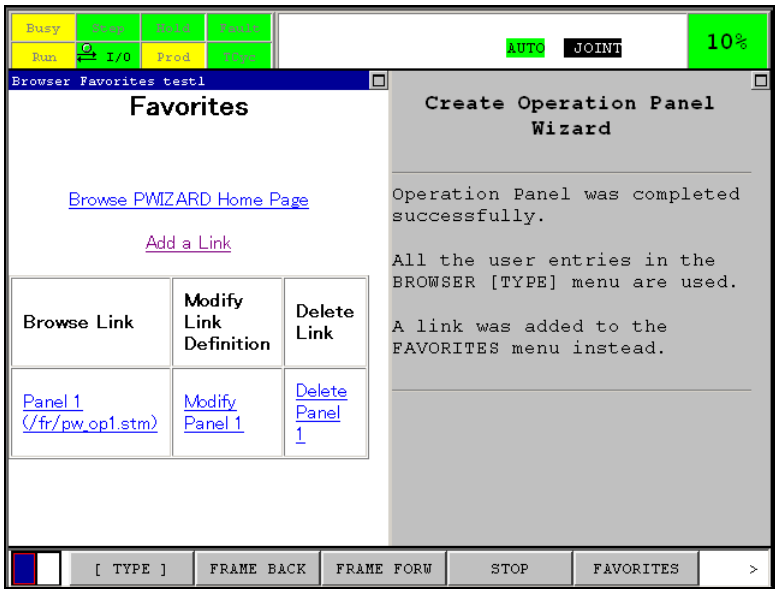


Fig. 23.3.1 (b) Panel1 added to favorite

23.3.2 Backup/Restore

Created panels can be saved and restored in file screen. It is categorized as “Application”.
Panel1…Panel 4 corresponds to PW_OP1.STM…PW_OP4.STM, respectively.

NOTE
PANEL1.STM,PANEL1.DT,……,PANEL4.STM, PANEL4.DT have nothing to do with this function.

- If you use image file you prepared, use it in GIF or JPG format. Copy them to FR: and refer to it. FR: *.GIF and FR: *.JPG are saved and restored in file screen. They are also in category of “Application”.

- Registration data to menu of MENU→BROWSER→[TYPE] is saved in SYSVARS.SV. If you want to backup/restore the registration, SYSVARS.SV should be backed up and restored.

- If you do not want to restore SYSVARS.SV, you have to register panel to [TYPE] menu manually. Use procedure below.
  - Restore backup of panel. (PW_*.STM and required image files)
  - Display Selection of operation screen (Fig. 23.1.12 (b)) by procedure 23-11.
  - Press the F5, FINISH. This means there is actually no change.
  - These steps add panel to [TYPE] menu of BROWSER.

24 ENHANCED MIRROR IMAGE

The enhanced mirror image option (A05B-2600-R698) allows you to translate an entire teach pendant program or portion of a teach pendant program to mirror-image the original programmed points. This option can be used to teach symmetrical parts easily. The enhanced mirror image option provides additional functionality so that existing frames such as user frames can be used directly with any specified XY, YZ, or XZ plane. Also, additional orientation shifting is provided for different application requirements.

> ⚠ **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.
Mirror Frame (used only for “FRAME” method)	World, User Frame, or Jog Frame can be selected as the mirror frame.

ITEM	DESCRIPTION
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).

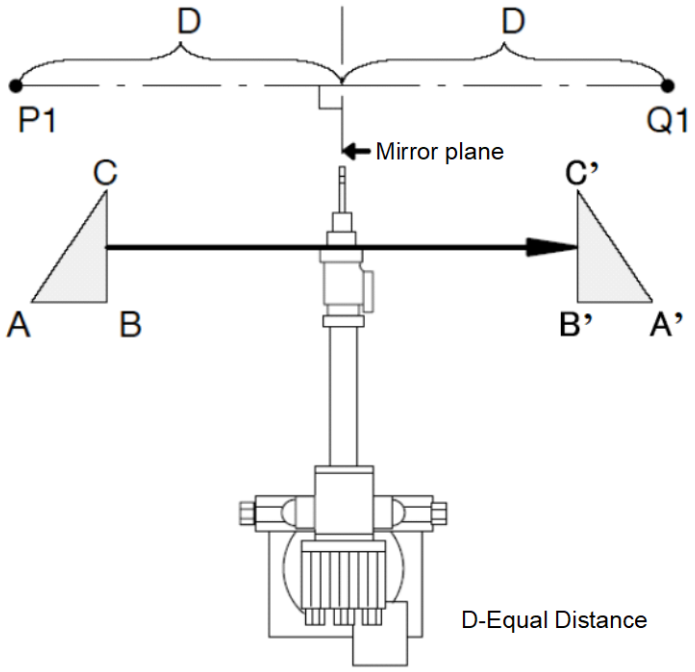


Fig. 24.1 (a) Parallel mirror image with mirror plane in center of robot

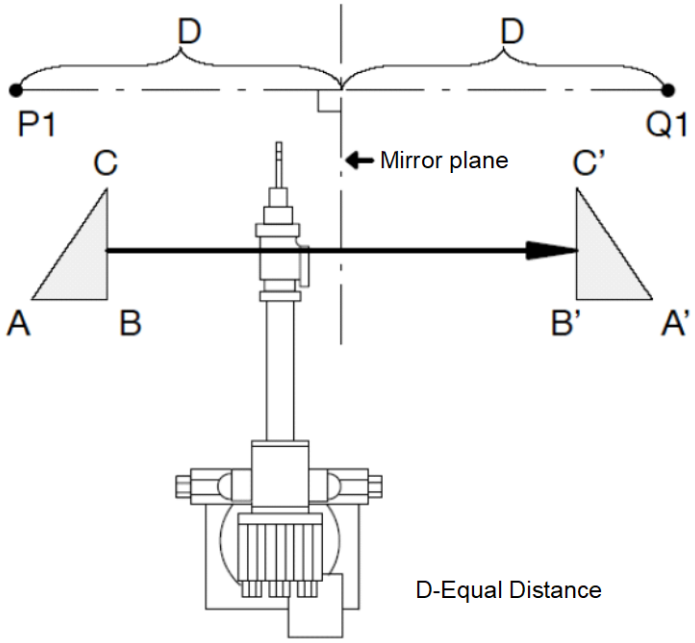


Fig. 24.1 (b) Parallel mirror image with mirror plane offset from center of robot

> ⚠ **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).

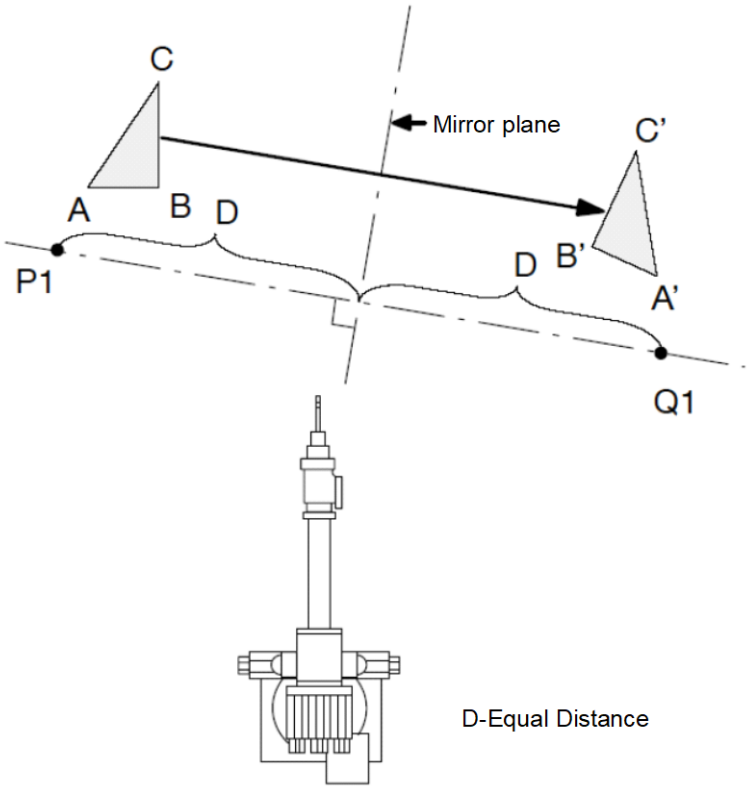


Fig. 24.1 (c) Parallel mirror image with offset

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''.

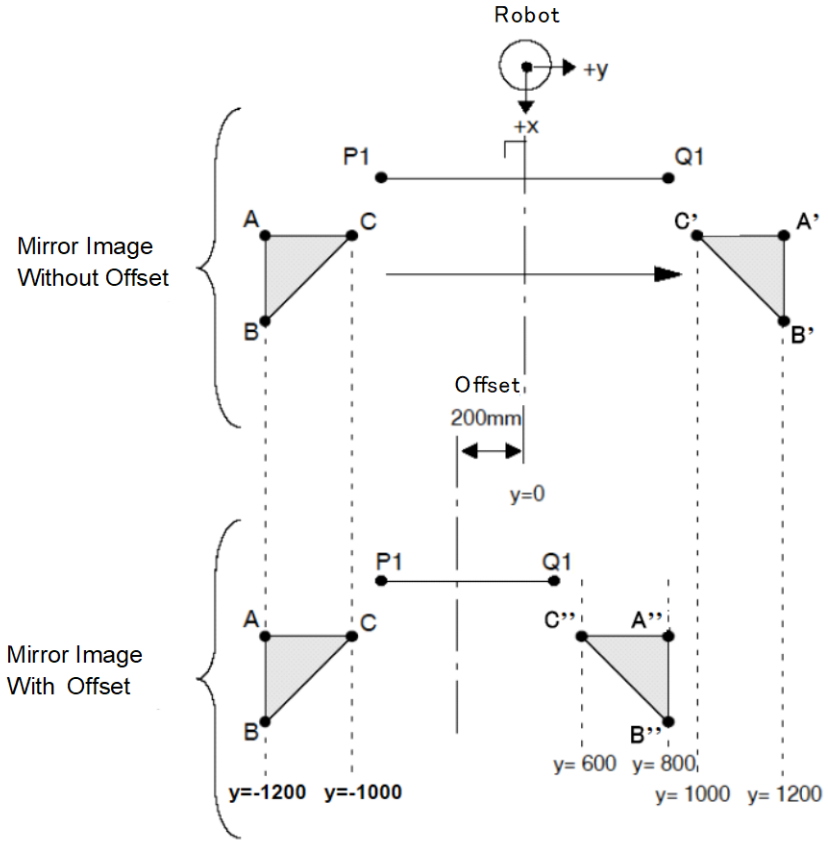


Fig. 24.1 (d) Positional mirror image

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).

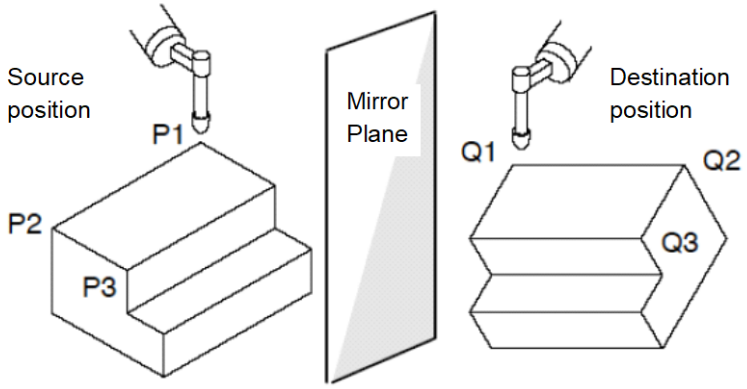


Fig. 24.2 (a) Rational mirror image

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'.

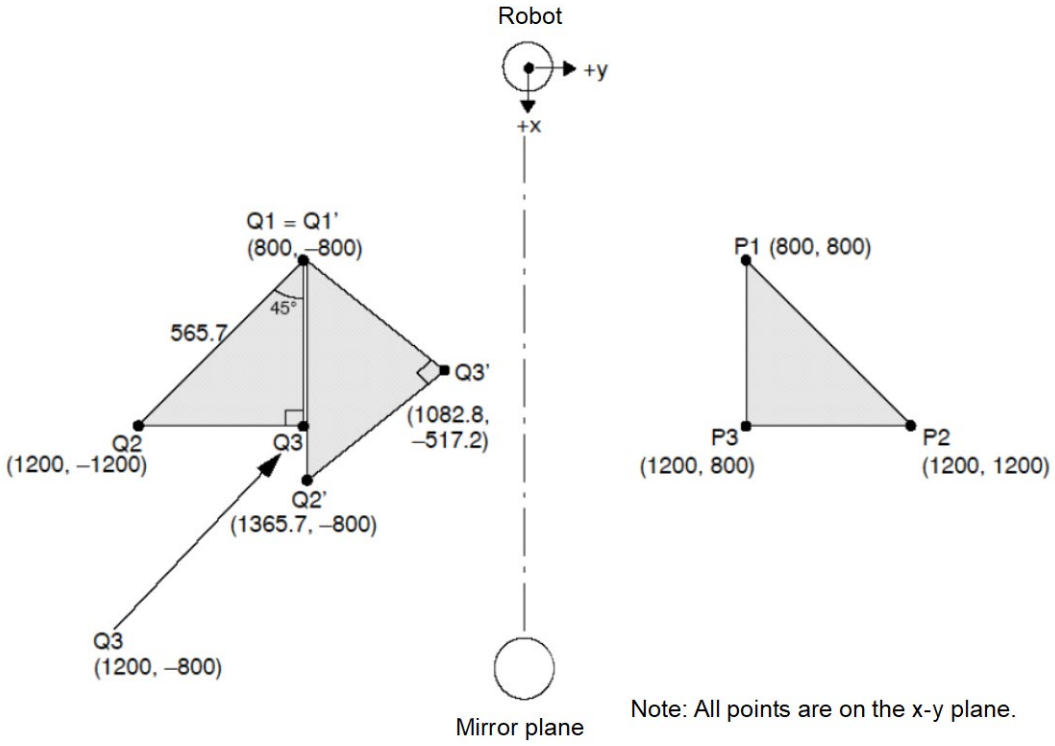


Fig. 24.2 (b) Rational mirror image

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).

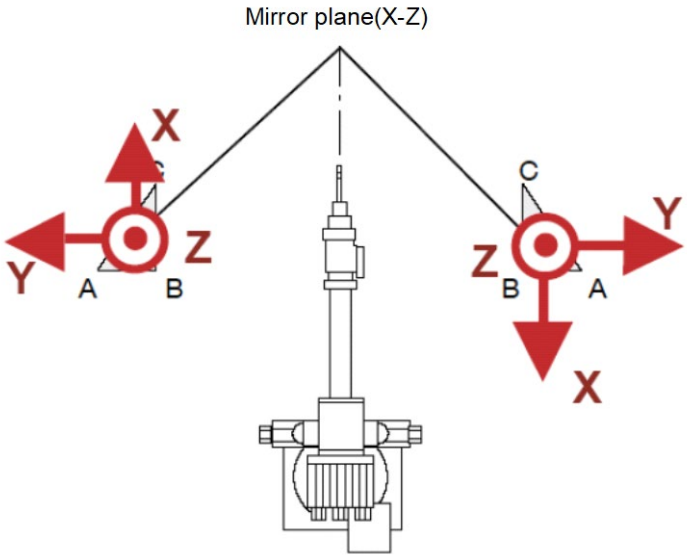


Fig. 24.3 (a) Mirror image shift with orientation mirrored (mirror method)

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).

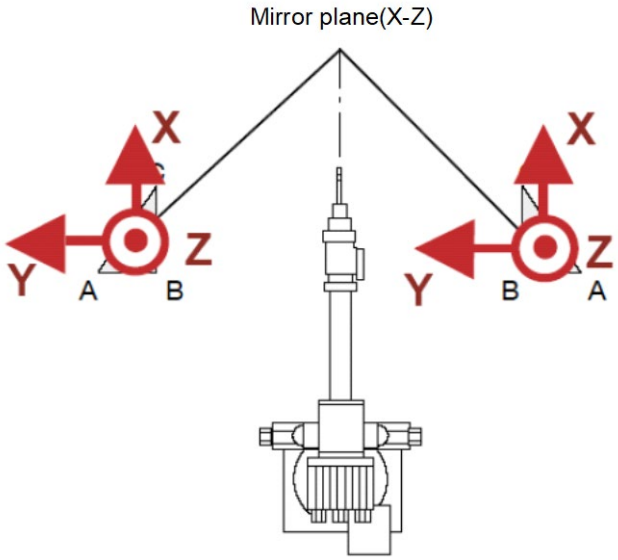


Fig. 24.3 (b) Enhanced mirror image with orientation controlled (fixture method)

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. Fig. 24.4 (a) displays a key for use with Fig. 24.4 (b) through Fig. 24.4 (d).

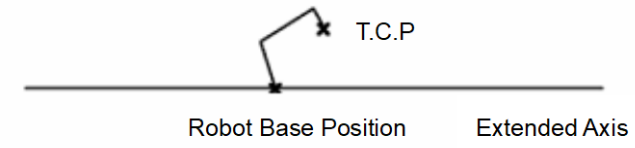


Fig. 24.4 (a) Mirror image key

Possible extended axes configurations are:

- **Robot axes only -** allows you to mirror the axes of the robot without mirroring any non-integrated extended axes such as a positioning table. The shift is calculated using the change in the robot tool center point (TCP). See Fig. 24.4 (b).

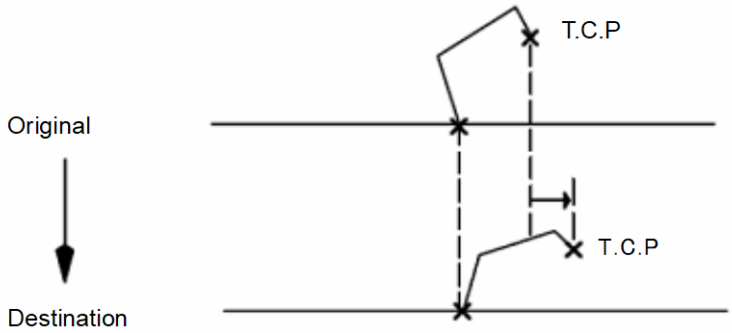


Fig. 24.4 (b) Example of robot axes only mirror image

- **Ext integrated -** allows you to mirror the axes of the robot and any integrated axes. The amount of mirror image for the robot and the extended axes is calculated using the change in the TCP. See Fig. 24.4 (c).

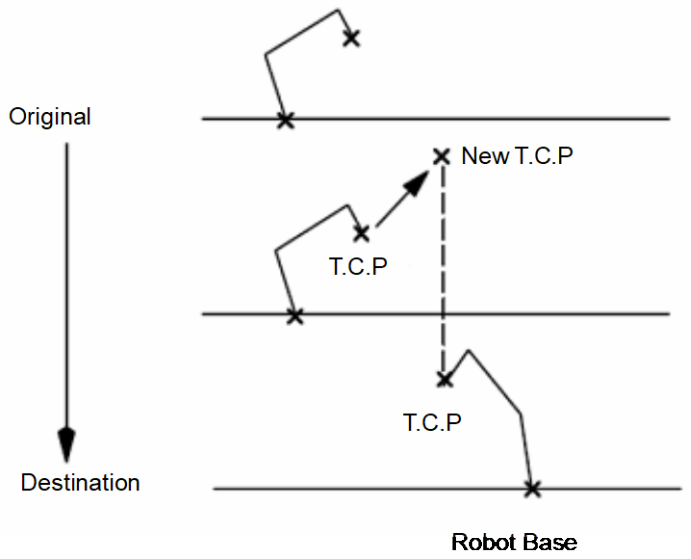


Fig. 24.4 (c) Example of extended axes integrated mirror image

- **With ext axes -** allows you to mirror positions for robot axes and any extended axes in your system. The amount of mirror image for the robot is calculated by using the change in the TCP. The amount of shift for the extended axes is calculated using the center of the difference between an original position (P1) and a new position (Q1) as the point where the mirror image occurs. See Fig. 24.4 (d).

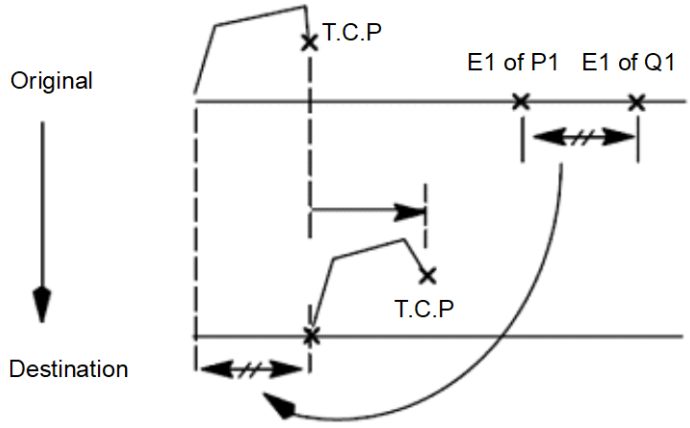


Fig. 24.4 (d) Example of with extended axes mirror image

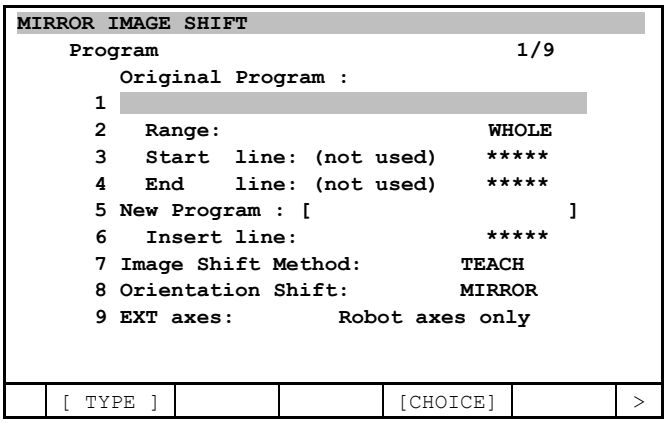
Procedure 24-1 Using Enhanced Mirror Image Shift

Conditions

- The program you want to mirror has been created and contains recorded positions.
- All robot joint axes are at zero degrees.

Steps

1 Press 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.

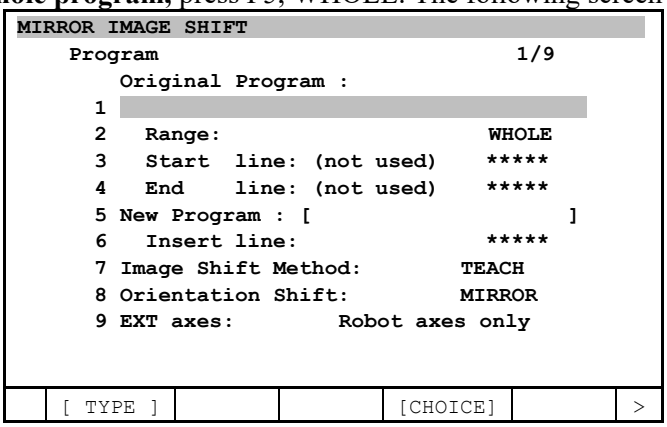


5 Move the cursor to 1 under Original Program. If the program you want to mirror is not selected, press F4, [CHOICE]. Select the program from the list and press the [ENTER] key.

NOTE
The last program selected using the SELECT menu will automatically be named as the original program.

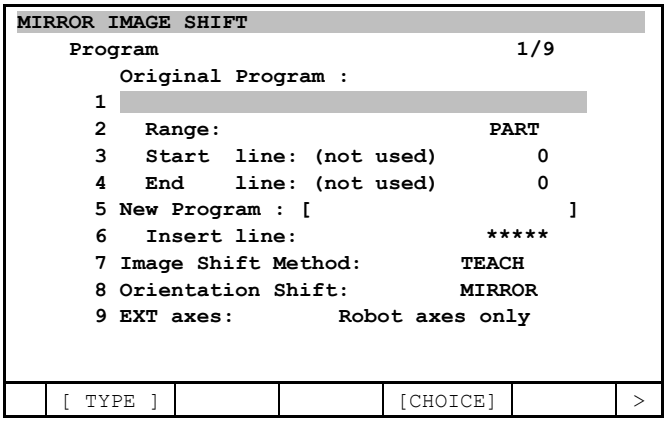
6 Move the cursor to Range. Use the function keys to select whether you want to mirror all or part of the original program.

- **To mirror the whole program,** press F5, WHOLE. The following screen will be displayed.



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.



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).

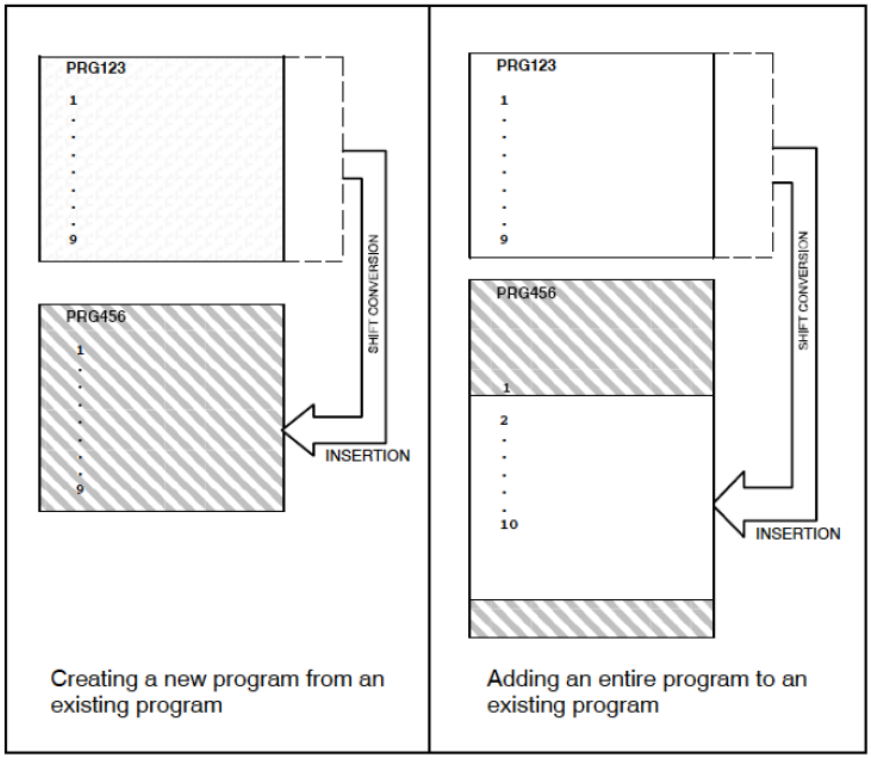


Fig. 24.4 (e) Mirroring an entire program

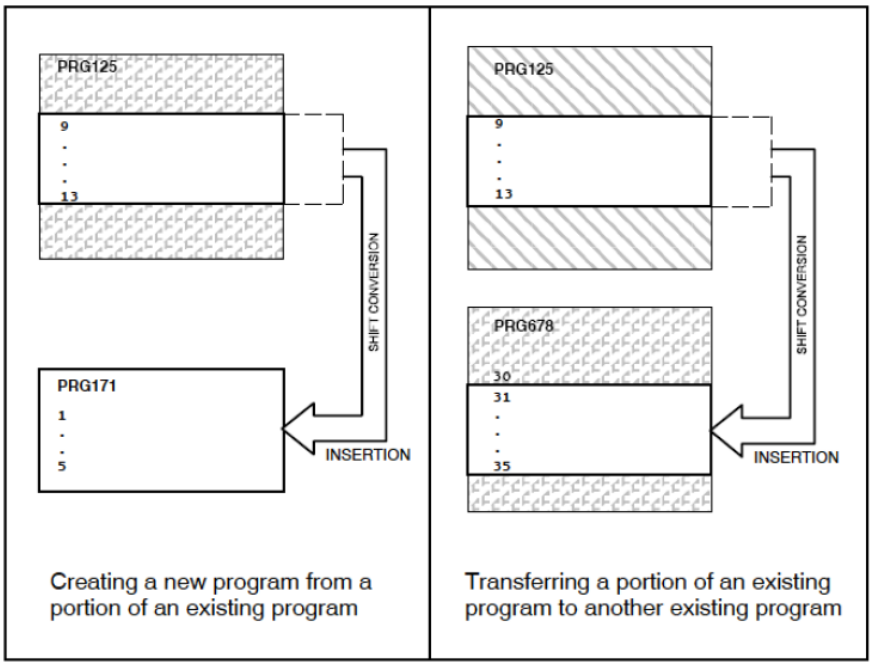


Fig. 24.4 (f) Mirroring a portion of a program

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:

```
MIRROR IMAGE SHIFT
    Shift amount/Teach                  1/3
      Position data
      X :********  Y :********  Z :********

      1 Rotation:                       OFF

      2 Source position         P1:

      3 Destination position    Q1:

[ TYPE ] | EXECUTE |  | ON | OFF | >
```

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:

```
MIRROR IMAGE SHIFT
    Shift Frame Settings                1/3


      1 Mirror Frame:                 WORLD
      2   Frame No.:  (not used)      *****
      3 Mirror Plane:                    XY

[ TYPE ] | EXECUTE |  | [CHOICE] |  |
```

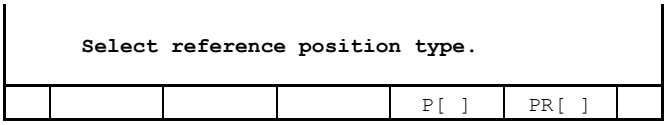
Continue to Step 14.

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.

```
MIRROR IMAGE SHIFT
    Shift amount/Teach                  1/3
      Position data
      X :********  Y :********  Z :********

      1 Rotation:                       OFF

      2 Source position         P1:

      3 Destination position    Q1:

[ TYPE ] | EXECUTE |  | ON | OFF | >
```

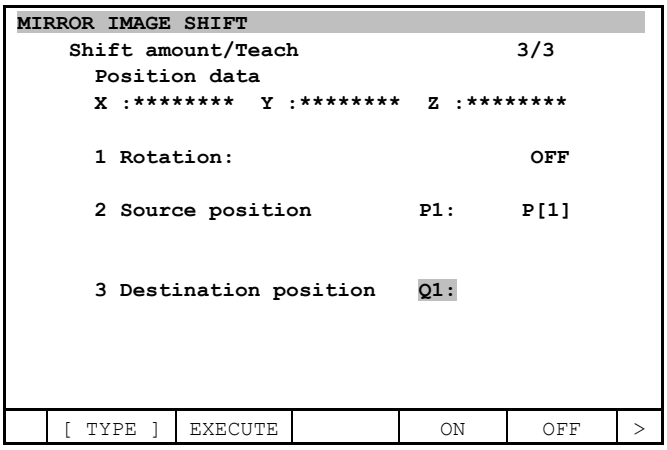
b. Move the cursor to source position (see Fig. 24.1 (a)).
c. Move the robot to the source position (P1) and either record or specify the position:

- **To record a position,** jog the robot to the position you want, press and hold in [SHIFT] key and press F5, RECORD.
- **To specify a previously recorded position or position register,** press F4, REFER. Type the number of a previously defined position or position register, and press [ENTER] key.

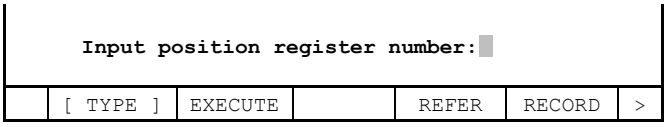


- **To clear a position setting,** press the [NEXT] key, >, and then press F1, CLEAR. At the prompt "Clear all data," press F4, YES, to confirm.

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.

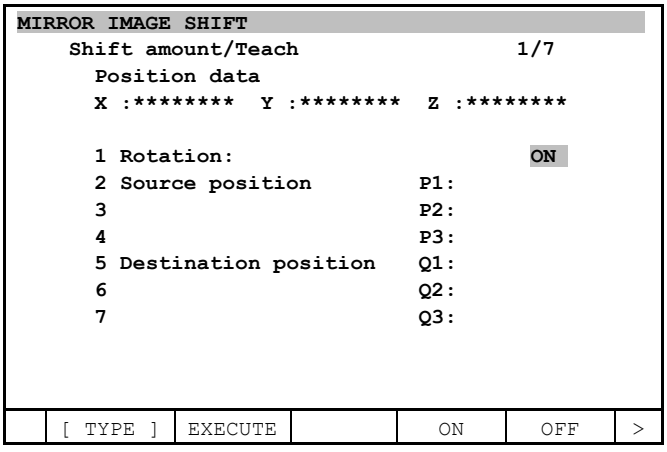


- **To clear a position setting,** press NEXT, >, and then press F1, CLEAR. At the prompt "Clear all data," press F4, YES, to confirm.

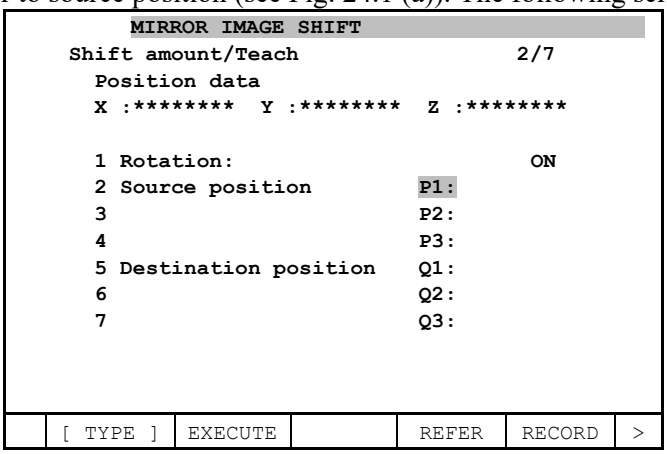
e. Record or specify the destination position.

13 If you want to rotate the positions,

a. Move the cursor to Rotation and press F4, ON. The following screen will be displayed.

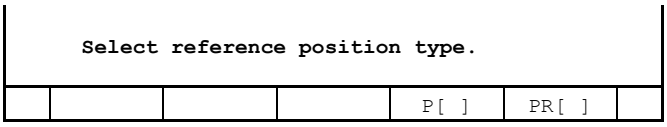


b. Move the cursor to source position (see Fig. 24.1 (a)). The following screen will be displayed.



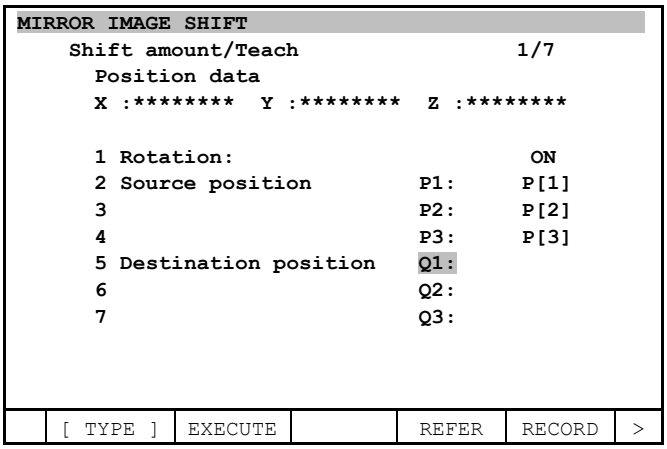
c. Move the robot to the first source position (P1) and either record or specify the position:

- **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.

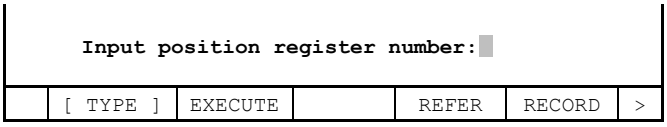


- **To clear a position setting,** press NEXT, >, and then press F1, CLEAR. At the prompt "Clear all data," press the F4, YES, to confirm.

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.

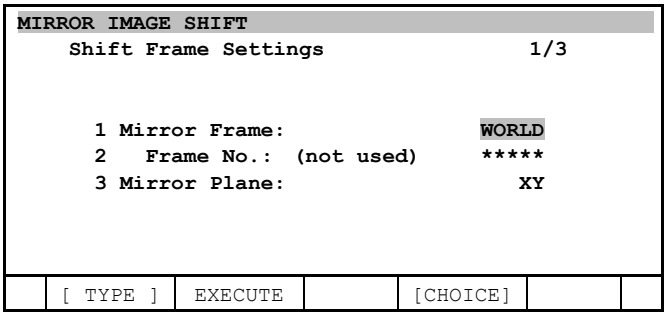


- **To clear a position setting,** press NEXT, >, and then press F1, CLEAR. At the prompt "Clear all data," press F4, YES, to confirm.

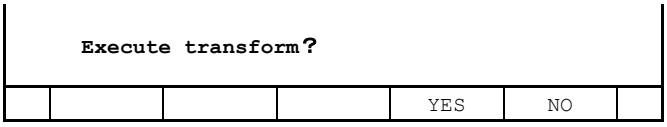
f. Record or specify all the destination positions.

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.

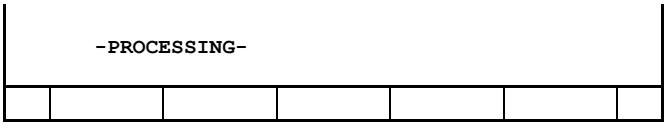


15 If you selected USER or JOG, move the cursor to Frame No. and type the number of the existing frame you want to use.
16 Move the cursor to Mirror Plane and press the F4, [CHOICE]. Select the plane you want to use and press the [ENTER] key.
17 **To mirror image the program,** press the F2, EXECUTE.

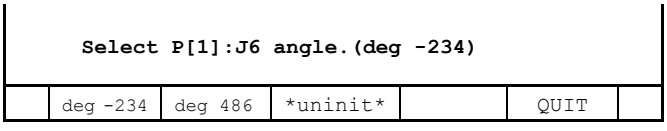


- **To execute the mirror image shift,** press the F4, YES.
- **If you do not want to execute the mirror image shift,** press the F5, NO.

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 CUSTOMIZE SUPPORT FUNCTION

In Software 7DC1 and 7DD0 software series Customize Support Function (J742) is included in KAREL Use Support Function.

This function supplies the necessary function for the one who develops the system specific function.
It supports following.

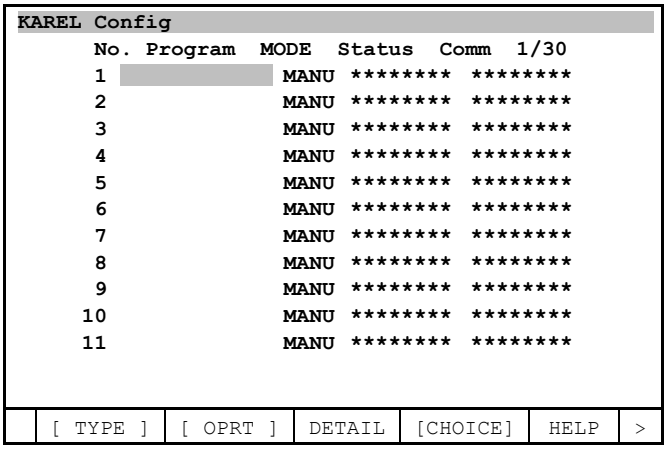
- Executing or aborting KAREL programs
- Auto running the KAREL programs at the time of power on
- Periodically monitor and output the specified KAREL program status
- Set the configuration of custom menu setting (cf. Section 25.3)

25.1 KAREL CONFIG

25.1.1 KAREL Config Screen

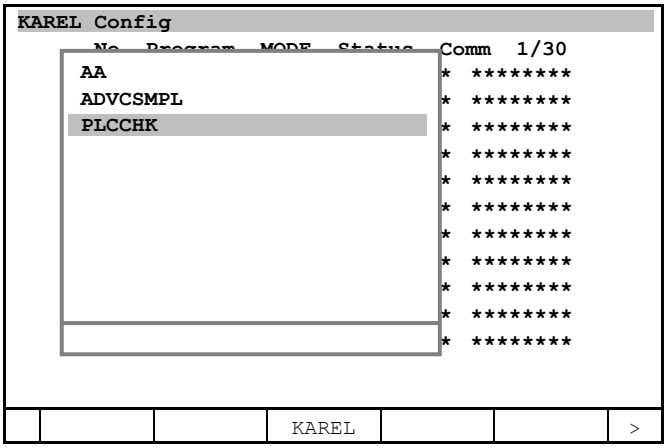
STEP

1 Select MENU – 6. SETUP – KAREL Config.
2 The following screen will be displayed.

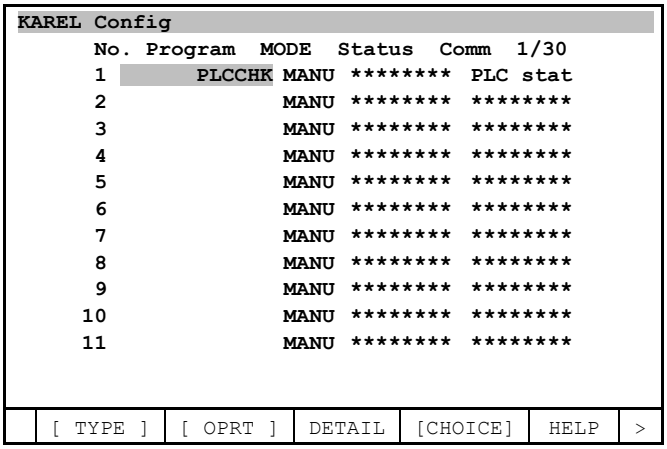


25.1.2 Use KAREL Config Screen

Select F4, [CHOICE], then you can see the list of KAREL program to be registered to run.



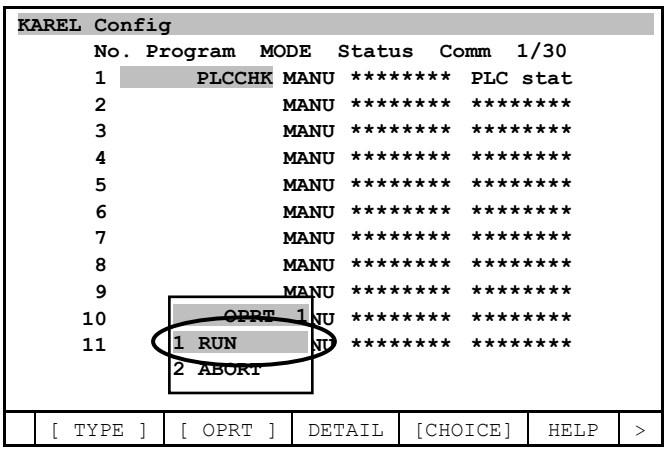
In this documentation, PLCCHK is used as the sample to explain. Set the cursor onto the PLCCHK, and press the [ENTER] key.



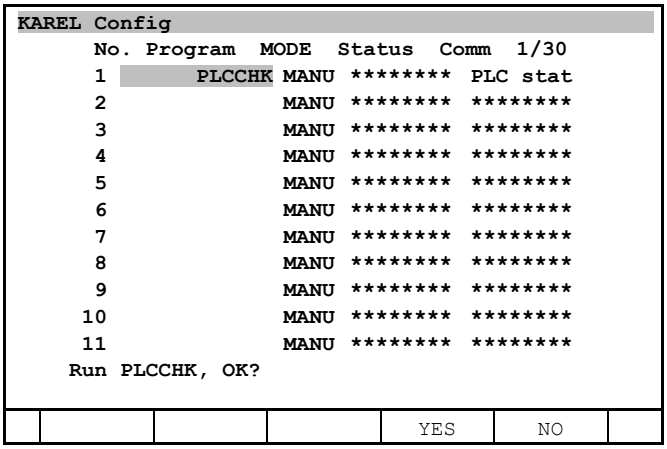
There are four status, running, paused, aborted and ********. ******** means that status is not other 3 ones.
Comm is the comment that is described in the KAREL source file as %COMMENT.

25.1.3 Run KAREL Program

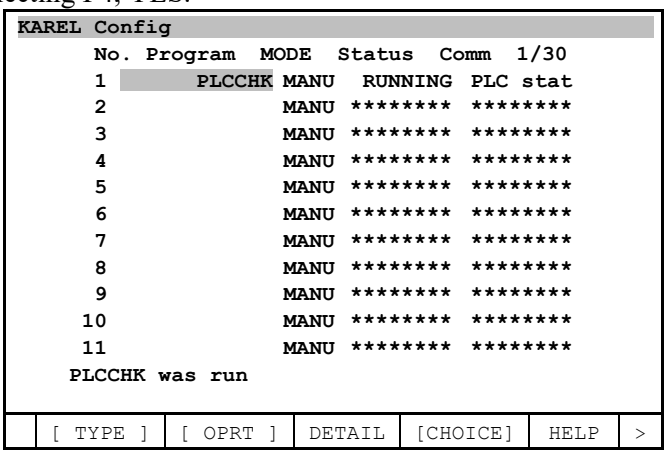
Set the cursor onto PLCCHK in Program row and press the F2, [OPRT].



Select 1. RUN. Then confirmation message is displayed.

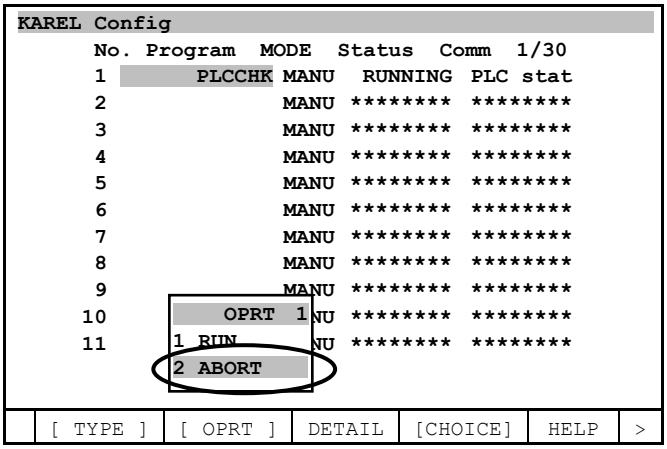


Program is run after selecting F4, YES.

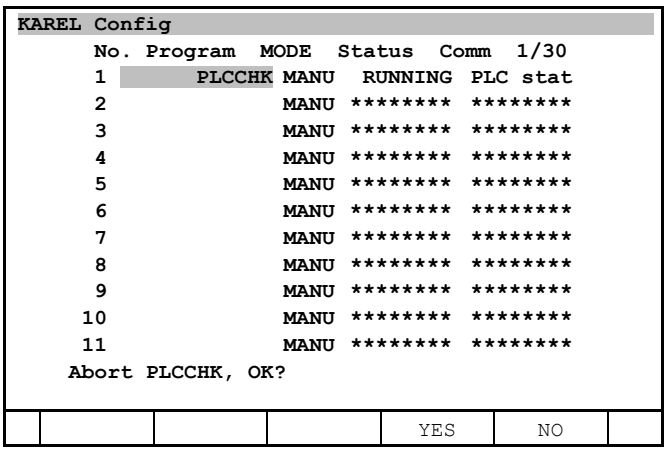


25.1.4 Abort KAREL Program

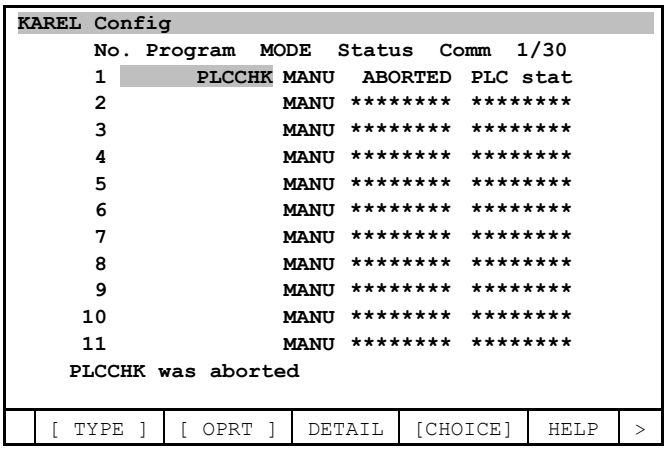
Set the cursor onto PLCCHK in Program row and press the F2, [OPRT].



Set the cursor onto 2. ABORT and press the [ENTER] key.
Confirmation message will be displayed.

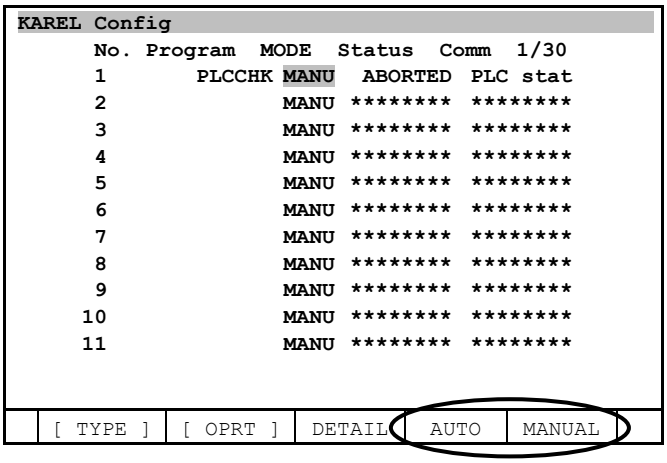


Select F4, YES, then KAREL program will be aborted.

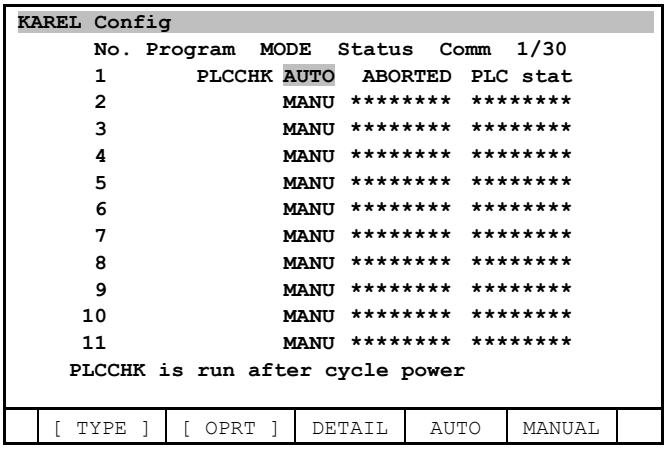


25.1.5 Start Mode Config of KAREL program

Set the cursor onto MODE row. F4 and F5 keys are turns to AUTO and MANUAL.



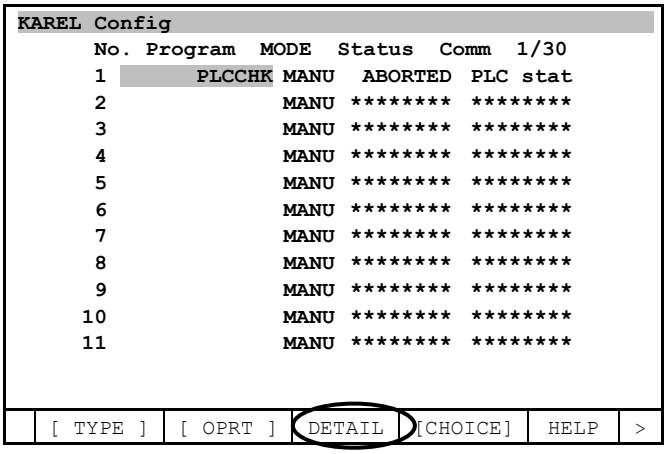
Default is MANUAL. If you select AUTO, KAREL program is run after cycling power of the controller.



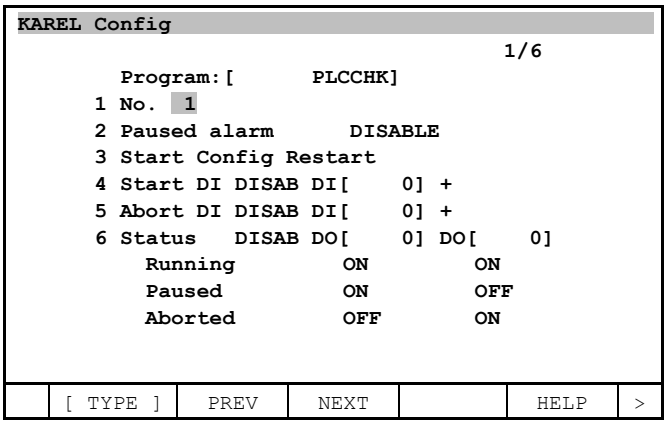
There is a limit that the number of running the KAREL simultaneously. It depends on the configuration of the robot system. “PROG-014 Max task number exceed” will be posted when the number of running KAREL is going to be over max number.

25.1.6 Detail Screen of KAREL Config

Press the F3, DETAIL.

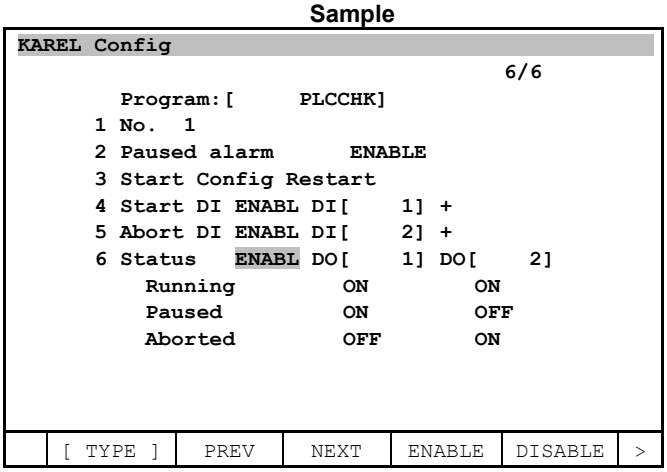


Like following screen will be displayed.



Item	Comment	
Paused alarm	If ENABLE, warning is posted when specified program, ex PLCCHK, is paused or aborted.	
	Paused	“SYST 294 Paused (program name)” “SYST 296 Specified KAREL was paused”
	Aborted	“SYST 295 Aborted (program name)” “SYST 297 Specified KAREL was aborted”
Start Config	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 can not be changed. If you would like to set them, please set “Start DI” disable once. After setting them, please set it enable. Then this configuration is enabled after cycle power.	
Abort DI	Sets the start operation. Detecting the rising edge (+) or falling edge (-) of specified DI starts the specified program. 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

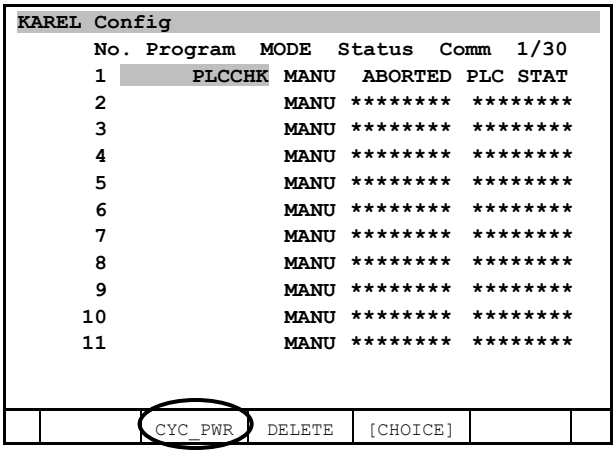
To switch previous screen, press the F2, PREV. In this case, screen is switched to No.30.
To switch next detail screen, press the F3, NEXT. In this case, screen is switched to No.2.

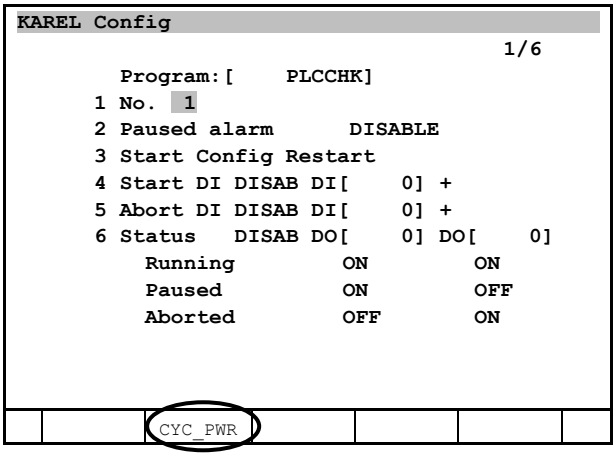
25.1.7 Limitation and Caution of KAREL Config

- Status is monitored at a constant frequency. Thus status is not changed at a real time. For example, program status is changed from running to paused, and aborted in one monitoring cycle, paused status is not detected at next monitoring cycle. Therefore status is changed from running to aborted in this case.
- Monitoring constant frequency can be changed. You can change it with system variable $KRL_NUM.$UPDT_TIME. It is able to change in the range 8 to 5000 msec. If you set it out of its range under 8 or above 5000, it is automatically set to 8 or 5000msec.
- In the case of above sample, after you set the detail of PLCCHK, return to KAREL Config screen with [PREV] key and change the registered No.1 program from PLCCHK to other program, such as C2, PLCCHK does not start if DI[1] is turned on. C2 is run instead. For prevention of miss operation, please do not change program after set the detail.

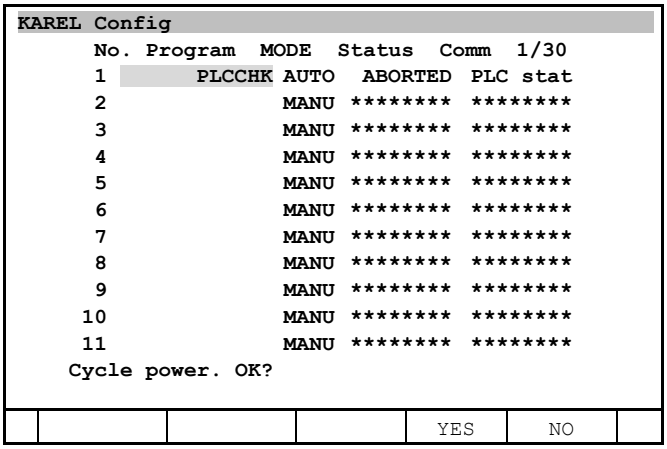
25.1.8 Cycle Power (R-30*i*B Controller)

Both KAREL config and detail screen enable to cycle power.
In R-30*i*B Mate controller, do cycle power manually.





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.

25.2 CUSTOM MENU

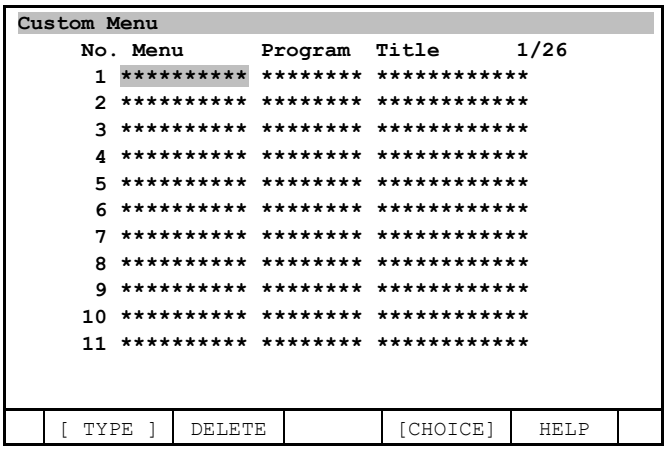
25.2.1 Overview

This function allows to set the TP or KAREL program to the menu. It is easy to run the specifed program to select the menu.

For example, user-created screen by KAREL program is allowed to display quickly by selecting the menu.

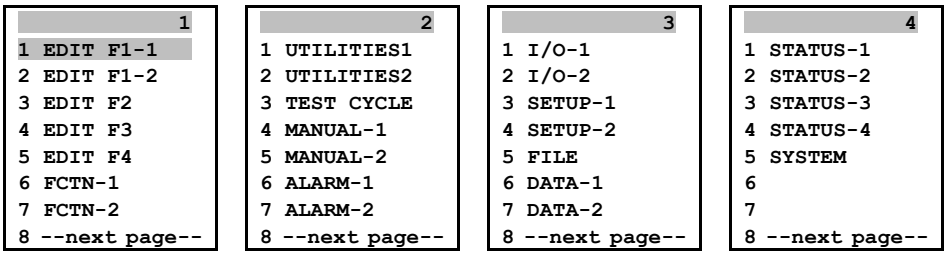
25.2.2 Starting Custom Menu

Select MENU – setup – Custom Menu.
The following screen will be displayed.

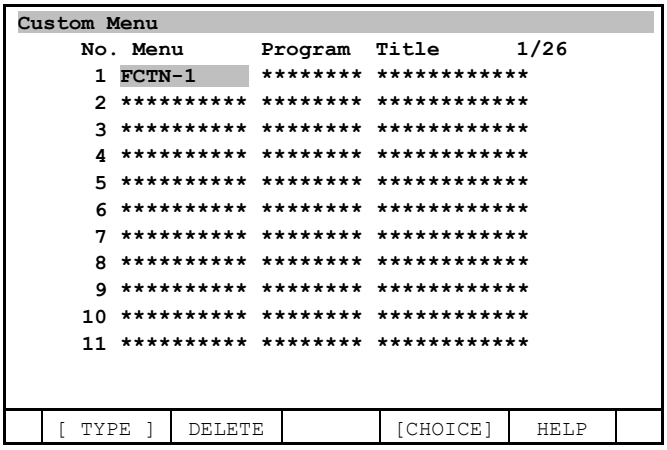


25.2.3 Set Custom Menu

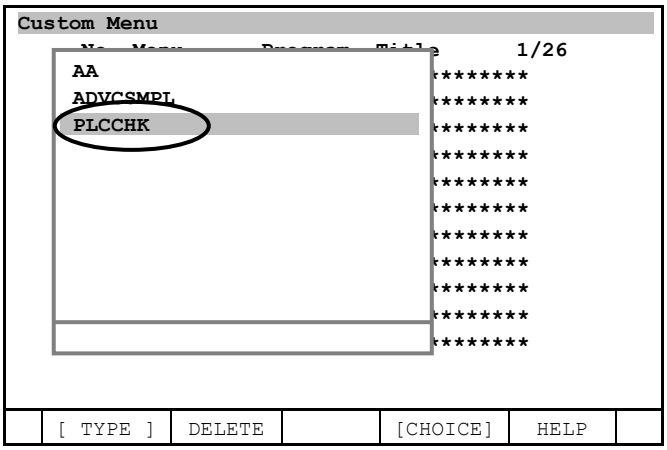
Press the F4, [CHOICE].



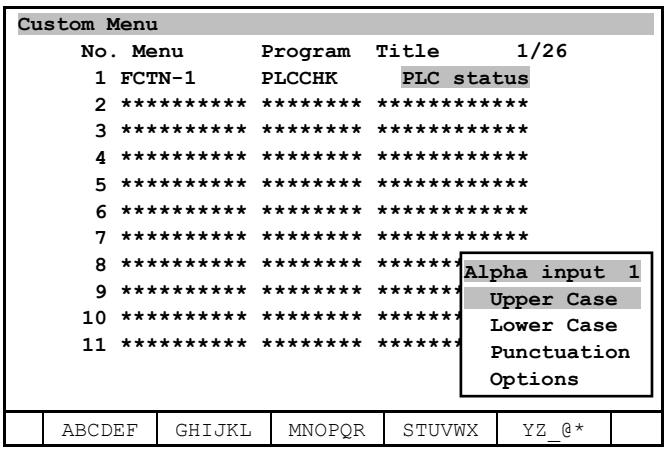
Select item that you would like to add. Select FCTN –1 as an example.



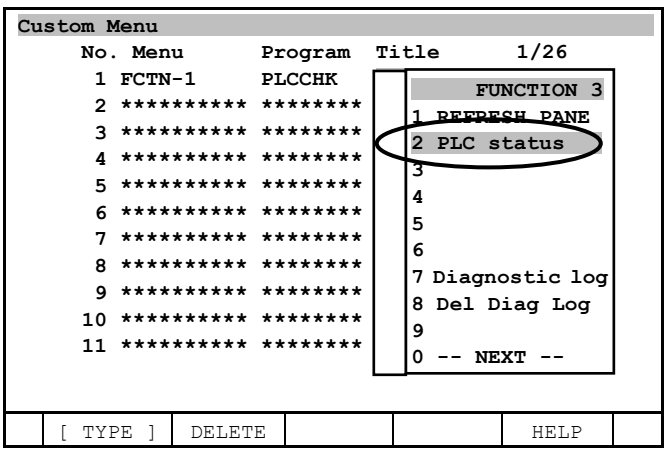
Set the cursor onto program row and press F4, [CHOICE] key. As an example PLCCHK is selected.



Next, set the cursor onto title row and press the [ENTER] key. PLCCHK is input for instance.

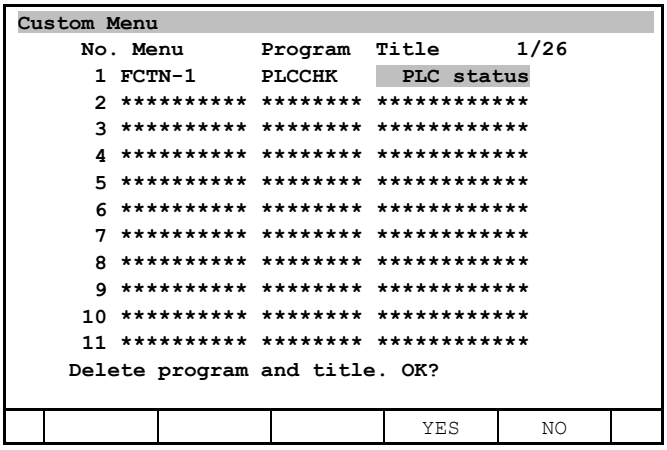


Select FCTN → 0 -- NEXT -- → 0 --NEXT --. Verify PLCCHK is added to FCTN menu.

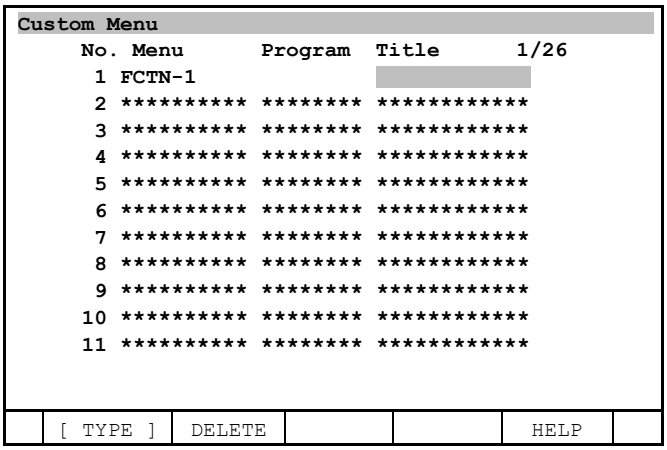


25.2.4 Delete Set

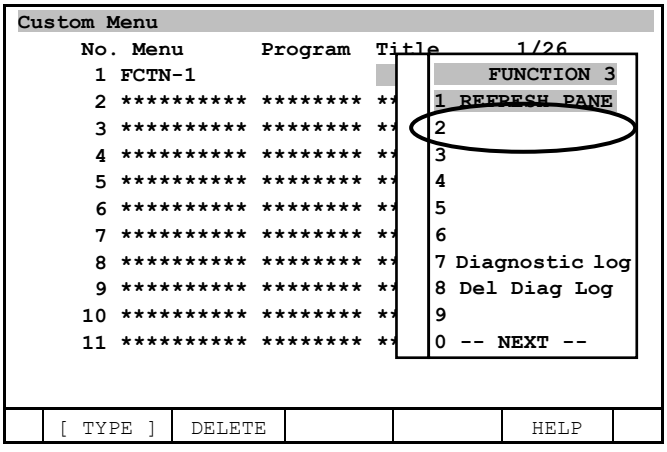
In the case of deleting set, press the F2, DELETE. Confirmation message will be displayed like following.



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.



26 KAREL PROGRAM EXECUTION HISTORY RECORD

KAREL program execution history record is a tool used to diagnose user application or controller system software problems. This option generates logs that show details about events, including the sequence in which they occur.

This option allows you to log certain events that occur during KAREL and TPP (TP program) execution. It also allows you to generate ASCII files that contain all or part of the logged data. The ASCII files contain detailed information about each event that is logged.

For example, you can log the following types of events.

- Start of execution of KAREL statements or TPP lines.
- Routine calls or returns.
- Internal system events for diagnosis of system software problems.

Refer to the following for additional information.

- For information about related hardware and software, refer to 26.1 HARDWARE AND SOFTWARE.
- For information about installing and setting up the debugging option, refer to 26.2 SETUP AND OPERATIONS.
- For information about events, refer to Section 26.3 LOGGING EVENTS.
- For examples of this option as related to KAREL programming, teach pendant programming, and the ASCII file format, refer to 26.4 EXAMPLES.

26.1 HARDWARE AND SOFTWARE

26.1.1 Hardware and Software Requirements

There are certain hardware and software requirements that must be met before you can run the KAREL Program Execution History Record.

NOTE
The hardware requirements in this section apply to all robot models.

26.1.2 Hardware

The KAREL Program Execution History Record requires the following hardware for all robot models:
- A FANUC Robot Controller
- A teach pendant
- 42000 bytes of Temporary memory
- 18000 bytes of CMOS
- 30000 bytes of FROM file system space

26.1.3 Software

The KAREL Program Execution History Recode function is included in Customize Support Function option (J742). However, in 7DC1 and 7DD0 series, this function is included in KAREL Use Support function option (J971).

A series of teach pendant is used to control the logging of diagnostic information. The teach pendant screens allow you to:

- Specify the kinds of events you want to log.
- Specify the tasks for which events are to be logged.
- Generate ASCII files containing the logged data.

26.1.4 Performance

The KAREL Program Execution History Record Option can affect the performance of your system.

- This option reduces execution speed of KAREL and teach pendant program logic by approximately 1 to 2 %.

NOTE
You can select “Disable all logging” on the Diagnostic Logging main menu to eliminate this reduction.

- Actual logging of events adds about .08 ms per event logged.
- Motion execution is not affected.

26.2 SETUP AND OPERATIONS

26.2.1 Setting Up the KAREL Program Execution History Record

Before you can log information and write data log files, you must set up the KAREL Program Execution History Record Option. This section contains information about how to set up the KAREL Program Execution History Record Option, including a detailed procedure for each setup screen.

The following screens are used to control diagnostic logging:

- Clear event log deletes all data from the log.
- Dump Selections screen allows you to write log data to an ASCII file, Subsection 26.2.2.
- Task Selection screen allows you to add tasks for which events are logged, Subsection 26.2.3.
- Stop Logging a task Selection screen allows you to select tasks to be removed from the list of currently logged tasks, Subsection 26.2.4.
- Currently Selected tasks screen lists tasks currently selected for logging, Subsection 26.2.5.
- Event Class Selections screen allows you to select or de-select classes or groups of related event types for logging, Subsection 26.2.6.
- Event Detail selections screen allows you to select or de-select individual event types for logging, Subsection 26.2.7.
- Enable or Disable all logging allows you to start or stop all events logging, Subsection 26.2.8.

For example, you can log the following types of events.
- Start of execution of KAREL statements or TPP lines.
- Routine calls or returns.
- Internal system events for diagnosis of system software problems.

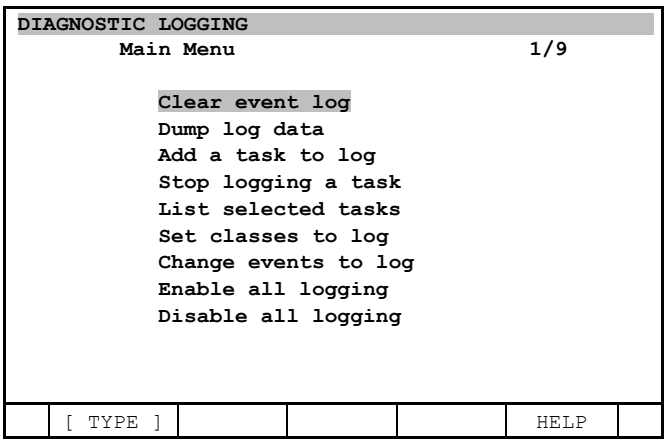
26.2.2 Dump Selections Screen

The Dump Selection Screen allows you to select a range of records to be dumped or logged. If the log is currently empty, an error message is displayed. Use Procedure 26-1 to dump records to the debug log.

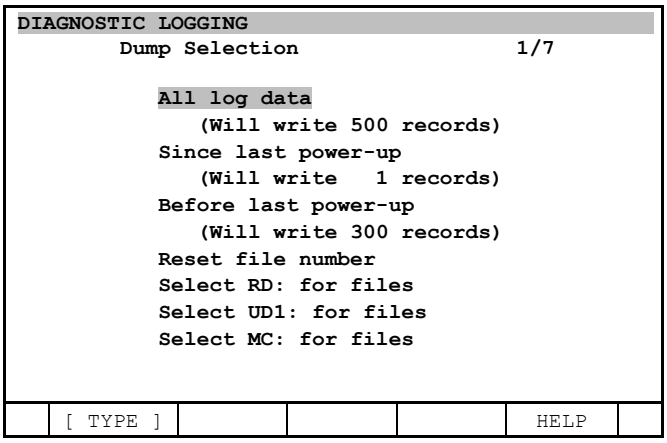
Procedure 26-1 Dumping Records to the Debug Log

Steps

1 Press the [MENU] key.
2 Select TEST CYCLE.
3 Press the F1, [TYPE].
4 Select Debug Ctl, the following screen will be displayed.



5 Select Dump log data. The number of records in each range is displayed on the screen. See the following screen for an example.



6 Select the range of data to be dumped or logged.
- To request a dump of all event information currently in the log, select All log data.
- To request a dump of all event information logged since the most recent power-up, select Since last power-up.
- To request a dump of all event information between the last two power-ups, select Before last power-up. This will log all events prior to the last power-down.

The information will be dumped to an ASCII file.

NOTE
By default, successive log files are written to MC:PGDBG201.DT, MC:PGDBG202.DT, and so forth.

7 To reset the file number in the file naming convention, select Reset file number. The starting point is MC:PGDBG201.DT.
8 To change the default saving device to RAM disk, select RD: for files. All files will be automatically saved to RAM disk.

9 To change the default saving device to USB memory, select UD1: for files. All files will automatically be saved to USB memory.
10 To change the default saving device to memory card, select MC: for files. All files will automatically be saved to a memory card.

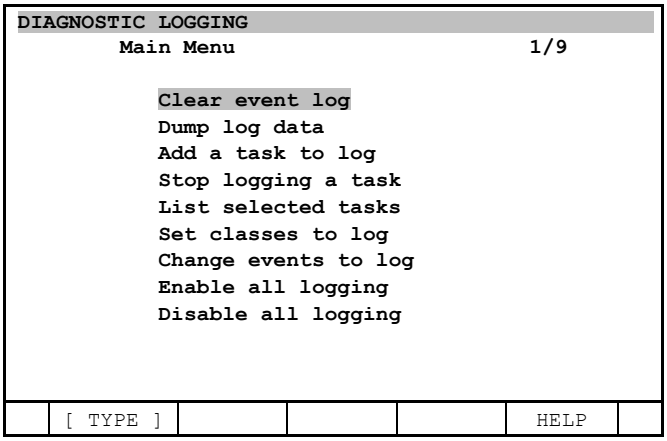
26.2.3 Task Selection Screen

The Task Selection screen allows you to select additional tasks for which you want to log information. Use Procedure 26-2 to select tasks to log.

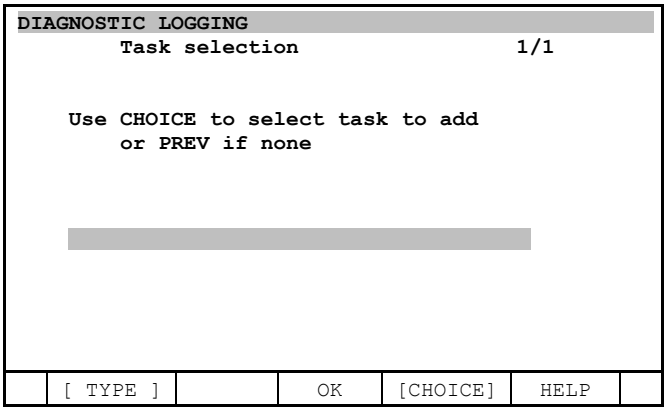
Procedure 26-2 Selecting Tasks to Log

Steps

1 Press the [MENU] key.
2 Select TEST CYCLE.
3 Press the F1, [TYPE].
4 Select Debug Ctl. The following screen will be displayed.

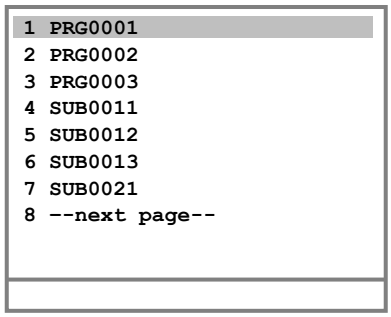


5 Select Add a task to log. The following screen will be displayed.

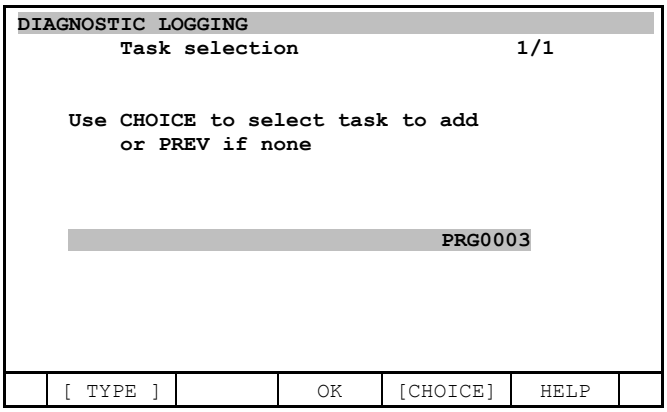


6 To display a list of tasks from which you can select, press the F4, [CHOICE]. The following screen will be displayed.

NOTE
Selecting a program means that events will be logged for a task that has the selected program as its MAIN program.



7 Use the arrow keys to select the task for which you want events logged.
8 Press the [ENTER] key when you have chosen a task. The following screen will be displayed.



9 To request logging for the displayed task, press the F3, OK.

⚠CAUTION
Do not exit this screen by pressing the [PREV], F1, [TYPE], or any other key before you press the F3, OK; otherwise, the selected task will not be logged.

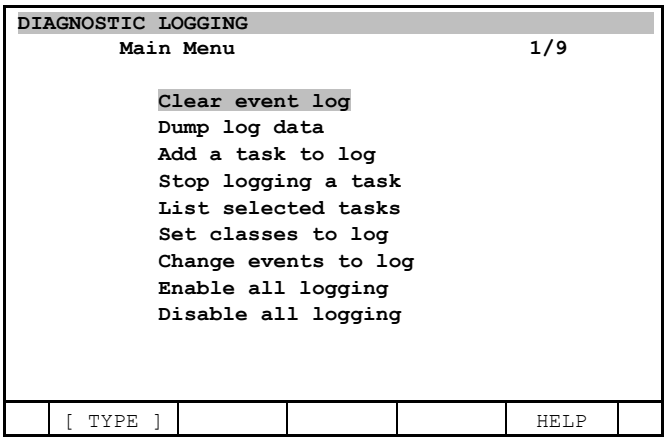
26.2.4 Stop Logging Tasks Screen

The Stop Logging Tasks screen allows you to select tasks to be removed from the list of currently logged tasks. Use Procedure 26-3 to stop logging tasks.

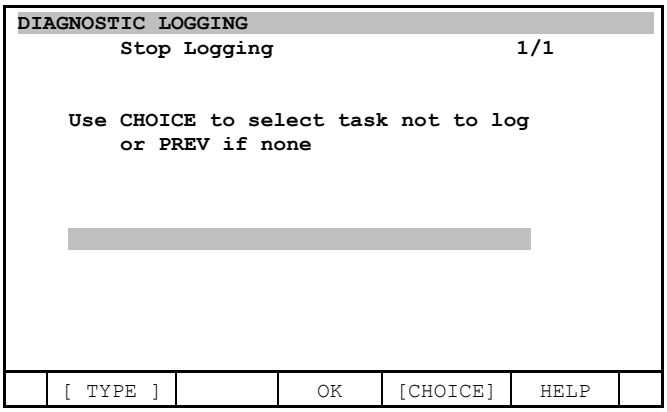
Procedure 26-3 Stop Logging Tasks

Steps

1 Press the [MENU] key.
2 Select TEST CYCLE.
3 Press the F1, [TYPE].
4 Select Debug Ctl. The following screen will be displayed.

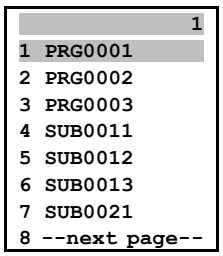


5 Select Stop logging a task. The following screen will be displayed.

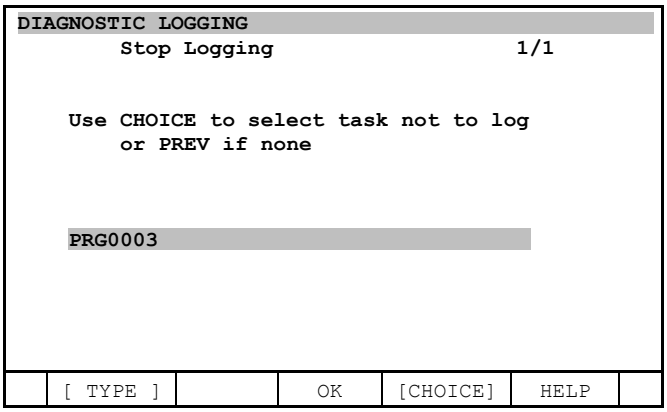


6 To display a list of tasks, from which you can select, press the F4, [CHOICE]. The following screen will be displayed.

NOTE
When you select a program, the events will no longer be logged for any tasks that have the selected program as their MAIN program.



7 Use the arrow keys to select the task for which you want to stop logging events.
8 Press the [ENTER] key when you have chosen a task. The following screen will be displayed.



9 To stop logging for the task that is current displayed, press the F3, OK.

⚠CAUTION
Do not exit this screen by pressing the [PREV], F1, [TYPE] , or any other key before you press the F3, OK; otherwise, logging for the selected task will not stop.

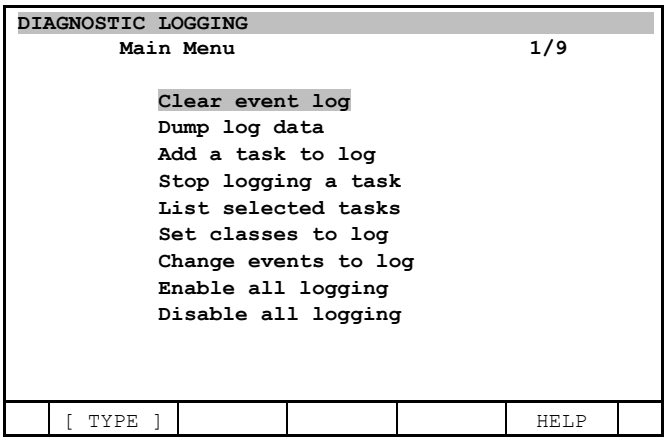
26.2.5 List Selected Tasks Screen

The List Selected Tasks Screen allows you to displays the tasks that are currently selected and being logged. Use Procedure 26-4 to list selected tasks.

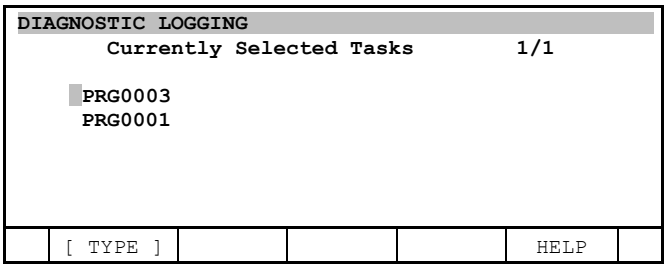
Procedure26-4 List Selected Tasks

Steps

1 Press the [MENU] key.
2 Select TEST CYCLE.
3 Press the F1, [TYPE].
4 Select Debug Ctl. The following screen will be displayed.



5 Select List selected tasks. The following screen will be displayed.



This screen displays the tasks that are currently selected and being logged.

6 Press the [PREV] key on the teach pendant to return to the main menu.

26.2.6 Event Class Selection Screen

The Event Class Selection Screen allows you to determine which classes of events should be logged (Procedure26-5). Each class of events enables or disables one or more detailed event types. The following is a list of event classes and the associated event types. Refer to Subsection Table 26.3.4 for more detailed information on the types of events that can be logged.

NOTE
An asterisk (*) indicates logging is for internal use only.

- Execute KAREL statement
  - Execute KAREL statement
- Execute TPP line
  - Execute TPP line
- Call/return
  - KAREL or TPP routine called
  - KAREL or TPP routine returned
- Motion *
  - Motion started *
  - Motion planned (teach pendant motion only) *
  - Motion cancel issued *
  - Motion stop issued *
  - Resume move *
  - Motion done received *
  - Motion completed normally *
  - MMR received *
- Condition Handler *
  - Condition handler triggered *
  - Condition handler enabled *
  - Condition handler disabled *
- Interrupt rtn
  - Before transferring to an ISR (Interrupt Sub-Routine)
  - After transferring to an ISR (Interrupt Sub-Routine)
  - Return from ISR (Interrupt Sub-Routine)
- Task start/end
  - KAREL or TPP task starts execution
  - KAREL or TPP task aborts
- Packet_rcd *
  - Interpreter receives packet *
- Pcode exec *
  - Starting execution of p-code *
- AMR activity *

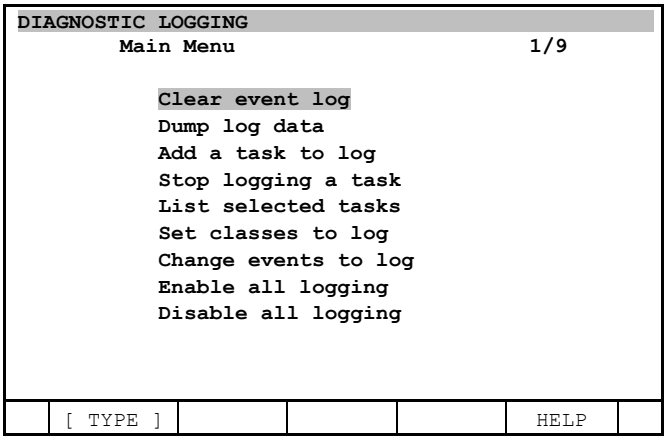
- Normal AMR recvd by AMGR *
- Start AMR recvd by AMGR *
- Stop AMR recvd by AMGR *
- AMR sent to AX *
- AMR rcvd from AX *

Use Procedure 26-5 to enable or disable event classes for logging.

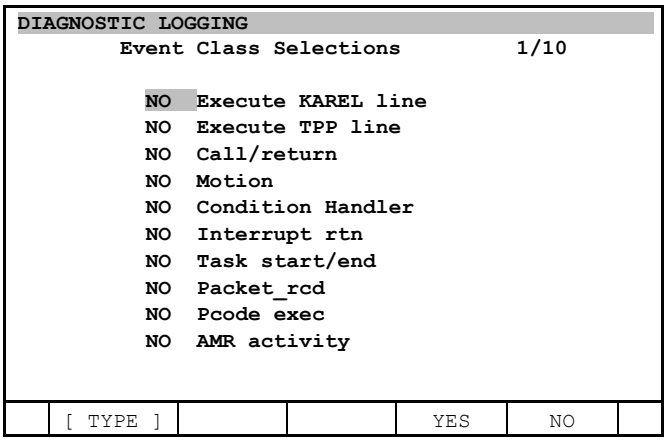
Procedure 26-5 Enable or Disable Event Classes for Logging

Steps

1 Press the [MENU] key.
2 Select TEST CYCLE.
3 Press the F1, [TYPE].
4 Select Debug Ctl. The following screen will be displayed.



5 Select Set classes to log. The following screen will be displayed.



6 Enable or disable classes of events to be logged.

- To turn on logging for a class of events, press the F4, YES.
- To turn off logging for a class of events, press the F5, NO.

The changes take effect immediately. The current status of whether an event is being logged is displayed to the left of the event.

NOTE
If you specify an entire class of events, you will enable or disable logging for all event types in that class.

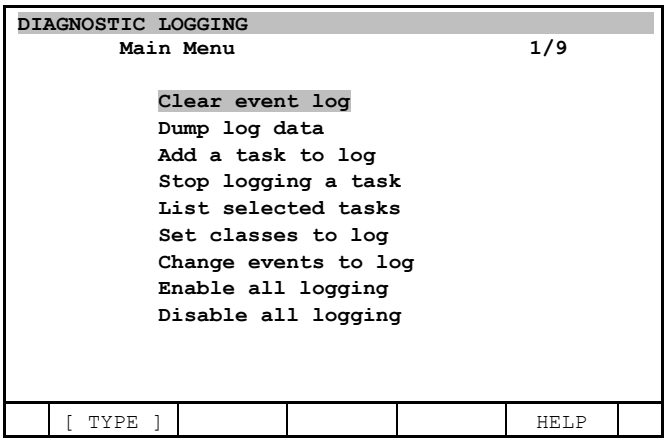
26.2.7 Event Detail Selection Screen

The Event Detail Selections Screen allows you to enable and disable logging of specific events. Use Procedure 26-6 to enable or disable a specific event type for logging.

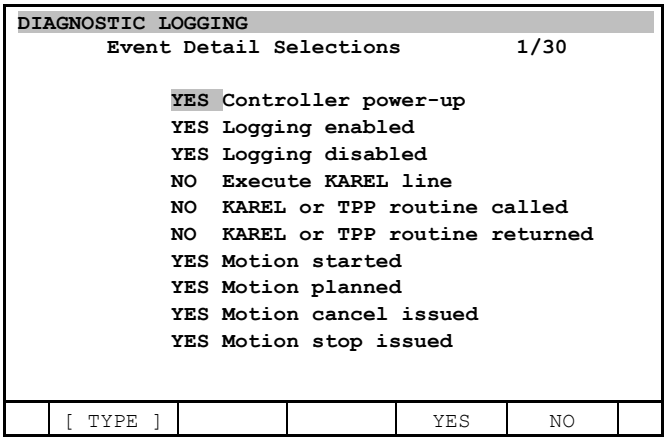
Procedure 26-6 Enable or Disable a Specific Event Type for Logging

Steps

1 Press the [MENU] key.
2 Select TEST CYCLE.
3 Press the F1, [TYPE].
4 Select Debug Ctl. The following screen will be displayed.



5 Select Change events to log. The following screen will be displayed.



6 Enable or disable a specific event type to be logged.
- To turn on logging for an event type, press F4, YES.
- To turn off logging for an event type, press F5, NO.

The changes take effect immediately.

26.2.8 Enable or Disable All Event Logging

The Enable/Disable All Logging Screen allows you to start or stop the automatic logging of all selected events. When the Program Diagnostic Option is installed, logging of selected events in selected tasks is enabled. The Disable all logging menu selection turns off all logging. It also eliminates the overhead added when logging is enabled. (Refer to Subsection 26.1.4 "Performance".) The Enable all logging menu selection can be used to start logging again.

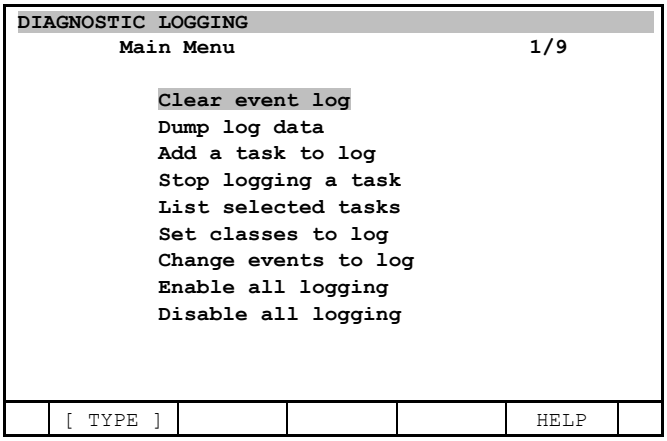
NOTE
Event and task selections are not changed by Disable all logging or Enable all logging.

Use Procedure 26-7 to enable and disable all event logging.

Procedure 26-7 Enabling and Disabling All Event Logging

Steps

1 Press the [MENU] key.
2 Select TEST CYCLE.
3 Press the F1, [TYPE].
4 Select Debug Ctl. The following screen will be displayed.



5 Select either Enable or Disable all logging.

⚠CAUTION
If you Enable or Disable all logging, you must perform a Cold Start for the change to take effect.

- If you select Enable all logging, selected events in selected tasks will automatically start being logged after you perform a Cold Start.
- If you select Disable all logging, selected events in selected tasks will automatically stop being logged after you perform a Cold Start.

6 Perform a Cold Start.

26.3 LOGGING EVENTS

Events are conditions or situations that occur in a KAREL or teach pendant program while the program is running. The KAREL Program Execution History Record Option allows you to record certain events, as they take place, to help you in debugging programs. The logged events can be dumped to an ASCII file.

26.3.1 Setting up Events

An event will be recorded in the log when all of the following are true:

- You have selected Enable all logging from the main menu.
- You have enabled event logging for the task.
- You have selected the event type to log either,
  - Specifically, using the Change Events To Log screen to select a specific event.

- Generally, using the Set Classes To Log screen to select a class of events.

Types of Events that can be Logged

With the Program Diagnostic Option, the following kinds of events can be recorded. Refer to Subsection 26.4.4 for additional information.

- Controller power-up
- Logging enabled
- Logging disabled
- Execute KAREL line
- KAREL or TPP routine called
- KAREL or TPP routine returned
- Motion started
- Motion planned
- Motion cancel issued
- Motion stop issued
- Resume move
- Motion done received
- Motion completed normally
- Condition handler triggered
- Condition handler enabled
- Condition handler disabled
- Before processing ISR
- After processing ISR
- Return from KAREL ISR
- KAREL or TPP task starts execution
- KAREL or TPP task aborts
- Interpreter receives packet
- Starting execution of p-code
- Start of execution of TPP line
- MMR received
- Normal AMR received by AMGR
- Start AMR received by AMGR
- Stop AMR received by AMGR
- AMR sent to AX
- AMR received from AX

26.3.2 Logging Events to an ASCII File

The last 500 events that are recorded can be written to an ASCII file. This can be done by requesting to write the logged data to a file on the Dump Selections screen (refer to Subsection 26.2.2).

26.3.3 ASCII File General Event Information

Specific information about each event is logged to the ASCII file. Table 26.3.4 describes the following information for each event:

- **Event name:** Event description displayed in the Event Detail Selections screen.
- **When recorded:** Conditions under which the event is recorded. For example, the STRT_K_LINE event is recorded when a KAREL statement is about to be executed.
- **Information recorded:** List of values logged with each event. The following are reported for all events (except as noted).
  - Event time (seconds) since power-up. This is always a multiple of 4 milliseconds (.004 second).
  - Number of the task associated with the event.
  - Name of the KAREL routine or TPP program, in which the statement was triggered.
  - Program line number at which statement event was triggered.

- **Enabled default:** YES if logging of the specified event type is Enabled by default.
- **Comments:** Additional information, if appropriate.

26.3.4 ASCII File Specific Event Information

Table 26.3.4 contains each event and the corresponding information that is logged.

Table 26.3.4 Event logging information

Event Name	When Recorded	Information Recorded	Enabled Default	Comments
Controller power-up	At every controller COLD start	Routine name, line number	YES	
		Clock time: at which power-up occurred, in "DD-MMM-YY HH:MM" format.		
		Task number will always be 1.		
Logging enabled	When logging is started. Typically, this is when the task starts.	Standard information only	YES	
Logging disabled	When logging is stopped. Typically, this is when the task ends.	Standard information only	YES	
Execute KAREL line	At beginning of execution of KAREL statement.	Internal information	NO	
		Internal information		
		Internal information		
KAREL or TPP routine called	When a KAREL routine or KAREL or TPP program is called. This is either directly or as a result of a condition handler action.	Standard information only	NO	Routine name is name of called routine; line number is line number from which call was made or interrupt occurred.
KAREL or TPP routine returned	When a KAREL PROGRAM or ROUTINE or TP program returns to the calling or interrupted program.	Standard information only	NO	The ROUTINE name is the name of the returning program or routine; the line number is the line in the calling or interrupted program to which the task is returning, generally the same as the line number shown for in the CALL event.
Motion started	When a request to initiate a motion is issued.	Group mask	NO	Intended for system level analysis.
		MMR address, in hexadecimal		
		MMR status, in hexadecimal; normally FFFFFFFF		
Motion planned	For TPP motion commands, this occurs typically before the motion is actually started.	Group mask	NO	Intended for system level analysis.
		MMR address, in hexadecimal		
		MMR status, in hexadecimal; normally FFFFFFFF		

Event Name	When Recorded	Information Recorded	Enabled Default	Comments
Motion cancel issued	Recorded in case planned motion isn't executed by stop request such as HOLD. When a CANCEL statement or condition handler action is executed, a MOVE... UNTIL condition is satisfied, a CANCEL severity error is posted, or a program is aborted with a motion planned or in progress.	Group mask MMR address, in hexadecimal MMR status, in hexadecimal; normally FFFFFFFF	NO	Intended for system level analysis.
Motion stop issued	Recorded in case motion is stopped by HOLD or error. When a STOP statement or condition handler action is executed or a STOP severity error is posted.	Group mask MMR address, in hexadecimal MMR status, in hexadecimal; normally FFFFFFFF	NO	Intended for system level analysis.
Resume move	When a stopped motion is resumed; typically when a RESUME statement or condition handler action is executed; also when a program is CONTINUED following a STOP error condition.	Group mask MMR address, in hexadecimal MMR status, in hexadecimal; normally FFFFFFFF	NO	Intended for system level analysis.
Motion done received	When the termination type is satisfied for a KAREL or TPP motion statement executed. For KAREL MOVE ... NOWAIT statements, this event if recorded when the motion starts.	Group mask MMR address, in hexadecimal MMR status, in hexadecimal; normally FFFFFFFF	NO	Intended for system level analysis.
Motion completed normally	When a motion is completed or cancelled. This is generally after MTN_DONE is recorded.	Group mask MMR address, in hexadecimal MMR status, in hexadecimal; normally FFFFFFFF	NO	Intended for system level analysis.
Condition handler triggered	When a global condition handler triggers.	Condition handler number	NO	Intended for system level analysis.
Condition handler enabled	When a global condition handler is enabled by an ENABLE statement or condition handler action.	Condition handler number	NO	Intended for system level analysis.

Event Name	When Recorded	Information Recorded	Enabled Default	Comments
Condition handler disabled	When a global condition handler is disabled by a DISABLE statement or condition handler action. It is not recorded when a condition handler triggers.	Condition handler number	NO	Intended for system level analysis.
Before processing ISR	Before calling a KAREL PROGRAM from a condition handler.	Internal information	NO	The routine name and line number indicate the code that was executing when the interrupt routine request was received.
		Internal information		
		Internal information		
After processing ISR	When the interrupt routine is ready to run.	Standard information only	NO	Routine name is the name of the interrupt routine. Line number will always be 1.
Return from KAREL ISR	When exiting from an interrupt routine.	Standard information only	NO	Routine name and line number are the interrupt routine name and the line number from which it returned.
KAREL or TPP task starts execution	When a task selected for logging starts executing.	Standard information only	YES	Routine name is the main program name; line number is always 1.
KAREL or TPP task aborts	When a task ends.	Standard information only	YES	Routine and line number indicate the last statement executed by the task. This might be an END statement of a KAREL program or the last line of a TPP program.
Interpreter receives packet	When an interpreter task receives a packet.	Packet address	NO	Intended for system level analysis.
		Packet status		
		Request code (including sub-system code)		
		Requestor id		
		ITR-level		
Starting execution of p-code	At start of execution of KAREL p-code instruction.	Number of words on the routine stack	NO	Intended for system level analysis.
		Number of words on the data stack		
		P-code mnemonic		

Event Name	When Recorded	Information Recorded	Enabled Default	Comments
Start of execution of TPP line	At start of execution of a line of a TPP program.	Number of words on the data stack	NO	
		Number of words on the routine stack		
		Number of words available for combined ROUTINE and DATA stacks		
MMR received	When an MMR is received back from the motion sub-system. This occurs when the motion completes, is cancelled, or is stopped.	Group mask	NO	Intended for system level analysis.
		MMR address, in hexadecimal		
		MMR status, in hexadecimal; normally FFFFFFFF		
Normal AMR recvd by AMGR	An AMR is received by AMGR.	Address of AMR	NO	Intended for system level analysis.
		AMR number		
		AMR AMGR_wk		
		AMR ax_phase		
Start AMR recvd by AMGR	A start AMR request is processed by AMGR.	Address of AMR	NO	Intended for system level analysis.
		AMR number		
		AMR AMGR_wk		
		AMR ax_phase		
Stop AMR recvd by AMGR	A stop AMR request is processed by AMGR.	Address of AMR	NO	Intended for system level analysis.
		AMR number		
		AMR AMGR_wk		
		AMR ax_phase		
AMR sent to AX	When an AMR is sent from AMGR to an AX task.	Address of AMR	NO	Intended for system level analysis.
		AMR number		
		AMR AMGR_wk		
		AMR ax_phase		
AMR rcvd from AX	When an AMR is receivedack from an AX task.	Address of AMR	NO	Intended for system level analysis.
		AMR number		
		AMR AMGR_wk		
		AMR ax_phase		

26.4 APPENDIX

26.4.1 Overview

This section contains example programs and the resulting log data that show how the KAREL Program Execution History Record Option operates.

- Subsection 26.4.2 contains a KAREL program (T) which calls a teach pendant program (TPP).
- Subsection 26.4.3 contains the teach pendant program that is called (TTT).
- Subsection 26.4.4 contains the log file that is generated after running the KAREL program (T) with the following Event class selections:
  - YES Execute KAREL line
  - YES Execute TPP line
  - YES Call/return
  - YES Motion
  - YES Condition Handler
  - YES Interrupt rtn
  - YES Task start/end
  - NO Packet_rcd
  - NO Pcode exec

26.4.2 KAREL Program Example

The KAREL program in Example 26.4.2 is the MAIN program from which the TPP program is called and executed.

Example 26.4.2 KAREL program example (T.KL)

```
 1 program t
 2 var
 3   i,j,k: INTEGER
 4
 5 routine ttt from ttt
 6 routine tt
 7 begin
 8   i = i + 1
 9 end tt
10
11 begin
12
13 condition[1]:
14   when k >= 500 DO
15     k = 0
16     tt
17     enable condition[1]
18   endcondition
19 k = 0
20 i = 0
21 connect timer to k
22 enable condition[1]
23 wait for i=2
24 disconnect timer k
25 disable condition[1]
26 ttt
```

```
27 DELAY 1000
28 end t
```

26.4.3 Teach Pendant Program Example

The TPP program in Example 26.4.3 is called by the KAREL program (T) in Example 26.4.2.

Example 26.4.3 Teach pendant program example (TTT.TP)

```
1:J P[1] 100% FINE
2:  WAIT   0.00(sec)
3:  R[1]=0
4:  LBL[1]
5:  R[1]=R[1]+1
6:  IF R[1] <= 3,JMP LBL[1]
7:  WAIT    .50(sec)
```

26.4.4 ASCII File Example

With logging task T enable and event classes selected as indicated in Example 26.4.2, if T is executed and the log dumped, the file show in Example 26.4.4 is generated.

NOTE
For detailed information about the fields on the ASCII report shown in Example 26.4.4, refer to Subsection 26.3.4.

Example 26.4.4 ASCII File Example

Event	Time	TID	Routine	Line				
POWER-UP	.104	1	***	***	09- 9-28 11:53			
TASK-START	84.888	8	T	1				
LOG-ENABLE	84.888	8	T	1				
STRT-K-LINE	84.888	8	T	13	0	0	19200	
STRT-K-LINE	84.904	8	T	19	0	0	19200	
STRT-K-LINE	84.904	8	T	20	0	0	19200	
STRT-K-LINE	84.904	8	T	21	0	0	19200	
STRT-K-LINE	84.904	8	T	22	0	0	19200	
STRT-K-LINE	84.904	8	T	23	0	0	19200	
CH-ENABLE	84.904	8	T	23	1			
CH-TRIGGER	85.912	8	T	23	1			
PRE-ISR	85.912	8	T	23	0	128	35	
CALL	85.912	8	TT	23				
IN-ISR	85.912	8	TT	1				
STRT-K-LINE	85.912	8	TT	8	0	68	1792	
STRT-K-LINE	85.928	8	TT	9	0	68	1792	
RETURN	85.928	8	TT	23				
RTN-INT-RTN	85.928	8	TT	23				
CH-TRIGGER	86.920	8	T	23	1			
PRE-ISR	86.920	8	T	23	0	128	35	
CALL	86.920	8	TT	23				
IN-ISR	86.920	8	TT	1				
STRT-K-LINE	86.920	8	TT	8	0	68	1792	
STRT-K-LINE	86.936	8	TT	9	0	8	1792	
RETURN	86.952	8	TT	23				
RTN-INT-RTN	86.952	8	TT	23				
STRT-K-LINE	86.952	8	T	24	0	0	19200	
STRT-K-LINE	86.952	8	T	25	0	0	19200	
STRT-K-LINE	86.952	8	T	26	0	0	19200	
CH-DISABLE	86.952	8	T	26	1			
CALL	86.952	8	TTT	26				
STRT-T-LINE	86.952	8	TTT	1	0	16	15104	
PLAN-MOVE	86.952	8	TTT	1	405E61A8		FFFFFFFF	6
START-MOVE	86.952	8	TTT	1	405E61A8		FFFFFFFF	6
MTN-DONE	88.088	8	TTT	1	405E61A8		FFFFFFFF	3
STRT-T-LINE	88.088	8	TTT	2	0	16	15104	
MMR_RCVD	88.104	8	TTT	1	405E61A8		00000000	3
MTN_ENDED	88.104	8	TTT	1	405E61A8		00000000	3
STRT-T-LINE	88.104	8	TTT	3	0	16	15104	
STRT-T-LINE	88.104	8	TTT	4	0	16	15104	
STRT-T-LINE	88.104	8	TTT	5	0	16	15104	
STRT-T-LINE	88.104	8	TTT	6	0	16	15104	
STRT-T-LINE	88.136	8	TTT	4	0	16	15104	
STRT-T-LINE	88.136	8	TTT	5	0	16	15104	
STRT-T-LINE	88.136	8	TTT	6	0	16	15104	
STRT-T-LINE	88.168	8	TTT	4	0	16	15104	
STRT-T-LINE	88.168	8	TTT	5	0	16	15104	
STRT-T-LINE	88.168	8	TTT	6	0	16	15104	
STRT-T-LINE	88.200	8	TTT	4	0	16	15104	
STRT-T-LINE	88.200	8	TTT	5	0	16	15104	
STRT-T-LINE	88.200	8	TTT	6	0	16	15104	
STRT-T-LINE	88.200	8	TTT	7	0	16	15104	
RETURN	89.208	8	TTT	26				
STRT-K-LINE	89.208	8	T	27	0	0	19200	
STRT-K-LINE	91.224	8	T	28	0	0	19200	
TASK-END	91.240	8	T	28				
LOG-DISABLE	91.240	8	T	28				

27 TORQUE LIMIT FUNCTION

Using a TP instruction, Torque Limit function can limit the maximum motor torque as desired. And Torque limit multi-axis setup function can set up torque limit for multi axes respectively. To use these functions, Torque Limit function option (A05B-2600-J611) is required.

You can realize followings by Torque Limit function.
This manual describes Torque Limit function using a Servo Hand which has a servo driven gripper.

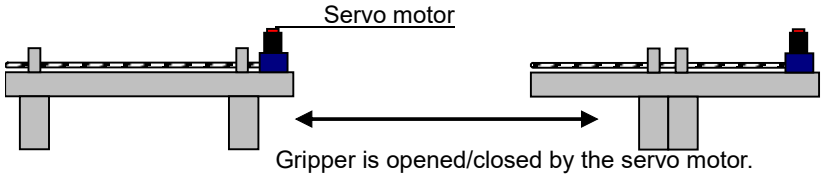


Fig. 27 (a) Example of servo hand

(1) Easy to teach the gripper position

Servo Hand can control the Open/Close position of gripper, but it is necessary to teach gripper position according to the work piece length (size). Then it is necessary to teach position for each work piece type, and it is not easy to operate and maintain the Servo Hand.

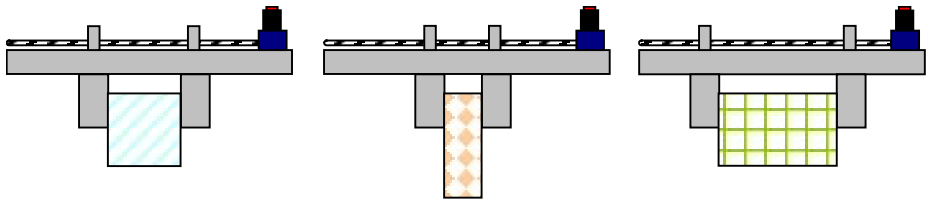


Fig. 27 (b) Teaching by legacy style

This function can stop the gripper movement when the external force (load) of the servo motor exceed the specified motor torque. Therefore, if you teach a closed gripper position and use Torque Limit for this motion instruction, the gripper will stop automatically at the position fit to the work length.

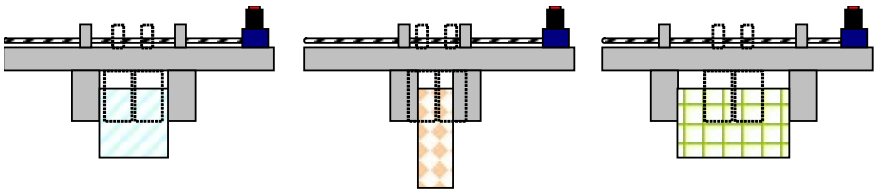


Fig. 27 (c) Teaching by torque limit function

Merit1: Easy to teach
One teaching point (where gripper is closed) can be used for all work pieces.

(2) Gripping force is adjustable.

By TORQ_LIMIT instruction, it is possible to change the maximum motor torque to arbitrary value. Therefore, it is possible to adjust the gripping force to the arbitrary value.

Merit2: Gripping force is adjustable.
It is possible to adjust the gripping force to the arbitrary value because the maximum motor torque can be limited as desired.

27.1 TORQUE LIMIT FUNCTION FEATURE

Torque Limit function can limit the maximum torque of servo motor which is controlled by a robot controller. You can set Torque Limit value in % of the maximum motor torque.

Syntax ... TORQ_LIMIT t% (t: 0.0 to 100.0)

NOTE

Depending on the combination of a motor and an amplifier, the maximum current of the amplifier may be limited to guard the motor. But, even in that case, the reference value (= 100%) of the Torque Limit function is the torque corresponding to the maximum current in the specification of the amplifier. For example, in case that the maximum current of the amplifier is limited to 20Ap though the maximum current in the specification is 40Ap, if you want to limit the torque to a value corresponding to 10Ap, specify TORQ_LIMIT 25% as the ratio of 10Ap to 40Ap is 25%.

If you specify the torque limit value which exceeds the limit for motor guard, the maximum current is limited automatically so that actually the current does not exceed the limit for motor guard. When executing the torque limit instruction with such a value, "INTP-271 Excessive torque limit value" warning is also shown. In the example above, even if you specify the value greater than 50%, the actual maximum current is clamped to 50% which corresponds to 20Ap.

Using following system variables, you can select the target axes of torque limit. In R-30*i*B/R-30*i*B Mate, J7 axis in Group 1 is enabled as default. In R-30*i*B Plus/R-30*i*B Mate Plus, any axis s isn't enabled as default.)

-Group number

$TORQUE_LMT.$GROUP[i]	Select the group number including the servo motor using Torque Limit. i : Group number

-Axis number

$TORQUE_LMT.$GAi[j]	Select the group number and axis number of the servo motor using Torque Limit. i : Group number, j : Axis number

For example, if J1 axis in Group 2 uses Torque Limit, set TRUE to $G2[1], and set others FALSE.

```
$TORQUE_LMT.$GROUP[2] = TRUE
$TORQUE_LMT.$GA2[1]   = TRUE
$TORQUE_LMT.$GA2[2]   = FALSE
:
$TORQUE_LMT.$GA2[9]   = FALSE
```

You can set multiple axes to TRUE.

In the case of the dual drive, set TRUE to both the master axis and slave axis.

⚠ CAUTION

Usually GA1[1] - [6] are robot axes. So, please do NOT set these variables to TRUE.

⚠ CAUTION
Check the intended axes of Torque Limit function in all axes in all groups that exist in your robot controller.

NOTE
The following software versions restrict usage of Torque Limit on robot axes. When TORQ_LIMIT instruction is executed to activate torque limiting on a robot axis, the alarm "INTP-663 **(%s^4, %d^5)** Torque Limit cannot be used" will be posted.
- 7DC1/33 (V8.10P/33) or later
- 7DC2/36 (V8.20P/36) or later
- 7DC3/63 (V8.30P/63) or later
- 7DF1/37 (V9.10P/37) or later
- 7DF3/21 (V9.30P/21) or later
- 7DF5/21 (V9.40P/21) or later

It is necessary for you to decide Torque Limit value.
It is necessary to decide Torque Limit value by gripping actual workpiece.
After a TORQ_LIMIT instruction is executed, the maximum motor torque will not be changed until another TORQ_LIMIT instruction will be executed.

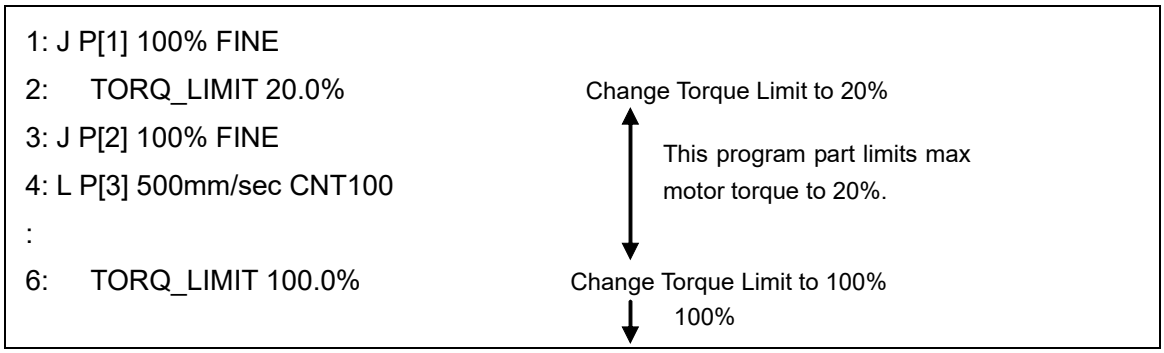


Fig.27.1 Example program of torque limit function

The upper limit of Torque Limit can be specified by the following system variables.
$TORQUE_LMT.$MAX_TRQ_LMT = Upper limit (0.0 - 100.0)
The lower limit is fixed to 0.0.

Depending on the workpiece or taught position of servo hand, the following alarms are more likely to be posted when torque limit function is used.
- SRVO-023 Stop error excess(G:%d A:%d)
- SRVO-024 Move error excess(G:%d A:%d)
- SRVO-036 Inpos time over (G:%d A:%d)

In order to prevent these alarms, it is necessary to change the following system variables to bigger values in advance.
- $PARAM_GROUP[group].$STOPTOL[axis]
- $PARAM_GROUP[group].$STOPERLIM[axis]
- $PARAM_GROUP[group].$MOVER_OFFST[axis]

However, if you change to too big values, these alarms may not be posted when they should. The following is an example of changing these system variables.

Procedure 27-1 Example of changing the system variables for a torque limit axis (in the case of a servo hand)

Step

1 Set the torque limit to 100%
2 Move the servo hand to the gripping position without a workpiece.
3 Select [MENU] → 0.--Next-- → 4 Status→ F4[PULSE].
4 Select F5[UTIL] → GROUP. Enter the group number of the servo hand.
5 Write down the "machine pulse" of the servo hand axis displayed on the TP screen.
6 Keep the torque limit at 100% and move the servo hand to the destination position without gripping the workpiece.
7 Write down the "machine pulse" at the destination position after moving.
8 Subtract the "machine pulse" of the gripping position from the destination one. Calculate the absolute value, and multiply it by 1.3 as a margin.
9 Set each system variable to the value calculated in step 8. Record the original value before the change so that you can undo it just in case.
10 Cycle the robot controller power.
11 After cycling power, perform the motion to actually grip the workpiece using the torque limit. At this time, monitor the "Position Error" of the servo hand axis on the PULSE screen.
12 Make sure that the "Position Error" in Step 11 is sufficiently smaller than the value calculated in Step 8. If it is not sufficiently small, increase the margin in Step 8 and perform steps 8 through 11 again.

By using the System Design Tool (J738), you can check the status (position, current, etc.) of the torque limit axis. For details, refer to Chapter 10 "CHECK SERVO HAND GRIP STATUS" in the "System Design Tool OPERETOR'S MANUAL (B-83274EN)".

27.2 TORQUE LIMIT MULTI-AXIS SETUP FUNCTION

You can set torque limit respectively to multiple axes using "CALL" from TP program.
Please set system variable $karel_enb to TRUE in order to select karel at "CALL" instruction.
Following explains how to use from TP program.

```
CALL TPTRQLIM(group number, axis number, torque limit value(%))
```

Fig. 27.2 (a) How to call "Torque limit multi-axis setup function"

- group number : Set the group number including the servo motor using Torque Limit.
- axis number : Set the axis number of the servo motor using Torque Limit.
  It cannot be used on robot axes. If the robot axis is specified and you execute this KAREL, the alarm "INTP-659 Axis not exist or is robot axis." will be posted.
- torque limit value : Set Torque Limit value in % of the maximum motor torque.
  (You can set 0.1% to 100.0%. You can also use integer.)

Following is an example TP problem.(This example TP program set 70% torque limit to axis 7 of group 1.)

```
CALL TPTRQLIM(1, 7, 70)
```

Fig. 27.2 (b) Example program of Torque limit multi-axis setup function

In the case of the dual drive, set TRUE to both the master axis and slave axis.

27.3 LIMITATIONS

- This function limits the maximum motor torque of the specified servo motor, but this function does NOT guarantee the motor torque while rotating the servo motor.
- Because this function is designed for FANUC servo motor, if you want to use this function to non FANUC servo motor, contact your local FANUC representative.

27.4 CAUTION

- Do NOT enable this function on robot axes.
- When an axis is stopped by Torque Limit, the external force is equal to the limited motor torque, and the distance between current position and destination position is considered as position error. (Please see the highlighted part in the figure below.) If you change Torque Limit to 100%, the axis moves to the destination with 100% of motor torque, therefore you may break Servo Hand or a work piece. If you want to change Torque Limit to 100% (or increase motor torque), you have to move the axis where above problem will not occur before changing Torque Limit.

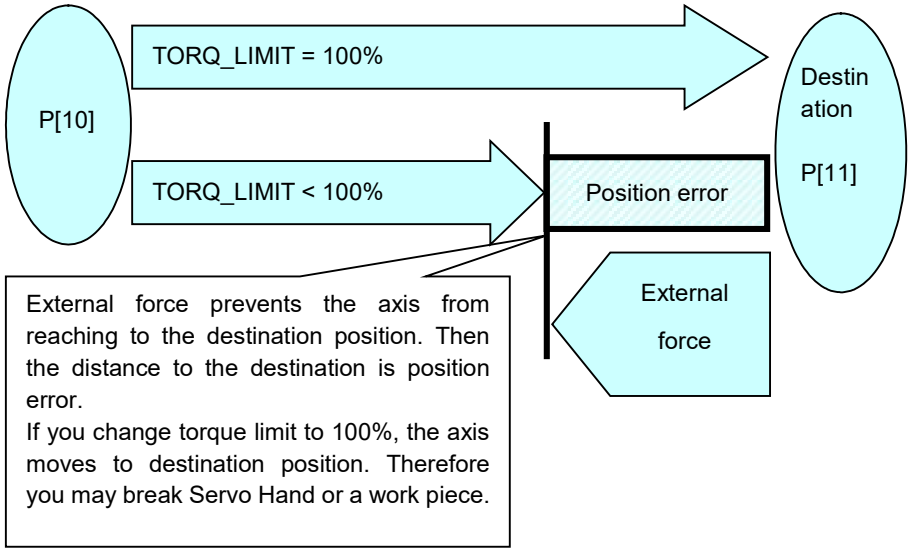


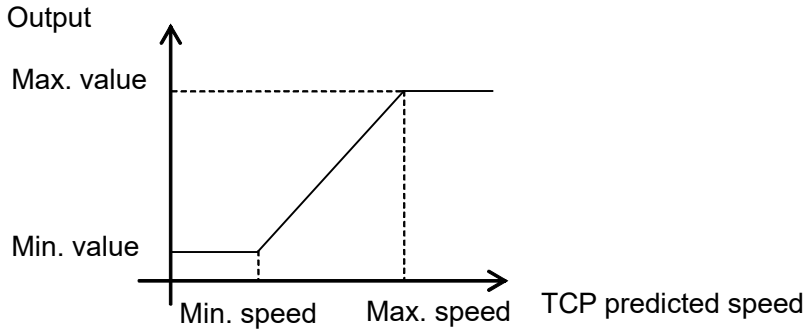
Fig. 27.4 Caution for torque limit function

28 TCP SPEED OUTPUT

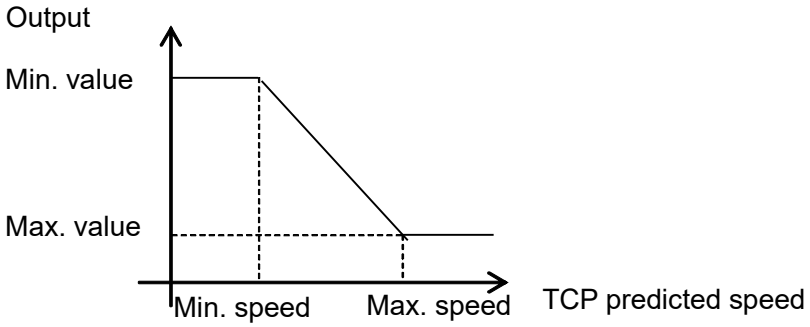
The TCP speed output function outputs the predicted speed of tool center point (TCP) via analog output signals, group output signals, or registers. A software option, TCP speed output function, is required to use this function. The TCP predicted speed is converted on the output condition that is defined by the following setup items:

- Min. value
- Max. value
- Min. speed (mm/sec)
- Max. speed (mm/sec)

The following figure shows the relationship between TCP predicted speed and output.



The Min value is output when TCP predicted speed is the specified Min speed or below. The Max value is output when TCP predicted speed is the specified Max speed or above. The following figure shows the relationship between TCP predicted speed and output when you set the Max value to a value of less than Min value.



28.1 LIMITATIONS

There are limitations on use of the TCP speed output function.

- This function is not available on DispenseTool or Dispense-enabled SpotTool+.
- This function is not available with tracking functions e.g. Line Tracking.
- TCP predicted speed is output only while the robot is moving by program execution.
- TCP predicted speed reflects only group 1 motion.
- TCP predicted speed does not reflect the extended axis motion except the integrated axis one.
- Accuracy of TCP predicted speed may lower when the motor speed limit alarms are posted.
- Accuracy of TCP predicted speed may lower while the frame is being switched.
- Accuracy of TCP predicted speed may lower while the motion is being switched between the normal and the remote TCP motion.
- Accuracy of the TCP predicted speed may lower while one of the following instructions is being executed:

WAIT instruction
Joint motion instruction
Motion instruction with Approach/Retract (Linear Distance) instruction
Motion instruction with Process Speed instruction
Motion instruction with Max Speed instruction
Motion control statement in KAREL program

- Up to 10 output conditions can be defined.
- Only one prediction time can be specified. Different prediction times cannot be specified among output conditions.

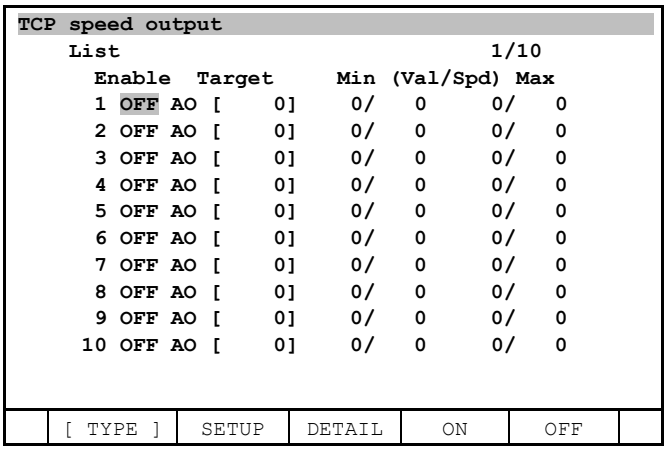
28.2 SETTING UP TCP SPEED OUTPUT

To use the TCP speed output function, you must set up the output condition using the procedure 28-1.

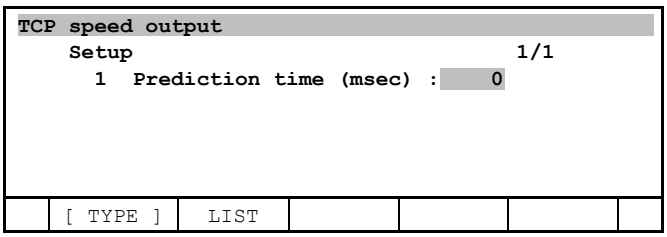
Procedure 28-1 Setting up TCP speed output conditions

Step

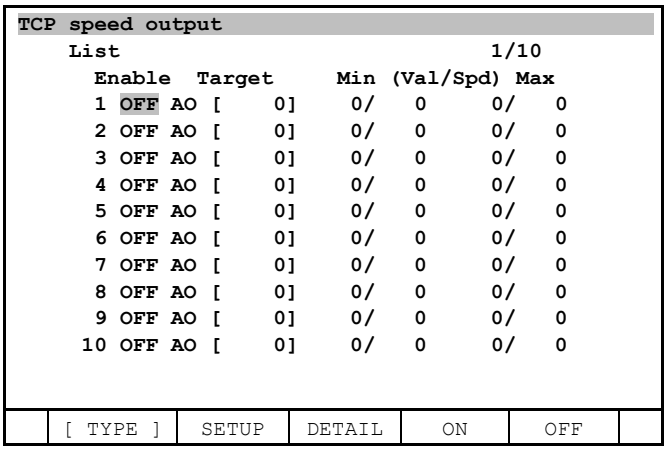
1 Press the [MENU] key.
2 Select I/O.
3 Press the F1, [TYPE].
4 Select TCP Speed. The following screen will be displayed.



5 Press the F2, SETUP. The following screen will be displayed.

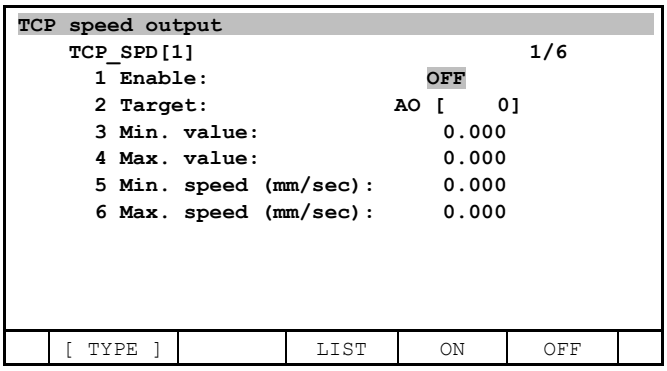


6 Set the value for Prediction time.
7 Press the F2, LIST. The following screen will be displayed.



NOTE
Each line in this screen indicates the status of one output condition.

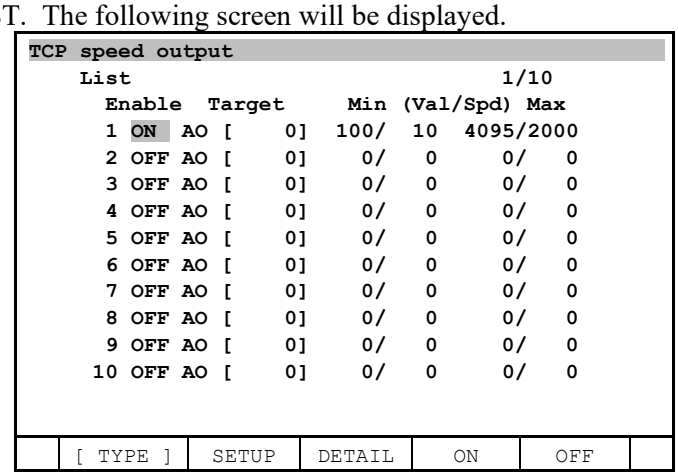
8 Move the cursor to an output condition that you want to set up, and press the F3, DETAIL. The following screen will be displayed.



9 Move the cursor to Target and select the output target.
10 Move the cursor to each item and set the value.
11 To start the output, set Enable to ON.

NOTE
You can also set Enable and Target by using an instruction. Refer to Procedure 28-2 for more information.

12 Press the F3, LIST. The following screen will be displayed.



NOTE
You can set Enable for each output condition on this screen.

28.3 TCP SPEED OUTPUT INSTRUCTION

You can set the Enable and Target of TCP speed output function by using an instruction.
The following instruction is used to start the output.

Target type[number]=(TCP_SPD[output condition number])

The following instruction is used to terminate the output.

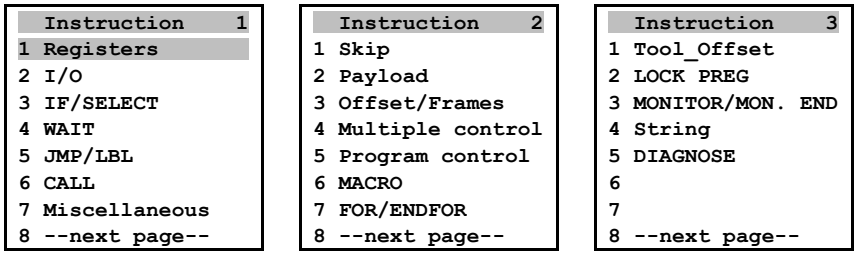
Target type[number]=constant value

Procedure 28-2 directs how to insert the TCP speed output instruction for a register type target.

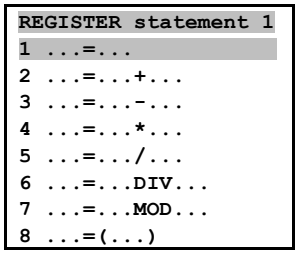
Procedure 28-2 Inserting TCP speed output instruction

Step

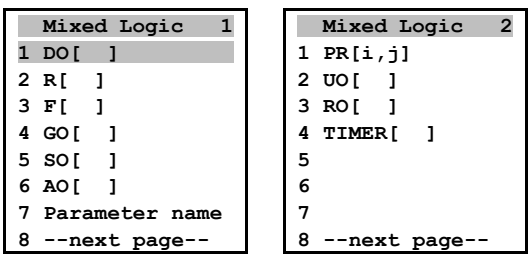
1 Open the program edit screen.
2 Press the [NEXT] key.
3 Press the F1, [INST]. The following menus will be displayed.



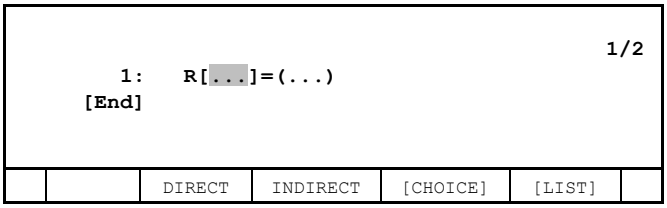
4 Select Registers. The following menu will be displayed.



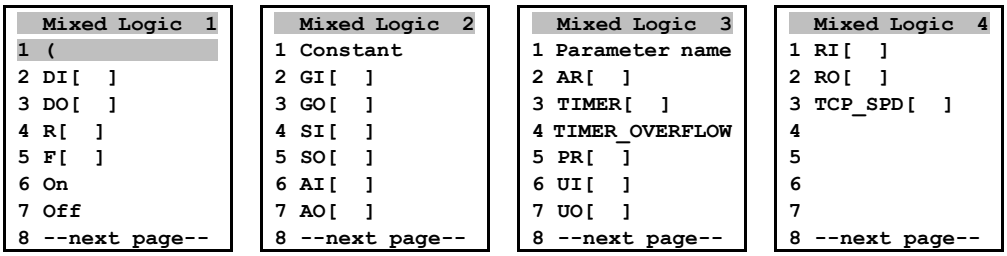
5 Select …=(...). The following menus will be displayed.



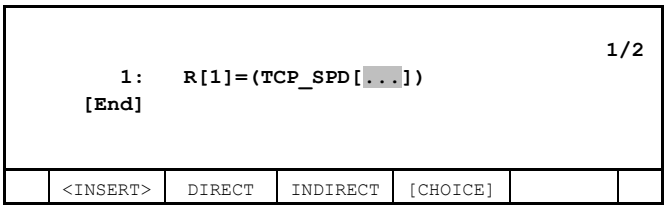
6 Select R[ ].



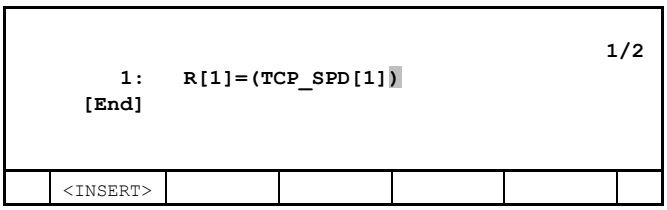
7 Set the register number for the output. The following menus are displayed.



8 Select TCP_SPD[…].



9 Set the output condition number.



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).

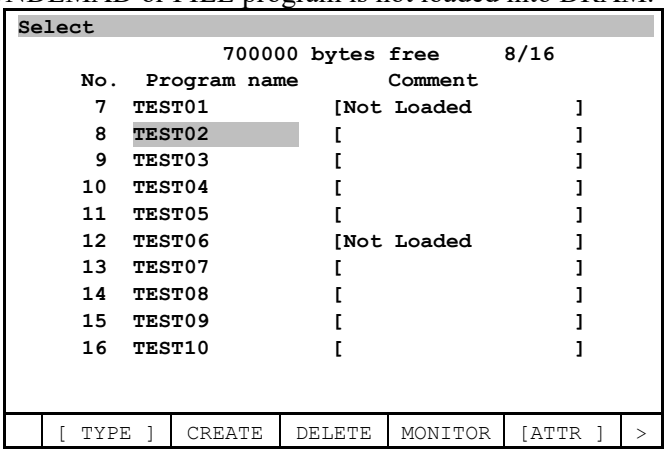
NOTE
Use a dedicated directory in external device for FILE storage path. Do not use shared directly to save TP program files in other storage and backup files.

29.2 STORAGE CONFIGURATION

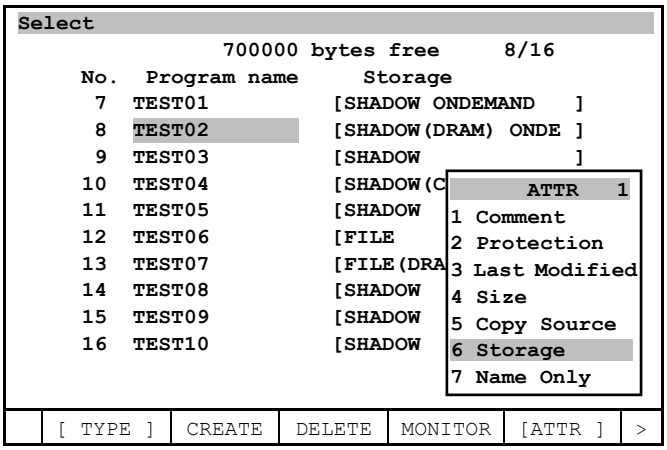
Procedure 29-1 Confirm storage device

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.



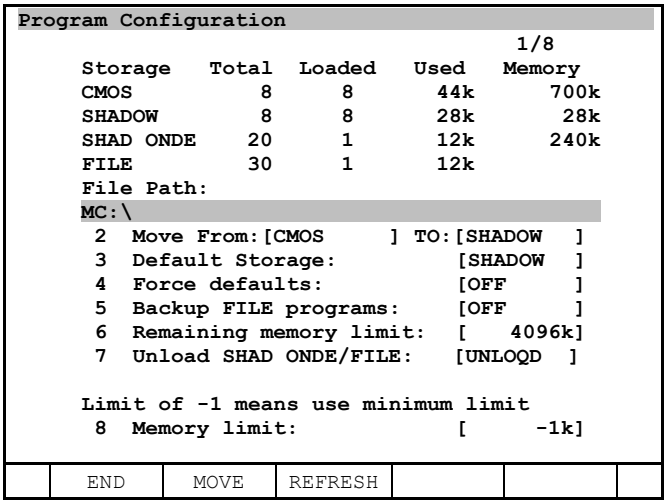
2 Press the F5, [ATTR], and select “Storage”. The following screen will be displayed.



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.

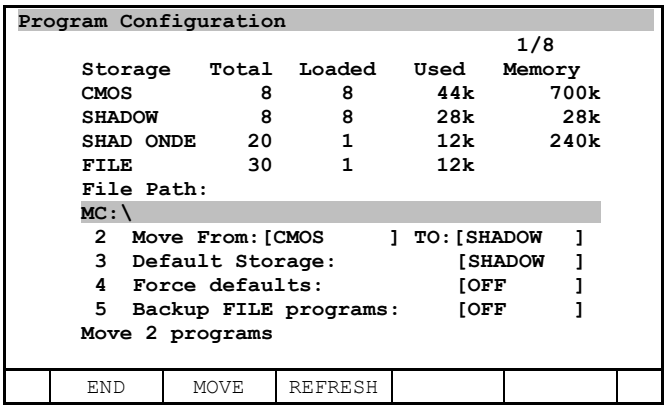
NOTE
These settings are reflected soon after changed.

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.

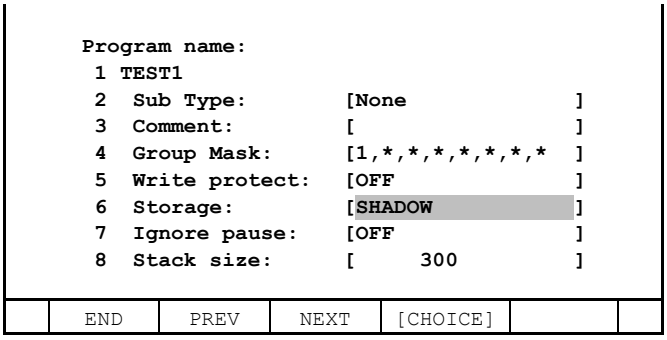
1 In the item “Move From To”, select the original storage in “From”, and select the new storage in “To”.
2 Press the F2, MOVE.
3 Press the F4, YES.
4 The following screen will be displayed and moving the storage finished.



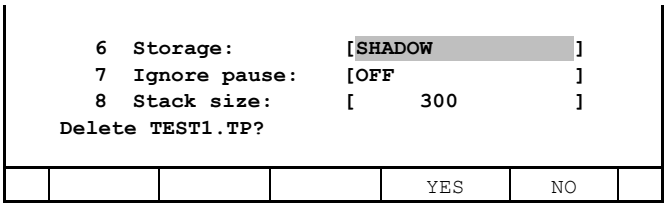
NOTE
Please confirm that there is enough memory space available in the new storage. If not, an alarm occurs and stops moving. Then, programs that are already moved do not go back the original storage. When you move large number of programs, please make backup in advance.

- 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.



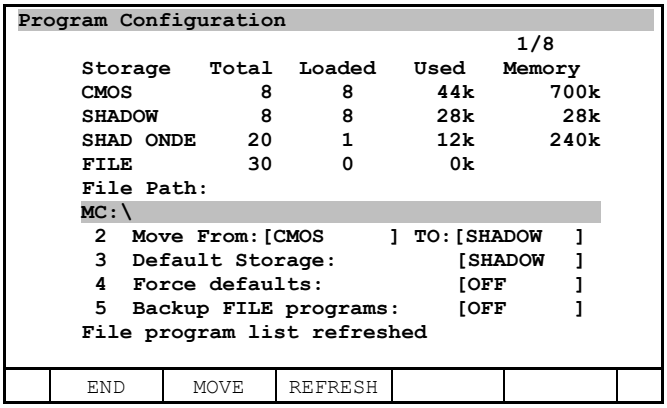
NOTE
Sometimes, the storage goes back to original storage soon after you change the storage. In such case, please confirm if the program is in edit, there is enough memory space in the new storage, or “File Path” is specified if FILE.

Procedure 29-4 Update FILE program list

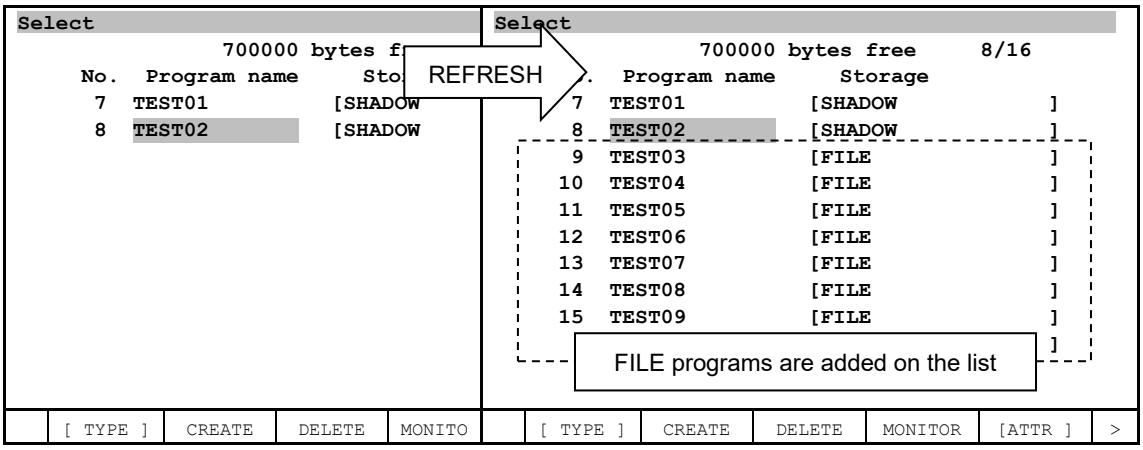
Step

- 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.



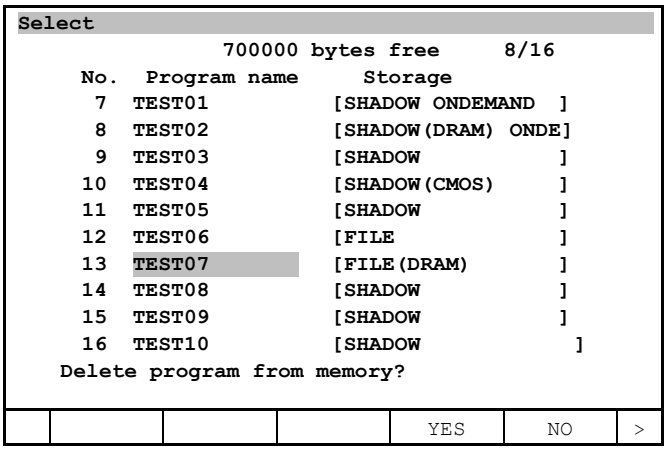
Procedure 29-5 Delete FILE, SHADOW ONDEMAND program

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.

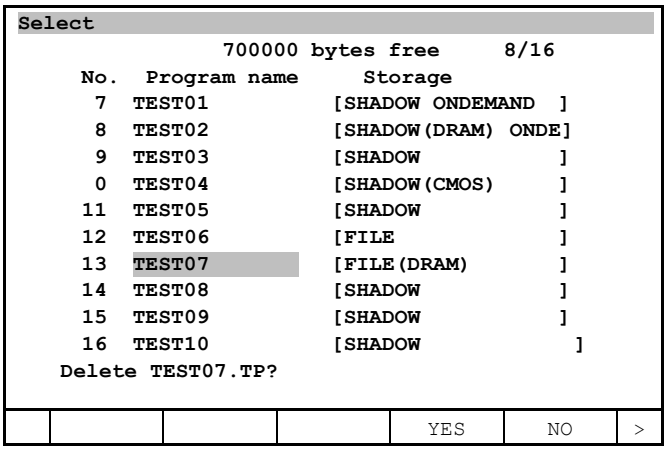
Please refer to the following process to delete FILE and SHADOW ONDEMAND program.

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.

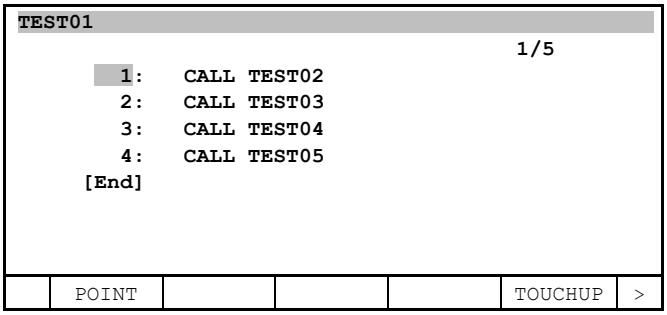
NOTE
”R709 TP DRAM/FILE Storage” option is necessary to load LS files that include “STORAGE” item. To load LS files into the controller without this option, you need to delete “STORAGE” item in advance.

29.3.4 Make Backup of Programs

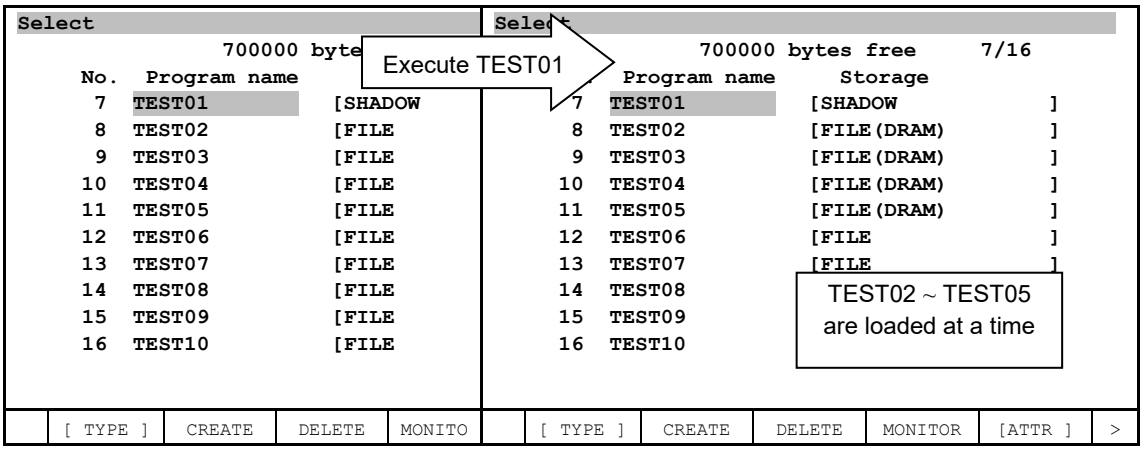
In File screen, you can make backup of SHADOW and SHADOW ONDEMAND programs as well as CMOS programs.
However, when you make backup of FILE programs, you need to set “Backup FILE programs” “ON” in Program Configuration screen.

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.

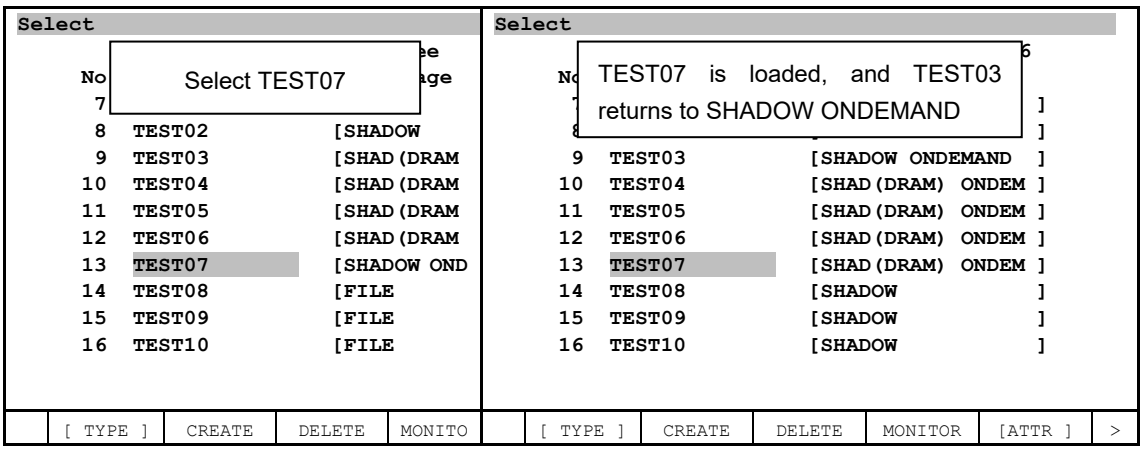


When TEST01 is executed, all the programs (TEST02~TEST05) that are called from TEST001 are loaded into DRAM.
If the size of a FILE program is large, it takes time to load it. So there is a time lag.
If these FILE programs are loaded in advance, TEST01 will start immediately.



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 value of system variable $TPDRAM_CFG.$MANAGE_MEM to “1”. By default this value is “0” 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.

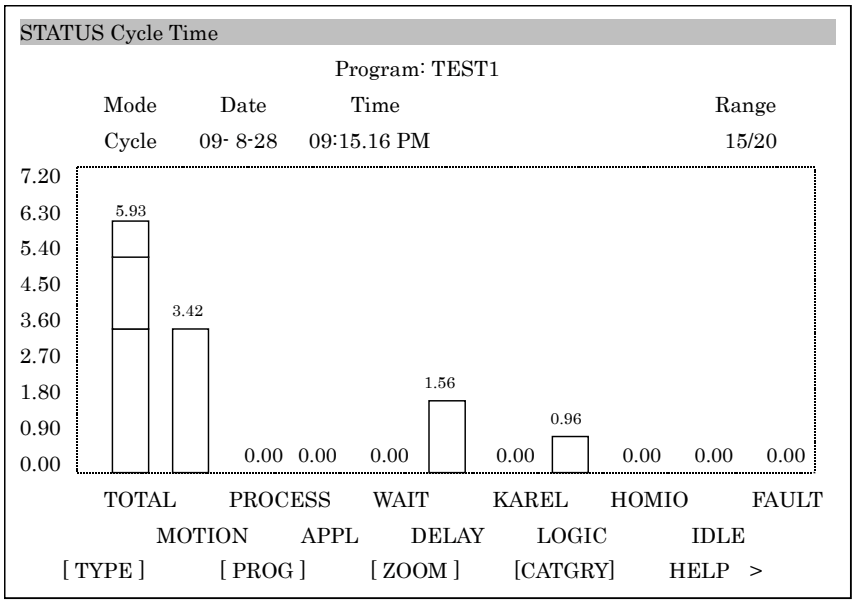
Item	Description
[CATGRY]	This function key allows you to choose which set of cycle time information is displayed. You can choose Total which displays the total of all of the categories using different color for each kind and the cycle time for each individual category. Or, you can choose an individual 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

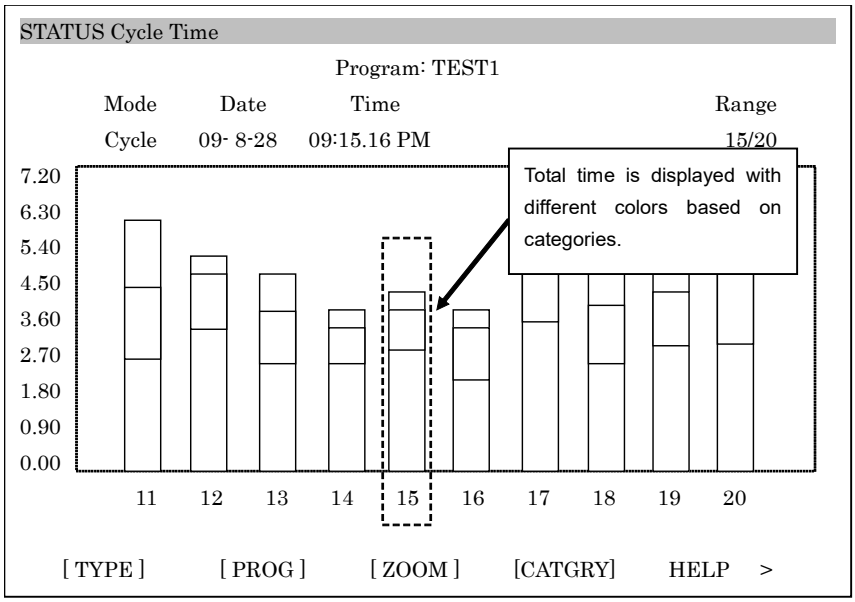
Cycle time data can be displayed in three modes by cycles, by hours, or line-by-line.

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.

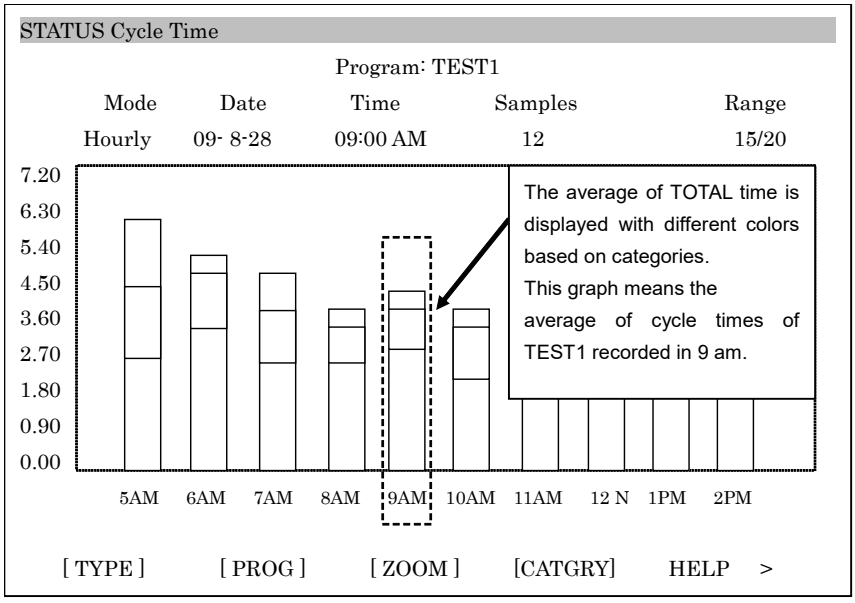


The following figure displays cycle times in ten cycles mode.



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, $CY_CONFIG.$LINEENABLE and $CY_CONFIG.$NUMLINES should be configured.
$CY_CONFIG.$LINEENABLE Valid when it is TRUE. (FALSE by default)
$CY_CONFIG.$NUMLINES Set the number you want to record. (0 by default)

After that,

1 In STATUS Cycle Time screen, press 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.

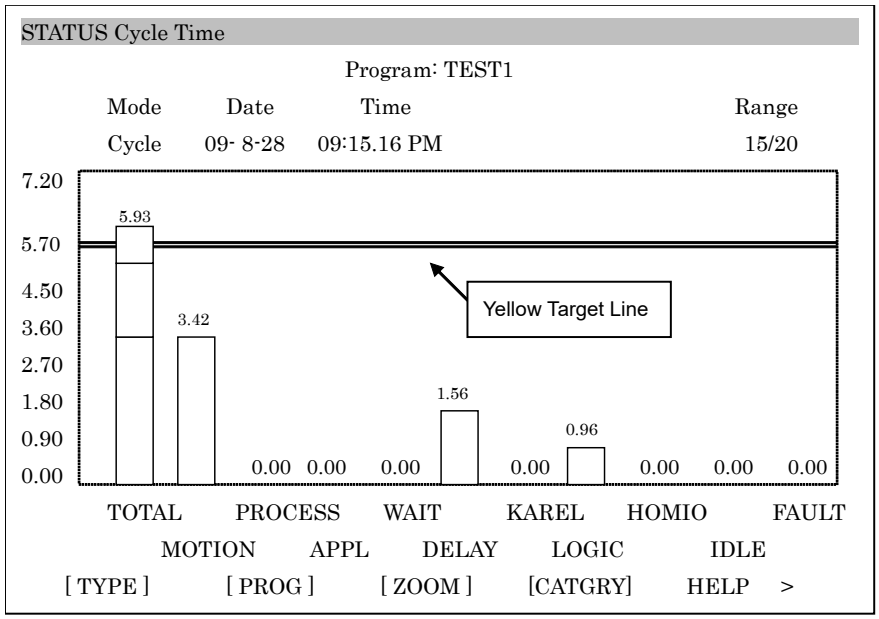
- Clear Line Data
  Clear the data for line by line mode. When you want to get a new line data, you should select Collect Line Data again.

- 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.

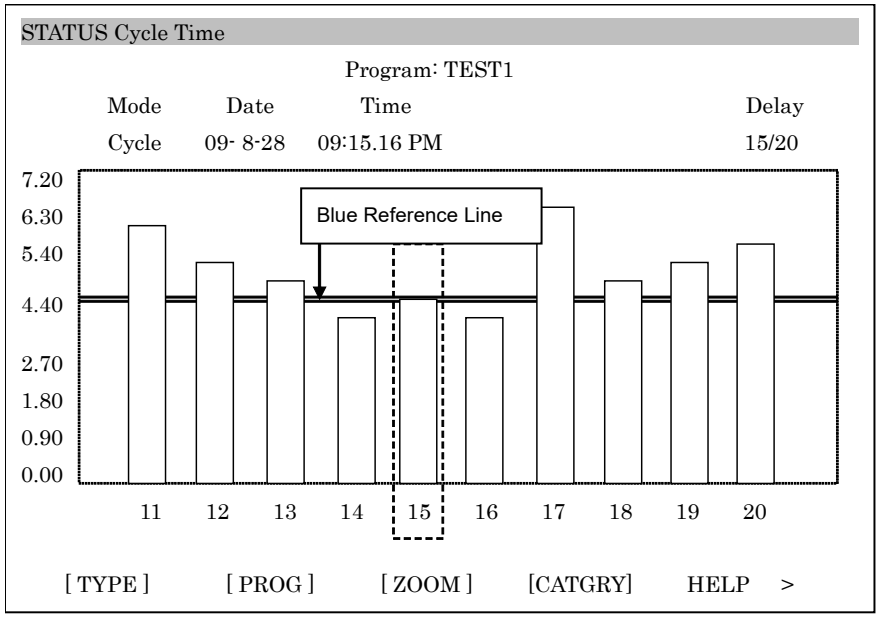
1 Press the [SELECT] key and select a program.
2 Press the [NEXT] key, and press the F2, DETAIL.
3 Press the F3, NEXT until Target Cycle Time (seconds) is displayed.
4 Press the [ENTER] key and type a new target cycle time.
5 Press the F1, END.
6 Press the [MENU] key, select STATUS, press the F1, [TYPE] 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

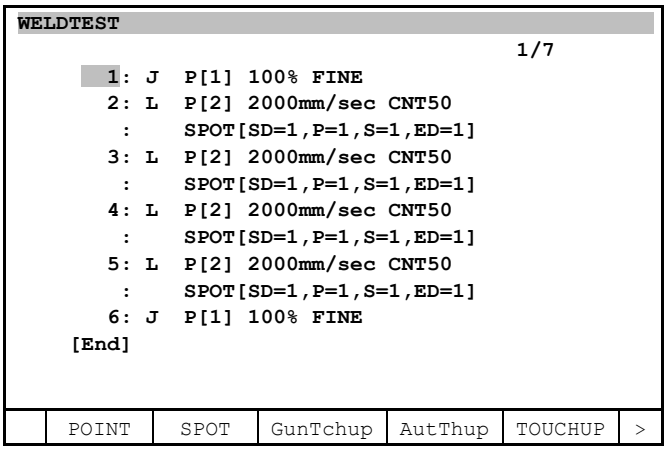
The cycle time of the current program can be recorded and be used as a reference. The recorded cycle time is displayed as a blue line. The reference cycle time line is displayed only in cycle mode and when an individual category is selected.

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.)

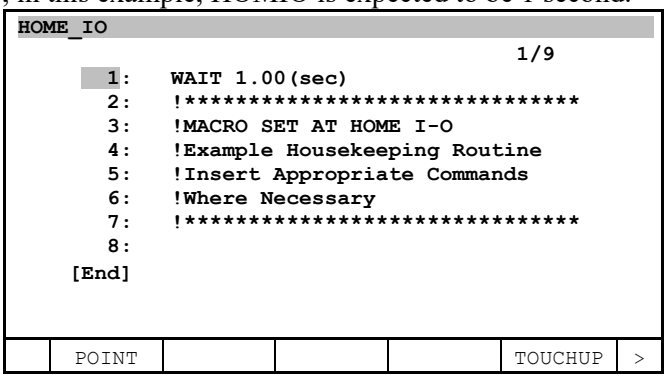


30.3 EXECUTION EXAMPLE OF SPOT PROGRAM

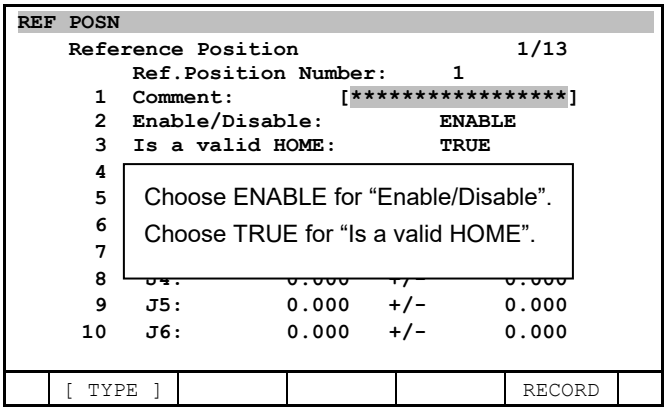
In this section, as an execution example of Cycle Time Logging function, the cycle time is recorded by executing the following Spot program, WELDTEST.
The cycle time of Spot program is not recorded by SHIFT-FWD execution. SOP cycle start, UOP_START, or PROD_START should be used.



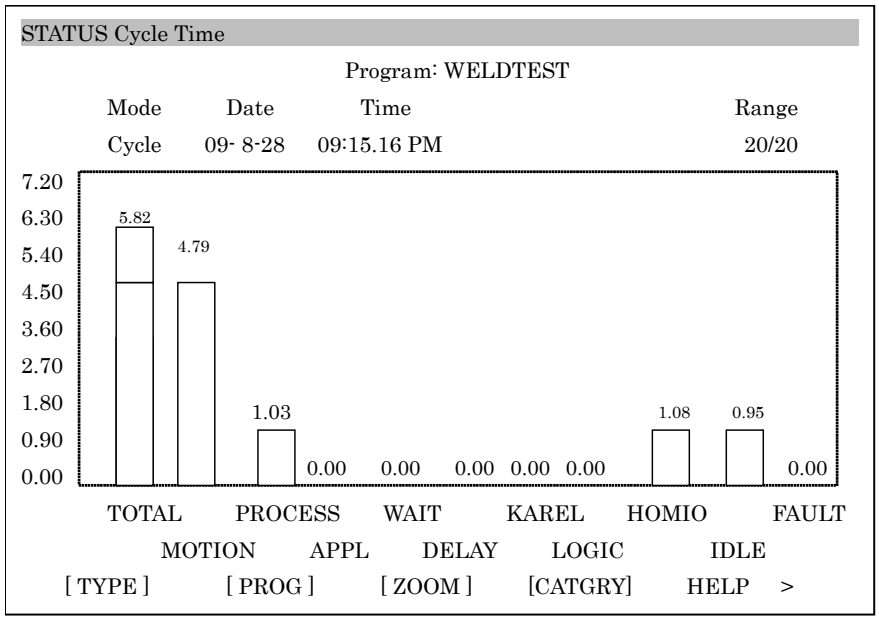
WELDTEST has 4 Spot instructions. In this example, the setting of welding time is 250 ms per one Spot instruction. So the total welding time is expected to 1 second and recorded as PROCESS cycle time.
The following HOME_IO program is also used. It has only one instruction of "WAIT 1.00 (sec)" and other 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 (a) 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

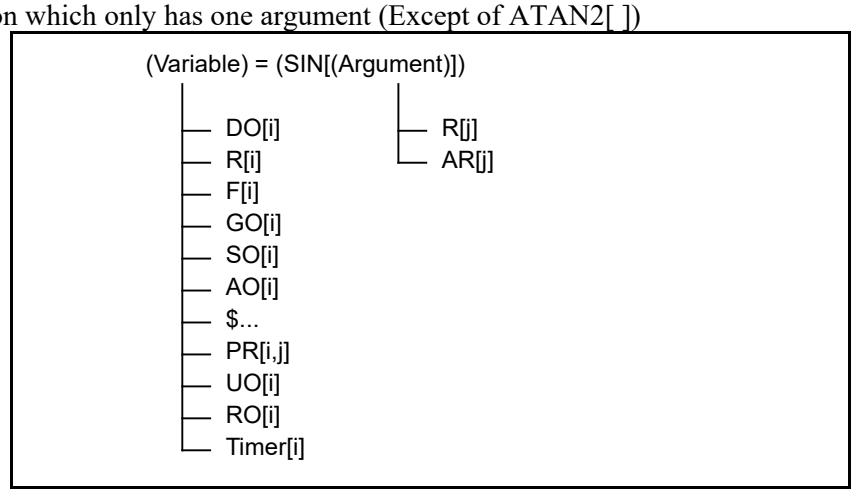
- Units of aforementioned functions are degrees.
- Conversion functions are not supported because the user can achieve them by multiplying a constant value (57.29579 or 0.017453 based on requirement).

31.2 INSTRUCTION FORMAT OF MATH FUNCTION

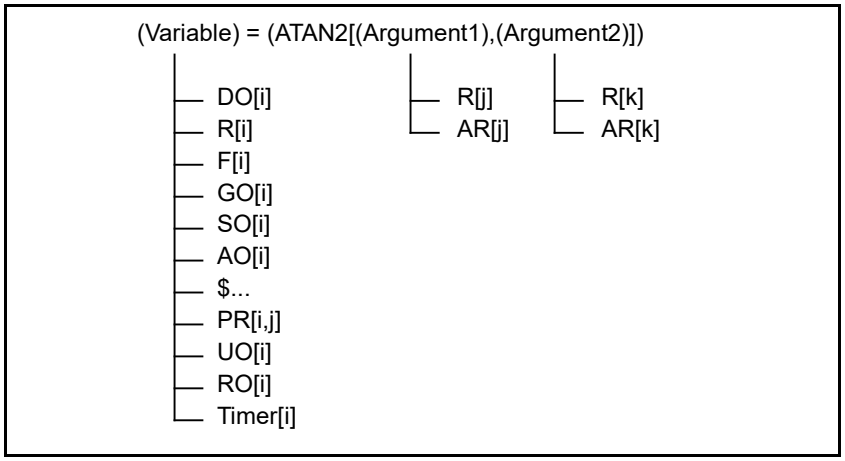
Instruction type of math functions is explained in the following example.

31.2.1 Instruction Format of Assignment Statements

- The function which only has one argument (Except of ATAN2[ ])

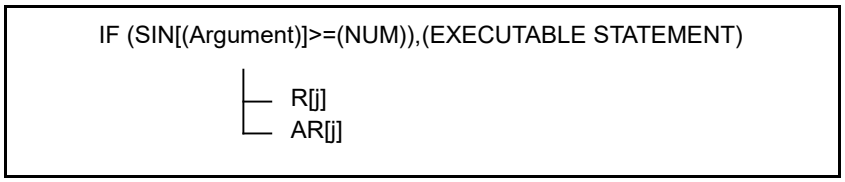


- The function which has two arguments (ATAN2[ ])



- 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

Mode	Maximum allowable number of items	Scanning time	Available data	Available operators
Standard mode	No restriction	(Number of items in standard mode / 300-Number of items in high-level mode)*ITP The number of items as used here refers to the total number of items in each mode background operation programs. One ITP is usually eight milliseconds.	SIN, COS, TAN, ASIN, ACOS, ATAN, ATAN2, SQRT, LN, EXP, ABS, TRUNC, ROUND	(,), =, <>, <, <=, >, >=, +, -, *, /, DIV, MOD
High-level mode	270	Eight milliseconds.		

- The scanning time of Math function is 2 or 3 times of normal items, because Math function instruction needs more time to calculate.
- The scanning time of ATAN2 instruction is 3 times of normal items.
- The other math function instructions of scanning time are 2 times of normal items.

31.5 TEACH MATH FUNCTION INSTRUCTION

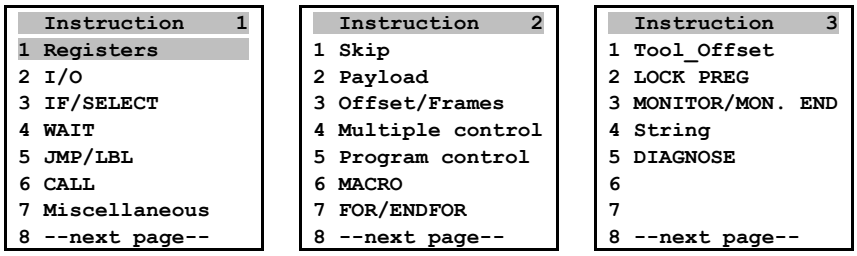
Math function can be used in mixed logic instruction.
As an example, the teaching process of following statement is as follows.

```
1 : R[1] = (SIN[R[2]])
```

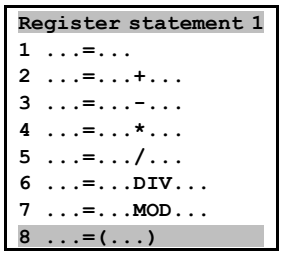
Procedure 31-1 Examples of teaching math function

Step

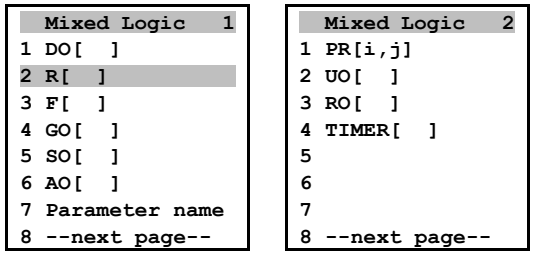
1 Move the cursor to a edited line, and press the F1, [INST] key.
2 Select Register.



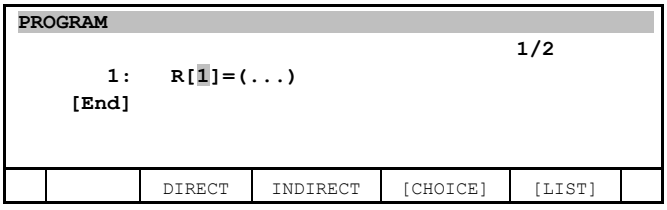
3 Select mixed logic instructions "…= (…)".



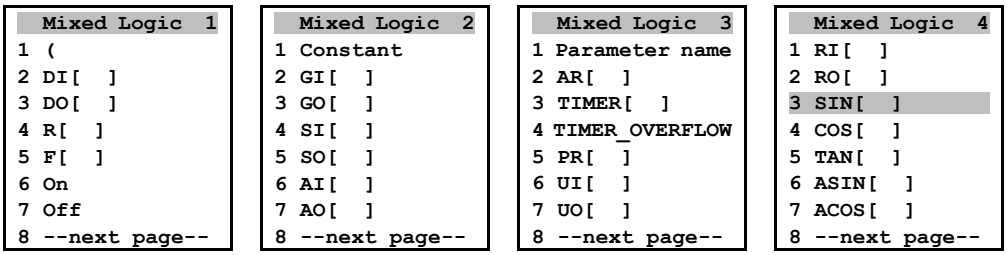
4 Select register "R[ ]". Then the cursor moves in the square bracket.



5 Input index 1.

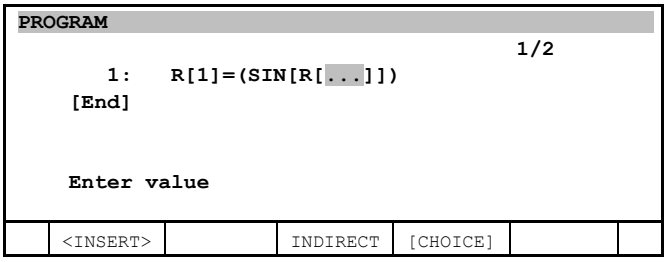


6 Then the cursor moves to the right side of equal sign.
7 Select "SIN[ ]".

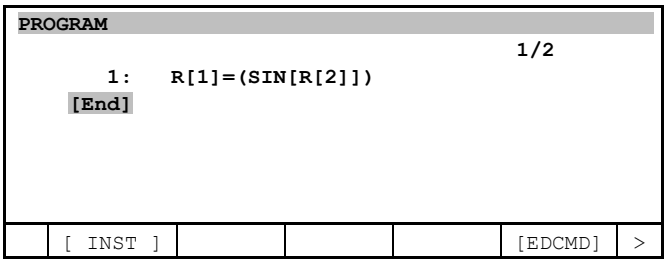


8 Then the cursor moves in the square bracket. And the default value R[…] is selected.
Arguments are "Register " and "Argument Register". (Not Constant)
The "Register" is default value of argument.
So to use "Argument Register", press the F3, [CHOICE] then select "Argument Register".

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

- It is possible to switch more than two servo tools without power off robot controller.
- Switching the air drive tool and the servo tool is also available.

What is a servo tool?

A servo tool is a tool having an axis that is driven by a servo motor controlled by a robot controller.

32.1.2 Basic Specification

The total number of changeable tool	9 for one-axis servo tool change 4 for two-axis servo tool change 3 for three-axis servo tool change Only one tool can be controlled at the same time.
The period of tool detach (standard)	0.4 sec (Battery type) / 0.4 sec(Battery less type) *1)
The period of tool attach (standard)	4 sec(Battery type) / 6 sec(Batteryless type) *1)
Tool axis pulse backup battery	Both battery type and batteryless type can be available.*2)

(*1) The period includes the internal processing period, not include the period of robot motion for changing the tools.

(*2) The batteryless type is not available for multi-axis servo tool change.

32.1.3 Restrictions

Restrictions matters

- Switching is possible only between servo tools that are set within the same motion group.
- Because servo tools that do not have a battery for tool axis pulse value backup require calibration after they are attached, these tools require longer processing time for tool change than battery-equipped tools.
- The motor of the tool change must be controlled by the same amp.
- The tool change instruction can not be used by multi task. (The tool change instruction can not be executed by plural tasks at the same time.)
- Installing both the servo tool change function and servo gun change function cannot be done.
- On a system with the servo tool change function installed, the dynamic brake release function cannot be used for servo guns.
- The motion group that the tool change function is enabled must be used a final group in the system.

Prohibition matters

- Do not force to detach the tool while the controller is off.
- The line tracking can not be used with the tool change function.

Notes on using the batteryless type

- When calibration motion type 1 or 2 is selected, calibration cannot be interrupted by a hold or emergency stop. (Program execution must not be stopped during tool attach processing.)
- When calibration motion type 3 or 4 is selected, the stroke limit must be expanded so that the tool axis touches the open limit.

- When calibration motion type 5 or 6 is selected, a detection switch such as a limit switch or dog switch must be installed in the tool.

For details of the calibration motion types, see Section 32.4, "TOOL CHANGE SETUP".

Notes on using multi-axis servo tool

- The servo tool number must be assigned to successive axes.
- The servo tool must have the pulse backup battery.
- To change two-axis servo tool, the two-axis tool change unit must be used.
- To change three-axis servo tool, the three-axis tool change unit must be used.
- Four (or more)-axis servo tool cannot be changed.

NOTE

When a Pulsecoder αiAR128(A860-2010-T341) shipped until May of 2006 is used, alarm SRVO-068(DTERR) occurs at turning power on and executing TOOL ATTACH instruction. Implement either of following countermeasures.

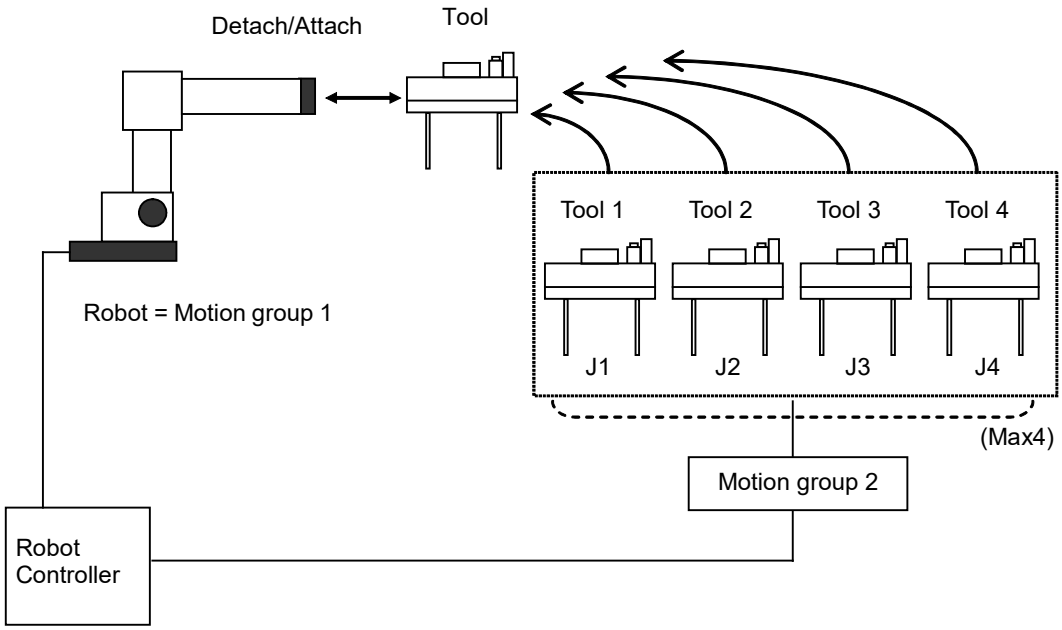
1. Replace Pulsecoder αiAR128 with one shipped after June of 2006.
2. Change system valuables as follows.
   - $SHC_ITF[#].$WAITTM_PCHK = 270
   - $SHC_ITF[#].$WAITTM_PWUP = 550
   - $SHC_ITF[#].$SVON_DELAY = 230
   - $SHC_ITF[#].$WAITTM_STRT = 650

   # is group number of servo tool group using Servo Tool Change Function.

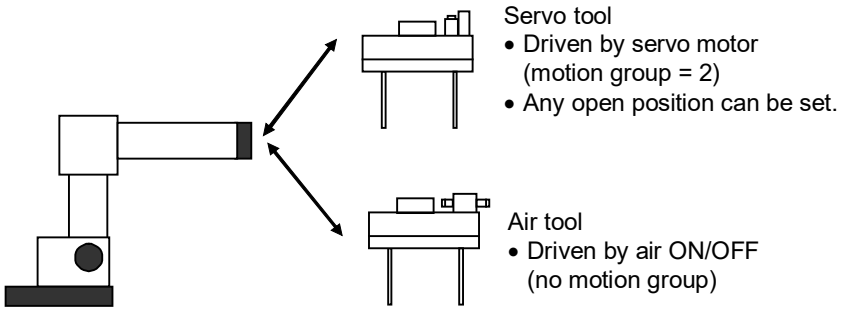
When you implement 2nd countermeasure, tool attach process is extended by about 2 seconds.

32.1.4 System Configuration

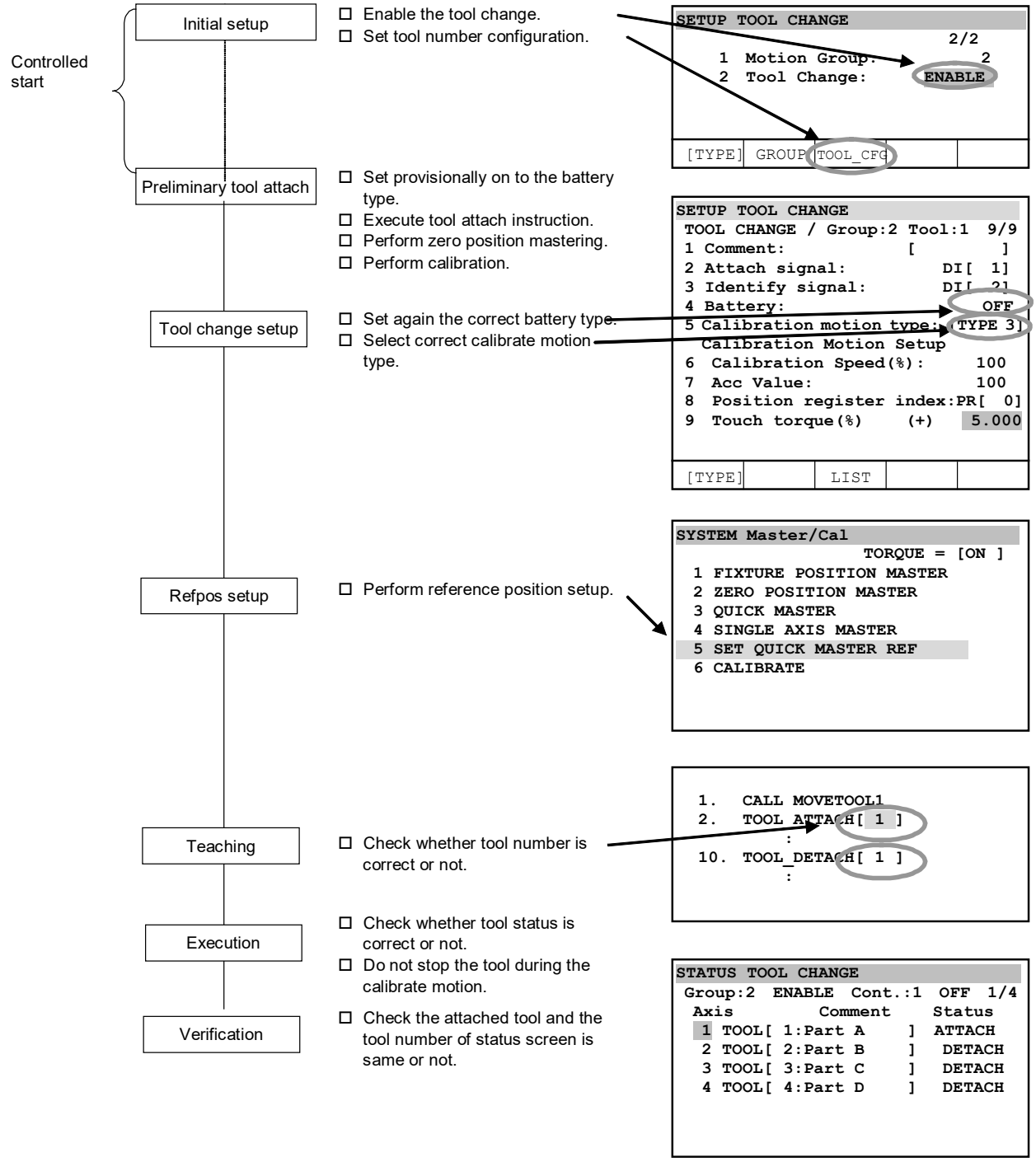
One-axis servo tool change system is constructed as follows.



You can replace the servo tool with the air tool by switching on the controller.



32.1.5 Outline of Installation



32.2 INITIAL SETUP

The following initial setup is needed to install the servo tool change system.

- Motion parameter setup of servo tool axes
- The configuration setup of servo tool number
- Enable the tool change function.

NOTE
These setup must be done at the beginning of system installation. These setup must be done only one time.

These initial setup is done at the controlled start. See Subsections 32.11.1, 32.11.2 and 32.11.3 for detailed information.

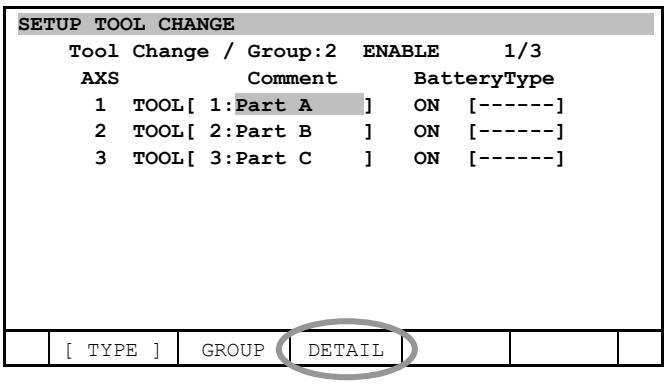
32.3 PRELIMINARY TOOL ATTACH OPERATION

When “Tool Change” is enabled as described in Section 32.2, "INITIAL SETUP", tools assigned to a target operation group are all detached.

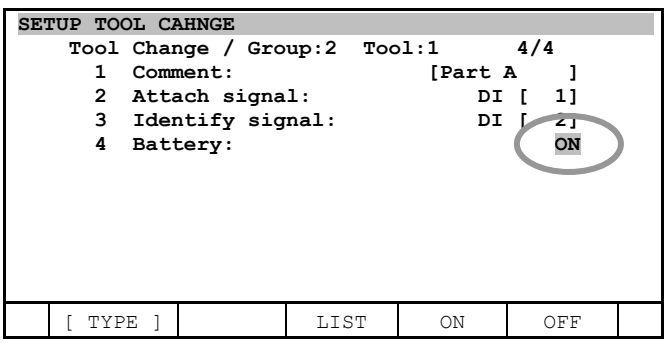
NOTE
Only when the tool change software is installed in a system in which servo tools have already been operated and a tool used previously is left attached, the tool is attached. In this case, preliminary tool attach steps 1 to 9 explained below are unnecessary.

When using a tool for the first time, be sure to follow the steps below to perform the preliminary attach operation, then proceed to Section 32.4, "TOOL CHANGE SETUP".

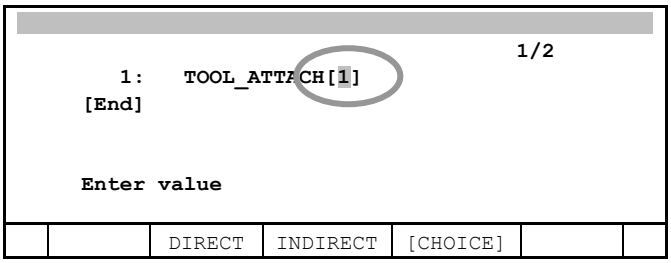
1 Press the [MENU] key to display the screen menu.
2 Select 6, SETUP.
3 Press the F1, [TYPE] to display the screen switch menu.
4 Select “Tool Change”. The tool change list screen will be displayed.
5 From the list, select one tool, and press the F3, DETAIL. The tool change setup detail screen will be displayed.



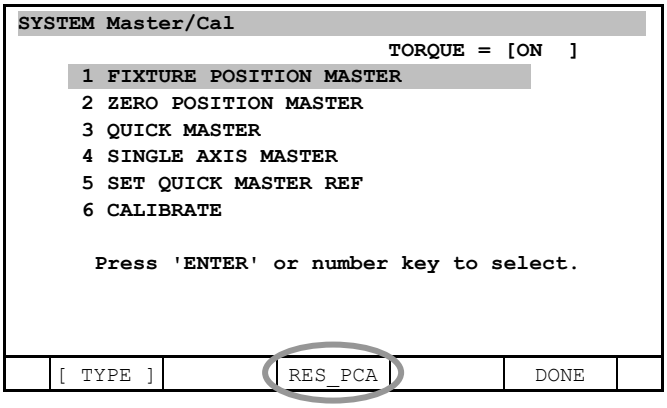
6 On the tool change setup detail screen, temporarily set the battery to ON. (Even when the actual tool has no battery, select ON in this step. This item is to be set again after the preliminary attach operation is completed.)



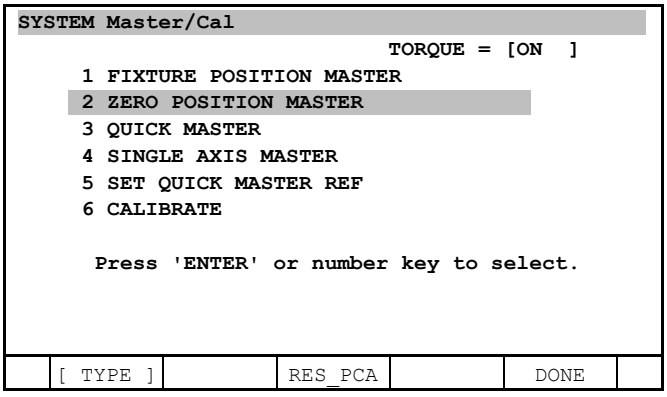
7 Attach a target tool to the tool changer.
8 Create a program to teach an attach instruction.
9 Execute the TOOL ATTACH instruction to attach the tool temporarily.



10 After the TOOL ATTACH instruction is executed, the following are issued:
SRVO-075 Pulse not established
SRVO-062 BZAL alarm
11 On the calibration screen, press the F3, RES_PCA, and select YES.



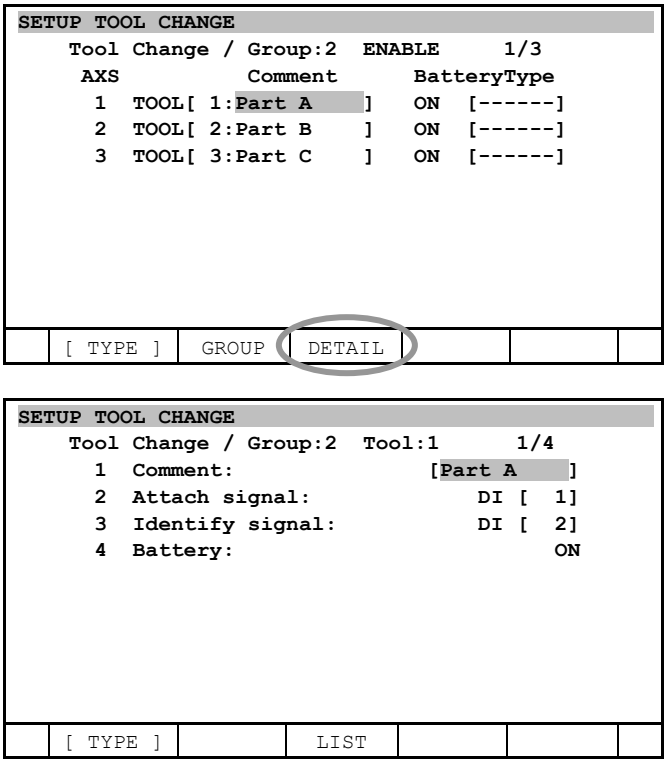
12 Cycle power of the controller.
13 Jog the servo tool axis.
Continue jogging the servo tool axis until the following message is displayed when the reset button is pressed:
SRVO-075 Pulse not established.
14 In the same manner as for ordinary servo tools, perform zero-position mastering for the servo tool axis.



15 After completing tool change setup and reference position setting for the first tool, detach the first tool by using the TOOL DETACH instruction, then perform the preliminary attach operation for the second and subsequent tools.

32.4 TOOL CHANGE SETUP

The tool change setup must be done for each tool.



1 Press the [MENU] key to display menu screen.
2 Select “6 SETUP”.
3 Press the F1, [TYPE] to display switch menu screen.
4 Select “Tool change” and the tool change setup screen will be displayed.
5 Select tool from the list of tools, then press the F3, DETAIL. The detail screen will be displayed.

Common setup (common setup item for all tools)

Comment	Comment for each tool.
Attach signal	This signal used for tool set verification. Available I/O type = DI / RI • This signal is check before tool attach/detach instruction. If this signal is OFF, the error occurs. • If the index is zero, the above checking is canceled.
Identify signal	This signal used for tool identification. Available I/O type = DI / RI / GI • This signal is check before tool attach/detach instruction. If this signal is OFF, the error occurs. • If the index is zero, the above checking is canceled.
Battery	Select battery type. ON = Battery backup type OFF = Batteryless type NOTE To change multi-axis servo tool, Battery must be set to ON.

(Select the following when Battery is set to OFF in the common settings.)

Calibration Motion Type	On battery-less tools, the position (pulse value) is lost when the tool is detached. So, calibration must be carried out when tool is attached. Calibration involves the following two operations: • Turning the motor two turns for positioning the Pulsecoder (This operation is simply referred to as "positioning" from here on.) • Calibrating at a preset reference position The user selects from the following types, 1 to 6, to suit the user's particular system requirements. (See Table 32.4 and Fig. 32.4.)

Calibration motion type Selection Criteria

- Types 1, 2

Select these types on systems capable of holding the tool detachment position as a result of braking the servo tool axis motor up to the next servo tool attachment.

> ⚠ **CAUTION**
> With type 1 and 2, if calibration is interrupted by a hold or emergency stop, reference position data is lost. In such a case, see Subsection 32.13.2, "The Robot Stopped during Calibration".

When these types are set, the position where the tool is detached is automatically memorized as the reference position, and calibration is carried out at that position after the tool is attached. For this reason, the position when the tool is attached must be the same as position when it is detached.
The allowable errors of the detach and attach positions must fall within the respective following ranges for the direct-acting axis and rotary axis:

Direct-acting axis:
±(gear ratio/2) mm or less around the detach position
Rotary axis:
±((360/gear ratio)/2) deg or less around the detach position

Note that positional misalignment occurs during tool attachment if the above ranges are exceeded.

The difference between types 1 and 2 is in the operating direction during calibration. (See Table 32.4 and Fig. 32.4.)

Type 1:
Specify this type when the tool is detached at a position where it is impossible to move the tool in the positive direction of manual feed by two or more turns of the motor.
Type 2:
Specify this type when the tool is detached at a position where it is impossible to move the tool in the negative direction of manual feed by two or more turns of the motor.

- Types 3, 4

Select these types on systems not capable of holding the tool detachment position as a result of the servo tool axis motor not having a brake up to the next servo tool attachment. On such a system, the reference position must be set to another position as the tool detachment position cannot be used as the reference position.

With these types, contact of the tool axis with the open limit is detected by using the motor torque, and the contact position is set as the reference position. (This is because, in the standard setting, the calibration motion causes the tool axis to move in the negative direction of manual feed.) For details on how to set the reference position, see Section 32.5, "SETTING THE REFERENCE POSITION".

The difference between types 3 and 4 is in the operating direction during calibration. (See Table 32.4 and Fig. 32.4.)

Type 3:
Specify this type when the tool is detached at a position where the distance to the open limit is two or more turns of the motor.

Type 4:
Specify this type when the tool is detached at a position where the distance to the open limit is two or less turns of the motor.

NOTE
1 In the calibration, to move the tool axis in the positive direction of manual feed (to use the limit on the positive side of manual feed), change the following system variable to 0:
$SVTCTOOLxx.$CLOSDIR = 1 → 0 (xx: Tool number)
2 With types 3 and 4, it is necessary to expand the stroke limit so that the tool axis touches the open limit.

- Types 5, 6

Like types 3 and 4, specify these types on systems not capable of holding the tool detachment position as a result of the servo tool axis motor not having a brake up to the next servo tool attachment. With types 5 and 6, however, a detection signal must be provided on the tool by means of a limit switch or dog switch, and the contact position must be set as the reference position. Install the detection switch in such a place that the detection switch is turned on when the tool axis is moved in the negative direction of manual feed (this is because, in the standard setting, the calibration motion causes the tool axis to move in the negative direction of manual feed). For details on how to set the reference position, see Section 32.5, "SETTING THE REFERENCE POSITION".
The difference between types 5 and 6 is in the operating direction during calibration. (See Table 32.4 and Fig. 32.4.)

Type 5:
Specify this type when the tool is detached at a position where the distance to the position where the detect signal turns ON is two or more turns of the motor.

Type 6: Specify this type when the tool is detached at a position where the distance to the position where the detect signal turns ON is two or less turns of the motor.

NOTE
1 In the calibration, to move the tool axis in the positive direction of manual feed, change the following system variable to 0:
$SVTCTOOLxx.$CLOSDIR = 1 → 0 (xx: Tool number)
2 With types 5 and 6, a switch such as a limit switch or dog switch must be installed in the tool.

Table 32.4 Calibration motion type

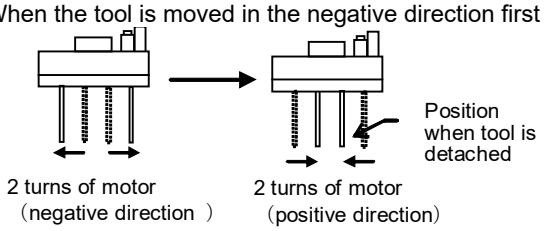
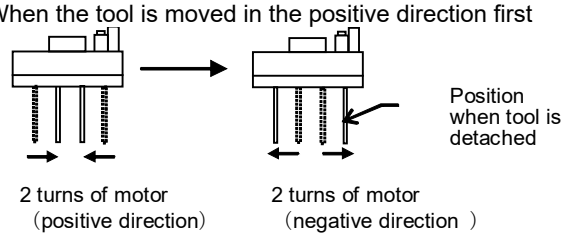
Type	Calibration Motion ($SVTCTOOLxx.$CLOSDIR is 1.)	Set Position of Reference position
Type 1	The tool is moved in the negative direction of manual feed by two turns of the motor and is subsequently moved in the positive direction by two turns, then the tool is calibrated. (NOTE 1) Set setting item (1) on the next page.	Position when tool is detached Reference position is not required. (NOTE 2)

Type	Calibration Motion ($SVTCTOOLxx.$CLOSDIR is 1.)	Set Position of Reference position
Type 2	The tool is moved in the positive direction of manual feed by two turns of the motor and is subsequently moved in the negative direction by two turns, then the tool is calibrated. Set setting item (1) on the next page. (NOTE 1)	Position when tool is detached Reference position is not required. (NOTE 2)
Type 3	The tool is moved in the negative direction of manual feed until touch torque is detected, then the tool is calibrated. Arrival at the limit is detected by motor torque. Set setting items (1), (2) and (3) on the next page.	Open limit position
Type 4	The tool is moved in the positive direction of manual feed by two turns of the motor and is subsequently moved to the limit on the negative side until touch torque is detected, then the tool is calibrated. Arrival at the limit is detected by motor torque. Set setting items (1), (2) and (3) on the next page.	Open limit position
Type 5	The tool is moved in the negative direction of manual feed until the detection signal is turned on, then the tool is calibrated. Set setting items (1), (2) and (4) on the next page.	Position where detection signal turns ON
Type 6	The tool is moved in the positive direction of manual feed by two turns of the motor and is subsequently moved in the negative direction until the detection signal is turned on, then the tool is calibrated. Set setting items (1), (2) and (4) on the next page.	Position where detection signal turns ON

NOTE

1 Tool master data is lost if the operation is canceled due to a hold or an emergency stop during calibration that is carried out during the tool change sequence when types 1 or 2 are selected. If this happens, manually carry out zero position mastering.

2 With types 1 and 2, the system automatically memorizes the position where the servo tool is detached as the reference position. This memorized reference position is used in calibration when the same tool is next attached. For this reason, when types 1 or 2 are selected, the detachment position and the attachment position must be the same. Do not select types 1 or 2 when a motor without a brake is used for the servo tool axis.

Type 1	Type 2
When the tool is moved in the negative direction first	When the tool is moved in the positive direction first
	

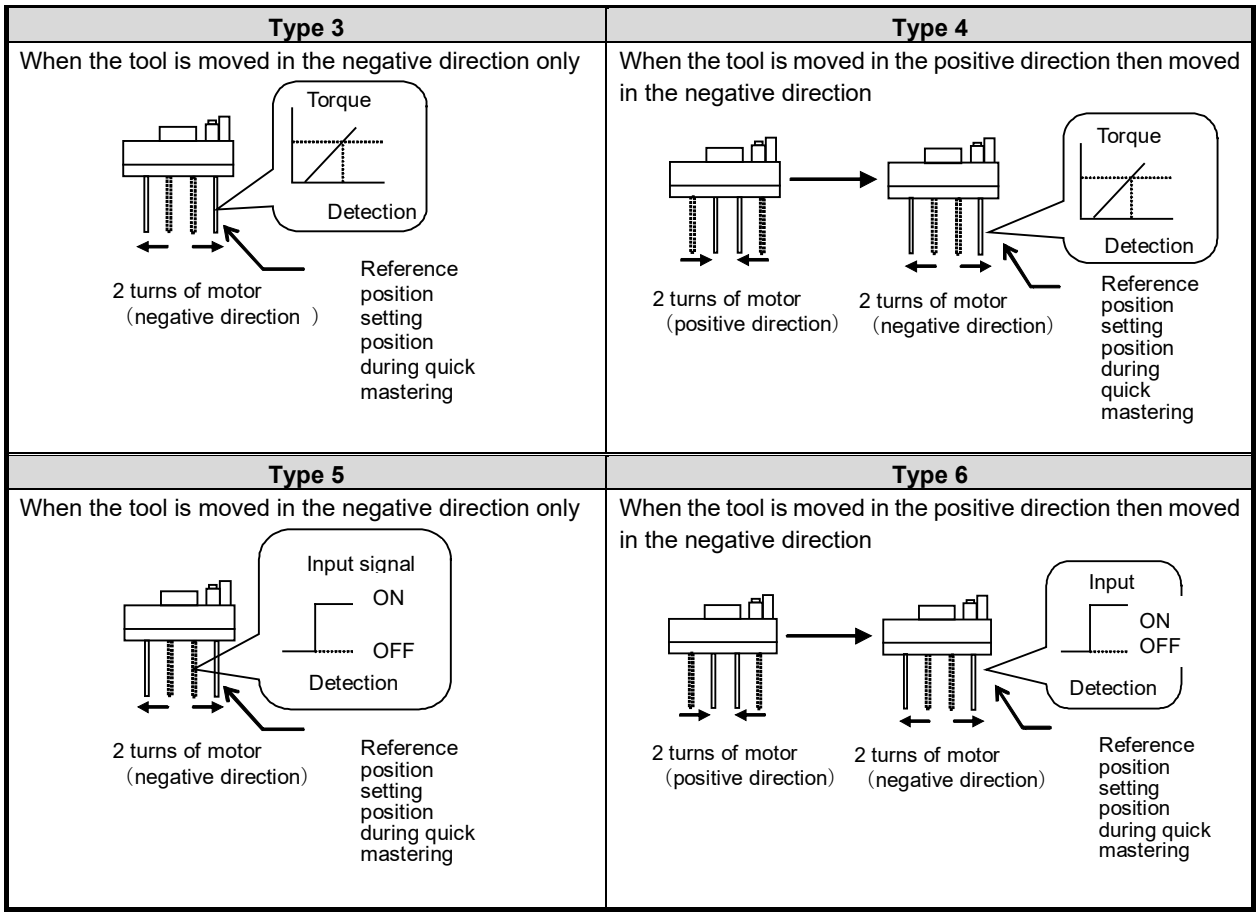


Fig. 32.4 Calibration motion type

The positive and negative directions in the above figure are determined on the assumption that $SVTCTOOLxx.$CLOSDIR = 1 is set. If this system variable is set to 0, the directions are reversed.

Screen for calibration types 1, 2

```
SETUP TOOL CHANGE
    Tool Change / Group:2  Tool:1       5/7
      1  Comment:                [          ]
      2  Attach signal:              DI [  1]
      3  Identify signal:            DI [  2]
      4  Battery:                       OFF
      5  Calibration motion type:    [TYPE 2]
         Calibration Motion Setup
      6    Calibration Speed(%):        100
      7    Acc Value:                 ACC100

 [ TYPE ]  |  LIST  [CHOICE]  |
```

Screen for calibration types 3, 4

```
SETUP TOOL CHANGE
    Tool Change / Group:2  Tool:1      5/9
      1  Comment:                [          ]
      2  Attach signal:              DI [  1]
      3  Identify signal:            DI [  2]
      4  Battery:                         OFF
      5  Calibration motion type:    [TYPE 4]
         Calibration Motion Setup
      6    Calibration Speed(%):         100
      7    Acc Value:                  ACC100
      8    Position register index:  PR[  1]
      9    Touch torque(%):     (+)    5.000
 [TYPE]          LIST  [CHOICE]
```

Screen for calibration types 5, 6

```
SETUP TOOL CHANGE
    Tool Change / Group:2  Tool:1      5/9
      1  Comment:                [          ]
      2  Attach signal:              DI [  1]
      3  Identify signal:            DI [  2]
      4  Battery:                         OFF
      5  Calibration motion type:    [TYPE 6]
         Calibration Motion Setup
      6    Calibration Speed(%):         100
      7    Acc Value:                  ACC100
      8    Position register index:  PR[  1]
      9    Detect signal:             DI [  0]
 [TYPE]          LIST  [CHOICE]
```

(1) Open limit operation setting (Set on all calibration types.)

Calibration speed	Specify the calibration motion operation speed.
Accelerate rate	Specify the calibration motion accelerate rate. Default is 100(%).

(2) Position register No. setting
(Set as follows when types 3 to 6 are set in the Calibration Setup.)

Position register No	Position register for holding the momentary position where arrival at the reference position (open limit or signal input position) was detected Set the position register specified here exclusively for calibration. (Do not use the same position register in the program, for example.)

(3) Open limit detection torque setting
(Set as follows when types 3 or 4 are set in the Calibration Setup.)

Touch torque (%)	Threshold value of the motor torque for detecting arrival at the open limit Set the value within range 0.0 to 100.0%.

(4) Detection signal setting (Set as follows when types 5 or 6 are set in the Calibration Setup.)

Detection signal	Type and No. of input signal for detecting arrival at the reference position • DI • RI Specify a No. available for each of the signals as the signal No.

32.5 SETTING THE REFERENCE POSITION

32.5.1 Battery-less Type Tools

On battery-less tools, the position (pulse value) is lost when the tool is detached. So, calibration must be carried out when tool is attached.
The position taken as the reference for calibration is called the "reference position".
On battery-less tools, the reference point must be set before the tool is detached.
How the reference point is set differs according to the calibration motion type.

Types 1, 2

The system automatically sets the reference position when the servo tool is detached.

Types 3, 4

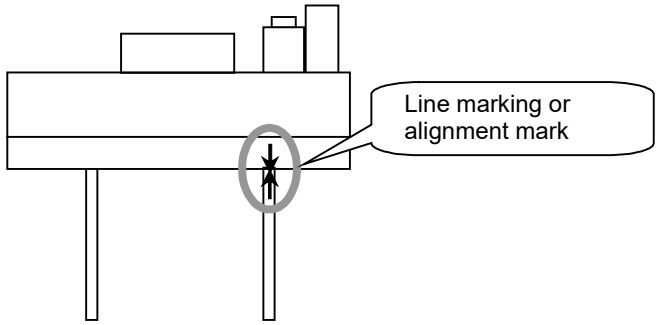
Create a program for detecting the reference position using the touch torque, and execute this program to calculate the reference position. For details, refer to Subsection 32.12.1.

Types 5, 6

Create a program for detecting the reference position using the detection signal, and execute this program to calculate the reference position. For details, refer to Subsection 32.12.2.

32.5.2 Battery-Mounted Type Tools

On battery-mounted type tools, the position (pulse value) is not lost even if the tool is detached. Therefore, the reference position used for calibration when the tool is attached need not be set as it needs to be on a battery-less type tool. However, that we recommend setting a reference position for quick mastering by marking a line or alignment mark as shown in the figure below for extra safety in the event that the battery comes loose. For details on how to set the reference position for quick mastering, refer to 32.12.3.



32.6 TOOL CHANGE INSTRUCTION

For changing servo tools, the following instructions are provided.

32.6.1 TOOL DETACH Instruction

This instruction can detach the servo tool without power off.

Format

TOOL DETACH[i]

i : Tool number (1 to 4)
Specify the tool number which you want to detach.
R[ ] and AR[ ] can be used.

Ex) TOOL DETACH[1]
TOOL DETACH[R[2]]
TOOL DETACH[AR[1]]

32.6.2 TOOL ATTACH Instruction

This instruction can attach the servo tool without power off. In case of batteryless tool, the mastering and calibration are performed automatically.

Format

TOOL ATTACH[i]

i : Tool number (1 to 4)
Specify the tool number which you want to detach.
R[ ] and AR[ ] can be used.

Ex) TOOL ATTACH[4]
TOOL ATTACH[R[3]]
TOOL ATTACH[AR[1]]

32.6.3 Sample Program

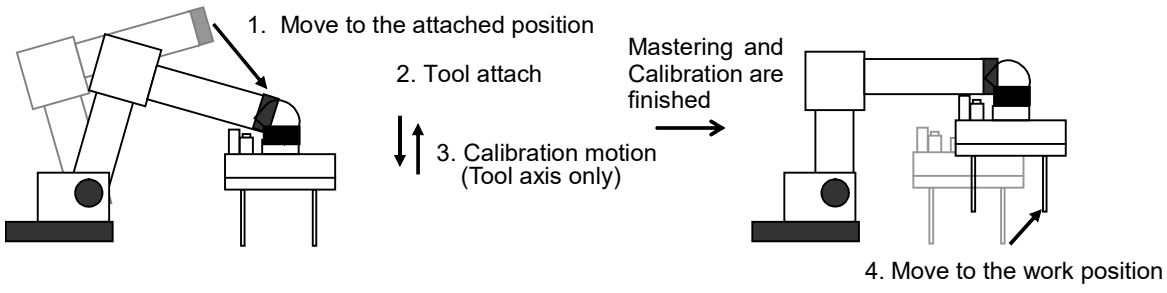
Sample program

	[Explanation]
1: J P[1:Approach] 100% FINE	1: Move to approach position.
2: L P[2:TOOL1 SET] 100mm/sec FINE	2: Set tool axis to 0 position.
3: TOOL DETACH[1]	3: Detach tool 1 (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 is merely a sample program. Create programs to suit the requirements of your system.

32.6.4 Forward Execution (FWD)

It is necessary to check that the tool number of instruction is same as the actual tool. And in the case of attach instruction, the calibration motion type must be set correct type.
In the case of batteryless tool, the calibration motion (3 in the following figure. See Fig. 32.4 for more detail) is done.



⚠CAUTION
In the case of calibration motion type is Type 1 or Type 2, if the motion is stopped by e-stop or HOLD during the calibration motion, the mastering data is lost. If the mastering data is lost, please do the zero mastering and calibration by manual again.
Otherwise,
1) The current position on TP is not matched to the actual position.
2) If the motion instruction is executed, the following alarm is occurred.
"MOTN-049 Attempt to move w/o calibrated"
3) The TOOL DETACH instruction does not work until manual mastering and calibration are finished.

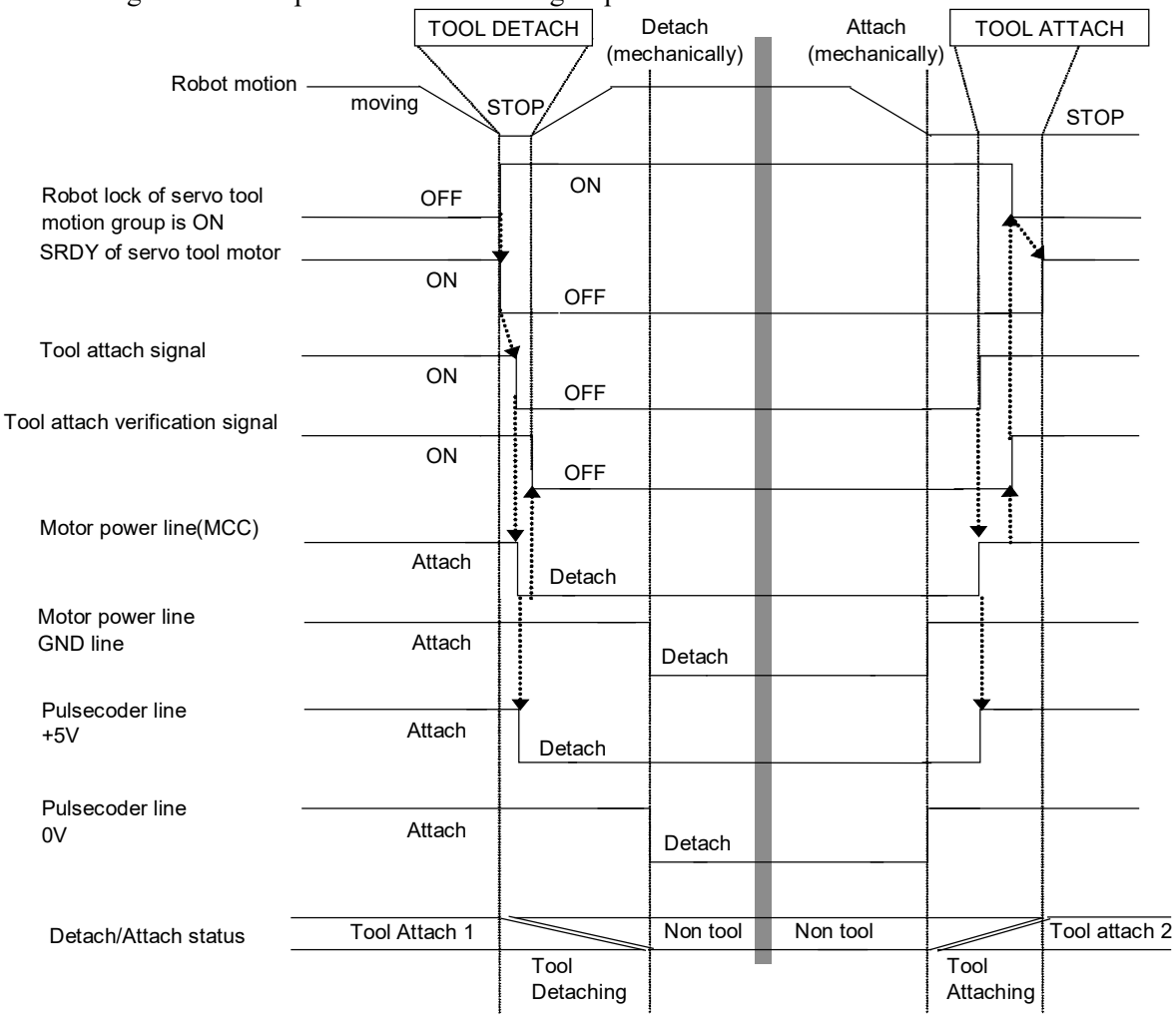
In the case of TOOL DETACH instruction, both robot and tool do not move at all.

32.6.5 Backward Execution (BWD)

Both TOOL ATTACH and TOOL DETACH instruction can not be execute by backward execution. Therefore, tool change can not be done in backward execution.

32.7 TOOL CHANGE SEQUENCE

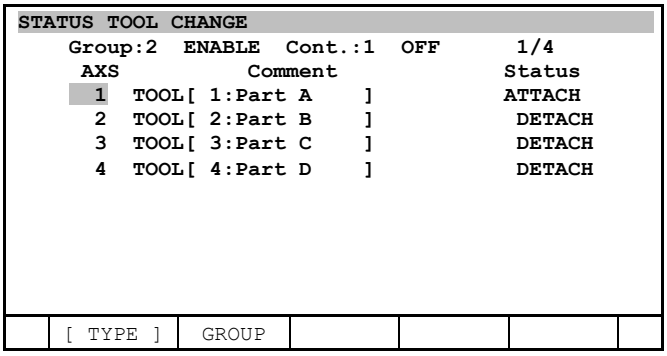
The tool change instruction performs the following sequence.



32.8 TOOL CHANGE STATUS

On the tool change status screen, verify that which tool is attached or detached now.

1 Press the [MENU] key to display menu screen.
2 Select “0 next page”, ”4 STATUS”.
3 Press the F1, [TYPE] to display menu switch screen.
4 Select “Tool change”, tool change status screen is displayed.



The following status are displayed.

AXS	The axis number of the motion group which is assigned to the current motion group.
Tool	The tool number which is assigned to the above axis.
Status	Tool status ATTACH : The tool is attached. DETACH : The tool is detached.
Comment	The comment of each tool.

32.9 TEACHING

In the tool change system, please note the following points.

32.9.1 Notice for Teaching

Position data

In the tool change system, the each tool is assigned to the each axis of the servo tool motion group. For example, if three kinds of tool are assigned to tool change system, the servo tool motion group has three axes data. This means three axes data are recorded when servo tool position is thought.

(Ex. The position data of tool1 which is assigned to first axis)

```
GP1:    UF : 0, UT : 1, Config : 'N, 0, 0, 0',
 X =   1879.25 mm,   Y = -1035.37 mm,   Z =    162.65 mm,
 W =   -178.67 deg,  P =   -13.68 deg,  R =    -37.32 deg
GP2:    UF : 0, UT : 1,
   J1 = 10.00 mm      J2 = 0.00 mm     J3 = 0.00 mm
```

Please note that this position data has not only tool 1 data but also tool 2 and tool 3 data. This means this position data can move tool 2 and tool 3. If tool 2 is attached, tool 2 moves to 0mm position. This specification is for the flexibility of the program. But, if the user attaches different tool, it may cause interfere of tool. To prevent this problem, please specify different tool number to the tool close direction for each tool.

By this function, the position data for each tool has the different tool number. Because of this, even if the user executes the program with different tool, the program is stopped by “INTP-253 Tool frame number mismatch”.

Re-teaching after the servo tool is added

When the servo tool is added, the program made before adding cannot use. Please re-teach positions on attaching the corresponding tool.

32.9.2 Sample Program

Sample program

```
          :
 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.

24:Change tool frame number to 2.

26:Handle with tool2.
27:Handle with tool2.
        :
99:Normal end
```

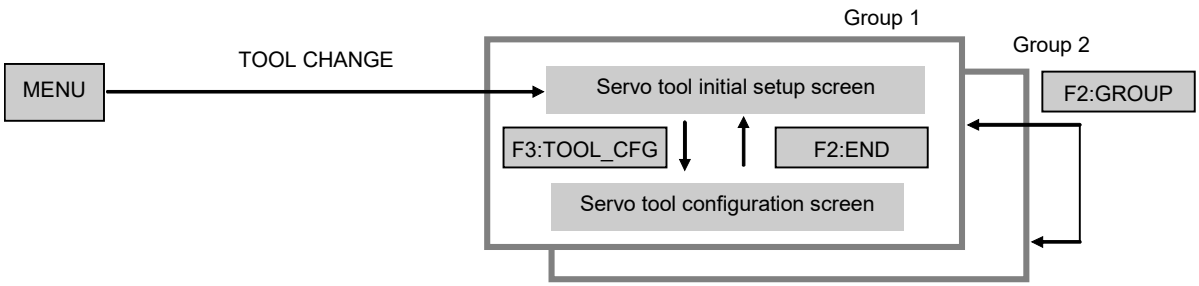
Move with tool number 1.

Move with tool number 2.

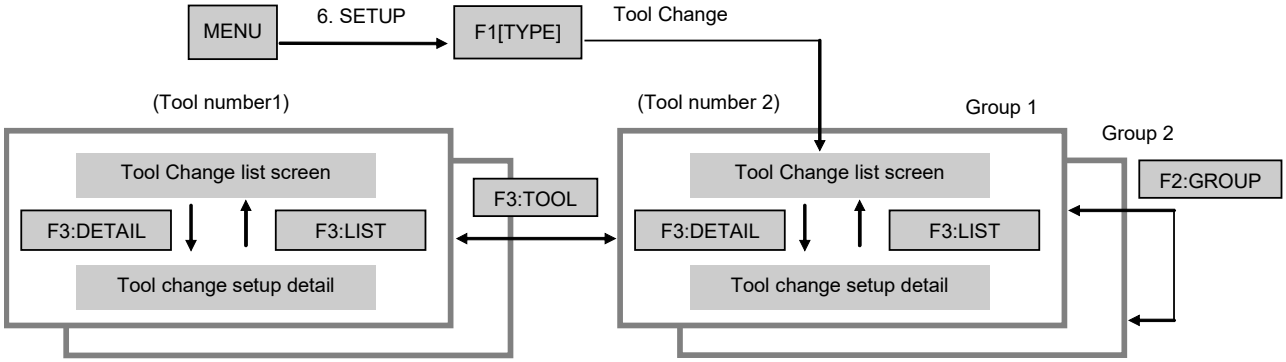
NOTE
This is sample program. Create the program related to your system.

32.10 CONSTRUCTION OF SERVO TOOL CHANGE SCREEN

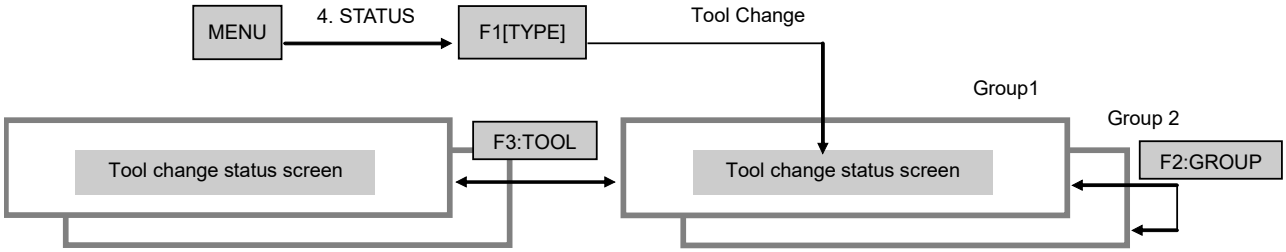
INITIAL SETUP SCREEN（This screen can be displayed at CONTROLLED START Menu.）



SETUP SCREEN



STATUS SCREEN



32.11 TOOL CHANGE INITIAL SETUP

The following explains how to make initial settings required for tool change system installation.

NOTE
The setup procedure explained below is required just once when the system is installed. Once settings are made, they need not be changed in ordinary use.

32.11.1 Setting Motion Parameters for Servo Tool Axes

Set tool axes used for the tool change function within each operation group according to the procedure below.

- 1st to 4th axes: Independent axis setting procedure

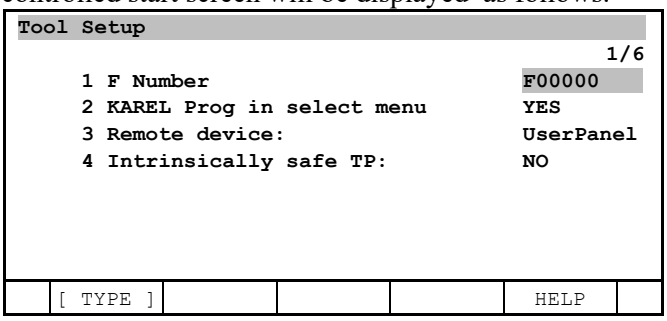
32.11.2 Assigning Tool Numbers to Servo Tool Axes

Open the servo tool initial setup screen from the controlled start menu, and set the following:

1 Turn off the power. While holding the [PREV] key and the [NEXT] key, turn on the power.
2 After a while, the following screen will be displayed. Select 3, Controlled start.

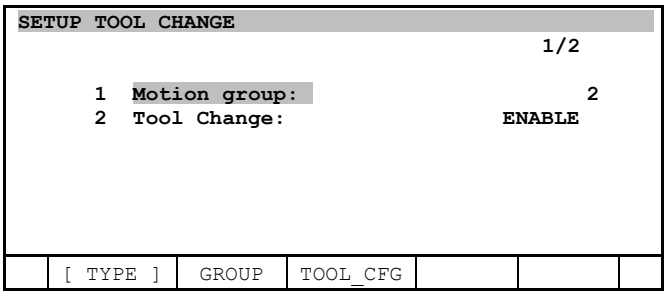
```
System version: V8.1035        2/10/2012

---------- CONFIGURATION MENU ----------

  1. Hot start
  2. Cold start
  3. Controlled start
  4. Maintenance

Select >
```

3 After a while, the controlled start screen will be displayed as follows.



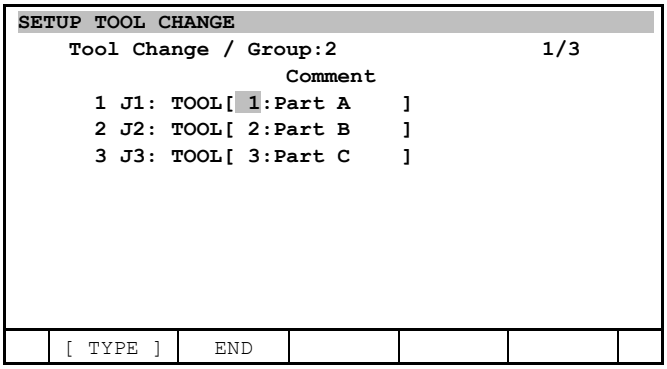
4 Press the [MENU] key to display the screen menu.
5 Select 0, NEXT then 4, TOOL CHANGE. The servo tool initial setup screen on the next page will be displayed .

Motion Group	Input the operation group number with which servo tools are set.
Tool Change	Enable or disable the tool change function.



The tool change function switches servo tools that are set within the same operation group. Note that it is impossible to switch between tools that are set in different operation groups.

6 Press the F3, TOOL_CFG, then the servo tool configuration screen will be displayed.



Set the items listed below for the tool axes included in the selected operation group.
If there is no problem with using the standard settings, the standard settings need not be modified.

TOOL	Set a number from 1 to 4.
Comment	Set a comment on the tool number. You can also set a comment on the tool change setup screen after cold start.

To change multi-axis servo tool, assign the tool number to multiple axes.

NOTE
1 The tool number must be assigned to successive axes for multi-axis servo tool change.
2 Four (or more)-axis servo tool cannot be changed.

32.11.3 Setting System Variables

To use servo guns on a system with the servo tool change function installed, set the following system variables to TRUE.

1. $SGCFG[equipment number of servo gun].$DUAL_AXIS[1] = FALSE -> TRUE
2. $SGCFG[1].$DUAL_AXIS[2] = FALSE -> TRUE

NOTE
The setting 1 must be done for all servo gun equipment.

32.12 TOOL CHANGE REFERENCE POSITION SETUP METHOD (BATTERY-LESS TYPE)

32.12.1 Reference Position Setup for Calibration Types 3 and 4

Reference Position Setup Program

Create the following program to calculate the reference position based upon the sign of the calculated touch torque.

NOTE
Only the servo tool axis operation group can be used as the operation group of the program to be used for setting the reference position.
Example) When the servo tool axis operation group is the 2nd group:
The operation group becomes [*,1,*,*,*,*,*,*].

Sample Program Operation Group=[*,1,*,*,*,*,*,*]

```
1: J P[1:ZERO POS] 10% FINE

   (when sign of touch torque is plus)
2: SKIP CONDITION $MISC[g].$HPD_TRQ[ga] > 2 *1)
   (when sign of touch torque is minus)
2: SKIP CONDITION $MISC[g].$HPD_TRQ[ga] < -2 *1)

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 manual feed touches the limit. Pay attention to the sign of the touch torque.
3: Move the tool in the negative direction of manual feed 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 user alarm.

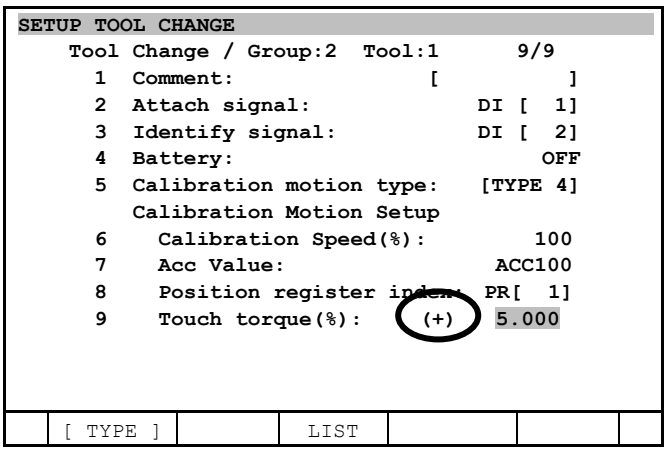
NOTE
The above is merely a sample program. Create programs to suit the requirements of your system.
g: Operation group number of currently attached tool
ga: Axis number of currently attached tool
If the currently attached servo tool axis is taken as the 1st axis of the 2nd group, then g and ga are as follows: g=2, ga=1

Program Operation Adjustment Method

Adjust the value at the shaded section in the sample program to adjust program operation. Operation adjusted here is used as calibration motion that is carried out when attaching the tool.

*1) Touch torque
Increase the value from about ±2, and calculate the value at which erroneous detection of contact does not occur.
The sign of touch torque is the sign that is currently displayed at Touch torque in the following Calibration Setup screen.

NOTE
After confirming the sign in this screen, teach it to the program. Do not make a mistake when entering the sign. If you do so, contact with the open limit may not be detected.



*2) Program operation speed
To set the same calibration motion that is carried out when attaching the tool, execute this program at the same override (normally, 100%) as when the attach instruction is executed for the tool used during actual production. For this reason, reduce the program operation speed so that contact is made with the open limit at a safe speed even if the program is executed at a high override.
For added safety, increase the program operation speed value successively from 1%.

*3) Accelerate rate
Decrease the accelerate rate if motion is quick even if the program operation speed is 1%.

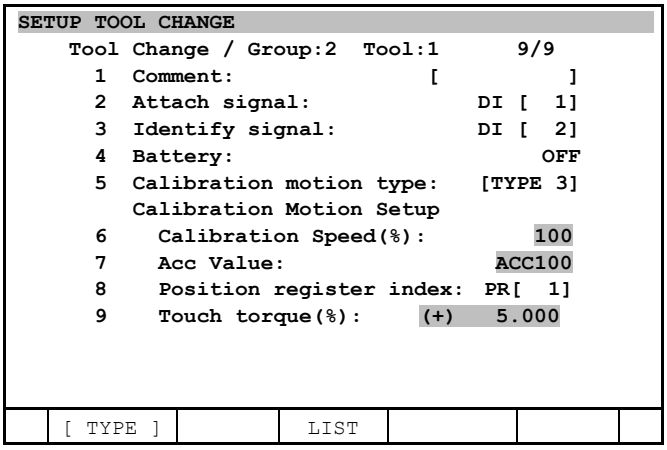
Reference Position Setup

After you have adjusted operation in the program, execute the program to bring the servo tool axis into contact with the open limit.
Set up the reference position according to the procedure described in 32.12.3 with the servo tool contacting the open limit.

Tool Change Setup

To set operation in the newly adjusted program to the same motion as calibration motion that is carried out when attaching the tool, reflect the program operation speed, accelerate rate and touch torque adjustment values in the tool change setup.

1 Press the [MENU] key to display the screen menu.
2 Select 6 SETUP.
3 Press the F1, [TYPE] to display the screen selection menu.
4 Select Tool Change. The Tool Change Setup List screen will be displayed.
5 Select the tool to be adjusted from this list, and press the F3, DETAIL to display the Tool Change Details Setup screen.
6 Set the operation speed, accelerate rate and touch torque adjustment values at the shaded sections in the following screen:



32.12.2 Reference Position Setup for Calibration Types 5 and 6

Provision of Detection Signal

With calibration types 5 and 6, the position where the detection signal is input is taken as the reference position during quick mastering.

The user must install a limit switch for providing the detection signal capable of assuring the absolute position of the tool.

Reference Position Setup Program

Create the following program to calculate the reference position based upon the provisioned detect signal.

NOTE

Only the servo tool axis operation group can be used as the operation group of the program to be used for setting the reference position.

Example)

When the servo tool axis operation group is the 2nd group: The operation group becomes [*,1,*,*,*,*,*,*].

Sample Program Operation Group=[*,1,*,*,*,*,*,*]

```
1:J P[1:ZERO POS] 10% FINE
2:SKIP CONDITION DI[ 1*1)] =ON
3:J P[2:SWITCH POS] 1% *2) FINE ACC100
   *3)SKIP,
  LBL[1],PR[1]=JPOS
4:END
5:LBL[1]
6:UALM[1]
```

[Explanation]
1: Servo tool axis 0 position (The position need not necessarily be the 0 position.)
2: Set turning ON of the provided detection signal as the scan condition.
3: Move tool to position where detection signal turns ON. (Jump to label[1] if the detection signal does not turn ON.)
4: Normal end
5: Jump to here if detection signal does not turn ON.
6: Generate user alarm

NOTE
The above is merely a sample program. Create programs to suit the requirements of your system.

Program Operation Adjustment Method

Adjust the value at the shaded section in the sample program to adjust program operation.
Operation adjusted here is used as calibration motion that is carried out when attaching the tool.

*1) Detection Signal
Assign the provided detection signal to DI[ ] or RI[ ].
*2) Program Operation Speed
To set the same calibration motion that is carried out when attaching the tool, execute this program at the same override (normally, 100%) as when the attach instruction is executed for the tool used during actual production. For this reason, reduce the program operation speed so that the switch will not be broken even if the program is executed at a high override.
For added safety, increase the program operation speed value successively from 1%.
*3) Accelerate rate
Decrease the accelerate rate if motion is quick even if the program operation speed is 1%.

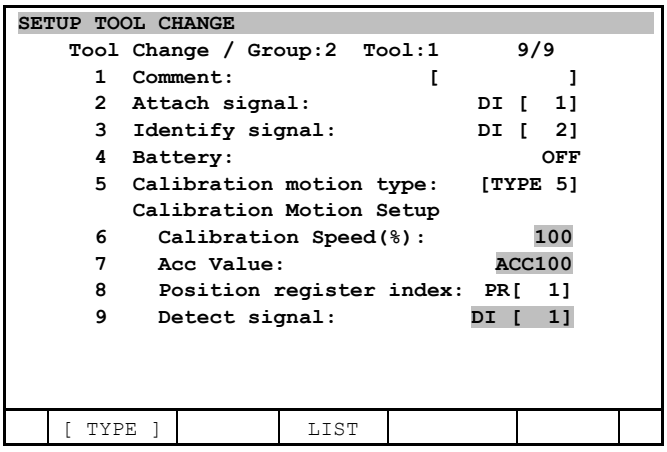
Reference Position Setup

After you have adjusted operation in the program, execute the program to stop the servo tool axis at the position where the detection signal turns ON.
Set up the reference position according to the procedure described in 32.12.3 as it is.

Tool Change Setup

To set operation in the newly adjusted program to the same motion as calibration motion that is carried out when attaching the tool, reflect the program operation speed, accelerate rate and detection signal adjustment values in the tool change setup.

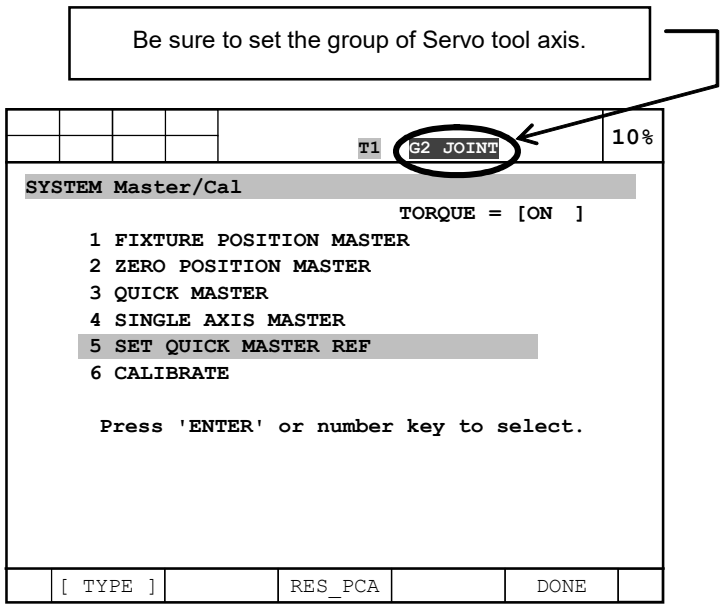
1 Press the [MENU] key to display the screen menu.
2 Select 6 SETUP.
3 Press the F1, [TYPE] to display the screen selection menu.
4 Select Tool Change. The Tool Change Setup List screen will be displayed.
5 Select the tool to be adjusted from this list, and press the F3, DETAIL to display the Tool Change Details Setup screen.
6 Set the operation speed, accelerate rate and detection signal adjustment values at the shaded sections in the following screen:



32.12.3 Quick Mastering Reference Position Setup

Calibration (quick mastering) is automatically carried out when the servo tool is changed. However, the reference points must be set to each of the tools to enable automatic calibration. The reference points are set with the robot in a cold start state.
Follow the procedure below to set the reference points:

1 Press the [FCTN] key to select 3 CHANGE GROUP. Change the operation group to the desired servo tool group (e.g. 2nd group).
2 Press the [MENU] key to display the screen menu.
3 Press 0 NEXT and 6 SYSTEM.
4 Press the F1, [TYPE] to display the Screen Selection menu.
5 If Master/Cal is not displayed in the menu, select Variables to display the system variables screen. Set the $MASTER_ENB system variable to "1".
6 Select Master/Cal at 4. to display the calibration screen.
7 Select “5 SET QUICK MASTER REF” and F4, YES.
8 The message "Quick master reference point is set." is displayed. This indicates that setting of the reference point is completed.



32.13 TROUBLESHOOTING

32.13.1 The Attach Instruction is Executed when the Tool is not Attached.

Symptom

The tool is not actually attached. So, a communications alarm such as "SRVO-068 DTERR alarm" occurs on the Pulsecoder. Correct this by Remedy 1 or Remedy 2 below according to the present conditions.

Remedy 1

1 Turn the machine lock ON in the servo tool operation group (normally, group 2) in the Test Cycle screen.
2 Move the robot by manual feed so that the robot and tool are physically attached.
3 Cycle power of the controller.

Battery-less tool (calibration motion types 1, 2)

4 The tool pulse values and mastering data are lost. So, first move the tool by manual feed to cancel the "SRVO-075 Pulse not established" alarm.
5 Then, carry out tool zero position mastering and calibration manually.

Battery-less tool (calibration motion types 3, 4, 5, 6)

6 The tool pulse values and mastering data are lost. So, move the tool by manual feed to cancel the "SRVO-075 Pulse not established" alarm.
7 Carry out tool zero position mastering and calibration manually. The tool 0 position may be an approximate position.
8 Execute the detach instruction to temporarily detach the tool. Also, physically detach the tool from the robot. Otherwise, the tool will not be calibrated at the next attach operation.
9 Execute the attach instruction again to attach the tool. Calibration this time is carried out at the reference position to restore the position.

Battery mounted tool

10 The tool is attached and positioned, so it can be used as it is.

Remedy 2

1 Directly change the system variable to detach the tool. To do this, set the following system variable to "0":
$SCR_GRP[servo tool operation group No. (normally, 2)].$AXISORDER[all 1 to 9] = 0
2 Cycle power of the controller.
3 The robot starts up with the tool in a detached state. Move the robot by manual feed to physically attach the robot and tool.
4 Execute the attach instruction to attach the tool.
5 The tool can be used as it is after it is attached regardless of the battery type.

32.13.2 The Robot Stopped during Calibration.

Symptom

This symptom occurs only on battery-less tools. Note that the symptom and remedy vary according to the preset calibration type.

Battery-less tool (calibration motion types 1, 2)

The "TOOL-012 Tool mastering data is lost" alarm has occurred.

Battery-less tool (calibration motion types 3, 4, 5, 6)

Alarm has not occurred.

Remedy

Calibration motion types 1, 2

1 The tool pulse values and mastering data are lost. So, first move the tool by manual feed to cancel the "SRVO-075 Pulse not established" alarm.
This alarm may not be occurring. If so, skip this step and proceed to step 2.
2 Then, carry out tool zero position mastering and calibration manually.

Calibration motion types 3, 4, 5, 6

1 Tool operation can be resumed as it is.

32.13.3 Calibration Motion Failed.

Symptom

The "TOOL-020 Calibrate motion is failed" alarm occurred.
This symptom occurs only when the calibration motion type is 3, 4, 5 or 6 on a battery-less tool.
This alarm occurs when the touch torque or detection signal conditions are not satisfied during calibration.
In both of the following Remedies 1 and 2, the tool is calibrated again. So, calibration motion will fail again if the touch torque or detection signal conditions are not satisfied. Before you remedy this symptom, make sure that the touch torque or detection signal conditions are correct.

Remedy 1

Carry out steps 6 to 9 of Subsection 32.13.1 Remedy 1.

Remedy 2

Same as Remedy 2 of Subsection 32.13.1.

32.13.4 A Different Tool from that Specified by the Attach Instruction is Attached.

Symptom

The tool is attached to the robot, but the tool's current position is the wrong value. On a battery-less tool, an alarm may cause the robot to stop during calibration.

Remedy

The following describes an example where tool 2 (actually attached tool) is attached to tool 1 (tool No. taught by the attach instruction).

1 In this state, the tool cannot be detached from the robot by the detach instruction. So, directly change the system variable to detach the tool. To do this, set the following system variable to "0":

$SCR_GRP[servo tool operation group No. (normally, 2)].$AXISORDER[all 1 to 9] = 0

2 Cycle power of the controller.
3 The tool starts up in a detached state.
4 Execute the TOOL ATTACH[2] instruction to attach tool 2. (In this state, tool 2 should be physically connected to the robot.)

When tool 2 is a battery-less tool and the calibration motion type is 1 or 2:
If tool 2 is moved when attached by mistake instead of tool 1, tool 2 will move and be at a position different to when it was detached. So, tool 2 may be calibrated to the wrong position by the attach

instruction executed in step 4 above, resulting is misalignment between its actual and desired positions. In this case, carry out tool zero position mastering and calibration manually.

When tool 2 is other than the above:
Tool 2 can be used as it is.

5 Tool 1 can be used as it is by executing the attach instruction to attach the tool.

32.13.5 The Attached Tool has been Detached by Mistake (without Using the Detach Instruction).

Symptom

The tool is not actually attached. So, a communications alarm such as "SRVO-068 DTERR alarm" occurs on the Pulsecoder. Correct this by Remedy 1 or Remedy 2 below according to the present conditions.

Remedy 1

Same as Remedy 1 of Subsection 32.13.1

Remedy 2

Same as Remedy 2 of Subsection 32.13.1

32.13.6 The Tool Axis of a Detached Tool has Moved.

Symptom

Battery-less tool (calibration motion types 1, 2):
Misalignment has occurred. Misalignment occurs as a result of calibration at the wrong position when the tool is attached with the tool axis having moved to a different position. Note, however, that misalignment does not occur when tool axis movement is within the allowable errors indicated below.
The allowable error must fall within the respective following ranges for the direct-acting axis and rotary axis:

Direct-acting axis: ±(gear ratio/2)mm or less around the detach position
Rotary axis: ±((360/gear ratio)/2)deg or less around the detach position

Other instances:
No problem has occurred.

Remedy

Battery-less tool (calibration motion types 1, 2):
1 Execute the attach instruction to attach the tool.
2 If misalignment occurs, carry out tool zero position mastering and calibration manually.

32.13.7 The Battery Voltage has Fallen.

Symptom

The "SRVO-065 BLAL" alarm has occurred with the tool in an attached state.

Remedy

1 Change the battery with the tool attached.
2 The tool can be used as it is.

32.13.8 The Battery Ran Low while the Tool was Detached.

Symptom

The "SRVO-062 BZAL" and "SRVO-075 Pulse not established" alarms occurred when the tool was attached.

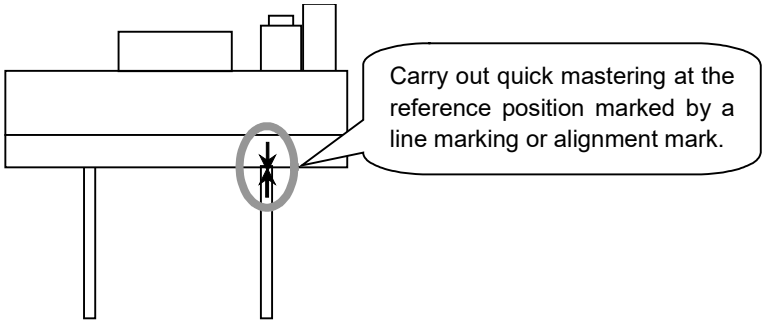
Remedy

When the reference position has not been set:

1 Press the F3:RES_PCA in the Calibration screen, and select YES.
2 Cycle power of the controller.
3 Move the tool by manual feed to cancel the "SRVO-075 Pulse not established" alarm.
4 Carry out tool zero position mastering and calibration manually.

When the reference position has been set:

1 Press the F3:RES_PCA in the Calibration screen, and select YES.
2 Cycle power of the controller.
3 Move the tool by manual feed to cancel the "SRVO-075 Pulse not established" alarm.
4 Move the tool by manual feed to the reference position marked by a line marking or alignment mark.
5 Execute “3 QUICK MASTER” in the Calibration screen.



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-83284EN) 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 6-7 Step test" and "Procedure 6-8 Continuous test (using the teach pendant)" of "FANUC Robot series OPERATOR's MANUAL(Basic Operation)"(B-83284EN) 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.

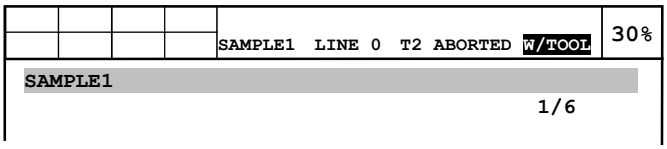
⚠WARNING
The robot starts its motion in the next step. If the jog feed of the robot needs to be stopped in an emergency in the order to avoid danger, the operator should release the deadman's switch or press the emergency stop button.

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 speed override (default value is 0) should be during high speed SHIFT operation to $HSP_SHIFT.$HSP_OVRD on system variable screen. It is possible to input 0-100 but 0 means disable high speed SHIFT function.

NOTE
Speed override is clamped to upper limit in case there is setting upper limit of speed override and value of $HSP_SHIFT.$HSP_OVRD is higher than this value. The speeds at the tool tip and flange surface are limited to 250 mm/sec or slower at T1 mode. So there is no situation that speed exceed 250 mm/sec by this function.

34.2 HIGH SPEED SHIFT JOG OPERATION

It is possible to change moving robot speed high during jog operation by pressing 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.
- Value of system variable $HSP_SHIFT.$HSP_OVRD is 1-100.
- Speed override isn’t either FINE or VFINE.

Step

1 Hold the teach pendant and press the deadman switch on the back of the teach pendant. Continue pressing the deadman switch during jog feed.
2 Turn on the teach pendant enable switch.
3 Release 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 value of system variable $HSP_SHIFT.$HSP_OVRD (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.
- Value of system variable $HSP_SHIFT.$HSP_OVRD 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 value of system variable $HSP_SHIFT.$HSP_OVRD (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.
Servo Diagnosis	Diagnose servo alarms to show possible causes of alarms.
Motion Profiler	Show motion data (OVC, overheat, power consumption) of each program.

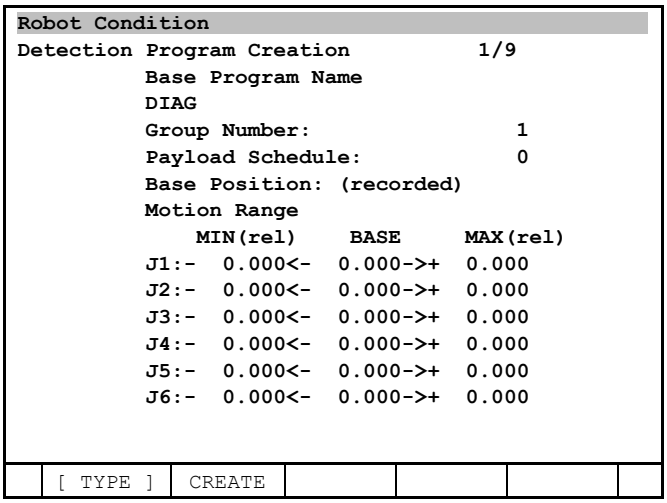
35.1 ROBOT CONDTION DETECTION

- This function detects abnormalities of reducers to help users decide time for preventive replacement in order to minimize down time due to a problem in reducers.
- By posting warning, it warns the users of abnormality in reducer, detected by periodically analyzing motion data.
- It requires creation of a diagnostic program, registration of initial data, and periodical execution of the diagnostic program.
- The diagnostic program is created by giving base position and allowable motion range.
- Some models do not support this function, contact your local FANUC representative about supported models.
- If enough motion range is not given, motion data cannot be analyzed.
- In case of impulsion-induced rapid degradation, or failure in certain parts, reducer failure may not be detected beforehand as abnormality.
- A failure due to aging may also progress rapidly in a few days before failure. Run the diagnostic program periodically, around once a day.

35.1.1 Setup

“CREATE”
This function requires creation of diagnostic program.
Create diagnostic program with following procedure.

1 Jog the robot to the base position, where diagnostic program starts and ends.
2 Press the [MENU] key.
3 Select “UTILITIES”.
4 Press the F1, [TYPE].
5 Select “Robot Condition”.

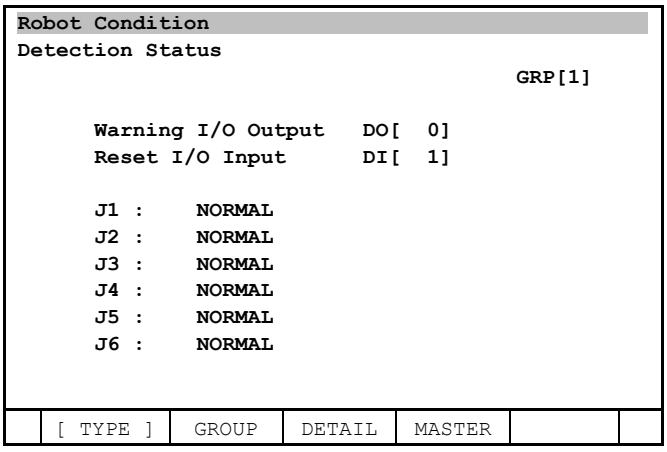


6 Enter to “Base Program Name” the name of the diagnostic program to be created.
Group number will be added to the base name to create program name.
Please make the base program name not more than 34 letters.
7 Enter to “Group Number” the number of the group for analysis.
8 Enter to "Payload Schedule" the number of the payload setting.
9 Move the cursor over “Base Position”, and press the F4, RECORD with [SHIFT] key.
10 Enter allowable motion range of each axis in degrees.
For each axis, enter allowable motion range relative to the base position, both MIN and MAX.
The robot will move axis by axis, one axis at a time.
The axis without wide enough motion range cannot be analyzed.
The axis that is set motion range of 0 degree will not be analyzed.
11 Set to AUTO mode and disable TP.
12 Set override to 100%.
13 Press the F2, CREATE with [SHIFT] key.

A program will be created with group number added to the Base Program Name.

NOTE
- Although the robot does not move during creation, the robot moving is simulated inside the controller. Please reset alarms and release hold etc. to enable to simulate the robot moving with 100% override. And, then create a diagnostic program.
- Please keep 100% override during creation.
- After the creation, please run the program with low override to assure safety of the program.

“MASTER”
Next, follow the procedure for the registration of the initial data.
1 Run the created program with 100% override (run with low override beforehand to assure safety).
2 Press the [MENU] key.
3 Select “STATUS”.
4 Press the F1, [TYPE].
5 Select “Robot Condition”.



6 If upper right display of 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 an individual program is used for each axis, input the each target axis number of the program executed this time in order to MASTER.
9 A message will ask to confirm removal of previous data and use of current data as base data.
"Please confirm to use current data as baseline data? Please note previous history will be removed after master."
Select [YES].

The latest analysis data is registered as base line data.
The axis not registered, shown with *s was not given enough motion range to analyze the condition.

NOTE

Please note the followings, for they may prevent the function from working properly.

- Do not edit the created diagnostic program.
- Creating the program with the same name will overwrite the old program. If overwritten after "MASTER," the function does not work properly. If the program needs to be changed during setup, go through both "CREATE" and "MASTER".
- Different programs may be created with arbitrary base positions. However, an axis can be analyzed with only one program. With different programs analyzing the same axis, the function does not work properly because of different characteristics of the data collected. When tried to analyze an axis with a program different from the one used for "MASTER," following warning will be posted.
"DIAG-008 Baseline prog changed (G:g A:a)"
where g is the group number and a is the axis number.
- Basically, go through "MASTER" only with initial setup and when replacing reducer.
- If it becomes impossible to use the program for some reason, such as moving the system or changing the robot's load, it would be necessary to go through "CREATE" and "MASTER". However, if there is a reducer already reaching an end of its operating life, the function may not work properly.

35.1.2 Execute Program

After the base line data is registered, the latest data is analyzed every time diagnostic program is executed with 100% override, and check if there are any abnormalities.
Run the program periodically to detect abnormality: for example, once a day.
Make sure that the program is run under the same condition as the time the baseline data was recorded, such as load.
In the case that the software version is 7DC1/01-18 or 7DC2/01-04, make sure that the UTOOL number is set to1 before executing the program.

If abnormality is detected, following warning will be posted,
"SRVO-394 Reducer Abnormal (Gg, Aa)"
where g is the group number and a is the axis number.
Abnormality is detected in corresponding axis. Please plan replacement.

35.1.3 Robot Condition Detection Status

Current condition can be viewed at Robot Condition Detection Status.
To view Robot Condition Detection Status,
1 Press the [MENU] key.
2 Select "STATUS".
3 Press the F1, [TYPE] key.
4 Select "Robot Condition".

Robot Condition Detection Status shows condition of each axis.
Moving the cursor over an axis and pressing F3, DETAIL will show the time of registration of the base line data and time of analysis of the latest data.

If "Warning I/O Output" DO is set, the specified DO turns ON when abnormality is detected.
If "Reset I/O Input" DI is set, the "Warning I/O Output" DO is cleared (i.e. turns OFF) when the specified DI turns ON.

35.1.4 After Replacement

When reducer is replaced, corresponding base line data need to be re-registered.
Follow the procedure to re-register the data.
1 Run the diagnostic program with 100% override. (Run with low override beforehand to assure safety.)
2 Press the [MENU] key.
3 Select "STATUS".
4 Press the F1, [TYPE].
5 Select "Robot Condition".
6 If upper right display of GRP[] does not show the group for analysis, press the F2, GROUP and enter the group number.
7 Move the cursor over the axis of replacement, and press the F3, DETAIL.
8 Check that the time of the Latest is the time diagnostic program was run in step 1.
9 Press the F4 MASTER.
10 Enter the axis of replacement when prompted to enter the axis number.
11 A message will ask to confirm removal of previous data and use of current data as base data.
"Please confirm to use current data as baseline data? Please note previous history will be removed after master."
Select [YES].
12 Press the F3, DETAIL and check that time of the Base is updated.

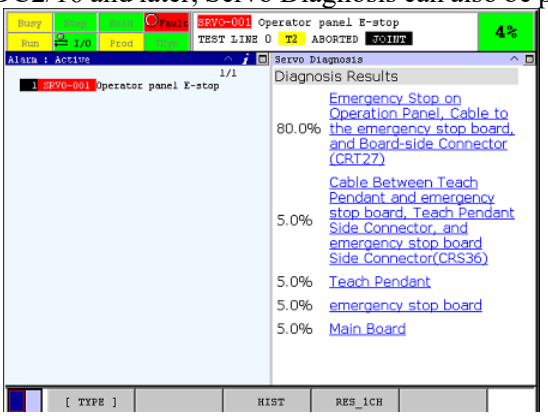
The new base line data is registered.

35.2 SERVO DIAGNOSIS

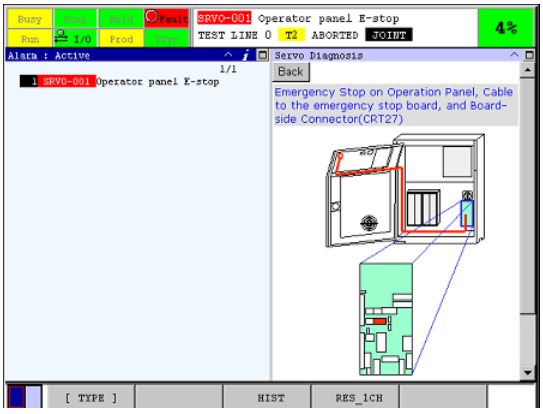
- This function diagnoses alarms posted, and list probable causes, aiming to decrease the down time before resetting alarm.
  - ➢ This function estimates where the problem is, based on the alarm posted and internal information acquirable by software. However, it may not be able to estimate the problem in some cases such as if problem rises where software cannot acquire information. Please use it to help solve problem when alarm is posted.
- When servo alarm is posted, follow the procedure to perform Servo Diagnosis.

(Alarm Posted.)

1 Press the [MENU] key.
2 Select “ALARM”. “Alarm: Active” would be displayed.
3 Select an alarm to diagnose.
4 Press the [FCTN] key with [i] key, then select "Servo Diagnosis" in the "Related Views" menu. The causes of the alarm would be listed. Please check the items to see if there is any trouble.
With software versions 7DC2/16 and later, Servo Diagnosis can also be performed by F5 “DIAG” key.



5 Some of the causes have the texts as hyperlinks. Tap the hyperlink to show the diagram about the cause. (7DC2 and later)



Click “Back” button to return back to the cause list view.

- The alarms covered by this function are following:
  - ➢ SRVO-001 Operator panel E-Stop
  - ➢ SRVO-005 Robot overtravel
  - ➢ SRVO-006 Hand broken
  - ➢ SRVO-009 Pneumatic pressure alarm
  - ➢ SRVO-018 Brake abnormal (G:%d A:%d)
  - ➢ SRVO-021 SRDY off (Group:%d Axis:%d)
  - ➢ SRVO-022 SRDY on (Group:%d Axis:%d)
  - ➢ SRVO-023 Stop error excess (G:%d A:%d)

- SRVO-024 Move error excess (G:%d A:%d)
- SRVO-036 Inpos time over (G:%d A:%d)
- SRVO-038 Pulse mismatch (G:%d A:%d)
- SRVO-043 DCAL alarm (Group:%d Axis:%d)
- SRVO-044 DCHVAL alarm (Group:%d Axis:%d)
- SRVO-045 HCAL alarm(Group:%d Axis:%d)
- SRVO-046 OVC alarm (Group:%d Axis:%d)
- SRVO-047 LVAL alarm(Group:%d Axis:%d)
- SRVO-049 OHAL1 alarm (Grp:%d Ax:%d)
- SRVO-050 Collision Detect alarm (G:%d A:%d)
- SRVO-051 CUER alarm(Group:%d Axis:%d)
- SRVO-055 FSSB com error 1 (G:%d A:%d)
- SRVO-056 FSSB com error 2 (G:%d A:%d)
- SRVO-057 FSSB disconnect (G:%d A:%d)
- SRVO-062 BZAL alarm(Group:%d Axis:%d)
- SRVO-064 PHAL alarm(Group:%d Axis:%d)
- SRVO-067 OHAL2 alarm (Grp:%d Ax:%d)
- SRVO-068 DTERR alarm (Grp:%d Ax:%d)
- SRVO-069 CRCERR alarm (Grp:%d Ax:%d)
- SRVO-070 STBERR alarm (Grp:%d Ax:%d)
- SRVO-071 SPHAL alarm (Grp:%d Ax:%d)
- SRVO-072 PMAL alarm(Group:%d Axis:%d)
- SRVO-073 CMAL alarm(Group:%d Axis:%d)
- SRVO-074 LDAL alarm(Group:%d Axis:%d)
- SRVO-076 Tip Stick Detection(G:%d A:%d)
- SRVO-105 Door open or E.Stop
- SRVO-136 DCLVAL alarm (G:%d A:%d)
- SRVO-156 IPMAL alarm (G:%d A:%d)
- SRVO-157 CHGAL alarm (G:%d A:%d)
- SRVO-204 External(SVEMG abnormal) E-stop
- SRVO-205 Fence open(SVEMG abnormal)
- SRVO-206 Deadman switch (SVEMG abnormal)
- SRVO-213 E-STOP Board FUSE2 blown
- SRVO-214 6ch amplifier fuse blown(R:%d)
- SRVO-215 Brake Unit fuse blown(R:%d)
- SRVO-216 OVC(total) (%d)
- SRVO-233 TP OFF in T1,T2/Door open
- SRVO-235 Short term Chain abnormal
- SRVO-251 DB relay abnormal(G:%d A:%d)
- SRVO-252 Current detect abnl(G:%d A:%d)
- SRVO-253 Amp internal over heat(G:%d A:%d)
- SRVO-277 Panel E-stop(SVEMG abnormal)
- SRVO-278 TP E-stop(SVEMG abnormal)
- SRVO-290 Dclink HC alarm(G:%d A:%d)
- SRVO-291 IPM over heat (G:%d A:%d)
- SRVO-292 EXT.FAN alarm (G:%d A:%d)
- SRVO-349 DCS MCC ON alarm %x,%x

35.3 MOTION PROFILER

Motion Profiler is a motion option providing important motion information for either total execution cycle or each executed program line. This option enables the easy diagnosis of program performance for the selected program. The exact motion lines can be identified and then adjusted easily using this option so the desired performance can be achieved.

The summary data for execution cycle provided by this option includes:

- Total Cycle Time
- Reducer Load Monitor (max. value)
- Joint Load Monitor (intended models only, max. value)
- M430 Load Monitor (intended models only, if warning is posted)
- Reducer Estimated Life (L10 life)
- Power Consumption
- Regenerative Power (if hardware is available)
- Steady-State OVC (the prediction, if continuously run in the same path, non-stop)
- Overheat

The detailed data shown in bar charts per program execution line includes:

- Execution Time per line
- Max. value of Reducer Load Monitor per line
- Power usage per line

Due to the memory constraints of controllers, the motion profiler data for the extremely long program may be overwritten and data may be missing. Please consider separating the very long program to reasonable program sizes.

35.3.1 Setup

Set up Motion Profiler with following procedure.

1 Press the [MENU] key.
2 Select “STATUS”.
3 Press the F1, [TYPE].
4 Select “Motion Profiler”.

With software prior to 7DC1/06, please set up Motion Profiler with following procedure.

1 Press the [MENU] key.
2 Select “SETUP”.
3 Press the F1, [TYPE].
4 Select “Motion Profiler”.

Motion profiler function will be turned off automatically after the selected program is aborted. Please enable the function again if the program needs to be analyzed again.

It is important that ambient temperature, special hardware option, and robot connection cable (RCC) length for each robot group are entered correctly. The prediction and calculation result will be affected by these settings. The results will not reflect those of non-supported groups.

Table 35.3.1 Menu items and descriptions for motion profiler's setup menu

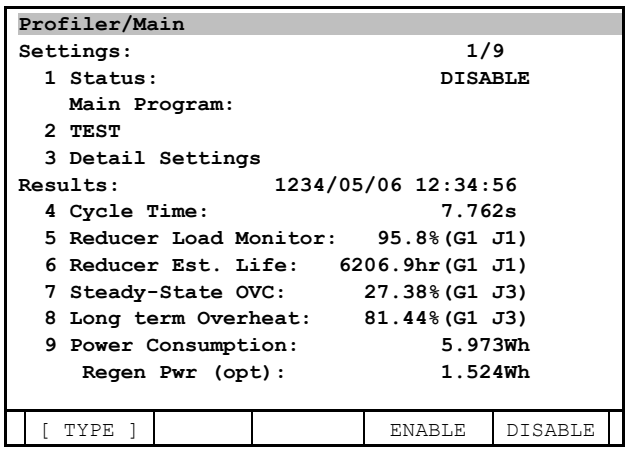
Item			Description
Status			Enable or disable Motion Profiler. This status will change to disable automatically after the main program is aborted.
Main Program			The main TP program to be used for Motion Profiler
Detail Settings	Group Settings	Group Number	Group number
		Ambient Temperature	The ambient temperature for this robot. With software versions 7DC3/34 and later or 7DF1 and later, please set the temperature from System Config Menu.
		Robot Connection Cable Length	The length of robot connection cable (RCC) for this robot This number may be printed on the robot connection cable.
		Hardware Configuration	Any special hardware configuration for this robot. For example, motor fan, motor cover, etc.
	Data Storage	Device	Device for comparison data to output to and read from. With software versions 7DF1/08 and later, comparison data can be output and compared with.
		Directory	Directory for comparison data to output to and read from. If not set, root directory will be used.

35.3.2 Results

The results of Motion Profiler function can be reviewed in either Status menu or directly go to the "related view" from TP editor.

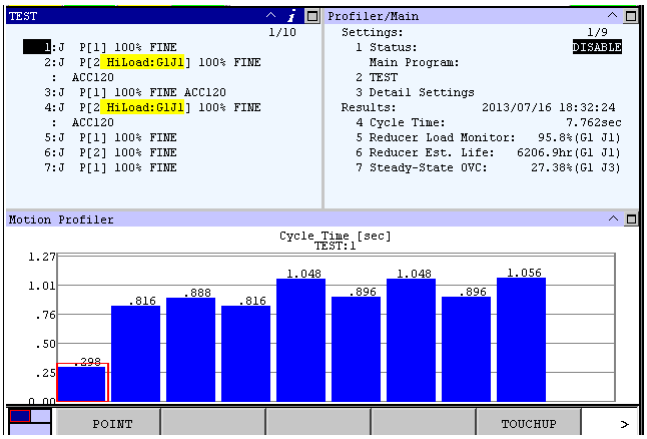
- Status screen

1 Press the [MENU] key.
2 Select "STATUS".
3 Press the F1, [TYPE].
4 Select "Motion Profiler" to show Main page.
5 Select data and press the F3, DETAIL to display the detail bar chart with program name: line number as horizontal axis, or data of each axis.



- Related view

1 Press the [SELECT] key.
2 Select TP program that Motion Profiler was enabled and press the [ENTER] key.
3 In TP program editor, press the [i] + the [FCTN] key to display Related View menu.
4 Select "Motion Profiler" to review triple pane with TP editor, Motion Profiler Main, and Motion Profiler detailed chart. Scrolling the cursor in the TP editor then Motion Profiler detailed chart will update to current line data automatically. This will work in the sub program as well.



Unsupported axes or data:

If there are unsupported axes or data, next to the “Results:” will be displayed (exclusions). Selecting (exclusions) will show the list of unsupported axes and/or data. (software versions 7DC1/09 and later)

Comparison Data Output:

With software versions 7DF1/08 and later, Motion Profiler data can be output to a file for comparison.

1. Navigate to the Main page of the Profiler described above.
2. Press F2 OUTPUT to save the file containing the Motion Profiler data.
   A subdirectory named after the Main Program will be created, if have not been, under the directory of the device specified in the Detail Settings, and a new file will be saved under it.
   When the operation is successful, a prompt will display the name of the file output.
   e.g. “Output File: ~2018_07_08T12_34_56.dat”
   If a file with the same name already exists, it will be overwritten.
   When the operation is not successful, following prompts may be displayed:
   - “Comparison data path is uninitialized”
     The device to output the data is not specified. Please specify in the Detail Settings.
   - “Subdirectory is too long”
     Due to the file system constraints, the directory names must be limited to 40 characters.
   - “Directory does not exist”
     The directory specified in the Detail Settings must exist on the device. Please create the directory and try again.
   - “Invalid device”
     The device specified in the Detail Settings is not accessible. Please make sure that the device is correctly specified and inserted in the controller.

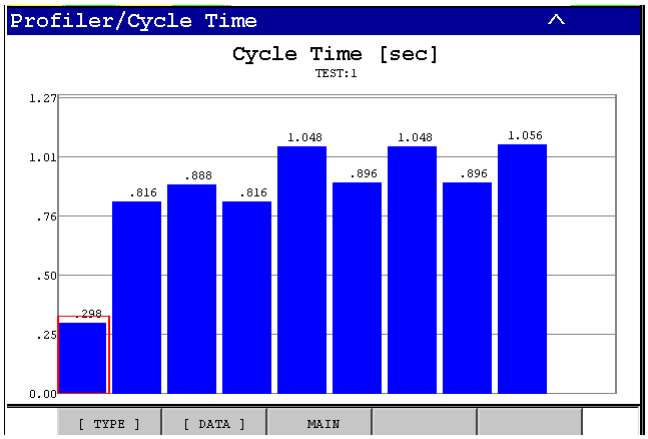
NOTE
Please do not rename the subdirectory with the Main Program name or any of the data files created by this function, or those files may not be used for comparison.

35.3.2.1 Detail

Main page displays summarized data of analyzed program. With cursor over each item, pressing the F3, DETAIL shows detailed data of each item.

[Cycle Time]

Shows execution time of the analyzed program. In DETAIL, bar chart shows execution time of each program line. The lines with very short execution time will not be shown, including some of command lines.

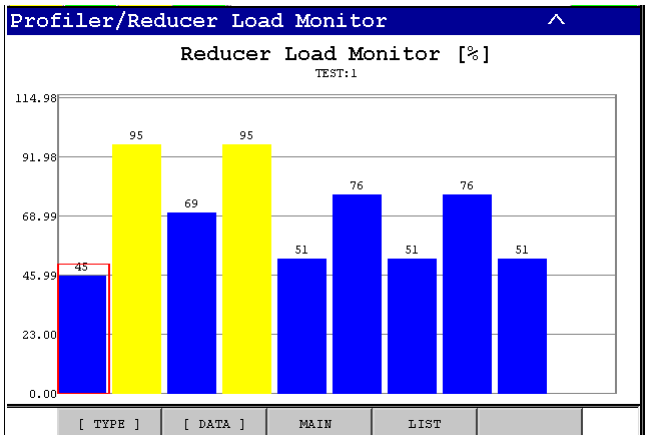


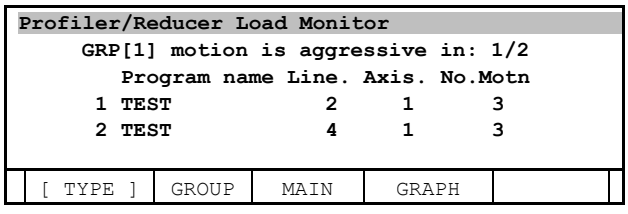
[Reducer Load Monitor] (shown only for supported models)

Calculates load level to drive element, such as reducer or gear, from motion command. Continued use over 100% of load level may result in premature failure of driving element, such as reducer or gear. If the load level exceeds 100%, change the program, such as acceleration override instruction (ACC) or feed rate, so that the load level is below 100%.

Main page displays maximum value within the analyzed program, and the group and axis. In DETAIL, bar chart shows maximum values per line. Lines with aggressive motion will be colored yellow and red. The yellow bar shows that the load level is near the limit, and the red bar shows that the load level exceeds the limit. In LIST, it shows list of aggressive motions with program name, line number, axis number, and number of motion. If motion command of aggressive motion is derived from more than one line, number of motion shows number of lines involved. After the analysis, if there are aggressive motions, corresponding line will have comment “HiLoad:G(group number)” added when EDIT is displayed with TP enabled.

When load level is over 100%, “DIAG-005 Reducer load excess” warning would be posted in addition to the display in this option.





[Joint Load Monitor] (For certain models only. Shown only for supported models)

Displays load level to joints. If it exceeds 100%, “MOTN-522 Joint load excess” alarm will stop the robot. There is no DETAIL.

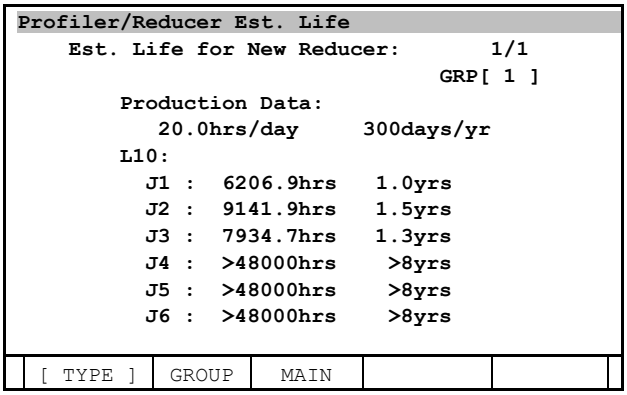
[M430 Load Monitor] (For certain models only. Shown only for supported models)

Calculates load to reducers in different way from Reducer Load Monitor. If “warning” is displayed, it means that “MOTN-520 J(number of axis)TqOver” is posted. Change the program in the same way as Reducer Load Monitor so that the warning would not be posted. There is no DETAIL.

[Reducer Estimated Life] (Some axes are not targeted. Shown only for supported models)

Displays L10 life of reducers if analyzed program is executed repeatedly using new reducers. Main page shows the shortest L10 life, its group and axis. In DETAIL, L10 life of each axis is shown. Also in DETAIL, production data, operating hours per day and operating days per year, which would be used for L10 life calculation, ca be changed.

L10 life is the time after which 10% of reducers fail. Evaluate the L10 life in the context of intended use.



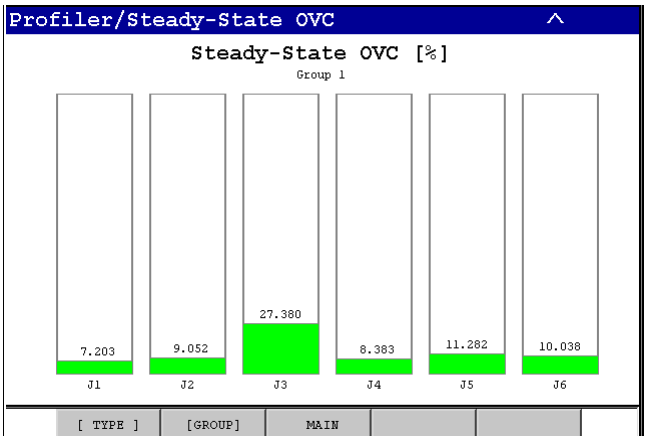
NOTE

Reducer Load Monitor, Joint Load Monitor, M430 Load Monitor, and Reducer Estimated Life require that the payload is set correctly. Otherwise, they do not work properly.

Also, the calculations are done with the assumption that there is no contact between robot and other objects. Therefore, a program that includes contact and causes external force (such as pushing a work piece while mounting) will not be analyzed properly.

[Steady-State OVC]

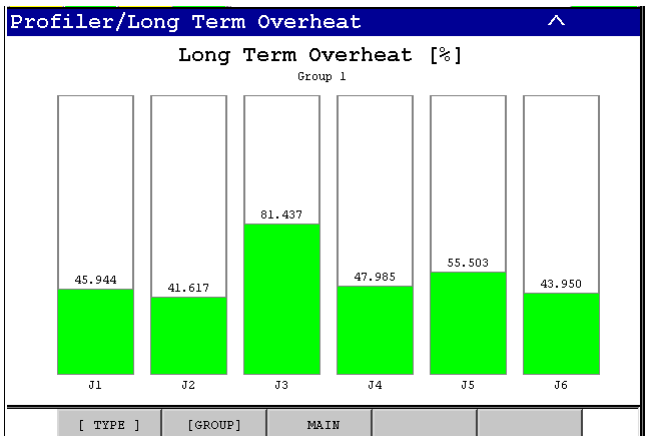
Shows estimation of OVC value(The ratio of the motor temperature simulated by the software to the alarm threshold) when analyzed program was run repeatedly. In Main, the largest value among the axes is shown. In DETAIL, the values of each axis are shown. The value over 100% shows estimated result of "SRVO-046 OVC alarm (Group:i Axis:j)" occurring.



[Overheat] (for supported models)

Actual overheat alarm(SRVO-067 OHAL2 alarm (Grp:i Ax:j)) is triggered by sensor in the motor, but this function estimates if overheat alarm would occur, by comparing electric current data to

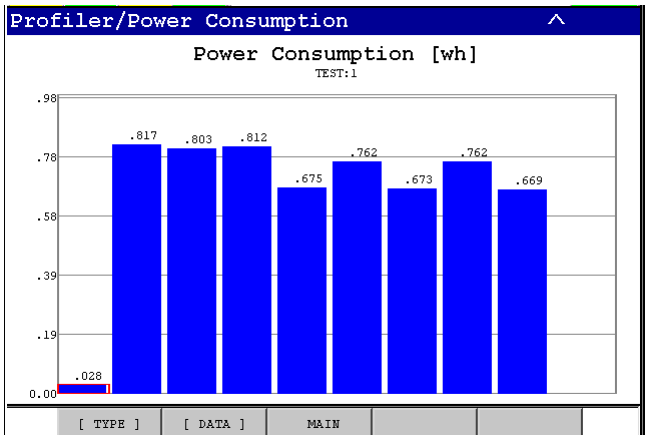
experimental data. In Main, the largest value among the axes is shown. In DETAIL, the values of each axis are shown. The values represent the ratio of calculated value against the thresholds. The value over 100% shows estimated result of overheat alarm occurring. The estimation is affected by Ambient Temperature and Hardware Configuration set in Detail Settings. Please set them accurately.



NOTE
Steady-State OVC and Overheat are results of estimation, and their accuracies are not guaranteed. Please use them only as guides.

[Power Consumption] (for supported models)

Shows power consumption while executing the analyzed program. In DETAIL, power consumption per axis is shown. They are estimated results and not measured data. In cases such as unsupported group included in motion groups of the program, the results do not include their power consumption. The results are affected by Robot Connection Cable Length set in Detail Settings. Please set it accurately.



[Regenerative Power]

Shows the estimation of regenerated energy while executing the analyzed program, if the hardware option is installed. The result is affected by Robot Connection Cable Length set in Detail Settings. Please set it accurately. There is no DETAIL.

35.3.2.2 Details Comparison

With software versions 7DF1/08 and later, the Cycle Time and the Power Consumption data may be compared against a previously saved data.

1. In the DETAIL bar charts of the Cycle Time or the Power Consumption, press F4 COMPARE.

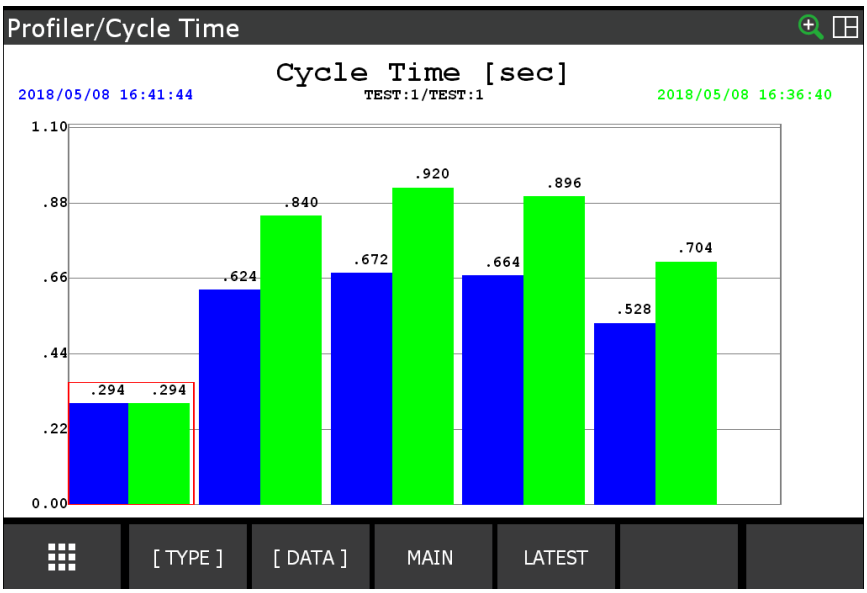
The list of the files in the directory, named after the Main Program of the last recorded data, under the directory of the device specified in the Detail Settings will be brought up.
They will be listed by the date the data was recorded.
Choose a file to compare to.

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 the slash "/".

NOTE
If the program has been edited to add/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 the number of difference. In Related View display, selecting line from the TPP Editor will only allow selection up to the end of the current program.

2. To stop comparing, press F4 LATEST

NOTE
If there are many files in the specified directory, it may take some time to show the file list, during which the Teach Pendant will appear non-responsive. Please consider using different subdirectories if the number of files grows too large.



36 MENU UTILITY FUNCTION

You can create your own teach pendant menus to help you application run more smoothly. To use your own menus, first perform menu setup then call the menu macro program from a teach pendant program and select the desired menu in the parameter.

See Fig. 36 (a)for an example program and example menus.

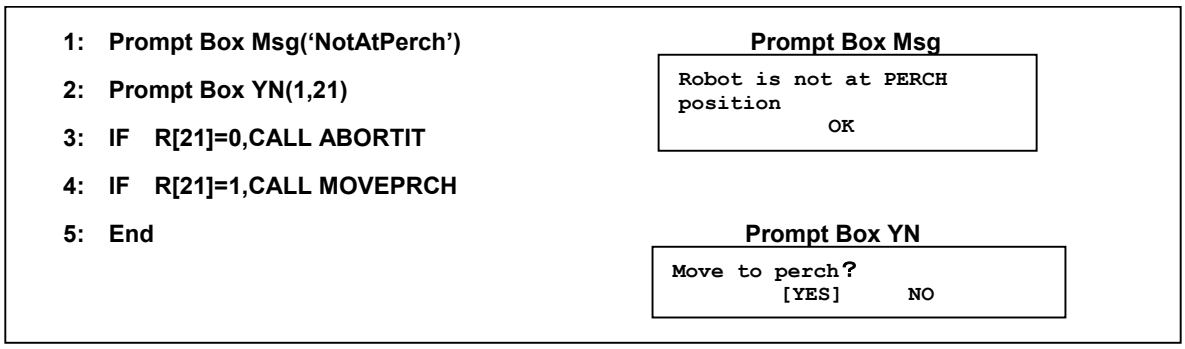
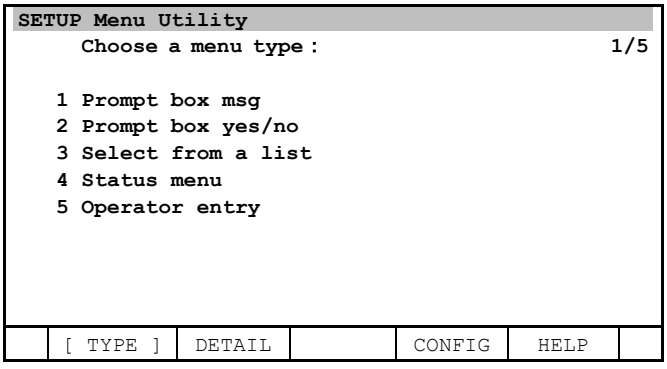


Fig.36 (a) Example for teach pendant program that uses menu utility function

Above example is in case position of robot isn't in safe area. If macro program in line 1 "Prompt box Msg('NotAtPerch')" is implemented, prompt box message screen is displayed so that to make sure the operator is aware of something. And if macro program in line 2 "Prompt Box YN(1.21)" is implemented, prompt box yes/no screen is displayed, so that the teach pendant program pauses until the operator selects YES or NO and presses the [ENTER] key. In this case, teach pendant program "MOVEPRCH" is called by line 3 is implemented if YES is selected and teach pendant program "ABORTIT" is called by line 4 is implemented if NO is selected. In this way prompt box message or menu is displayed so that it is possible to prompt the operator to be caution or let the operator do work smoothly.

36.1 SETUP ABOUT MENU UTILITY

If you press the [MENU] key and select "6 SETUP" and select "Menu Utility", the following menu utility display will be displayed. There are five menu type "Prompt box msg", "Prompt box yes/no", "Select from a list", "Status menus" and "Operator entry".



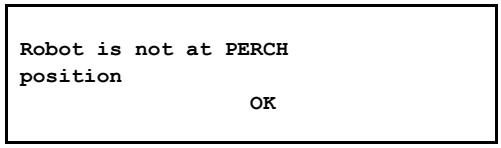
Each menu type corresponds with macro program as following Table 36.1, so each behavior of macro program will be changed if each setting of menu type is changed.

Table 36.1 Description for each menu utility screen item

ITEM	Macro Program	DESCRIPTION
Prompt box msg	Prompt Box Msg (Menu)	This item is used to make sure the operator is aware of something while executing a TP program. When a prompt box message screen is displayed, the teach pendant program pauses until the operator pressed the [ENTER] key.
Prompt box yes/no	Prompt Box YN (menu,answer_reg)	This item asks the operator a question that can be answered YES or NO while executing a TP program. When a prompt box yes/no screen is displayed, the teach pendant program pauses until the operator selects YES or NO and presses the [ENTER] key.
Select from a list	List Menu(menu,answer_reg)	This item asks the operator to select an item from a list of items while executing a TP program. When a list menu is displayed, the teach pendant program pauses until the operator selects an item form the list.
Status menus	Status Menu(menu)	This item displays status information to the teach pendant screen while executing a TP program. When a status menu is executed, the teach pendant displays the status menu and then continues executing. The values that are displayed are refreshed at least once every second.
Operator entry	Op. Entry Menu(menu)	This item requires the operator to enter INTEGER,REAL,BOOLEAN, or text values while executing a TP program. When an operator entry menu is executed, the program pauses until the operator types the information and then presses the F2 DONE.

36.1.1 Prompt Box Msg

A prompt box message menu is used to make sure the operator is aware of something. When a prompt box message screen is displayed, the teach pendant program pauses until the operator presses the [ENTER] key. In addition, when the prompt box is displayed, the operator cannot access any other menus.



Macro: Prompt Box Msg (menu)
A prompt box message menu uses the Prompt Box Msg(menu) macro.

Parameter 1: menu
This is the menu number or menu name. The menu can be chosen by number or name.

Usage example:
Example 1 (setting menu number to parameter):
PROG TEST1
1: Prompt Box Msg(2)

Example 2 (setting menu name to parameter):
PROG TEST2
1: Prompt Box Msg('NotAtPerch')

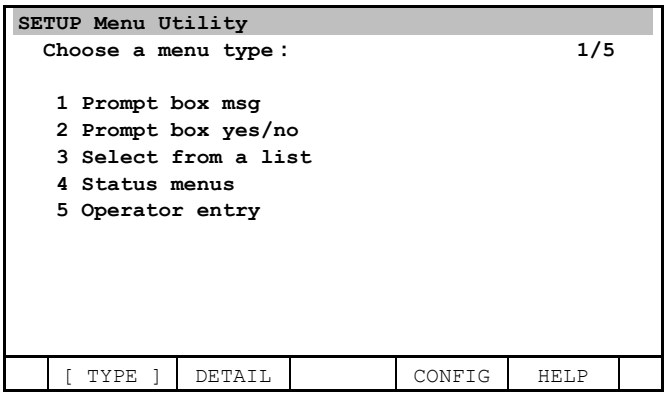
Save Menu setting:

Each prompt box message menu is stored in its own variable file that begins with the prefix “MENU1”. The next two characters of the filename are the menu number. For example, the prompt box message menu 1 is stored in the variable file named “MENU101.VR”.

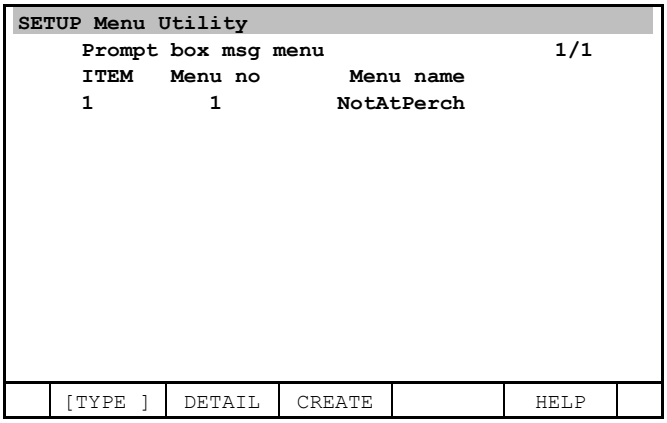
Procedure 36-1 Setup Prompt Box Msg

Step

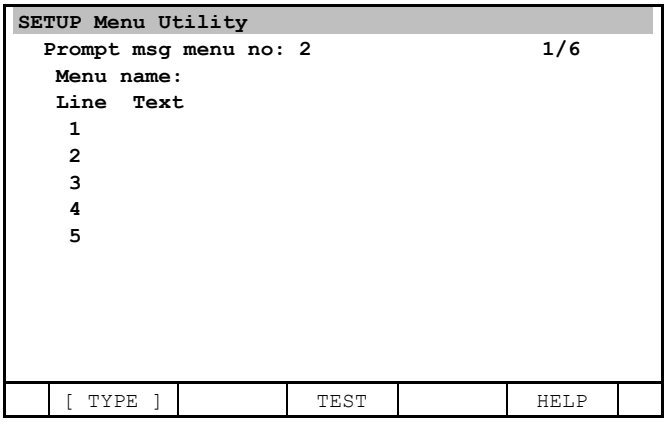
1 Press the [MENU] key and select ”6 SETUP” and select ”Menu Utility” so that Menu utility screen will be displayed.



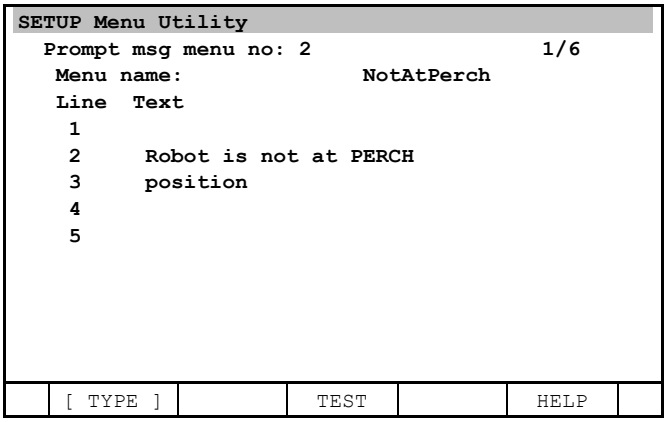
2 Set cursor to “1 Prompt box msg” and press the F2, DETAIL so that Prompt box msg menu will be displayed. You will see a screen similar to the following.



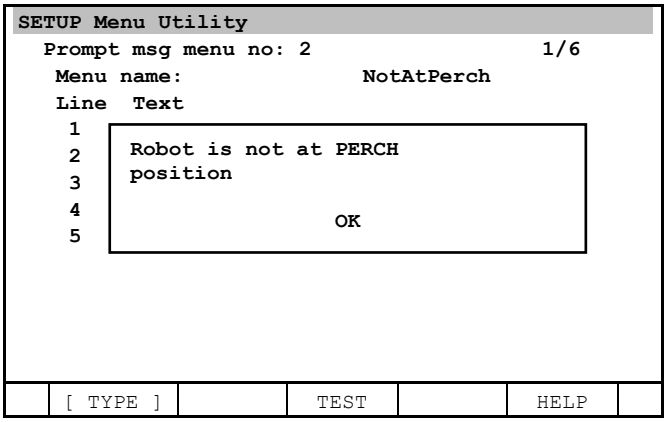
3 Press the F3, CREATE if you want to create new menu. Press the F2, DETAIL if you want to change menu that already exists. The following screen will be displayed.



4 Input menu name and message that you want to display. (ex. Setting menu no.2 as menu name "NotAtPerch")

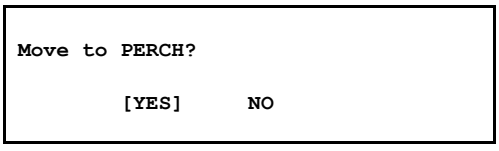


5 Press the F3, TEST if you want to confirm the message that will be displayed. The following screen will be displayed.



36.1.2 Prompt Box Yes/No Menu

A prompt box yes/no menu asks the operator a question that can be answered YES or NO. When a prompt box yes/no screen is displayed, the teach pendant program pauses until the operator selects YES or NO and presses the [ENTER] key. In addition, when the prompt box yes/no screen is displayed, the operator cannot access any other menus.



Macro: Prompt Box YN (menu, answer_reg)
A prompt box yes/no menu uses the Prompt Box YN(menu, answer_reg) macro.

Parameter 1: menu
This is the menu number or menu name. The menu can be chosen by number or name.

Parameter 2: answer_reg
This is the register that will contain the answer from the operator. When prompt box yes/no menu displays, 1 will be input to assigned register if you select YES, 0 will be input if you select NO.

Save Menu setting:
Each prompt box yes/no menu is stored in its own variable file that begins with the prefix “MENU2”. The next two characters of the filename are the menu number. For example, the prompt box message menu 1 is stored in the variable file named “MENU201.VR”.

Usage example:
Example 1 (setting menu number to parameter):

```
PROG   TEST1
   1: Prompt Box YN(2, 2)
   2: IF R[2] = 0, CALL ABORTIT
```

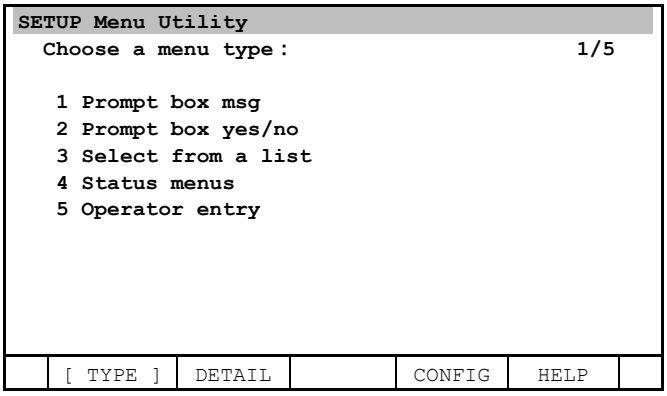
Example 2 (setting menu name to parameter):

```
PROG   TEST2
   1: Prompt Box YN('IsItSafe', 6)
   2: IF R[6] = 0, JMP LBL[5]
```

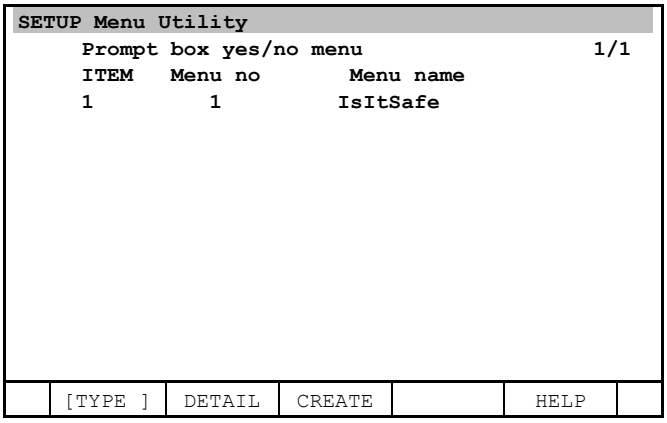
Procedure 36-2 Setup Prompt Box YN

Step

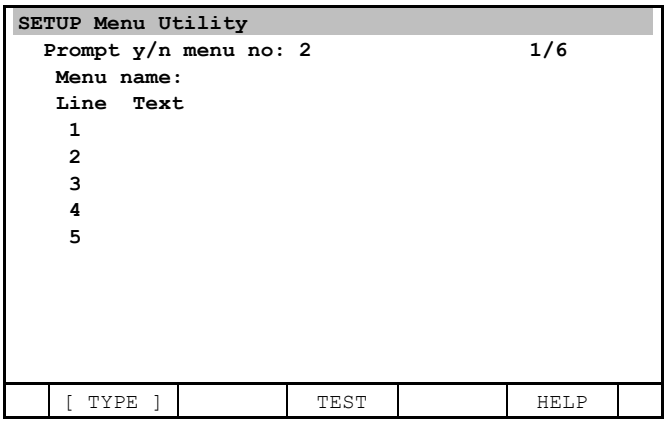
1 Press the [MENU] key and select ”6 SETUP” and select “Menu Utility” so that Menu utility screen will be displayed.



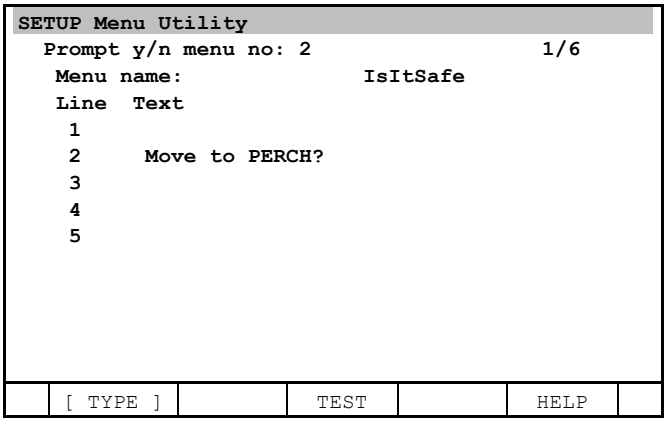
2 Set cursor to “2 Prompt box yes/no” and press the F2, DETAIL so that Prompt box yen/no menu will be displayed. The following screen will be displayed.



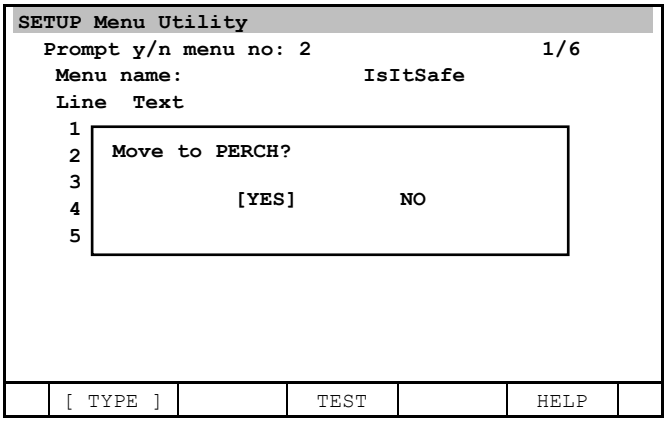
3 Press the F3, CREATE if you want to create new menu. Press the F2, DETAIL if you want to change menu that already exists. The following screen will be displayed.



4 Input menu name and message that you want to display. (ex. Setting menu no.2 as menu name "IsItSafe")

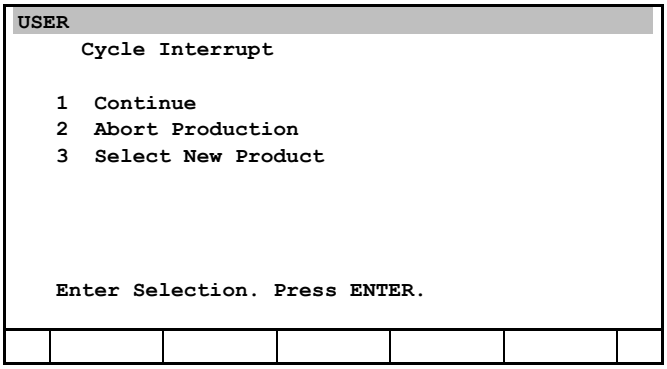


5 Press the F3, TEST if you want to confirm the message that will be displayed. The following screen will be displayed.



36.1.3 List Menu

A list menu asks the operator to select an item from a list of items. When a list menu is displayed, the teach pendant program pauses until the operator selects an item from the list.



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.

Save Menu setting:
Each list menu is stored in its own variable file that begins with the prefix “MENU3”. The next two characters of the filename are the menu number. For example, the list menu 1 is stored in the variable file named “MENU301.VR”.

Usage example:
Example 1:

```
 PROG   TEST1
      1: List Menu (2,21)
      2: SELECT R[21] = 1, CALL TOOL1
      3:              = 2, CALL TOOL2
      4:              = 3, JMP LBL[2]
      5:              ELSE, CALL ABORTIT
```

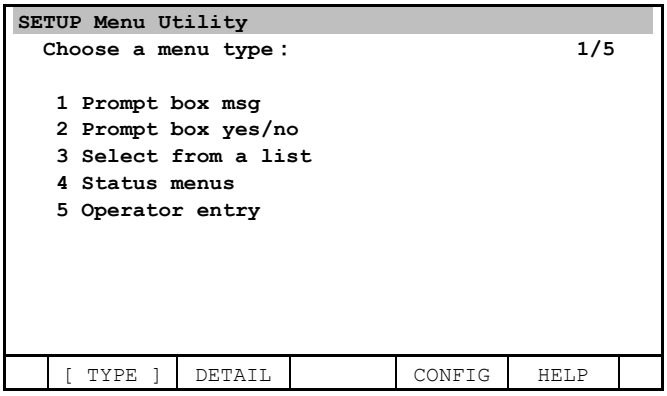
Example 2:

```
 PROG   TEST2
      1: List Menu (‘ErrorRecov’,2)
      2: SELECT R[2] = 1, JMP LBL[2]
      3:             = 2, CALL MOVEMANT
      4:             = 3, CALL MOVEPRCH
      5:             ELSE, CALL ABORTIT
```

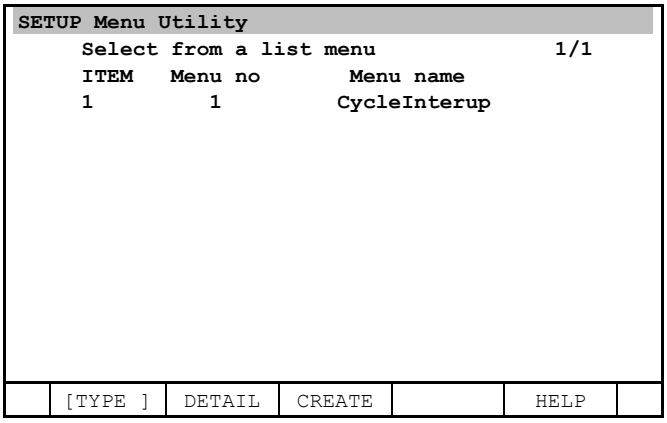
Procedure 36-3 Setup List Menu

Step

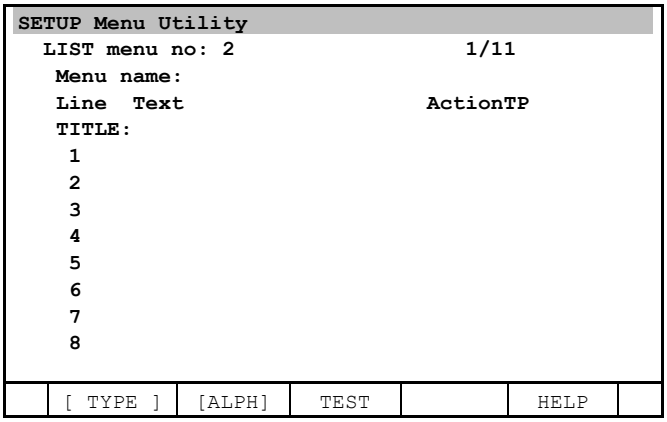
1 Press the [MENU] key and select “6 SETUP” and select ”Menu Utility” so that Menu utility screen will be displayed.



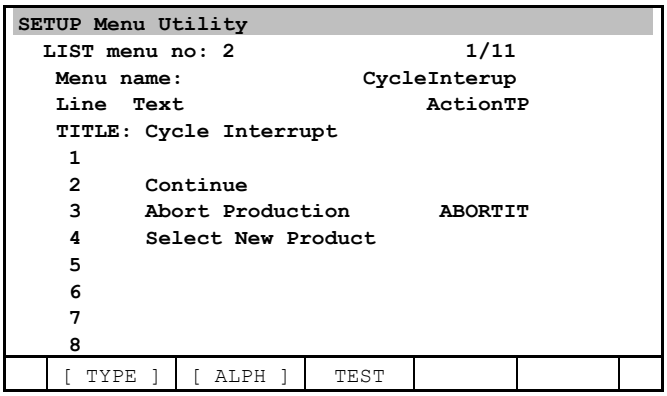
2 Set cursor to “3 Select from a list” and press the F2, DETAIL so that Prompt box will be displayed. The following screen will be displayed.



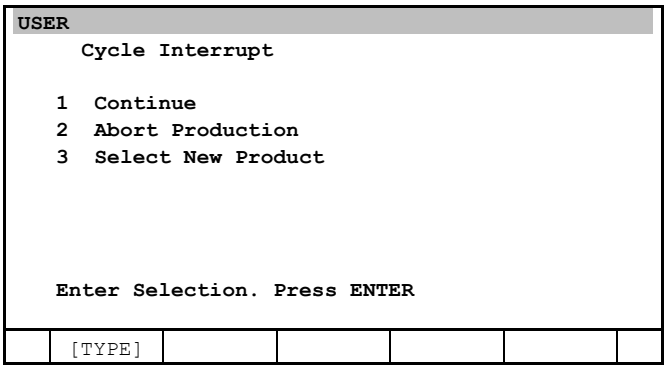
3 Press the F3, CREATE if you want to create new menu. Press the F2, DETAIL if you want to change menu that already exists. The following screen will be displayed.



4 Input menu name to line “Menu name” and text to each line. Line that is input text becomes item when list menu is displayed. Set cursor to Action TP and press the F4, CHOICE and select teach pendant program if you want to associate an Action TP program with a list item.

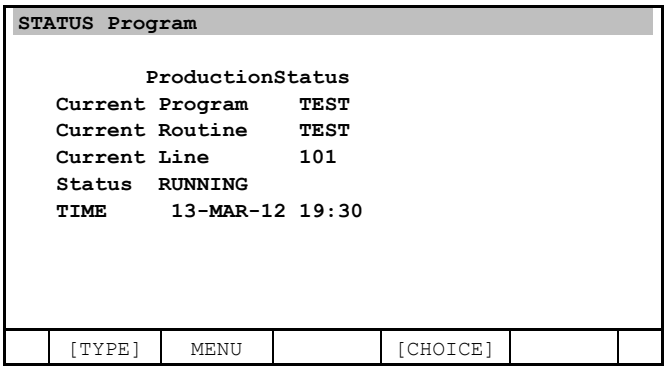


5 Press the F3, TEST if you want to confirm edited menu list. The following screen will be displayed.



36.1.4 Status Menu

A status menu displays status information to the teach pendant screen. When a status menu is executed, the teach pendant displays the status menu and then continues executing. The values that are displayed are refreshed at least once every second. This is read-only menu.



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.

Save Menu setting:
Each list menu is stored in its own variable file that begins with the prefix “MENU4”. The next two characters of the filename are the menu number. For example, the list menu 1 is stored in the variable file named “MENU401.VR”.

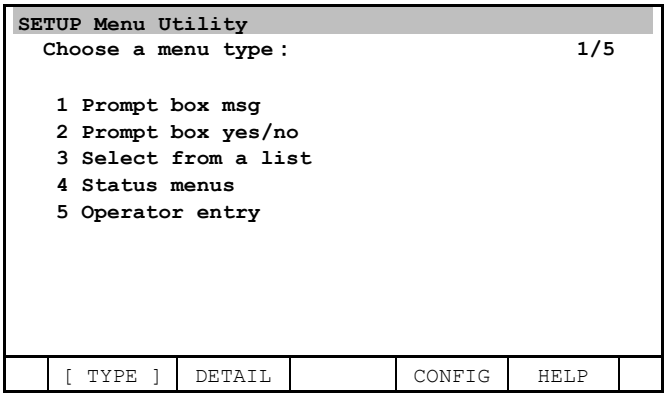
Usage example:
Example 1:
PROG TEST1
1: Status Menu (2)

Example 2:
PROG TEST2
1: Status Menu (‘ProductionStatus’)

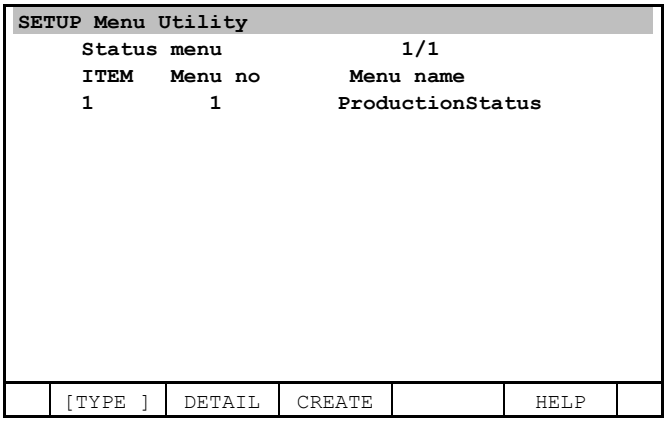
Procedure 36-4 Setup Status Menu

Step

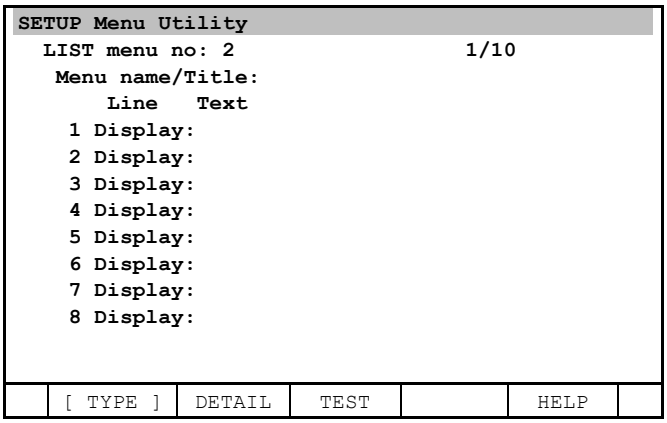
1 Press the [MENU] key and select ”6 SETUP” and select ”Menu Utility” so that Menu utility screen will be displayed .



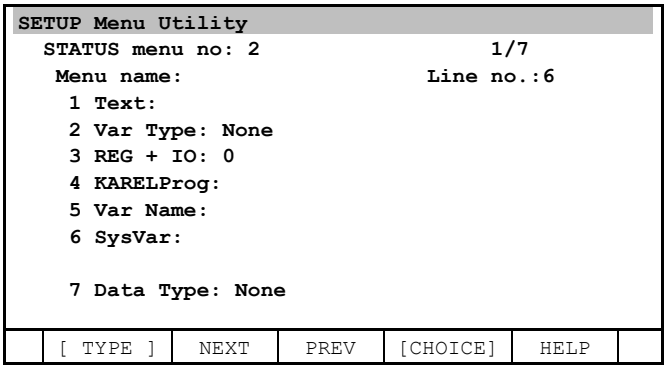
2 Set cursor to “4 Status menus” and press the F2, DETAIL so that Prompt box will be displayed.



3 Press the F3, CREATE if you want to create new menu. Press the F2, DETAIL if you want to change menu that already exists.



4 Set cursor to line that you want to edit, and press F2, DETAIL. The following screen will be displayed.



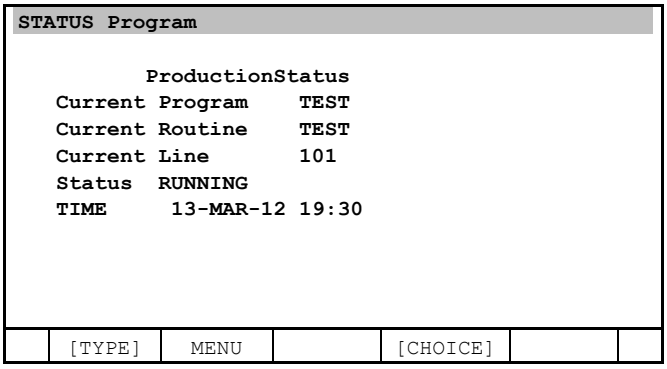
5 Setup each item. It is possible to setup if you set cursor objective item and press the F4, [CHOICE]. Please refer to table 36.1.4 and setup each item.

Table 36.1.4 Status menu variable detail information

ITEM	DESCRIPTION		
Text	This item allows you to specify the text that will be displayed for the status item. You can type up to 17 characters of display text.		
Var Type	This item allows you to specify the kind of variable for the status item. You must assign the following parameter and data type to depending on Type of Variable.		
	Type of Variable	**What to Define**	**Data Display Type**
	NONE	None	None
	R[]	REG Number	INTEGER, REAL, BOOLEAN
	DIN[]	I/O Number	INTEGER, BOOLEAN, TEXT
	DOUT[]	I/O Number	INTEGER, BOOLEAN, TEXT
	RI[]	I/O Number	INTEGER, BOOLEAN, TEXT
	RO[]	I/O Number	INTEGER, BOOLEAN, TEXT
	GIN[]	I/O Number	INTEGER
	GOUT[]	I/O Number	INTEGER
	KAREL VAR	Prog. Name, Var. Name	INTEGER, REAL, BOOLEAN,TEXT
	SYSTEM VAR	System Variable Name	INTEGER, REAL, BOOLEAN,TEXT
	TIME	None	----
	CURR PROGRAM	None	----
	CURR ROUTINE	None	----
	TIMER (SEC)	Prog. Name, Var. Name	----
	TIMER (MIN)	Prog. Name, Var. Name	----
	TIMER (HR)	Prog. Name, Var. Name	----
	CURR LINE	None	----

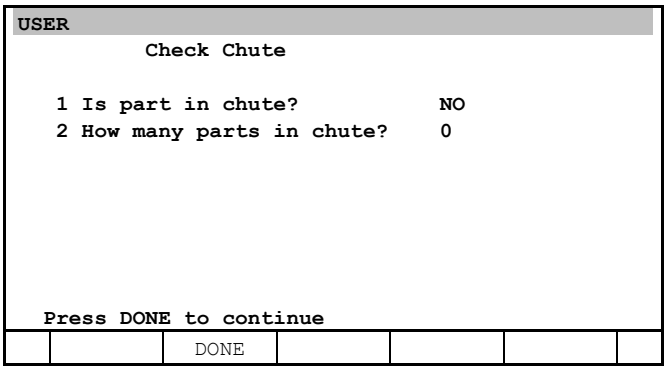
ITEM	DESCRIPTION
REG+IO	This item allows you to specify the number of the register or the I/O signal, when the variable type is R[ ],DIN[ ],DOUT[ ],RI[ ],RO[ ],GIN[ ], or GOUT[ ].
KAREL Program	This item allows you to specify the name of the KAREL program that contains the variable you are displaying, when the variable type is KAREL VAR.
Variable Name	This item allows you to specify the name of the variable you are displaying.
System Variable	This item allows you to specify the name of the system variable you are displaying, when the variable type is SYSTEM VAR.
Data Type	This item allows you to specify the variable data types: ・ NONE ・ INTEGER ・ REAL ・ TEXT ・ BOOLEAN

6 Press the F3, TEST if you want to confirm the Status menu edited in the step 3. The following screen will be displayed.



36.1.5 Operator Entry Menu

An operator entry menu requires the operator to enter INTEGER, REAL, BOOLEAN, or text values. When an operator entry menu is executed, the program pauses until the operator types the information and then presses the F2, DONE.



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.

Save Menu setting:
Each operator entry menu is stored in its own variable file that begins with the prefix “MENU5”. The next two characters of the filename are the menu number. For example, the list menu 1 is stored in the variable file named “MENU501.VR”.

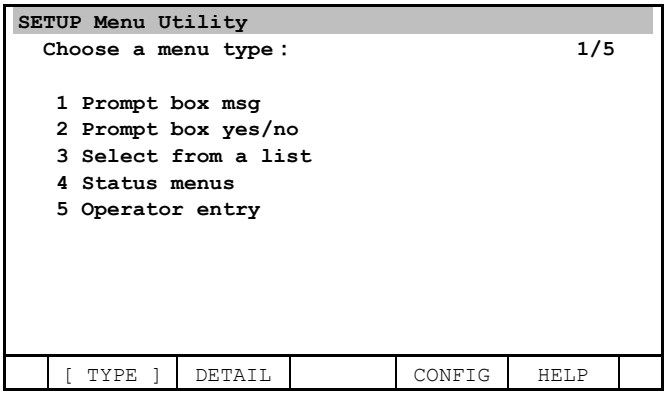
Usage example:
Example 1:
PROG TEST1
1: Status Menu (2)

Example 2:
PROG TEST2
1: Status Menu (‘ProductionStatus’)

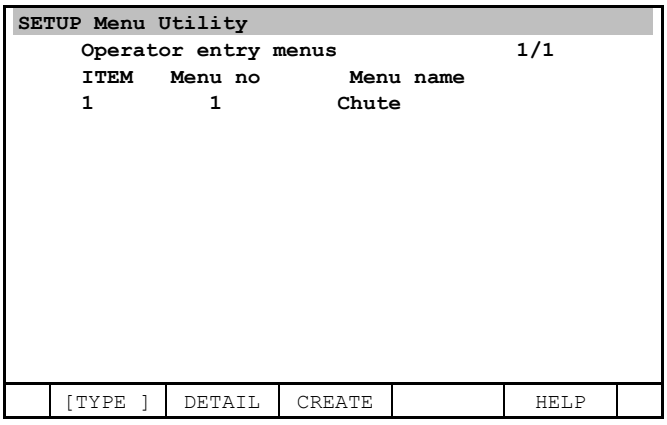
Procedure 36-5 Setup Operator entry

Step

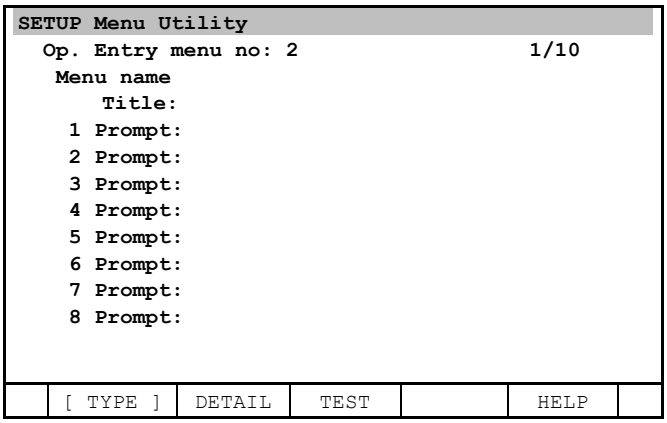
1 Press the [MENU] key and select “6 SETUP” and select ”Menu Utility” so that Menu utility screen will be displayed .



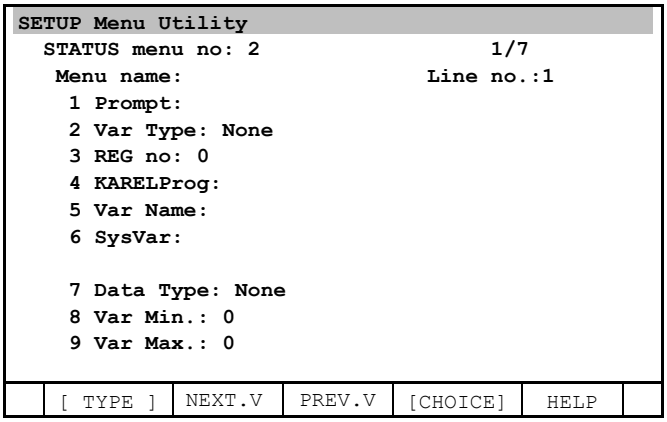
2 Set cursor to “5 Operator entry” and press the F2, DETAIL so that Prompt box will be displayed.



3 Press the F3, CREATE if you want to create new menu. Press the F2, DETAIL if you want to change menu that already exists.



4 Set cursor to line that you want to edit, and press the F2, DETAIL.



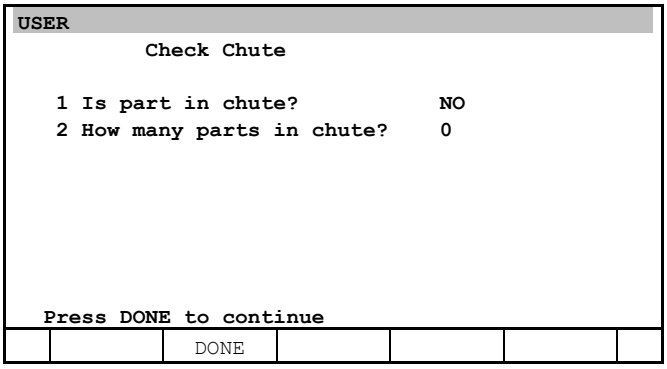
5 Setup each item. It is possible to setup if you set cursor objective item and press the F4, [CHOICE] or [ENTER] key. Please refer to the table 36.1.5 and setup each item.

Table 36.1.5 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	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
	SYSTEM VAR	System Variable	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.		
System Variable	This item allows you to specify the name of the system variable you are using, when the variable type is SYSTEM 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 variables.		

ITEM	DESCRIPTION
Variable Maximum	This item allows you to specify the maximum value allowed for the INTEGER or REAL variables.

6 Press the F3, TEST if you want to confirm the screen that will be displayed in the step 3. The following screen will be displayed.



37 4D GRAPHICS FUNCTION

In 4D graphics function, robots, tools, parts and various work cells can be displayed as 3D models on *i*Pendant.
And various invisible internal data such as positions taught in a program can be visualized as 4th dimensional information on *i*Pendant.
4D means the fusion of 3D robot model and 1D internal data as 4th dimension of information.
The robot model moves as the real robot moves. In machine lock, only the robot model can be moved and the direction of the movement of the robot can be previewed.

To use this function, 4D GRAPHICS option (R764) is required.

> ⚠ **WARNING**
> When moving the robot with the pendant enabled, be sure to watch the robot instead of watching the TP screen. After the robot is in a safe state you can examine the pendant graphics.

> ⚠ **WARNING**
> 4D graphics might no be an accurate representation of the real world. Regarding the path of the robot arm, actually execute the program and check it.

The invisible data that can be displayed on *i*Pendant with 4D graphics function are:

1	Logical tool center point (TCP).	- 4D graphics display function
2	Frame settings	- 4D visual jog function
3	Robot jog settings	- 4D jog preview function
4	The points (node) in a TP program	- 4D node map function
5	Exact path that the robot followed	- 4D TCP trace function
6	Position registers	- 4D position register function
7	DCS safety zone (*1)	- 4D DCS function.

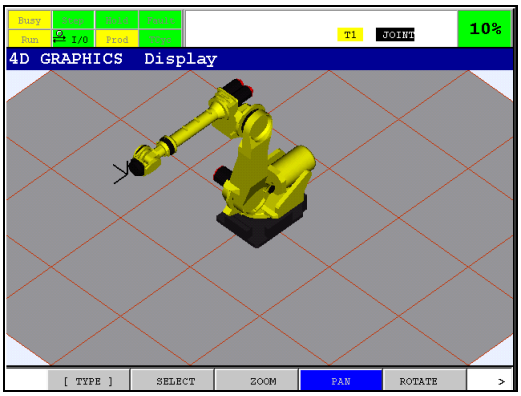
*1: This function requires DCS Pos./Seed check (J567) option.

To display 4D screen, please follow the operation 37-1 below.

Procedure 37-1 4D GRAPHICS Display screen

Step

1 Press the [MENU] key to display the screen menu.
2 Press the “0 NEXT,” then select “5 4D GRAPHICS”. The following screen will be displayed.
3 Alternatively, instead of steps 1 and 2 above, the user can press the [*i*] key and [POSN] key at the same time.



37.1 OVERVIEW

37.1.1 Graphic Models

4D GRAPHICS screen displays the following graphic models.
- Robot model
- Internal data
- Cell floor
- Peripheral devices

Robot model
A graphic model of ordered robot is displayed.

Internal data
Various internal data such as positions taught in a program or tool center point are displayed on 4D graphics screen.

Cell floor
This is a reference floor with a 1 meter grid. The position of cell floor can be set in SETUP Frames screen.

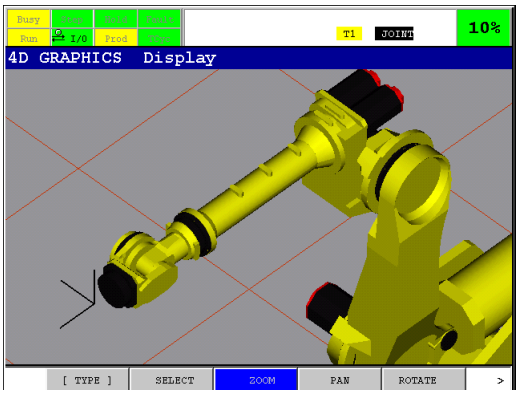
Peripheral devices
Peripheral device graphic models such as a works, fences and tools can be located anywhere in 4D GRAPHICS scene by using 4D Editor function in ROBOGUIDE. Please refer to section 37.3 4D Editor Function for details.

37.1.2 Operation Procedure

In 4D GRAPHICS screen, view is adjusted by the following operation modes.
- ZOOM
- PAN
- ROTATE

And the following functions are prepared.
- Preset Views
- User Views

ZOOM



In ZOOM mode, you can change the magnification of the viewpoint. Increasing the magnification makes the objects larger but the field of view is narrow. Press the [F3], ZOOM to set the system to zoom mode. Then, the label F3, ZOOM turns blue in color.
There are two ways for ZOOM operation. One is the key input and the other is touch panel operation.

Zoom by key input

Increase the magnification

- Press the up arrow key on iPendant.
- Press the [SHIFT] + up arrow key on iPendant. (High magnification)

Decrease the magnification

- Press the down arrow key on iPendant.
- Press the [SHIFT] + down arrow key on iPendant. (High magnification)

Zoom by touch panel

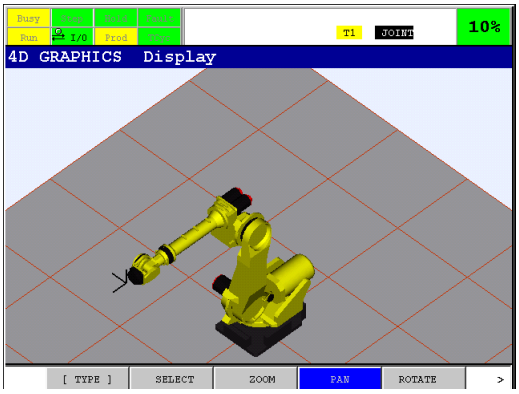
Increase the magnification

- Touch and release near the top of the screen.
- Touch the screen and drag your finger up or right,

Decrease the magnification

- Touch and release near the bottom of the screen.
- Touch the screen and drag your finger down or left,

PAN



In PAN mode, you can move up, down, left and right without changing the angle of the viewpoint. Press the [F4] key, PAN to set the system to PAN mode. Then, the label F4, PAN turns blue in color.

There are two ways for PAN operation. One is the key input and the other is touch panel operation.

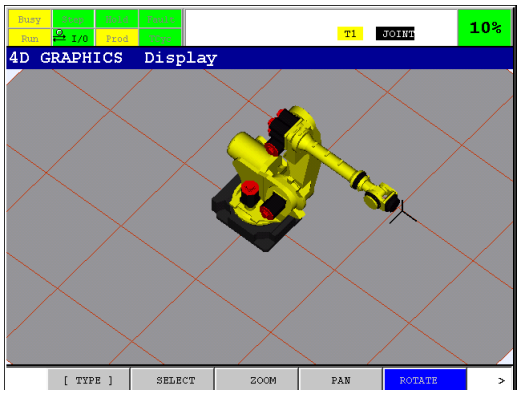
PAN by key input

- Press the arrow key on iPendant up, down, left and right.

PAN by touch panel

- Touch the screen and drag your finger up, down, left and right.
- Click the screen so the point you click becomes center on the screen.

ROTATE



In ROTATE mode you can change the angle without changing the position of the viewpoint. Press the [F5], ROTATE to set the system to ROTATE mode. Then, the label F5, ROTATE turns blue in color.

There are two ways for ROTATE operation. One is the key input and the other is touch panel operation.

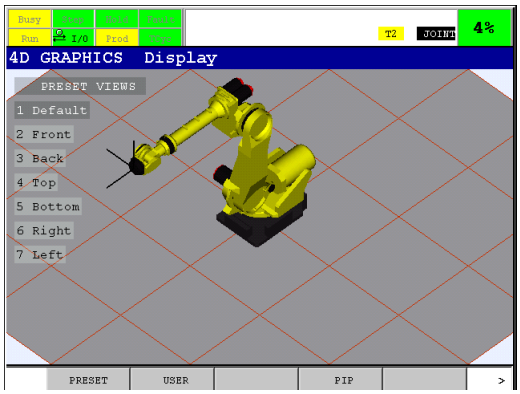
ROTATE by key input

- Press the arrow key on iPendant up, down, left and right.
- Press the [SHIFT] + arrow key on iPendant up, down, left and right. (High magnification)

ROTATE by touch panel

- Touch the screen and drag your finger up, down, left and right.

Preset Views



4D GRAPHICS screen provides seven preset views. The default view provides a view from 45 degree. This view is good starting point for setting the view. It also put information back on the screen in the case where it has inadvertently been lost. All preset views will center the floor in the middle of the view.

- Default Set the view to default
- Front Set the view in front of the robot
- BackSet the view behind the robot
- Top Set the view right above the robot
- Bottom Set the view right below the robot
- 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 the [NEXT] key and press the F2, [VIEWS]. Select one of the preset views above.

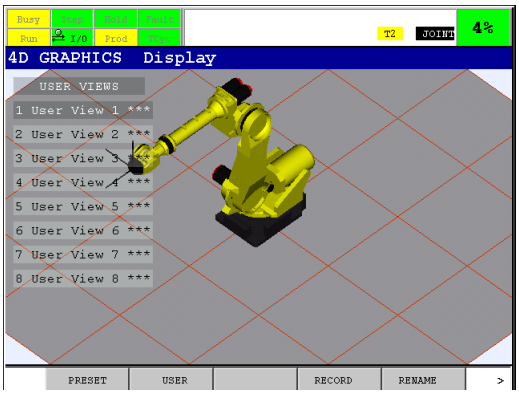
User Views

4D GRAPHICS screen provides user view function that can be recorded eight views. These views can be recorded and then retrieved by selecting them. You can also name the recorded user views.

Procedure 37-2 Record User Views

Step

1 Move to the view you want to record.
2 Press the [NEXT] key and press F2, [VIEWS].
3 Press the F2, [USER] so following screen will be displayed.



4 Select one of User Views you want to record and press the F4, [RECORD].

When User View was recorded, "***" displayed the right of the User View was removed.
Once a User View is recorded, it is available to be retrieved at any time.
Press the F5, [RENAME] on one of User Views, then User View can be named.

37.2 4D GRAPHICS SCENE

In 4D GRAPHICS function, the following scenes are available.
- 4D GRAPHICS Display
- 4D GRAPHICS Edit Node Map
- 4D GRAPHICS Select Node Map
- 4D GRAPHICS Position Register
- 4D GRAPHICS TCP Trace
- 4D GRAPHICS Frame Display
- 4D GRAPHICS CDS (This function requires DCS Pos./Seed check (J567) option.)

Pressing the F1, [TYPE] in any 4D GRAPHIC scene and selecting one of them display the selected scene.

37.2.1 4D GRAPHICS Display

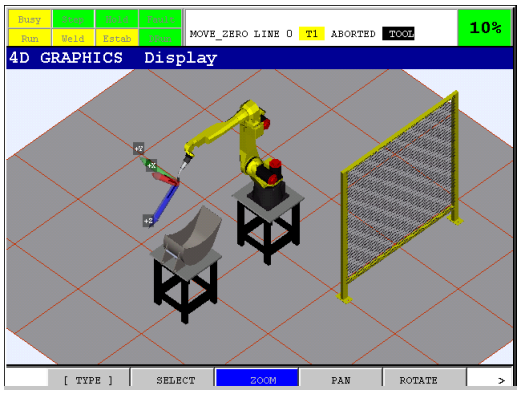
4D GRAPHICS Display scene is the standard scene and it can composite together any multiple scenes to be viewed together.

In 4D GRAPHICS Display scene, the following functions are available.
- Visual Jog
- Jog Preview

37.2.1.1 Visual jog

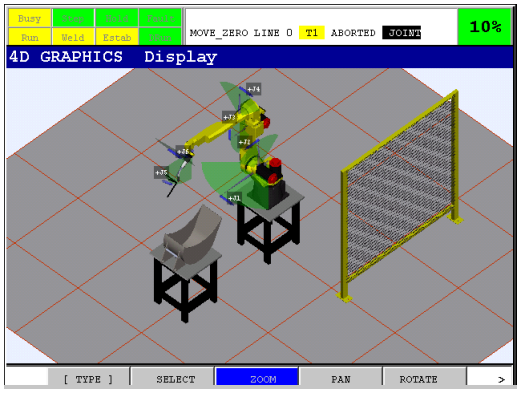
Visual Jog allows the coordinate system and the group selected for jogging to be indicated on the 4D GRAPHICS Display. It allows you to preview the direction that the robot will move before actually jogging the robot.
Visual Jog is available when the teach pendant is enable.

Cartesian Visual Jog



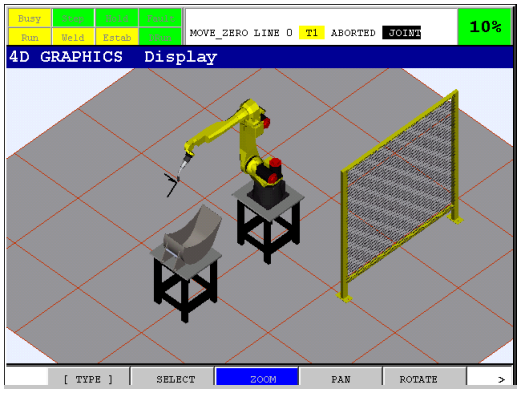
In Cartesian visual jog the red, green and blue axes correspond to the X, Y and Z directions.

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.

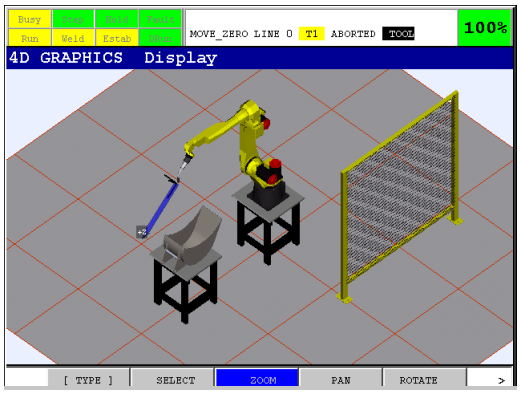
Pressing the [*i*] key and the COORD key simultaneously turn this function on and off.



37.2.1.2 Jog preview

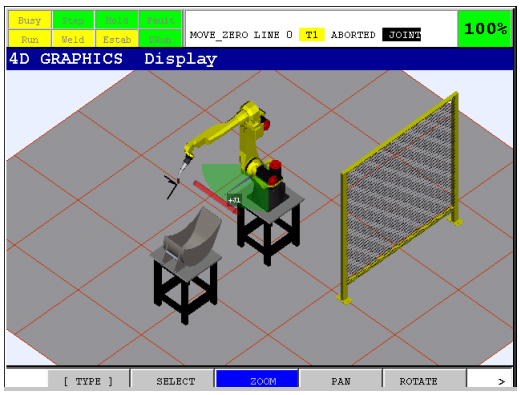
Jog Preview can preview the direction of movement of the robot when a particular jog key is pressed. It is activated when pressing a single jog key at the same time as the *i* key. Pressing more than one jog key will display the jog direction of the last key pressed. It is not necessary to enable the DEADMAN or clean errors. Jog Preview is available when the teach pendant is enable.

Cartesian Jog Preview



Cartesian Jog Preview shows a single arrow indicating the direction that the robot will move. Depending on the setting of the speed override, the size of the arrow will change. The figure shows an example when *i* key and +Z jog key.

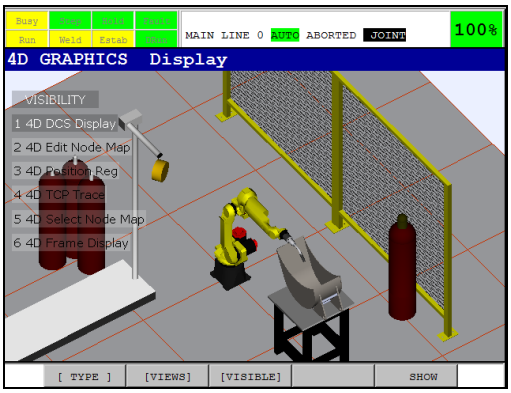
Joint Jog Preview



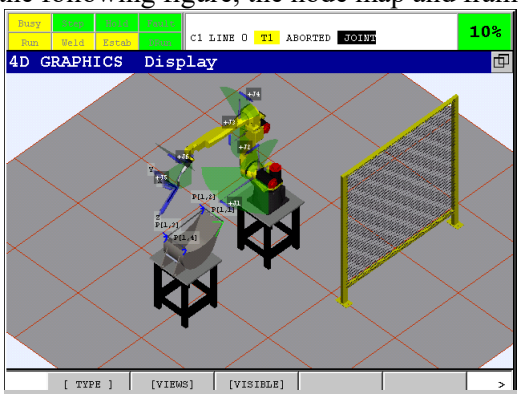
Joint Jog Preview shows a single red arrow tangent to the fan indicating the direction that the joint will move. Depending on the setting of the speed override, the size of the arrow will change. The figure shows an example when *i* key and +J1 jog key.

37.2.1.3 Set visibility: 4D GRAPHICS display

The elements which are visible in 4D GRAPHICS Display scene is controlled by the visibility softkey on page two of the softkeys. Pressing the F3, [VISIBLE] on page two of the softkeys shows the following menu.



For example, selecting 4D Edit Node Map adds the node map in the scene. Selecting the same item again to make the element invisible. In the following figure, the node map and frame elements are added.



NOTE
When there are too much information displayed, the performance of 4D GRAPHICS function declines.
In that case, make unnecessary data invisible.

37.2.2 4D GRAPHICS Node Map

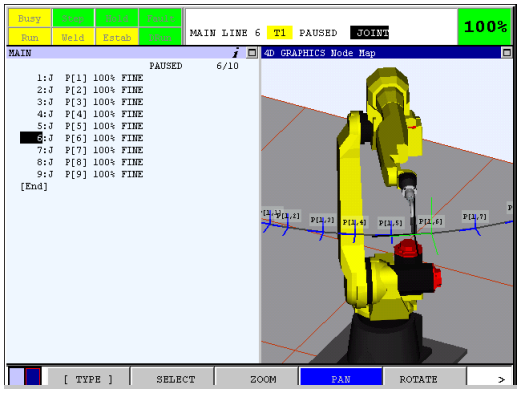
Node map is a 4D visualization of a TP program. This shows all the positions as nodes and connect them with lines. Each node has its position number in the text label elements. The main purpose of node map is to see the relationship among the positions in a program without moving a robot.

NOTE
Node Map connects two nodes in the program in descending order. Therefore, it is sometimes different from the real motion path.

NOTE
When INC, OFFSET or Position Register is used in a motion statement or when frame number is changed in a program, the node will be displayed in the different position from the actual position.

Each node is colored in blue, green or yellow. The blue node means normal, the green one means the position focused in EDIT screen and the yellow one means the position is unspecified, such as the position register. The line between nodes is usually colored in gray or red. The red one means there is an unspecified position between the two nodes.
Edit node map and select node map are functionally a little different from each other.

37.2.2.1 4D GRAPHICS edit node map



In 4D GRAPHICS Edit Node Map scene, the currently selected program is displayed. It is linked to EDIT screen. So the focused position is displayed as a green node in node map. As the cursor is moved in EDIT screen, the green node also moves. The node map will be updated immediately when you add, delete or correct the position on the edit screen.

Procedure 37-3 4D GRAPHICS Edit Node Map

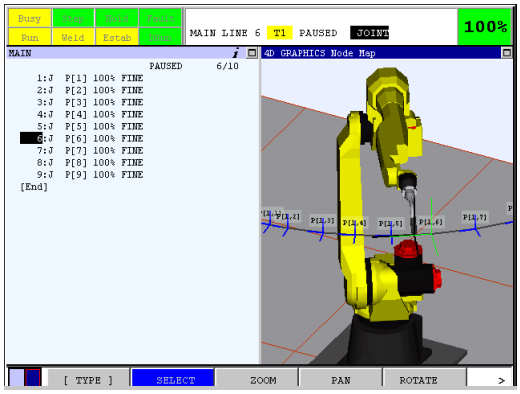
Step

There are some other ways to display 4D GRAPHICS Edit Node Map in addition to selecting the item from F1, [TYPE].

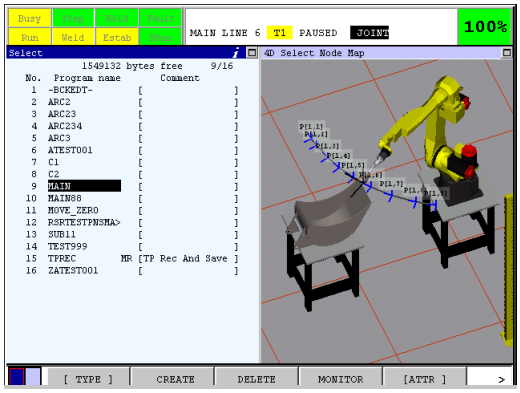
1 Press the [EDIT] key. When a program is selected, EDIT screen will be displayed.
2 Press the [*i*] key and [FCTN] key on EDIT screen. Related View menu will be displayed.
3 Select “4D Edit Node Map”.
4 Instead of step 1 to step 3, Pressing the [*i*] key and [EDIT] key displays the node map.

SELECT Data

Selecting a node on the 4D graphics edit node map screen can make the cursor in EDIT screen 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 this functionality, select F2, SELECT. Then, the label F2, SELECT turns blue in color.



37.2.2.2 4D GRAPHICS select node map



In 4D GRAPHICS Select Node Map scene, the currently focused program in SELECT screen is displayed. It is linked to SELECT screen. As the cursor is moved in SELECT screen, displayed node map also changes. When a program is selected in SELECT screen, EDIT screen of the program is displayed. Then, 4D GRAPHICS Edit Node Map is also displayed automatically.

Procedure 37-4 4D GRAPHICS Select Node Map

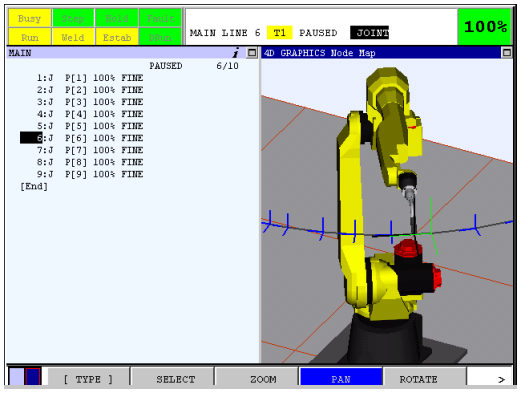
Step

There are some other ways to display 4D GRAPHICS Select Node Map in addition to selecting the item from F1, [TYPE].

1 Press the [SELECT] key. SELECT screen will be displayed.
2 Press the [*i*] key and [FCTN] key on SELECT screen. Related View menu will be displayed.
3 Select “4D Display”.
4 Instead of step 1 to step 3, Pressing [*i*] key and [SELECT] key displays the node map.

37.2.2.3 Set Visibility: node map position number

The position number element can be turned on and off. To do it, press the [NEXT] key and then press the F3, [VISIBLE]. The following figure shows a node map without the position number.



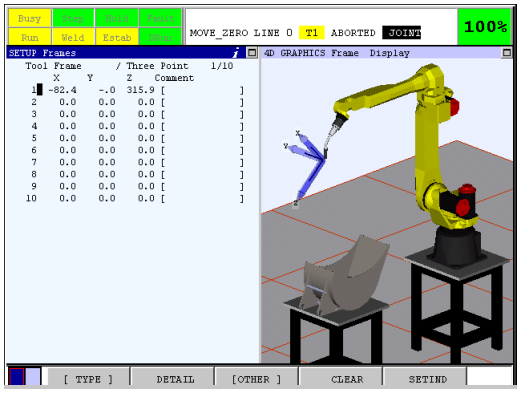
37.2.2.4 Data supported by node map

The following data are supported in node map.

- Normal positions
- Positions specified by Position Register
- Via positions and end positions in Circular motion
- Incremented position by INC option

37.2.3 4D GRAPHICS Frame Display

4D GRAPHICS Frame Display scene shows the following graphics of the current frame setup operation. The comments for the frame is also displayed.



Procedure 37-5 4D GRAPHICS Frame Display

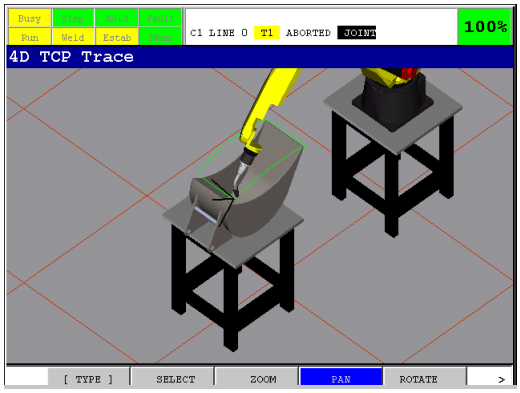
Step

There are another way to display 4D GRAPHICS Frame Display in addition to selecting the item from F1, [TYPE].

1 Press the [MENU] key to display the menu screen.
2 Select “6 Setup”.
3 Press the F1 [TYPE] to display the type menu.
4 Select “Frames”.
5 Press the [*i*] key and [FCTN] key on Frame setup screen. Related View menu will be displayed.
6 Select “4D Frame Display”.

37.2.4 4D GRAPHICS TCP Trace

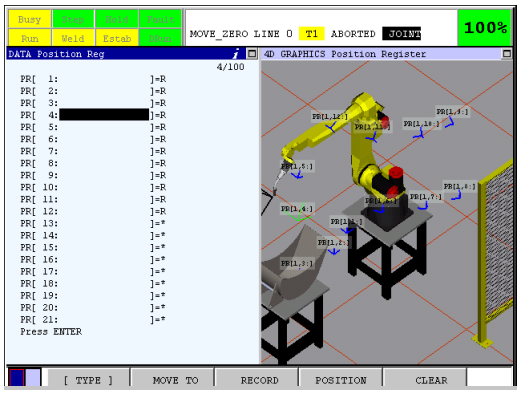
4D GRAPHICS TCP Trace scene provides the capability to view the path that the robot took when executing a program. The path is displayed in green. The path is updated every time a program is executed.



NOTE
The path provided in 4D GRAPHICS TCP Trace are possibly different from actual path.

37.2.5 4D GRAPHICS Position Register

In 4D GRAPHICS Position Register scene, position registers are displayed as nodes like node map. This helps understand where the position registers are located spatially. Each node has its position register number.
4D GRAPHICS Position Register is linked to Position Register screen. Focused position register is displayed as green highlighted node. When a position register is newly taught or modified, the graphics are automatically updated.



Procedure 37-6 4D GRAPHICS Position Register

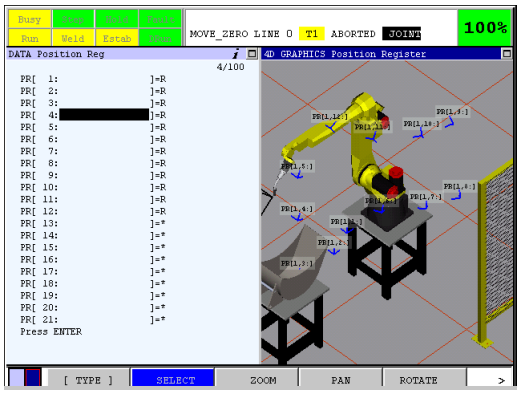
Step

There are another way to display 4D GRAPHICS Frame Display in addition to selecting the item from F1, [TYPE].

1 Press the [DATA] key and display DATA screen.
2 Press the F1 [TYPE] to display the type menu.
3 Select “Position Reg”.
4 Press the [*i*] key and [FCTN] key on Position Register screen. Related View menu will be displayed.
5 Select “4D Position Reg”.
6 Instead of step 1 to step 5, pressing the [*i*] key and DATA key display the screen.

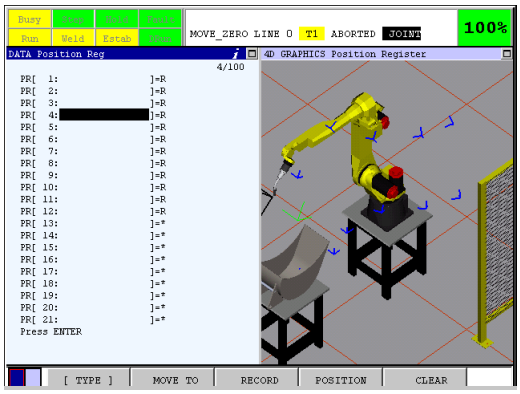
SELECT Data

Selecting a node can make the cursor in Position Register screen move to the corresponding line. 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 this functionality, select F2, SELECT. Then, the label F2, SELECT turns blue in color.



37.2.5.1 Set Visibility: position register number

The position register number element can be turned on and off. To do it, press the [NEXT] key and then press the F3, [VISIBLE]. The following figure shows a node map without the position register number.

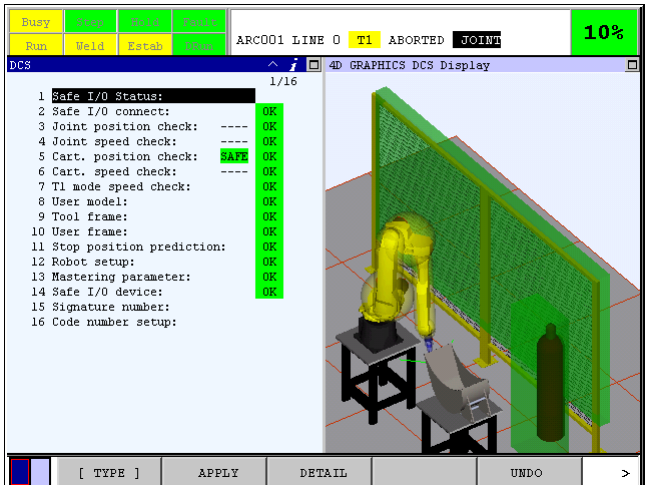


NOTE
Position registers are always considered to be in the “current” frame. The position register graphical display reflects the position of the position register in the current frame. This may not be how the position register is used. Changing the current frame will be reflected in the position of the position register triad in the graphics display.

NOTE
Any register which has been set to joint representation will not be displayed.

37.2.6 4D GRAPHICS DCS

In 4D GRAPHICS DCS scene, DCS safety zones are displayed. By this function DCS safety zones can be set and check graphically. This section describes the way of displaying 4D DCS screen.
The robot controller must have the DCS Pos./Speed check (J567) option installed to use this function.
Please refer to section 8 DCS VISUALIZATION in R-30*i*B/R-30*i*B Mate CONTROLLER Dual Check Safety Function OPERATOR’S MANUAL (B-83184EN) for details of 4D graphics DCS function.



Procedure 37-7 4D GRAPHICS DCS

Step

There is another way to display 4D GRAPHICS DCS in addition to selecting the item from F1 [TYPE] in 4D graphics display screen.

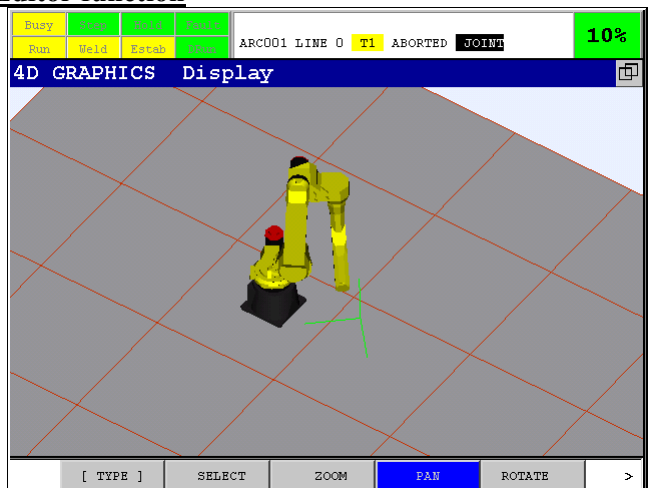
1 Press the [MENU] key and display MENU pop up screen.
2 Select [0 -- NEXT --] and select [6 SYSTEM].
3 Press the F1 [TYPE] to display the type menu and select [DCS].
4 Press the [*i*] key and [FCTN] key on DCS screen. Related View menu will be displayed.
5 Select “4D DCS Display”.

37.3 4D EDITOR FUNCTION

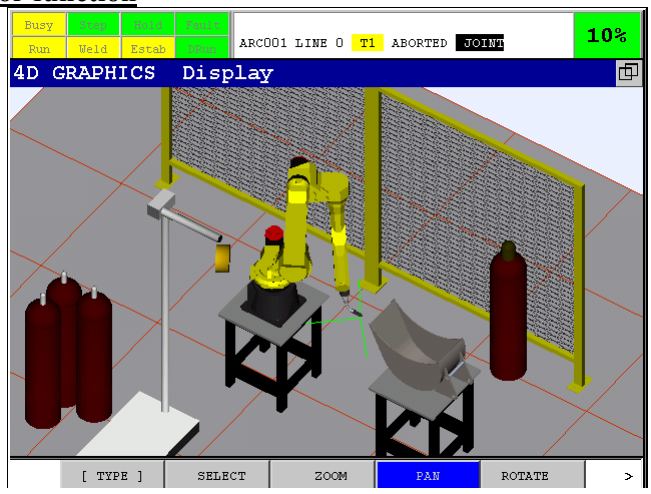
37.3.1 Overview

Peripheral device graphic models such as a works, fences and tools can be located anywhere in 4D GRAPHICS scene by using 4D Editor function in ROBOGUIDE. This section describes the screen using 4D Editor function.
Please refer to the ROBOGUIDE helps of 4D Editor for details of 4D Editor function.

Graphics not using 4D Editor function



Graphics using 4D Editor function



4D Editor function enabled to display graphics more closely to the real system.

37.4 FULL SCREEN 4D DISPLAY

37.4.1 Overview

By this function, 4D graphics screen can be displayed on a PC.
This function requires the Internet Conn/Cust (R558) and R764 4D GRAPHICS function and (R764) option.

NOTE
For early releases, display of 4D information requires that the PC have an advanced graphics card capable of displaying OpenGL. If your PC does not work, updating your PC graphics drivers might fix it.

37.4.2 Setup

37.4.2.1 Requirements

The following are the requirements for Full Screen 4D Display.

- PC must have Microsoft® Internet Explorer 5.5 or greater installed.
- PC must have the *i*Pendant Controls installed. Please refer to section 15.2 iPENDANT CONTROLS INSTALLATION in Ethernet Function OPERATOR'S MANUAL (B-82974EN) for installation instructions.
- PC must have an advanced graphics card capable of displaying OpenGL.
- PC must be connected to a network, and be properly configured to allow a TCP/IP connection to the Robot Controller with the *i*Pendant connected.
- The robot controller must be connected to a network and be properly configured for Network access to the above PC.
- The robot controller must have the Internet Conn/Custo (R558) option installed.
- The robot controller must have the 4D Graphics (R764) option installed. And the robot controller must have the DCS Pos./Speed check (J567) option installed to display 4D DCS Display.
- The HTTP Authentication must be properly configured on the robot to allow *i*Pendant access. Please refer to section 15.2 HTTP AUTHENTICATION in CONTROLLER Ethernet Function OPERATOR'S MANUAL (B-82974EN) for configuration information.

37.4.2.2 Configuring Microsoft® Internet Explorer

Please refer to section 15.3.2.2 Configuring Internet Explorer® in Ethernet Function OPERATOR'S MANUAL (B-82974EN) for configuration information.

37.4.3 Operation

After you have properly configured Internet Explorer ® and verified that you can connect to the robot, you can now display the Full Screen 4D Display screen. The following sections detail the operation and the limitations of this feature.

37.4.3.1 Accessing the Full Screen 4D Display

This section will describe the method to connect to the robot controller and display the Full Screen 4D Display screen.

Procedure 37-8 Accessing The Full Screen 4D Display

Steps

1 Bring up Internet Explorer on the PC.

2 In the Internet Explorer Address field, type the following: http://<myrobot_name_ or 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 (i.e. pderob111.frc.com) or the IP address of your robot (i.e. 192.168.1.100).
If the item is “Full Screen 4D Display : Unavailable”, check if 4D Graphics (R764) option is ordered.
If the item is not displayed, check if Internet Conn/Custo (R558) option is ordered.

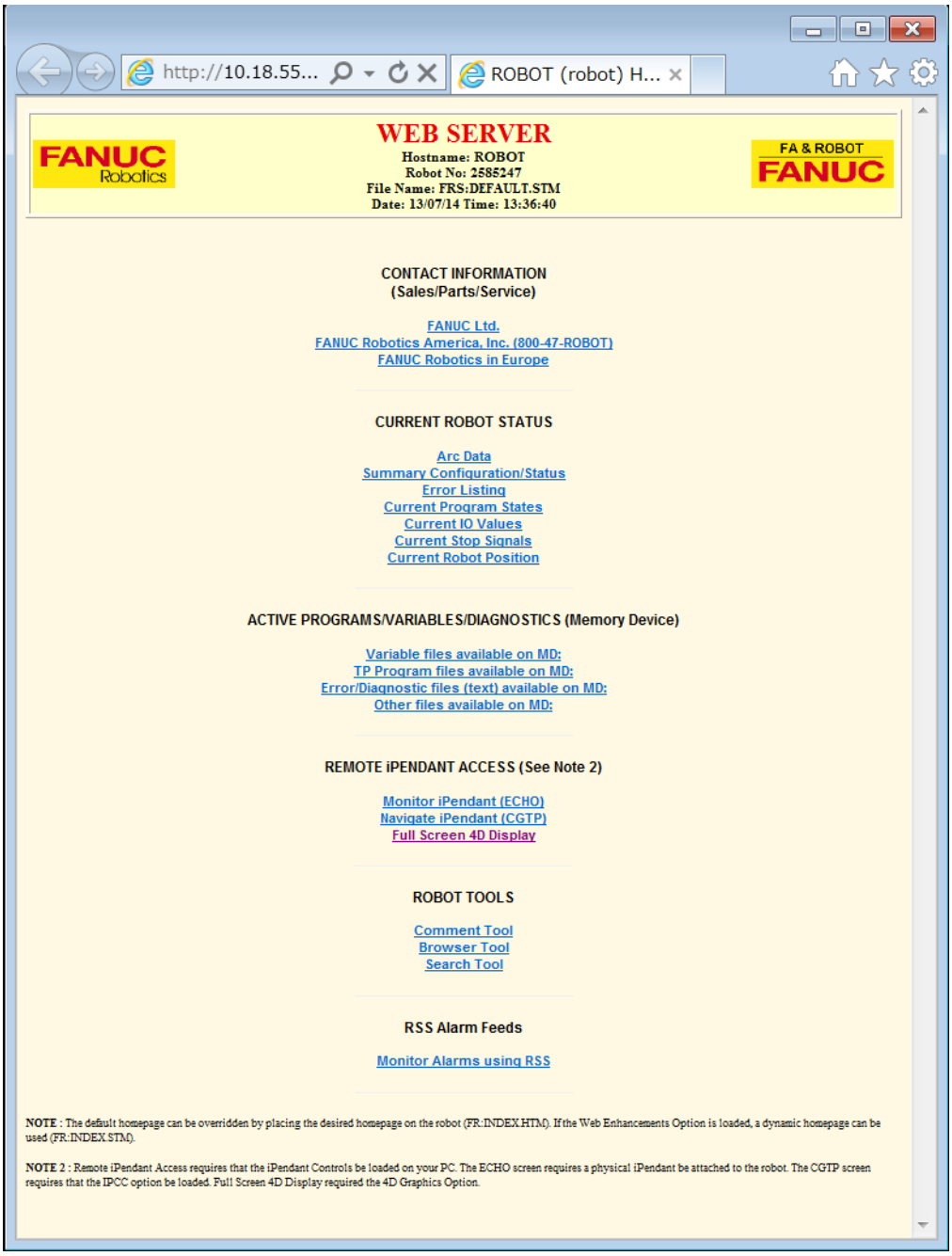


Fig. 37.4.3.1(a) Robot Home Page

3 From the HOME page, when you select Full Screen 4D Display, 4D graphics display screen similar to the one shown in Figure 37.4.3.1(b) is displayed on your PC.

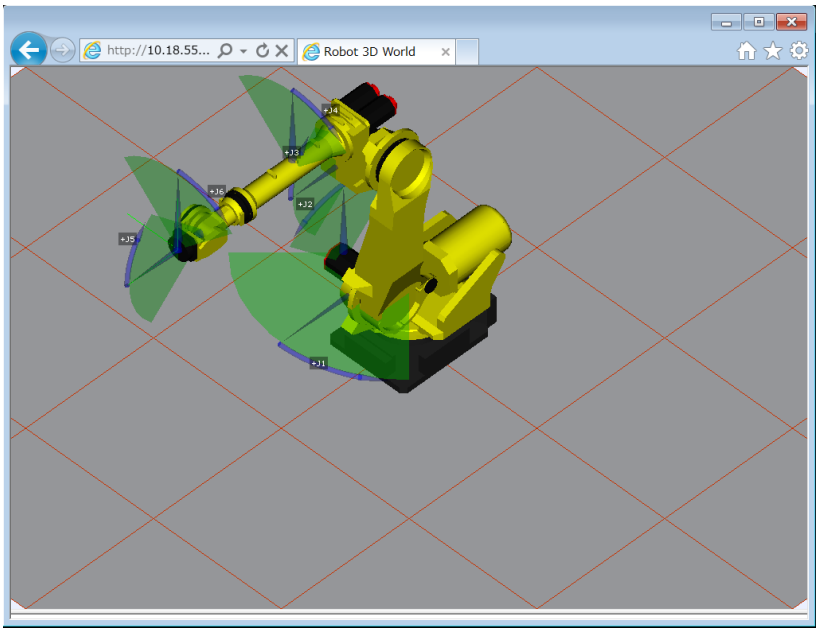


Fig. 37.4.3.1(b) Full Screen 4D Display Screen

37.4.3.2 View adjustment mode

The view of the 4D scene on PC can be adjusted with the mouse. The mouse can PAN, ROTATE or ZOOM the view.

Procedure 37-9 Selecting View Adjustment Mode

Steps

1 Click with the right (alternate) mouse button. Menus of 4D graphic display will be displayed at the upper left of the screen. Right click again to dismiss the menu.

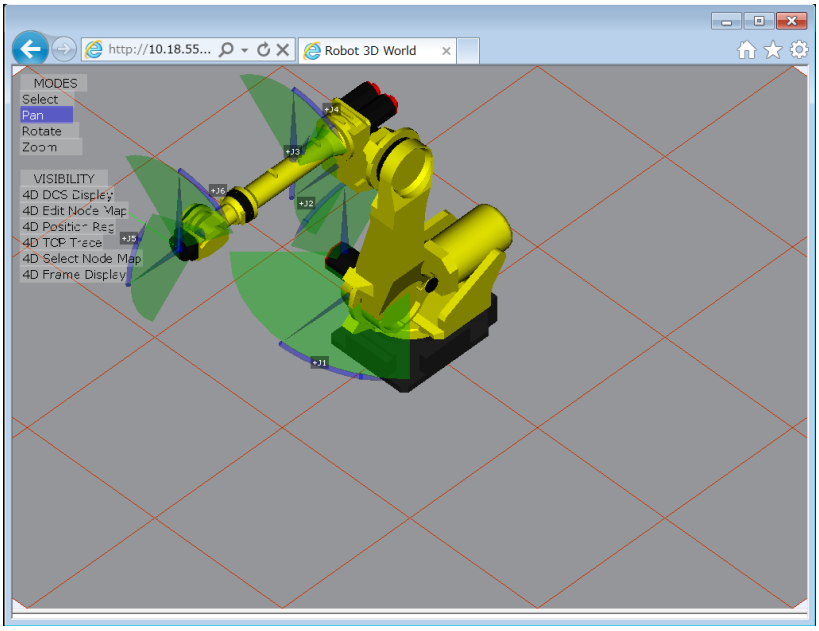


Fig. 37.4.3.2 Full Screen 4D Display Screen

2 Use either mouse button to select the view adjustment mode you want from the following three 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 selection of any combination of 4D display entities. The highlighted (white on blue) elements in the menu are the ones which are currently being displayed. Selecting an element will toggle the visibility on or off.

Procedure 37-10 Selecting Scene Visibility

Steps

1 Click with the right (alternate) mouse button to display the menu.

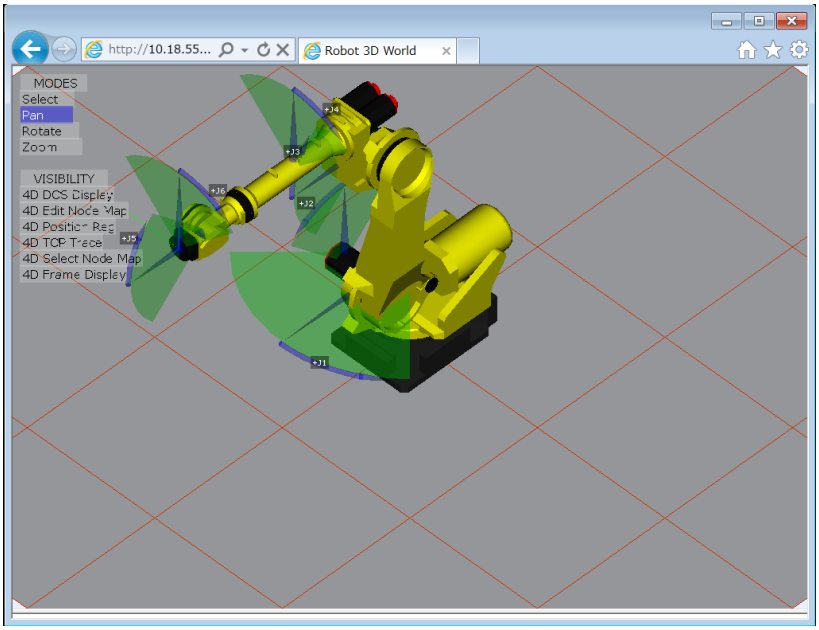


Fig. 37.4.3.3(a) 4D Graphics Node Map Scene

2 Use the left or right mouse button to select the scene you want to toggle:

- To change visible/invisible status of each data, select them.
- When you select multiple items in the scene information, the all selected scene will be displayed in 4D screen.

Please refer to section 37.2 4D GRAPHICS SCENE for details of each scene.

For example, at the system like Fig. 37.4.3.3(b), when you select 4D Select Node Map, the node map is displayed on your PC.

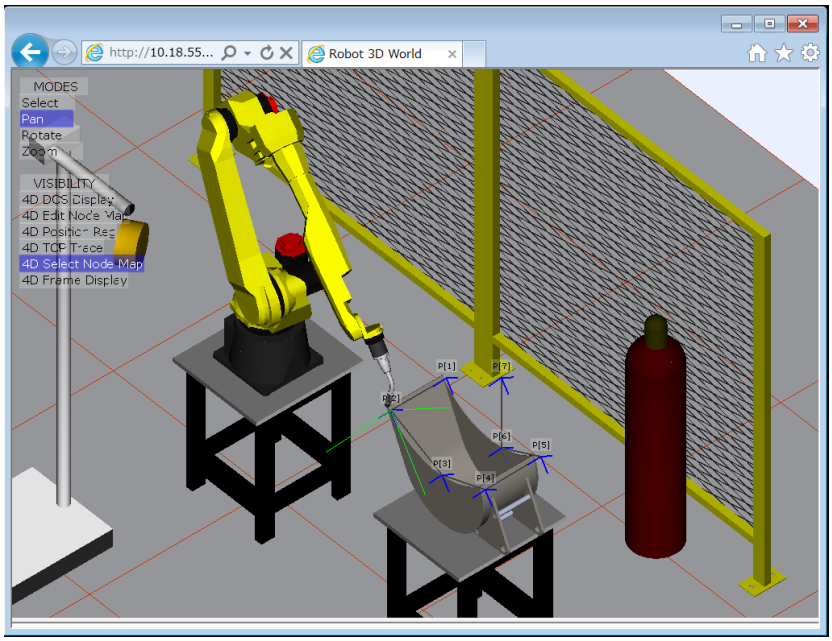


Fig. 37.4.3.3(b) 4D Graphics Node Map Scene

3 Click anywhere outside the menu to dismiss the menu

NOTE
If DCS Pos./Speed check (J567) option is not loaded, the item “4D DCS Display” is not displayed.

38 DATA TRANSFER BETWEEN ROBOTS FUNCTION

This section explains data transfer between robots function. This function is optional (J740).

This function enables you to transfer data between robots over Ethernet.
By calling KAREL program, you can transfer register and position register between robot controllers.

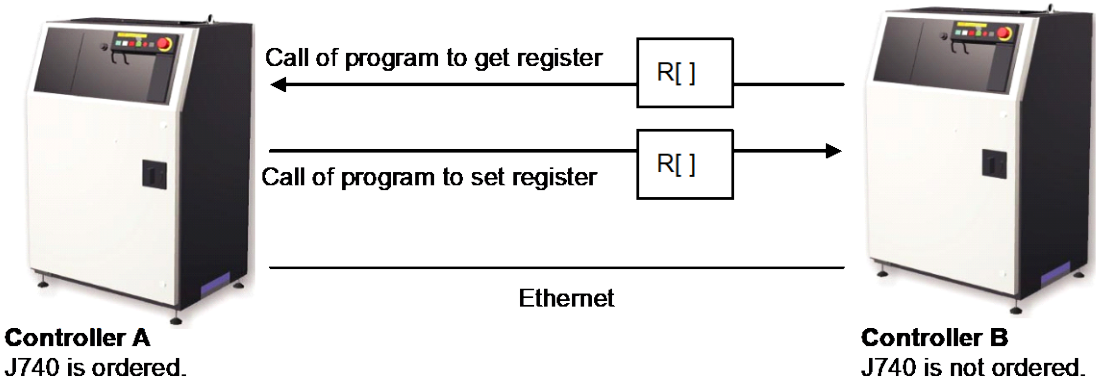


Fig. 38 (a) Data transfer between robots

In Fig. 38, controller A runs program to get register from controller B. In addition, Controller A call program to transfer register to controller B.
In Fig. 38, only controller A starts data transfer. Controller B is just responding to request from controller A. In this case, only controller A needs this option.

NOTE
In case that robot controller to communicate with (controller B in Fig. 38(a)) is R-30*i*A or R-30*i*A Mate controller, series of the system software of that controller must be 7DA5 or later.

NOTE
Position register cannot be transferred between R-30*i*B Plus/R-30*i*B Mate Plus controller and R-30*i*B/R-30*i*B Mate/R-30*i*A/R-30*i*A Mate controller.

This function also provides KAREL built-in for data transfer. By KAREL programming you can transfer register and comment of I/O of remote controller. You can also read and write value of I/O. KAREL is program language for robot. It is used to for construction of robot system. For more detail on KAREL, please refer to “KAREL function OPERATOR’S MANUAL” (B-83144EN). Please refer to “38.7 KAREL built-in” for built-in routines provided by this function.

38.1 TERMINOLOGY

To simplify and clarify explanation of this function we use following term.

Client

Client is a robot controller that starts communication by this function. In Fig. 38, controller A is client. Client runs program to request various service to another controller.

Server

Server is a robot controller that receives request from client and serve for it. In Fig. 38, controller B is server.

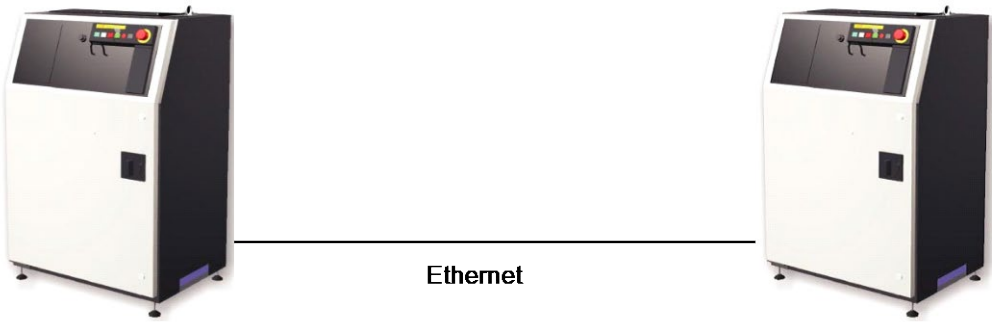
38.2 SETUP

Connect robot controllers by Ethernet. Set TCP/IP parameters so that controllers can communicate over Ethernet by TCP/IP.

1 Press the [MENU] key.
2 Select SETUP.
3 Select Host Comm.
4 Display TCP/IP setup screen.
5 Input host name (Robot name) and IP address and those of controllers to communicate with.
6 Cycle power.

For more detail, please refer to “Ethernet Function OPERATOR’S MANUAL (B-82974EN)”, Chapter2 “Setting up TCP/IP”.

NOTE
Don’t use underscore (“_”) in host name.



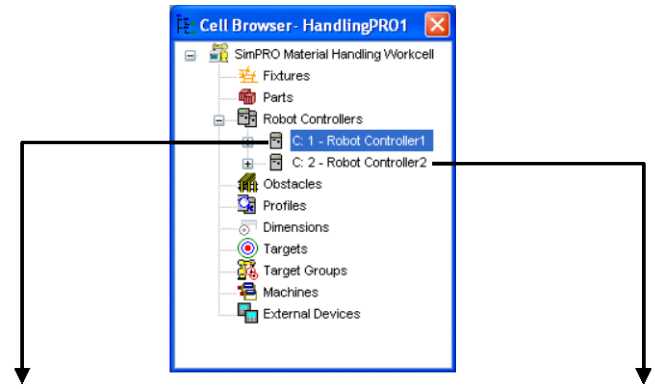
```
SETUP Host Comm
    TCP/IP                                1/40
        Robot name:                        RC1
        PORT#1 IP addr:          192.168.0.2
        Subnet Mask:             255.255.255.0
        Board address:           ******************
        Router IP addr:          ******************

      Host Name (LOCAL)  Internet Address
        1          RC1             190.168.0.2
        2          RC2             190.168.0.3
        3 **********     ******************
        4 **********     ******************

[ ] [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 work cell of ROBOGUIDE. Please note that IP address of each host must be 127.0.0.x.

Following figure is an example of TCIP/IP setup for ROBOGUIDE. There are two robot controllers in your work cell. Host name of the 1st one is RC1 and the 2nd one is RC2.




```
SETUP Host Comm
    TCP/IP                                1/40
        Robot name:                        RC1
        PORT#1 IP addr:          127.0.0.2
        Subnet Mask:             255.255.255.0
        Board address:           ******************
        Router IP addr:          ******************

      Host Name (LOCAL)  Internet Address
        1          RC1             127.0.0.2
        2          RC2             127.0.0.3
        3 **********     ******************
        4 **********     ******************

[ ] [TYPE] [ ] PORT PING HELP >
```

```
SETUP Host Comm
    TCP/IP                                1/40
        Robot name:                        RC2
        PORT#1 IP addr:          127.0.0.3
        Subnet Mask:             255.255.255.0
        Board address:           ******************
        Router IP addr:          ******************

      Host Name (LOCAL)  Internet Address
        1          RC1             127.0.0.2
        2          RC2             127.0.0.3
        3 **********     ******************
        4 **********     ******************

[ ] [TYPE] [ ] PORT PING HELP >
```

You have to re-start robot controllers after TCP/IP setup.

If you run following program, value and comment of R [1] of RC1 is transferred to R [5] of RC2.

```
1:  CALL RGETNREG('RC1',1,5,0) ;
```

RGETNREG is KAREL program this function provides. For more detail, please refer to Section 38.4 STANDARD DATA TRANSFER PROGRAM.

38.4 STANDARD DATA TRANSFER PROGRAM

This function includes programs to transfer register and position register. Following table lists the programs.

Program	Function	Syntax
RGETNREG	To get numeric register	RGETNREG('host/IP address', source index, destination index, option)
RSETNREG	To set numeric register	RSETNREG('host/IP address', destination index, source index, option)
RGETPREG	To get position register	RGETPREG('host/IP address', source index, source group number, destination index, destination group, option)
RSETPREG	To set position register	RSETPREG('host/IP address', destination index, destination group, source index, source group number, option)

We describe each program in subsection 38.5.1 and later.

NOTE
If arguments are not proper, it causes error. Followings are example of wrong arguments.
- Type of argument (integer, string and the like) is wrong.
- The number of arguments is less than expected.
- Content of argument is wrong.

Please call program with proper argument.
When you restart execution of program, please move cursor to 1 line before CALL and restart from the line.

38.4.1 Program to Get Numeric Register

Program name: RGETNREG

Overview

Client controller gets register from server and sets it to register of client.

Syntax

RGETNREG (host_name, src_idx, dest_idx, option)
Input/Output parameters
The 1st parameter: [IN] String of host name or IP address of server.
The 2nd parameter: [IN] Index of source register. This integer is index of register of server.
The 3rd parameter: [IN] Index of destination register. This integer is index of register of client.
The 4th parameter: [IN] This integer specifies function of this program.

Value	Description
0	Value and comment are got.
1	Value is got.
2	Comment is got.

Detail

If there isn't R [src_idx] on server, "DTBR-014 Bad variable or register index (host name, index)" is posted.
If there isn't R [dest_idx] on client, "VARS-024 Bad variable or register index" is posted.

Example

Execution of following program gets value and comment of R [10] of Host “SERVER”.

```
CALL RGETNREG (‘SERVER’, 10, 20, 0)
```

38.4.2 Program to Set Numeric Register

Program name: RSETNREG

Overview

Client controller sets client’s register to server’s register.

Syntax

RSETNREG (host_name, dest_idx, src_idx, option)
Input/Output parameters
The 1st parameter: [IN] String of host name or IP address of server.
The 2nd parameter: [IN] Index of destination register. This integer is index of register of server.
The 3rd parameter: [IN] Index of source register. This integer is index of register of client.
The 4th parameter: [IN] This integer specifies function of this program.

Value	Destination
0	Value and comment are set.
1	Value is set.

Detail

If there isn’t R [dest_idx] on server, “DTBR-014 Bad variable or register index (host name, index)” is posted.
If there isn’t R [src_idx] on client, “VARS-024 Bad variable or register index" is posted.

Example

Execution of following program sets value and comment of R [20] of client to R [10] of host “SERVER”.

```
CALL RSETNREG (‘SERVER’, 10, 20, 0)
```

38.4.3 Program to Get Position Register

Program name: RGETPREG

Overview

Client controller gets position register from server and set it to position register of client. Position data of specified group only is got.

Syntax

RGETPREG (host_name, src_idx, src_grp, dest_idx, dest_grp, option)
Input/Output parameters
The 1st parameter: [IN] String of host name or IP address of server.
The 2nd parameter: [IN] Index of source position register. This integer is index of PR of server.
The 3rd parameter: [IN] Group number of source position register. This integer is group number of server.
The 4th parameter: [IN] Index of destination position register. This integer is index of PR of client.
The 5th parameter: [IN] Group number of destination position register. This integer is group number of client.
The 6th parameter: [IN] This integer specifies function of this program.

Value	Description
0	Value and comment are got.
1	Value is got.
2	Comment is got.

Detail

If contents of the 2nd through the 5th parameters are wrong, following errors are posted.

Error	Cause
DTBR-014 Bad variable or register index (host name, index)	There isn't PR [(the 2nd parameter) src_idx] on server.
DTBR-015 Illegal group number (host name, index)	There isn't group "(the 3rd parameter) src_grp" on server.
VARS-024 Bad variable or register index	There isn't PR [(the 4th parameter) dest_idx] on client.
ROUT-026 Illegal group number	There isn't group "(the 5th parameter)dest_grp"on client.

NOTE

Position data of specified groups of client and server may be different in its components. For example, following points may differ.

- The number of axes
- Configuration of Cartesian position
- Type of axis (rotary or linear)

You may not be able to use position register that is got or set as it is. Please be careful to use transferred data.

NOTE

Even if Cartesian position data is got and stored to client's group that does not support Cartesian data (independent axis for example), Cartesian data is set.

NOTE

Position register cannot be transferred between R-30iB Plus/R-30iB Mate Plus controller and R-30iB/R-30iB Mate/R-30iA/R-30iA Mate controller.

Example

Execution of following program by client transfers position data of group 2 of PR [10] of host "SERVER" to client's group 1 of PR[20]. Comment is not acquired.

```
CALL RGETPREG ('SERVER', 10, 2, 20, 1)
```

38.4.4 Program to Set Position Register

Program name: RSETPREG

Overview

Client controller sets client's position register to server's position register. Position data of specified group only is set.

Syntax

RSETPREG (host_name, dest_idx, dest_grp, src_idx, src_grp, option)

Input/Output parameters

The 1st parameter: [IN] String of host name or IP address of server.

The 2nd parameter: [IN] Index of destination position register. This integer is index of PR of server.
The 3rd parameter: [IN] Group number of position data of destination PR of server. This is INTEGER.
The 4th parameter: [IN] Index of source position register. This integer is index of PR of client.
The 5th parameter: [IN] Group number of position data of source PR of client. This is INTEGER.
The 6th parameter: [IN] This integer specifies function of this program.

Value	Description
0	Position data and comment is set.
1	Position data is set.

Detail

If contents of the 2nd through 5th parameters are wrong, following errors are posted.

Error	Cause
DTBR-014 Bad variable or register index (host name, index)	There isn't PR [(the 2nd parameter) dest_idx] on server.
DTBR-015 Illegal group number(host name, index)	There isn't group"(the 3rd parameter) dest_grp" on server.
VARS-024 Bad variable or register index	There isn't PR [(the 4th parameter) src_idx] on client.
ROUT-026 Illegal group number	There isn't group "(the 5th parameter) src_grp" on client.

NOTE
Position data of specified groups of client and server may be different in its components. For example, following points may differ.
- The number of axes
- Configuration of Cartesian position
- Type of axis (rotary or linear)

You may not be able to use position register that is got or set as it is. Please be careful to use transferred data.

NOTE
Even if Cartesian position data is set to server's group that does not support Cartesian data (independent axis for example), Cartesian data is set.

NOTE
Position register cannot be transferred between R-30*i*B Plus/R-30*i*B Mate Plus controller and R-30*i*B/R-30*i*B Mate/R-30*i*A/R-30*i*A Mate controller.

Example

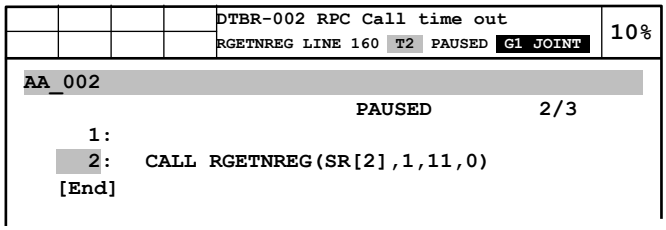
Execution of following program by client transfers position data of group 1 of PR [20] of client to group 2 of PR [10] of host "SERVER".

```
CALL RSETPREG ('SERVER', 10, 2, 20, 1, 0)
```

38.5 RECOVERY FROM ERROR

When error happened at execution of standard data transfer program, most of errors are following 2 cases.

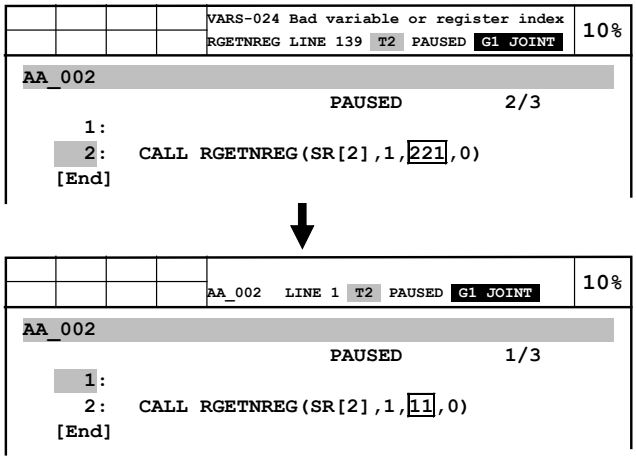
1 “DTBR-002 RPC Call timed out" is posted because communication failed or was not in time.



Please check if the controller (client) can always, stably communicate with specified host (server). You can perform “PING” in PING screen of Host Comm setup screen.
Please improve network traffics.
If standard data transfer program or built-in of this function are used very often, please lessen frequency.
If standard data transfer program or built-in of this function is executed consecutively, there should be interval. If there is already interval, please lengthen it.

When program is resumed without moving cursor, the same transfer is performed again.
If parameter and TCIP/IP setup has no problem and network had trouble, resume after removal of network problem can be successful.

2 Arguments were wrong.
In this case, arguments must be modified. In program edit screen, cursor is on the call of standard data transfer program. However, execution stopped inside of called program. Arguments were already handed to sub program (in figure below, RGETNREG).
Even if you modify arguments in program edit screen, resume from the same line doesn’t solve the problem. It is because value of arguments was already handed to sub program. If arguments are register or string register, value was handed at call. Even if you change value of register or string register after error happened, the change is not reflected to resume from the same line.
After modification of arguments, please move cursor to 1 line before CALL and resume from the line.



38.6 KAREL BUILT-IN

KAREL built-in routine (or function) is pre-defined function, which is built in KAREL language. This function provides built-in routines to transfer data between robots. KAREL program using the built-in enables followings.

- Sending and receiving register
- Sending and receiving position register
- Sending and receiving string register
- Read and write of I/O value
- Read of I/O simulation status
- Simulating and un-simulating I/O
- Read and write of comment of following data
  - Register
  - Position register
  - String register
  - I/O

Following table lists built-in routines provided by this function.

Built-in	Description
RGET_PORTCMT	gets I/O comment.
RGET_PORTSIM	gets I/O simulation status.
RGET_PORTVAL	gets I/O value.
RGET_PREGCMT	gets comment of position register.
RGET_REG	gets register.
RGET_REG_CMT	gets comment of register.
RGET_SREGCMT	gets comment of string register.
RGET_STR_REG	gets string register.
RNREG_RECV	transfers server's register to client's register.
RNREG_SEND	transfers client's register to server's register.
RPREG_RECV	transfers server's position register to client's position register.
RPREG_SEND	transfers client's position register to server's position register.
RSET_INT_REG	sets INTEGER to register.
RSET_PORTCMT	sets I/O comment.
RSET_PORTSIM	simulates I/O.
RSET_PORTVAL	sets I/O value.
RSET_PREGCMT	sets comment of position register.
RSET_REALREG	sets REAL value to register.
RSET_REG_CMT	sets comment of register.
RSET_SREGCMT	sets comment of string register.
RSET_STR_REG	sets string register.

38.6.1 RGET_PORTCMT Built-in ROUTINE

Purpose:

To allows a KAREL program to determine comment that is set for a specified logical port of remote host.

Syntax:

RGET_PORTCMT (host_port, port_type, port_no, comment_str, status)

Input/Output parameters

[in] host_port: STRING
[in] port_type: INTEGER

[in] port_no: INTEGER
[out] comment_str: STRING
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- "host_port" is IP address or host name of remote robot controller.
- "port_type" specifies the code for the type of port whose comment is returned. Codes are defined in kliotyps.kl.
- "port_no" specifies the port number whose comment is being returned.
- "comment_str" is returned with the comment for the specified port. This should be declared as a STRING with a length of at least 16 characters. Only 16 characters are returned even if server's comment is longer.
- "status" explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program gets comment of DO[10] of host RC1.

```
PROGRAM RGTPCMT
%COMMENT='RC1 DO[10] CMT'
%NOLOCKGROUP
%ENVIRONMENT RPCC
%INCLUDE kliotyps

VAR
  status  : INTEGER
  comment : STRING[16]
BEGIN
  RGET_PORTCMT('RC1', io_dout, 10, comment, status)
END RGTPCMT
```

38.6.2 RGET_PORTSIM Built-in ROUTINE

Purpose:

To get port simulation status from remote controller.

Syntax:

RGET_PORTSIM (host_port, port_type, port_no, simulated, status)
Input/Output parameters
[in]host_port: STRING
[in]port_type: INTEGER
[in]port_no: INTEGER
[out]simulated: BOOLEAN
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- "host_port" is IP address or host name of remote robot controller.
- "port_type" specifies the code for the type of port whose simulation status is returned. Codes are defined in kliotyps.kl.
- "port_no" specifies the port number whose simulation status is being returned.
- "simulated" returns TRUE if the port is being simulated, FALSE otherwise.
- "status" explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program gets simulation status of DI[8] of host RC1.

```
PROGRAM RGTPSIM
%COMMENT = 'RC1 DIN[8] SIM'
%NOLOCKGROUP
%ENVIRONMENT RPCC
%include kliotyps
VAR
  status  : INTEGER
  sim_stat: BOOLEAN
BEGIN
  RGET_PORTSIM('RC1', io_din, 8, sim_stat, status)
END RGTPSIM
```

38.6.3 RGET_PORTVAL Built-in ROUTINE

Purpose:

To allow KAREL program to determine the current value of a specified logical port of remote host.

Syntax:

RGET_PORTVAL (host_port, port_type, port_no, port_value, status)
Input/Output parameters
[in]host_port: STRING
[in]port_type: INTEGER
[in]port_no:INTEGER
[out]port_value:INTEGER
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- "host_port" is IP address or host name of remote robot controller.
- "port_type" specifies the code of port whose value is being returned. Codes are defined in kliotyps.kl.
- "port_no" specifies the port number whose value is returned.
- "port_value" is returned with the current value (status) of the specified port. For BOOLEAN port types (DIN for example), this will be 0 = OFF, or 1 = ON.
- "status" explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program gets value of RO [3] of host RC1.

```
PROGRAM RGTPVAL
%COMMENT='GET RC1 RO[3]'
%NOLOCKGROUP
%ENVIRONMENT RPCC
%include kliotyps
VAR
  status  : INTEGER
  port_val: INTEGER
BEGIN
  RGET_PORTVAL('RC1', io_rdo, 3, port_val, status)
END RGTPVAL
```

38.6.4 RGET_PREGCMT Built-in ROUTINE

Purpose:

To retrieve comment of a position register of remote host.

Syntax:

RGET_PREGCMT (host_port, register_no, comment_str, status)
Input/Output parameters
[in]host_port: STRING
[in]register_no: INTEGER
[out]comment_str : STRING
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- "host_port" is IP address or host name of remote robot controller.
- "register_no" specifies which position register to retrieve the comment from. The comment of the given position register is returned in parameter "comment_str".
- "comment_str" should be declared as a STRING with a length of at least 16 characters.
- "status" explains the status of the attempted operation. If not equal to 0, then an error occurred.

NOTE
Position register cannot be transferred between R-30*i*B Plus/R-30*i*B Mate Plus controller and R-30*i*A/R-30*i*A Mate/R-30*i*B/R-30*i*B Mate controller.

Example:

Following program gets comment of PR [5] of host RC1.

```
PROGRAM RGTPRCMT
%COMMENT='RC1 PR[5] CMT'
%NOLOCKGROUP
%ENVIRONMENT RPCC
VAR
  status  : INTEGER
  comment : STRING[16]
BEGIN
  RGET_PREGCMT('RC1', 5, comment, status)
END RGTPRCMT
```

38.6.5 RGET_REG Built-in ROUTINE

Purpose:

To get an INTEGER or REAL value from the specified register of remote host.

Syntax:

RGET_REG (host_port, register_no, real_flag, int_value, real_value, status)
Input/Output parameters
[in]host_port: STRING
[in]register_no: INTEGER
[out]real_flag: BOOLEAN
[out]int_value: INTEGER
[out]real_value: REAL
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- "host_port" is IP address or host name of remote robot controller.
- "register_no" specifies the register to get.
- "real_flag" is set to TRUE and "real_value" is set to the register content if the specified register has a real value. Otherwise, "real_flag" is set to FALSE and "int_value" is set to the contents of the register.
- "status" explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program gets value of R [2] of host RC1.

```
PROGRAM RGTNREG
%COMMENT='GET RC1 R[2]'
%NOLOCKGROUP
%ENVIRONMENT RPCC
VAR
  status   : INTEGER
  is_real  : BOOLEAN
  int_val  : INTEGER
  real_val : REAL
BEGIN
  RGET_REG('RC1', 2, is_real, int_val, real_val, status)
END RGTNREG
```

38.6.6 RGET_REG_CMT Built-in ROUTINE

Purpose:

To get comment from the specified register of remote host.

Syntax:

RGET_REG_CMT (host_port, register_no, comment_str, status)
Input/Output parameters
[in] host_port: STRING
[in] register_no: INTEGER
[out] comment_str: STRING
[out] status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- "host_port" is IP address or host name of remote robot controller.
- "register_no" specifies which register to retrieve the comment from. The comment of the given register is returned in "comment_str".
- "comment_str" should be declared as a STRING with a length of at least 16 characters.
- "status" explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program gets comment of R [2] of host RC1.

```
PROGRAM RGTNRCMT
%COMMENT='GET RC1 R[2] CMT'
%NOLOCKGROUP
%ENVIRONMENT RPCC
VAR
  status  : INTEGER
  comment : STRING[16]
BEGIN
  RGET_REG_CMT('RC1', 2, comment, status)
END RGTNRCMT
```

38.6.7 RGET_SREGCMT Built-in ROUTINE

Purpose:

To get comment from the specified string register of remote host.

Syntax:

RGET_SREGCMT (host_port, register_no, comment_str, status)
Input/Output parameters
[in]host_port: STRING
[in]register_no: INTEGER
[out]comment_str: STRING
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- "host_port" is IP address or host name of remote robot controller.
- "register_no" specifies which string register to retrieve the comment from.
- "comment_str" contains the comment of the specified string register.
- "comment_str" should be declared as a STRING with a length of at least 16 characters.
- "status" explains the status of the attempted operation. If not equal to 0, then an error occurred.

Examples:

Following program gets comment of SR [20] of host RC1.

```
PROGRAM RGTSRCMT
%NOLOCKGROUP
%COMMENT = 'GET CMT SR[20]'
%ENVIRONMENT RPCC
VAR
  status  : INTEGER
  comment : STRING[16]
BEGIN
  RGET_SREGCMT('RC1', 20, comment, status)
END RGTSRCMT
```

38.6.8 RGET_STR_REG Built-in ROUTINE

Purpose:

To get value from the specified string register of remote host.

Syntax:

RGET_STR_REG (host_port, register_no, value, status)
Input/Output parameters
[in] host_port: STRING

[in] register_no: INTEGER
[out] value : STRING
[out] status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- "host_port" is IP address or host name of remote robot controller.
- "register_no" specifies string register to get.
- "value" contains the value of the specified string register.
- "status" explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program gets value of SR [10] or host RC1.

```
PROGRAM RGTSREG
%NOLOCKGROUP
%ENVIRONMENT RPCC
%COMMENT = 'GET SR[10]'
VAR
  status   : INTEGER
  sreg_val : STRING[253]
BEGIN
  RGET_STR_REG('RC1', 10, sreg_val, status)
END RGTSREG
```

38.6.9 RNUMREG_RECV Built-in ROUTINE

Purpose:

To transfer server's (the other controller's) register to client's (the controller itself, which executes built-in) register.

Syntax:

RNUMREG_RECV (host_port, src_idx, dest_idx, option, status)
Input/Output parameters
[in]host_port: STRING
[in]src_idx : INTEGER
[in]dest_idx : INTEGER
[in]option : INTEGER
[in]status : INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- "host_port" is IP address or host name of remote robot controller.
- "src_idx" specifies index of register of server.
- "dest_idx" specifies index of register of client. Acquired data is stored in this register.
- "option" specifies function of this built-in.

Value	Description
0	Value and comment are got.
1	Value is got.
2	Comment is got.

- "status" explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program gets value and comment of R [3] of RC1 and set them to R [12].

```
PROGRAM RNREGRCV
%COMMENT='RC1 R3->12'
%NOLOCKGROUP
%ENVIRONMENT RPCC

VAR
  STATUS : INTEGER
BEGIN
  RNUMREG_RECV('RC1', 3, 12, 0, STATUS)
END RNREGRCV
```

38.6.10 RNUMREG_SEND Built-in ROUTINE

Purpose:

To transfer client's (the controller itself, which executes built-in) register to server's (the other controller's) register.

Syntax:

RNUMREG_SEND (host_port, dest_idx, src_idx, option, status)
Input/Output parameters
[in]host_port: STRING
[in]dest_idx : INTEGER
[in]src_idx : INTEGER
[in]option : INTEGER
[in]status : INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- "host_port" is IP address or host name of remote robot controller.
- "dest_idx" specifies index of register of server.
- "src_idx" specifies index of register of client. Data of this register is sent to server.
- "option" specifies function of this built-in.

Value	Descrition
0	Value and comment is set.
1	Value is set.
2	Comment is set.

- "status" explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program sends value of R [11] to R [2] of host RC1.

```
PROGRAM RNREGSND
%COMMENT='R11->RC1 R1'
%NOLOCKGROUP
%ENVIRONMENT RPCC

VAR
  STATUS : INTEGER
BEGIN
  RNUMREG_SEND('RC1', 2, 11, 1, STATUS)
END RNREGSND
```

38.6.11 RPOSREG_RECV Built-in ROUTINE

Purpose:

To transfer position register of specified group of server (the other controller's) to position register of specified group of client (the controller itself, which executes 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 IP address or host name of remote robot controller.
- "src_idx" specifies index of position register of server.
- "src_grp" specifies group number of server. Position data of specified group is transferred.
- "dest_idx" specifies index of position register of client. The specified position register stores acquired data.
- "dest_grp" specifies group number of client. Position data of specified group of PR ["dest_idx"] is changed to received data.
- "option" specifies function of this built-in.

Value	Description
0	Position data and comment are got.
1	Position data is got.
2	Comment is got.

- "status" explains the status of the attempted operation. If not equal to 0, then an error occurred.

NOTE
Position register cannot be transferred between R-30*i*B Plus/R-30*i*B Mate Plus controller and R-30*i*A/R-30*i*A Mate/ R-30*i*B/R-30*i*B Mate controller.

Example:

Following example program performs following transfer. Comment is transferred, too.
From: Group 1 of PR [5] of host RC1
To: Group 2 of PR [6]

```
PROGRAM RPREGRCV
%COMMENT='RC1 PR5G1->PR6G2'
%NOLOCKGROUP
%ENVIRONMENT RPCC

VAR
  STATUS : INTEGER
BEGIN
  RPOSREG_RECV('RC1', 5, 1, 6, 2, 0, STATUS)
END RPREGRCV
```

38.6.12 RPOSREG_SEND Built-in ROUTINE

Purpose:

To transfer position register of specified group of client to position register of specified group of server.

Syntax:

RPOSREG_SEND (host_port, dest_idx, dest_grp, src_idx, src_grp, option, status)
Input/Output parameters
[in]host_port: STRING
[in]dest_idx: INTEGER
[in]dest_grp: INTEGER
[in]src_idx: INTEGER
[in]src_grp: INTEGER
[in]option: INTEGER
[in]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- "host_port" is IP address or host name of remote robot controller.
- "dest_idx" specifies index of position register of server.
- "dest_grp" specifies group number of server. Position data of specified group is changed by sent data from client.
- "src_idx" specifies index of position register of client. Position data of group "src_grp" of PR ["src_idx"] is sent to server.
- "src_grp" specifies group number of client.
- "option" specifies function of this built-in.

Value	Description
0	Position data and comment are set.
1	Position data is set.

- status explains the status of the attempted operation. If not equal to 0, then an error occurred.

NOTE
Position register cannot be transferred between R-30*i*B Plus/R-30*i*B Mate Plus controller and R-30*i*A/R-30*i*A Mate/R-30*i*B/R-30*i*B Mate controller.

Example:

Following example program performs following transfer. Comment is transferred, too.
From: Group 2 of PR [7] of client
To: Group1 of PR [8] of host RC1

```
PROGRAM RPREGSND
%COMMENT='PR7G2->RC1 PR8G1'
%NOLOCKGROUP
%ENVIRONMENT RPCC

VAR
  STATUS : INTEGER
BEGIN
  RPOSREG_SEND('RC1', 8, 1, 7, 2, 0, STATUS)
END RPREGSND
```

38.6.13 RSET_INT_REG Built-in ROUTINE

Purpose:

To store an INTEGER value in the specified register of remote host.

Syntax:

RSET_INT_REG (host_port, register_no, int_value, status)
Input/Output parameters
[in]host_port: STRING
[in]register_no: INTEGER
[in]int_value: INTEGER
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- "host_port" is IP address or host name of remote robot controller.
- "register_no" specifies the register into which "int_value" will be stored.
- "status" explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program sets 100 to R[20] of host RC1.

```
PROGRAM RSTNRI
%COMMENT = 'RC R[20]=100'
%NOLOCKGROUP
%ENVIRONMENT RPCC
VAR
  status : INTEGER
BEGIN
  RSET_INT_REG('RC1', 20, 100, status)
END RSTNRI
```

38.6.14 RSET_PORTCMT Built-in ROUTINE

Purpose:

To allows a KAREL program to set comment of specified logical port of remote host.

Syntax:

RSET_PORTCMT (host_port, port_type, port_no, comment_str, status)
Input/Output parameters
[in]host_port: STRING
[in]port_type: INTEGER
[in]port_no: INTEGER
[in]comment_str: STRING
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- "host_port" is IP address or host name of remote robot controller.
- "port_type" specifies the code of port whose comment is being set. Codes are defined in kliotyps.kl.
- "port_no "specifies the port number whose comment is being set.
- "comment_str" is a string whose value is the comment for the specified port. This must not be over 16 characters long.
- "status" explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program sets comment to DO [10] of host RC1.

```
PROGRAM RSTPCMT
%COMMENT ='RC1 DO[10] cmt'
%NOLOCKGROUP
%ENVIRONMENT RPCC
%INCLUDE kliotyps
VAR
  status : INTEGER
BEGIN
  RSET_PORTCMT('RC1', io_dout, 10, 'RC1 DOUT[10]', status)
END RSTPCMT
```

38.6.15 RSET_PORTSIM Built-in ROUTINE

Purpose:

To set port simulated on remote host.

Syntax:

RSET_PORTSIM (host_port, port_type, port_no, value, status)
Input/Output parameters
[in]host_port: STRING
[in]port_type: INTEGER
[in]port_no: INTEGER
[in]value: INTEGER
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- "host_port" is IP address or host name of remote robot controller.
- "port_type" specifies the code for the type of port to be simulated. Codes are defined in kliotyps.kl.
- "port_no" specifies port number to be simulated.
- "value" specifies the initial value to set.
- "status" explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program sets GO [3] of RC1 simulated and set its initial simulated value to 5.

```
PROGRAM RSTPSIM
%COMMENT ='RC1 SIM GO[3], 5'
%NOLOCKGROUP
%ENVIRONMENT RPCC
%INCLUDE kliotyps

VAR
  status  : INTEGER
BEGIN
  RSET_PORTSIM('RC1', io_gpout, 3, 5, status)
END RSTPSIM
```

38.6.16 RSET_PORTVAL Built-in ROUTINE

Purpose:

To allow KAREL program to set a specified output (or simulated input) for a specified logical port.

Syntax:

RSET_PORTVAL (host_port, port_type, port_no, port_value, status)
Input/Output parameters
[in]host_port: STRING
[in]port_type: INTEGER
[in]port_no: INTEGER
[in]port_value: INTEGER
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- "host_port" is IP address or host name of remote robot controller.
- "port_type" specifies the code of port whose value is being set. Codes are defined in kliotyps.kl.
- "port_no" specifies the port number whose value is being set.
- "value" indicates the value to be assigned to a specified port. If the port_type is BOOLEAN (DOUT for example), this should be 0 = OFF, or 1 = ON. This field can be used to set input ports if the port is simulated.
- "status" explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program sets DO [3] of host RC1 to ON.

```
PROGRAM RSTPVAL
%COMMENT ='RC1 DO[3]=ON'
%NOLOCKGROUP
%ENVIRONMENT RPCC
%INCLUDE kliotyps

VAR
  status  : INTEGER
BEGIN
  RSET_PORTVAL('RC1', io_dout, 3, 1, status)
END RSTPVAL
```

38.6.17 RSET_PREGCMT Built-in ROUTINE

Purpose:

To set comment of a position register of remote host.

Syntax:

RSET_PREGCMT (host_port, register_no, comment_str, status)
Input/Output parameters
[in]host_port: STRING
[in]register_no: INTEGER
[in]comment_str : STRING
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- "host_port" is IP address or host name of remote robot controller.
- "register_no" specifies position register of remote robot controller.
- "comment_str" is comment to be set. If "comment_str" exceeds 16 characters, it is truncated.
- "status" explains the status of the attempted operation. If not equal to 0, then an error occurred.

NOTE
Position register cannot be transferred between R-30*i*B Plus/R-30*i*B Mate Plus controller and R-30*i*A/R-30*i*A Mate/R-30*i*B/R-30*i*B Mate controller.

Example:

Following program sets comment "RC1 PR[5]" to PR [5] of host RC1.

```
PROGRAM RSTPRCMT
%COMMENT ='RC1 PR[5] CMT'
%NOLOCKGROUP
%ENVIRONMENT RPCC
VAR
  status  : INTEGER
BEGIN
  RSET_PREGCMT('RC1', 5, 'RC1 PR[5]', status)
END RSTPRCMT
```

38.6.18 RSET_REALREG Built-in ROUTINE

Purpose:

To store a REAL value in the specified register of remote host.

Syntax:

RSET_REALREG (host_port, register_no, real_value, status)
Input/Output parameters
[in] host_port: STRING
[in] register_no: INTEGER
[in] real_value: REAL
[out] status: INTEGER
%ENVIRONMENT RPCC

Detail:

- "host_port" is IP address or host name of remote robot controller.
- "register_no" specifies the register into which "real_value" will be stored.
- "status" explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program sets -12.3 to R [20] of RC1.

```
PROGRAM RSTNRR
%COMMENT = 'RC1 R[20]=-12.3'
%NOLOCKGROUP
%ENVIRONMENT RPCC
VAR
  status : INTEGER
BEGIN
  RSET_REALREG('RC1', 20, -12.3, status)
END RSTNRR
```

38.6.19 RSET_REG_CMT Built-in ROUTINE

Purpose:

To set comment of numeric register of remote host.

Syntax:

RSET_REG_CMT (host_port, register_no, comment_str, status)
Input/Output parameters
[in] host_port: STRING
[in] register_no: INTEGER
[in] comment_str: STRING
[out] status: INTEGER
%ENVIRONMENT RPCC

Detail:

- "host_port" is IP address or host name of remote robot controller.
- "register_no" specifies which register to set the comments to.
- "comment_str" represents the data which is to be used to set the comment of the given register. If "comment_str" exceeds more than 16 characters, the built-in will truncate the string.
- "status" explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program set comment "RC1 R[15]" to R [15] of host RC1.

```
PROGRAM RSTNRCMT
%COMMENT ='RC1 R[15] CMT'
%NOLOCKGROUP
%ENVIRONMENT RPCC
VAR
  status  : INTEGER
BEGIN
  RSET_REG_CMT('RC1', 15, 'RC1 R[15]', status)
END RSTNRCMT
```

38.6.20 RSET_SREGCMT Built-in ROUTINE

Purpose:

Sets the comment for the specified string register of remote robot controller.

Syntax:

RSET_SREGCMT (host_port, register_no, comment_str, status)
Input/Output parameters
[in]host_port: STRING
[in]register_no: INTEGER
[in]comment_str : STRING
[out]status: INTEGER
%ENVIRONMENT Group:RPCC

Detail:

- "host_port" is IP address or host name of remote robot controller.
- register_no specifies string register to set.
- comment_str contains the comment to set to the specified string register. If the comment_str exceeds more than 16 characters, the built-in will truncate the string.
- status explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program sets comment “RC1_STRING_REG_3” to SR[3] of host RC1.

```
PROGRAM RSTSRCMT
%COMMENT ='RC1 SR[3] CMT'
%NOLOCKGROUP
%ENVIRONMENT RPCC
VAR
  status  : INTEGER
BEGIN
  RSET_SREGCMT('RC1', 3, 'RC1_STRING_REG_3', status)
END RSTSRCMT
```

38.6.21 RSET_STR_REG Built-in ROUTINE

Purpose:

To set specified value for the specified string register of remote host.

Syntax:

RSET_STR_REG (host_port, register_no, value, status)
Input/Output parameters
[in] host_port: STRING
[in] register_no: INTEGER
[in] value : STRING
[out] status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- “host_port” is IP address or host name of remote robot controller.
- “register_no” specifies string register to set.
- “value” contains the value to set to the specified string register.
- “status” explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program sets SR [4] of host RC1.

```
PROGRAM RSTSREG
%COMMENT ='Set RC1 SR[4]'
%NOLOCKGROUP
%ENVIRONMENT rpcc
CONST
  ERR_SUCCESS = 0
VAR
  status    : INTEGER
BEGIN
  RSET_STR_REG('RC1', 4, 'RC1 of SR[4] was set by remote host', status)
END RSTSREG
```

38.7 TIME OUT AND RETRY

When communication doesn’t complete in time because of some kind of problem like network trouble, this function disconnect connection and tries to communicate again. After 2 times retry, this function gives up communication. Standard data transfer program automatically posts "DTBR-002 RPC Call timed out". Status of built-in of this function will be set to corresponding value.
Timeout value is 2 seconds by default. The number of retry is 2. When communication completely failed, this function gives up in about 6 seconds.

Time out value is stored in $DTBR_CFG.$RPC_TIMEOUT in seconds. Minimum timeout value is 1 second regardless of value of the system variable.
If controller cannot recognize host name at all, communication can be given up before time out happens.
Detection of communication error takes more time than usual if timeout value is longer than default.
This function does not give up communication and retry even if program is paused or aborted. If timeout time longer than usual, time you have to wait before starting next communication becomes longer.

38.8 LIMITATIONS

A) Performance of communication is not guaranteed. It depends network traffic and load of CPU of controller.
B) This function affects to performance of other functions that uses Ethernet and vise versa.
C) Request from multiple tasks (programs) doesn't cause parallel execution of data transfer.
They are actually processed sequentially.
D) In case that the controller to communicate with (server) is R-30*i*Aor R-30*i*A Mate controller. series of the system software of that controller must be 7DA5 or later.
E) You cannot restrict data transfer by the other functions, for example password function.
F) A robot controller cannot send request to itself by this function.
G) Transfer of position register pays attention to the number of axis or extended axis so that data to be set to destination does not exceeds axes destination controller has. However, the other elements of position data are not considered.
For example, followings are not considered.
- Whether robot can have Cartesian data or not.
- Axis limit
- Configuration of position available and range of turn number.
- Type of axis (rotary or linear)

H) Pause or force abort of program does not stop communication once it started.
Next request should be issued after completion of previous one.
I) Position register cannot be transferred between R-30iB Plus/R-30iB Mate Plus controller and R-30iB/R-30iB Mate/R-30iA/R-30iA Mate controller.

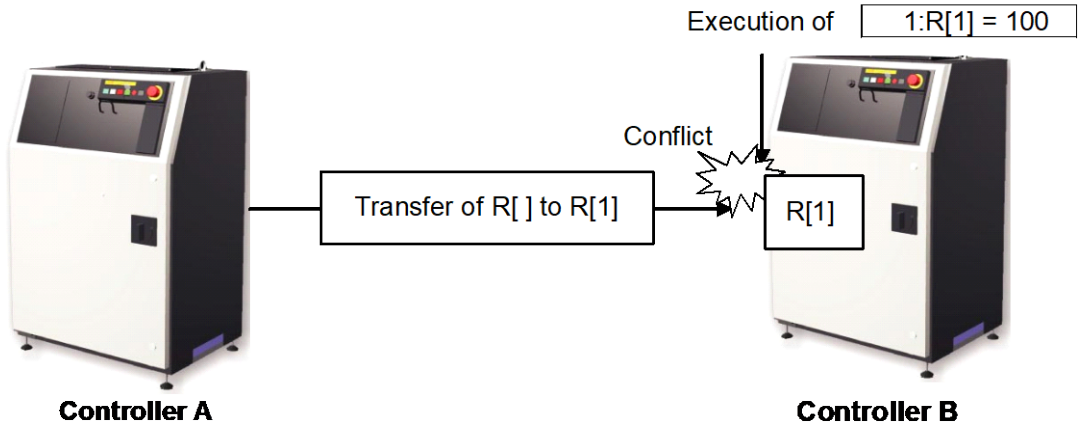
38.9 CAUTION

A) This function can not be used instead of hardwired I/O, field network.
This function offers ability to read/write I/O value of remote controller. However, hardwired I/O and field network I/O is always far faster.
B) Cyclic execution increases load of network and CPU of controller. Consecutive or cyclic execution of data transfer should be avoided.
C) Caution of transfer of position register
Content of position register depends on type of robots and configuration of position.
Please be careful for use of transferred position register.
Especially, you should be very careful when you use transferred position register as destination point of motion statement.
Robot of server and client are not always robot of the same type. Interpretation of position register depends on currently selected tool and user frame.

38.10 CONFLICT BETWEEN DATA WRITES

When you transfer register or position register, please note that only one controller changes a register or position register. This is also true to string register and I/O if you handle them by built-in.

If both server and client write to the same data, they overwrite value written by the other controller. This might look as if data transfer was not working properly.
Especially, if timing of write by client and server is very close, the earlier one may not be recognizable.



When client writes register, position register or string register of server, value and comment is acquired first. Then the client overwrites necessary part of acquired data and request server to write the updated value as a whole.
If server writes slightly before client, value or comment may be changed back to those were got by client.

38.11 TROUBLE SHOOTING

"DTBR-002 RPC Call timed out" is posted.

Please check if the controller (client) can always, stably communicate with specified host (server). You can perform "PING" in PING screen of "Host Comm" setup screen.
Please improve network traffics.
If standard data transfer program or built-in of this function are used very often, please lessen frequency of use.
If standard data transfer program or built-in of this function is executed consecutively, there should be interval. If there is already interval, please lengthen it.

Modification of arguments of program is not reflected after resume.

If you modified arguments of standard data transfer program, please move cursor to one line before the CALL and resume the program from the line.

Communication takes time or speed is unstable.

Speed of data transfer by this function is not guaranteed.
Performance depends on network traffic, load of CPU of client and server.
Please improve network traffics.
Please improve timing of data transfer. For example, data transfer that is not needed during motion of robots can be done when robot is not moving.

"INTP-320 (program name, line number) Undefined built-in" is posted.

If this symptom happens when standard data transfer program or built-in of this function is executed, please confirm whether this function is ordered or not. This function is optional (J740).

Time out does not happen. Program keeps running.

Pause or abort of program during data transfer does not stop the transfer once started. It tries to transfer data until the last retry fails.

The next request of data transfer is not processed until previous data transfer completes (fails).

Suppose following procedure happened.
1 Ethernet cable is not plugged in properly.
2 Standard data transfer program, RGETNREG is called but the user aborted the program immediately by FCTN menu.
3 Run the same program again and RGETNREG is called immediately.

In this case, the 2nd RGETNREG times out after about 2 times longer times than usual has passed.
By default setting, it is about 12 seconds.
Once communication is started, waiting for completion or error would be better lather than forcing abort.

Time out error happens just after program resume.

Program pause does not stop data transfer once it started. If timeout occurs while program is paused, timeout error is posted at resume.

Value of axis of joint position of PR[ ] is not transferred properly.

If type of axis (rotary or linear) is different between client's and server's corresponding axes, transferred value is not interpreted in the same way. It is not treated as the same value.

"INTP-311 (program name, line number) Uninitialized data is used" is posted when transferred position register is used.

1 Please confirm source data is initialized.
2 The number of total axes or extended axes may be different.
If source data does not have enough axes to set all axis data to destination data, axes that cannot be set because of lack of data becomes un-initialized. In this case, proper data should be set before use.
Suppose group 1 of host RC1 has an extended axis and group 1 of RC2 have no extended axis.
If Cartesian position of group1 of RC2 is set to group1 of RC1, E1 is set to un-initialized value.
RC1 have to set proper value by it self. If 0 is appropriate, PR[1,7]=0 can be used.

Position data of sent/received position register is partially un-initialized.

Please refer to previous item.
Position register screen doesn't display "R" unless data of all groups are initialized.
Standard transfer program and built-in transfers only specified group.

"VARS-037 Position register is locked" is posted.

Please check if position register is locked on server or client.

"HRTL-047 Address family not supported" is posted.

Please check if specified host is client itself.
Client cannot operate data of client itself by this function.

Data is sent to/received from a controller different from specified one.

If you didn't cycle power after change of IP address and host name, please cycle power.
If host name includes underscore, don't use it.

"HRTL-049 Can't assign requested address" is posted.
"HOST-108 Internet address not found" is posted.

Please check if specified host name or IP address is correct.
If you didn't cycle power after change of IP address and host name, please cycle power.
Please check if the controller (client) can always, stably communicate with specified host (server). You can perform "PING" in PING screen of "Host Comm" setup screen.

39 TOUCH SENSING

Touch sensing allows the robot to change a path automatically to compensate for object displacement. Touch sensing consists of two phases: Search Motion phase and Touch Offset phase.

To use this function, Touch sensor function (A05B-2600-J536) is required. This function was originally developed for ArcTool. Therefore, explanations and figures on this chapter are close to arc welding. But, it is also possible to use this function for HandlingTool.

Search Motion

Search Motion can detect the current position of workpiece.

- Move touch sensor part (on arc welding robot, the top of wire = TCP) toward the workpiece using pre-defined robot motion, speed, and direction.
- Use an input signal to indicate that the robot has come into the contact with the object.
- Store the found location of the workpiece, or position offset information, in position registers.

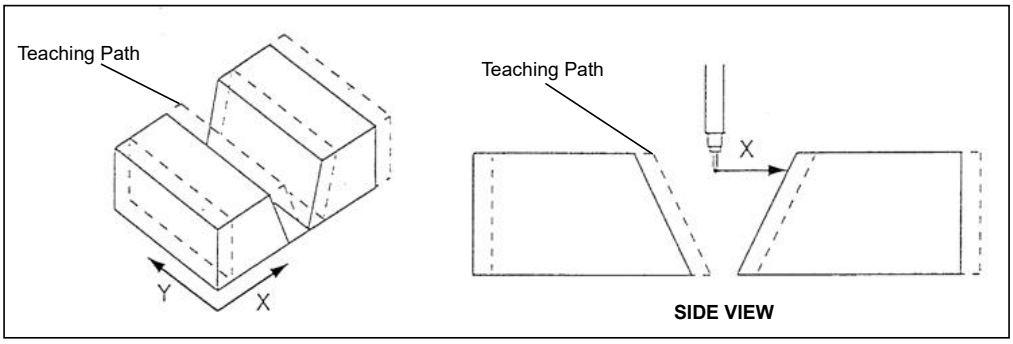


Fig. 39(a) Search motion for detecting shift of X direction

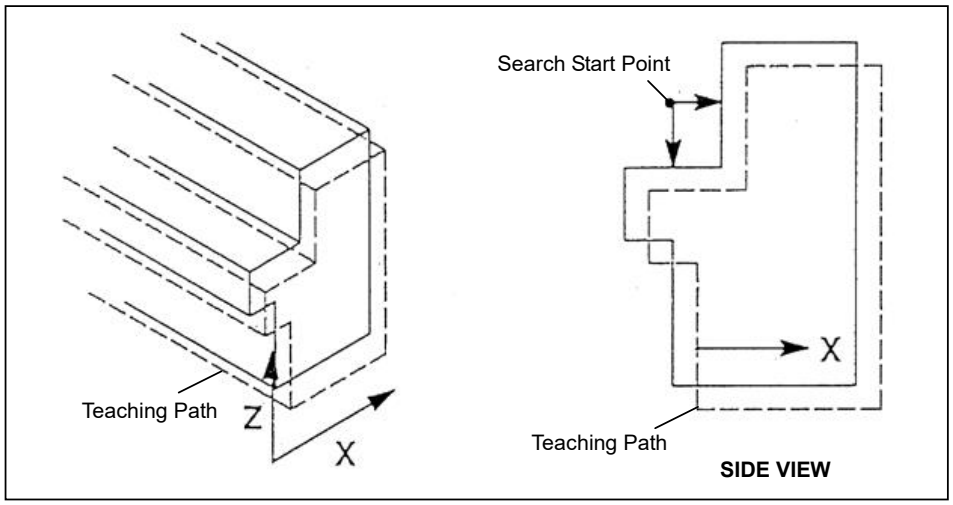


Fig. 39(b) Search motion for detecting shift of X, Z directions

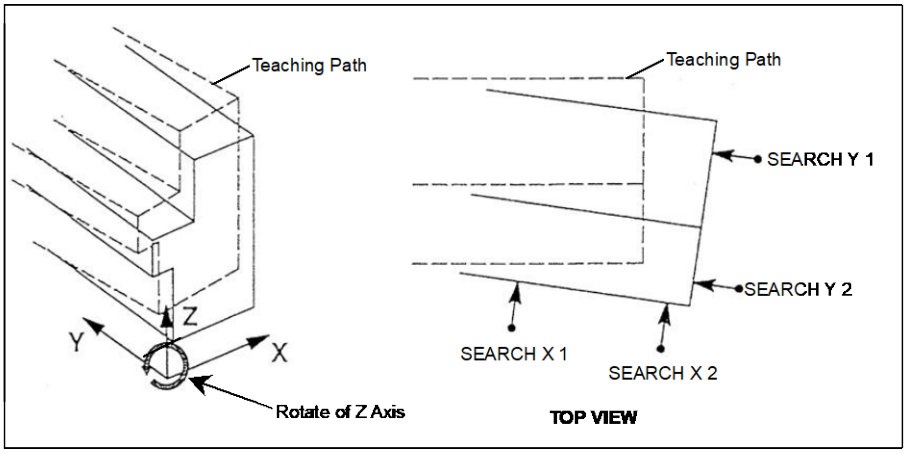


Fig. 39(c) Search motion for detecting shift of X, Y directions and Z rotation

Touch Offset

Shift one or more teaching positions in your welding program by using the stored position the stored position offset information.

Flow of Touch Sensing

To use touch sensing you must:

- Set up the robot Tool Center Point (TCP) properly.
- Set up touch sensing hardware. The hardware monitors an input signal to determine when the robot comes into contact with the object. There are some weld equipment models which install touch sensing hardware as standard such as Lincoln weld equipment.

After finishing the above preparations, please perform the following setups by referring to this chapter.

- **Assignment of Touch Sensing I/O (Section 39.1)**
  Assign I/O to enable and use the electrical interface circuit.
- **Setup of Touch Frame (Section 39.2)**
  Create Touch Sensing Frame to decide search direction during search motion.
- **Touch Schedule (Section 39.4)**
  Set the method to approach the workpiece (Search Pattern) and define the storing type of obtained position information (absolute position or position offset).
- **Touch Sensing Programming (Section 39.5)**
  Create touch sensing program by using Search instructions and Touch Offset instructions.
- **Execution of Touch Sensing Program (Section 39.6)**
  Execute Search Motion with Master Flag ON and obtain the master position. After that, set Master Flag OFF.

```
TOUCH_SENSING_1                                    13/13
     1:J  P[1] 50% FINE
     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 Home Position
Start Search Motion. Use Touch Sch 3, PR[3]
Teach a search start position
Search to Y direction
Teach an another search start position
Search to X direction
Finish Search Motion.
(If needed) Teach an intermediate point
Following points will be offset by PR[3]
P[7] is offset by PR[3]

P[8] is offset by PR[3]

Finish offset of positions

Fig. 39(d) Example program including touch sensing routine

39.1 ASSIGNMENT OF TOUCH SENSING I/O

To use touch sensing you must assign the
- Input signal that the touch sensing circuit monitors to indicate when the robot has reached the object.
- Output signal that enables and disables the touch sensing circuit.

You must wire the necessary connections for the input and output signals to be used for touch sensing. The wire stick detection circuit on the process I/O board also can be used for touch sensing.

Input Signal (Touch Detect Signal)

The touch sensing input signal indicating contact with a part is monitored by the touch sensing circuit. When the input is received, the current robot position is stored in a position register.

RI, DI, WI WS can be used as the touch sensing input signal:

REFERENCE: You can also set up touch sensing to monitor the condition of any RO or DO signal as an input signal. When the selected output turns on during a touch sensing routine, the controller reads this as a received input signal.

Output Signal (Touch Sensing Enable/Disable Command Signal)

RO, DO, WO, WSE can be used to enable the touch sensing circuit:

Assigning the Touch Sensing Inputs and Outputs

You must assign touch sensing inputs and outputs to match the hardware interface at your site. This involves assigning both input and output type and port number. Please refer to Procedure 39-1.

After assignment of Touch Sensing I/O is finished, please check whether touch sensing I/O can properly work or not. Please refer to Procedure 39-2.

NOTE

To use touch sensor circuit on welding power supply for Touch Sensing function, specify the port number of I/O of Touch Sensing signal on Touch sensing setup screen after confirming them on Weld I/O screen.

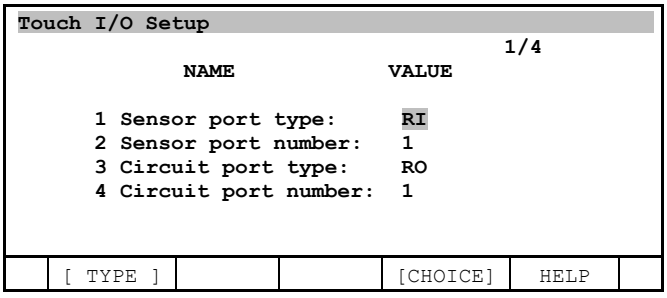
Procedure 39-1 Assigning Touch Sensing Input and Output Signals

Condition

- (Only for multi group system) Motion Group No. is selected for assignment of Touch Sensing I/O.

Step

1 Press the [MENU] key.
2 Select SETUP.
3 Press the F1, [TYPE] and then select Touch I/O. The following screen will be displayed.



4 Assign Sensor (input) and Circuit (output) types as follows:
  a Move the cursor to the line you want to assign.
  b Press the F4, CHOICE.

c Move the cursor to the desired input/output type.
d Press the [ENTER] key.

5 Assign Sensor and Circuit number:
a Move the cursor to the line you want to assign.
b Type the value and press the [ENTER] key.

NOTE
This screen shows the settings of the currently selected motion group. To view the settings of another motion group, change the motion group by selecting an auxiliary menu item CHANGE GROUP.

NOTE
After deciding Touch Sensing I/O, please enter the comment which indicates that this signal is used for Touch Sensing. This operation should be performed on each I/O signal screen.

NOTE
The ArcTool software checks the validity of the port type and port number when running your program that includes touch sensing. If the port type or number is invalid, the system displays an I/O invalid error message.

Procedure 39-2 Test for Touch Sensing I/O

Condition

- Workpiece that can be detected by Touch Sensing is placed.
- Assignment of Touch Sensing I/O is finished.

Step

1 Move the robot near the workpiece by jog operation.
2 Display the Output Signal screen for Touch Sensing Output signal set by Procedure 39-1 (If port type is DO, display Digital Output Signal screen). Move the cursor on the line of the port number for Touch Sensing Output signal and then press the F4, ON key. The signal is output and Touch Sensing circuit becomes Enabled.
3 Jog the robot and touch the touch sensor part to the workpiece. In this operation, you must set low override value.
4 Display the Input Signal screen for Touch Sensing Detect signal set by Procedure 39-1 (If port type is DI, display Digital Input Signal screen). Next, check the status of port number for Touch Sensing Input signal. If the status is ON, Touch Sensing will correctly work. If the status is OFF even when the touch sensor part touches to the workpiece, please check hardware connection or assignment of Touch Sensing I/O by Procedure 39-1.
5 Move the cursor on the output signal set ON by Step 2 and then press the F5, OFF.

⚠WARNING
If Touch Sensing Output signal is set to ON by Step 2, current flow is generated in Touch Sensing Circuit. Therefore, do not touch the touch sensor part (on ArcTool, the top of wire). Additionally, do not forget to return the output signal to OFF.

39.2 SETUP OF TOUCH FRAME

A touch frame determines the motion direction of the robot TCP (touch sensor part). A touch frame is defined by three points. The first point defines the origin, or starting point. The second point defines the positive X direction of the touch frame. The third point defines the positive X-Y plane. Z direction is created by defined X, Y axes.
Fig. 39.3 shows a touch frame and how it is used in a touch sensing program. The orientation of the touch frame to the object is arbitrary in Fig. 39.3. The positive X axis could be aligned with the current z direction. This would re-define positive z to be in the opposite direction of the current positive x direction.

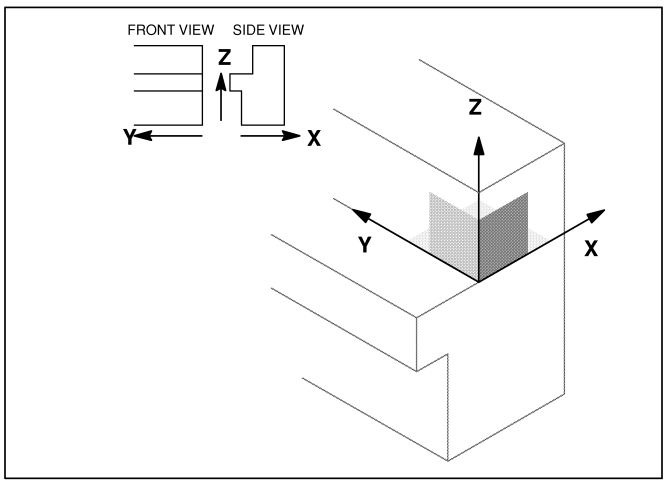


Fig. 39.2 Touch frame

NOTE
You can set up a maximum of 32 touch frames for each robot group. Touch frames are set up using the touch frame screen in the Setup menu.

NOTE
You must define a touch frame before you perform a search motion in a program.

There are two ways to define touch frames: The teaching method and the direct entry method. The teaching method defines the touch frame by recording three points. The direct entry method defines the touch frame by the rotation angle value you enter in the touch sense setup screen.
Normally, Reference Group in Touch frame setup screen should be the same group number with Robot Group (Touch frame relates to the WORLD frame of the robot).
In coordinated motion system, if you want to define touch frame to the coordinated (reference) group, you should set leader group number to Reference Group and follower group number to Robot Group.
Table 39.2 lists and describes the items you must set to define the touch frame. About more detail, refer to Section 39.10.

Table 39.2 Touch frame setup items

ITEM	DESCRIPTION
Frame Number	This item specifies the number of the touch frame you want to define.
Reference Group	This item specifies the reference group you want to make the relationship to Touch Frame. Normally, you should set the same group number of below Robot Group item. If you want to perform Coordinated Touch Sensing, setup the item by referring to Section 39.10.
Robot Group	This item specifies a motion group for which the touch frame is used. You cannot select non-robot group (Ex: positioner group). In single group system, always 1 should be set.

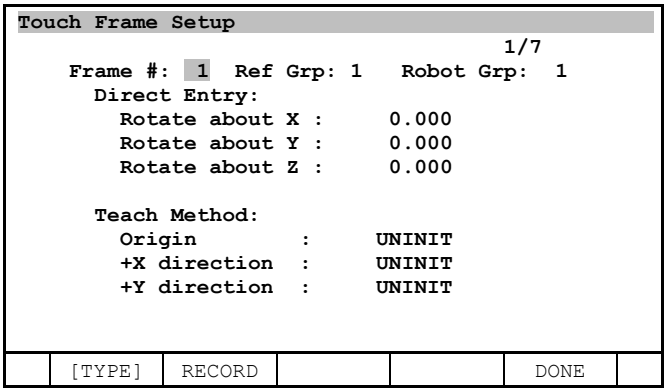
ITEM	DESCRIPTION
Direct Entry (Procedure 39-4)	
Rotate about X	This item specifies the rotation about X for the touch frame.
Rotate about Y	This item specifies the rotation about Y for touch frame.
Rotate about Z	This item specifies the rotation about Z for touch frame.
Teach Method (Procedure 39-3)	
Origin	This item allows you to record the origin of the touch frame.
+X direction	This item allows you to define the +X direction of the touch frame.
+Y direction	This item allows you to define the +Y direction of the touch frame.

Use Procedure 39-3 to define your touch frame by using the teaching method. Use Procedure 39-4 to define your touch frame by using the direct entry method.

Procedure 39-3 Setup of Touch Frame Using Teaching Method

Step

1 Press the [MENU] key.
2 Select Setup.
3 Press the F1, [TYPE].
4 Select Touch Frame. The following screen will be displayed.

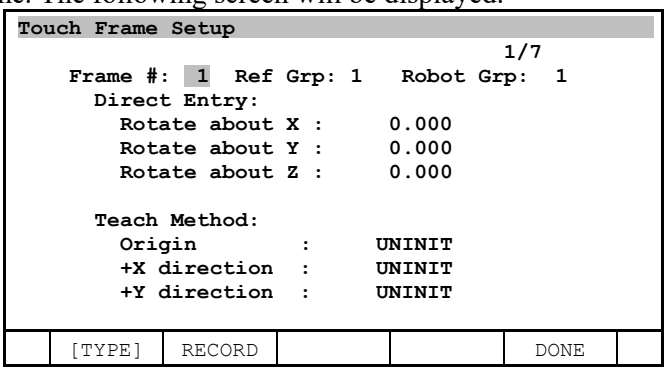


5 Move the cursor to Robot Grp. Enter the robot group number and press the [ENTER] key.
6 Move the cursor to Frame Number. Enter the number of the frame to define and press the [ENTER] key.
7 Move the cursor to Ref Grp. Normally, enter the group number which is the same as Robot Group item, and then press the [ENTER] key. If you want to perform Coordinated Touch Sensing, enter the group number of leader group.
8 Define the origin point of the Touch Frame.
   a Move the cursor to Origin.
   b Jog the Robot TCP to the desired starting point (origin).
   c Press the F2, RECORD. (Then, UNINIT will be changed to RECORDED.)
9 Define the +X direction.
   a Move the cursor to +X direction.
   b Jog the robot TCP to a point along the +X axis of the touch frame.
   c Press the 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 the F2, RECORD.
11 Press the F5, DONE to complete the definition of the frame. (Then, RECORDED will be changed to USED.)

Procedure 39-4 Setup of Touch Frame Using Direct Entry Method

Step

1 Press the [MENU] key.
2 Select SETUP.
3 Press the F1, [TYPE].
4 Select Touch Frame. The following screen will be displayed.

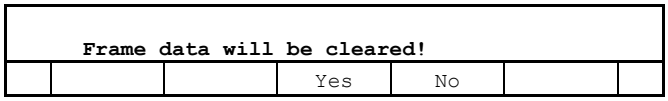


5 Move the cursor to Robot Grp. Enter the robot group number and press the [ENTER] key.
6 Move the cursor to Frame Number. Enter the number of the frame to define and press the [ENTER] key.
7 Move the cursor to Ref Grp. Normally, enter the group number which is the same as Robot Group item, and then press the [ENTER] key. If you want to perform Coordinated Touch Sensing, enter the group number of leader group.
8 Define the rotation angle about X.
   a Move the cursor to Rotate about X.
   b Enter the value (in degrees).
9 Define the rotation angle about Y.
   a Move the cursor to Rotate about Y.
   b Enter the value (in degrees).
10 Define the rotation angle about Z.
   a Move the cursor to Rotate about Z.
   b Enter the value (in degrees).
11 Press the F5, DONE to complete the definition of the frame. (UNINIT will be changed to USED.)

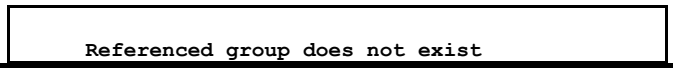
⚠CAUTION
Do not forget DONE operation at the last timing of Procedure 39-3 and 39-4. If you forget this operation, touch frame can not be defined.

NOTE

If you change the value of Ref Grp for an initialized frame at Step 7 of Procedure 39-3 and 39-4, the following warning message will be displayed:



If you press the F3, Yes, the frame data will be reinitialized.
If you set different number from Robot Group number to Ref Grp, but no coordinated pair calibration is not finished for the group number, the value of Ref Grp will return to the original value and the following warning message will be displayed:



39.3 SEARCH PATTERN

Search Pattern and Pattern Type determine the offset direction by Touch Sensing function. You should select Search Pattern and Pattern Type after checking the expected direction for shift of workpiece.
For detecting horizontal shift of workpiece, one search motion per 1 search direction is necessary. If the detection of rotational shift of workpiece is also requested, two search motions per 1 search direction are necessary.

Search Pattern

Four types of search patterns are available. You should select one of following search patterns by the shape of workpiece.

- Simple search
- Fillet/lap search
- V-Groove search
- Outside/inside diameter search

By selected search pattern, the position data type stored to a position register is defined. The type becomes Absolute position or offset value.
If you select Fillet/lap search, you should also select search type by the possibility of shift directions of workpiece.

Simple Search

For a simple search, a two-dimensional search is executed to find the actual location of one position on a workpiece. A simple search stores the found absolute position (x, y, z, w, p, r) into a position register PR[ ]. Once completed, the robot is programmed to move to the position stored in that position register.

Simple search requires:

- Surfaces being searched are perpendicular to each other.
- Searches to be done in two different directions.
- The second search motion to be performed with the desired torch angle.

The first search defines the positional information for that search direction only (x, for example). The second search defines the other direction positional information (z, for example). The starting position of the second search defines the remaining positional information, (y, w, p, r, for example) that determines the torch angle for welding and, in this case, the Y value.

Simple search is typically used to find the starting point of a weld path that uses the Thru-Arc Seam Tracking (TAST) option or Automatic Voltage Control (AVC) Tracking option.

A two-dimensional search is programmed in the software as the only valid search pattern type when a simple search is used. Changing the search pattern type has no effect.

> **⚠WARNING**
> Do not use simple search when you use the multipass option with touch sensing because both simple search and multipass use position registers. Simple search stores the computed position in a position register. Multipass cannot use position registers to plan paths. Use the 2D fillet search pattern when using multipass with touch sensing.

See Fig. 39.3(a) for an illustration of a simple search routine. Refer to Section 39.5 for example programs using simple search.

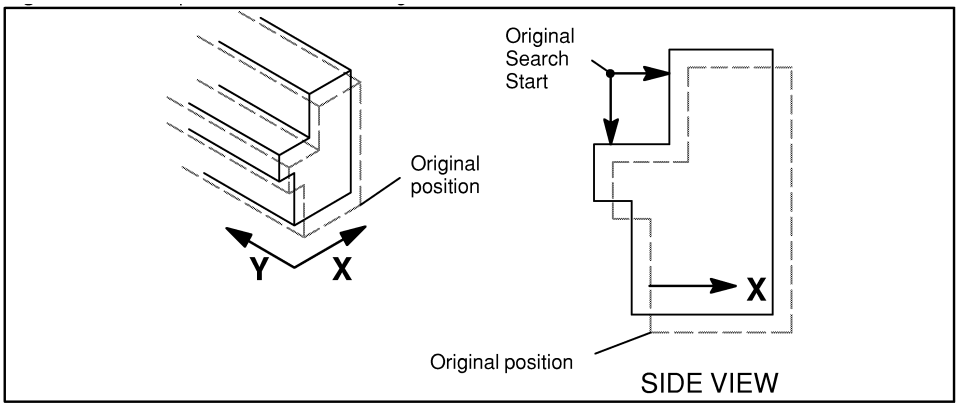


Fig. 39.3 (a) Simple search routine using searches in two directions

Fillet/Lap Search

For Fillet/Lap Search a one, two, or three-dimensional searches are executed to obtain positional offset information. Fillet/Lap Search stores positional offset information in a positional register PR[ ]. This offset can be applied to one or more positions in a programmed path.

1D Shift, 2D Shift, 3D Shift, 1D + Rot, 2D + Rot, 3D + Rot can be used as Search Pattern Type for Fillet/Lap Search. You can offset for on, two or three dimensions. Additionally, rotational offset for compensation can be also used.

For example, when 1D + Rot is selected as Search Pattern Type, the fillet search can offset for a rotation about Z by searching 2 points to X direction. Note that is this type of search, the first touch point is used as the arc start point. See Fig. 39.3(b).

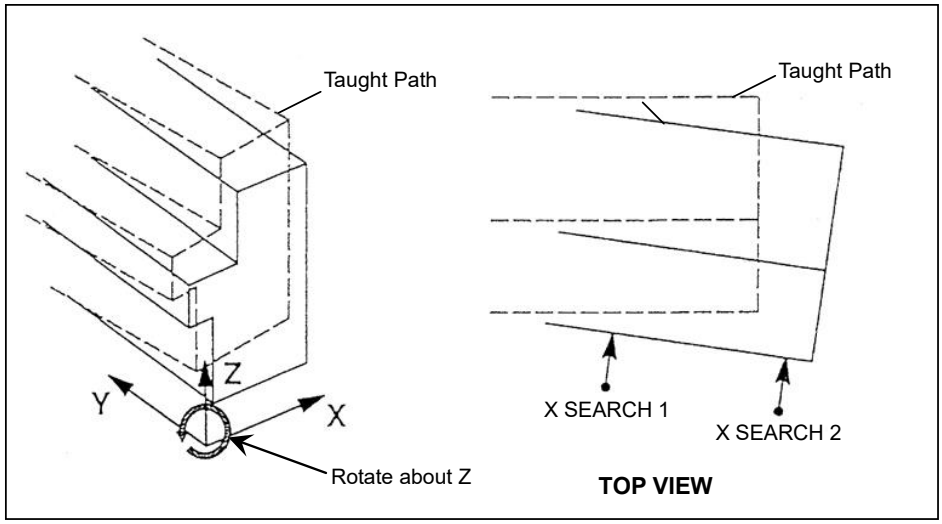


Fig. 39.3 (b) Fillet search in one direction (x) with rotation about z

When 2D + Rot is selected as Search Pattern Type, the fillet search can offset for a rotation about Z by searching 2 points per one direction to X and Y directions. See Fig. 39.3(c).

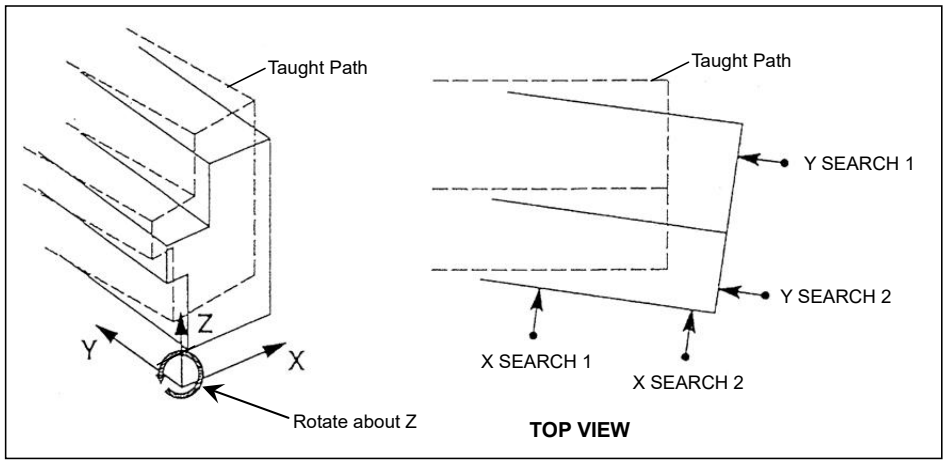


Fig. 39.3 (c) Fillet search in two directions (X and Y) with rotation about Z

When 3D + Rot is selected as Search Pattern Type, the fillet search can offset for rotations about all directions by searching 3 points to another direction in addition to 2D + Rot search motion. See Fig. 39.3(d).

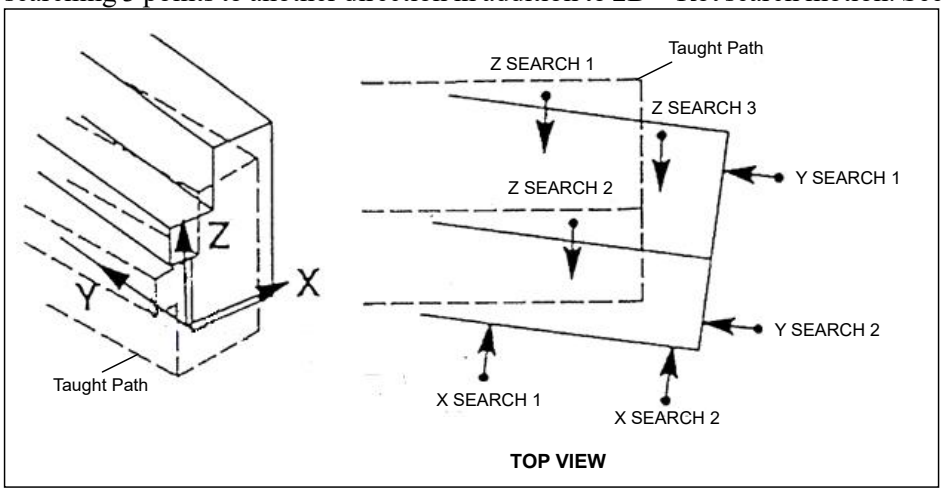


Fig. 39.3 (d) Fillet search in three directions (X, Y, Z) with rotation about all directions

Refer to Section 39.5 for example programs using Fillet/Lap Search.

V-Groove Search

For V-Groove Search a one-dimensional search is executed to obtain positional offset information. A V-Groove Search stores positional offset information in a positional register [PR]. This offset can be applied to one or more positions in a programmed path. Only 1D Search Pattern Type is allowed for V-Groove Search.

See Fig. 39.3(e) for an illustration of a V-Groove Search. Refer to Fig. 39.3(c) of Section 39.6 for example programs using V-Groove Search.

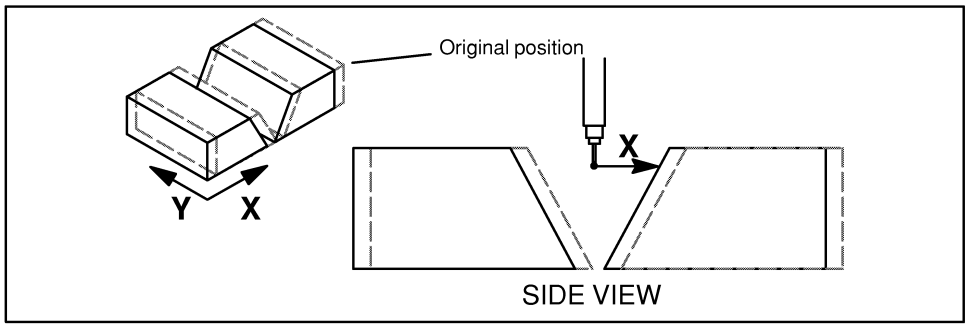


Fig. 39.3 (e) V-groove search

Outside/Inside Diameter Search (OD/ID)

For Outside/Inside Diameter Search (OD/ID Search) a two-dimensional search is executed to obtain the positional offset information of the center point of a circular path relative to the original (master) location. An Outside/Inside Diameter Search stores positional offset information in a positional register [PR]. This offset can be applied to one or more positions in a programmed path.

Only 2D Search Pattern Type is allowed for OD/ID Search. About more detail, refer to Table 39.4.

See Fig. 39.3 (f) for an illustration of a OD/ID Search.

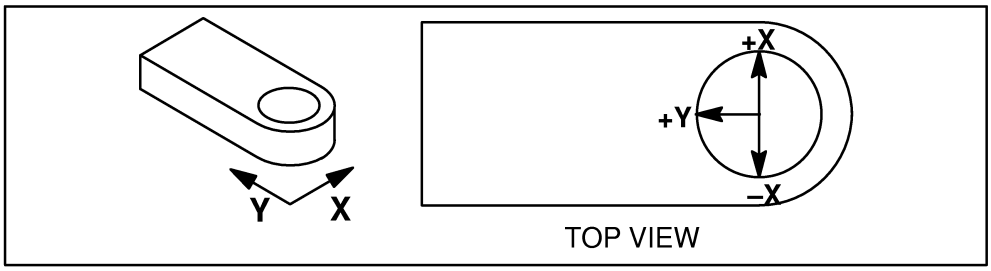


Fig. 39.3 (f) OD/ID search in two directions (x and y)

Table 39.4 shows a matrix of possible search pattern and valid pattern types. Select a combination that you would like to use on your application and verify that it will provide the proper results.

Table 39.3 Search pattern and valid pattern type

Search Patterns	Pattern Type 1_D	Pattern Type 2_D	Pattern Type 3_D	Pattern Type 1_D and Rotation	Pattern Type 2_D and Rotation	Pattern Type 3_D and Rotation
Simple Search	Not Valid	Requires 2 different search directions. 1 search per direction.	Not Valid	Not Valid	Not Valid	Not Valid

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
Fillet/Lap	Requires 1 search direction. 1 search.	Requires 2 different search directions, x and y, x and z, y and z. 1 search per direction.	Requires 3 different search directions, x, y, and z. 1 search per direction.	Requires 1 search direction. 2 searches per direction.	Requires 2 different search directions. 2 searches per direction.	Requires 3 different search directions. 3 searches in one direction (usually -z) 2 searches in each of the remaining directions.
V-Groove	Requires 1 search direction. 1 search	Not Valid	Not Valid	Not Valid	Not Valid	Not Valid
OD/ID	Not Valid	Requires 3 different searches in 2 different directions. For example, +x, -x, +y, NOT x, y, z. 1 search per direction.	Not Valid	Not Valid	Not Valid	Not Valid

39.4 TOUCH SCHEDULE

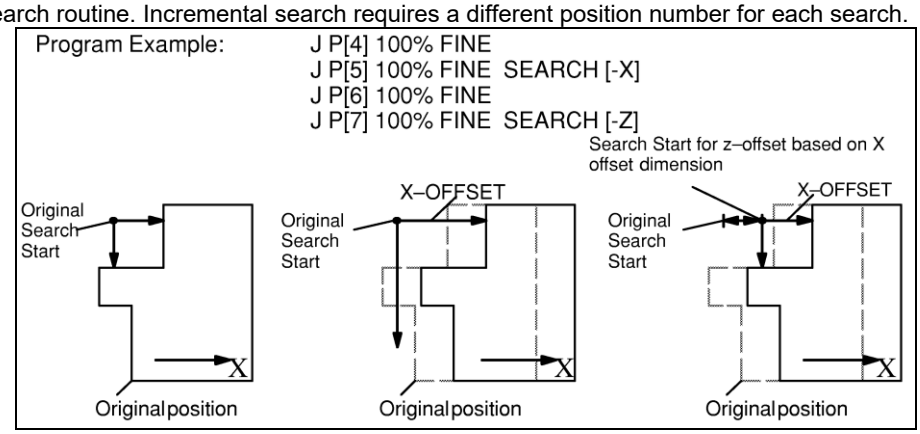
A touch schedule is a series of conditions that control how the search motion is completed. In this schedule, Touch Frame, Search Pattern and Pattern Type are included.

Thirty-two touch schedules are available. You can access touch schedules from the DATA menu. There are two screens associated with touch schedules: the SCHEDULE screen and the DETAIL screen.
The SCHEDULE screen allows you to view and set limited information for all schedules at once. DETAIL allows you to view and set the complete information for a single schedule.
Table 39.4 lists and describes each DETAIL screen condition. The items of SCHEDULE screen are also included in touch sensing DETAIL screen.
Use Procedure 39-5 to define touch schedules.

Table 39.4 Touch schedule screen Items

ITEM	DESCRIPTION
Touch Schedule	This item indicates the number of the displayed schedule. A comment can be entered.
Master Flag Default=OFF	This item enables the search routine to be used as a mastering routine for those touch sensing programs that generate positional offset information. If set to ON, when the search routine is executed, the touched positions are recorded on TP program internally as the reference positions to be used by future searches. This flag must be set to OFF after the master search is completed in order to generate the information of position offset on the objects to be searched. When the search is performed with Master Flag ON, the information of position offset in the position register is set to all zero values. The Master Flag condition has no effect on simple search pattern.

ITEM	DESCRIPTION
Search Speed Default = 50.0 mm/sec	This item specifies how fast the robot will move when performing a Search Motion. The accuracy of search will be improved when this speed is slow. **⚠ CAUTION** A search motion is programmed as a motion option at the end of a motion instruction. The speed at which the robot will move is determined by the search speed, not by what is indicated in the motion instruction. During testing, when dry run is in effect, this search speed is also used. The dry run speed has no effect.
Search Distance Default = 100 mm	This item defines how far the robot can move when it is performing a search. Error code "THSR-017 No contact with part." is posted when this distance is reached without making contact with the workpiece.
Touch Frame Default = 1	This item defines the touch frame number to be used in the touch schedule. This determines the X, Y, and Z directions for the search motion. The same touch frame can be used in more than one touch schedule.
Search Pattern Default = SIMPLE	This item defines the type of object to be searched and causes the ArcTool software to compute the found position or positional offset information dependent on the search pattern selected. The computed data is stored in a position register. There are four available search patterns: - Simple Search - Fillet/Lap Search - V-Groove Search - OD/I D Search About more detail, refer to the Section 39.3.
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 with Fillet/Lap Search Pattern. Six pattern types are available: Stores a one-dimensional offset. Offsets can be in the X, Y, or Z direction. Stores a two-dimensional offset. Offsets can be in two of the X, Y, or Z direction. Stores a three-dimensional offset to a program. Offsets are in the all of X, Y, and Z directions. Stores a one-dimensional offset with rotation about the axis of which the search is not performed. Stores a two-dimensional offset with rotation about the axis of which no searches are performed. For example, if the workpiece is being searched for an offset in both the X and Y directions, a 2_D Shift & Rotate search can offset for a rotation about the z axis. Stores a three-dimensional offset with rotation about the all axes are performed. NOTE: Simple, OD/ID, and V-Groove search patterns are pre-defined. Changing the pattern type for these searches has no effect.

ITEM	DESCRIPTION
Incremental Search Default = ON	Offsets the starting position of the second etc. search in a search routine by the amount of offset found by the first search motion. If set to OFF, the robot returns to the original starting position. The following illustration shows how the incremental search affects the search routine. Incremental search requires a different position number for each search.  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.
Reference Group	This item specifies how the offset is recorded. When Search Pattern is not Simple Search, you should specify the same group number as Ref Grp on selected Touch Frame. When Search Pattern is Simple Search, you should specify the same group number as Robot Group item on this schedule. **NOTE** When search pattern is not simple search, if Reference Group is not equal to Ref Grp on Touch Frame, an error message "THSR-039 Reference grp mismatch," will be posted. When search pattern is simple search, if Reference Group is not the same as the specified number of Robot Group on this touch schedule, an error message, "THSR-037 Illegal motion ref. grp," will be posted.
Return Term Type Default = FINE	This item specifies the termination type the robot will use to return to the search start position. Four Return Term Types are available: - FINE - CNT20 - CNT40 - CNT100

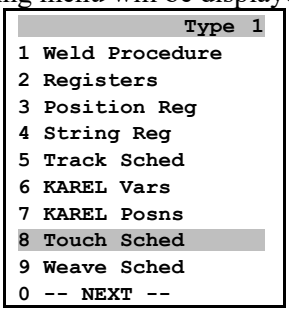
ITEM	DESCRIPTION
Contact Record PR Default = 32	The search output position register is used as a temporary buffer to hold the last search contact position. The purpose for this temporary position register buffer is to provide the ability to look at the positional data of an individual search, or to extract data from the buffer in a program. By default, this register is position register 32. **⚠ CAUTION** The data in the position register is overwritten at each search motion so the same position register should not be used to store the final positional data from the search motion. Also, the contents of this temporary buffer is a real position, not an offset. Do not program motion instructions to use this position register data as an offset.
Error on Failure Default = ON	This item posts error code “THSR - 017 No contact with part”, if the search move exceeds the distance set in Search Distance. When this item is OFF, the program execution continues with the next instruction even if the Search Distance is exceeded. Programming Hint: If this is set to OFF, the next instruction in the program looks at the contents of the Error Register and branch accordingly.
Error Register Number Default = 32	When Error On Failure is set to OFF, this register is set to 1 when the search distance is exceeded. A successful search sets this register to 0.
Robot Group Default = [1,*,*,*,*,*,*,*]	This item specifies the robot group that uses the touch sensing schedule.

Procedure 39-5 Setup of Touch Schedule

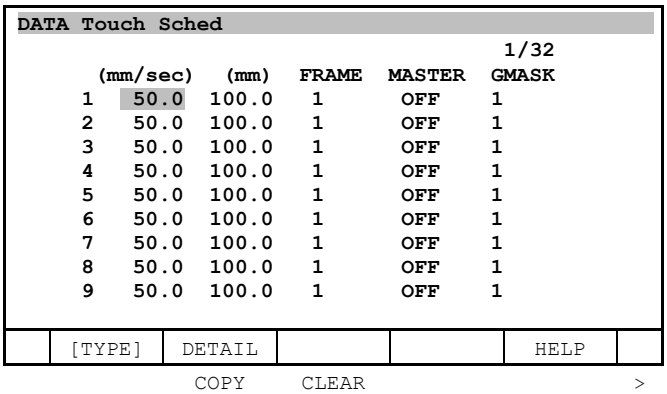
Step

1 Press the [DATA] key.

2 Press the 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 the [NEXT] key.

b Move the cursor to the schedule you want to copy.

c Press F2, COPY. Following message is displayed.

```
Enter schedule number :
```

d Enter the schedule number from which you want to copy the data.
e Press the [ENTER] key. 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 the [NEXT] key.
c Press the F3, CLEAR. Following message is displayed. If F4, Yes is pressed, the data will be cleared, but the comment will remain.

```
Clear this schedule? [NO]

                                  Yes     No
```

6 Move the cursor to the desired schedule number.
7 To display more information about the schedule, press the F2, DETAIL. See the following screen for an example.

```
DATA Touch Sched
                                          1/17

     1 Touch Schedule:1  [ Touch Schdule 1]

     2 Master flag:         OFF
     3 Search speed         50.0  mm/sec
     4 Search distance:     100.0  mm
     5 Touch frame:         1
     6 Search pattern:      Simple
     7 Pattern type:        1_D Shift
     8 Incremental search:  ON
     9 Auto return:         ON
    10 Return speed:        100.0  mm/sec
    11 Return distance:     2000.0 mm
    12 Reference Group:     1
    13 Return term type:    Fine
    14 Contact record PR:   32
    15 Error on failure:    ON
    16 Error register num:  32
    17 Robot Group Mask:   [1,*,*,*,*,*,*,*]

  [TYPE]  SCHEDULE                    HELP
```

8 Set each schedule item as desired.
9 To add a comment:
a Move the cursor to the to the comment line and press the [ENTER] key.
b Select a method of naming the comment.
c Press the appropriate function keys to add the comment.
d When you are finished, press the [ENTER] key.
10 To select [Search pattern], [Pattern type], [Return Term Type]:
a Move the cursor to each item and press F4, SELECT.
b Select the item and press the [ENTER] key.

NOTE
Touch Schedule is only for defining method of Touch Sensing, and information of master positions is never saved on Touch Schedule (the information is saved on TP program). Therefore, you can use one Touch Schedule in multiple programs and multiple places.

39.5 TOUCH SENSING PROGRAMMING

A touch sensing routine consists of search instructions to locate an object, and offset instructions to displace programmed positions.

> **⚠WARNING**
> Recorded positions and position registers are affected by UFRAME, and UFRAME has an affect during playback. If you change UFRAME, any recorded positions and position registers will also change.

> **NOTE**
> Any changes to the tool frame affects the search start position.

Touch Sensing Instructions

Following touch sensing instructions are provided:

- Search Start [ ] PR [ ] (Search Start instruction)
- Search End (Search End Instruction)
- Touch Offset PR[ ] (Touch Offset Start instruction)
- Touch Offset End (Touch Offset End instruction)

Additionally, following instruction is used as additional motion instruction between Search Start – Search End instructions.

- Search [ ] (Search instruction)

> **⚠CAUTION**
> Do not use Replace in (EDCMD) menu on program edit screen to the program that has Touch Sensing instructions. This will cause memory write error. If you want to replace motion instructions, insert Touch Sensing instructions after deleting the motion instruction that you want to replace.

Search Start/End Instructions

Search Start instruction commands the start of search motion. Specify Touch Schedule number for this search motion, and also specify Position Register number for storing positional information by Touch Sensing.

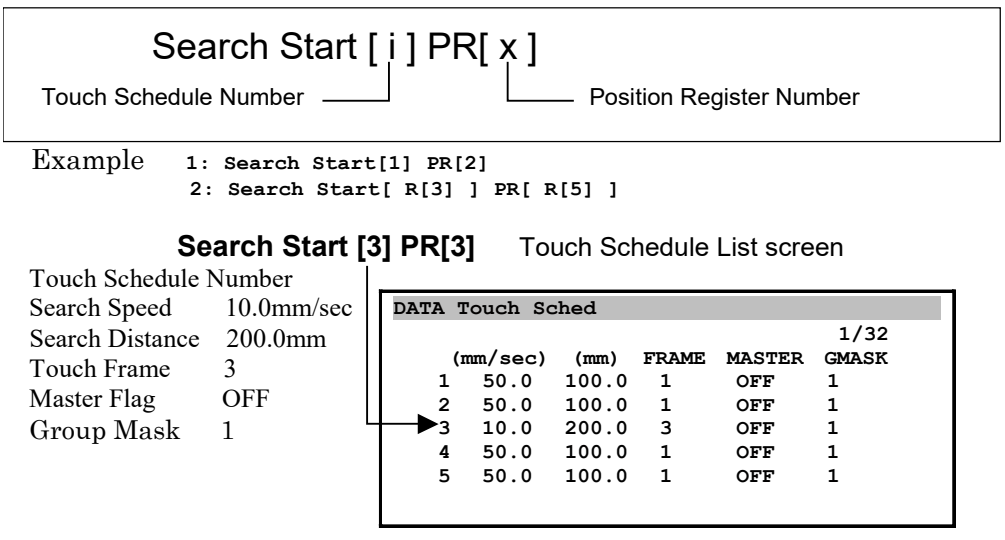


Fig. 39.5(a) Search start instruction

One Search End instruction is required for one Search Start instruction. If you want to execute Search Start instruction again for new search motion, you must execute Search End instruction for finishing previous search motion.

Search End

Example `1: Search End`

Fig. 39.5(b) Search end instruction

Search instruction must be taught between this Search Start instruction and Search End instruction.

Procedure 39-6 Teach Search Start/End Instructions

Condition

- Program edit screen has been displayed.
- Teach pendant is enabled.

Step

1 Press the [NEXT] key and press the F1, INST. The following menu will be displayed.

```
      Instruction 4
1 Stick Detect
2 String
3 Touch Sensor
4
5
6
7
8 --next page--
```

2 Select Touch Sensor. List of touch sensing instructions are displayed. If you teach Search Start instruction, select Search Start.

```
     Touch Sensor 1
1 Search Start
2 Search End
3 Touch Offset
4 Touch Offset End
5
6
7
8
```

3 Enter Touch Schedule number to [] just after Search Start. Additionally, enter position register number to [] of PR.
4 If you teach Search End instruction, select Search End on Step 2.

Search Instruction

There is one Touch Sensing motion option: Search [ ]..The Search [ ] motion option directs the motion of the robot (in a positive or negative X,Y or Z direction) to search for the object. The X, Y and Z vectors are defined by the touch frame assigned in the touch schedule.

The taught position in motion instruction that has the search instruction does not has actual meaning, so motion to the search start position must be taught by a separate motion instruction. Additionally, motion speed in motion instruction that has the search instruction does not have actual meaning, either. Instead of

this, Search Speed in Touch Schedule (refer to Table 39.5) is used as motion speed. When the touch sensor part touches the workpiece during Search Motion, current position is recorded to Contact Record PR.

```
TOUCH_SENSING_1
                                        13/13
    3:J  P[2] 50% FINE
    4:J  P[3] 50% FINE Search[-X]
 [End]
```

Move to Search Start Position
Search Motion (P[3] is never used)

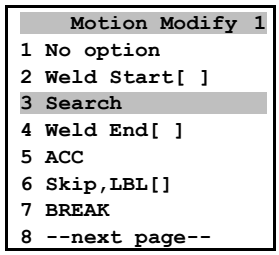
Fig. 39.5(c) Program example using search instruction as additional motion instruction

About how to teach Search instruction, refer to Procedure 39-7.

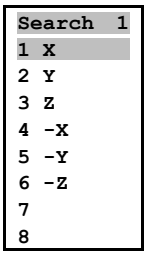
Procedure 39-7 Teach Search Instruction

Step

1 Jog the robot to the search start position and teach motion instruction.
2 Teach another motion instruction again at the same position.
3 Move the cursor to the end of the motion instruction taught on Step 2, and press the F4, [CHOICE] to view the motion option choices.



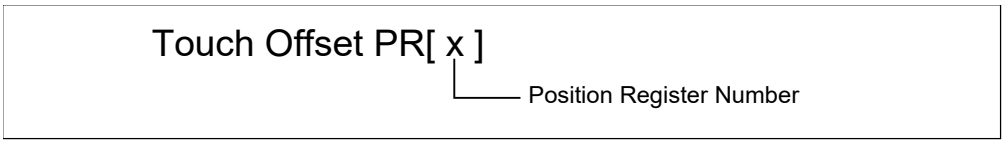
4 Select Search.
5 Select the direction of the search to be performed and press the [ENTER] key.



⚠CAUTION
Teach Search instruction at the last position of additional motion instruction. Additionally, motion instruction for moving to search start position (motion instruction which is taught just before motion instruction which has Search instruction) must use FINE termination type. If CNT termination type is used for their instructions, correct offset information cannot be calculated.

Touch Offset Start/End Instructions

Touch Offset Start/End instructions are used for shifting positions on the program by using position offset which is obtained by search motion. The position offset by Touch Offset is defined by the position register specified in Touch Offset Start instruction. Normally, specify position register number which record position offset by Search Start/End instructions.



Example
```
1: Touch Offset PR[3]
2: Touch Offset PR[ R[5] ]
```

Fig. 39.5(d) Touch offset start instruction

Example
```
1: Touch Offset End
```

Fig. 39.5(e) Touch offset end instruction

About how to teach Search instruction, refer to Procedure 39-8.

Procedure 39-8 Teach Touch Offset Start/End Instructions

Condition

- Program edit screen has been displayed.
- Teach pendant is enabled.

Step

1 Press the [NEXT] key and press the F1, INST. The following menu will be displayed.

```
      Instruction 4
1 Stick Detect
2 String
3 Touch Sensor
4
5
6
7
8 --next page--
```

2 Select Touch Sensor. List of touch sensing instructions are displayed. If you teach Touch Offset Start instruction, select Touch Offset.

```
     Touch Sensor 1
1 Search Start
2 Search End
3 Touch Offset
4 Touch Offset End
5
6
7
8
```

3 Enter position register number to [] of PR.
4 If you teach Touch Offset End instruction, select Touch Offset End on Step 2.

> **⚠CAUTION**
> Do not use CALL instruction between Touch Offset Start – End instructions. On the motion instructions after the execution of CALL instruction, Touch Offset becomes invalid.

Program Example of Simple Search

After a Simple Search routine, the touch sense software will calculate an absolute (real) position (X,Y,Z,W,P,R) and put the data in the position register defined by the Search Start instruction.

Since this is a real position, the robot will be commanded to move to the position in the position register.

Example: J PR [4] 100% FINE Weld Start [1,1]

Above example shows where position register 4 is the position register specified in the simple search routine.

Simple Search has different points from other Search Patterns.

- Master Flag is always OFF.
- Position values stored in position register is absolute position data.
- Touch Offset Start/End instructions are never required.

```
SIMPLE_SEARCH
                                        13/13
     1:J  P[1] 50% FINE
     2:  Search Start[4] PR[4]
     3:J  P[2] 50% FINE
     4:J  P[3] 50% FINE Search[Y]
     5:J  P[4] 50% FINE Search[-Z]
     6:  Search End
     7:J  PR[4] 50% FINE
      :  Weld Start[1,1]
```

Teach Home Position
Start Search Motion. Use Touch Sch 4, PR[4]
Teach a search start position
Search to Y direction
Search to -Z direction
Finish Search Motion
Move the robot to PR[4] position,
And start Arc Welding

Fig. 39.5(f) Program example of simple search (Refer to Fig. 39.4(a))

Program Example of 1D Search

In Touch Sensing program except simple search pattern, use the offset information by search motion on Touch Offset instruction, and offset the taught path.
In 1D Search, one point to one direction to which the workpiece will shift is searched, and taught path is compensated by Touch Offset instruction. Refer to the following program example.

```
1D_SEARCH
                                        12/13
     1:J  P[1] 50% FINE
     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 to 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)

Program Example of 2D Search

In 2D Search, search to two directions are performed, and the search point is one point per one direction. Then taught path is compensated by Touch Offset instruction. Refer to the following program example.

```
2D_SEARCH
                                        12/13
     1:J  P[1] 50% FINE
     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 to y direction
Teach an another search start position
Search to 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)

Program Example of 2D + Rotation Search

In 2D + Rotation Search, search to two directions are performed, and the search points are two points per one direction. Then taught path is compensated by Touch Offset instruction. Refer to the following program example.

```
2D+ROT_SEARCH
                                        25/26
     1:J  P[1] 50% FINE
     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 an another search start position
Search to X direction
Teach an intermediate position
Teach an another search start position
Search to Y direction
Teach an another search start position
Search to 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)

Program Example of 3D + Rotation Search

In 3D + Rotation Search, 3 points search to another direction is added to 2D + Rotation Search. Rotations of all directions are also obtained as offset data.

```
3D+ROT_SEARCH
                                        25/26
    1:J  P[1] 50% FINE
    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 to X direction
Teach an another search start position
Search to X direction
Teach an intermediate position
Teach an another search start position
Search to Y direction
Teach an another search start position
Search to Y direction
Teach an intermediate position
Teach an another search start poison
Search to Z direction
Teach an another search start poison
Search to Z direction
Teach an another search start poison
Search to 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)

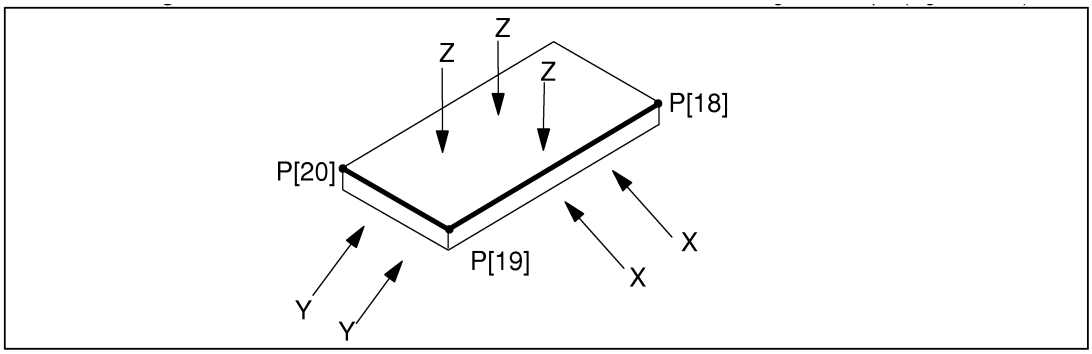


Fig. 39.5(k) Outline of 3D + rotation search

⚠CAUTION
For Fillet/Lap, V-Groove, OD/ID search pattern programs, you should perform Search Motion (Search Start – Search End section) first with Master Flag ON to saving information of master positions into TP program. You must establish master positions for all search motion. Then execute Touch Sensing program with Master Flag OFF.
If the program has ever been executed with Master Flag ON, the Touch Sensing program cannot work correctly (Refer to Section 39.6).

39.6 EXECUTION OF TOUCH SENSING PROGRAM

Touch Sensing Mastering

If except simple search pattern is used, offsets generated by touch sensing are relative to the position obtained by mastering. An offset is computed by comparing the position of the current workpiece (offset position) and the original workpiece position (mastering position).
Mastering is necessary once for obtaining the original workpiece position before executing Touch Sensing program. Perform Mastering by referring to Procedure 39-9.

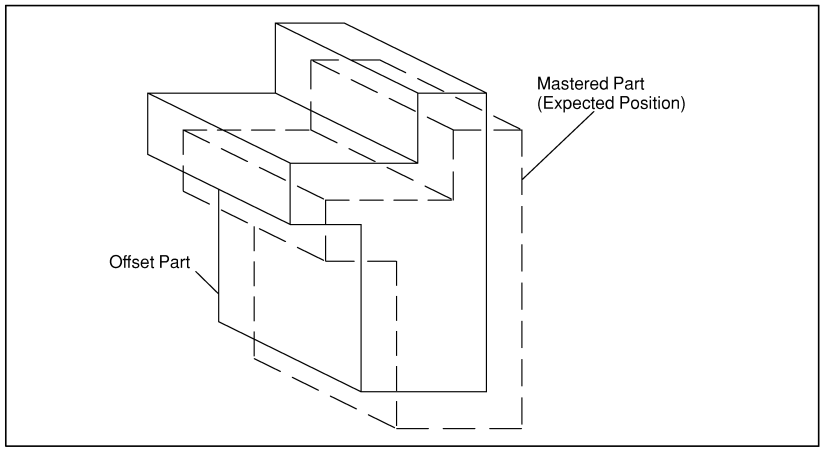


Fig. 39.6(a) Part in mastered position and offset position

NOTE
A simple search does not require mastering since it produces an absolute position stored in a position register.

Procedure 39-9 Touch Sensing Mastering

Condition

- The workpiece is placed without shift (placed on taught path).
- Touch Sensing Program is already created.
- Except simple search pattern is selected on Touch Schedule.

Step

1 Display Touch Schedule Detail screen for used Touch Schedule number by referring to Procedure 39-5.
2 Move the cursor on Master Flag and then press the F4, ON. Master Flag becomes ON.
3 Display the program edit for mastering.
4 Execute program between Search Start – Search End instruction (it is also OK with Single Step). Mastering position is stored to the TP program. Then, data in specified position register in Search Start instruction become all zero.
5 Display Touch Schedule Detail screen, move the cursor on Master Flag, and then press the F5, OFF. Master Flag becomes OFF.

NOTE
Incremental search is disabled while the Master Flag is turned ON. If incremental search is not performed, check Master Flag.

Execution of Touch Sensing Program after Mastering

After finishing Mastering, the taught path is compensated every time by executing Search Motion and Touch Offset. The process details of Touch Sensing program after Mastering are explained by referring to the program example of 2D Fillet Search (Fig 39.6(b), (c)).

```
2D_FILLET_SEARCH
                                          12/13
    1:J  P[1] 50% FINE
    2:  Search Start[3] PR[1]
    3:J  P[2] 50% FINE
    4:J  P[3] 50% FINE Search[X]
    5:J  P[2] 50% FINE
    6:J  P[4] 50% FINE Search[-Z]
    7:  Search End
    8:
    9:  Touch Offset PR[1]
   10:J  P[5] 50% FINE
   11:L  P[6] 50cm/min CNT100
   12:L  P[7] 50cm/min FINE
   13:  Touch Offset End
```

Fig. 39.6(b) Program example of 2D fillet search

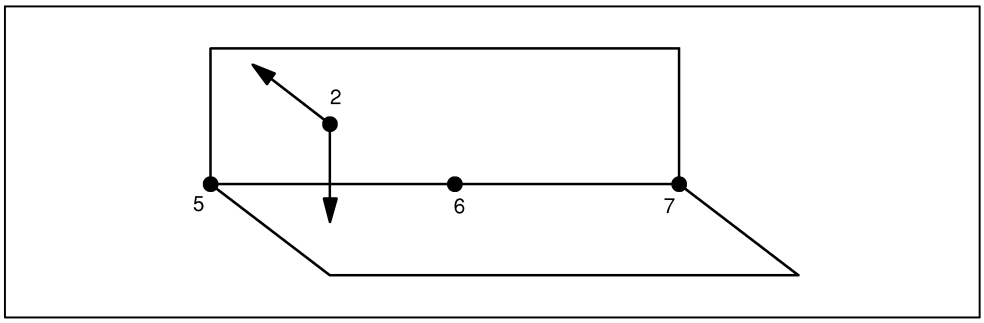


Fig. 39.6(c) Part with one touch start position, 2, and three points along a path, 5, 6, 7

- After finishing Mastering, Mastering position and current position (Offset position) obtained by Search Motion are compared and offset data is calculated when Search Motion (between Search Start – End instruction) is performed with Master Flag OFF. The offset data obtained by this Search Motion becomes relative data between Mastering position and Offset position.

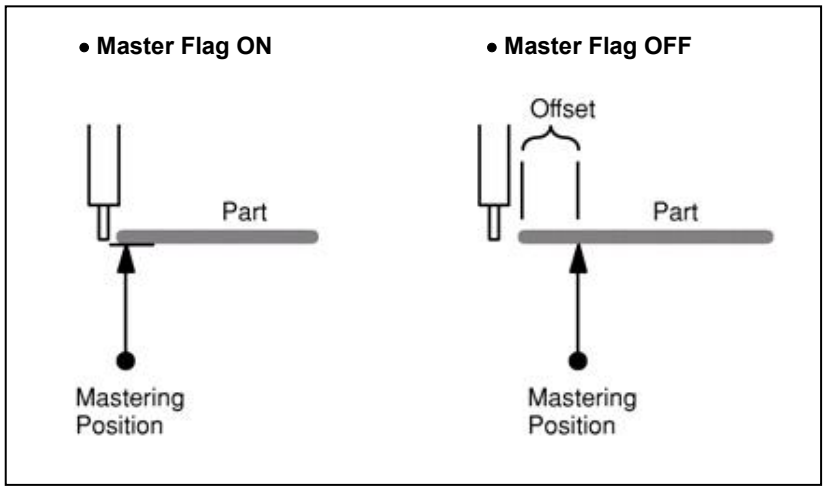


Fig. 39.6(d) Offset value

- After the execution of Search End instruction, offset data is set to position register specified on Search Start instruction.
- After that, compensated motion for current workpiece position is done by applying offset data to all motion instruction placed between Touch Offset Start – End instructions (position 5, 6 and 7 on Program Example).

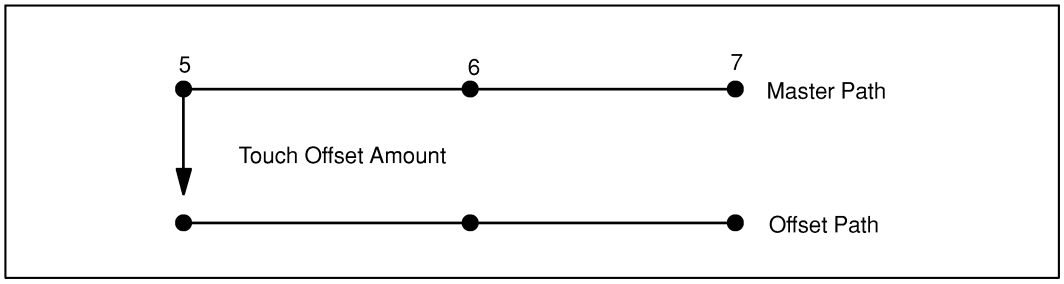


Fig. 39.6(e) Illustration of the path when an offset Is applied

Details of Touch Sensing Program Execution

- If BWD execution is performed for Search Start instruction, Search Motion becomes temporarily disabled. If you want to restart Search Motion, restart the program from the line just before Search Start instruction by FWD execution again.
- Touch Offset finishes when Touch Offset End instruction is executed or program is aborted. On the other hand, Touch Offset continues when program is paused and then resumed.
- If BWD execution is performed for Touch Offset Start instruction, Touch Offset becomes temporarily disabled. If you want to restart Touch Offset, restart the program from the line just before Touch Offset Start instruction by FWD execution again.
- After the FWD execution of Touch Offset Start instruction, if cursor is moved to except between Touch Offset Start – End instructions and then program is restarted, offset specified by Touch Offset Start instruction is applied (that is, the offset is taken over).

Remastering

The touch up procedure described in Section 39.8 should work for most instances where the search start positions do not need to be moved or if the parts do not change drastically.
Remastering is required if the search start positions must be taught again. If the path must be altered significantly, it is recommended to remaster to ensure a correct path.

The procedure of Remastering is not different from normal Mastering (refer to Procedure 39.9). After finishing Remastering, turn Master Flag OFF.

39.7 TOUCHING UP OF TOUCH SENSING PROGRAM

Mastering position data are stored internally into TP program using Touch Sensing. Therefore, take care to change the program. In this section, notifications for touching up new position on Touch Sensing program are explained by referring to Fig. 39.7 (a) and (b).

Touching Up Path Positions

Occasionally the part or its placement on a fixture will change requiring adjustment of the path. The entire process of remastering is not need to accommodate these changes.
When touch up of motion instructions placed between Touch Offset instructions is required, it is necessary to consider the current offset data of workpiece and teach the new master path. Therefore, the operation of touch up must be done after the completion of correct Search Motion and when Touch Offset continues (offset is compensated). In this situation, if touch up is done by jogging the robot, the only difference information between original offset position is reflected to the position data in the program and new correct master path is generated (refer to Fig. 39.7(a), (b)).

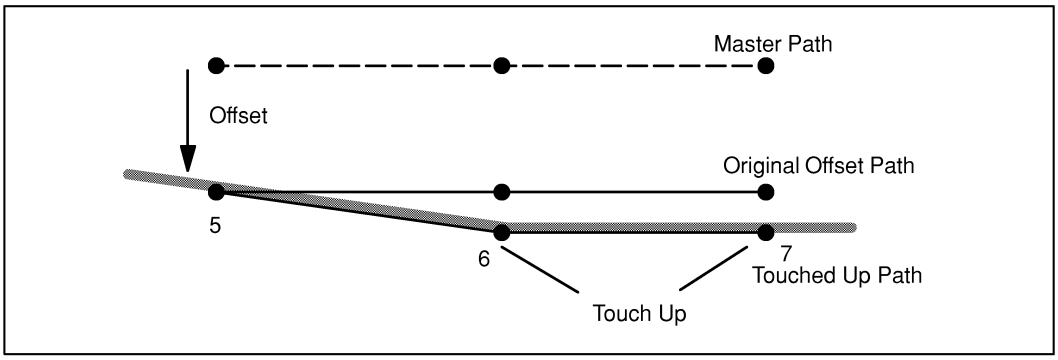


Fig. 39.7 (a) Touch up of position 6 and 7

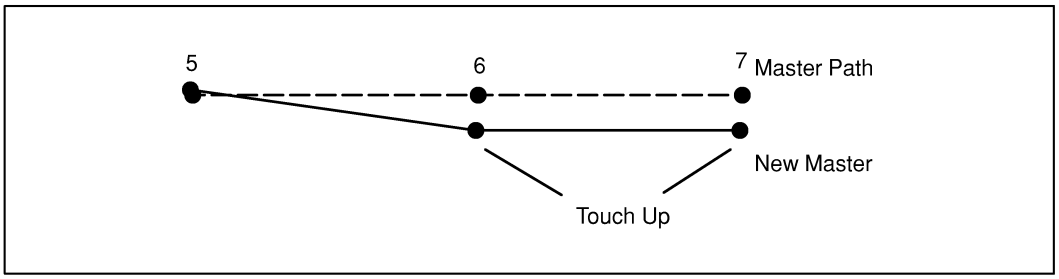


Fig. 39.7 (b) New master path after touch up of position 6 and 7

Procedure 39-10 shows the method to touch up the robot on Touch Sensing program. Additionally, Fig. 39.7(c) shows the example which requires Procedure 39-10 during touch up.

```
2D_FILLET_SEARCH
                                        9/14
     9:  Touch Offset PR[1]
    10:J  P[5] 50% FINE
    11:L  P[6] 50cm/min CNT100
    12:L  P[7] 50cm/min CNT100
    13:  Touch Offset End
```

When you touch up these positions, Procedure 39-10 is required.

Fig. 39.7(c) Example of touch up during touch offset

Procedure 39-10 Touch up of Touch Sensing Program

Condition

- Program edit screen has been displayed.
- Teach pendant is enabled.

Step

1 Execute Search Motion (between Search Start – End instructions) and store the offset data to position register.
2 Execute Touch Offset Start instruction.
3 Forward the program to motion instruction whose position is required to touch up with Single Step.
4 Jog the robot to the position which you want to touch up. Then, move the cursor to position number of current line and press the [SHIFT] key and the F5, TOUCHUP key. Touch up is done.
5 Touch up all required position between Touch Offset Start – End instructions by Step 4.

⚠CAUTION
Do not touch up after BWD execution during execution between Touch Offset Start – End instructions with Single Step. Otherwise, offset data will become wrong value.

Incorrect Touch Up

A common error is to alter the path without the correct offset being applied. Touch up must be done after executing the search and while the Touch Offset is applied.
An example of an incorrect touch up is as follows:

- Execute the program without executing Search Motion.
- Perform touch up the positions between Touch Offset Start – End instructions without executing Touch Offset Start instruction.
- Perform touch up with the position register which has incorrect offset data.

If incorrect touch up is performed, new master position become wrong. Therefore, you cannot obtain the correct compensated path even if Search Motion and Touch Offset are properly done.

Adding New Positions

Additional points can be added in the same manner of touch up Procedure 39-10.

- The search must be completed and an accurate offset must be generated.
- This operation should be done during Touch Offset.
- New position should be added to the offset path.

Fig. 39.7(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 Touch Up and correct path compensation cannot be obtained.

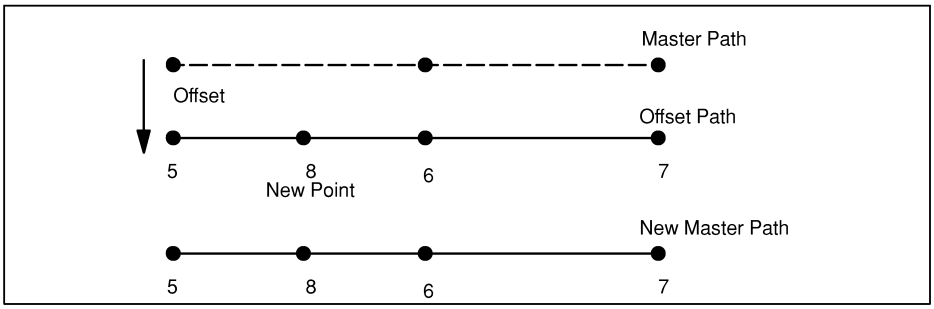


Fig. 39.7(d) Add new point while executing touch offset

Touch Up of Search Start Position

The touch up of a search start position is different from the touch up of the Touch Offset path position. If the search start position is touched up, remastering must be executed. But, if the search start position is moved along the axis of the search direction, remastering is not necessary.

Example that does not require Remastering

If the search start position is too close to the part due to poor programming, changes in the part, or a change in the part location, then the search start position needs only to be moved back along the search direction. This can be accomplished with no effect on the path positions and remastering will not be required.
The following example shown in Fig. 39.7(e) shows a part and search start position.

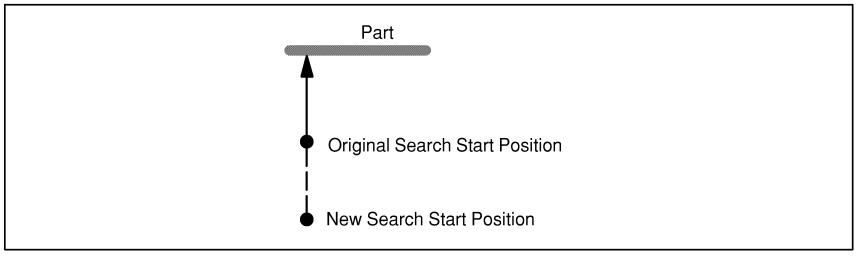


Fig. 39.7(e) Moving a search start position along the search direction

Example which requires Remastering
The example shown in Fig. 39.7(f) shows the search start position moved to the position separated the axis of the search direction. If the search position is moved off the axis of the search direction, Remastering is required. About Remastering, refer to Section 39.7.

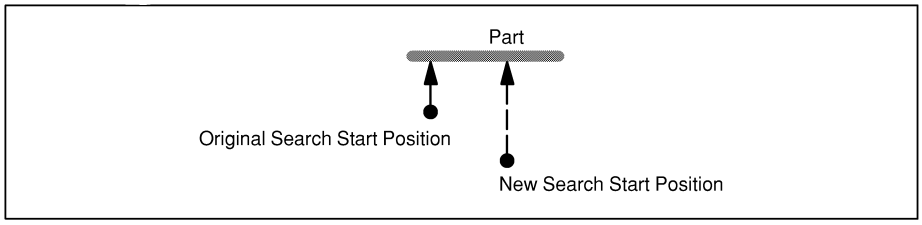


Fig. 39.7(f) Search start position Is separated from original search direction axis

39.8 MULTIPLE SEARCHES

Complex program can have multiple searches generating several offsets.

The following program example shown in Fig. 39.8(a) shows two searches that can be performed for complex shapes. When search is performed for that complex workpiece, Search Motion may not be able to complete by one Touch Frame. Therefore, multiple searches are required.
- The first search stores the offset data in position register 1 with positions 10, 11, 12 using the offset.
- The second search stores offset data in position register 2 with positions 13, 14, 15 using the offset.

```
MULTIPLE_SEARCHES
                                          25/26
     1:J  P[1] 50% FINE
     2:  Search Start[3] PR[1]
     3:J  P[2] 50% FINE
     4:J  P[3] 50% FINE Search[X]
     5:J  P[4] 50% FINE Search[-Z]
     6:  Search End
     7:J  P[5] 50% FINE
     8:  Search Start[4] PR[2]
     9:J  P[6] 50% FINE
    10:J  P[7] 50% FINE Search[-X]
    11:J  P[8] 50% FINE
    12:J  P[9] 50% FINE Search[-Z]
    13:  Search End
    14:
    15:  Touch Offset PR[1]
    16:J  P[10] 50% FINE
    17:L  P[11] 50cm/min CNT100
    18:L  P[12] 50cm/min CNT100
    19:  Touch Offset End
    20:
    21:  Touch Offset PR[2]
    22:J  P[13] 50% FINE
    23:L  P[14] 50cm/min CNT100
    24:L  P[15] 50cm/min CNT100
    25:  Touch Offset End
```

Fig. 39.8(a) Program example of multiple searches

If a position of the taught path is to be touched up, the corresponding search must be performed.

- The first search must be executed to obtain an accurate offset.
- The offset is applied and the positions, 10, 11, and 12 can be touched up as normal.
- The master will be correctly updated.

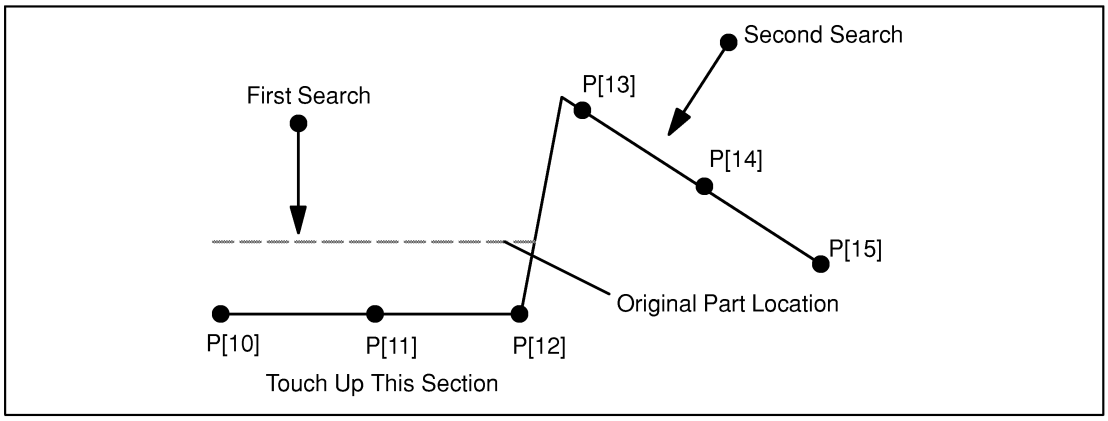


Fig. 39.8(b) Touch up on multiple searches

39.9 COORDINATED TOUCH SENSING

Touch Offset by Touch Sensing can also apply to a motion sentence which has a coordinated motion instruction. To achieve this, you must set Touch Frame and Touch Schedule for Coordinated Touch Sensing. The setup and program are changed by search pattern (Simple Search, or others).

Coordinated Touch Sensing except Simple Search

Fig. 39.9(a) shows Search Motion of leader group (Group 2: Positioner) and follower group (Group 1: Robot) for 2D+Rotation with coordinated motion. Search Motion is done based on coordinated frame of leader group. Touch Offset motion after Search Motion is shown in Fig. 39.9(b).

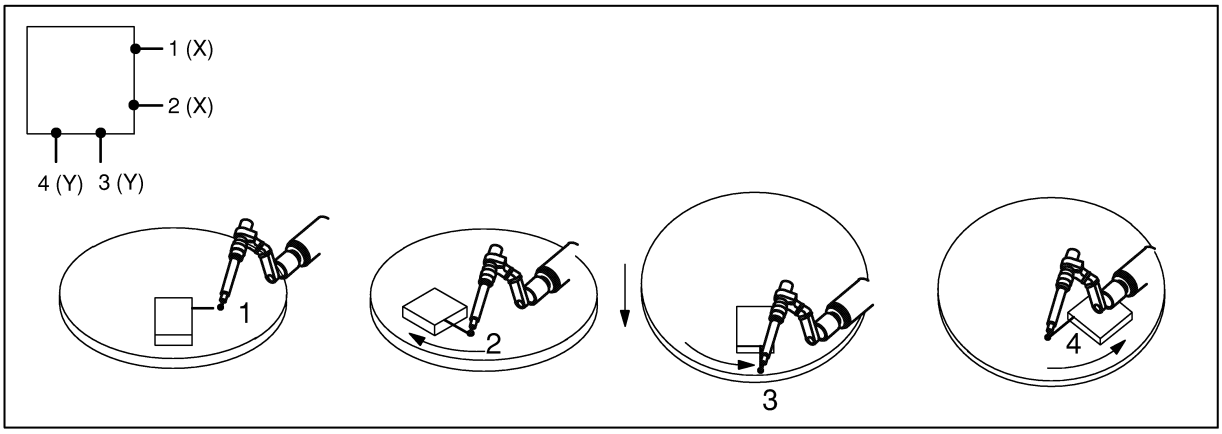


Fig. 39.9(a) 2D+rotation search with coordinated motion

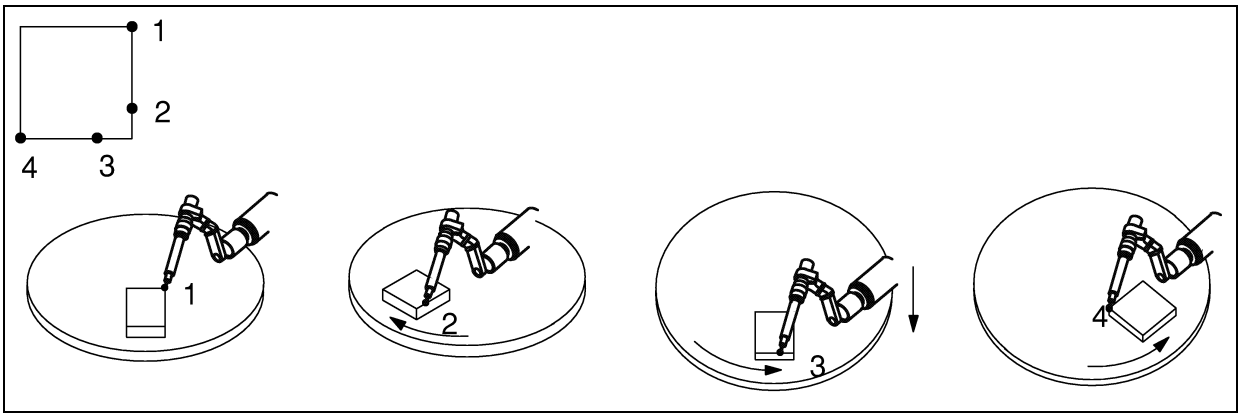


Fig. 39.9(b) 2D+rotation Touch offset with coordinated motion

First, Touch Frame setup is shown.
Enter the group number of robot group (follower group) that performs Search Motion and Touch Offset to "RoBot Grp" item. After that, enter the group number of leader group of coordinated motion to "Ref Grp" item. That is, if Coordinated Touch Sensing is performed, "Ref Grp" number and "Robot Grp" number" in Touch Frame are different. About more detail, refer to Procedure 39-3 and 39-4.
Before the setup, you should already complete the calibration of coordinated pair between leader group and follower group in coordinated motion. If you change "Ref Grp" item on Touch Frame without calibration of coordinated pair, "Referenced group does not exist" message will be displayed. About the calibration of coordinated pair, please refer to Coordinated Motion Function OPERATOR'S MANUAL (B-83484EN).

Additionally, "Ref Grp" number and "Robot Grp" number on Touch Frame must be also entered on Touch Schedule. Set the robot group for Touch Sensing to "Robot Group Mask" item in Touch Schedule Detail screen. Next, enter Touch Frame number to "Touch Frame" item. Then, enter the same number as "Ref Grp" number (Leader group number) on Touch Frame to "Reference Group" item.

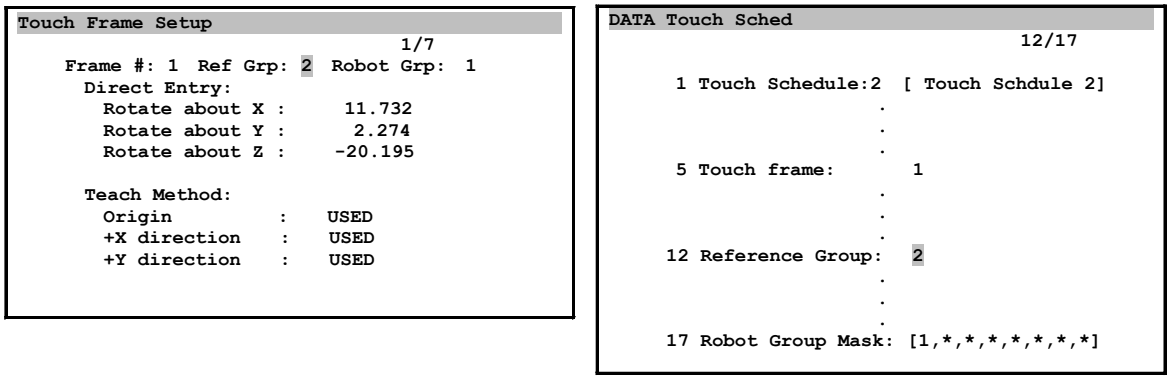


Fig. 39.9(c) Setup example of touch frame and touch schedule (Coord, except simple search)

Fig. 39.9(d) is the program example. Search direction is derived from the direction of leader group (positioner in the figure). Even if leader group moves, search direction is not changed relatively. It is also possible to teach the motion instruction which is not related to Search Motion (refer to Line 7 of Fig. 39.9(d)). The offset data obtained by Search Motion is compensated by the direction of leader group.

Group Mask in the Program[1,1,*,*,*,*,*,*]

```
COORD_SEARCH                                    19/20
     1:J  P[1] 20% FINE
     2:  Search Start[2] PR[2]
     3:J  P[2] 50% FINE
     4:J  P[3] 50% FINE[X]
     5:J  P[4] 50% FINE
     6:J  P[5] 50% FINE Search[X]
     7:J  P[6] 50% CNT100
     8:J  P[7] 50% FINE
     9:J  P[8] 50% FINE Search[Y]
    10:J  P[9] 50% FINE
    11:J  P[10] 50% FINE Search[Y]
    12:  Search End
    13:J  P[1] 50% FINE
    14:  Touch Offset PR[2]
    15:L  P[11] 100mm/sec FINE
      :  Weld Start[1,1]
    16:L  P[12] 80cm/min CNT100
      :  COORD
    17:L  P[13] 80cm/min FINE
      :  COORD Weld End[1,2]
    18:  Touch Offset End
    19:J  P[1] 50% FINE
```

Teach Home Position (Leader/Follower groups)
Start Search Motion: Touch Sch 2, PR[2]
Teach a search start position(Leader/Follower)
Search to X direction (Only follower group)
Teach an another search start point(Leader/Follower)
Search to X direction (Only follower group)
Teach an intermediate point (Leader/Follower)
Teach an another search start point(Leader/Follower
Search to Y direction (Only follower group)
Teach an another search start point(Leader/Follower
Search to Y direction (Only follower group)
Finish Search Motion
Move to Home Position
Following points are offset by PR[2]
P[11] is offset by PR[2]

P[12] is offset by PR[2]

P[13] is offset by PR[2]

Finish offset of positions
Teach Home Position

Fig. 39.9(d) 2D + rotation search with coordinated motion

NOTE
When reference group setup is performed for coordinated motion, mastering information of all groups related to reference group are stored in TP program. If you change reference group on Touch Frame or Touch Schedule, please perform remastering.

Coordinated Touch Sensing with Simple Search

Touch Frame for Simple Search can select world frame of follower group or coordinated frame of leader group as basis.
If you select world frame of follower group as basis, enter the group number of follower group. If you select coordinated frame of leader group as basis, enter the group number of leader group. In this case, you must complete the calibration of coordinated pair. Additionally, absolute position data in position register by Search Motion is based on follower group.

In Touch Schedule for Simple Search, reference group number must be the same number as Robot Group Mask.
Following Fig. 39.9(e) is the program example of Simple Search with coordinated motion.

Group Mask of TP Program[1,1,*,*,*,*,*,*]

```
COORD_SEARCH2
                                        11/12
     1:J  P[1] 50% FINE
     2:  Search Start[2] PR[2]
     3:J  P[2] 50% FINE
     4:J  P[3] 50% FINE Search[X]
     5:J  P[4] 50% FINE
     6:J  P[5] 50% FINE Search[Y]
     7:  Search End
     8:J  P[6] 50% FINE
     9:L  PR[2] 100mm/sec FINE
      :  Weld Start[1,1]
    10:L  P[7] 80cm/min FINE
      :  COORD
    11:  Weld End[1,2]
```

Teach Home Position (Leader/Follower groups)
Start Search Motion, Touch Sch 2, PR[2]
Teach a search start position (Leader/Follower)
Search to X direction (Only follower group)
Teach an another search start point(Leader/Follower)
Search to Y direction (Only follower group)
Finish Search Motion
Teach an intermediate point (Leader/Follower groups)
Move the robot to PR[2]
Start Arc Welding
Coordinated motion

Fig. 39.9(e) Program example of simple search with coordinated motion

⚠CAUTION
Search Motions and Touch Offsets for coordinated motion cannot be mixed with non-coordinated motion. That is, you cannot do searches for coordinated motion, then do searches for non-coordinated motion, and then use the offset from the coordinated motion search. You must use the offsets for coordinated motion searches before doing any non-coordinated searches. Similarly, you must use any offsets from non-coordinated motion searches before doing any coordinated motion searches.

NOTE
The leader group is not allowed to move between the searches.

40 TOUCH SKIP FUNCTION

Touch skip function can watch the disturbance of any plural axes at the same time and when any axes' disturbance exceed threshold high-speed skip can be executed.
The threshold to judge the touch can be set in the special screen for touch skip function.

When touch skip function is ordered,
- "SETUP/ TOUCH SKIP" screen is able to be shown.
- Touch skip programs are able to be called.

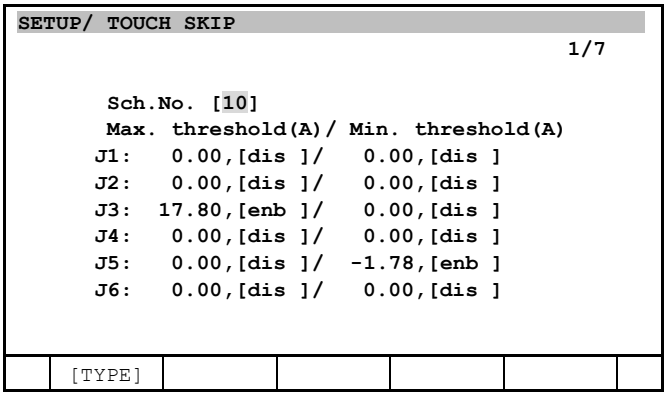
Note that system variable $MISC_MSTR.$HPD_ENB must be TRUE when Touch skip is used. (This variable is FALSE by default.)
If $MISC_MSTR.$HPD_ENB is not TRUE, please set it to TRUE by system variable screen or parameter instruction, and cycle power.

To use this function, Touch skip function option (A05B-2600-J921) is required.
Some robot models do not support this function.

40.1 TOUCH SKIP SCREEN

In touch skip screen, the conditions to judge the touch can be set.

- Show Touch skip screen by [MENU] key → "SETUP" → F1 [TYPE] → "Touch Skip".



- To change the schedule number, enter a value in "Sch. No." field.
- In lines below, set values and enable/disable of thresholds for disturbance of each axis.
  Check the disturbance during the motion by STATE → AXES → DISTURBANCE screen and set thresholds that is added margins if necessary.

In the example above, a condition of "J3 disturbance is larger than 17.8[A] or J5 disturbance is smaller than –1.78[A]" is set.

40.2 TOUCH SKIP PROGRAM

Following programs are able to be called.

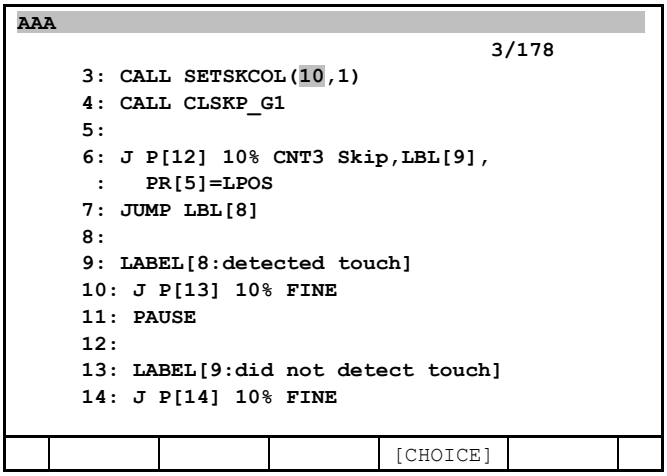
- SETSKCOL(s, g)

This is a program to set schedule No. [s], which is set on touch skip screen, to threshold of group [g].

- CLSKP_Gg

These are programs to set skip condition as "disturbance exceeds threshold in an axis of group [g]".

For example, you can use these programs as follows.



- In line3, schedule No. [10] which is set on touch skip screen is set on threshold of group [1].
- In line4, skip condition "disturbance exceeds threshold in an axis of group [1]" is set.
  Same as SKIP CONDITION instruction, this condition has been effective until program ends or another condition is set.
- A motion with high-speed skip instruction in line6 works all the same as high-speed skip instruction of standard function.
  When disturbance exceeds threshold in an axis of group [1], the robot stops and then the next line (line7) is executed.
  When disturbance does not exceed threshold in any axis of group [1] and the robot reaches destination point, the jump to the label [9] is executed.

40.3 CAUTIONS

- To decrease impact of stopping when skip condition is satisfied, speed limit is set for the motion with touch skip instruction. The speed limit is adjusted for each robot models. When taught speed exceeds the limit, speed limit works automatically. When speed limit works, the following warning is displayed.

  MOTN-560 Hspd.skip speed limit (G:group number)

- Some robot models decreased stop when high-speed skip instruction is executed. With such a robot, touch skip function can not be used. When touch skip function is executed with such a robot, the following alarm is displayed.

  MOTN-561 Touch skip NOT available (G:group number)

- Touch skip can not be used with skip instruction (Skip, LBL). When touch skip is executed with skip instruction (Skip, LBL), the following alarm is displayed. When you use touch skip function, you must teach high-speed skip instruction (Skip, LBL, PR).

  MOTN-561 Touch skip NOT available (G:group number)

41 MROT INSTRUCTION

MROT (Minimal Rotation) is a motion option to be used with the Joint motion type, or Cartesian motion with the WJNT option. It generates the shortest joint angle move for the wrist axes within axes limits during Joint and WJNT motion, and ignores the turn numbers of the wrist. For example, the start and destination angles are 150° and -130°, respectively. In the case, the shortest move is the motion from 150° to 230°. Because, 230° and -130° are the same position for the axis physically, 230° can be reached with less rotation.

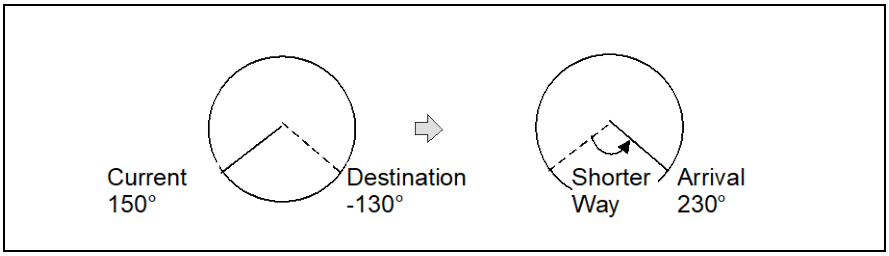


Fig. 41 (a) MROT instruction

To use this function, MROT Instruction options required.

41.1 HOW TO USE MROT

Add MROT instruction to Joint or WJNT motion.

J P[1] 100% FINE MROT
L P[2] 2000mm/sec CNT100 WJNT MROT

41.2 LIMITATIONS

The MROT motion option has following limitations.

- This function is supported only in Handling Tool.
- Before MROT can take effect, the destination position of the Joint or WJNT motion must be represented in Cartesian space. Otherwise, if the destination position is represented in Joint angles, the system will attempt to reach the specified destination joint angles regardless of the MROT option.
- MROT generates shortest joint move within axes limits. For example, the upper limit of wrist axis is 180°, and the start and destination angle are 150° and -130°, respectively. In this case, a warning massage "MOTN-330 MROT Limit Warn" will be posted and the axis will move from 150° to -130°. Because 230° is out of limit.

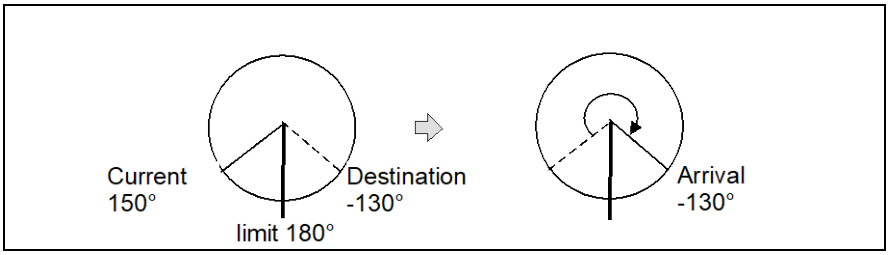


Fig. 41.2 Shorter way is out of limit.

- Single step Backward motion may not be the same as forward motion.

For example:
1: J P[1] 100% FINE
2: J P[2] 100% FINE MROT
3: J P[3] 100% FINE

If the actual reached joint angles at P[2] are different from the taught joint angles (in terms of turn number), then backward motion from P[3] to P[2] would be different from the forward motion from P[2] to P[3]. Because the backward motion from P[3] to P[2] is not MROT motion, the arrival position is taught position of P[2].

42 ROBOT ISOLATION FUNCTION

Robot Isolation function can make certain robot be isolated state by the flick of a switch when multiple robot is controlled by single controller. In teaching in the system with multiple robot, when just partial robots are taught, it is possible to keep the other robots unmoving. On this account, the user can have safety and convenience in teaching.

This function is usable if the hardware of the switch for Robot Isolation is attached. No software option is required.

42.1 SPECIFICATION

- Isolated state
  Robot turns into machine lock state by isolating. Servo power is turned off and robot comes into unmoving state. And the user can't teach new point to isolated robot. By unlocking isolated state, robot turns into normal state.

- Method of operating
  Isolation switch is prepared up to 4 on the operation panel. And by Isolation switch, the user can change certain robot state between isolated and connected. When robot is isolated, robot turns into machine lock state after alarm is displayed. And when robot is connected, robot turns into normal state after alarm is displayed.

NOTE
1 If robot state is changed from isolated to connected in power-off, robot remains in machine lock state after power is turned on. In this instance, it is needed to change robot state by the Teach Pendant to be normal state.
2 If robot state is changed from connected to isolated in power-off, robot turns into machine lock state during power up after power is turned on.

- Setting
  Each Isolation switch corresponds certain motion group. User can see correspondence between Isolation switch and motion group by system variable $ROBOT_ISOLC.

42.2 METHOD OF OPERATING

42.2.1 Operation Panel

As is often the case, Isolation switch is mounted on Operation panel.

By turning switch ENABLE side, robot become operable.
By turning switch DISABLE side, robot become isolated.

42.2.2 DCS Safety Signal: RPI

If DCS Safe I/O connect function (J568) is ordered, DCS safety signal: RPI is usable.

It is possible to see the state of Isolation switch by RPI[1-4].

RPI is ON: Isolation switch is ENABLE side (connected).
RPI is OFF: Isolation switch is DISABLE side (isolated).

Isolation switch 1 - RPI[1]
Isolation switch 2 - RPI[2]
Isolation switch 3 - RPI[3]
Isolation switch 4 - RPI[4]

42.3 SYSTEM VARIABLE: $ROBOT_ISOLC

Depending on ON/OFF of each Isolation switch, corresponding motion group state become normal / machine lock. Correspondence between Isolation switch and motion group is decided by system variable $ROBOT_ISOLC.
Which robot's power current to servo is turned ON/OFF by Isolation switch is decided by hardware configuration. Therefore, it is decided uniquely which motion group should be machine lock state by software when certain isolation switch is turned.
So $ROBOT_ISOLC is needed to be set just one time when robot system is established.

$ROBOT_ISOLC have 4 alignment, and each one corresponds one Isolation switch. If the number of Isolation switch (abbreviated as SW) is smaller than 4, $ROBOT_ISOLC alignments are available as many as the number of Isolation switch, and the other alignments is ineffective.

SW1: $ROBOT_ISOLC[1]
SW2: $ROBOT_ISOLC[2]
SW3: $ROBOT_ISOLC[3]
SW4: $ROBOT_ISOLC[4]

The each bit of $ROBOT_ISOLC value is equivalent to each motion group.

Bit31	・・・	Bit1	Bit0
Motion Group32	・・・	Motion Group2	Motion Group1

For example, the default values of $ROBOT_ISOLC are set as the following, it is assumed that one Isolation switch is equivalent to one motion group.

Example1 (Default)

SW1: $ROBOT_ISOLC[1] = 1 [Motion group 1]
SW2: $ROBOT_ISOLC[2] = 2 [Motion group 2]
SW3: $ROBOT_ISOLC[3] = 4 [Motion group 3]
SW4: $ROBOT_ISOLC[4] = 8 [Motion group 4]

In case that SW1 is equivalent to motion group 1, 3,
and SW2 is equivalent to motion group 2, 4,
$ROBOT_ISOLC value is the following.

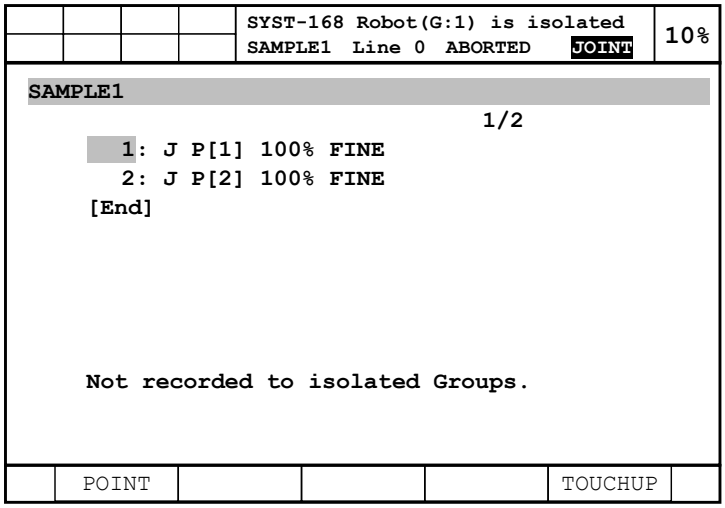
Example 2

SW1: $ROBOT_ISOLC[1] = 5	[Motion group 1,3]
SW2: $ROBOT_ISOLC[2] = 10	[Motion group 2,4]
SW3: $ROBOT_ISOLC[3] = 0	
SW4: $ROBOT_ISOLC[4] = 0	

42.4 TEACHING TO MOTION INSTRUCTION

When the user teaches motion instruction, current position isn't recorded for isolated robot.
And when teaches target position again by [SHIFT] key + F5 "TOUCHUP", position isn't recorded similarly.

Then, the message that informs isolated group can't be taught is displayed.



43 ANTI-DEFLECTION FOR EXTERNAL FORCE

Anti-Deflection for External force suppresses the replacement of TCP due to external force for application which works under external force. For example, this function is usable for tool which drills a hole by pressing the work with a cylinder and fixing the robot to the work in advance. This function compensates the replacement due to reaction force in which presses the work with a cylinder, and improves accuracy of drilling. To use this function, Anti-Deflection for External force function option is required.

NOTE
Only M-900*i*B/700 is supported.

When robot is acted by external force, TCP is displaced by RV reducer torsion and so on. This function estimates displacement of TCP and compensates it.

NOTE
No special hardware is required.

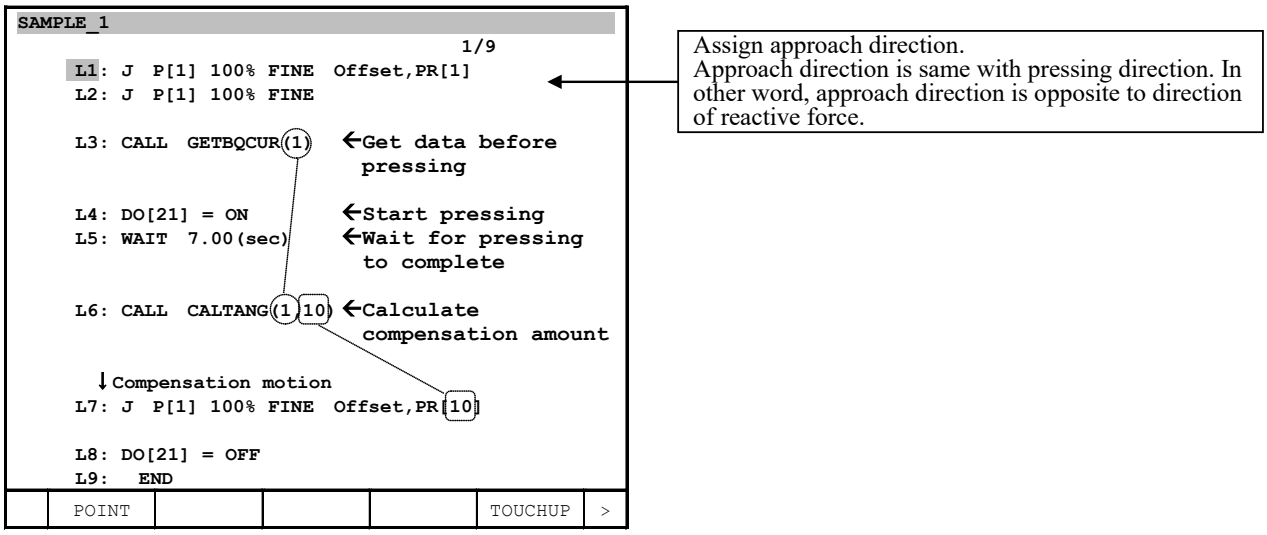
43.1 PREPARATION TO CREATE COMPENSATION PROGRAM

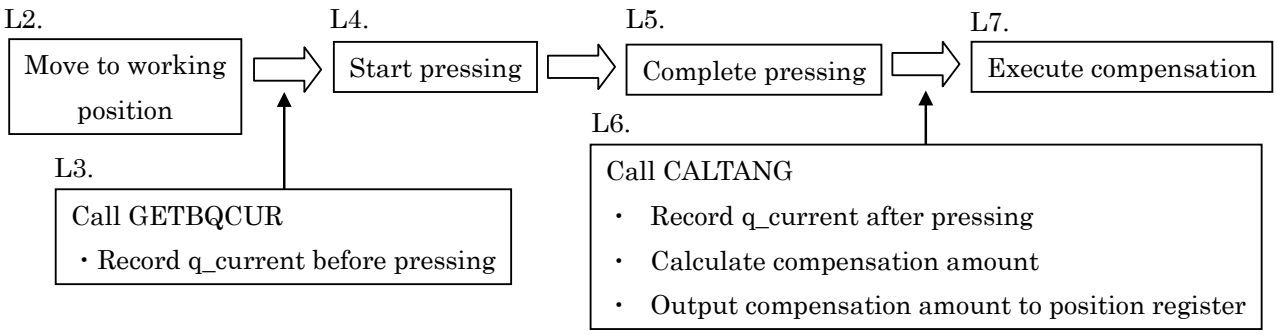
Two KAREL programs, GETBQCUR and CALTANG, are loaded if this function is ordered. Compensation amount is calculated by executing GETBQCUR before being acted by external force and executing CALTANG after being acted by external force. Please enter following arguments to these KAREL programs.

GETBQCUR (n)
n: Group number of target robot.
CALTANG (n, m)
n: Group number of target robot.
Set same number with group number assigned at GETBQCUR.
m: Position register number for output compensation amount

43.2 HOW TO COMPENSATE

This section shows sample program and flow.





Following explanation is details for each line.

L1: Specify approach direction to approach from constant direction. Approach direction is same with pressing direction. In other word, approach direction is opposite to direction of reactive force.
L2: Move to working position.
L3: Call KAREL program GETBQCUR and get data before pressing.
L4: Start pressing.
L5: Wait for the completion of pressing.
L6: Call KAREL program CALTANG and get data after pressing and calculate compensation amount of each axis. Compensation amount is output to assigned position register with JOINTPOS type.
L7: Compensate using direct offset condition instruction. Please assign position register number assigned at argument2 of CALTANG.
-----Task to the application -----
L8: Finish pressing
L9: Finish compensation sequence

⚠ CAUTION
Data type of compensation amount output at position register is JOINTPOS.

43.3 SETUP CUSTOMIZATION

It is possible to change following variables from KAREL variable screen. The method to display KAREL variable is following.

Step

1 Press the [MENU] key.
2 Press [0 NEXT] and select [6 SYSTEM].
3 Press the F1 [TYPE], and select [SYSTEM Variable] from the menu.
4 Place the cursor on $KAREL_ENB, then key in "1" and press [ENTER] key.
5 Select CALTANG of KAREL program in SELECT screen.
6 Press the [DATA] key.
7 Press the F1, [TYPE] and select "KAREL Vars".

Please refer to Subsection 9.21.6 of B-83284EN Operator's manual (Basic operation) for details of KAREL variable screen.

Name	Type	Explanation
offset_mode	INTEGER	Add offset to compensation amount with following type. offset_mode = 1 :XYZWPR offset_mode = 2 :JOINT (Default : 0)
offset	ARRAY[MAX_GRP_AXES] OR REAL	Offset amount. If XYZWPR type is selected (offset_mode=1), then offset[1] to offset[6] correspond to such as X:offset[1], Y:offset[2], … , P:offset[5],R:offset[6]. If JOINTPOS type is selected (offset_mode=2), then offset[1] to offset[6] correspond to such as offset[axis number]. Offset amount is calculated based on assigned frame. (Default : 0)
utool_num	INTEGER	Tool frame number. (Default : Selected number)
ufram_num	INTEGER	User frame number. (Default : Selected number)
max_comp_deg	REAL	high limit of compensation amount (Default : 3.0)
spr_const	ARRAY[6] OF REAL	Spring constant of each axis. There is no need to change this variables unless the circumstances are exceptional. (Default : Tuned value)
backlush	ARRAY[6] OF REAL	Backlush of each axis. There is no need to change this variables unless the circumstances are exceptional. (Default : Tuned value)
margin	REAL	Final compensation amount of each axis is given by multiplying compensation amount by margin. (default : 1.0)

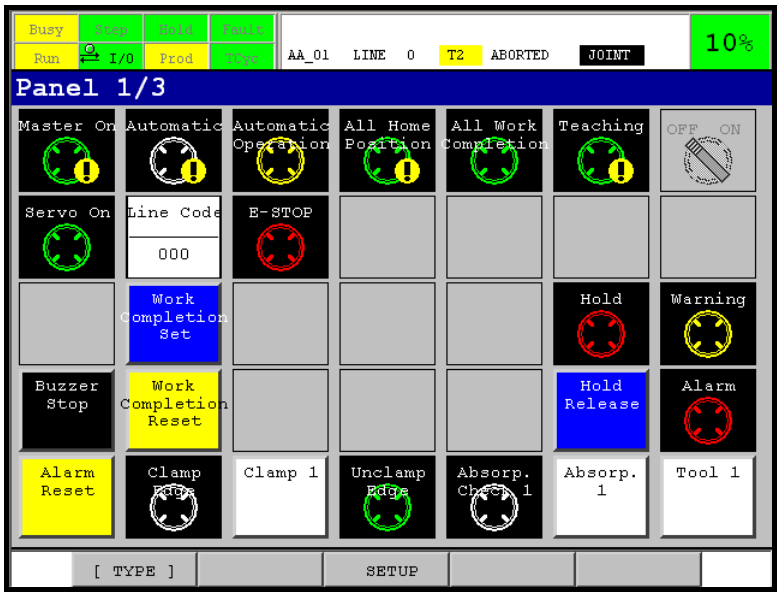
⚠CAUTION
When restarting controller, listed above KAREL variables return to uninitialized variables. These variables return to default when KAREL program CALTANG is executed. If you want to use variables before restarting, please reconfigure it.

43.4 RESTRICTIONS

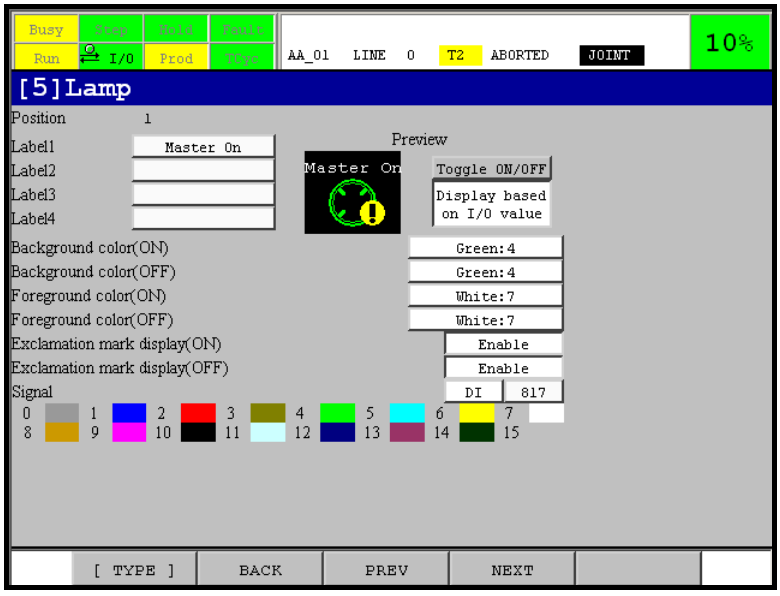
(1) It does not need to calculate compensation amount each time if the condition and motion is thoroughly same. However, it is needed to calculate compensation amount periodically because compensation amount changes depending on temperature and so on.
(2) It is needed to calculate compensation amount in the case where position is changed or approach direction is changed.
(3) Accuracy of position at TCP is influenced by tool rigidity. If tool rigidity is low, that might prevent to suppress the replacement of TCP.
(4) If offset_mode = 1, then this function does not work near singular points. In this case, 'Convert offset fail' is displayed on status window.

44 INTERFACE PANEL FUNCTION

This is a function to create panel similar to following figure. You can place buttons on grid like panel. The number of panels is 3 by default. You can create up to 10 panels. Please refer to Section 44.16 for change of the number of panels.



You can setup buttons by setup screen. Labels and signals for example.



You can put buttons on panel to display and change value of I/O.
Please refer to Section 44.7 ”TYPE OF BUTTON” for type of buttons.
This function can use button (which can be pressed) and lamp (which cannot be pressed).
In this document both button and lamp are often referred as button.

Hardware Requirement

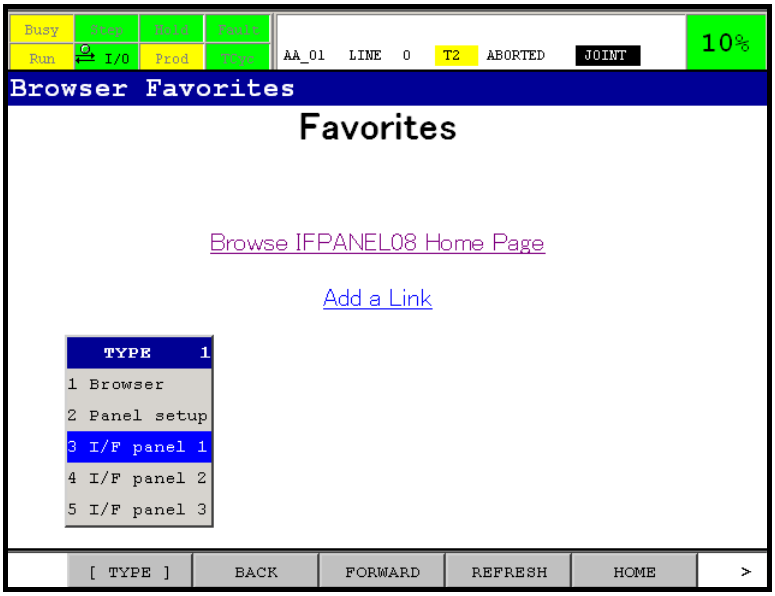
*i*Pendant with touch panel is required.

Software Requirement

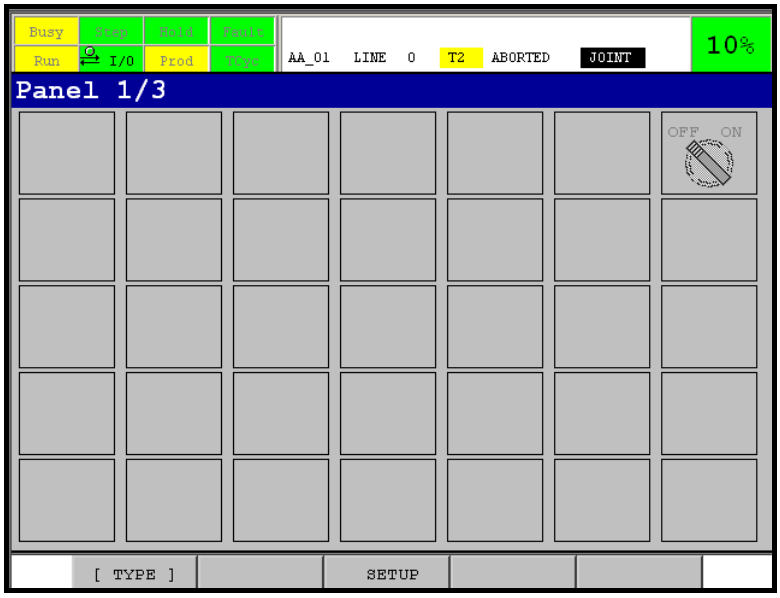
This function is optional (J741). This function is supported by system software version of 7DC2/02 or later.

44.1 HOW TO DISPLAY PANEL

1 Press the [MENU] key.
2 Select BROWSER.
3 Press the F1 [TYPE].
With default setting, you can see “I/F panel 1”, “I/F panel 2” and “I/F panel 3” in pull up menu. They are page 1, 2 and 3 of panel of this function. Select page to display.

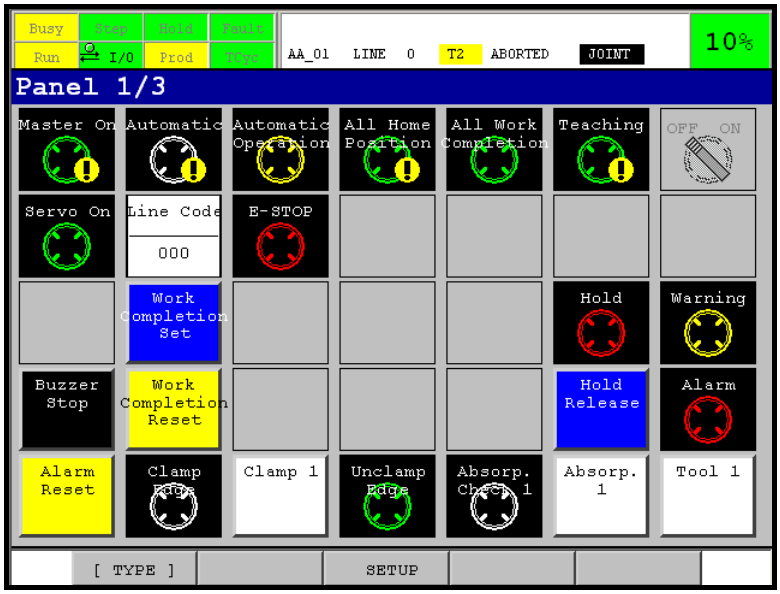


At first, there is no buttons on panels. You have to setup buttons to place them on panels.
Press the F3 to display button type setup screen. You can select position you place button and type of button.
Properties of buttons are setup in button detail setup screen.
For button type setup screen and button detail setup screen, please refer to Section 44.4 “BUTTON TYPE SETUP SCREEN” and 44.6 “BUTTON DETAIL SETUP SCREEN” respectively.



44.2 INTERFACE PANEL

Interface panel is operator's panel displayed on teach pendant. You can put lamps and buttons to display and turn ON/OFF I/O signal. Please refer to Section 44.7 for types of buttons.



This section explains elements that compose panel and condition to press buttons.

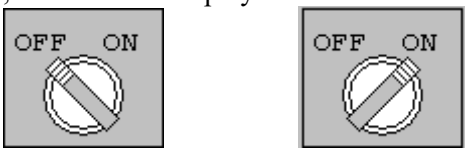
Page number

It is number displayed on title bar. In the figure above, title bar displays 1/3. It means this is the 1st page out of 3 pages.

Enable switch

This is a button that is on the 1st row and the last column (right upper corner). By default setting, you have to turn on this button to press any button on panel.
When the [SHIFT] key is not pressed, the button is displayed like the following figure.

When the [SHIFT] key is pressed, the button is displayed like the following figure.



Panel enable switch is turned off when screen is changed. It is also true when page of pane is changed.

Conditions to press button

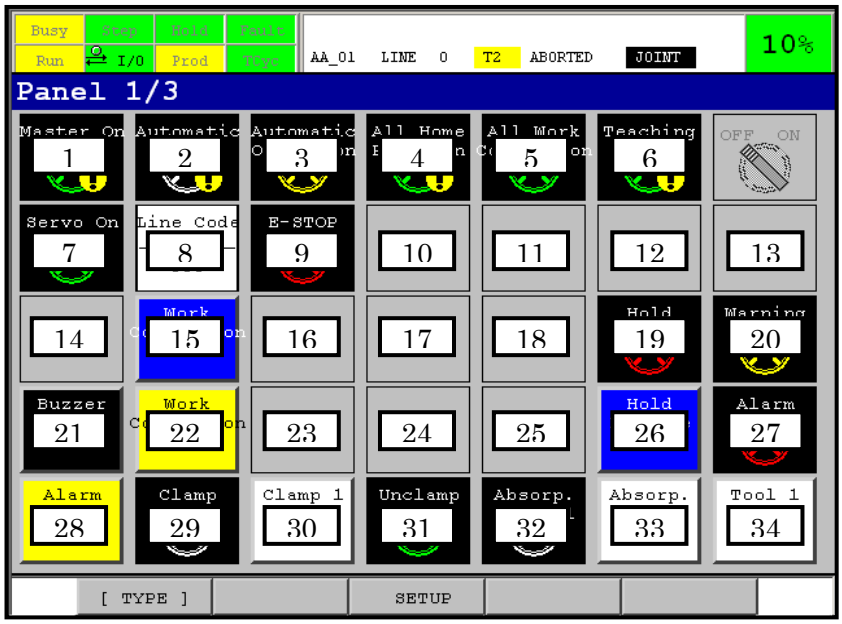
Condition to press buttons can be setup for each button. The condition is setup in operation condition setup screen. By default, following conditions have to be satisfied to press a button.

- Enable switch of Interface panel is ON
- The [SHIFT] key is pressed.

Please refer to section 44.9, “OPERATION CONDITION” for detail of operation condition and its setup screen.
By default setting, operation condition of momentary button is checked only when it is pressed. Lamp does not have operation condition setup screen because it cannot be pressed.

Button

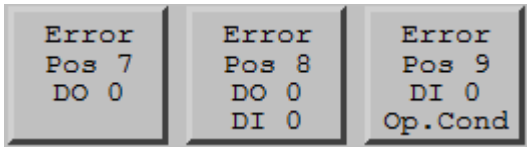
Interface panel is grid like panel. On each tile, button can be placed. Each tile has position number.
Following figure shows position numbers of the 1st page of 7 column panel.



When you setup a button, you have to decide where to place it by selecting position number. Then type of the button on the position must be set. After the button is setup by detail screen, the button is displayed on panel on the position.
For setup, please refer to Section 44.3 ”INTERFACE PANEL SETUP SCREEN", 44.4 ” BUTTON TYPE SETUP SCREEN" and 44.5 ”SETTING OF TYPE OF BUTTON”.

Error button

If setup is wrong, button is displayed as error button. They are similar to the following figure.



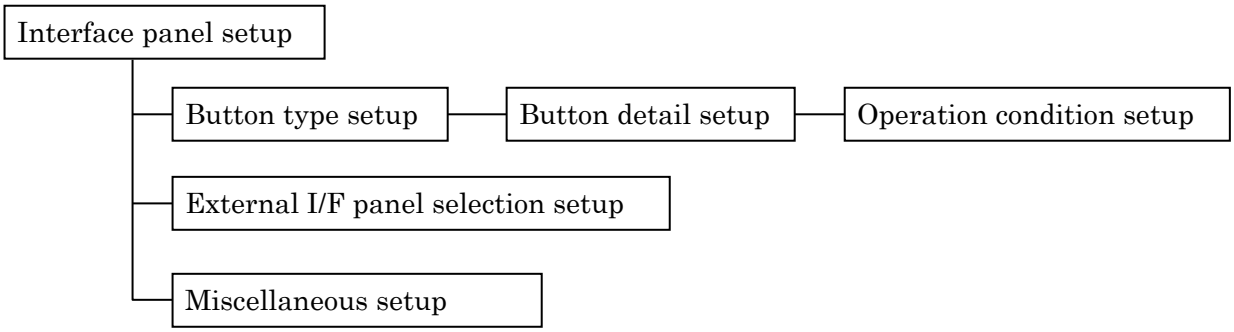
- The button displays position number and I/O that can not be read.
- If the button has 2 signals and both cannot be read, both signals are displayed.
  Button type 3 (2 contact point switch) displays only the 1st signal if index of the 2nd signal is 0.
- Button type 6 and 7 (digital switch and digital lamp) displays only the 1st found signal that cannot be read.
- If signal is specified and enabled as operation condition, and if the signal cannot be read, error button is displayed
  Signal for actual operation (output signal for example) is also wrong, it has priority.
- Press of the button displays detail setup screen.

F3 SETUP

Pressing the function key displays button type setup screen of the same page. For button type setup screen, please refer to section 44.4, “BUTTON TYPE SETUP SCREEN”.

44.3 INTERFACE PANEL SETUP SCREEN

Panel is setup by Interface panel setup screen. It consists of following screens.



Following table is overview of each screen.

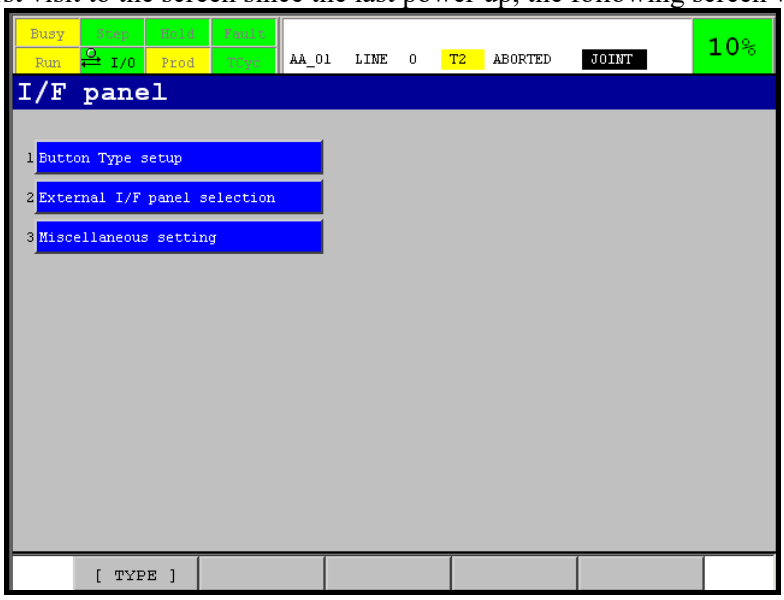
Screen	Description
Interface panel setup screen	The top level screen. This screen bundles the other screens in the table.
Button type setup screen	Type of buttons of each position number is set. You can copy and paste button in this screen.
External I/F panel selection setup screen	Digital input signals to display each page is setup
Misc. setup screen	Various setup items that affect all pages. The number of columns and pages are included.
Button detail setup screen	Various properties of buttons are set here.
Operation condition setup screen	This screen setup condition to press a button. This is part of button detail setup screen. Only buttons that can be pressed has this screen.

NOTE

Panel is setup for each language separately. When language is changed, setup for new language is used. Suppose that you setup panel in English only. If you change language to Japanese, there are no buttons on panels for Japanese. Setup in External I/F panel and Misc. setup screen don’t depend on language.

You can display Interface panel setup screen by the following procedure.

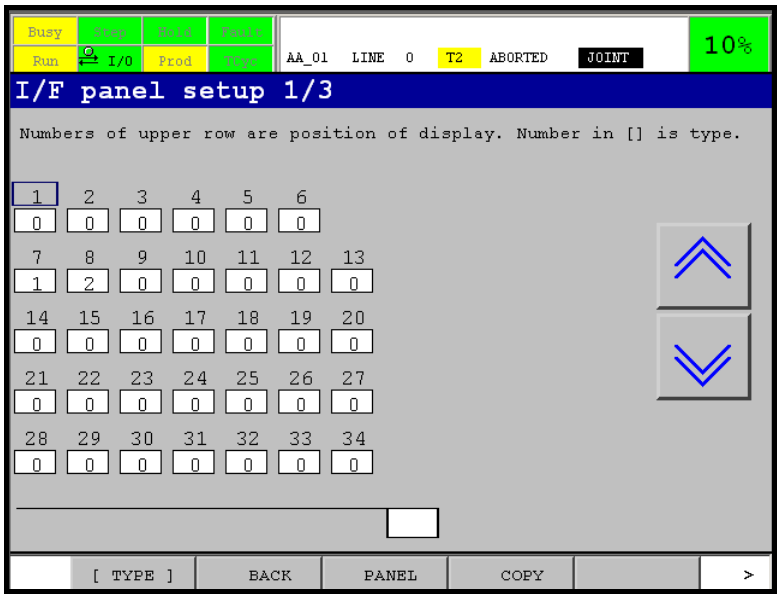
1 Press the [MENU] key.
2 Select SETUP.
3 Press the F1 [TYPE].
4 Select I/F panel setup.
If this is the first visit to the screen since the last power up, the following screen will be displayed.



Press of each button leads you to each setup screen.

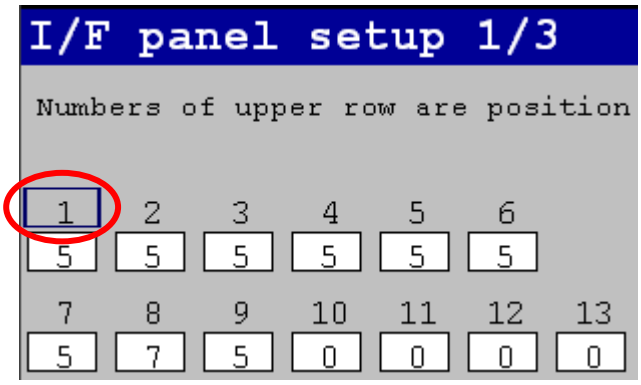
44.4 BUTTON TYPE SETUP SCREEN

This screen decides type of buttons. It is screen similar to the following figure.
Press of “Button type setup” button of Interface panel setup screen displays the screen.



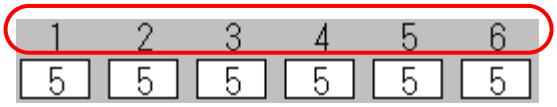
Numbers in white boxes show the type of button of the position. Number just above each white box is position number.
Press of arrows on the right side of the screen changes pages.
The arrows are displayed when cursor is on a position number.

Cursor

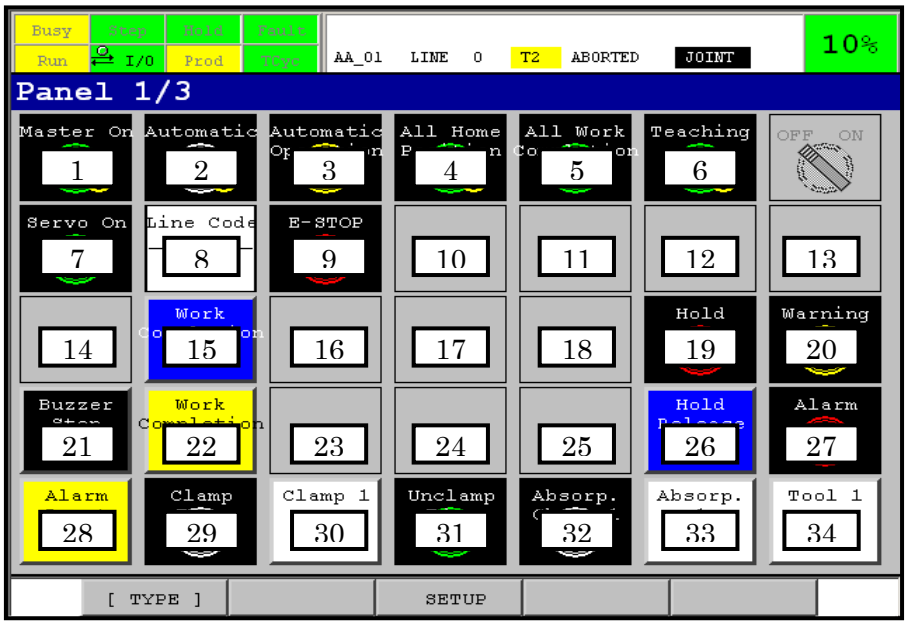


In the figure above, blue rectangle that encloses “1” is cursor. You can move the cursor by arrow keys.

Position number

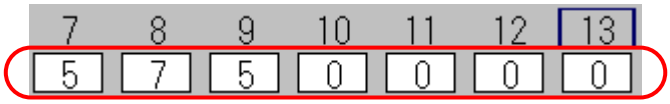


On the left side of the screen, there are many numbers. White box and the number above it are pair. Position numbers are the number above white boxes. When cursor is on a position number, arrow buttons to change page are displayed on the right side of screen. When cursor is on a position number, shift+ up arrow or down arrow key can be used to change pages.
Following figure shows position number of the 1st page of 7 column panel.

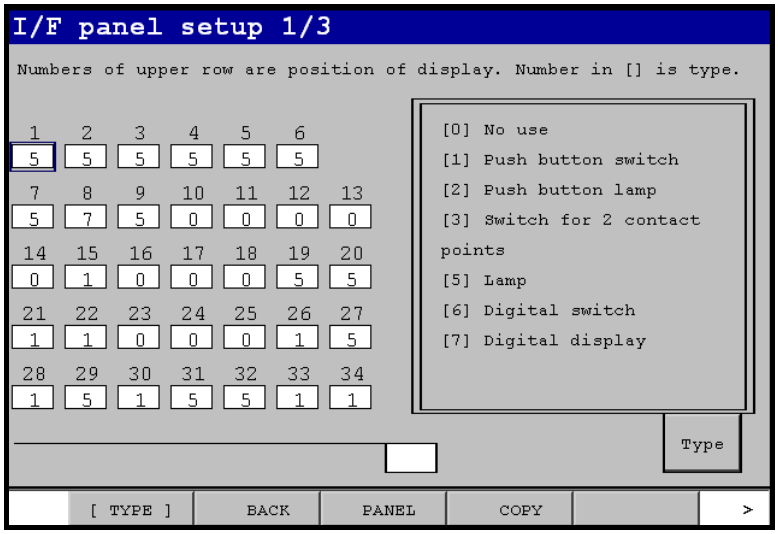


Likewise, left upper corner of the 2nd page is position number 35. Right lower corner corresponds to 68. You can display button detail setup screen if you press the [ENTER] key when cursor is on a position number.

Type code field



This is white box on the left side of the screen. The box displays button type of corresponding position number, which is displayed above each white box. In the figure above, position 7 and 9 are type 5 (lamp). Position 8 is type 7 (digital lamp). From position 10 to 13 are type 0. It means the positions are not used. When cursor is on type code field, explanation of type code is shown on the right side of the screen.

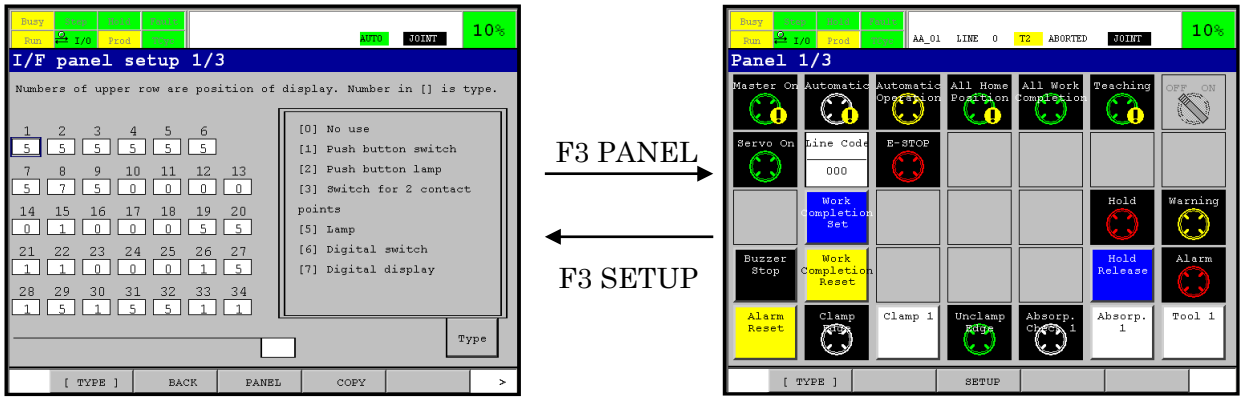


F2 BACK

The highest level screen, Interface panel setup screen is displayed.

F3 PANEL

By pressing the F3 PANEL, panel of the same page is displayed.



F3 SETUP of panel displays back to button type setup screen of the same page.
Even in this case, F2 BACK always displays interface panel setup screen (the highest level screen).

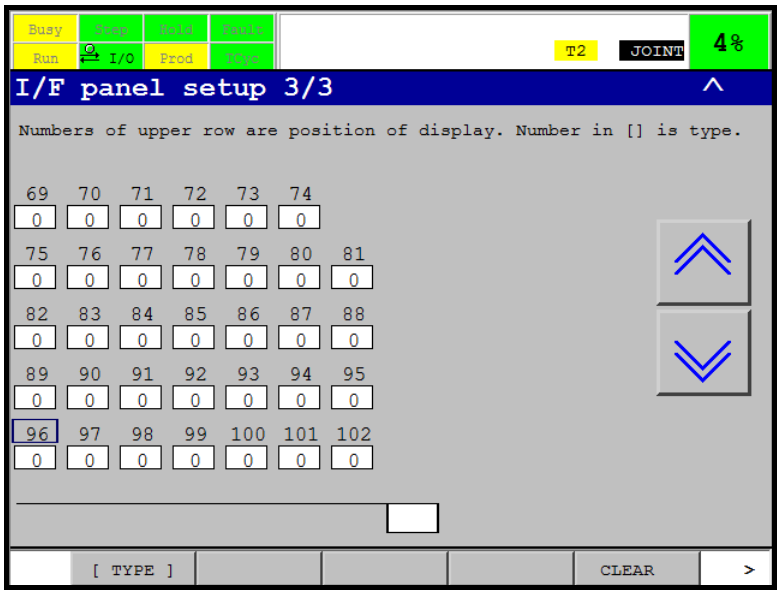
F4 COPY

You can use this function key to copy a button. Please refer to Subsection 44.5.1 for more details.

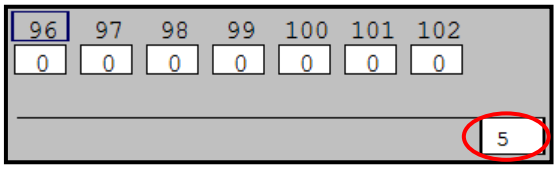
44.5 SETTING OF TYPE OF BUTTON

Type of button is setup by following procedure.

1 Display button type setup screen.
2 Display page you want to setup.
3 Move cursor to position number or type code field (white box) of the position.

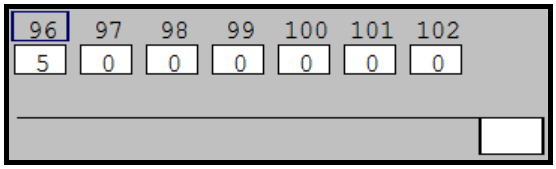


4 Input type code.



Type code you are inputting is displayed in white box at the bottom of the screen.

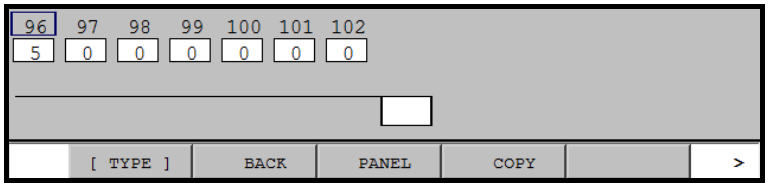
5 Press the [ENTER] key to complete input.
It is reflected to type code field of the position you selected.



44.5.1 Copy and Paste

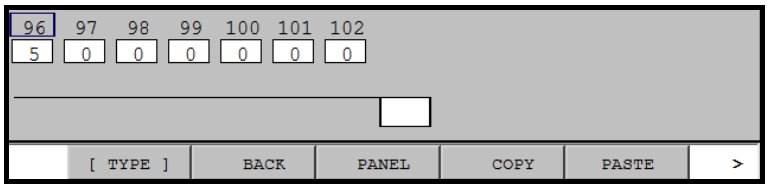
You can copy and paste a button in button type setup screen.

1 Display button type setup screen.
2 Display the page that includes button to copy.
3 Move cursor to position number or type code field of the button.



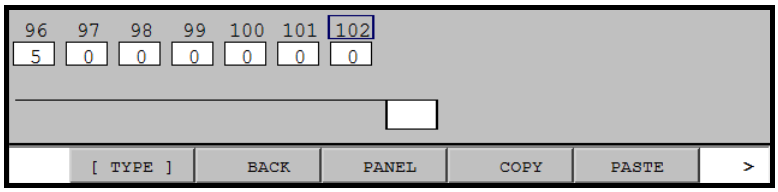
In the figure above, the button of position 96 (type 5, lamp) is copied

4 Press the F4, COPY.

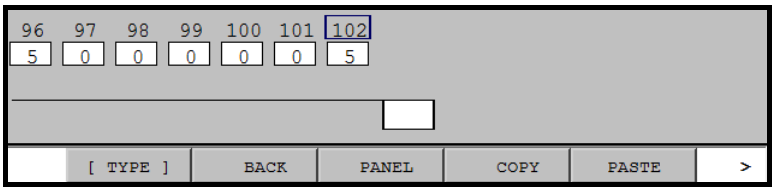


F5 PASTE is displayed. If you have copied another button before, PASTE must be already displayed before you press the F4 copy. Copy data is overwritten to previous data.

5 Move cursor to position to paste.



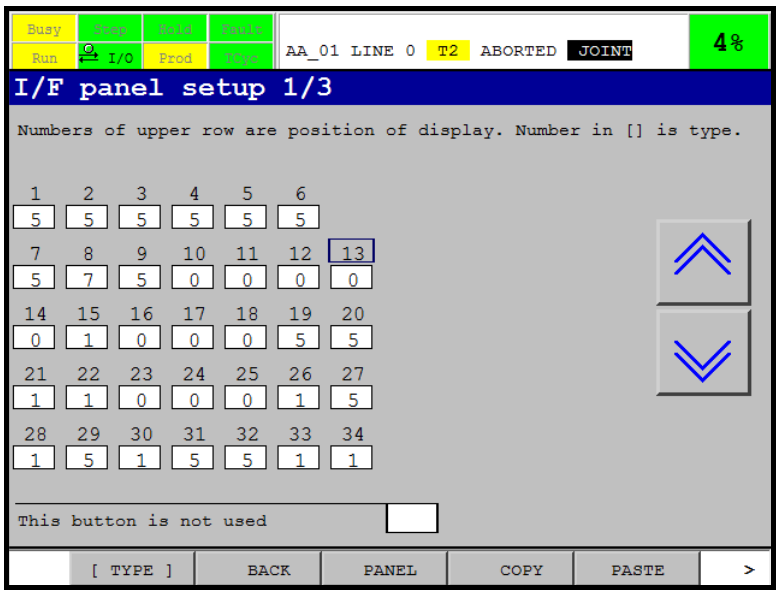
6 Press the F5, PASTE.



The button was copied. You can paste the same data repeatedly.

NOTE

You cannot copy button of type 0 (not used). Warning message is displayed as following figure. When you set type code to 0, input 0 one by one, confirming if the button can be cleared.

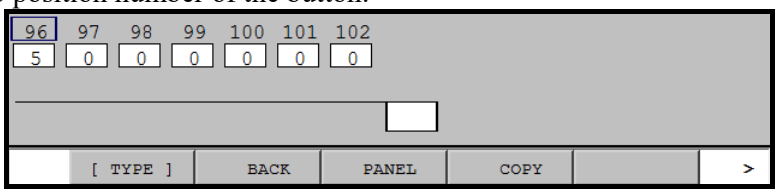


You cannot paste if current language is different from the one selected at copy. Suppose that you copied a button when English was selected. You cannot paste the data if the other language is selected.

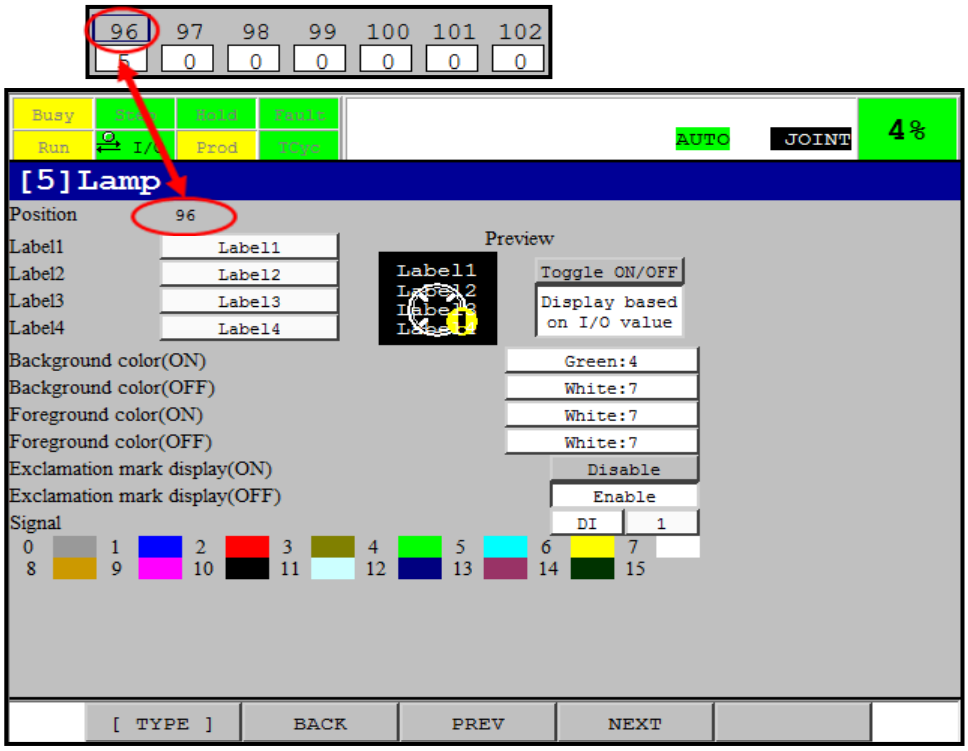
44.6 BUTTON DETAIL SETUP SCREEN

Button detail setup screen is a screen to setup various property of a button. To display the screen please use following procedure.

1 Display button type setup screen.
2 Change pages to display panel that includes the button to be setup.
3 Move cursor to position number of the button.



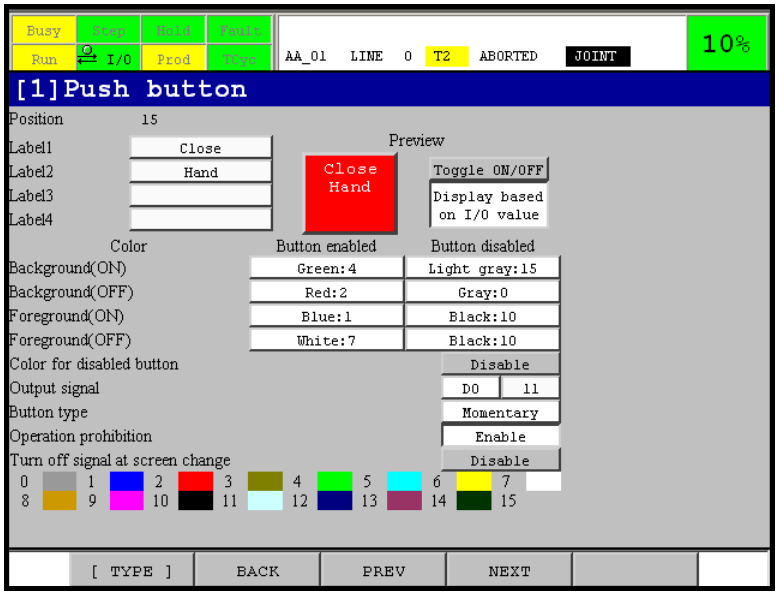
4 Press the [ENTER] key.
Screen for the button on the position (position number) is displayed. According to type of button, proper screen is displayed. In the following figure, screen for type 5 (lamp) is displayed.



NOTE
After change of button type, index of input or output signals are defaulted to 0.
Preview of button always shows error.

44.6.1 Preview of Button

Button detail setup screen has preview of button. You can get a rough idea how the button actually looks on panel. Some items cannot be reflected to preview. They are mainly items to decide action of button when it is pressed. This subsection explains overview of preview.



“Toggle ON/OFF” button

The button exists in detail screen of buttons that has ON/OFF status. They have colors for ON and OFF state.

The button toggles “ON” state and “OFF” state of button without changing actual signal the button should refer to. The button works only when “Display based on I/O value” button is OFF.

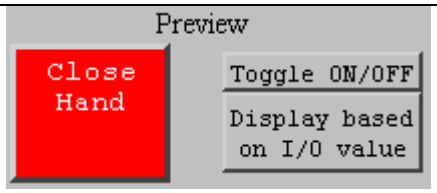
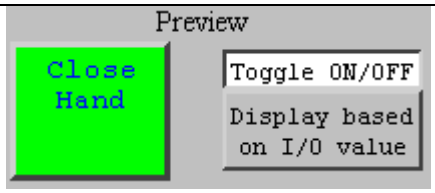
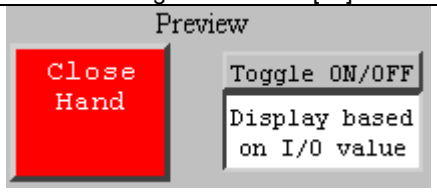
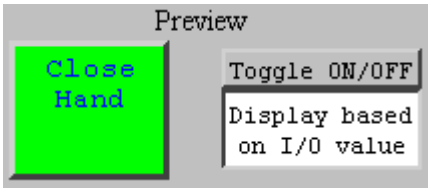
In case of 2 contact switch (type 3), left/right or up/down state are toggled.

“Display based on I/O value” button

When the button is ON, preview is displayed based on current status of I/O specified. After this button is pressed, “Toggle ON/OFF” button is turned off automatically.

Based on the setup in the previous figure, we explain how the button works. DO [11] is specified as output signal. When panel is actually displayed, button is displayed based on DO [11].

The following table shows how “Toggle ON/OFF” and “Display based on I/O value” button change preview.

“Display based on I/O value”	“Toggle ON/OFF” button” is OFF	“Toggle ON/OFF” button” is ON
OFF (“Toggle ON/OFF” button decides status of preview button.)	 "OFF" state regardless of DO[11]	 "ON" state regardless of DO[11]
ON	  Status of DO [11] decides status of preview button.	This status doesn’t exist. You cannot turn on both “Toggle ON/OFF” button and “Display based on I/O value” at the same time. If you press one of them when the other is ON, the latter is turned off automatically.

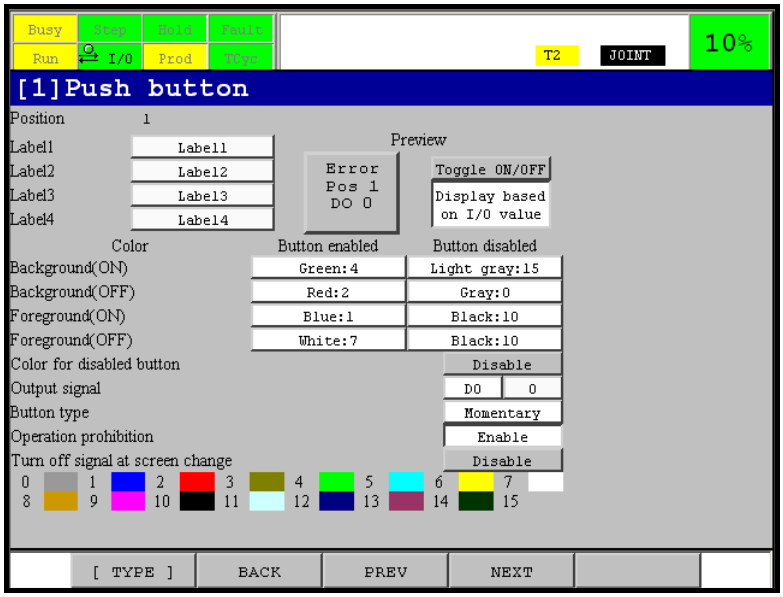
NOTE
You cannot press a button displayed as preview of button detail setup screen. The button doesn't respond.

Color for disabled button

A few types of buttons have setup item, "Color for disabled button". If the item is set to "Enable", color of button is changed according to whether condition to press button (operation condition) is satisfied or not. You can compare the two looks by preview. In preview, panel enable switch is treated as if it is always ON in operation condition check.
When "Color for disabled button" is enabled and operation condition is default setting, preview button is displayed in color of "Button enabled" column when [SHIFT] key is pressed.

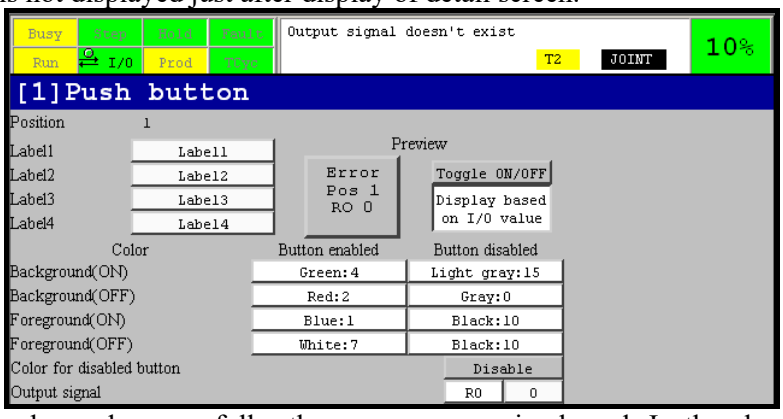
44.6.2 Error Display of Button

When "Display based on I/O" is pressed and specified signal cannot be read, preview of detail screen displays error button.



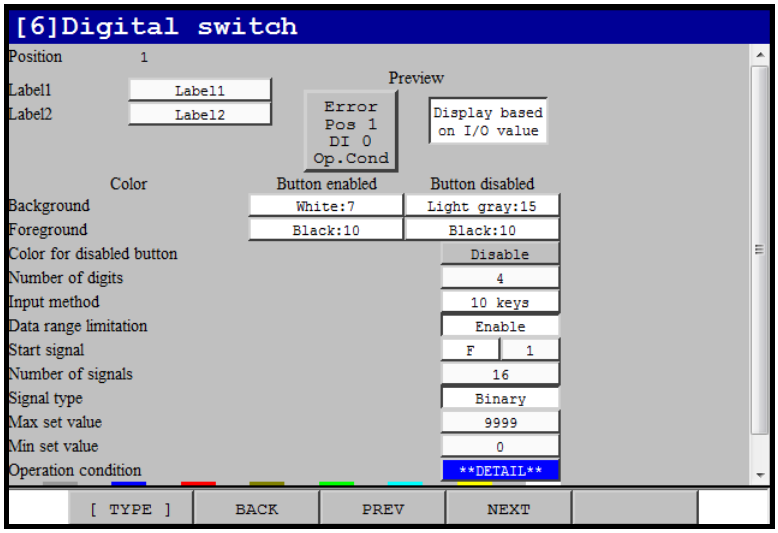
When "Display based on I/O" button is not pressed, preview is not displayed as error button.

When setup item is changed but the changed item still causes error, message is displayed on error line. This error message is not displayed just after display of detail screen.



When setup item is changed successfully, the error message is cleared. In the above figure, any color change would clear error message on error line. However, preview still shows error button.

When operation condition signal cannot be read, preview displays error button. Signal for actual operation (output signal for example) is also wrong, it has priority.



In the figure above, error button shows DI 0 is wrong. But DI[0] is not specified in the screen. It is specified in operation condition setup screen. When error button is displayed, please check operation condition screen, too.

44.7 TYPE OF BUTTON

This function can use following button and lamp.

- Push button (type 1)
- Push button lamp (type 2)
- 2 contact points switch (type 3)
- Lamp (type 5)
- Digital switch (type 6)
- Digital lamp (type7)

Overview of properties that is common to some button types are explained in the next section.

44.8 COMMON PROPERTIES

Label

This is string shown on button. The number of labels depends on type of button.
They are shown as Label1,…, Label4 in button detail setup screen. Label n is displayed at the nth row of the button.
When panel has 7 columns, each label should have less than 10 characters.
If the number of column is 8, less than 8 characters.

Foreground color and background color

Following type of buttons have two sets of colors. “Button enabled” and “Button disabled” column of button detail setup screen.

- Push button (type 1)
- Push button lamp (type 2)
- 2 contact points switch (type 3)
- Digital switch (type 6)

Following figure is extract of button detail setup screen of type 1, 2 and 3.

Color	Button enabled	Button disabled
Background(ON)	Green:4	Light gray:15
Background(OFF)	Red:2	Gray:0
Foreground(ON)	Blue:1	Black:10
Foreground(OFF)	White:7	Black:10
Color for disabled button		Disable

Digital switch doesn't have ON/OFF status. There are only Foreground, Background and color for disabled button.
Colors of "Button enabled" column are for when button can be pressed (when button is enabled).
"Button disabled" column is for when button can NOT be pressed (when button is disabled).
Please refer to section 44.9 "OPERATION CONDITION" for conditions to press a button.

Item	Description
Background(ON)	Background color for when signal is ON. It depends on button type which signal decides color. Left column is for when button is enabled (when button is enabled). Right column is for when button is disabled.
Background(OFF)	Background color for when signal is OFF.
Foreground(ON)	Foreground color for when signal is ON.
Foreground(OFF)	Foreground color for when signal is OFF.
Color for disabled button	Enable Color of button Is changed according to whether button can be pressed or not. Disable Colors of "Button enabled" column are always used.

Signal	Button is enabled	Button is disabled
ON		
OFF		

NOTE
If "Color for disabled button" is disabled, colors of "Button enabled" column are always used.

Turn off signal at screen change

If this item is enabled, specified output is turned off at following event
- When panel is displayed
- When screen is changed to another screen (This includes page change between panels)
- Any operation that re-displays panel (Refresh pane of FCTN menu, Zoom for example)

Supported signal type

Buttons of this function supports digital type I/O only. I/O that has integer value are not supported. For example, analog I/O and group I/O are not supported. Digital switch and lamp treat a series of digital signal as a group and treat them as numeric value.
Following I/O types are supported by output signal of push button (type 1), output signal of push button lamp (type 2), both signals of 2 points contact switch (type 3) and signals of digital switch (type 6).
- DO

- RO
- F (Flag)

Status signal of push button lamp (type 2), signal of lamp (type 5) and signals of digital lamp (type 7) support following I/O types.

- DI
- DO
- RI
- RO
- SI
- SO
- TPOUT
- UI
- UO
- F (Flag)
- M (Marker)

TPOUT[1-8] corresponds to status of software LED, which is at left upper corner of iPendant screen. For software LED, please refer to "Teach Pendant" subsection of OPERATOR'S MANUAL (Basic Operation) (B-83284EN).

For Flag and Marker, please refer to section of "BACKGROUND LOGIC" of OPERATOR'S MANUAL (Basic Operation). Flag and Marker are explained in subsection of "Other Instructions and Functions".

Operation condition

This item exists in detail screen of button type that can be pressed. You can display operation condition setup screen by pressing detail button on the row of operation condition. Please refer to section 44.9 OPERATION CONDITION for more information.

44.9 OPERATION CONDITION

Interface panel function can specify condition to press buttons on panel.

- Conditions are specified by combining following keys and I/O.
  - Panel enable switch
  - [SHIFT] key
  - TP enable switch
  - DEADMAN switch
  - Signal specified by user is ON or OFF
- Conditions are specified for each button.
- If a button is pressed when its operation condition is not satisfied, warning is posted.
  Some button may look the button was successfully pressed, but the function of the button does not work. Output signal is not turned on for example.

Signal specified by operation condition setup screen is referred as operation condition signal.

44.9.1 Button Types Supported

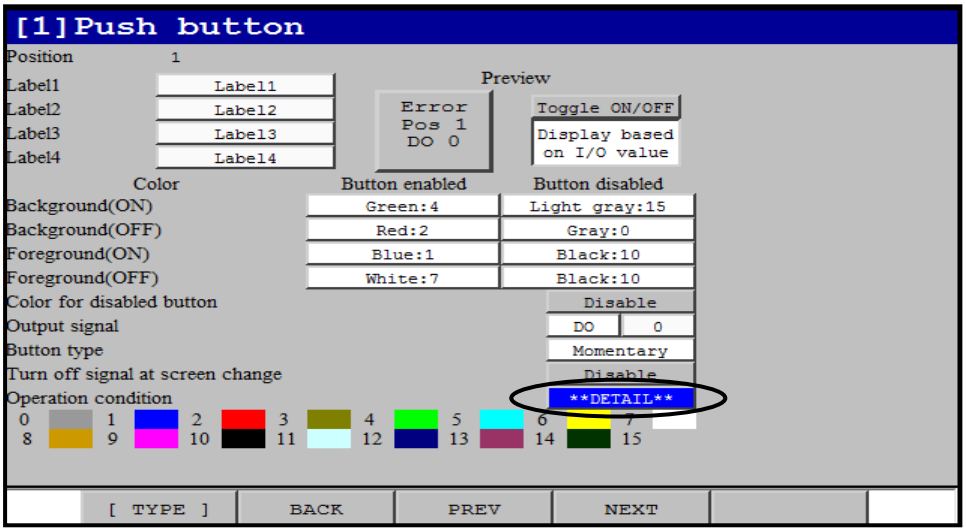
Operation condition is supported by following button types.

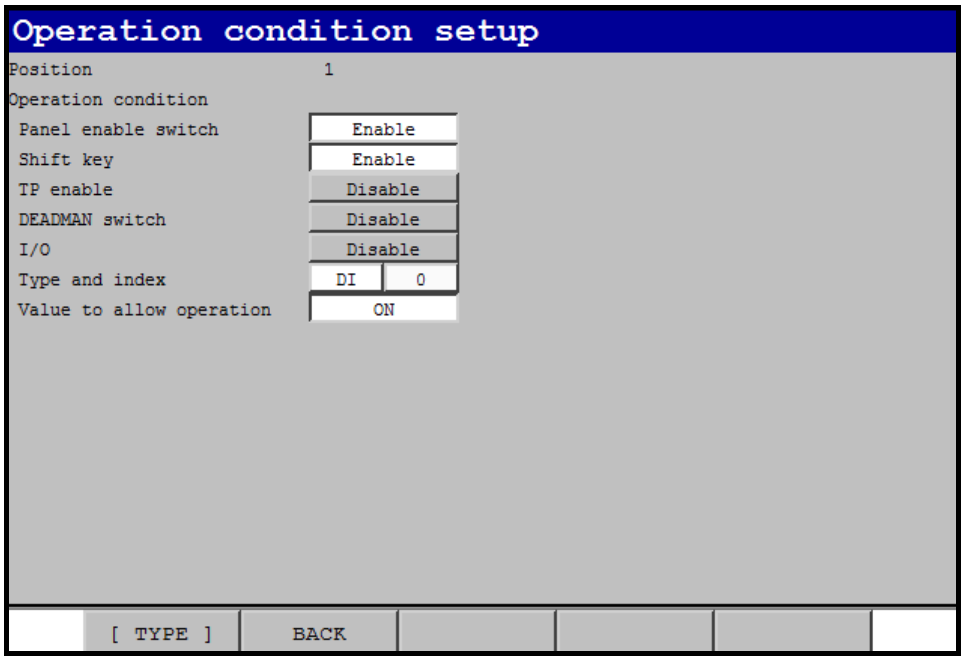
- Push button (type 1)
- Push button lamp (type 2)
- 2 contact point switch (type 3)
- Digital switch (type 6)

Lamp (type 5) and Digital lamp (type 7) don't support operation condition. They are lamp and cannot be pressed.

44.9.2 Operation Condition Setup Screen

The screen is displayed by “Detail” button of button detail setup screen.



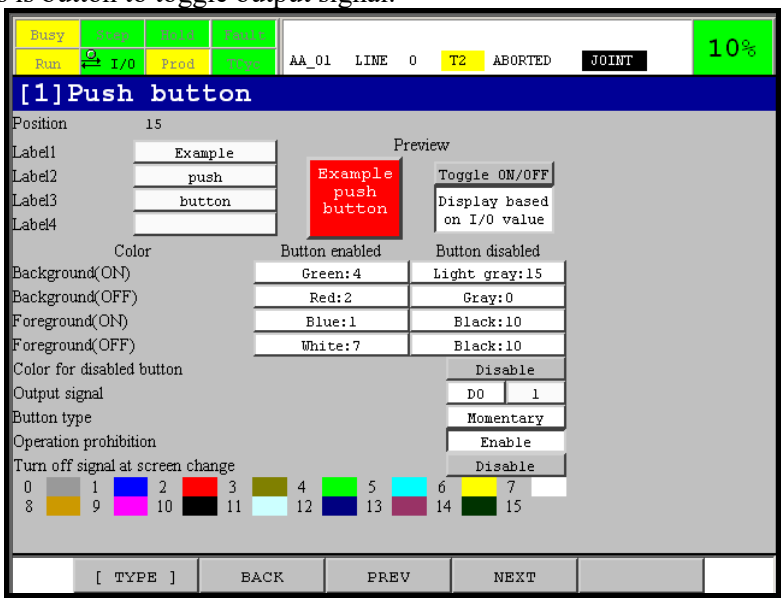


Item	Default value	Description
Position	Display only	Position number of button under setup
Panel enable switch	Enable	Enable Button cannot be pressed if panel enable switch is OFF. If pressed, “IFPN-010 Enable Interface panel” is posted. Disable Status of panel enable switch does not affect whether button can be pressed or not.

Item	Default value	Description
SHIFT key	Enable	Enable The Button cannot be pressed if the [SHIFT] key is not pressed. If the button is pressed, “IFPN-009 Press the [SHIFT] key” is posted. Disable Status of the [SHIFT] key does not affect whether button can be pressed or not.
TP enable	Disable	Enable Button cannot be pressed if TP is disabled. If pressed, “IFPN-019 TP is disabled” is posted. Disable Status of TP enable switch does not affect whether button can be pressed or not.
DEADMAN switch	Disable	Enable Following conditions must be satisfied to press the button. - TP E-Stop button is OFF - TP is enabled - TP DEADMAN switch is gripped properly Otherwise, one of following warning is posted when the button is pressed. - IFPN-018 TP Estop is pressed - IFPN-019 TP is disabled - IFPN-020 Deadman switch is not gripped properly Disable The conditions above aren’t checked.
I/O	Disable	Enable At press of button, value of signal specified by item “Type and index” (Operation condition signal) is checked. If the value isn’t equal to value specified by item “Value to allow operation”, button cannot be pressed. If pressed, “IFPN-022 Condition signal is OFF” or “IFPN-023 Condition signal is ON” is posted. Disable Signal specified by “Type and index” is not checked.
Type and index	DI and 0	This item is used only when item “I/O” is enabled. Signal to check at press (Operation condition signal) is specified.
Value to allow operation	ON	This item is used only when item “I/O” is enabled. This item decides operation condition signal should be ON or OFF to press the button.

44.10 PUSH BUTTON

Type code is 1. This is button to toggle output signal.



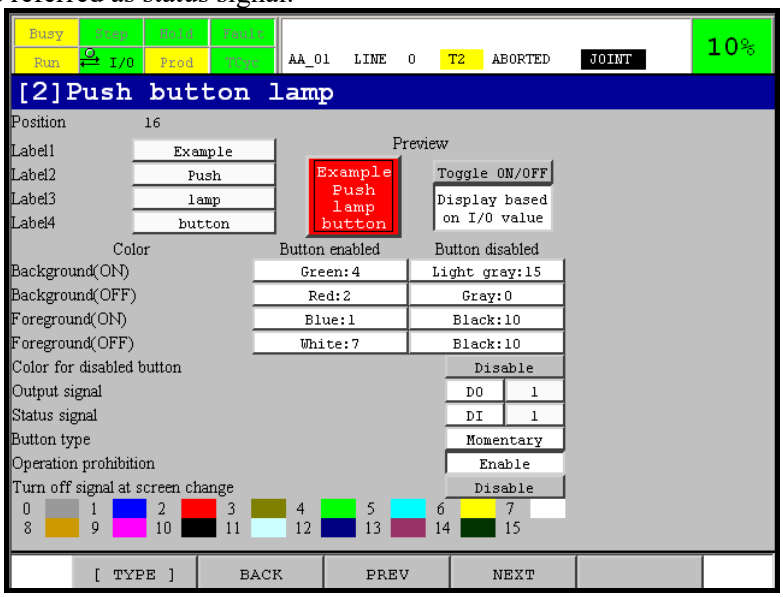
Following table is list of setup items.
Two default values are written in default value column of color items. The 1st one is value for “Button enabled” column of detail setup screen. The 2nd one is for “Button disabled column”.
Please refer to “Foreground color and background color” of section 44.8 "COMMON PROPERTIES” for more information of setup of the colors.

Item	Default value	Description
Label1, …, Label4	Label1,…,Label4	These are strings displayed on the button.
Background (ON)	Green(4) and Light gray(15)	Background color of when output signal is ON.
Background (OFF)	Red(2) and Gray(0)	Background color of when output signal is OFF.
Foreground (ON)	Blue(1) and Black(10)	Foreground color of when output signal is ON. This is color of label.
Foreground (OFF)	White (7) and Black(10)	Foreground color of when output signal is OFF
Color for disabled button	Disable	When this item is enabled, color of “Button disabled” column is used when the button is disabled. Please refer to section 44.8 "COMMON PROPERTIES” for details.
Output signal	DO[0]	Output signal to be toggled by this button.
Button type	Momentary	This item decides how button operates. Choices are “Momentary” or “Alternate”. Momentary When button is pressed, the button turns on. When button is released, the button turns off. Alternate Every time the button is pressed, the button turns on and off.
Turn off signal at screen change	Disable	If this item is enabled, specified output is turned off at a few events. Please refer to 44.8 “COMMON PROPERTIES”
Operation condition	(It isn’t a setting item.)	“Detail” button displays operation condition screen.

NOTE
If button type is momentary (default), specified output signal turns off when;
- The panel is displayed.
- Screen is changed to another one, including another page of panel.
- Any operation that re-displays panel
  (Refresh pane of FCTN menu, Zoom for example)

44.11 PUSH BUTTON LAMP

Type code is 2. This is push button to turn on and off output signal. Color is changed based on another signal. The signal is referred as status signal.



Foreground and background color is decided by status signal (input signal). If button type is alternate, button is pressed state when output signal is on. Button is released state when output signal is off.

Status signal	Released state	Pressed state
OFF	Example Push lamp button	Example Push lamp button
ON	Example Push lamp button	Example Push lamp button

Following table is list of setup items.
Two default values are written in default value column of color items. The 1st one is value for “Button enabled” column of detail setup screen. The 2nd one is for “Button disabled” column.
Please refer to “Foreground color and background color” of section 44.8 "COMMON PROPERTIES” for more information of setup of the colors.

Item	Default value	Description
Label1, …, Label4	Label1,…,Label4	These are strings displayed on the button.
Background (ON)	Green(4) and Light gray(15)	Background color of when status signal is ON. This is similar to Background (ON) of push button lamp but color doesn't depend on output signal. Status signal is referred to decide color.
Background (OFF)	Red(2) and Gray(0)	Background color of when status signal is OFF.
Foreground (ON)	Blue(1) and Black(10)	Foreground color of when status signal is ON. This is color of label.
Foreground (OFF)	White (7) and Black(10)	Foreground color of when status signal is OFF.
Color for disabled button	Disable	When this item is enabled, color of "Button disabled" column is used when the button is disabled. Please refer to section 44.8 "COMMON PROPERTIES" for details.
Output signal	DO[0]	Output signal to be toggled by this button.
Status signal	DI[0]	Color of this button changes according to value of this signal.
Button type	Momentary	This item decides how button operates. Please refer to explanation for same item of push button lamp.
Turn off signal at screen change	Disable	Please refer to explanation for the same item of push button lamp.
Operation condition	(It isn't a setting item.)	"Detail" button displays operation condition screen.

NOTE

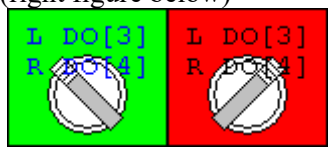
If button type is momentary (default), specified output signal turns off when;

- The panel is displayed.
- Screen is changed to another one, including another page of panel.
- Any operation that re-displays panel
  (Refresh pane of FCTN menu, Zoom for example)

44.12 2 CONTACT POINT SWITCH

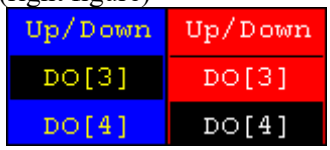
Type code is 3. This is switch to toggle status of 2 output signals.
Suppose DO [3] and DO [4] are specified as signals of 2 contact point switch.
The button can toggle following 2 status

- DO [3] is ON and DO [4] is OFF (left figure below)
- DO [3] is OFF and DO [4] is ON (right figure below)

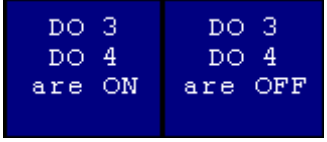


This type of button has another display format, up-down. If you use up-down format, the number of labels are 3. Following figures are example of up-down format with same signals specified.

- DO [3] is ON and DO [4] is OFF (left figure)
- DO [3] is OFF and DO [4] is ON (right figure)

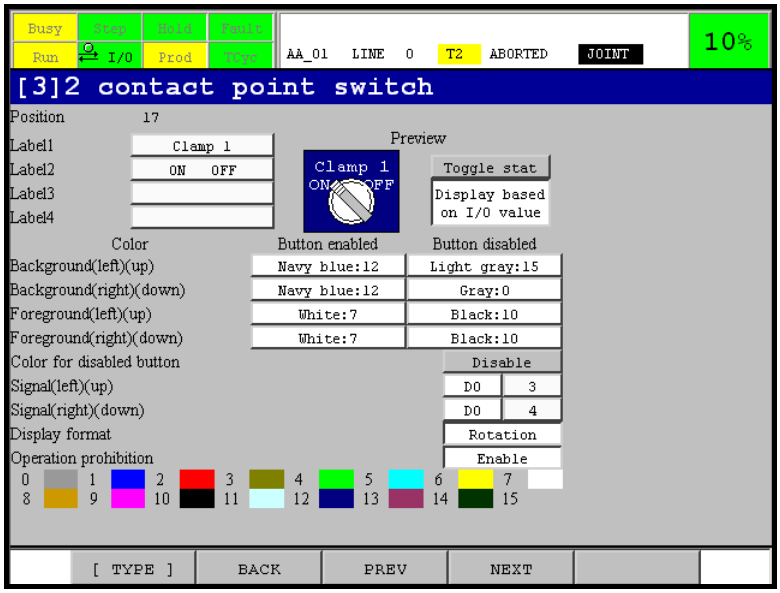


If both signals are ON or OFF, the following button will be displayed.



NOTE
The 2 signals setup for 2 contact point switch should not be ON (or OFF) at the same time.
It is not proper state for the type of button. If this is not expected in your system, please confirm design of your system. I/O configuration and TP program for example. If it is expected status, don't use 2 contact point switch for the signals.

If you press the button in this status, signal (left) (up) is changed. If it is OFF, it is turned on. If it is ON, it is turned off.



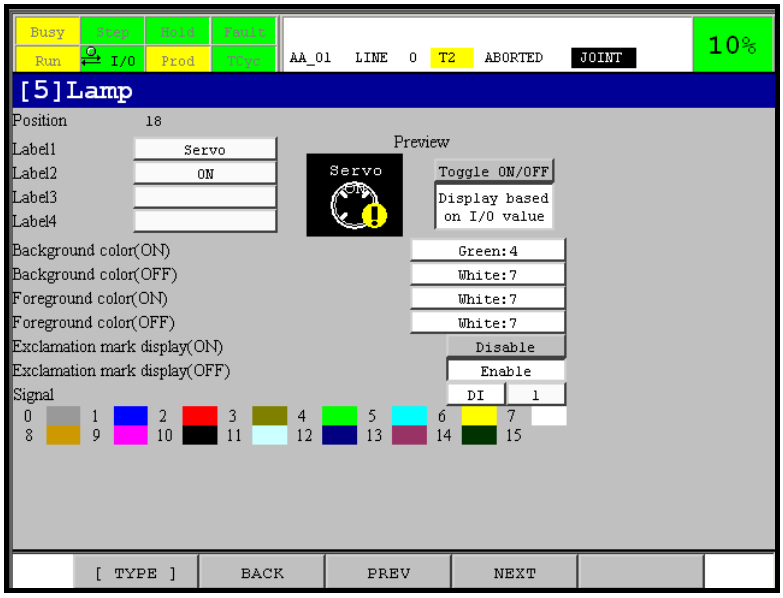
Following table is list of setup items.
Two default values are written in default value column of color items. The 1st one is value for "Button enabled" column of detail setup screen. The 2nd one is for "Button disabled" column.
Please refer to "Foreground color and background color" of 44.8 "COMMON PROPERTIES" for more information of setup of the colors.

Item	Default value	Description
Label1, …, Label4	Label1,…,Label4	These are strings displayed on the button. If format is Up/Down, Label 4 is not used.
Background (left)(up)	Navy blue (12) and Light gray(15)	Background color of when signal (left) (up) is ON.
Background (right)(down)	Navy blue (12) and G ray(0)	Background color of when signal (right) (down) is ON.
Foreground (left)(up)	White(7) and Black(10)	Foreground color of when signal (left) (up) is ON. This is color of label.
Foreground (right)(down)	White(7) and Black(10)	Foreground color of when signal (right) (down) is ON.
Color for disabled button	Disable	When this item is enabled, color of "Button disabled" column is used when the button is disabled. Please refer to section 44.8 "COMMON PROPERTIES" for details.
Signal (left) (up)	DO[0]	Output signal to be turned on and off by this button. Please refer to the explanation at the beginning of this section.
Signal (right) (down)	DO[0]	
Display format	Rotation	The choices are "Rotation" and "Up/Down". Please refer to the explanation at the beginning of this section.
Operation condition	(It isn't a setting item.)	"Detail" button displays operation condition screen.

If index of signal (right) (down) is 0, this button works as switch that just turns on and off signal (left) (up). Even if setup of signal (left) (up) is correct and index of signal (right) (down) is 0, it isn't recognized as error.

44.13 LAMP

Type code is 5. This is lamp to show status of signal.

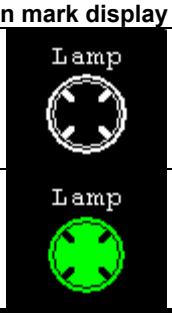
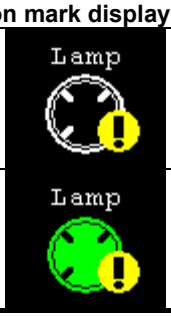


Following table is list of setup items.

Item	Default value	Description
Label1, …, Label4	Label1,…,Label4	These are strings displayed on the lamp.
Background (ON)	Green(4)	Background color of when specified signal is ON. This is color of circular part at center of the lamp.
Background (OFF)	White(7)	Background color of when specified signal is OFF.
Foreground (ON)	White(7)	Foreground color of when specified signal is ON. This is color of label.
Foreground (OFF)	White(7)	Foreground color of when specified signal is OFF.
Exclamation mark display (ON)	Disable	Please refer to the explanation just after this table.
Exclamation mark display (OFF)	Enable	Please refer to the explanation just after this table.
Signal	DI[0]	This lamp shows status of this signal.

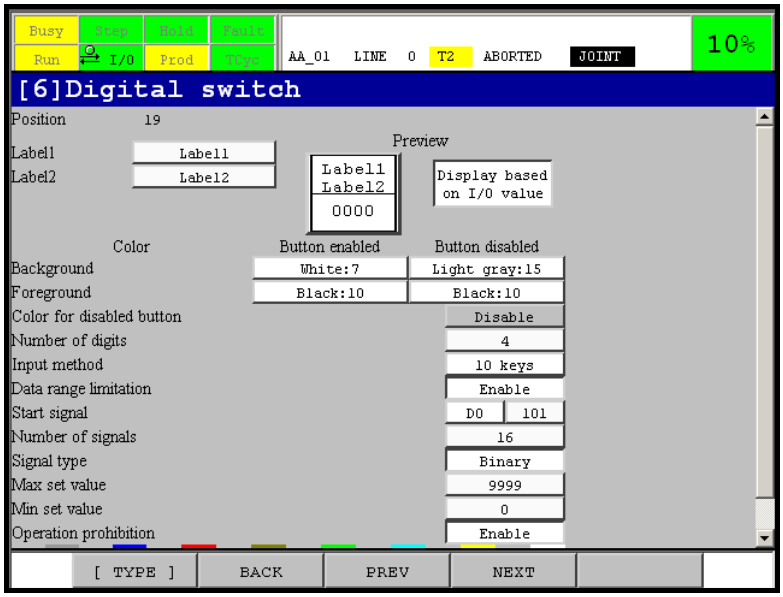
Exclamation mark display

These items specify if exclamation mark is displayed or not when signal is ON or OFF, respectively.

Signal	Exclamation mark display is disabled	Exclamation mark display is enabled
OFF		
ON		

44.14 DIGITAL SWITCH

Type code is 6. This switch regards a series of output signals as one group. This switch output the signals as number of specified format.



Following table is list of setup items.
Two default values are written in default value column of color items. The 1st one is value for “Button enabled” column of detail setup screen. The 2nd one is for “Button disabled” column.
Please refer to “Foreground color and background color” of Section 44.8 "COMMON PROPERTIES” for more information of setup of the colors.

Item	Default value	Description
Label1, Label2	Label1, Label2	These are strings displayed on the button.
Background	White(7) and Light gray(15)	Background color of this button.
Foreground	Both Black(10)	Foreground color of this button.
Color for disabled button	Disable	When this item is enabled, color of “Button disabled” column is used when the button is disabled. Please refer to section 44.8 "COMMON PROPERTIES” for details.
Number of digits	4	Value is displayed in specified number of digits.
Input method	10 keys	Choices are “10 keys” and “Switch”. 10 keys When you press the button, numeric keyboard (dialog box) is displayed. Input number to set to a series of signals. Switch Press of the button increments value by one. If value is maximum value, value is not incremented.”IFPN-011 Out of range(%d - %d)” is posted.
Data range limitation	Enable	If this is enabled, value is limited between maximum and minimum value, which are setup in this screen. Even if this is disabled, you cannot set value that cannot be represented by specified number of signals.
Start signal	DO [0]	This button treats a series of signals as a group to compose integer. This is the first signal of the group.

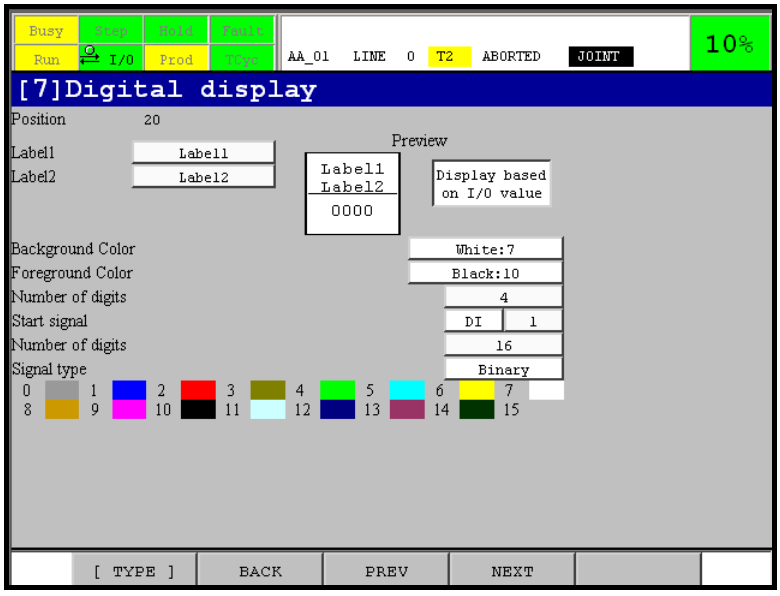
Item	Default value	Description
Number of signals	16	This is how many signals are included to group of signals. The number is counted from the signal specified by another setup item “Start signal”. If start signal is DO [1] and this item is 4, This button changes from DO [1] to DO [4].
Signal type	Binary	Encoding of signals as integer number. BCD The group of signals is considered as BCD (binary coded decimal). Binary Each signal is considered as bit of binary number. This is similar to GO [ ].
Max set value	9999	This is the maximum value that this button can set when data range limitation is enabled.
Min set value	0	This is the minimum value that this button can set when data range limitation is enabled.
Operation condition		“Detail” button displays operation condition screen.

When “Display based on I/O value” button is OFF, value in preview button as follows.

- If index of start signal is 0, *** is displayed.
- Otherwise, 0 is displayed.

44.15 DIGITAL DISPLAY

Type code is 7. This lamp regards a series of output signals as numeric value and displays the value.



Following table is list of setup items.

Item	Default value	Description
Label1, Label2	Label1, Label2	These are strings displayed on the lamp.
Background	White(7)	Background color of this lamp.
Foreground	Black (10)	Foreground color of this lamp.
Number of digits	4	Value is displayed in specified number of digits.
Start signal	DI [0]	This lamp treats a series of signals as a group. This is the first signal of the group.

Item	Default value	Description
Number of signals	16	This is how many signals are included to group of signals. If start signal is DI [1] and this item is 4, This button changes from DI [1] to DI [4].
Signal type	Binary	This is encoding of signals. BCD The group of signals is considered as BCD (binary coded decimal). Binary Each signal is considered as bit of binary number. This is similar to GI [ ]. Ver. The number of “Binary” is divided by 100 and is displayed as real number to the second decimal place. The number of digits must be 3 or 4.

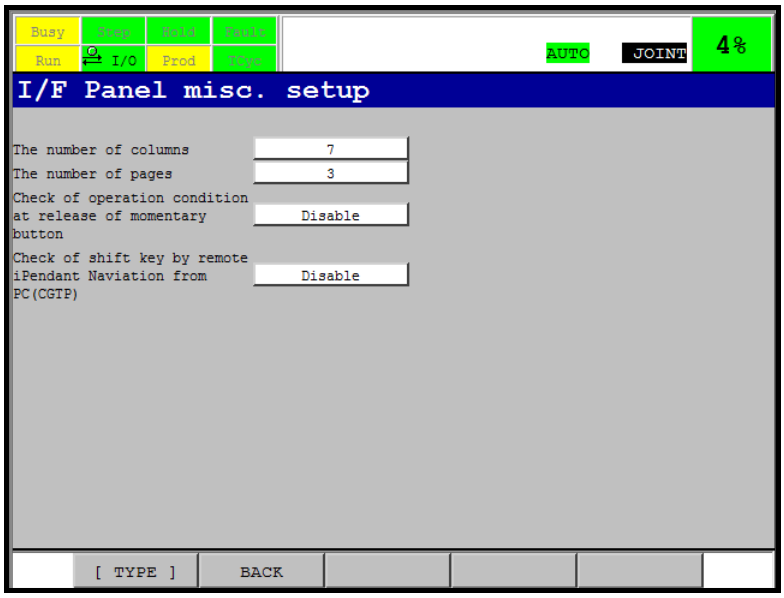
When “Display based on I/O value” button is OFF, value in preview button as follows.

- If index of start signal is 0, *** is displayed.
- Otherwise, 0 is displayed.

44.16 MISCELLANEOUS SETTING SCREEN

Miscellaneous setting screen is a screen to setup items that affects all panels. The screen is displayed by following procedure.

1 Display interface panel setup screen. Please refer to Section 44.3.
2 Press the “Miscellaneous setting” button. The following screen will be displayed.



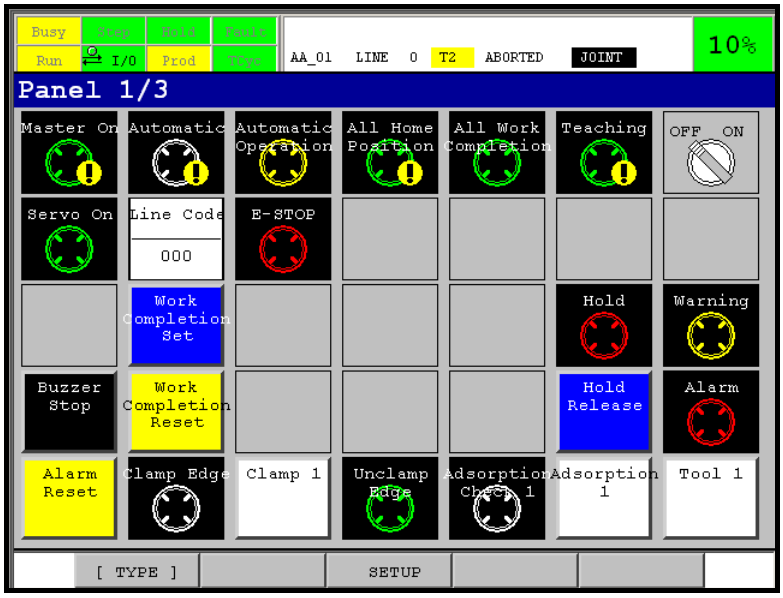
Item	Default value	Description
The number of columns	7	Press of the button displays submenu to select the number of columns. Choices are 7 or 8.
The number of pages	3	Minimum value is 1. Maximum value is 10. You can display from page 1 to the page specified here. For example, if this item is set to 3, you can display from page 1 to 3.

Item	Default value	Description
Check of operation condition at release of momentary button	Disable	Disable At release of momentary button, operation condition is not checked. For example, panel enable switch and [SHIFT] key are not checked. Enable At release of momentary button, operation condition is checked. If button is released when operation condition isn't satisfied, output signal does not change. Button becomes released status.
Check of [SHIFT] key by remote iPendant Navigation from PC (CGTP)	Disable	Disable Suppose Internet Explorer of personal computer displays Interface panel using "Navigate iPendant (CGTP)" of robot homepage. On CGTP, operation condition does not check the [SHIFT] key. You can press buttons without pressing the [SHIFT] key. You can also press panel enable switch without pressing [SHIFT] key. If "Color for disabled button" is enabled, change of color also ignores status of the [SHIFT] key. Enable CGTP checks [SHIFT] key. You cannot press button and panel enable switch without pressing the [SHIFT] key. However, you cannot press the [SHIFT] key on CGTP. You can hardly operate Interface panel. For *i*Pendant operation of robot homepage (CGTP), refer to "ADVANCED *i*Pendant FUNCTIONS" chapter on Ethernet Function OPERATOR'S MANUAL (B-82974EN).

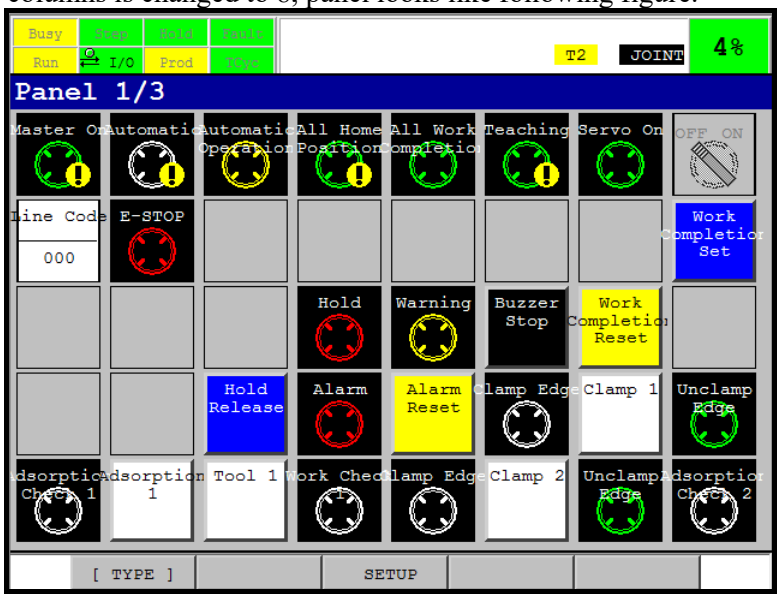
NOTE
1 8 column mode can display more buttons than 7 column mode can. However, width of button is narrower.
2 Change of the number of columns causes the change of layout of buttons. Please refer to explanation in this section.
3 Don't change the number of columns and pages when the other panes are displaying Interface panel or its setup screens.

The number of columns

The number of columns of interface panel is 7 by default. You can change it to 8. Following figure is a panel of 7columns



After the number of columns is changed to 8, panel looks like following figure.

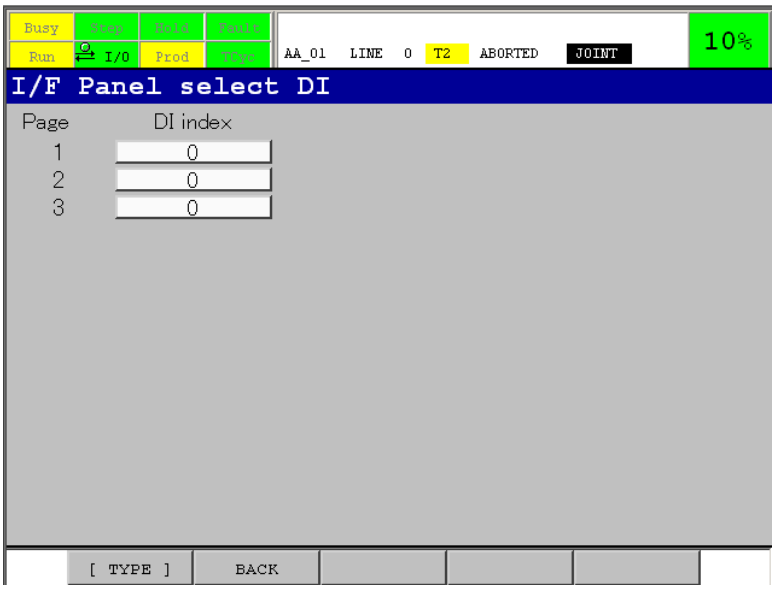


Setup of each buttons is common to both display. Both panels display buttons sequentially from position number 1. The number of buttons displayed on each panel is different.
For example, the 1st page of 7 column panel displays from position number 1 to 34. That of 8 column displays from position number 1 to 39.
When the number of columns is 8, the last 5 buttons of the 1st page are buttons that were the first 5 buttons on the 2nd page of 7 column panel.

44.17 EXTERNAL I/F PANEL SELECTION SETUP SCREEN

You can display Interface panel at rise of digital input signal (DI). The signal is specified in external Interface panel selection setup screen. The screen is displayed by the following procedure.

1 Display interface panel setup screen. Please refer to Section 44.3.
2 Press the “External I/F panel selection” button. The following screen will be displayed.



- The number of signals alters based on the number of pages.
- At the rise of specified DI, corresponding page of Interface panel is displayed.
  The panel is displayed in the 1st pane of iPendant.
- Signal is specified for each page
- Interface panel may not be displayed when this function conflicts with another function to force display of a screen.

NOTE
Cycle power is needed for change to take effect.

44.18 BACKUP AND RESTORE

Files for backup and restore are classified into two groups.

- IFP0**.XML (** is 2 characters that are language suffix).
  For example, it is “eg” for English and “kn” for Japanese.
- IFP_VARS.SV

IFP0**.XML

- This file contains setup of panel.
  Position number, type and parameters for buttons are saved into these files.
- Contents of panels are saved into file for each language.
- IFP**.XML can be restored at cold start.

NOTE
1 Don’t load the XML file when panel or setup screens are displayed.
2 Setup of type setup screen and button detail setup screens are saved at exit from the screens. Backup after exiting from the screens.

IFP_VARS.SV contains following setup.
- Signal number of external Interface panel selection
- Items of miscellaneous setting screen.
- IFP_VARS.SV should be read in controlled start mode.

44.19 LIMITATIONS

1 *i*Pendant with touch panel is required.
2 Buttons and lamp of this function only support digital type signal.
Digital type signal means signals that have ON/OFF value, DI/DO and RI/RO for example.
This function does not support I/O type that has integer value, GI/GO and AI/AO for example.
The other data of robot such as register and system variable cannot be handled by buttons of this function.
3 You cannot use “<” and “>” in string of label.
4 After pressing a button, release it after confirming change of color and pressed/released status of button.
5 After pressing a button, release touch panel on the same button.
If you drag your finger and release touch panel on another button or empty place, button doesn’t work properly.
Especially, momentary button doesn’t turn off specified signal.
Press and release the button again, properly.
6 Display panel and all setup screens in single pane mode.
Otherwise, whole screen cannot be displayed. Cursor may be displayed outside of screen.
Changed setup items that is not saved may be lost by display or operation on another pane.
You cannot change focus by touching panel and type setup screen. Use DISP key.
7 Setup of type setup screen and button detail setup screens are saved at exit from screen.
Exit from the screens before display panel, power off or backup.
8 Zoom function does not zoom text and image of button.

45 SPECIAL JOG SEQUENCE

This function enables Jog operation without pressing 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 following system variable.

$JOG_SP_MODE:
0: This function is invalid.
1-100: 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%.

⚠WARNING
If large value is set in this system variable, the robot is possible to move with high speed when jog key is pressed. Don't set the value over 5 for this system variable.

45.3 APPLICATION

Automatic Override setup when changing frames:
Normally, override is automatically initialized to 10% when Jog frame is changed in Teach Pendant. With this function, as you can confirm the direction in low speed, you may want to make this automatic override setup invalid.
Setting the following system variable, the current override can be kept when frame is changed.

$SCR.$COORDOVRD = 100 (Standard value 10)

Meaning of this system variable:

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

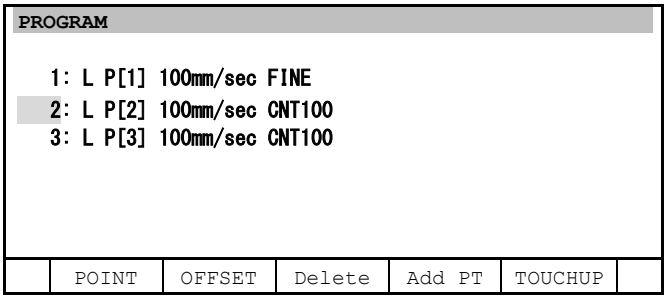
This function includes two functions, “Motion instruction insert and instruction delete function” and “Data offset function”. These are described below.
It is possible to order Enhanced Edit option (J560) independently.
In that case, this function is available only in handling tool and LR handling tool.

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

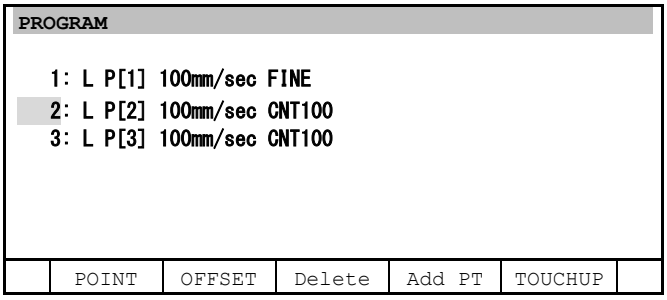
Function keys are displayed as follows in EDIT screen.



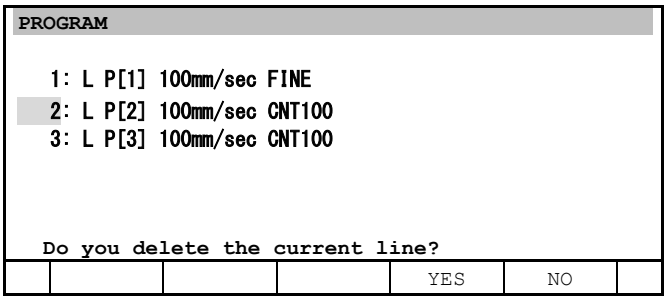
Procedure 46-1 Delete instructions

Step

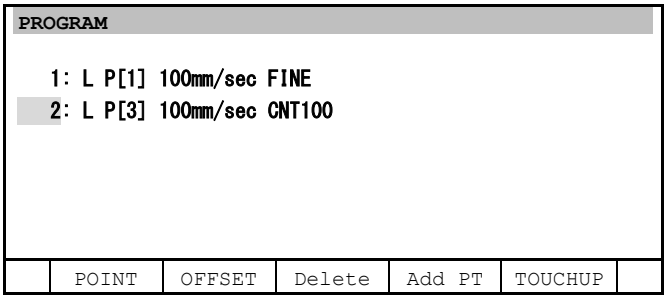
1 Move the cursor on the line you want to delete.
2 Enable Teach Pendant, and press the [SHIFT] key and the F3, Delete.



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 .



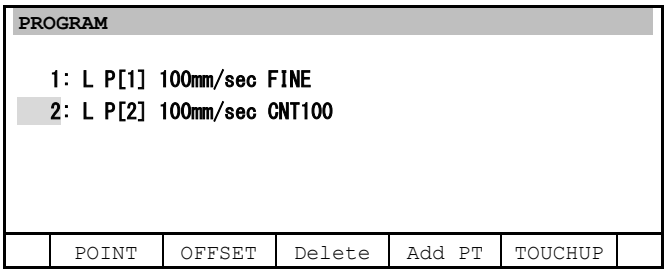
4 The line that the cursor was on is deleted.



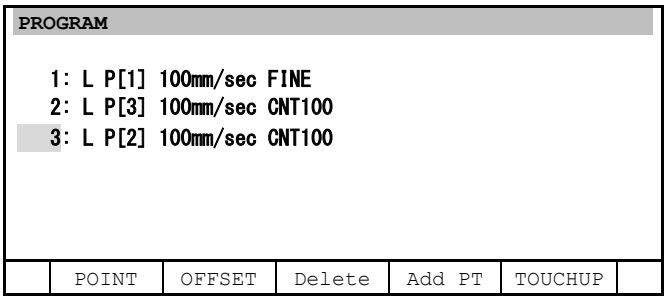
Procedure 46-2 Insert motion instructions

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)”.

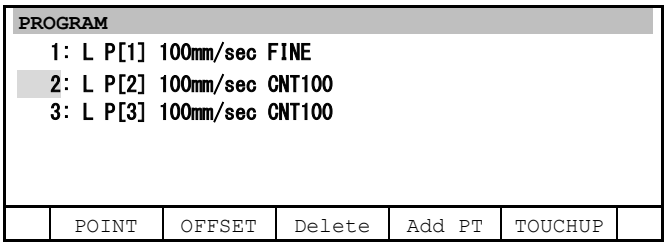
46.2.1 Feed Rate Conversion

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.

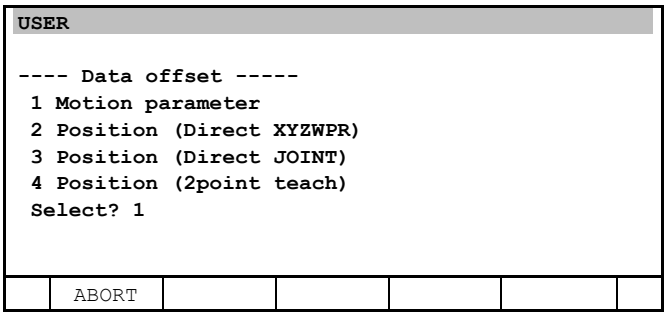
Procedure 46-3 Feed rate conversion

Step

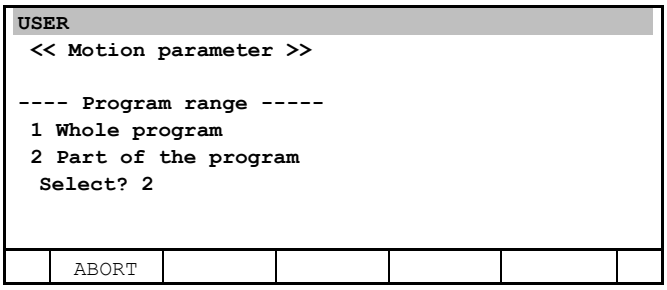
1 Press the F2, OFFSET in EDIT screen.



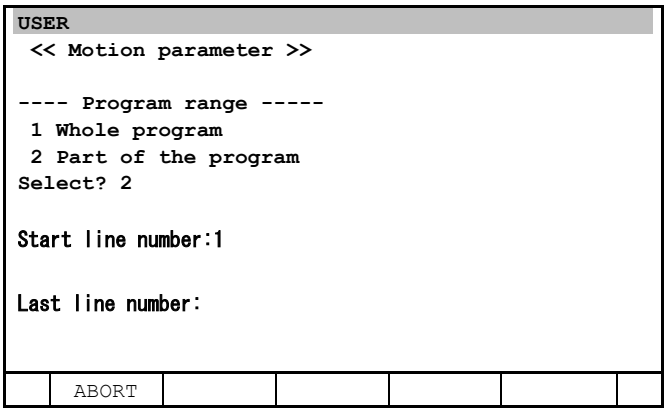
2 Input “1” to select “1 Motion parameter”.



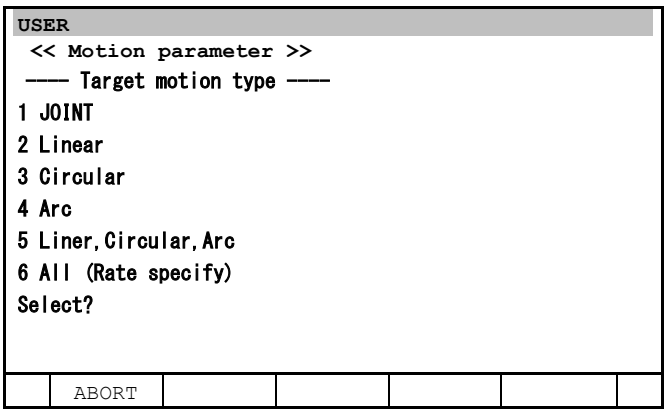
3 To convert a whole program, input “1”. Then go to step 5. To convert a part of a program, input “2”. Then go to step 4.



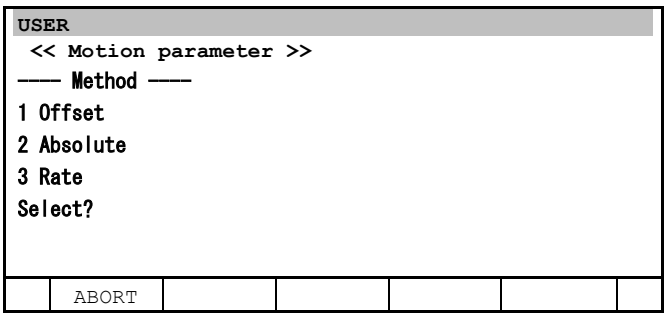
4 Input start line number and last line number.



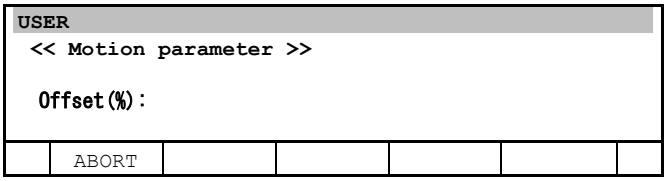
5 Select target motion type or All (Rate specify). If you select All (Rate specify), go to step 11.



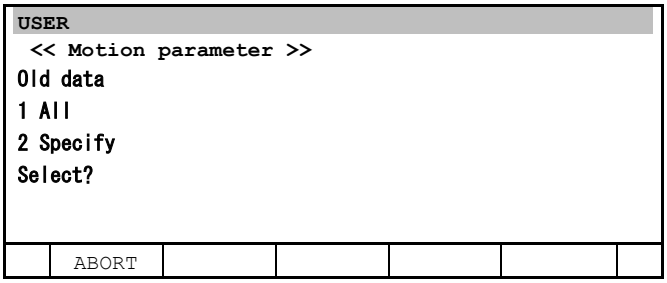
6 Select the method for converting feed rate.
Input “1” to specify the offset to the current feed rate. Then go to step 7.
Input “2” to convert feed rate to specified value. Then go to step 8.
Input “3” to specify the rate to the current feed rate. Then go to step 11.



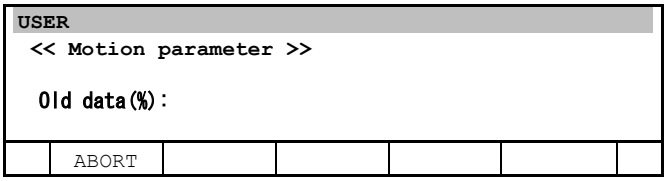
7 Input the offset. When you want to decrease the value, input negative value. The unit is % when Joint motion or mm/sec when Linear motion or Circular motion. Go to step 12.



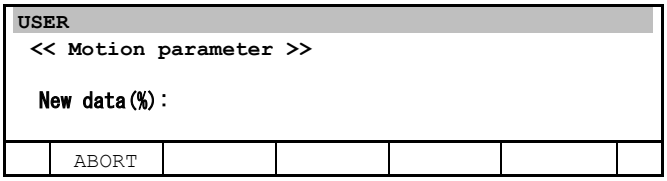
8 Input “1” to convert all the motion instructions. Then go to step 10.Input “2” to convert only the motion instructions whose current feed rate is equal to the value specified by operation 9. Then go to step 9.



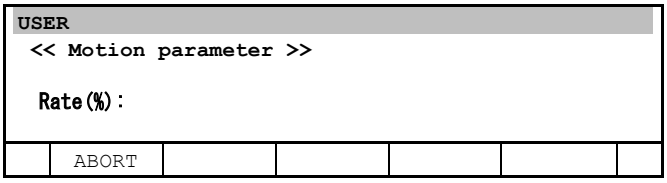
9 Input current feed rate of the motion instructions that you convert. The unit is % when Joint motion or mm/sec when Linear or Circular motion.



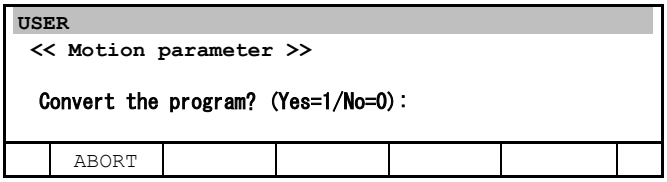
10 Input new value of feed rate. The unit is % when Joint motion or mm/sec when Linear or Circular motion. Go to step 12.



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)

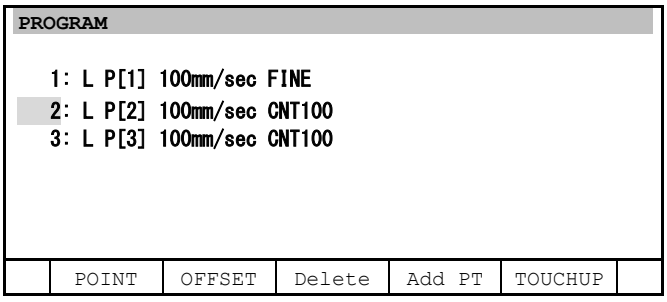
Position data conversion (2 point teach method) converts multiple position data in the program at once. Translation is specified by teaching 2 point, the original position and the destination position.

- 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.

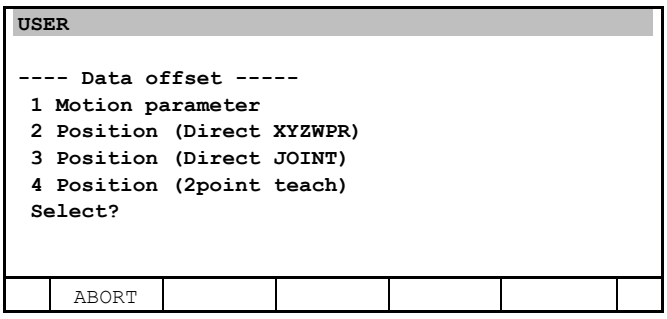
Procedure 46-4 Position data conversion (direct method)

Step

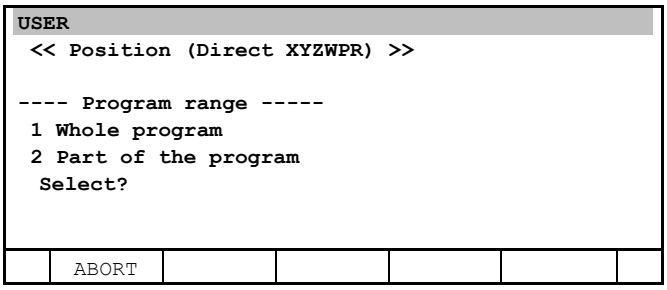
1 Press the F2, OFFSET in EDIT screen.



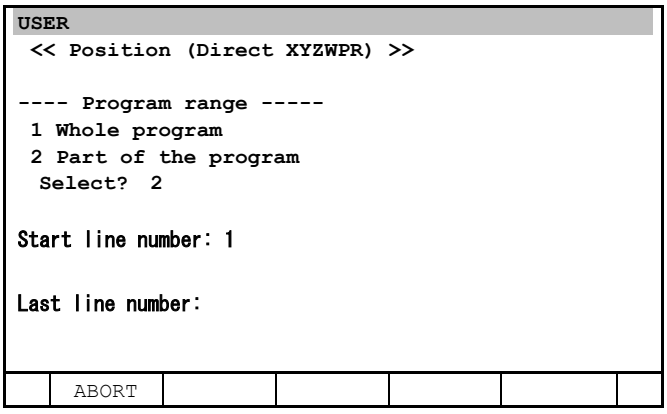
2 Input “1” to select “2 Position (Direct XYZWPR). Input “2” to select “3 Position (Direct JOINT).



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.

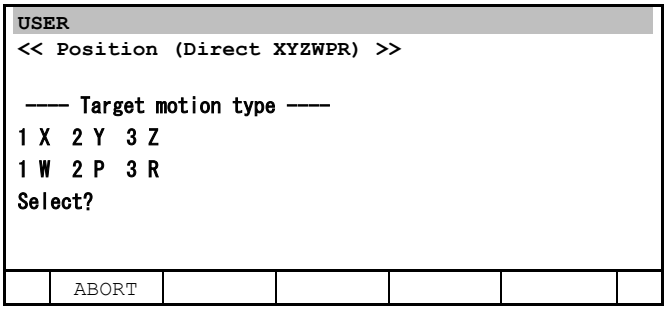


4 Input start line number and last line number.

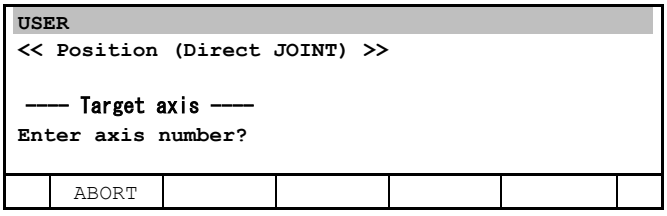


5 Input the number of the element or axis number to convert.

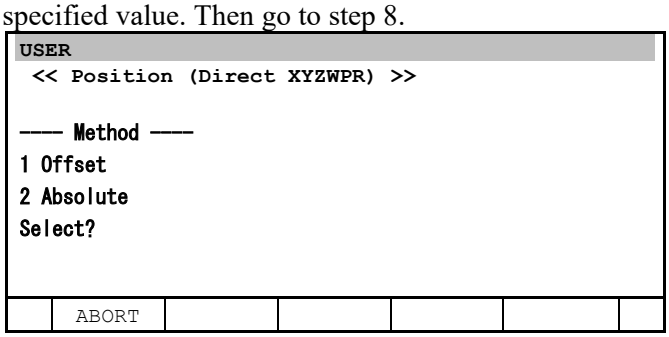
When “Direct XYZWPR” is selected



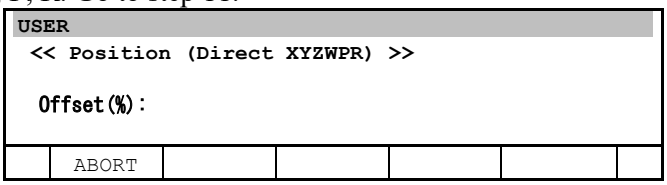
When “Direct JOINT” is selected



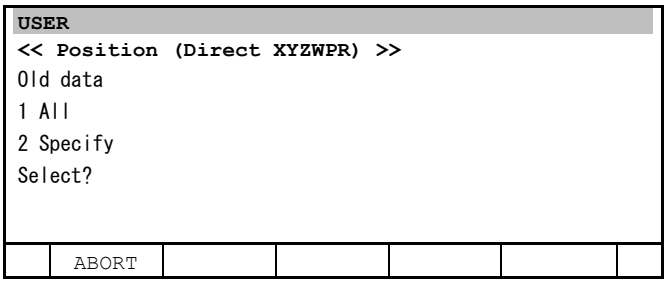
6 Input “1” to specify the offset to current position data. Then go to step 7. Input “2” to specify the position data into specified value. Then go to step 8.



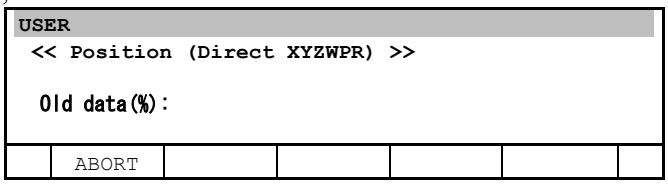
7 Input the offset. When you want to decrease the value, input negative value. The unit is mm when X, Y, Z or deg when W, P, R. Go to step 11.



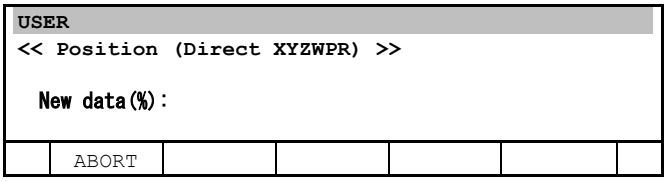
8 Input “1” to convert all the motion instructions. Then go to step 10. Input “2” to convert only the motion instructions whose current feed rate is equal to the value specified by operation 9. Then go to step 9.



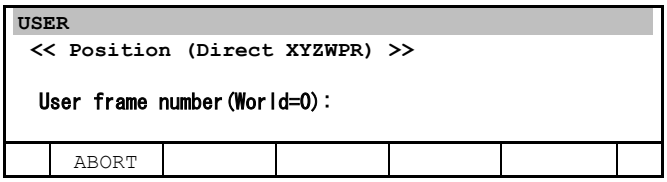
9 Input the current position data of the motion instruction that you convert. The unit is mm when X, Y, Z or deg when W, P, R.



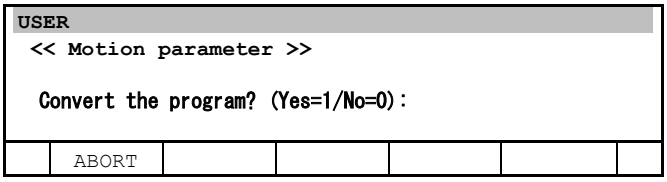
10 Input new position data. The unit is mm when X, Y, Z or deg when W, P, R.



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)

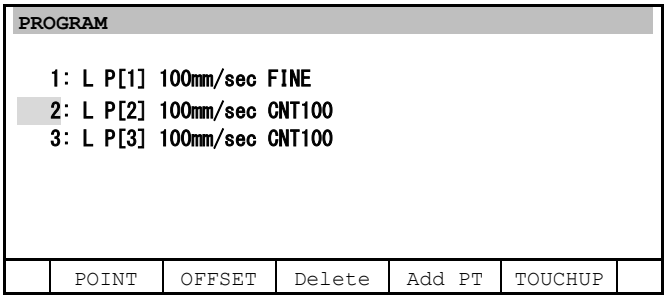
Position data conversion (2 point teach method) converts multiple position data in the program at once. Translation is specified by teaching 2 point, the original position and the destination position.

- 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.

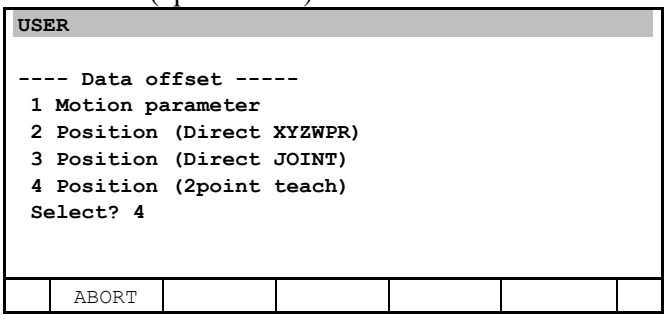
Procedure 46-5 Position data conversion (2 point teach method)

Step

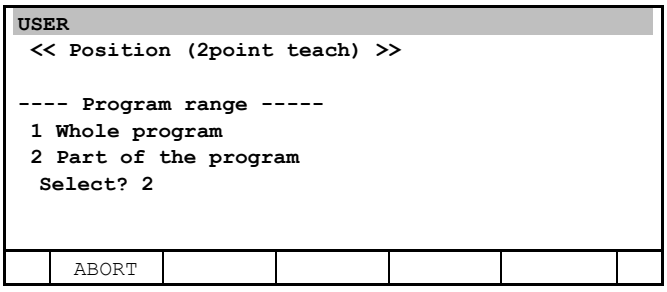
1 Press the F2, OFFSET in EDIT screen.



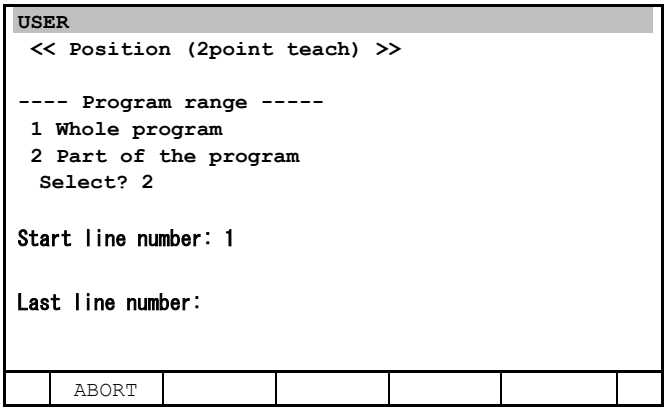
2 Input “4” to select “4 Position (2point teach)”.



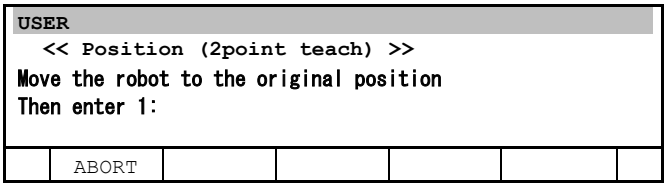
3 Input “1” to convert a whole program. Then go to step 5. Input “2” to convert a part of a program. Then go to step 4.



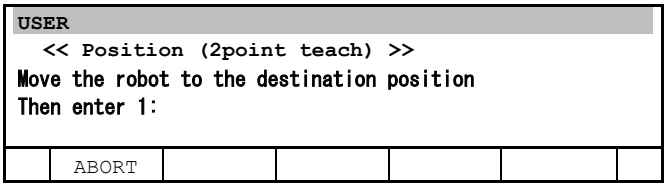
4 Input the start line number and last line number.



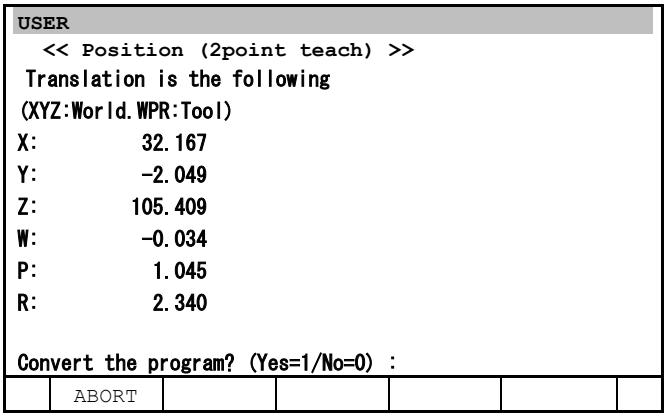
5 Input “1” after moving the robot to the original position.



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 JOINT POSITION OUTPUT FUNCTION

Joint Position Output Function outputs current commanded position information of a specified axis with group output signal (GO).

48.1 HOW TO USE

48.1.1 Setting

If you would like to use this function, please set following system variables on SYSTEM Variables screen.

- $SV_GUN_CTRL[n].$OUTPUT_ENB
  This is the flag to enable or disable this function. When this variable is TRUE, this function is enabled.

- $SV_GUN_CTRL[n].$GROUP_NUM
  This is used to specify the group number.

- $SV_GUN_CTRL[n].$AXIS_NUM
  This is used to specify the axis number.

- $SV_GUN_CTRL[n].$GO_VALUE
  This is used to specify the index of GO.

- $SV_GUN_CTRL[n].$IO_SCALE
  This is used to change precision of the value which stored in GO. You need to set this system variable to a positive integer. The default value of this system variable is “10”. The specified axis value is multiplied by this system variable and decimals are omitted from the result. The absolute value of the result is stored in GO.

“n” is integer from 1 to 5. It is possible to specify up to 5 axes freely with the above system variables.

48.1.2 Note

- Up to 5 axes can be specified.
- You need to assign 16 digital signals for the specified GO in advance.
- If a specified axis is linear axis, the unit is [mm]. If a specified axis is rotary axis, the unit is [deg].
- Default value of $SV_GUN_CTRL[n].$IO_SCALE is 10, so default precision is 0.1[mm] or 0.1[deg].
- The specified axis value is multiplied by this system variable and decimals are omitted from the result. The absolute value of the result is stored in GO. When the specified axis value is negative, it is defined by [32768 + (the absolute value of the result)].
- The specified axis value is multiplied by this system variable and decimals are omitted from the result. When the absolute value of the result exceeds 32767, GO is not updated.

48.1.3 Example of a Setting

In the case that store the value of axis 1 of group 2 in GO[5], the system variable is set as followings.

```
$SV_GUN_CTRL[1].$OUTPUT_ENB =    TRUE
$SV_GUN_CTRL[1].$GROUP_NUM =     2
$SV_GUN_CTRL[1].$AXIS_NUM =      1
$SV_GUN_CTRL[1].$GO_VALUE =      5
$SV_GUN_CTRL[1].$IO_SCALE =      10
```

If position of the axis is 125.64[mm], the value of GO[5] should be “1256”. If position of the axis is -80.56[mm], the value of GO[5] will be “33573 (= 32768 + 805)”.

49 EXPANDED REGISTERS FUNCTION

This option function expands the maximum number of numeric register to 5000 and position register to 2,000. However, the maximum number of position register depends on the number of group. Please refer to the Table 49 (a) for the maximum numbers of numeric and position register in each group system.

Table 49 (a) The maximum numbers of registers in each group system

Groups	Position registers	Numeric registers
1	2000	5,000
2	1000	
3	666	
4	500	
5	400	
6	333	
7	285	
8	250	

When this option was loaded, the numbers of numeric and position register were expanded as table 49.1 automatically. And expanded numbers of registers are available in all registers instructions. Please refer to section 4.5 REGISTER INSTRUCTIONS of FANUC Robot series OPERATOR'S MANUAL (Basic Operation) (B-83284EN) for details of numeric and position registers function.

NOTE
This option function requires 3MB SRAM memory.

49.1 SETTING THE NUMBER OF REGISTERS

The maximum numbers of numeric and position register can be changed. However, the numbers can't be set beyond the numbers of table 49.
The numbers of these registers can be change by following procedures.

Procedure 49-1 Setup the maximum numbers of numeric and position register

Step

1 Turn ON the controller with the [PREV] and the [NEXT] key pressed. Then select [3 Controlled start].
2 Press the [MENU] key.
3 Select "0 – NEXT --" and select "1 PROGRAM SETUP", so Program Limits screen will be displayed.

Program Limits screen

```
Program Limits
      Program Limits Setup                1/14

       1   User Tasks                    9
       2   Numeric Registers             5000
       3   Position Registers            2000
       4   String Registers              25
       5   Macros                        150
       6   User Alarm                    10
       7   Trace Length                  200
       8   Num. Dig. Ports               512
       9   Error Severity Table          20
      10   Program Adjust Schedule       10

    [ TYPE ]
```

4 Put the cursor on numeric registers or position registers and input new value.
5 Perform a cold start.

> **⚠WARNING**
> If the number of numeric or position register is decreased, reduced registers data are lost. For example the number of numeric registers decrease from 5,000 to 4,000, numeric registers data from 4,001 to 5,000 are lost.

49.2 SAVEING AND LOADING FILES

By executing all backup in FILE screen, numeric registers and position registers data are saved as files. When all backup is executed, numeric registers data are saved as a file named of "NUMREG.VR" and position registers data are saved named of "POSREG.VR". By loading these files, numeric and position registers data are overwritten to the saved data. Please refer to section 8.4 SAVING FILES and 8.5 LOADING FILES of FANUC Robot series OPERATOR'S MANUAL (Basic Operation) (B-83284EN) for details.

However, there are some limitations to load these files. These limitations are described in following subsections.

49.2.1 LOADING NOT EXPANDED xxxREG.VR

NUMREG.VR and POSREG.VE files saved from a system not loading Expanded Registers (R830) are enabled to load to a system loading Expanded Registers (R830).

However, in this case, the numbers of numeric and position registers are translated to the numbers of loaded files.

> **⚠CAUTION**
> When you load these files, if the warning message "VARS-206 ***REG.VR will load at startup" is output, you have to reboot the controller to apply the registers data.

> **NOTE**
> At Program Limits screen in controlled start, the number of registers can be changed again. Please refer to Subsection 49.1 SETTING THE NUMBER OF REGISTERS for changing.

49.2.2 LOADING EXPANDED xxxREG.VR

Loading to the system loaded Expanded Registers

NUMREG.VR and POSREG.VE files saved from a system loading Expanded Registers (R830) are enabled to load to a system loading Expanded Registers (R830).

However, the numbers of numeric and position registers are translated to the numbers of loaded files if the number of registers is changed.

> **⚠ CAUTION**
> When you load these files, if the warning message "VARS-206 ***REG.VR will load at startup" is output, you have to reboot the controller to apply the registers data.

NOTE
At Program Limits screen in controlled start, the number of registers can be changed again. Please refer to Subsection 49.1 SETTING THE NUMBER OF REGISTERS for changing.

Loading to the system NOT loaded Expanded Registers

NUMREG.VR and POSREG.VE files saved from a system loading Expanded Registers (R830) are disabled to load to a system not loading Expanded Registers (R830).

However, in the system loading Expanded Registers, if the numbers of registers are set below the values of the table 49.2.2, the saved file is enabled to load.

Table 49.2.2 The maximum numbers of registers in each group system

Groups	Position registers	Numeric registers
1	372	999
2	372	
3	248	
4	186	
5	148	
6	124	
7	106	
8	93	

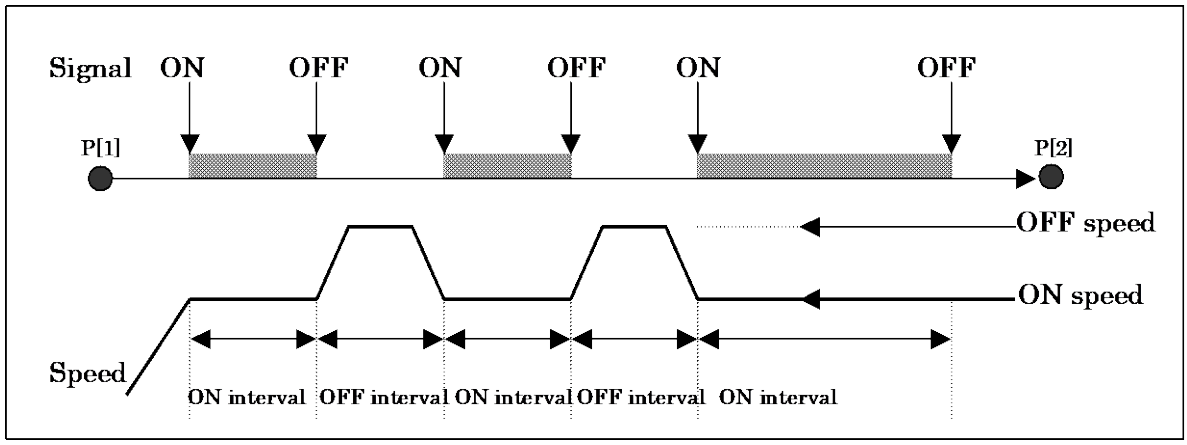
The numbers of numeric and position registers are translated to the number of loaded file.

⚠ CAUTION
When you load these files, if the warning message “VARS-206 ***REG.VR will load at startup” is output, you have to reboot the controller to apply the registers data.

NOTE
At Program Limits screen in controlled start, the number of registers can be changed again. Please refer to Subsection 49.1 SETTING THE NUMBER OF REGISTERS for changing.

50 STITCH FUNCTION

For stitch-like sealing and laser welding, stitch function enables easy teaching of such sealing/welding part (as ON interval and OFF interval) and turning signal on/off in each interval.



50.1 SPECIFICATION

50.1.1 Instruction

Following two instructions specify an interval in which stitch function is used.
This interval is referred as stitch interval hereafter.

- **STITCH[i]**
  - Start stitch process (change of speed, signal control) according to stitch condition.
  - Stitch process doesn't stop until "STITCH END" instruction is executed.
  - Stitch condition is specified in data screen. Please refer to 50.1.2）
  - This instruction is taught as usual instruction only. This cannot be taught as motion option.
- **STITCH END**
  - By this instruction, stitch process is stopped. Signal is turned off.

Example

```
1: J P[1] 100% FINE
2:    STITCH[1]
3: L P[2] 2000mm/sec CNT100
4: L P[3] 2000mm/sec FINE
5:    STITCH END
```

Stitch interval.
Only linear and circular motions are available. If joint motion type is used, error is posted

50.1.2 Stitch Condition

- Stitch condition is set in data screen. Screens for stitch condition are Stitch Data List screen, Stitch Data Detail screen and Stitch Data Common screen.（Please refer to 50.4 APPENDIX 1 for structure of screens.）
- Items on each screen are as follows.

Stitch Data List screen

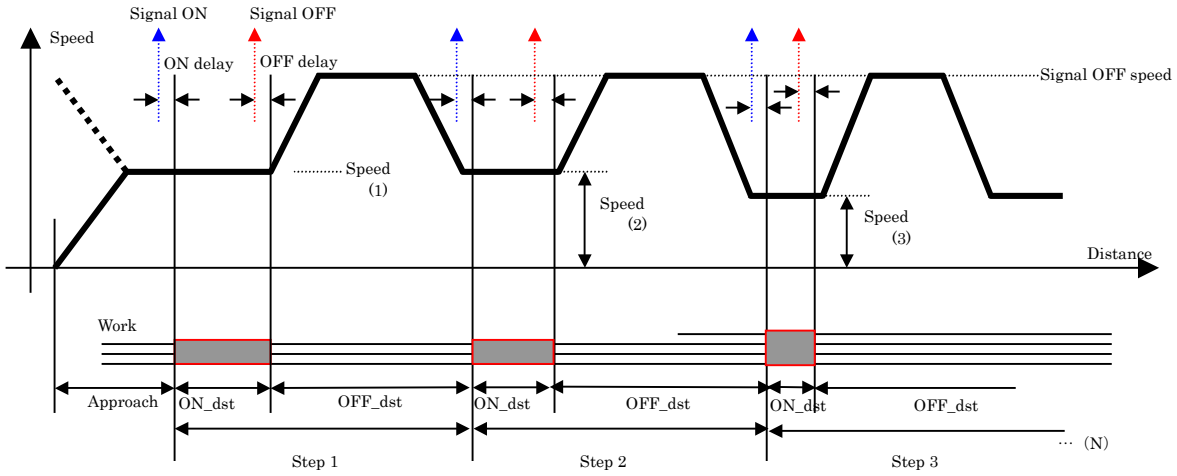
This screen displays list of stitch conditions. Stitch instruction specifies condition by number (1-20) displayed in this screen. To display detail screen of each condition, set cursor to a condition and press the [F4] key.

Comment	Comment for each condition is displayed.	
F3（COMMON）	To display Stitch Data Common screen.	
F4（DETAIL）	To display detail screen for selected condition (condition which cursor is set to).	
F5（COPY）	User can copy selected condition to another condition. **Caution) Even if condition that selected condition is copied to is already set, the former one is overwritten.**	

Stitch Data Detail screen

◆ In this screen, following items can be set.

Item	Explanation	Range
Comment	Comment for each condition is displayed	
Equipment condition	Type of equipment signal (DO/RO/GO/AO) and index	DO/RO/GO/AO 1-32766
Signal ON delay time	Delay time between signal ON and actual action of equipment can be set here.	0-9999[msec]
Signal OFF delay time	Delay time between signal OFF and actual stop of equipment can be set here.	0-9999[msec]
Approach distance	Approach distance to stabilize speed.	0-9999[mm]
Signal OFF speed.	Speed in OFF intervals. “Speed unit” in Stitch Data Common screen decides unit.	0.1 - $PARAM_GROUP [*].$SPEEDLIM Usually, this corresponds to 2000mm/sec
Following item decide condition for each step.		
ON_dst	Distance of signal ON interval	0-9999[mm]
Speed	Speed of signal ON interval. Unit is decided in Stitch Data Common screen.	0.1 - $PARAM_GROUP[*].$SPEEDLIM Usually, this corresponds to 2000mm/sec
OFF_dst	Distance of signal OFF interval.	0-9999[mm]
Count	This specifies how many times this step is executed. If you want to repeat a sequence, input time of repetition. If 0 is specified, this step is neglected. Next step with non-zero counter is executed.	0-99[times]
Cond.	Output value of equipment signal is specified here. This value is output in ON interval.0 is output in OFF interval. If equipment signal is DO or RO, set 1. In case of GO, value specified here is output as binary code.	1-9999
F2（DATA）	You can display another Stitch Data Detail screen for different condition.	
F3（LIST）	Go back to Stitch Data List screen.	



NOTE

- Do not change ON speed and OFF speed during executing the program. Change them after pause. If it is changed during the program execution. There is the case that a robot does not work at designated speed.
- If stitch motion is performed by the controller installed TCP speed estimation function, warning “TCPP-018 Begin Error Mode at line” or “TCPP-019 Speed Ovrd Mode at line” could occur. No remedy is required for these warning.

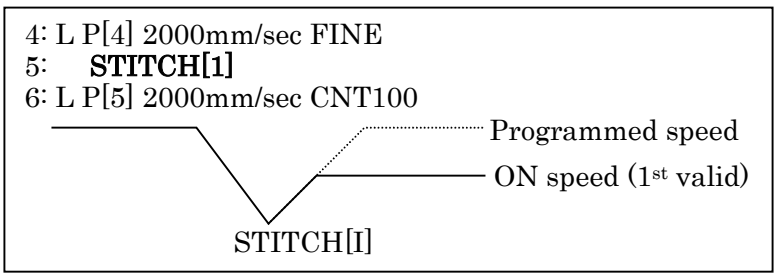
Stitch Data Common screen

◆ In this screen, following items can be set.

Stitch enable	Only when specified signal is ON, stitch process (speed change and signal control) is performed. If this signal is OFF, speed of motion in stitch interval is programmed speed. If index is 0, it corresponds to signal ON.	DI/RI 1-32766
Equipment enable	Only when specified signal is ON, equipment signal is turned on. If this is OFF, only speed control is done. It index is 0, it corresponds to signal ON.	DI/RI 1-32766
Speed unit	Unit of Signal OFF speed and ON speed is specified here.	cm/min mm/sec inch/min

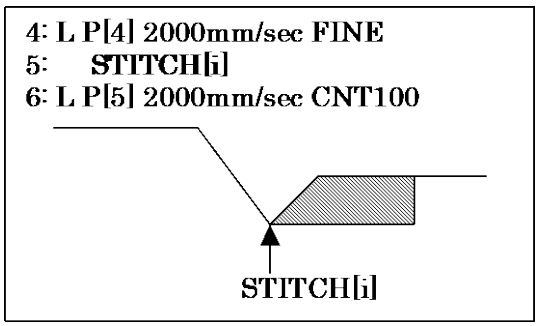
50.1.3 Flow of Stitch Process

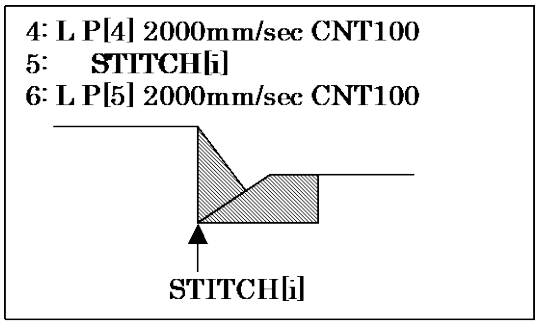
1 After execution of STITCH[i], speed of TCP is changed to ON speed of first valid step of stitch condition i. Valid step is a step with non-zero count.
Speed specified by program has no meaning.



2 When distance which TCP traveled after STITCH[i] exceeds {approach distance – ON delay (converted to distance) }, R-30iA output signal.

NOTE
If STITCH[i] instruction is executed before complete of motion, like STITCH[I] just after CNT motion, approach distance is shorter than expected.





NOTE
ON delay time is interpreted after conversion to distance. On delay time is converted to a distance that TCP travels at On speed and 100% override. Timing of signal output is based on this converted on delay time. Please adjust ON delay time with override 100%.

3 After this, change of speed and ON/OFF of signal is done according to accumulated distance TCP traveled.
4 If motion statement doesn't complete after completion of all steps, TCP moves at OFF speed until STITCH END instruction is executed.

50.1.4 Other Specifications

◆ Several motions in a stitch interval.
Stitch function doesn't care how many motion statements are in a stitch interval. See following example. TCP moves from P[1] to P[3] according to stitch condition

```
1: J P[1] 100% FINE
2:    STITCH[1]
3: L P[2] 2000mm/sec CNT100
4: L P[3] 2000mm/sec FINE
5:    STITCH END
```

◆ Changing stitch condition in a stitch interval
When STITCH[i] instruction is executed in a stitch interval, New stitch condition is used from the instant. Approach distance for new stitch condition is also executed.

◆ Halt and resume
When program is halted, equipment signal is turned off.
After resume, the rest of stitch sequence is processed. But timing of stitch may be different from non-halted case due to effect of acceleration and deceleration

◆ Single step execution
◆ Backward execution
◆ Abort of program
◆ Change of line
In these state or if one of these is done when stitch is in process, stitch function is automatically disabled. Equipment signal is turned off.
If stitch function is disabled, programmed speed is used.

After resume, stitch process isn't performed. To use stitch function, change execution line to one that is before STITCH[i] instruction and run halted (aborted) program.

◆ T1 Mode
When one of the followings is executed in T1 mode, INTP-427 WARN (program, line) Stich disable(T1 mode) occurs.
execute STICH[]
Hold during stich, then change to T1 mode and resume

NOTE
When holded during stich, then change to T1 mode and resume, resume speed is the same speed as stich is enabled.

◆ Saving stitch condition
Stitch conditions can be saved by "All of above" in file screen. File name is as follows.
SYSSTCH.SV

50.2 ADJUSTMENT

Change of speed needs time for acceleration and deceleration. For this reason, it is desired to start deceleration at appropriate timing in order to maintain specified speed in ON interval. Speed of robot depends on specified speed. Appropriate acceleration/deceleration time depends on pose of robot even if specified speed is same.

We recommend users to adjust timing of deceleration for improved performance.

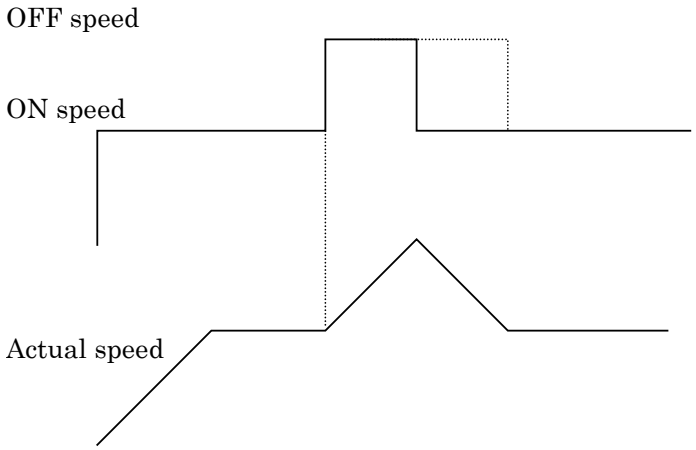
- Timing of deceleration from OFF interval to ON interval
It is possible to decelerate earlier than usual by hastening deceleration to ON speed.
For this adjustment, following system variables are available.

$STCH[i]. $STEP[s].$DEC_NUM=1-10（default value: 2）
OFF interval is divided into pieces of this number in order to change deceleration rate by divisions.
Following system variables specify deceleration rates for each division.
$STCH[i]. $STEP[s].$DEC_RATE[1-10]（[1]=100, [2]=0, the others =100）
For each division, speed is set to following speed.
(OFF speed –On speed)*$DEC_RATE/100+On speed
This means that DEC_RATE[*] = 100 corresponds to OFF speed and 0 corresponds to ON speed.

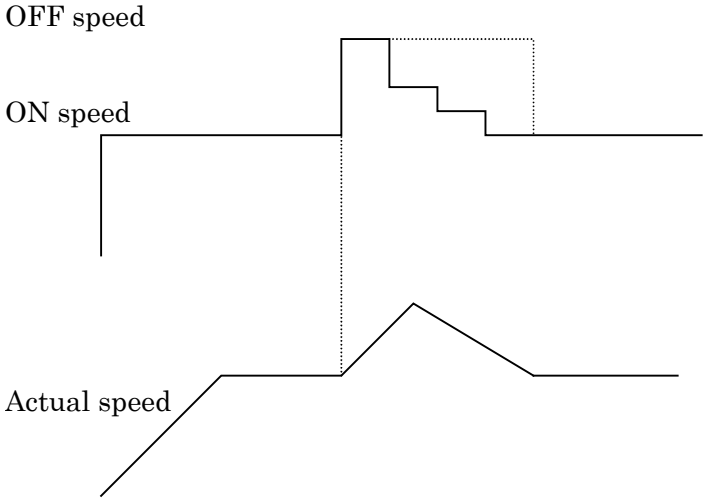
NOTE
DEC_RATE value should be in the range from 0 to 100.

Meaning of $DEC_NUM, $DEC_RATE[]

DEC_NUM= 2
DEC_RATE[1]=100, DEC_RATE=[2]=0



DEC_NUM= 4
DEC_RATE[1]=100, DEC_RATE=[2]=50, DEC_RATE=[3]=25, DEC_RATE=[4]=0

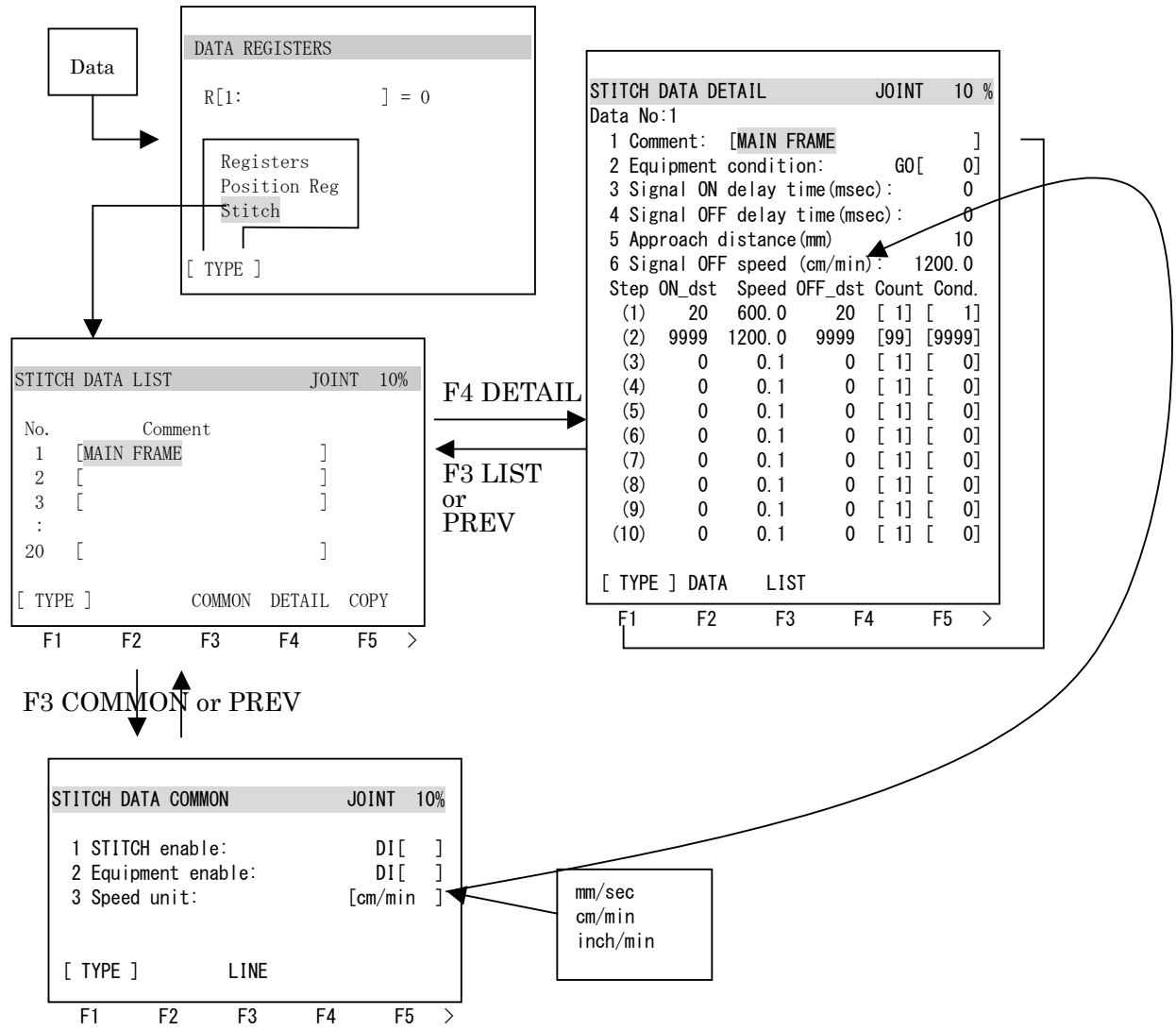


50.3 LIMITATIONS

- After E-stop, distance cannot be calculated correctly. After resume distance that TCP moved is calculated. But there may be errors due to overrun after E-stop.
- JOG doesn't accumulate distance of stitch intervals.
- In stitch interval, following operations are not available.
  JOINT motion
  Change of tool frame and user frame.
- Stitch function cannot be used with following functions.
  - ✓ Continuous turn
  - ✓ Line tracking
  - ✓ Coordinated motion
  - ✓ Remote TCP
  - ✓ Robots which doesn't have Cartesian coordination (Independent axis)
  - ✓ Extended axis
  - ✓ Distance Before (In stitch interval, Distance Before is not available.)
  - ✓ Constant path function

50.4 APPENDIX

Appendix1 Structure of screens for stitch function



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 (a) for another example.

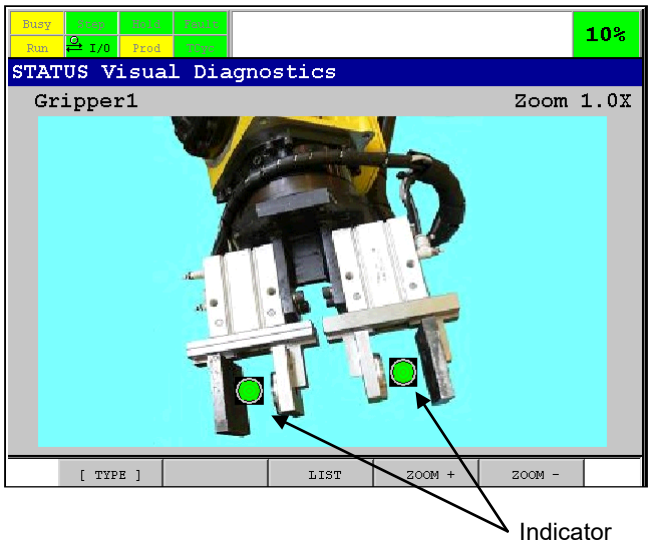


Fig.51 (a) Simple Gripper Example

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.

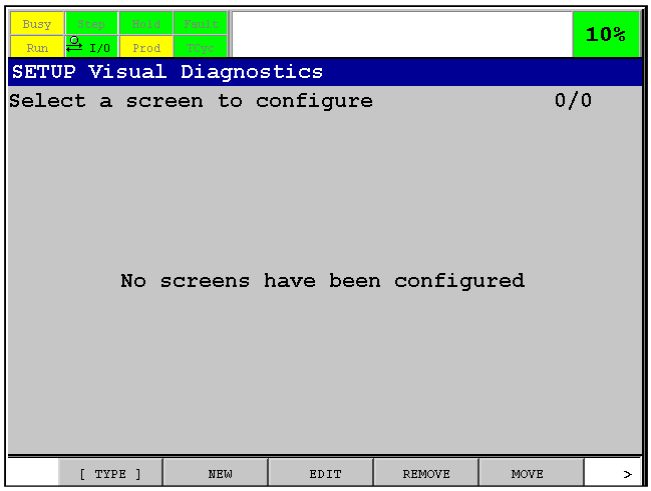


Fig. 51.1 (a) Select a Screen to Configure

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.

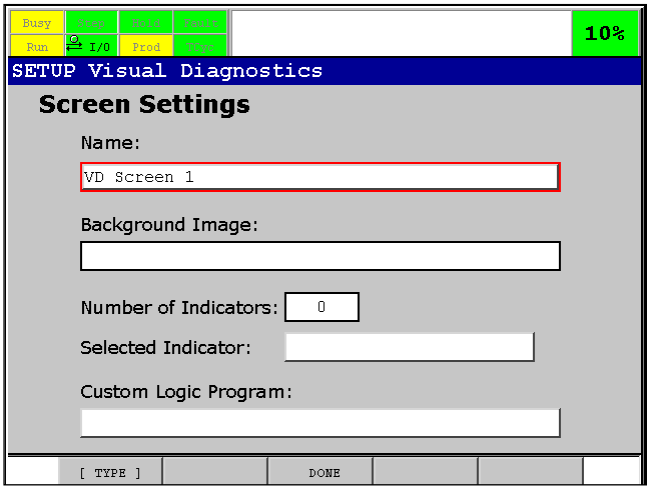


Fig. 51.1 (b) Screen Settings Page

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.

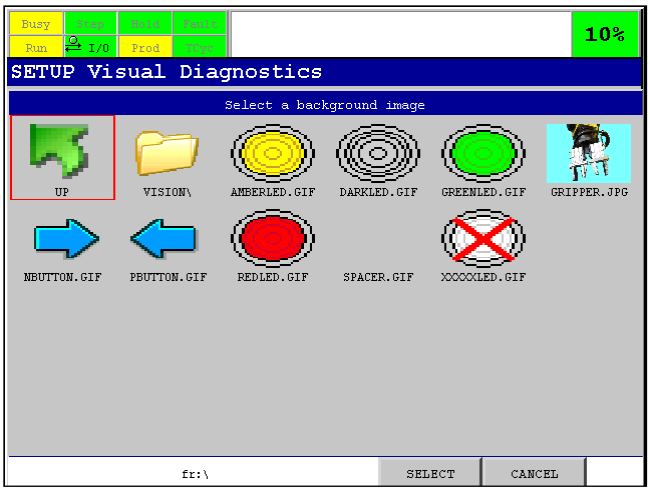


Fig. 51.1 (c) Selecting an Image File

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.

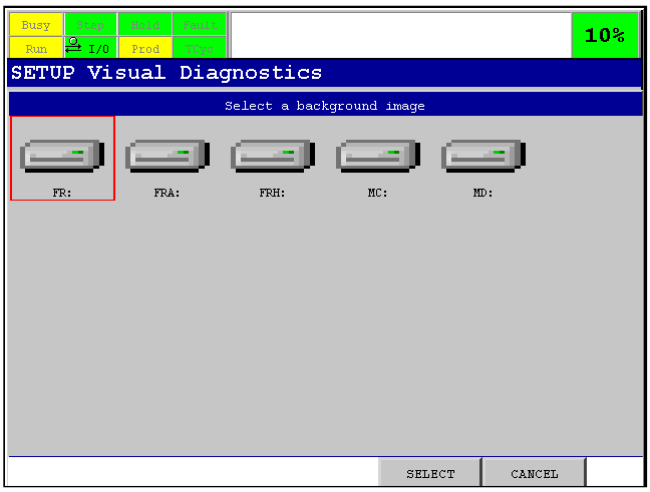


Fig. 51.1 (d) File Devices

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).

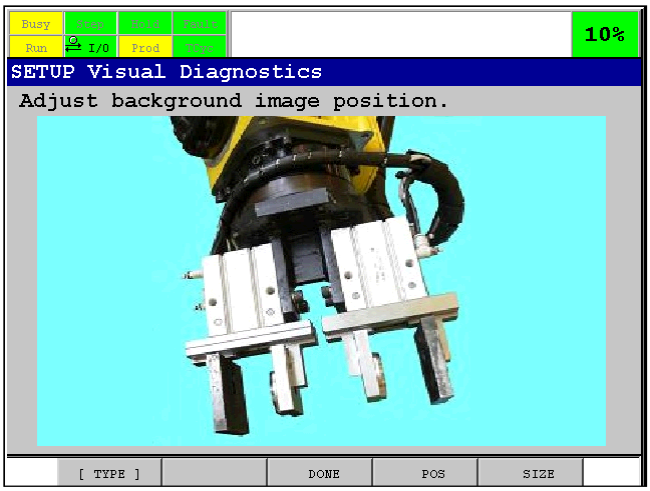


Fig. 51.1 (e) Adjust

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.

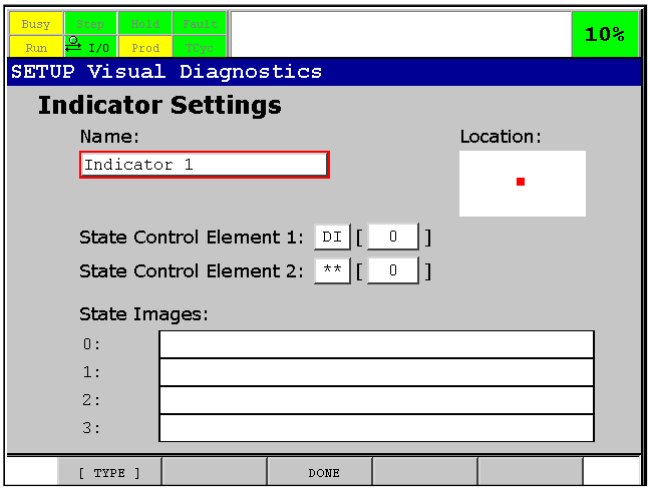


Fig. 51.1 (f) Indicator Settings Page

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.

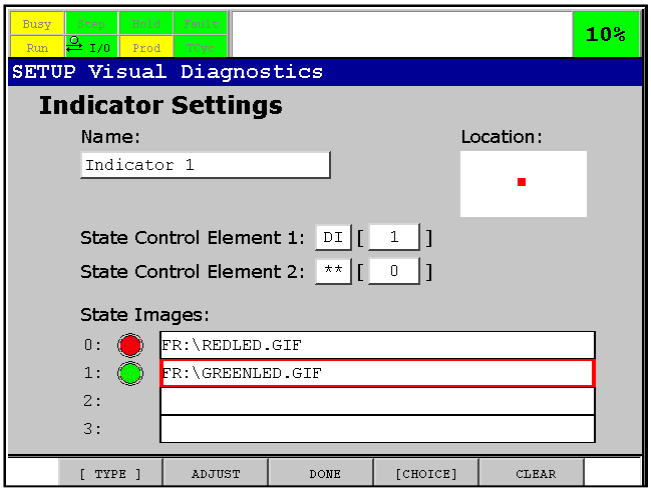


Fig. 51.1 (g) Indicator Settings

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).

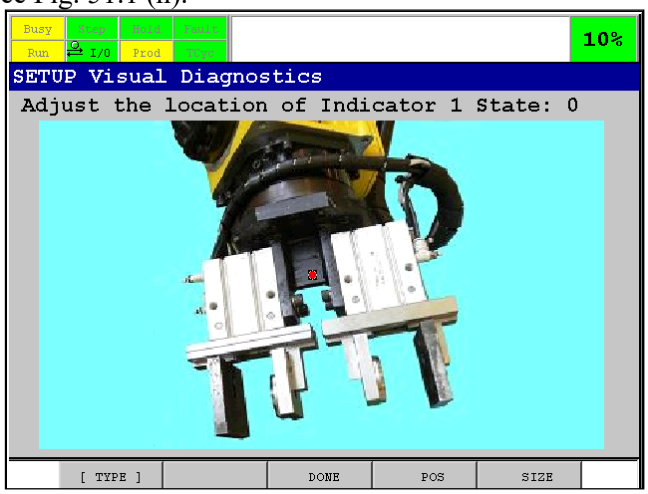


Fig. 51.1 (h) Adjust the Location of LED Indicator State: 0

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.

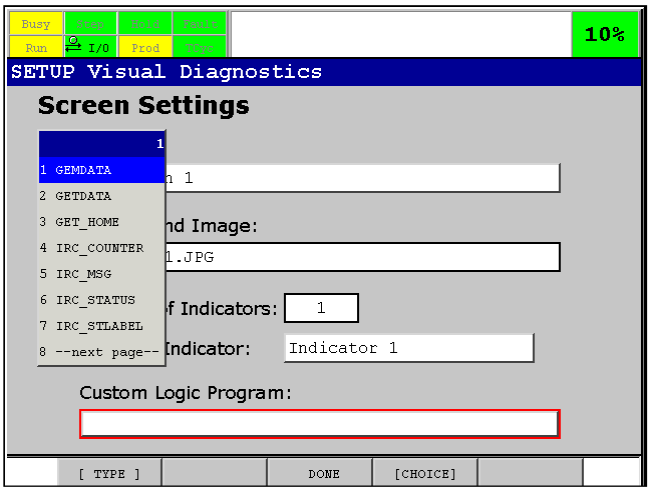


Fig. 51.1 (i) Custom Logic Program Field

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.

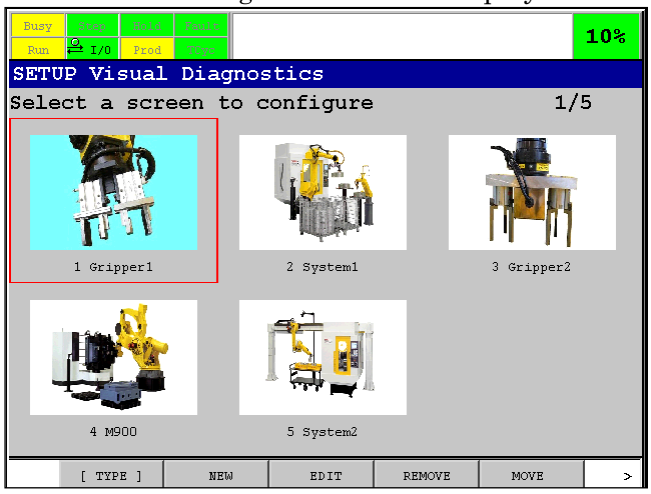


Fig.51.2(a) SETUP Visual Diagnostics

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.

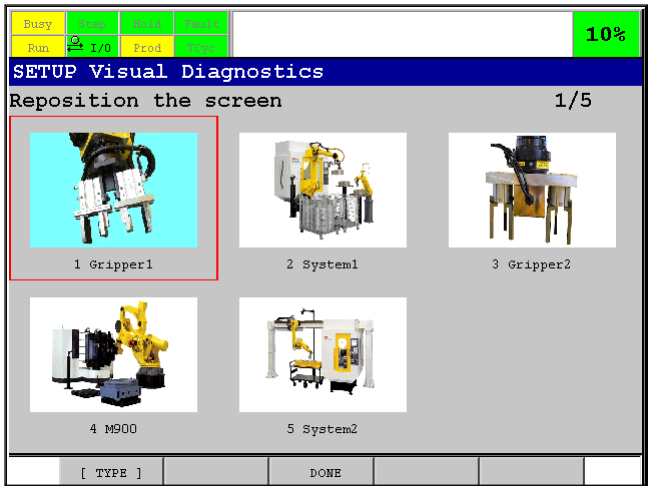


Fig.51.2(b) Reposition the Screen

   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.

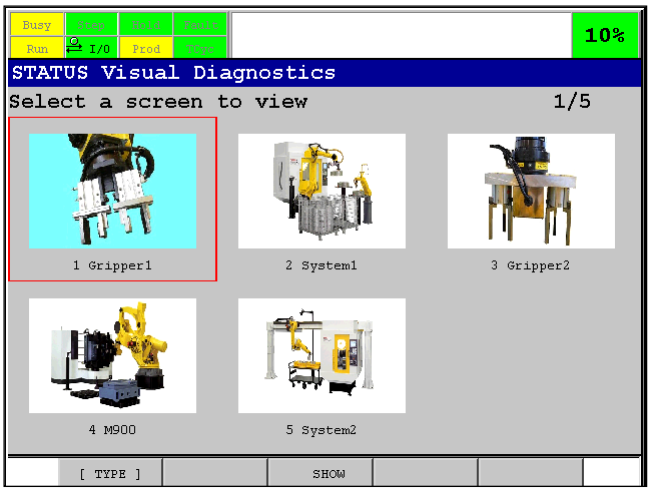


Fig. 51.3 (a) Select a Screen to View

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.

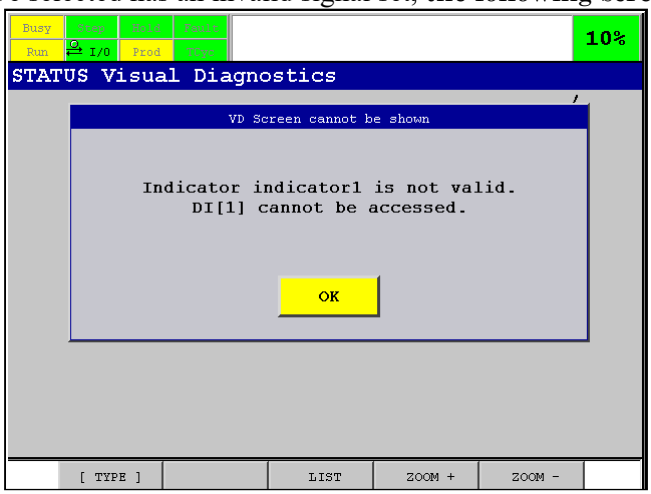


Fig.51.3(b) Invalid Indicator

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.

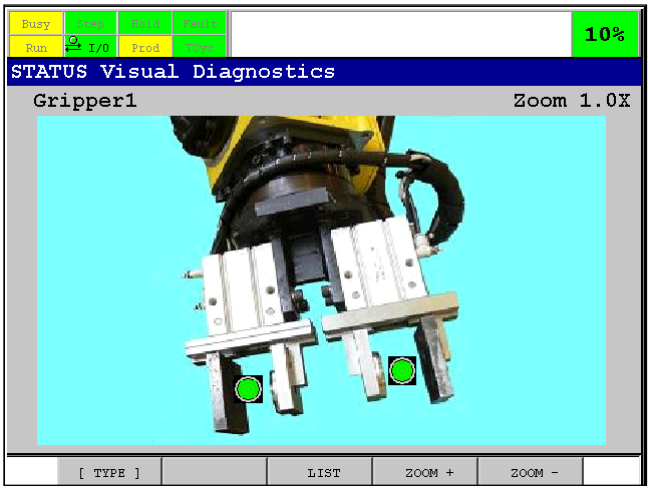


Fig.51.3(c) Visual Diagnostics Screen Display

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 R-30*i*B storage device.

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.

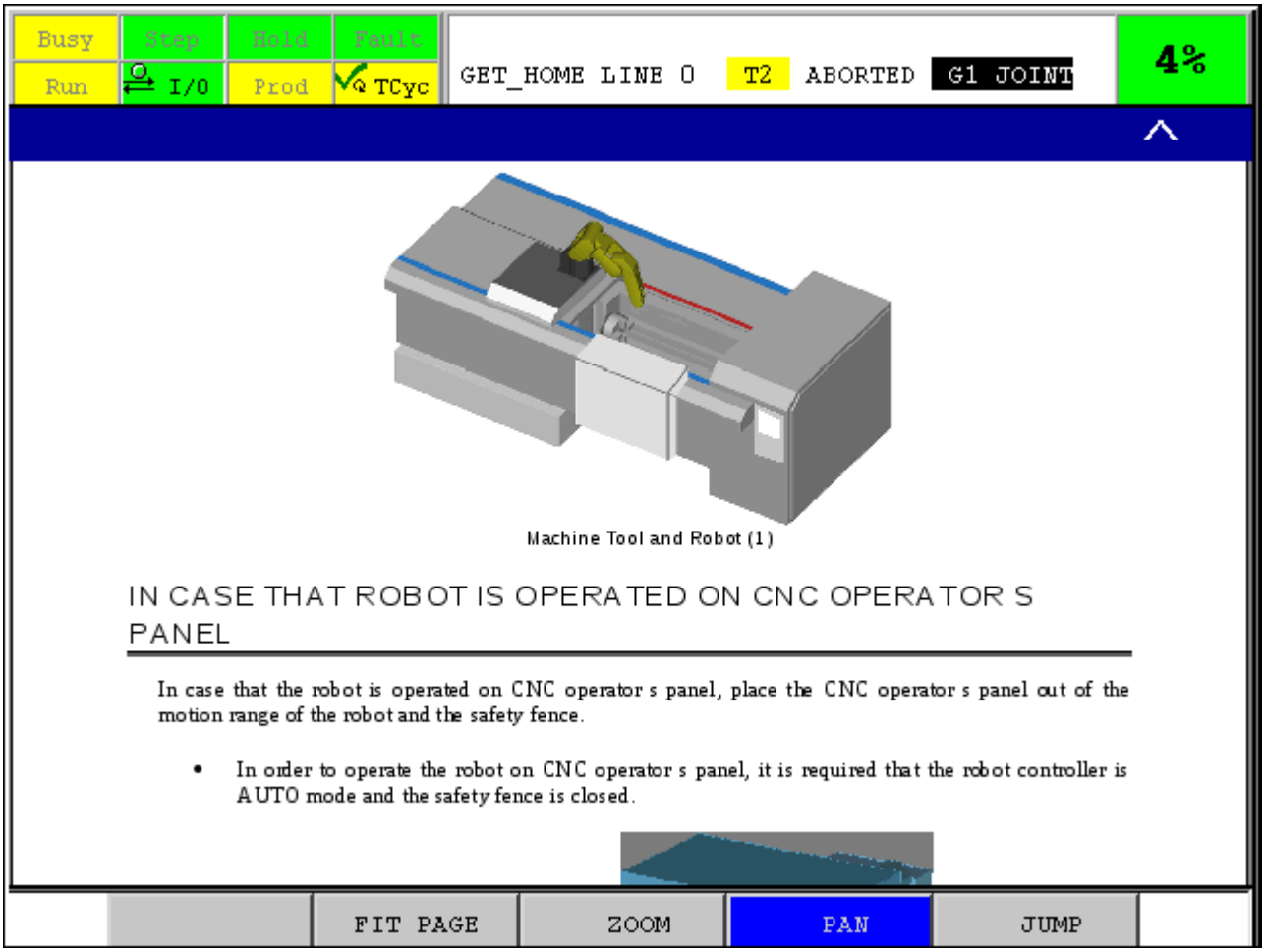


Fig. 52.2 PDF Viewer

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.

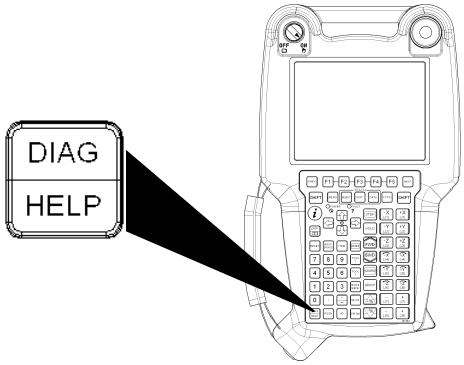


Fig. 53 DIAG/HELP key

- 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-2600-R664)
- Diagnostics Display: Alarm Cause/Remedy (A05B-2600-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.

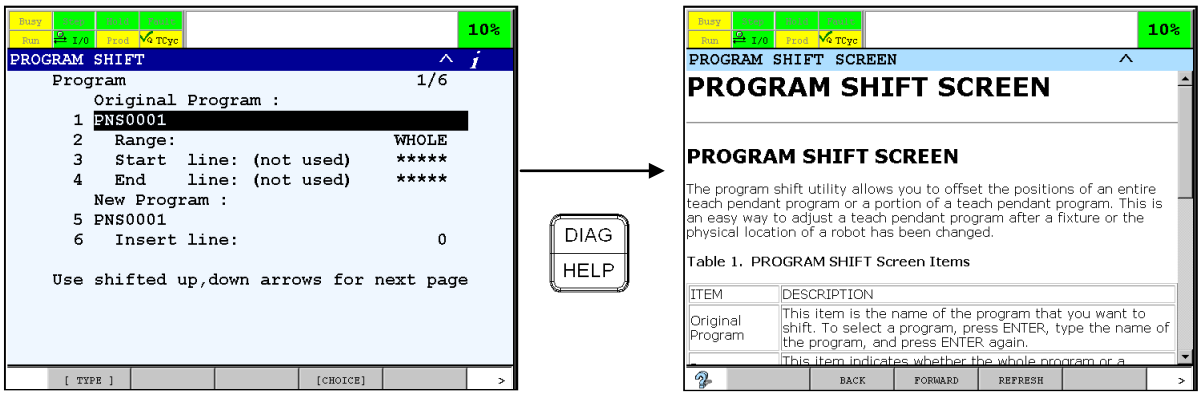


Fig. 53.1 Help for Program Shift Screen

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.

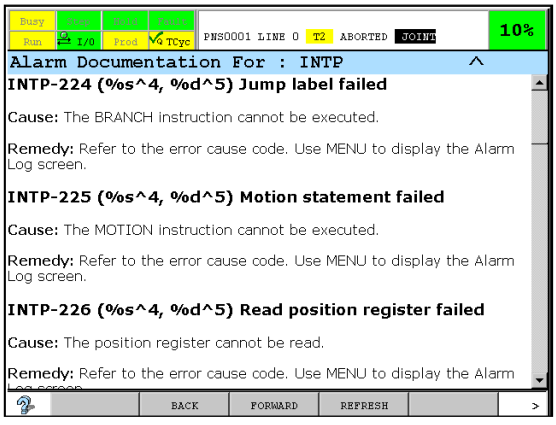


Fig. 53.2 (a) Alarm Cause and Remedy Display

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 DIAG alarm is displayed as follows.

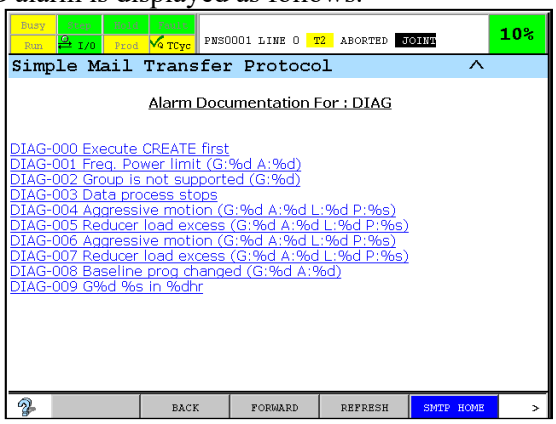


Fig. 53.2 (b) List screen for DIAG alarm

3 Press SMTP HOME (F5). The following screen will be displayed.

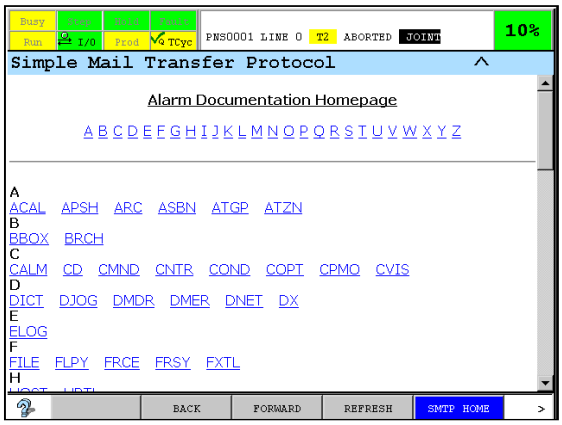


Fig. 53.2 (c) Alarm List screen

4 You can display the list of the alarm that you want to display the cause and remedy by selecting facility name in this screen.

5 Select the alarm to display the cause and remedy in the alarm list screen.

Procedure 53-3 Diagnostics display when the screen other than alarm screen is displayed and an alarm occurs.

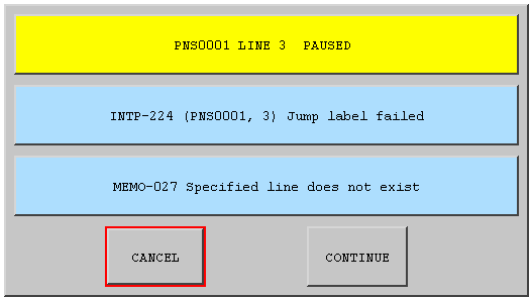
Conditions

- The screen other than the alarm screen is displayed 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.



Program Name and Status

The most recent alarm

Cause Code

Fig.53.2(d) Popup screen (When an alarm occurs)

3 In case that the “CONTINUE” button is pressed, the cause and remedy for the active alarm is displayed as follows. In case that the “CANCEL” button is pressed, the popup screen disappears and the previous screen will be displayed.

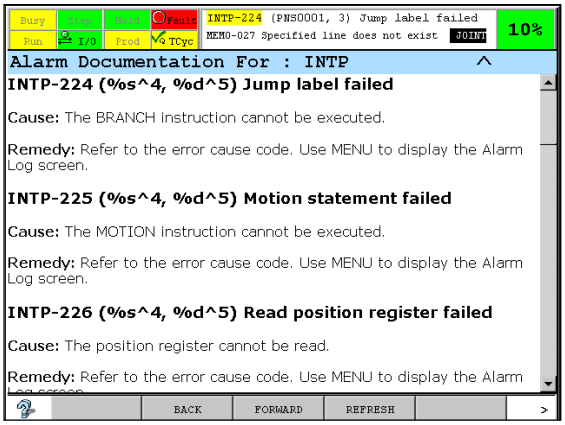


Fig.53.2(e) Alarm Cause and Remedy Display

Procedure 53-4 Diagnostics display when the screen other than alarm screen is displayed and no alarm occurs.

Conditions

- 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.

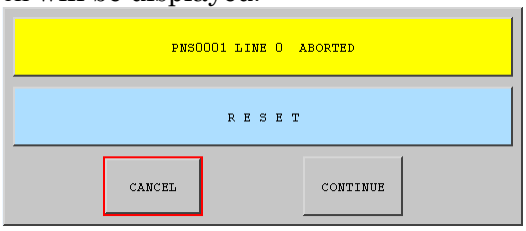


Fig.53.2(f) Popup screen (When no alarm occurs)

3 In case that the “CONTINUE” button is pressed, the list screen for DIAG alarm is displayed. Please refer to Procedure 53-2 to display the alarm cause and remedy from this screen. In case that the “CANCEL” button is pressed, the popup screen disappears and the previous screen will be displayed .

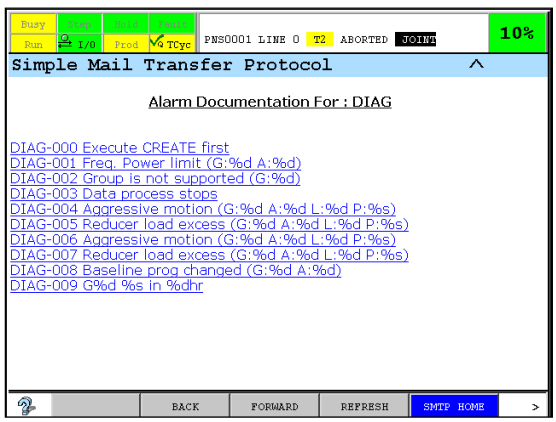


Fig.53.2(g) List screen for DIAG alarm

53.3 HELP/DIAGNOSTICS SCREEN

The operations in the online help screen and the alarm cause and remedy screen are as follows.

Change of the display by the link

By the selecting the link, you can display the figure, the table, or the other page specified by the link.

- To highlight the link that you want to select, use the up and down cursor keys.
- If the teach pendant equips the touch panel, touch the link to select.

Scroll of the screen

If the screen exceeds the window, the scroll bar is displayed. Please scroll the screen by the following ways.

- To scroll the screen, press the up or down cursor key while holding down [SHIFT] key.
- If the teach pendant equips the touch panel, touch the scroll bar.

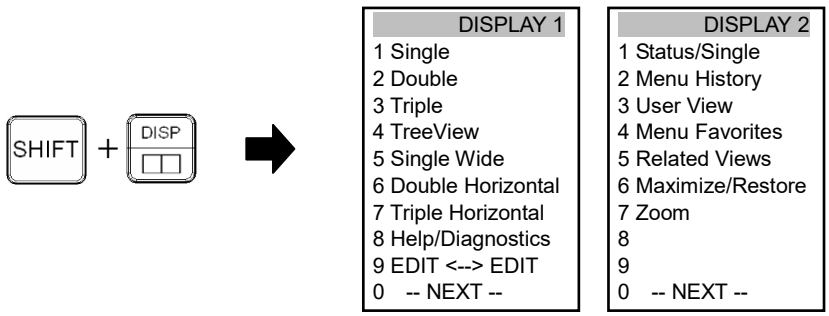
If the window on the teach pendant is divided, the Help/Diagnostics screen will be displayed as follows.

Table 53.3 Help/Diagnostics Display

Window mode	Display Location
Single	The Help/Diagnostics information replaces the current information in the window.
Status/Single	The Help/Diagnostics information replaces the current information in window on the right.
Double	The Help/Diagnostics information replaces the current information in the inactive window (window without focus).
Triple	The Help/Diagnostics information replaces the current information in the window without focus. For example in triple mode, the left window is window 1, the upper right window is window 2, and the lower right window is window 3. Then if the active window is window 2, the Help/Diagnostics information replaces the contents of window 3. If the active window is window 3, the Help/Diagnostics information replaces the contents of window 1, and so forth.

53.4 HELP/DIAGNOSTICS MENU

You can display an online help screen and an alarm cause and remedy screen by selecting an item in the Help/Diagnostics menu. The Help/Diagnostics menu will be displayed when the Help/Diagnostics item is selected in the DISPLAY menu that is displayed by pressing the [DISP] key with the [SHIFT] key.



(The items in the menu and the order of the items may differ depending on the software series and version.)

Fig. 53.4 (a) DISAPLY menu 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.

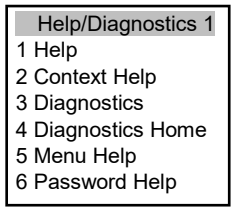


Fig. 53.4 (b) Help/Diagnostics menu

Table 53.4 describes the displayed screen by selecting each item is the Help/Diagnostics menu.

Table 53.4 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

Maintenance Reminder reminds users of periodic maintenances or part replacement for a robot system. By notifying preconfigured and customized maintenances with TP warnings, output to DO, and Notification Icon (software versions 7DC3/10 and later), it helps to maintain with appropriate intervals.
Maintenance items can be configured by the user. Some robot models have predefined items setup for recommended maintenance tasks (Required Maintenances).
To use this function, Maintenance Reminder option (A05B-2600-J771, A05B-2670-J766 for software versions 7DF1/04 and later) is required. When ordered with Maintenance Reminder/ Intelligent Grease Change Reminder Add-on (A05B-2600-J766, for software versions 7DC3/44 and later, for 7DC3 only) or Maintenance Reminder/ Intelligent Cable Change Reminder Add-on (A05B-2600-S511, for software versions 7DC3/44 and later or 7DF1/15 and later), the maintenance items will be changed to Intelligent Grease Change Reminder and Intelligent Cable Change Reminder respectively.
Also, when used in conjunction with Zero Down Time (ZDT) / ***i***RConnect, the maintenances can be checked on mobile devices other than Teach Pendant.

54.1 ADDITIONAL OPTION

Maintenance Reminder can expand its function by ordering additional options together. With additional options, the Required Maintenances of supported models change according to the option.

54.1.1 Intelligent Grease Change Reminder(J766)

Intelligent Grease Change Reminder (hereinafter referred to as i-grease change) estimates and notifies the appropriate grease change timing with operating conditions taken into account, instead of conventional elapsed time and running time. With severe conditions, changing grease earlier will prevent premature failure of reducers. With moderate conditions, it can prolong the grease change interval.
i-grease change shows estimated remaining life in %. It also shows the year and month the remaining life is expected to reach 0%, considering recent usage trend. Please refer to those values to plan grease change.
i-grease change requires ambient temperature to be set. Please refer to 54.3.3 to set ambient temperature.
On software series 7DF1 and later, the use of this function does not require additional option. With the Maintenance Reminder (A05B-2670-J766), the supported models will have i-grease change.
On software versions 7DC3/44 and later, the use of this function requires Intelligent Grease Change Reminder Add-on (A05B-2600-J766) in addition to Maintenance Reminder (A05B-2600-J771)
Please contact your local FANUC representative for supported models and software versions.

54.1.2 Intelligent Cable Change Reminder (S511)

Intelligent Cable Change Reminder (hereinafter referred to as i-cable change) estimates and notifies the appropriate cable change timing with operating conditions taken into account, instead of conventional elapsed time and running time.
i-cable change shows estimated remaining life in %. It also shows the year and month the remaining life is expected to reach 0%, considering recent usage trend. Please reference to those values to plan cable change.
The use of this function requires Intelligent Cable Change Reminder Add-on (A05B-2600-S511).
Please contact your local FANUC representative for supported models and software versions.

54.2 MAIN MENU

Main menu of Maintenance Reminder is displayed with following procedures.

1 Press the [MENU] key.
2 Select “STATUS”.
3 Press the F1, [TYPE].
4 Select “Reminder”.

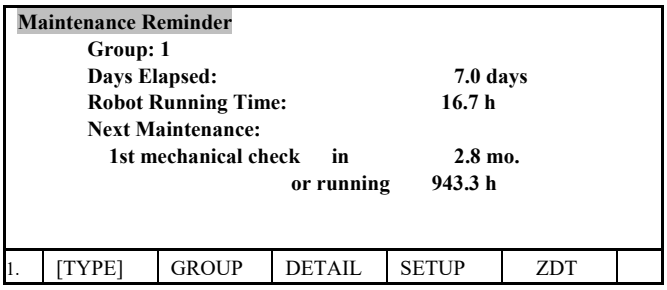


Table 54.2 items in main menu

Item	Description
Group	The group number, of which the maintenance items are currently shown.
Days Elapsed	It shows the days elapsed since the robot has started being used. It includes the time while the power is off. Adjusting the clock of the controller may result in increased elapsed time than in reality. All the groups in the controller share the same elapse time.
Robot Running Time	It shows the total time the robot has been moving. It does not include waiting time. Each group has individual counter.
Next Maintenance	Among enabled maintenance items, the item with shortest remaining time to the next maintenance will be shown with the remaining time. The items displayed will be limited to those setup with “Days Elapsed” or “Robot Running Time”.

If multiple groups exist, press F2 [GROUP] and the group number to change the group to show.
Press F3[DETAIL] to show menu for checking maintenance time and completing maintenance, described in the Section 54.4.
Press F4[SETUP] to show setup menu described in the Section 54.3.
Press F5[ZDT] to show ZDT setup menu described in the Section54.5. The F5 key may change to [iRConn] depending on the options.
On software versions 7DC3/10 and later, pressing [NEXT] key then F3 “OUT LIST” displays Monitored Out screen, F4 “RECORD” displays Maintenance Record screen. They will be described in later sections.

54.3 SETUP

- There are settings common to groups and items, such as when to send reminder notice.
- There are settings individual to each maintenance items, such as maintenance intervals.
- If passcode, described in the next section, is set, showing setup menu asks for the passcode. Set the passcode for the cases such as if the maintenance personnel and operators differ.

54.3.1 Common Setting

This menu is for settings common among all the groups of the controller, such as the timing to output the maintenance reminder warning.

Setup menu is displayed with following procedures.

1 Display the main menu in the Section 54.2.

2 Press the F4, [SETUP］.
3 If the passcode is set, the prompt line will ask “Please enter the passcode: “. Enter the passcode to display the setup menu.

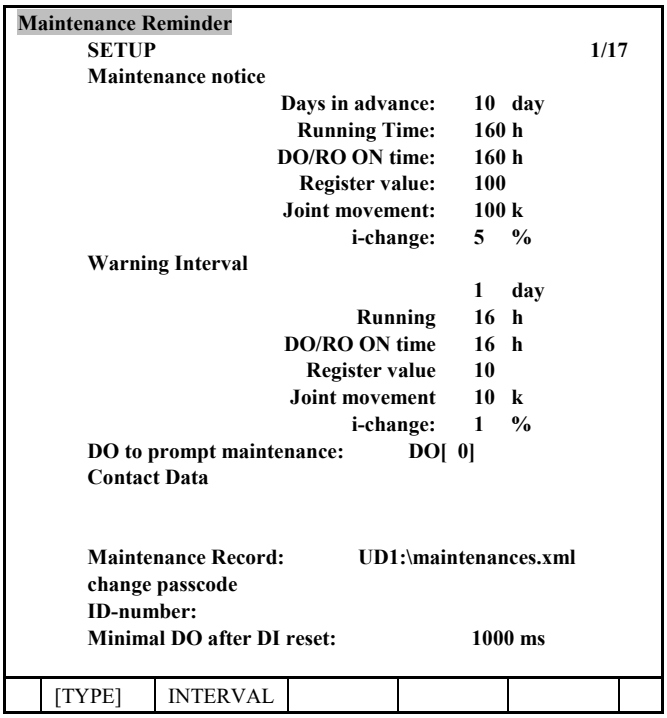


Table 54.3.1 Common setting items

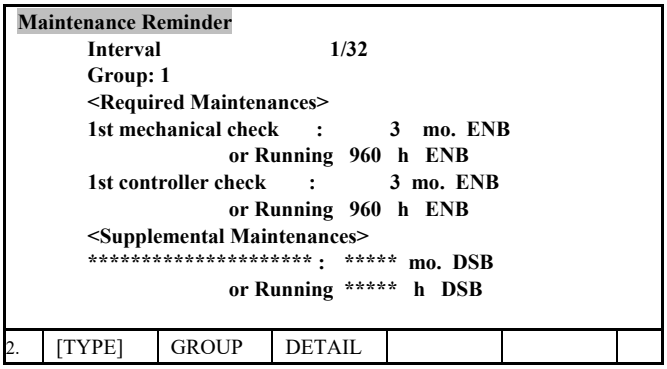
Item	Description
Maintenance notice	Set timing to post warnings and output DO. On Software versions 7DC3/10 and later, DO/RO ON time, Register value, and Joint movement can be set in addition to days and Running Time. On software series 7DF1 and later, or software versions 7DC3/44 and later with i-grease change (J766) ordered, or when i-cable change (S511) is ordered, if the groups include supported model, i-change is 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 start to remind maintenances. The unit of Joint movement is thousand revolutions (kRev) for rotational axis, km for linear axis. If item specific setup, described in later section, is done, it will be used instead of this common setting.
Warning Interval	Set interval to post warnings after the maintenance notice time, explained above, has been reached. If item specific setup, described in later section, is done, it will be used instead of this common setting.
DO to prompt maintenance	Set DO to be output after the maintenance notice time, explained above, has been reached. DO will be ON when any item reaches maintenance notice time, and will stay ON until “DONE” , or “COMPLETE” (on software series 7DF1 and later), operation, to be explained in the later section. Setting DO[0] will disable DO output. On software versions 7DC3/10 and later, item specific DO can be setup, which will be described in later section.
Contact Data	Any text up to 37 letters can be set for these 2 lines. When set, they will be displayed on the main menu in the Section 54.2. (They will not be displayed when no data is set.) Any data such as contact data for maintenance personnel can be set.
Maintenance Record	On software versions 7DC3/10 and later, Maintenance Record can be output to a file. This item sets destination file of the records. Maintenance Record will be output only after the file is created in Maintenance Record screen, which will be described in later section.
change passcode	Select this item and press ENTER to set passcode. (If passcode is already set,) After entering the current passcode, enter the new code twice to set the new passcode. Set “1111” to disable passcode protection. Do not forget the passcode, or settings cannot be changed and “DONE” , or “COMPLETE” (on software series 7DF1 and later), operation, to be explained in the later section, cannot be executed.

Item	Description
ID-number	Any text up to 10 letters can be set as ID of the controller. When set, it will be displayed on the main menu in the Section 54.2. (It will not be displayed when no ID is set.) It can be used for any purpose, such as for maintenance management.
Minimal DO after DI reset	On software versions 7DC3/10 and later, maintenance items can be reset by DI, which will be described in later section. This item sets the minimal time length of DO output when DI reset is done. When checking the DO by external PLC and such, set the length to be longer than the cycle length of the checking. The actual time length of DO output may be longer depending on the processing status of the controller.

54.3.2 Maintenance Item Setting

Interval setting menu is displayed with following procedures.

1 Display setup menu in the Subsection 54.3.1.
2 Press the F2 [INTERVAL］.



There are two types of maintenance item, Required Maintenances and Supplemental Maintenances. Required Maintenances are prepared for the robot models, and any maintenance can be added to Supplemental Maintenances. The configurability of two types differs, but they act in the same way after the setup.

Table 54.3.2(a) Maintenance Item settings

Item	Description
Group	The group number, of which the maintenance items are currently shown.
Name	The name of the maintenance item. The names of the Required Maintenances are fixed. The names of Supplemental Maintenances can be set freely, up to 21 letters.
Interval	The interval, with which the maintenance should be done. The intervals can be set for “Days Elapsed” and/or “Robot Running Time”, “DO/RO ON Time”, “Register value”, or “Joint movement”. The units of interval for “Days Elapsed” are days (day) or months (mo.) which can be switched by pressing F4/F5 keys while selecting the unit. The unit of interval for “Robot Running Time” and “DO/RO ON Time” is hours (h). The unit of interval for “Joint movement” is thousand revolutions (kRev) for rotational axis, or km for linear axis. Required Maintenances have the intervals setup. They can be shortened, but cannot be made longer. The interval for i-greae change and i-cable change will not be shown and cannot be changed.
ENB/DSB	Enable or disable the interval type of the maintenance item. Select the item and press F4/F5 key to switch. Required Maintenances are enabled and cannot be disabled. However, the maintenances that are meant to be done once, such as 1st mechanical check, will be disabled by “DONE” , or “COMPLETE” (on software series 7DF1 and later), operation, to be explained in later section.

If multiple groups exist, press F2 [GROUP] and the group number to change the group to setup.
On software versions 7DC3/10 and later, pressing F3 DETAIL displays Maintenance Item Setup screen, where type of interval and notification details can be setup.

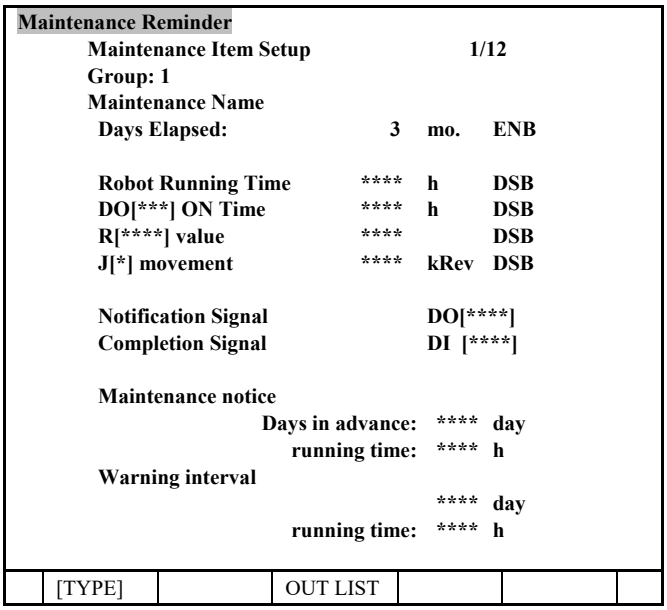
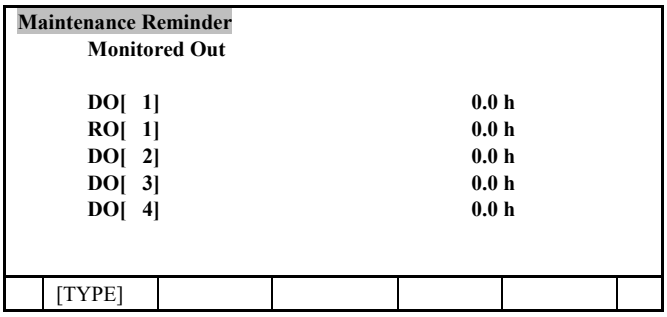


Table 54.3.2(b) Maintenance Item setup

Item	Description
Group	The group number, of which the maintenance item is currently shown.
Name	The name of the maintenance item. The names of the Required Maintenances are fixed. The names of Supplemental Maintenances can be set freely, up to 21 letters or 36 letters on software versions 7DC3/35 and later or software series 7DF1 and later.
Interval	The interval, with which the maintenance should be done. The intervals can be set for “Days Elapsed” and/or “Robot Running Time”, “DO/RO ON Time”, “Register value”, or “Joint movement”. i-greae change and i-cable change will not show the interval and it cannot be changed.
DO/RO ON Time	Count the time specified DO or RO is ON, and notify maintenance based on the time length. With the cursor over “DO” (“RO”) pressing F5 “RO” (F4 “DO”) will change the type of out. Enter the number of out to monitor ON time inside [] , and length of interval to the right. Up to 5 different outs can be monitored.
Register value	Notify the maintenance based on increase of specified register. Enter the number of register inside [], and length of interval, increase of register to the right. At the completion process, described later, the register value itself will not be reset. The next interval starts from the value at the time, and the notification will be based on the difference the current value and that value.
Joint movement	Notify the maintenance based on the amount of motion of the specified axis. Enter the number of the axis inside [], and length of interval, amount of motion to the right. The unit of interval is thousand revolutions (kRev) for rotational axis, and km for linear axis.
ENB/DSB	Enable or disable the interval type of the maintenance item. Select the item and press F4/F5 key to switch. Required Maintenances are enabled and cannot be disabled. However, the maintenances that are meant to be done once, such as 1st mechanical check, will be disabled by “DONE” , or “COMPLETE” (on software series 7DF1 and later), operation, to be explained in later section. “Robot Running Time”, “DO/RO ON Time”, “Register value”, and “Joint movement” can only be enabled when the other three are disabled.
Notification Signal	This item sets item specific DO to output when the item reaches notice time. Setting 0 disables item specific DO.
Completion Signal	“Completion of Maintenance”, described in later section, can be processed through DI input. To process completion with DI, Notification Signal, described above, needs to be set. Completion Signal needs to be input until the falling edge of the Notification Signal is detected. If Completion Signal is input when Notification Signal is ON, Notification Signal will be OFF after the process. If Completion Signal is input when Notification Signal is OFF, Notification Signal will be ON after the process, then OFF after the time set in “Minimal DO after DI reset”, described in 54.3.1.

Item	Description
Maintenance Notice	This item sets timings to post warnings and output DO. The timing for “Days Elapsed” and for the enabled type of interval can be set. When the remaining time or value to the next maintenance reaches the set time, warnings, DO, and Notification screen start to remind maintenances. If not set, the common settings described in 54.3.1 will be used.
Warning interval	Set interval to post warnings after the maintenance notice time, explained above, has been reached. If not set, the common settings described in 54.3.1 will be used.

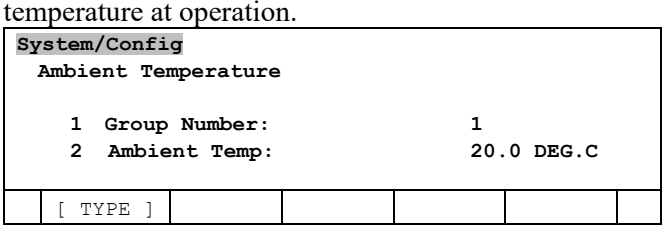
Pressing F3 “OUT LIST” displays the list of DO/ROs, of which ON time is counted. This screen is same screen as the one displayed when “OUT LIST” is pressed in Main Menu.



54.3.3 Ambient Temperature Setting

On software series 7DF1 and later, or on software versions 7DC3/44 and later with i-grease change (J766) ordered, for certain models, i-grease change is supported. This type of maintenance item requires ambient temperature setting. Set the ambient temperature with following procedures.

1 Press the [MENU] key.
2 Select “SYSTEM”.
3 Press the F1, [TYPE].
4 Select “Config”.
5 Select DETAIL of “Ambient Temperature”.
6 Enter 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.

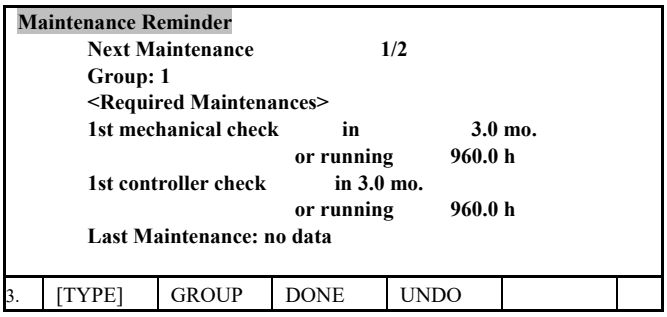
54.4 CHECK MAINTENANCE TIME AND COMPLETE MAINTENANCE

In the detail menu, remaining time to the next maintenance is displayed for each maintenance items. Also in this menu, an item can be “DONE”, or “COMPLETE” (on software series 7DF1 and later), after the completion of the corresponding maintenance, to reset the counter to start counting for the next maintenance.

54.4.1 Check Maintenance Time

Detail menu is displayed with following procedures.

1 Display the main menu in the Section 54.2.
   This menu only displays the item of the group with the shortest remaining days in “Days Elapsed” and the item with the shortest remaining hours in “Robot Running Time”, if different from the item with the shortest remaining days.
2 Press the F3 [DETAIL]



In detail menu, all the enabled items in the group are displayed with remaining time, beginning with Required Maintenances followed by Supplemental Maintenances. Upper lines represent the remaining time in “Days Elapsed” and the lower lines represent the remaining time in “Robot Running Time”. On software versions 7DC3/10 and later, the lower line shows “Robot Running Time”, “DO/RO ON Time”, “Register value”, or “Joint movement”. If only one type of the counter is enabled, the other type will display *’s.
On software series 7DF1 and later or software versions 7DC3/44 and later with i-grease change (J766) ordered, supported models also display i-grease changes. For i-grease changes, in addition to remaining life in %, it displays estimated month the grease reaches its life. The estimation is based on the 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 from the day before. It will not be updated other than on once-a-day estimation.
For supported models with i-cable change (S511) ordered, i-cable change that shows the life of mechanical cable is displayed. Similar to i-grease change, it will show the remaining life in % and estimated month that the cable reaches its life. The cable life is calculated while in motion, and will be updated when the display is shown.
If multiple groups exist, press F2 [GROUP] and the group number to change the group to show.

54.4.2 Maintenance Remind

When the maintenance time approaches, and remaining time/value is less than Maintenance notice configured in the Subsection 54.3.1, or 54.3.2 (on software versions 7DC3/10 and later) following warning is posted.
“DIAG-009 G(group number) (Name of the maintenance) in (remaining time) h”.
On 7DC3/10 and later:
“DIAG-009 G(group number) (Name of the maintenance) in (type of interval)(remaining time/value) (unit)”.

This warning is posted repeatedly with Warning Interval set in the section 54.3.1, or 54.3.2 (on 7DC3/10 and later) until the maintenance is processed “DONE”, or “COMPLETE” on software series 7DF1 and later, in the following section.
Also, if the output DO is set, the DO will be ON. This DO comes back ON even if switched OFF in I/O menu, unless the maintenance is processed “DONE.”, or “COMPLETE” on software series 7DF1 and later.
On software versions 7DC3/10 and later, Notification screen will notify the maintenance in addition to warnings and DO.

NOTE
This function only posts warning for the maintenance with the shortest remaining time per group. Even if several maintenances are within “Maintenance notice”, the warning for other maintenances will not be posted, unless the maintenance with the shortest remaining time is processed “done”, or “COMPLETE” (on software series 7DF1 and later).
On software versions 7DC3/10 and later, all maintenance items post warnings.

54.4.3 Upon Completion of Maintenance

When the Required Maintenances reach their maintenance time, follow the instructions in the mechanical unit operator’s manual to complete the corresponding maintenance, or contact your local FANUC representative. For Supplemental Maintenances, complete the corresponding maintenance in accordance with the design intent. Upon completion of the maintenance, “DONE”, or “COMPLETE” on software series 7DF1 and later, operation is necessary. If not, maintenance reminder warning will continue to be posted with set interval, and the next maintenance cannot be advised appropriately.

Follow the procedures to process the maintenance item as “DONE”, or “COMPLETE”.
1 Display the detail menu in the Subsection 54.4.1.
2 Select the item of which the maintenance is completed.
3 Press F3 [DONE], or [COMPLETE].
4 If set, enter the passcode, when asked “Please enter the passcode:”.
5 The prompt line will ask, “(Name of the maintenance) done?” Press F4 [yes].
If wrong item is selected, press F5 [no].
The selected item is processed as “DONE” and the count to the next maintenance starts.
For the item that requires maintenance only once, such as “1st mechanical check,” the count will not restart, but displays “done”.
On software versions 7DC3/46 and later, or 7DF1/17 and later, the name of the maintenance will not be shown in the prompt line.

If an item is processed “done” by mistake, without completing the maintenance, it can be canceled for once. Undo the “done” process with following procedures.
1 Display the detail menu in the Subsection 54.4.1.
2 Select the item, which is processed “done” by mistake.
3 Press the F4 [UNDO].
4 If set, enter the passcode, when asked “Please enter the passcode:”.
5 The prompt line will ask, “undo the last maintenance?” Press F4 [yes].
If wrong item is selected, press F5 [no].
The remaining time changes back to as if it was not processed “done”.
After processing the item as “done,” selecting the item in the detail menu displays the date of “done” at the bottom of the screen after “Last Maintenance:”. If the item is never “done” or “done” is canceled with UNDO, it displays “no data”. When it displays “no data”, it cannot be undone.

On software versions 7DC3/10 and later, the completion of the maintenance can be processed by Completion Signal DI input. When the DI set in Fig. 54.3.2 is input, the item will be processed done. DI needs to be input until the process is confirmed. If the Notification Signal DO is ON, it will be OFF when the process is finished. If it is OFF, it will be ON when the process is finished, then OFF after “Minimal DO

after DI reset" time set in the subsection 54.3.1. In either case, the process can be confirmed by detecting falling edge of Notification Signal DO.
Fig. 54.4.3 shows the flow of the system user should set up to utilize the completion process by DI input. If maintenance is done after the notification, with DO ON, DI should be input until DO turns OFF. If maintenance is done before the notification, with DO OFF, DI should be input until DO turns ON, then OFF. The DO would be ON at least for the time set in "Minimal DO after DI reset". It should be long enough so that the user system can detect DO ON. Since the signal for stopping DI input is always the DO change from ON to OFF, the user system for completion process can be set up as inputting DI until detecting falling edge of corresponding DO, regardless of whether DO was ON at the maintenance.
If Completion Signal DI is input by mistake, it needs to be "UNDONE" by operating the TP as described above.

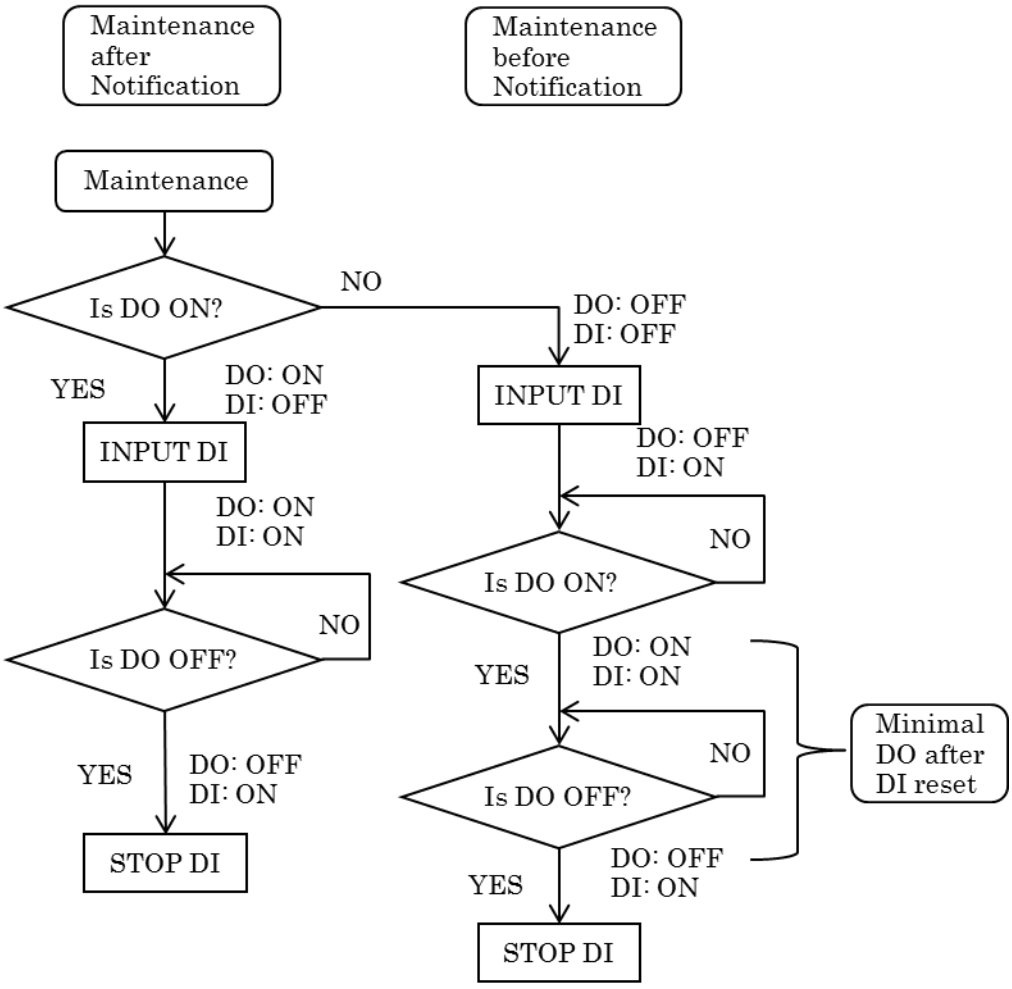


Fig. 54.4.3 The user system to utilize completion process by DI input

54.5 ZDT / *i*RConnect

Zero Down Time (ZDT) collects and analyzes data for preventing unexpected downtime, from robots through network. Used with ZDT, Maintenance Reminder enables checking maintenance timing remotely from devices other than Teach Pendant, such as PC or smart devices. It sends periodic data and reminder data to the configured server. The data can be checked from Web portal or ZDT application. Please refer to the ZDT manual for setting up ZDT. ZDT requires software option, Zero Down Time option (R854).
iRConnect is a function that sends email with information on the controller, which can be viewed with ***i***RConnect mobile application. Using in conjunction with ***i***RConnect, maintenance time can be checked on mobile devices. By sending periodic and reminder emails to the configured address, it enables checks on mobile devices. Please refer to the chapter on ***i***RConnect from Ethernet Function Operator's Manual for the use of ***i***RConnect, such as setup. ***i***RConnect requires software option, ***i***RConnect (R818).

54.5.1 Setup for Maintenance Reminder

To check Maintenance Reminder information on ZDT or ***i***RConnect, in addition to setup on ZDT or ***i***RConnect such as server or address, some setup is required on Maintenance Reminder side.
Display ZDT / ***i***RConnect settings menu with following procedures.

1 Display the main menu in the Section 54.2.
2 Press the F5 [ZDT] or [*i*RConn]

When ZDT / ***i***RConnect option is not ordered, [ZDT] / [*i*RConn] will not appear.
The display will differ depending on the option ordered, but settings are basically the same.

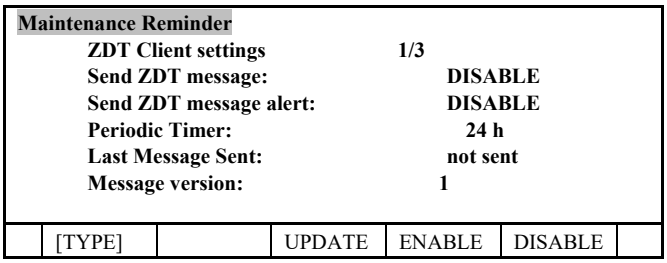


Table 54.5.1 Maintenance Reminder setup items for ZDT

Item	Description
Send ZDT (*i*RConnect) message	Enable/Disable ZDT (***i***RConnect) message of Maintenance Reminder information. When All Emails in SETUP ***i***RConnect menu is disabled, it cannot be enabled.
Send ZDT (*i*RConnect) message alert	Enable/Disable message to "Alert Addr" (SMS) when Maintenance Reminder warning is posted. When "Send ZDT (*i*RConnect) message" is disabled, it cannot be enabled.
Periodic Timer	Set interval with which ZDT (***i***RConnect) message is to be sent to "To Addr" when Send ZDT (*i*RConnect) message is enabled.
Last Message Sent	It displays the last time Maintenance Reminder created ZDT (***i***RConnect) message. If it is different from the time stamp on Maintenance Reminder screen on ZDT (***i***RConnect) application, it indicates email was not sent/received properly.
Message version	The version of the message sent to the ZDT server. Set it to 0, if Zero Down Time package version 1.0 is used.

This setup is valid for all groups and maintenance items.
If for some reason, such as trouble on servers, email is not sent/received properly and information on ZDT(***i***RConnect) is not updated (when time stamp on ZDT(***i***RConnect) and "Last Message(Email) Sent" mentioned above do not correspond), new Message(Email) can be created/sent at any time by pressing F3 [UPDATE].

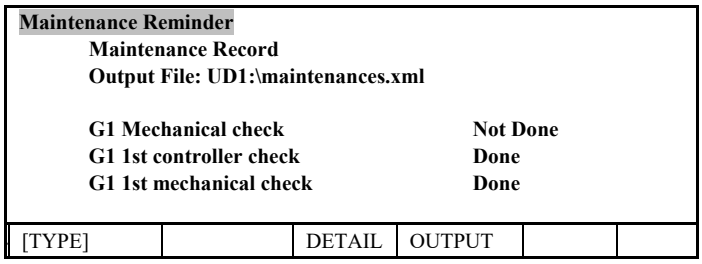
54.6 MAINTENANCE RECORD

On software versions 7DC3/10 and later, maintenance records are saved. The record is created at the time of first warning, and updated when past due and when completed. 10 last records can be viewed on TP. By creating maintenance record file beforehand, old records can be saved and viewed as xml file.

54.6.1 Display Maintenance Record

Display Maintenance Record screen with following procedures.

1 Display the main menu in the Section 54.2.
2 Press the [NEXT] key.
3 Press the F4 RECORD



Pressing F3 DETAIL shows detail information of the item selected.
Pressing F4 OUTPUT creates the file specified and writes the information of the items that are “Done”. Once the file is created, the information of the items “Done” thereafter are added to the file.

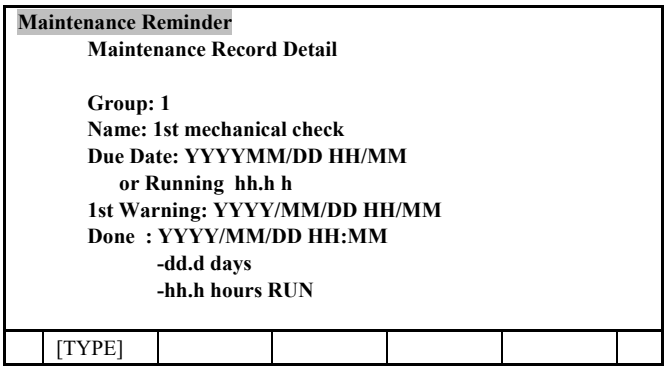


Table 54.6.1 Maintenance Record items

Item	Description
Group	Group number of the item displayed.
Name	Name of the item displayed.
Due Date	Due date of the item displayed. The first line shows the due date calculated from the interval set in “Days Elapsed”. If not set, *’s will be displayed. The second line shows remaining time/value of the other type of interval. The negative value shows time/value past the due. If not set, the second line will not be shown.
1st Warning	It displays the date the first warning of the item is posted. If no warning is posted before processed “DONE”, *’s will be displayed.
Done	It displays the date the item is processed “DONE”. The second and third line shows the time/value past due at the completion process. The negative value shows remaining time/value when the item is processed “DONE” before due.

54.6.2 Maintenance Record File

Maintenance Record File is created in xml format at specified directory with specified name.

Table 54.6.2 Items in Maintenance Record File

Node/Element	Description
MAINT_RECORD	Maintenance Record Data
CONT	Controller
ID	ID set in 54.3.1. If not set, it would be other type of ID that is automatically set.
ROBOT	The group of the Maintenance Record Data
Model	The name of the model of the group. Ex. R-2000*i*C/165F
Group	The group number
ITEM	Maintenance Item
ID	The ID number of the maintenance item. It corresponds to the order the item is shown in the screen described in 54.3.2.
Name	The name of the maintenance item
Type	The type of the maintenance item expressed in numbers. 1: “Days Elapsed” only. 2: “Robot Running Time” is set. 3: “DO/RO ON Time” is set. 4: “Register value” is set. 5: “Axis motion” is set. On software series 7DF1 and later or on software versions 7DC3/44 and later with i-grease change (J766) ordered, 7: “i-grease change” is set. Supported models with i-cable change (S511) ordered, will have 8: “i-cable change” set.
HIST	The record of maintenance
DueAct	Due date and time in terms of “Days Elapsed”
DueOther	Due in other types of timer. The negative value means time/value till the due. The positive value means time/value past due. If the item is “done”, it is same as DonePast.
DueType	The type of the timer that reached due, expressed in number. 0:Not due. 1: “Days Elapsed”. 2: “Robot Running Time”. 3: “DO/RO ON Time”. 4: “Register value”. 5: “Axis motion”. On software series 7DF1 and later or on software versions 7DC3/44 and later with i-grease change (J766) oredered, with supported models, 7: “i-grease change”. With i-cable change (S511) ordered, 8: “i-cable change”.
Warn	The date and time of the first warning.
DoneDate	The date and time when it is “done”.
DonePast	The difference between due and the timer when it is “done”. It is same as DueOther.

Example:

```
<?xml version="1.0"?>
<MAINT_RECORD>
<CONT ID="TEST ">
 <ROBOT Group="1" Model="R-2000iC/165F">
  <ITEM ID="1" Name=" 1st mechanical check" Type="2">
   <HIST DueAct="2016/3/16 16:08" DueOther="-3839.8 hours RUN" DueType="0" Warn=""
DoneDate="2015/03/18 15:44:38" DonePast="-3839.8 hours RUN"/>
  </ITEM>
  <ITEM ID="11" Name="TEST1" Type="5">
   <HIST DueAct="" DueOther=" -100.0 kRev J[1]" DueType="0" Warn="2015/03/18 15:43:50"
DoneDate="2015/03/18 15:44:06" DonePast=" -100.0 kRev J[1]"/>
  </ITEM>
 </ROBOT>
</CONT>
</MAINT_RECORD>
```

54.7 ROBOT SETTING

When executing the robot setting in the Controlled Start, settings of the required maintenances, such as intervals, are reset with R-30*i*B. The data such as remaining days will not be reset. With R-30*i*B Plus and later, both the setting and the data will not be reset.
If the software is updated, there is a possibility that the required maintenances are changed. In such cases, the new required maintenances will be set. In doing so, R-30*i*B may overwrite the supplemental maintenances by the new required maintenances. With R-30*i*B Plus and later, normally, it will not overwrite. However, if the sum of the numbers of the required maintenances and the supplemental maintenances exceed the maximum, it will overwrite.

54.7.1 i-grease Change

The software versions that supports i-grease change depend on the models.
On software series 7DF1 and later, if the software is updated from the version that does not support i-grease to the one that does, and the robot setting is executed, the conventional grease change will be changed to the i-grease change. (If the robot setting is not executed after the backup is restored, it will hold the conventional grease change.) On software series 7DC3, adding i-grease change (J766) will change the grease change to i-grease change without robot setting.
In doing so, it will set the ratio of the remaining hours until the maintenance in the conventional grease change as the initial value of the remaining life of the i-grease change. (After the grease change, when the Completion process is done, the remaining life will be reset to 100%.) Therefore, if changed to i-grease in the middle of the use, the maintenance remind will be at the different time from if i-grease is used from the beginning, even if the usage conditions are the same.

54.7.2 i-cable Change

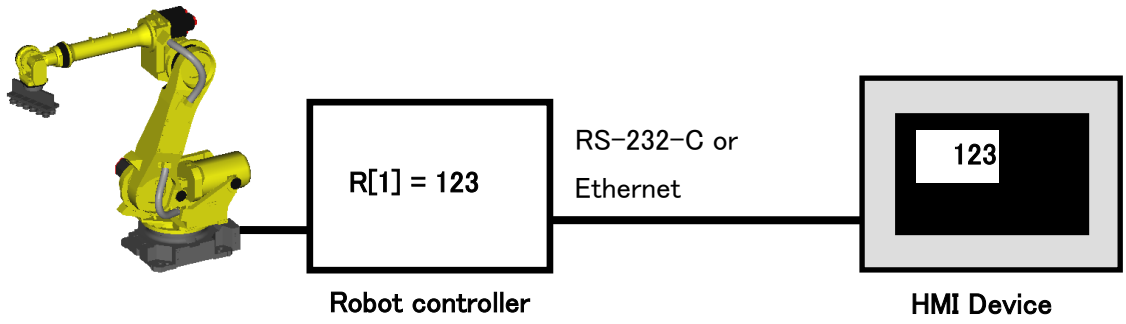
The software version that supports i-cable change depends on the models.
When i-cable change (S511) is added, or for multi-arm configuration, when the software is updated from the version that does not support one of the models to the one that does, the conventional cable change will be changed to the i-cable change.
In doing so, similar to 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 use will result in the different timing of the maintenance due compared to if used from the beginning, even if the usage conditions are the same.

55 HMI DEVICE COMMUNICATION

The HMI device communication function allows to communicate with external HMI devices etc. that are connected via RS-232-C or Ethernet via Modbus/TCP or OPC UA. Various robot data, such as DI, DO, Register and so on, can be read or written from the HMI device etc. by using this function.

This function operates as both:
Modbus slave, communicating with HMI device etc. that is the Modbus master
And,
OPC UA Server, communicating with OPC UA Clients.

HMI device can design the screen by putting lumps, switches and so on. The assignment of robot data to the lumps, switches and so on can be defined by user. For example, if robot register is assigned to the display panel on HMI device, the register value can be displayed on the screen of HMI device.



This function is standard when "Standard setting (R651)" is selected, but the following version conditions must be met in order to use each function. When "North America Setting (R650)" is selected, "HMI device (R553)" option is needed to use this function.

- Modbus slave function is supported from version 7DC2 series P07 and all versions after 7DC2 series. “HMI device (R553)” doesn’t support Modbus slave function in 7DC1 series or before.
- OPC UA server function is supported from 7DF1 series P19 or after and all versions after 7DF1 series.
- OPC UA server on ROBOGUIDE is supported from 7DF1 series P24 or after, 7DF3 series P04 or after, and all versions after 7DF3 series.

The response time of this function may 3-18ms usually but may take 100ms or more and is not guaranteed. It depends on the robot controller load.
If you want to get real time current position, use “RSI INTERFACE” in section 58 or “HIGH SPEED POSITION OUTPUT” in section 68.

OPC UA server function supports Publisher and MonitoredItem so client can use subscriber.
Minimum data acquiring interval of MonitoredItem is 100ms.
In OPC UA server function, OPC UA client will read/write large array data so we recommend to use the client soft which support index range, which enables to read/write specific range only.

⚠ WARNING
This function checks the received data and performs the proper process only when the robot controller receives the data sent from HMI device etc. This function cannot detect the situation that the communication from HMI device etc. is stopped accidentally. Therefore, this function must not be used to transfer the safety data. For example, when HMI device etc. sends STOP request to robot, there is no guarantee that the robot is stopped by the request. It is very dangerous to make safety system by using this function.

NOTE
When this function is used to communicate with HMI device etc., please test the whole 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 followings 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 followings 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 OF HMI DEVICE

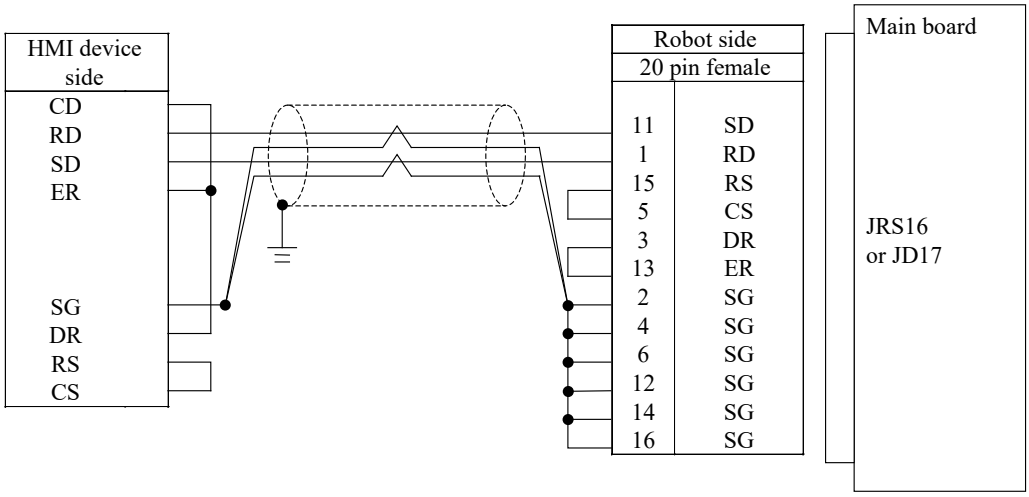
55.1.1 Modbus RS-232-C Connection

Cable

The following diagram shows the connection of RS-232-C. Please refer to the manual of the HMI device about the connection of HMI device side.

A-cabinet

The cable of the ROBOT CONTROLLER side is connected to JRS16 or JD17 connector on the main board.

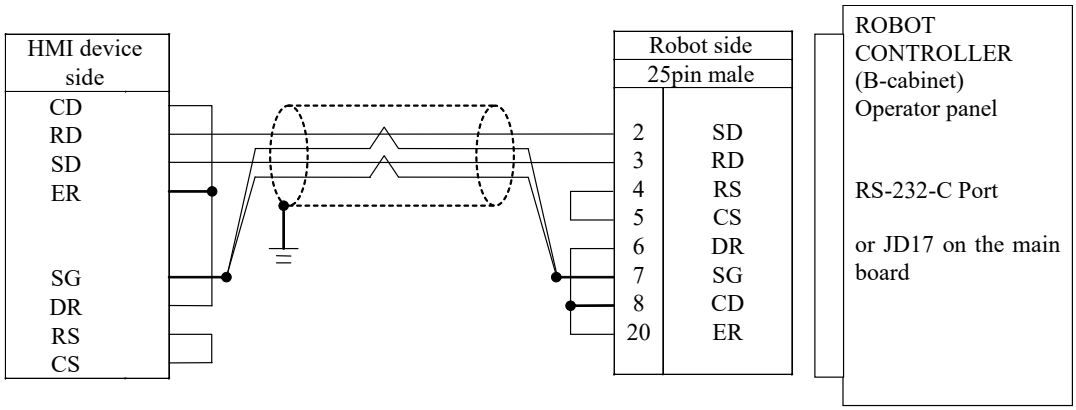


Specification of the connectors and the cable are as following:

- •Connector PCR-E20FS (Honda Tsushin Kogyo)
- •Housing PCR-V20LA (Honda Tsushin Kogyo), or compatible connector
- •Shielding quality Entirely shielded twisted pair cable is recommended.
  For protection against the noise, the twisted pair wire should be used for pairs of RD and SG, SD and SG.

B-cabinet

The D-Sub 25pin is connected to the RS-232-C port in front side of ROBOT CONTROLLER operator panel or JD17 connector on the main board.

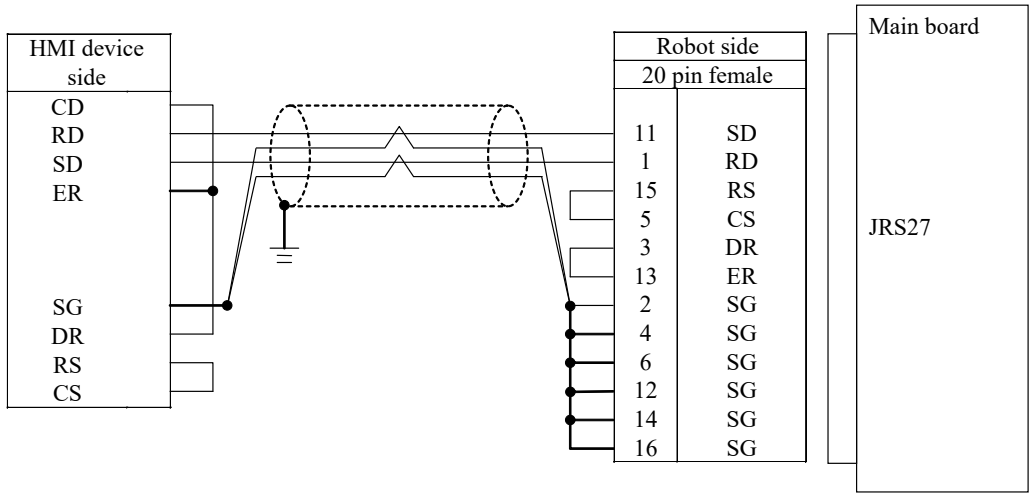


Specification of the connectors and the cable are as following (Operator panel):

- •Connector D-Sub25pin male DBM-25P (ANSI/EIA-232)
- •Housing DB-C2-J9 (ANSI/EIA-232)
- •Shielding quality Entirely shielded twisted pair cable is recommended.
  For protection against the noise, the twisted pair wire should be used for pairs of RD and SG, SD and SG.

R-30*i*B (Mate / Compact / Mini) (Plus)

The cable of the ROBOT CONTROLLER side is connected to JRS27 connector on the main board.



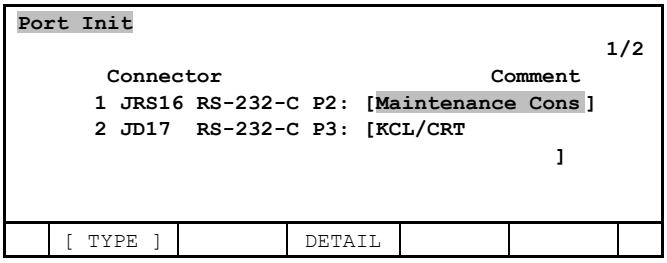
Specification of the connectors are as following:

- •Connector PCR-E20FS (Honda Tsushin Kogyo)
- •Housing PCR-V20LA (Honda Tsushin Kogyo), or compatible connector
- •Shielding quality Entirely shielded twisted pair cable is recommended.
  For protection against the noise, the twisted pair wire should be used for pairs of RD and SG, SD and SG.

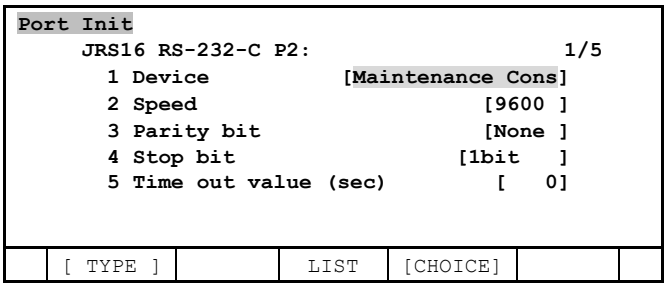
Setting of Serial Port

To use HMI device communication via RS-232-C, it is necessary to set "HMI dev(MODBUS)" in Port Init menu on Robot controller.

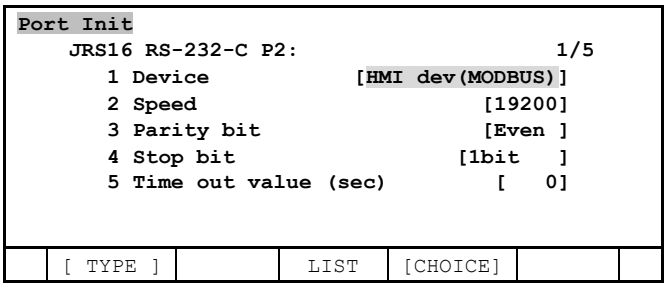
1 Press the [MENU] key on teach pendant, and select "6 SETUP".
2 Press the F1[TYPE] key, and select "Port Init".



3 Set cursor on "JRS16 RS-232-C P2:" line, and press the F3(DETAIL) key. The detail menu of "JRS16 RS-232-C P2:" is displayed.



4 Set cursor on "Device" line, and press the F4[CHOICE], and select "HMI dev(MODBUS)" from the list. There are "HMI device" and "HMI dev(MODBUS)" in the list. It is necessary to set "HMI dev(MODBUS)" for Modbus communication. It is not possible to use "HMI device" and "HMI dev(MODBUS)" at the same time.



5 Setup "Speed", "Parity bit" and "Stop bit" as same with the HMI device.

Setup slave address

The slave address of this function is set in the system variable $SNPX_PARAM.$MODBUS_ADR. The default value is 1. The slave address of Modbus RTU can be changed by this system variable. This function responds to the request for the specified slave address only. The request for the slave address 0 is processed as broadcast.

55.1.2 Modbus/TCP Slave Ethernet Connection

Setting of TCP/IP

To use HMI device communication via Modbus/TCP Slave Ethernet, it is necessary to setup TCP/IP on Robot controller in advance.
Details on the Ethernet interface and TCP/IP configuration can be found in "SETTING UP TCP/IP" section in the *Internet Options Setup and Operations Manual*.

Setting of Modbus/TCP Slave Ethernet connection

Modbus/TCP Slave Ethernet connection is disable by default. To use Modbus/TCP Slave Ethernet connection, please set the number of HMI devices etc. that are connected at the same time to the system variable $SNPX_PARAM.$NUM_MODBUS (Default is 0). When a Modbus/TCP Slave Ethernet connection is requested from HMI devices etc. which would exceed the setting value, the HMI device etc. whose latest communication is the oldest is disconnected automatically.

Unit ID

Robot controller does not use Unit ID in Modbus TCP communication data. Robot controller responds all requests from an HMI device etc. regardless of Unit ID.

Port number

The port number of this function is set in the system variable $SNPX_PARAM.$MODBUS_PORT. The default value is 502. The port number of this function can be changed by this system variable.

Modbus TCP Server (R800)

Modbus TCP Server (R800) option is independent from this function. But when this option is loaded, if $SNPX_PARAM.$NUM_MODBUS is set to 1 or greater value, "PRIO-090 SNPX communication error", "HRTL-048 Address already in use" occurs and Modbus/TCP Slave Ethernet communication of this function does not work. It is because this function and Modbus TCP Server use the same port number (502). By changing the port number of this function, both this function and Modbus TCP Server can be used at the same time.

55.1.3 OPC UA Server Ethernet Connection

OPC UA is multi-purpose and flexible industrial communication protocol which enables data exchange between multi-vendor products and different operating systems. This function enables the 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.

Limitation

The following limitations exist when using the FANUC robot OPC UA Server.

- Maximum of ten (10) OPC UA clients can communicate simultaneously.
- This function can be used only with CD38A and CD38B RJ45 Ethernet port connections on the robot controller. This function cannot be used with the CD38C RJ45 Ethernet port connection.
- Up to 2 subscriptions can be registered per 1 connection. (It also supports the Client/Server method.)
- 50 items can be monitored per 1 subscription but minimum data acquiring interval for the item is increased by 100ms per 1 item like 100ms for 1st item, 200ms for 2nd item, 300ms for 3rd item. Normal read / write (Data access(DA)) doesn’t have such limitations.

Setting of TCP/IP

To use HMI device communication via the FANUC robot OPC-UA Server, it is necessary to setup TCP/IP on the Robot controller in advance. See "2.4 SETTING UP TCP/IP" section in the *Ethernet Function OPERATOR'S MANUAL (B-82974EN)* or "2.5 SETTING UP TCP/IP" section in the *Internet Options Setup and Operations Manual*.

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. Please see the "55.2 MODBUS DATA MODEL" sections for details about assigning and accessing robot data.

55.2 MODBUS DATA MODEL

55.2.1 Modbus data model

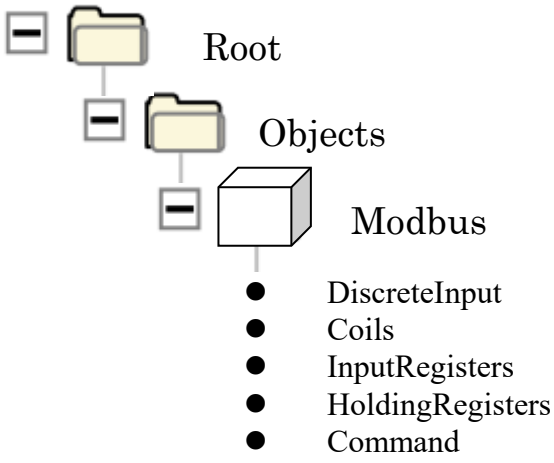
Modbus bases its data model on a series of tables.

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 Modbus data model

This section applies only when using 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 displayed below.



There are other folders just under root folder but is not used.

The following table describes node and data types of the five (5) leaf nodes under the "Modbus" object.

Node	NameSpace	Node ID	Data type
DiscreteInput	1	301	Boolean
Coils		302	Boolean
InputRegisters		303	UInt16
HoldingRegisters		304	Int16
Command		305	String[]

- Leaf nodes other than Command node are array. The index range should be supported on the OPC UA client to support partial read or write for the leaf nodes.
- For example, each X, Y, Z in position register cannot be assigned discrete Node ID.

Please Note: 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 address area 1 ~ 65536. This function defines the following correspondence of Modbus address to Robot data. The HMI device etc. can read or write the corresponded robot data by accessing Modbus address.
OPC UA clients can read or write the desired robot data by accessing the index which 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	
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	55.4 DYNAMIC ASSIGNMENT TO HOLDING REGISTERS using this area.	
	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 1 ~ 16384 of Holding Registers can be assigned to various Robot data. (→ Assignment of Holding Registers)

The HMI device etc. cannot write to the signals corresponded 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, address range is specified as 1 ~ 65536. But some HMI devices etc. specify as 0000 ~ FFFF (start from 0 and specified as hexadecimal). In this case, please convert the specified address as subtraction of 1 and change to hexadecimal.

55.2.4 Modbus Function Code

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) memory space. It is possible to assign 32 bit data to the continuous 2 addresses, and to assign string data to the continuous plural addresses.

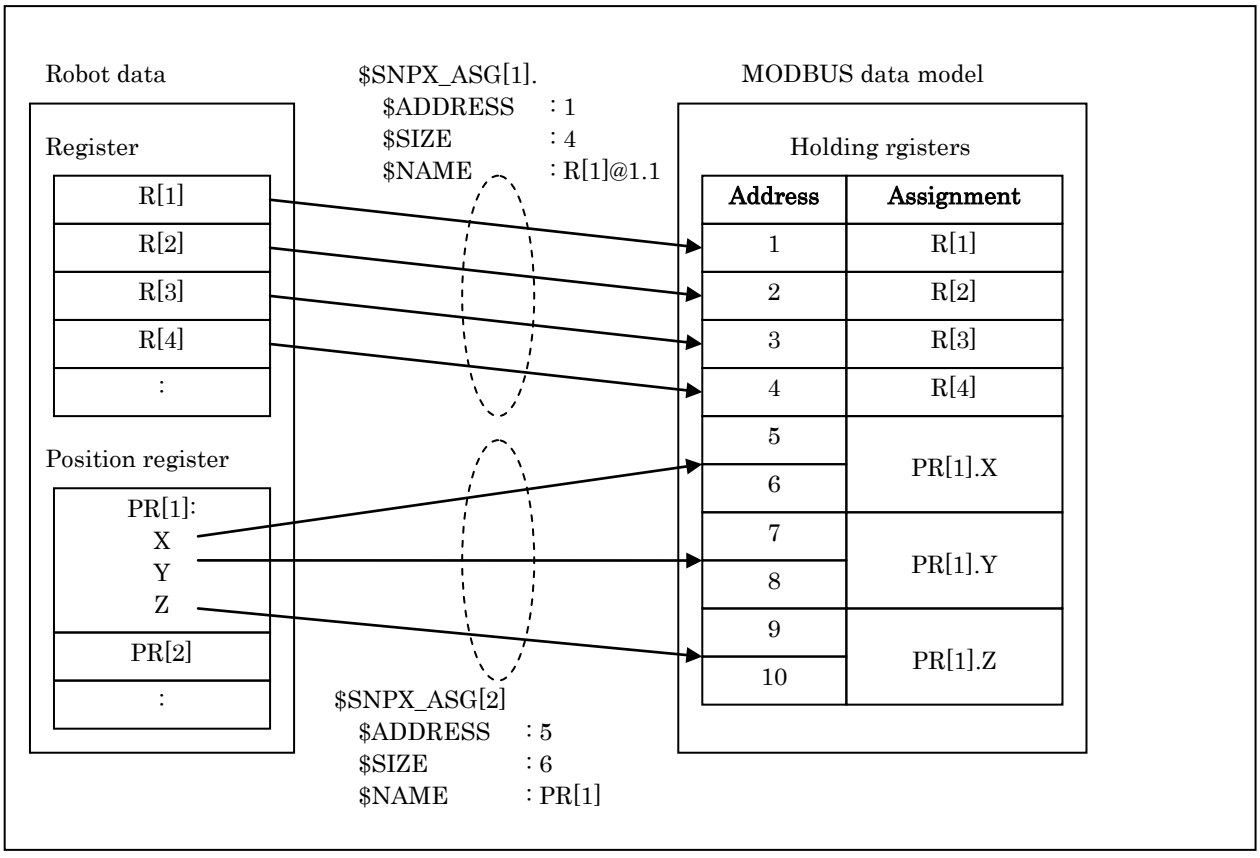
The following robot data can be assigned to the Holding Registers.
- Register data
- Position resister data
- String register data
- Current position
- Alarm history
- Program execution status (Running program name and line number)
- System variable data
- Comment of Register, Position register, String register and I/O
- I/O data and simulation status
- Integrated PMC address data
- Symbol and comment of Integrated PMC address.

The assignment between Robot data and Holding Registers is defined by setting of $SNPX_ASG. The $SNPX_ASG is 80 arrays of $SNPX_ASG[1]~[80], every element has the following variables. Various robot data can be assigned to the Holding Registers by setting $SNPX_ASG.

Variable in $SNPX_ASG	Description
$ADDRESS	Meaning: Start address of Holding Registers to assign Range: 1~16384
$SIZE	Meaning: Number of Holding Registers to assign. Range: 1~16384

Variable in $SNPX_ASG	Description
$VAR_NAME	Meaning: String to specify robot data. It specifies the robot data type and index by string. The meaning of this variable is different according to the assigned data type. Please refer to the description in each robot data. Example: R[1] : Register R[1] PR[1] : Position register PR[1] POS[1] : Current position of group 1 robot
$MULTIPLY	Meaning: Multiply for assigned data. It specifies the conversion type from assigned data to Holding Registers. The meaning of this variable is different according to the assigned data type. Please refer to the description in each robot data. Example: When register is assigned and the register value is 123.45, the value of the Holding Registers is the following. When $MULTIPLY is 1, Holding Register is 123. When $MULTIPLY is10, Holding Register is 1235 When $MULTIPLY is 0.1, Holding Register is 12

The following figure shows a example that robot register R[1~4] are assigned to Holding Register address 1 ~ 4 and X,Y,Z of robot position register PR[1] are assigned to Holding Register address 5 ~ 10.



The default setting of $SNPX_ASG is the following.

System variable	Value
$SNPX_ASG[1].$ADDRESS	1
$SNPX_ASG[1].$SIZE	10000
$SNPX_ASG[1].$VAR_NAME	R[1]@1.1
$SNPX_ASG[1].$MULTIPLY	1

By the default setting, the assignment of Holding Registers is the following.

Robot data	Address of Holding Registers	Example
Register R[*x*]	*x*	R[1] ⇔ Address 1

Note: Data type is 16 bits signed integer. Value is rounded off to no decimal place. Range of value is from -32768 to 32767. If value is out of the range, value cannot be read and write crrectly.

55.3.1 Data type of Holding Registers

Data type of Holding Register is 16 bits integer. Data range of Holding Register is -32768 ~ 32767. The value of assigned robot data is converted as follows according to the data type of the assigned data.

Assigned data	Description
16 bits signed integer Range: -32768 ~ 32767	The Holding Register data is the same as the assigned data.
32 bits signed integer Range: -2147483648 ~ 2147483647	The continued 2 Holding Registers express a 32 bits signed integer. The former address is lower 16 bits data, and the latter address is upper 16bits data.
Real	The continued 2 Holding Registers express a single precision binary real format of IEEE 754. The former address is lower 16 bits data, and the latter address is upper 16bits data. In case of that real type robot data is assigned to Holding Registers, when $MULTIPLY is 0, the robot data is assigned to Holding Registers as real type data. When $MULTIPLY is not 0, the robot data is assigned to Holding Registers as 32 bits signed integer. In this case, the Holding Registers has the value that the robot data is multiplied by $MULTIPLY and rounded off to no decimal place.
String	The continued Holding Registers express a string data. One Holding Register expresses 2 characters. Byte order of string data is specified by $MULTIPLY as follows. $MULTIPLY = 1 (0 or more): Lower byte is the first character, upper byte is the second character. $MULTIPLY = -1 (less than 0): Upper byte is the first character, lower byte is the second character. The last of the string data must be 0. When string data is written to Holding Registers, it is necessary to write 0 to the last of the string data.

> ⚠ **WARNING**
> If Integer data is written to the Holding Registers of Real data type, the assigned robot data may be set to the value that is too big or cannot be processed. And it may cause serious effect such as the robot moves to unexpected position because the assigned position register has abnormal position data. Please set $MULTIPLY to the value of not 0 normally, and assign 32 bit signed integer data to Holding Registers.

55.3.2 Assign Robot Registers

$SNPX_ASG setting to assign Robot Registers to Holding Registers is the following.

Variable in $SNPX_ASG	Description
$ADDRESS	Meaning: Start address of Holding Registers to assign. Range: 1~16384
$SIZE	Meaning: Number of Holding Registers to assign. One Register uses two Holding Registers. (You can change the number of Holding Registers for one robot register by adding "@" in $VAR_NAME) Range: 1~16384

Variable in $SNPX_ASG	Description
$VAR_NAME	Meaning: String to specify robot data. Please set ”R[1]” to assign Register R[1]. The number in [ ] is the index of register. Continued registers such as R[2]-R[5] can be assigned by one $SNPX_ASG element. In this case, please set $SIZE to 8 because the number of registers to assign is 4. And please set $VAR_NAME “R[2]”. The index 2 means the starting index of the continuous assignment. If “@1.1” is added just after the string, the number of Holding Registers for one register is changed from 2 to 1. In this case, accessing data size becomes 16 bits. Example: R[1]@1.1 “R[1]” part means to assign robot registers from index 1. “@1.1” part means one register is accessed as 16 bits data.
$MULTIPLY	Meaning: Multiplier The value accessed by an HMI device etc. is multiplied by $MULTIPLY. When $MULTIPLY is 0, it means that an HMI device etc. can access Holding Registers as 32bits REAL data. When it is not 0, an HMI device etc. accesses Holding Registers as 32bits signed integer. And value is rounded off to no decimal place. Range: 0.0001~10000, 0 Example: Register value is 123.45. When $MULTIPLY is 1, Holding Register is 123 When $MULTIPLY is 10, Holding Register is 1235. When $MULTIPLY is 0.1, Holding Register is 12. When $MULTIPLY is 0, Holding Register is 123.45 of REAL

For example, $SNPX_ASG is set as follows.

	$ADDRESS	$SIZE	$VAR_NAME	$MULTIPLY
$SNPX_ASG[1]	1	2	R[1]@1.1	1
$SNPX_ASG[2]	3	4	R[1]	100
$SNPX_ASG[3]	7	4	R[2]	0.1
$SNPX_ASG[4]	11	2	R[1]	0

In this case, address of Holding Registers are corresponded to robot registers as follows.

Address	Assigned data
1	R[1] as 16bits signed integer.
2	R[2] as 16bits signed integer.
3-4	R[1] multiplied by 100 as 32bits signed integer.
5-6	R[2] multiplied by 100 as 32bits signed integer.
7-8	R[2] divided by 10 as 32bits signed integer.
9-10	R[3] divided by 10 as 32bits signed integer.
11-12	R[1] as 32bits REAL

$SNPX_ASG[1] defines that 2 Holding Registers from address 1 to 2 are assigned to robot registers from R[1] multiplied by 1 as 16 bits signed integer. One Holding Register address is 16 bits data, and one register uses one Holding Register. So, address 1 is assigned to R[1] as 16 bits signed integer and address 2 is assigned to R[2] as 16bits signed integer.

$SNPX_ASG[2] defines that 4 Holding Registers from address 3 to 6 are assigned to robot registers from R[1] multiplied by 100 as 32 bits signed integer. One robot register uses 2 Holding Registers. So, addresses 3-4 are assigned to R[1] multiplied by 100 as 32 bits signed integer and address 5-6 are assigned to R[2] multiplied by 100 as 32 bits signed integer.

$SNPX_ASG[3] defines that 4 Holding Registers from 7 to 10 are assigned to robot registers from R[2] divided by 10 as 32 bits signed integer. One robot register uses 2 Holding Registers. So, addresses 7-8 are assigned to R[2] divided by 10 as 32 bits signed integer and address 9-10 are assigned to R[3] divided by 10 as 32 bits signed integer.

$SNPX_ASG[4] defines that 2 Holding Registers from 11 to 12 are assigned to robot register R[1] as 32bit REAL. So, addresses 11 to 12 are assigned to R[1] as 32bits REAL. OPC UA clients can read this bit array as Int16, so the OPC UA client should change this value to REAL number.

55.3.3 Assign Position Registers

$SNPX_ASG setting to assign Position Registers to Holding Registers is the following.

Variable in $SNPX_ASG	Description
$ADDRESS	Meaning: Start address of Holding Registers to assign. Range: 1~16384
$SIZE	Meaning: Number of Holding Registers to assign. One Position Register uses 50 Holding Registers. (You can change the number of Holding Registers for one position register by adding "@" in $VAR_NAME) Range: 1~16384
$VAR_NAME	Meaning: String to specify robot data. Please set "PR[1]" to assign Position Register PR[1]. The number in [ ] is the index of position register. Continued position registers like PR[2]-PR[5] can be assigned by one $SNPX_ASG element. In this case, please set $SIZE 200 because the number of position registers to assign is 4. And please set $VAR_NAME "PR[2]". The index 2 means the starting index of the continuous assignment. In multi group system, PR[1] means group 1 data of PR[1]. Please set PR[G2:1] to assign group 2 data of PR[1]. If "@" is added just after the string, the specified part of the position register is assigned. This is explained later.
$MULTIPLY	Meaning: Multiplier Only REAL values like X, Y, Z and J1 are effected 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 $MULTIPLY. When $MULTIPLY is 0, it means special that an HMI device etc. can access Holding Registers as 32bits REAL data. When it is not 0, an HMI device etc. accesses Holding Registers as 32bits signed integer. And value is rounded off to no decimal place. Range: 0.0001~10000, 0 Example: Position register member value is 123.45. When $MULTIPLY is 1, Holding Register is 123 When $MULTIPLY is 10, Holding Register is 1235. When $MULTIPLY is 0.1, Holding Register is 12. When $MULTIPLY is 0, Holding Register is 123.45 of REAL

⚠ WARNING
If Integer data is written to the Holding Registers of Real data type, the assigned robot data may be set to the value that is too big or cannot be processed. And it may cause serious effect such as the robot moves to unexpected position because the assigned position register has abnormal position data. Please set $MULTIPLY to the value of not 0 normally, and assign 32 bit signed integer data to Holding Registers.

NOTE
When the position register is locked by LOCK PREG instruction, the writing to the position register by this function is ignored. To confirm if the writing is succeeded or not, please read the position data just after the writing and confirm the read data.

One position register uses 50 Holding Registers. Contents of the 50 Holding Registers are the following.

Address	Description		Effect of $MULTIPLY
Cartesian data			
1-2	X	32bits signed integer or real (mm)	Yes
3-4	Y	32bits signed integer or real (mm)	Yes
5-6	Z	32bits signed integer or real (mm)	Yes
7-8	W	32bits signed integer or real (deg)	Yes
9-10	P	32bits signed integer or real (deg)	Yes
11-12	R	32bits signed integer or real (deg)	Yes
13-14	E1	32bits signed integer or real (mm, deg)	Yes
15-16	E2	32bits signed integer or real (mm, deg)	Yes
17-18	E3	32bits signed integer or real (mm, deg)	Yes
19	FLIP	16bits signed integer (1:Flip, 0:Non flip)	No
20	LEFT	16bits signed integer (1:Left, 0:Right)	No
21	UP	16bits signed integer (1:Up, 0:Down)	No
22	FRONT	16bits signed integer (1:Front, 0:Back)	No
23	TURN4	16bits signed integer (-128~127)	No
24	TURN5	16bits signed integer (-128~127)	No
25	TURN6	16bits signed integer (-128~127)	No
26	VALIDC	16bits signed integer (→NOTE1)	No
Joint data			
27-28	J1	32bits signed integer or real (mm, deg)	Yes
29-30	J2	32bits signed integer or real (mm, deg)	Yes
31-32	J3	32bits signed integer or real (mm, deg)	Yes
33-34	J4	32bits signed integer or real (mm, deg)	Yes
35-36	J5	32bits signed integer or real (mm, deg)	Yes
37-38	J6	32bits signed integer or real (mm, deg)	Yes
39-40	J7	32bits signed integer or real (mm, deg)	Yes
41-42	J8	32bits signed integer or real (mm, deg)	Yes
43-44	J9	32bits signed integer or real (mm, deg)	Yes
45	VALIDJ	16bits signed integer (→NOTE2)	No
Frame number			
46	UF	16bits signed integer (-1~62) (→NOTE3)	No
47	UT	16bits signed integer (-1~30) (→NOTE4)	No
48-50	Reserve		No

NOTE
1 VALIDC shows whether the position data is valid for Cartesian or not. This becomes 0 at the following situation, and it becomes 1 in the other situation.
   - Position data has uninitialized data (displayed as "******" on Teach Pendant).
   - Position data is Joint representation and it can not be converted to Cartesian.

   If HMI device etc. writes any data to VALIDC, position representation is changed to Cartesian. In this case, any value is OK to write.
2 VALIDJ shows whether the position data is valid for Joint or not. This becomes 0 at the following situation, and it becomes 1 in the other situation.
   - Position data has uninitialized data (displayed "******" on Teach Pendant).
   - Position data is Cartesian representation and it can not be converted to Joint.

   If HMI device etc. writes any data to VALIDJ, position representation is changed to Joint. In this case, any value is OK to write.
3 UF is user frame number.
   If UF is 0, world frame is used.
   If UF is -1, the current selected user fame is used.
   UF of Position Register is always -1. (In *i*Pendant it is displayed as F)
   This value cannot be changed.
4 UT is tool frame number
   If UT is 0, mechanical interface frame is used.
   If UT is -1, the current selected tool fame is used.
   UT of Position Register is always -1. (In *i*Pendant it is displayed as F)
   This value cannot be changed.

- Position register has 2 representations, Cartesian or Joint. If detail of the position register is displayed as X,Y,Z,W,P,R, it is Cartesian representation. If it is displayed as J1-6, it is Joint representation.
- An HMI device etc. can always access to any member without regarding to current representation. Position representation is changed by reading from HMI device etc. But if the representation of position register and the representation of accessed member from an HMI device etc. are different, please note the following.
- If the position data is out of stroke limit or there is uninitialized member, this position data cannot be converted to another representation. In this case, the all members in the part of another representation becomes 0.
  For example, a position register is Cartesian representation and X is 10000, it is out of stroke limit, J1-J9 is read as 0 by an HMI device etc. In this case, the members of Cartesian part or UF/UT part can be read correctly.
- If the representation of position register and the representation of accessed member from an HMI device etc. are different, communication response time is increased. Because position representation conversion takes much time. If an HMI device etc. reads many position registers, this conversion time may make big effect to response time.

Uninitialized member of position data is displayed as "******" on position register menu of Teach Pendant. These members are read as 0 from an HMI device etc. Please read VALIDC or VALIDJ to check whether the value is 0 or uninitialized. If position data has uninitialized member, VALIDJ and VALIDC are 0.

If you write data to the member in Cartesian part from an HMI device etc., the position representation becomes Cartesian. If you write data to member in Joint part from an HMI device etc., the position representation becomes Joint. If you write data to UF or UT from an HMI device etc., position representation is not changed.

Extract a part by adding "@" in $VAR_NAME

If "@" is added to $VAR_NAME, the specified part of data structure is extracted. It was already used for Register assignment (R[1]@1.1).

For example, you need to access only X, Y and Z of PR[1-3]. In this case, normally the setting of $SNPX_ASG is the following.

	$ADDRESS	$SIZE	$VAR_NAME	$MULTIPLY
$SNPX_ASG[1]	1	150	PR[1]	1

The following is the correspondence between position registers and Holding Registers of this setting.

Address	Assigned data
1-2	X of PR[1] as 32bits signed integer
3-4	Y of PR[1] as 32bits signed integer
5-6	Z of PR[1] as 32bits signed integer
51-52	X of PR[2] as 32bits signed integer
53-54	Y of PR[2] as 32bits signed integer
55-56	Z of PR[2] as 32bits signed integer
101-102	X of PR[3] as 32bits signed integer
103-104	Y of PR[3] as 32bits signed integer
105-106	Z of PR[3] as 32bits signed integer

Actual read data are only 18 Holding Registers, but this assignment occupies 150 Holding Registers. And, some HMI devices access address 7-50 of Holding Registers that is not necessary to read, because reading one big data is more efficient than reading several small data from the communication point of view. But the address 27-45 includes Joint representation part, and if position data is Cartesian representation, representation conversion is needed to read this unnecessary data.

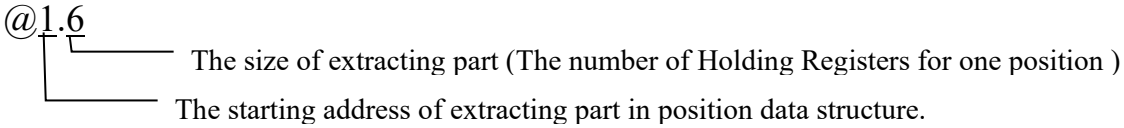
To communicate efficiently, please set $SNPX_ASG as follows.

	$ADDRESS	$SIZE	$VAR_NAME	$MULTIPLY
$SNPX_ASG[1]	1	18	PR[1]@1.6	1

The following is the correspondence between position registers and Holding Registers of this setting.

Address	Accessed data
1-2	X of PR[1] as 32bits signed integer
3-4	Y of PR[1] as 32bits signed integer
5-6	Z of PR[1] as 32bits signed integer
7-8	X of PR[2] as 32bits signed integer
9-10	Y of PR[2] as 32bits signed integer
11-12	Z of PR[2] as 32bits signed integer
13-14	X of PR[3] as 32bits signed integer
15-16	Y of PR[3] as 32bits signed integer
17-18	Z of PR[3] as 32bits signed integer

The change of $SNPX_ASG is that “@1.6” is added in $VAR_NAME and $SIZE is changed from 150 to 18. By this change, the number of Holding Registers for one position register is changed from 50 to 6. The “6” in ”@1.6” means the number of Holding Registers for one position register. And the “1” in “@1.6” means the starting address of extracting part in position data structure.
By specifying “@” in $VAR_NAME, you can extract the specified part of position data structure.



You can specify “@” not only for position register, but also for all data that is assigned by $SNPX_ASG.

The following is another example to assign J1-J6 of PR[1-3].

	$ADDRESS	$SIZE	$VAR_NAME	$MULTIPLY
$SNPX_ASG[1]	1	96	PR[3]@27.12	1

55.3.4 Assign String Registers

$SNPX_ASG setting to assign String Registers to Holding Registers is the following.

Variable in $SNPX_ASG	Description
$ADDRESS	Meaning: Start address of Holding Registers to assign. Range: 1~16384
$SIZE	Meaning: Number of Holding Registers to assign. One String Register uses 40 Holding Registers. It is possible to deal with first 80 characters of one string register in default setting. (You can change the number of Holding Registers for one string register by adding "@" in $VAR_NAME) Range: 1~16384
$VAR_NAME	Meaning: String to specify robot data. Please set "SR[1]" to assign String Register SR[1]. The number in [ ] is the index of string register. Continued string registers such as SR[2]-SR[5] can be assigned by one $SNPX_ASG element. In this case, please set $SIZE 160 because the number of string registers to assign is 4. And please set $VAR_NAME "SR[2]". The index 2 means the starting index of the continuous assignment. If "@" is added just after the string, the specified part of the String Register is assigned. Example: SR[1]@1.50 "SR[1]" means to assign String Registers from index 1. "@1.50" means the first 100 characters of String Register data are assigned.
$MULTIPLY	Meaning: Specify byte order of string. Range: 1,-1

55.3.5 Assign Current Position

$SNPX_ASG setting to assign Current Position to Holding Registers is the following.

Variable in $SNPX_ASG	Description
$ADDRESS	Meaning: Start address of Holding Registers to assign. Range: 1~16384
$SIZE	Meaning: Number of Holding Registers to assign. Current Position uses 50 Holding Registers. (You can change the number of Holding Registers for Current Position by adding "@" in $VAR_NAME) Range: 1~16384

Variable in $SNPX_ASG	Description
$VAR_NAME	Meaning: String to specify robot data. Please set ”POS[0]” to assign Current Position. The number in [ ] is user frame number. When user frame number is 0, the Current Position on World frame is assigned. This is the same as when "WORLD" is selected in the current position screen on the teaching pendant. When user frame number is -1, the Current Position on the User frame that is currently selected. This is the same as when "USER" is selected in the current position screen on the teaching pendant. When user frame number is 1-61, the Current Position on the specified user frame. Data structure is the same as position register. In multi group system, POS[0] means current position group 1 robot. Please set POS[G2:0] to assign current position of group 2 robot. If “@” is added just after the string, the specified part of current position is assigned.
$MULTIPLY	Meaning: Multiplier Only REAL values like X, Y, Z and J1 are effected 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 $MULTIPLY is read and written. If $MULTIPLY is 0, it means that the data is assigned to Holding Registers as 32bits REAL data. If it is not 0, it means that the data is assigned to Holding Registers as 32bits signed integer. And value is rounded off to no decimal place. Range: 0.0001~10000, 0 Example: Current position member value is 123.45. When $MULTIPLY is 1, Holding Register is 123 When $MULTIPLY is 10, Holding Register is 1235. When $MULTIPLY is 0.1, Holding Register is 12. When $MULTIPLY is 0, Holding Register is 123.45 of REAL

NOTE
Current position is read only. If you write it, nothing occurs.

Current position uses 50 Holding Registers, and data structure is the same as position register.
UT of current position is always -1.
UF of current position is the specified user frame number in $VAR_NAME.

Current position is not assigned continuously. For example, the following setting of $SNPX_ASG assigns current position of user frame 1 to address 1-50, but current position of user frame 2 is not assigned to address 51-100. Please set “POS[2]” in another $SNPX_ASG.

	$ADDRESS	$SIZE	$VAR_NAME	$MULTIPLY
$SNPX_ASG[1]	1	100	POS[1]	1

55.3.6 Assign Alarm History

$SNPX_ASG setting to assign Alarm history to Holding Registers is the following.

Variable in $SNPX_ASG	Description
$ADDRESS	Meaning: Start address of Holding Registers to assign. Range: 1~16384
$SIZE	Meaning: Number of Holding Registers to assign. One alarm uses 100 Holding Registers. (You can change the number of Holding Registers for one alarm by adding “@” in $VAR_NAME) Range: 1~16384
$VAR_NAME	Meaning: String to specify robot data. Please set ”ALM[1]” to assign alarm. The number in [ ] is line number in alarm menu. The latest alarm is 1. “ALM[1]” assigns active alarm. Alarms displayed in active alarm menu are assigned. “ALM[E1]” assigns alarm history. Alarms displayed in alarm history menu are assigned. “ALM[M1]” assigns motion alarm history. Alarm displayed in motion alarm menu are assigned. “ALM[S1]” assigns system alarm history. Alarm displayed in system alarm menu are assigned. “ALM[A1]” assigns application alarm history. Alarm displayed in application alarm menu are assigned. “ALM[P1]” assigns password log. Password log displayed in password log menu are assigned.
$MULTIPLY	Meaning: Specify byte order of string. Range: 1,-1

NOTE
Alarm history is read only. If you write it, nothing occurs.

One alarm uses 100 Holding Registers. Contents of the 100 Holding Registers are the following.

Address	Description
1	Facility code 16 bits signed integer When alarm is “SRVO-001”, facility code is 11 that means “SRVO”. Please refer the "2.2.2 Facility Name and Code" in "OPERATOR'S MANUAL (Alarm Code List)" (B-83284EN-1) about the number of facility code.
2	Alarm number 16 bits signed integer When alarm is “SRVO-001”, alarm number is 1.
3	Facility code of Cause Code 16bits signed integer When alarm occurs, alarm message is displayed on top of teach pendant. Sometimes a message is also displayed on the second line, it is cause code. This shows facility code of the cause code. If the alarm does not have cause code, this becomes 0. For example, when cause code is “SRVO-001”, facility code is 11 that means “SRVO”. Please refer the "2.2.2 Facility Name and Code" in "OPERATOR'S MANUAL (Alarm Code List)" (B-83284EN-1) about the number of facility code.
4	Alarm number of cause code 16 bits signed integer Alarm number of cause code. If the alarm does not have cause code, this becomes 0. When cause code is “SRVO-001”, alarm number of cause code is 1.

Address	Description
5	Alarm severity 16 bits signed integer The value shows alarm severity. NONE 128 WARN 0 PAUSE.L 2 PAUSE.G 34 STOP.L 6 STOP.G 38 SERVO 54 ABORT.L 11 ABORT.G 43 SERVO2 58 SYSTEM 123
6	Occurred Time (year) 16 bits signed integer
7	Occurred Time (month) 16 bits signed integer
8	Occurred Time (day) 16 bits signed integer
9	Occurred Time (hour 24 hour system) 16 bits signed integer
10	Occurred Time (minutes) 16 bits signed integer
11	Occurred Time (second) 16 bits signed integer
12-51	Alarm message string of up to 80 characters Alarm message string can be read. It shows the string as the same as that is displayed on top line of teach pendant such as “SRVO-001”.
52-91	Cause code alarm message string of up to 80 characters Alarm message of cause code.
92-100	Alarm severity word 80 characters string String of alarm severity such as “WARN”.

Facility code and alarm number of “RESET” is read as 0. Alarm message is “RESET”.

When there is only 2 lines on active alarm menu, all members of alarms after ALM[3] are 0.

In string item, addresses after string are 0.
One alarm uses 100 Holding Registers, if you would like to read only facility code and alarm number, most of 100 Holding Registers area is not used. To communicate efficiently, please use “@”.

Example: $SNPX_ASG is set as follows.

	$ADDRESS	$SIZE	$VAR_NAME	$MULTIPLY
$SNPX_ASG[1]	1	12	ALM[E1]@1.4	1

The correspondence of address and alarm data of this setting is the following.

Address	Assigned data
1	Alarm ID of alarm1
2	Alarm number of alarm1
3	Alarm ID of cause code of alarm1
4	Alarm number of cause code of alarm1
5	Alarm ID of alarm2
6	Alarm number of alarm2
7	Alarm ID of cause code of alarm2
8	Alarm number of cause code of alarm2
9	Alarm ID of alarm3
10	Alarm number of alarm3
11	Alarm ID of cause code of alarm3
12	Alarm number of cause code of alarm3

55.3.7 Assign Program Execution Status

$SNPX_ASG setting to assign program execution status to Holding Registers is the following.

Variable in $SNPX_ASG	Description
$ADDRESS	Meaning: Start address of Holding Registers to assign. Range: 1~16384
$SIZE	Meaning: Number of Holding Registers to assign. One task uses 18 Holding Registers. (You can change the number of Holding Registers for one task by adding "@" in $VAR_NAME) Range: 1~16384
$VAR_NAME	Meaning: String to specify robot data. Please set "PRG[1]" to assign program execution status. The number in [ ] is task number. In single task system, program execution status always can be assigned by PRG[1]. In multi task system, when 2 tasks are running at the same time, PRG[1] and PRG[2] shows the execution status of each task. Which task is PRG[1] is decided by timing of execution and communication. But the task that is read as PRG[1] is always read as PRG[1] until it is aborted.
$MULTIPLY	Meaning: Specify byte order of string. Range: 1,-1

NOTE
Program execution status is read only. If you write it, nothing occurs.

One task uses 18 Holding Registers. Contents of the 18 Holding Registers are the following.

Address	Description
1-8	Program name 16 characters string Name of running program. When sub program is called, it shows sub program name.
9	Line number 16bits signed integer. Execution line number. When sub program is called, it shows line number of sub program.
10	Execution status 16bits signed integer Aborted 0 Paused 1 Running 2
11-18	Parent program name 16 characters string Name of the started program When sub program is not called, it is the same as program name.

When program is aborted, all members become 0.

By assigning each strings to $VAR_NAME, the type of the program can be selected. And the meaning of line number is changed depending on the situation.

$VAR_NAME	Program name and line number
PRG[1]	Program name Name of running program. Line number Execution line number.

$VAR_NAME	Program name and line number
PRG[M1]	Program name When running program is MACRO program, it shows name of parent program that calls the running program. When the parent program is also MACRO program, it shows name of parent program that is not MACRO program. Line number Line number of CALL instruction that calls the running program. When all parent programs are MACRO program. It becomes 0.
PRG[K1]	Program name When running program is KAREL program, it shows name of parent program that calls the running program. When the parent program is also KAREL program, it shows name of parent program that is not KAREL program. Line number Line number of CALL instruction that calls the running program. When all parent programs are KAREL program. It becomes 0.
PRG[MK1] or PRG[KM1]	Program name When running program is MACRO or KAREL program, it shows name of parent program that calls the running program. When the parent program is also MACRO or KAREL program, it shows name of parent program that is not MACRO nor KAREL program. Line number Line number of CALL instruction that calls the running program. When all parent programs are MACRO or KAREL program. It becomes 0.

By adding "@", a part of this structure is assigned to Holding Registers.
Example: $SNPX_ASG is set as follows.

	$ADDRESS	$SIZE	$VAR_NAME	$MULTIPLY
$SNPX_ASG[1]	1	4	PRG[1]@9.2	1

The correspondence of address and program execution status of this setting is the following.

Address	Accessed data
1	Line number of task 1
2	Execution status of task 1
3	Line number of task 2.
4	Execution status of task 2.

55.3.8 Assign System Variables

$SNPX_ASG setting to assign system variables to Holding Registers is the following.

Variable in $SNPX_ASG	Description
$ADDRESS	Meaning: Start address of Holding Registers to assign. Range: 1~16384
$SIZE	Meaning: Number of Holding Registers to assign. The number of Holding Registers for one system variable is defined by data type of the system variable. Please refer the following data type list. (You can change the number of Holding Registers for one system variable by adding "@" in $VAR_NAME) Range: 1~16384

Variable in $SNPX_ASG	Description
$VAR_NAME	Meaning: String to specify robot data. Please set system variable name, for example “$SNPX_ASG[1].$ADDRESS”. If array system variable such as “$SNPX_ASG[1]” is set, array elements are assigned continuously like Registers. KAREL string can be specified by the following format. $[KAREL program name]variable name.
$MULTIPLY	Meaning: The meaning of $MULTIPLY is defined according to the data type of system variable. Please refer the following data type list.

To access to system variable, data type of assigned Holding Registers and meaning of $MULTIPLY are changed according to data type of system variable. The following is the list of data type of system variables that can be assigned to Holding Registers, and it also explain the number of Holding Registers and meaning of $MULTIPLY.

Data type of system variables	The number of Holding Registers for one system variable	Meaning of $MULTIPLY
INTEGER 32bits signed integer	2	The data is multiplied by $MULTIPLY. The data is accessed as 32 bits signed integer. And value is rounded off to no decimal place. If $MULTIPLY is 0, it is same as $MULTIPLY is 1.
SHORT 16bits signed integer	2	
BYTE 8bits signed integer	2	
REAL 32bits real	2	The data is multiplied by $MULTIPLY. If $MULTIPLY is 0, it means that the data is accessed as 32 bits REAL data. If it is not 0, the data is accessed as 32 bits signed integer. And value is rounded off to no decimal place
BOOLEAN TRUE/FALSE	2	$MULTIPLY is not used for this data type. The data is accessed as 32bit signed integer. If value is TRUE, Holding Register is 1. If it is 0, Holding Register is 0. If 0 is written, the system variable becomes FALSE. If not 0 is written, the system variable becomes TRUE.
POSITION Position data	50	Data structure is the same as position register. Meaning of $MULTIPLY is also same as position register.
STRING String data	40	Specify byte order of string.

INTEGER, SHORT and BYTE are accessed by the same way. The system variable that integer value is displayed in system variable menu on teach pendant is INTEGER, SHORT or BYTE.
The variable that real value is displayed in system variable menu on teach pendant is REAL.
The variable that TRUE or FALSE is displayed in system variable menu on teach pendant is BOOLEAN.
The variable that string is displayed on system variable menu on teach pendant is STRING.
The variable that is displayed as POSITION in system variable menu on teach pendant is POSITION.

> ⚠ **WARNING**
> The system variable that is protected in system variable menu on teach pendant can be changed from Holding Registers. If illegal value is set to system variable, system may be harmed seriously. Please check value enough to write system variables.

> ⚠ **WARNING**
> If Integer data is written to the Holding Registers of Real data type, the assigned robot data may be set to the value that is too big or cannot be processed. And it may cause serious effect such as the robot moves to unexpected position because the assigned POSITION type variable has abnormal position data. Please set $MULTIPLY to the value of not 0 normally, and assign 32 bit signed integer data to Holding Registers.

By adding "@", a part of this structure is assigned to Holding Registers.
For example. the following $SNPX_ASG assigns X, Y and Z of user frame 1 and 2 of group 1.

	$ADDRESS	$SIZE	$VAR_NAME	$MULTIPLY
$SNPX_ASG[1]	1	12	$MNUFRAME[1,1]@1.6	1

This assigns system variables to Holding Registers as follows.

Address	Accessed data
1-2	X of user frame 1 of Group 1 (X of $MNUFRAME[1,1])
3-4	Y of user frame 1 of Group 1 (Y of $MNUFRAME[1,1])
5-6	Z of user frame 1 of Group 1 (Z of $MNUFRAME[1,1])
7-8	X of user frame 2 of Group 1 (X of $MNUFRAME[1,2])
9-10	Y of user frame 2 of Group 1 (Y of $MNUFRAME[1,2])
11-12	Z of user frame 2 of Group 1 (Z of $MNUFRAME[1,2])

Special system variables

The following table shows the list of the special system variables that are not actual system variables but it can be assigned to Holding Registers to read the robot internal information. The special system variables are read only. The value is not changed, when the assigned Holding Registers is written. The special system variables of $RBTIF.MEM_~ are supported from 7DC1 series P06 or later. The other special system variables are supported from 7DC3 series P43 or later and 7DF1 series P15 or later and all versions after 7DF1 series.

Variable name	Data Type	Description
$RBTIF.MEM_TPP_TOT	INTEGER	Meaning: Total storage capacity of the TPP. The unit displayed in the holding register is byte.
$RBTIF.MEM_TPP_FREE	INTEGER	Meaning: Available storage capacity of the TPP. The unit displayed in the holding register is byte.
$RBTIF.MEM_TEMP_TOT	INTEGER	Meaning: Total storage capacity of the TEMP. The unit displayed in the holding register is byte.
$RBTIF.MEM_TEMP_FREE	INTEGER	Meaning: Available storage capacity of the TEMP. The unit displayed in the holding register is byte.
$RBTIF.MEM_PERM_TOT	INTEGER	Meaning: Total storage capacity of the PERM. The unit displayed in the holding register is byte.
$RBTIF.MEM_PERM_FREE	INTEGER	Meaning: Available storage capacity of the PERM. The unit displayed in the holding register is byte.
$RBTIF.MEM_FROM_TOT	INTEGER	Meaning: Total storage capacity of the F-ROM module. The unit displayed in the holding register is byte.
$RBTIF.MEM_FROM_FREE	INTEGER	Meaning: Available storage capacity of the F-ROM module. The unit displayed in the holding register is byte.
$RBTIF.SERIAL_NUM	STRING	Meaning: Robot serial number.
$RBTIF.SOFT_VERSION	STRING	Meaning: Robot software version.
$RBTIF.ROBOT_ID_GRP1 ~ $RBTIF.ROBOT_ID_GRP8	STRING	Meaning: Robot ID. The last number corresponds to the number of groups. If invalid last number is set, value 0 is set.
$RBTIF.OVERRIDE	INTEGER	Meaning: Override. Range is 1 to 100%.

Variable name	Data Type	Description
$RBTIF.STIME_PWR_TOT1 ~ $RBTIF.STIME_PWR_TOT8	INTEGER	Meaning: Total time of controller power on. The last number corresponds to the number of groups. If invalid last number is set, value 0 is set. The unit displayed in the holding register is minute.
$RBTIF.STIME_SRV_TOT1 ~ $RBTIF.STIME_SRV_TOT8	INTEGER	Meaning: Total time of servo on. The last number corresponds to the number of groups. If invalid last number is set, value 0 is set. The unit displayed in the holding register is minute.
$RBTIF.STIME_RUN_TOT1 ~ $RBTIF.STIME_RUN_TOT8	INTEGER	Meaning: Total time of program running. The last number corresponds to the number of groups. If invalid last number is set, value 0 is set. The unit displayed in the holding register is minute.
$RBTIF.STIME_WIT_TOT1 ~ $RBTIF.STIME_WIT_TOT8	INTEGER	Meaning: Total time of program waiting. The last number corresponds to the number of groups. If invalid last number is set, value 0 is set. The unit displayed in the holding register is minute.
$RBTIF.STIME_SHM_TOT1 ~ $RBTIF.STIME_SHM_TOT8	INTEGER	Meaning: Total time of brake release. The last number corresponds to the number of groups. If invalid last number is set, value 0 is set. The unit displayed in the holding register is minute.
$RBTIF.STIME_PWR_LAP1 ~ $RBTIF.STIME_PWR_LAP8	INTEGER	Meaning: Lap time of controller power on. The last number corresponds to the number of groups. If invalid last number is set, value 0 is set. The unit displayed in the holding register is second.
$RBTIF.STIME_SRV_LAP1 ~ $RBTIF.STIME_SRV_LAP8	INTEGER	Meaning: Lap time of servo on. The last number corresponds to the number of groups. If invalid last number is set, value 0 is set. The unit displayed in the holding register is second.
$RBTIF.STIME_RUN_LAP1 ~ $RBTIF.STIME_RUN_LAP8	INTEGER	Meaning: Lap time of program running. The last number corresponds to the number of groups. If invalid last number is set, value 0 is set. The unit displayed in the holding register is second.
$RBTIF.STIME_WIT_LAP1 ~ $RBTIF.STIME_WIT_LAP8	INTEGER	Meaning: Lap time of program waiting. The last number corresponds to the number of groups. If invalid last number is set, value 0 is set. The unit displayed in the holding register is second.
$RBTIF.STIME_SHM_LAP1 ~ $RBTIF.STIME_SHM_LAP8	INTEGER	Meaning: Lap time of brake release. The last number corresponds to the number of groups. If invalid last number is set, value 0 is set. The unit displayed in the holding register is second.
$RBTIF.Q_CURRENT11 ~ $RBTIF.Q_CURRENT19, $RBTIF.Q_CURRENT21 ~ $RBTIF.Q_CURRENT29, … $RBTIF.Q_CURRENT81 ~ $RBTIF.Q_CURRENT89	REAL	Meaning: 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 invalid last number is set, value 0 is set. The unit displayed in the holding register is ampere. Example: $SNPX_ASG.$VAR_NAME : $RBTIF.Q_CURRENT23 Group number is 2, Axis number is 3.

Variable name	Data Type	Description
$RBTIF.CURRENT11 ~ $RBTIF.CURRENT19, $RBTIF.CURRENT21 ~ $RBTIF.CURRENT29, … $RBTIF.CURRENT81 ~ $RBTIF.CURRENT89	REAL	Meaning: Current of each axis. The last number corresponds to the number of groups and axis number. If invalid last number is set, value 0 is set. The unit displayed in the holding register is ampere. Example: $SNPX_ASG.$VAR_NAME : $RBTIF.CURRENT23 Group number is 2, Axis number is 3.

55.3.9 Assign comment

$SNPX_ASG setting to assign comment of Register, Position register, String register and I/O to Holding Registers is the following.

Variable in $SNPX_ASG	Description
$ADDRESS	Meaning: Start address of Holding Registers to assign. Range: 1 to 16384
$SIZE	Meaning: Number of Holding Registers to assign. One comment uses 40 Holding Registers. Please set according to the number of comments to assign. (You can change the number of Holding Registers for one comment by adding “@” in $VAR_NAME) Range: 1 to 16384

Variable in $SNPX_ASG	Description
$VAR_NAME	Meaning: String to specify robot data. Please set ”R[C1]” to assign comment of register. The number in [ ] is index of register. To assign comment of position register, string register and I/O, please set the following string. Register R[C1] Position register PR[C1] String register SR[C1] DI DI[C1] DO DO[C1] RI RI[C1] RO RO[C1] UI UI[C1] UO UO[C1] SI SI[C1] SO SO[C1] WI WI[C1] WO WO[C1] WSI WSI[C1] WSO WSO[C1] GI GI[C1] GO GO[C1] AI AI[C1] AO AO[C1] Comment of the continued registers such as R[2]~R[5] can be assigned by one $SNPX_ASG element. In this case, please set $SIZE 160 because the number for one comment to assign is 40. And please set $VAR_NAME “R[C2]”. The index 2 means the starting index of the continuous assignment. If “@” is added just after the string, the specified part of the comment is assigned. Example: R[C1]@1.5 “R[C1]” means to assign comment of registers from index 1. “@1.5” means the first 10 characters of comment are assigned.
$MULTIPLY	Meaning: Specify byte order of string. Range: 1,-1

55.3.10 Assign I/O data and simulation status

$SNPX_ASG setting to assign I/O data and simulation status to Holding Registers is the following.
Input signal can't be changed from this function even though it is assigned to Holding Registers.
If you want to change input signal from this function, there is a way to change input signal through flag with assigning the input signal to flag, which is rack 34 and slot 1.

Variable in $SNPX_ASG	Description
$ADDRESS	Meaning: Start address of Holding Registers to assign. Range: 1~16384
$SIZE	Meaning: Number of Holding Registers to assign. The number of Holding Registers for one I/O value or simulation status is changed by $MULTIPLY setting. When $MULTIPLY is 1, one I/O value or simulation status uses one Holding Register. When $MULTIPLY is 0, one Holding Register includes 16 I/O value or simulation status as bit image. (Note: Value of GI/O and AI/O use one Holding Register even if $MULTIPLY is 0.) Range: 1~16384
$VAR_NAME	Meaning: String to specify robot data. Please set "DI[1]" to assign DI[1] data. The number in [ ] is index of DI. To assign I/O data or simulation status of the other type of I/O, please set the following string. DI: DI[1] DI[S1] DO: DO[1] DO[S1] RI: RI[1] RI[S1] RO: RO[1] RO[S1] UI: UI[1] UO: UO[1] SI: SI[1] SO: SO[1] WI: WI[1] WI[S1] WO: WO[1] WO[S1] WSI: WSI[1] WSI[S1] WSO: WSO[1] WSO[S1] GI: GI[1] GI[S1] GO: GO[1] GO[S1] AI: AI[1] AI[S1] AO: AO[1] AO[S1] Flag: F[1] Marker: M[1] Example: Assign DI[11]~DI[42]: When $MULTIPLY is 1, please set 32 to $SIZE because the number of DI is 32, and set "DI[11]" to $VAR_NAME. In this case, index 11 means that the start index of the assignment is 11. When $MULTIPLY is 0, 16 DI data are assigned to one Holding Register, so please set 2 to $SIZE, and set "DI[11]" to $VAR_NAME.
$MULTIPLY	Meaning: One Holding Register has one I/O data or 16 I/O data as bit image. One I/O data is assigned to one Holding Register 1 16 I/O data are assigned to one Holding Register 0

You can select by $MULTIPLY setting whether one I/O data is assigned to one Holding Register or 16 I/O data are assigned to one Holding Register.

When $MULTIPLY is 1, one I/O data or one simulation status are assigned to one Holding Register. When the value is ON, the corresponded Holding Register is set to 1. When the value is OFF, the corresponded Holding Register is set to 0.

When $MULTIPLY is 0, 16 I/O data or 16 simulation status are assigned to one Holding Register. When the value is ON, the corresponded bit of Holding Register is set to 1. When the value is OFF, the corresponded bit of Holding Register is set to 0. I/O of lower index is assigned to lower bit of Holding Register.
Example: When DI[1-16] is assigned to Holding Register, if only DI[1] is ON and the others are OFF, the Holding Register is 1. If only DI[16] is ON and the others are OFF, the Holding Register is 32768 (in case of unsigned 16 bit integer).

55.3.11 Assign Integrated PMC address data

$SNPX_ASG setting to Integrated PMC address data to Holding Registers is the following.

Variable in $SNPX_ASG	Description
$ADDRESS	Meaning: Start address of Holding Registers to assign. Range: 1~16384
$SIZE	Meaning: Number of Holding Registers to assign. The number of Holding Registers corresponded to PMC address is different by the specified PMC address that is byte address or bit address. When bit address is specified, one bit PMC address is assigned to one Holding Register. When byte address is specified, two bytes PMC addresses are assigned to one Holding Register. Range: 1~16384
$VAR_NAME	Meaning: String to specify robot data. Set ”PMC1:X0” to assign data of PMC address X0. The "PMC1" part is PMC path number. "PMCS" means DCS Safety PMC. Data of bit address is assigned by specifying bit address such as ”PMC1:X0.0”. Example: Assign X0.0~X1.7 data of 1st path PMC When $SIZE is set to 1 and $VAR_NAME is set to ”PMC1:X0”, two byte address X0~X1 are assigned to one Holding Register address. Lower byte is assigned to X0, upper byte is assigned to X1. When $SIZE is set to 16 and $VAR_NAME is set to ”PMC1:X0.0”, 16 bit address X0.0~X1.7 are assigned to 16 Holding Registers. X0.0 is assigned to Address 1, X0.1 is assigned to Address 2, X1.7 is assigned to Address 16.
$MULTIPLY	Not used

NOTE
1 When K900.4 (Change address value) is 0, writing to PMC address data is ignored. To check if the writing is ignored or not, please read the PMC address after writing and confirm the read value.
2 Address of DCS safety PMC is read only. If you write it, nothing occurs.

Example: $SNPX_ASG is set as follows.

	$ADDRESS	$SIZE	$VAR_NAME	$MULTIPLY
$SNPX_ASG[1]	1	2	PMC1:R0.0	1
$SNPX_ASG[2]	3	2	PMC1:D0	1

The correspondence between Holding Registers and robot data by this setting is the following.
Bit address is assigned to the address 1 - 2, therefore 1 bit of PMC address is assigned to 1 Holding Register continuously.

Byte address is assigned to the address 3 - 4, therefore 2 bytes of PMC address is assigned to 1 Holding Register continuously.

Address	Assigned robot data
1	R0.0
2	R0.1
3	D0 - D1 (Lower byte:D0, Upper byte:D1)
4	D2 - D3 (Lower byte:D2, Upper byte:D3)

55.3.12 Assign Symbol and Comment of Integrated PMC address

$SNPX_ASG setting to symbol and comment of Integrated PMC address to Holding Registers is the following.

Variable in $SNPX_ASG	Description
$ADDRESS	Meaning: Start address of Holding Registers to assign. Range: 1~16384
$SIZE	Meaning: Number of Holding Registers to assign. One symbol or comment uses 40 Holding Registers. Please set according to the number of comments to assign. (You can change the number of Holding Registers for one symbol and comment by adding “@” in $VAR_NAME.) Range: 1~16384
$VAR_NAME	Meaning: String to specify robot data. Set ”PMC1:S:X0” to assign symbol of PMC address X0. Set ”PMC1:C:X0” to assign comment of the PMC address. The "PMC1" part is PMC path number. "PMCS" means DCS Safety PMC. Symbol and comment of bit address is assigned by specifying bit address such as ”PMC1:S:X0.0”. Example: PMC1:S:X0.0@1.5 The “PMC1:S:X0.0” part means that symbol of bit address from X0.0 is assigned continuously. The @1.5 part means that the top 10 characters are assigned.
$MULTIPLY	Meaning: Specify byte order of string. Range: 1,-1

NOTE
Symbol and comment of Integrated PMC are read only. If you write it, nothing occurs.

Example: $SNPX_ASG is set as follows.

	$ADDRESS	$SIZE	$VAR_NAME	$MULTIPLY
$SNPX_ASG[1]	1	10	PMC1:S:R0.0@1.5	1
$SNPX_ASG[2]	11	10	PMC1:C:D0@1.5	1

The correspondence between Holding Registers and robot data by this setting is the following.

Address	Assigned robot data
1-5	Symbol of R0.0
6-10	Symbol of R0.1
11-15	Comment of D0
16-20	Comment of D1

55.3.13 Hints

Assigned data is not read correctly

If an HMI device etc. reads a Holding Register that is not assigned to any data, this Holding Register is always read as 0. If Holding Register that the problem occurs is not 0, this Holding Register is assigned to the other data.
For example, when $SNPX_ASG is set as follows, addresses 101-150 of Holding Registers are assigned by both $SNPX_ASG[1] and $SNPX_ASG[2].

	$ADDRESS	$SIZE	$VAR_NAME	$MULTIPLY
$SNPX_ASG[1]	1	1000	R[1]@1.1	1
$SNPX_ASG[2]	101	50	PR[1]	100

In this case, the $SNPX_ASG whose index is smaller is used.
Therefore, addresses 101-150 are assigned to R[101]-R[150], and PR[1] can not be accessed by an HMI device etc.
If the read data of Holding Register is 0, there may also be duplicated assignment as above. Please check duplication of $SNPX_ASG.
If $SNPX_ASG setting has problem even though assignment is not duplicated, please check $VAR_NAME of $SNPX_ASG.
The following list shows the correct format of $VAR_NAME setting. If the setting is not match to them, robot data is not assigned.

Correct format	Note
“R[n]”	
“PR[n]” “PR[Gn:n]	If you specify group number, “:” must be specified between group number and index.
“SR[n]”	
“POS[n]” “POS[Gn:n]	If you specify group number, “:” must be specified between group number and index.
“ALM[n]” “ALM[En]” “ALM[Pn]	Alarm line number must be specified just after E, M, A, S or P. The ":" must not be specified.
“PRG[n]” “PRG[Mn]”, “PRG[Kn]” “PRG[MKn]”, “PRG[KMn]”	Index must be specified just after M, K, MK or KM. The ":" must not be specified.
System variable name	The first character of system variable must be “$”.
$[KAREL program name]variable name	Specify KAREL program name just after ‘$[‘, then specify ‘]’ and specify KAREL variable name just after ‘]’.
“DI[n]”, “DI[Sn]”, “DI[Cn]” “DO[n]”, “DO[Sn]”, “DO[Cn]” “RI[n]”, “RI[Sn]”, “RI[Cn]”, “RO[n]”, “RO[Sn]”, “RO[Cn]” “UI[n]”, “UI[Cn]” “UO[n]”, “UO[Cn]” “SI[n]”, “SI[Cn]” “SO[n]”, “SO[Cn]” “WI[n]”, “WI[Sn]”, “WI[Cn]”, “WO[n]”, “WO[Sn]”, “WO[Cn]” “WSI[n]”, “WSI[Sn]”, “WSI[Cn]” “WSO[n]”, “WSO[Sn]”, “WSO[Cn]” “GI[n]”, “GI[Sn]”, “GI[Cn]” “GO[n]”, “GO[Sn]”, “GO[Cn]” “AI[n]”, “AI[Sn]”, “AI[Cn]” “AO[n]”, “AO[Sn]”, “AO[Cn]” “F[n]”, “F[Cn]” “M[n]”, “M[Cn]”	Index must be specified just after S or C. The ":" must not be specified.

Correct format	Note
PMC<path>:address PMC<path>:address.bit PMC<path>:S:address PMC<path>:S:address.bit PMC<path>:C:address PMC<path>:C:address.bit	The <path> is from 1 to 5 or "S". The ":" must be specified just after PMC path. And PMC address must be specified just after the ":". For symbol or comment, ":S" or ":C" must be specified just after PMC path.

- If there is space character in $VAR_NAME, data is not assigned. If you specify "@", please do not add any space before "@".
- If you use continuous array assignment, the number of Holding Registers for one element is defined according to data type, please check explanation of the assigned robot data.
- The format of "@" is "@n.n". A "." must be specified between starting address and the number of Holding Registers. A "@" must be specified just after variable name.

Method to improve response time

- Position representation conversion takes much time. If you access position register, please assign only necessary members of position structure by using "@". And please change position representation of the position registers to the representation that is accessed by an HMI device etc.

55.4 DYNAMIC ASSIGNMENT TO HOLDING REGISTERS

Setting $SNPX_ASG is necessary to read or write except for I/O signal. This setting is done by the system variable screen normally. However, dynamic settings, which is equivalent to $SNPX_ASG, is available for each connection after communication connection from Modbus master or from OPC UA client.
Please 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 with keeping connection or set every connection. Many masters / clients disconnect after every writing, if dynamic assignment doesn't work correctly, you have to understand and control master / client operation for the level of Modbus or TCP/IP protocol.

This setting takes precedence over $SNPX_ASG setting.
This function is supported from 7DF1 series P19 or after and all versions after 7DF1 series.

When 128 characters or less are written to the address 16385 or after of Holding Registers (It is necessary to start writing from address 16385) with Modbus from master or Command node with OPC UA from client, robot regards the written characters as command and processes it. Byte order of string data when Modbus can be changed with $SNPX_PARAM.$CMD_ENDIAN.
Other than 0 (default) : Lower byte is the first character, upper byte is the second character.
0 : Upper byte is the first character, lower byte is the second character.

The following commands can be executed.
Each command should be written in 1 time. Write some times with segment is invalid.

CLRASG Clear all assignment and initialization
Format: CLRASG
Function: Clear all assignment and initialization.

Do once before following SETASG command sequence.

SETASG Set assignment
Format: SETASG ($ADDRESS) ($SIZE) ($VAR_NAME) [($MULTIPLY)]
Function: Set assignment.

The specification of $ADDRESS, $SIZE, $VAR_NAME and $MULTIPLY are same with the specification of the same variables under $SNPX_ASG.
$MULTIPLY can be abbreviated. If abbreviated, $MULTIPLY is set to 1.
Do CLRASG before executing SETASG sequence.

Example: If assignments are set as following

No.	$ADDRESS	$SIZE	$VAR_NAME	$MULTIPLY
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 following strings in order.

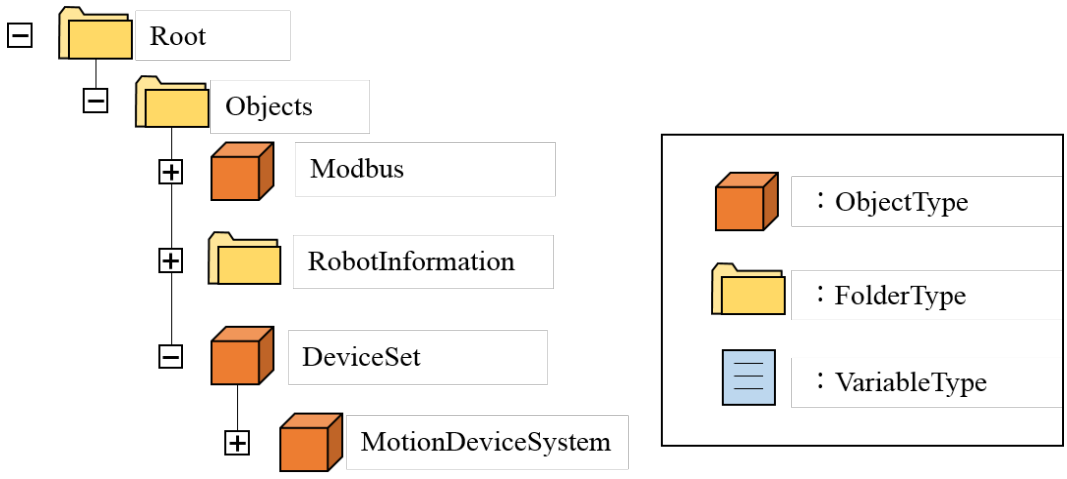
```
CLRASG
SETASG 1 2 R[1]@1.1 1
SETASG 3 4 R[1] 100
SETASG 7 4 R[2] 0.1
SETASG 11 2 R[1] 0
```

If some commands are separated with CR (13) or LF (10), some commands can be written in 1 sequence but the number of max character in 1 sequence is 128.

55.5 OPC UA DATA MODEL

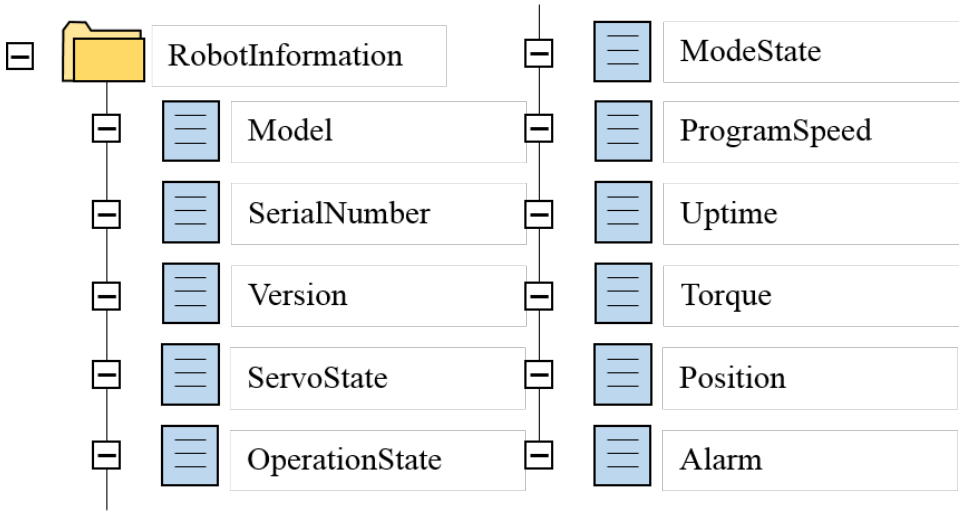
The OPC UA server for FANUC robots has Modbus node that provides read/write of IO and registers using the Modbus data model, RobotInformation node that provides basic robot information, and MotionDeviceSystem node of the OPC UA for Robotics specification.

- RobotInformation node and MotionDeviceSystem node are supported from 7DF1 series P28 or after, 7DF3 series P06 or after, and all versions after 7DF3 series.



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

The Model node provides the model name for the group 1 robot.
e.g. R-2000iC/165F

SerialNumber

The SerialNumber node provides the robot controller serial number.

Version

The Version node provides the software version of the robot controller.
e.g. 7DF1/19

ServoState

The ServoState node provides the status of servo on / off.
The node value is set to 1 when the servo is on and 0 when the servo is off.

OperationState

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, and the value converted to decimal is represented by one node value.
E-STOP is set to 1 when any one 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 ~ BUSY is the same as UO of peripheral I / O.
The OperationState node is set in the following order from the lower bits.
E-STOP (1) | Abort (2) | PROGRUN (4) | PAUSED (8) | HELD (16) | FAULT (32) | TPENBL (64) | BATALM (128) | BUSY (256)

ModeState

The ModeState node provides the operation mode of the robot.
The current operation mode is set to the node value.
Operation modes are Auto, T1 (Manual), and T2 (Manual).

ProgramSpeed

The ProgramSpeed node provides the speed override of robot.
The value from 0 to 100 (%) is set to the node value.

Uptime

The Uptime node provides the total energization time of the robot controller.
The value of the total energization time (min) is set to the node value.

Torque

The Torque node provides the Q-phase current value of each axis for the robot (until 6 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

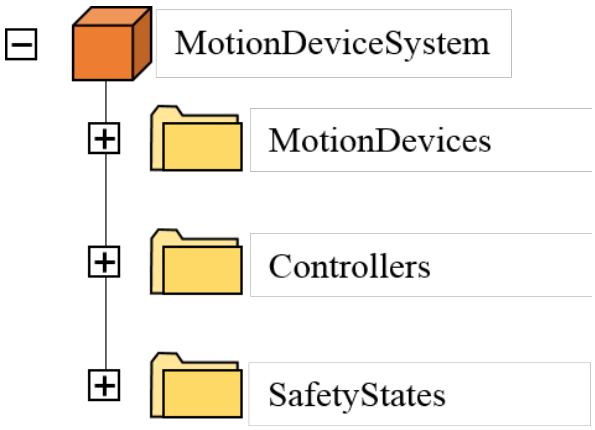
The Position node provides the position of each axis for the robot (until 6 axis) of group 1 and independent axis for group 1.

Alarm

The Alarm node provides the current alarm.
The current alarm string is set to the node value.
e.g. SRVO-001 Operator panel E-stop

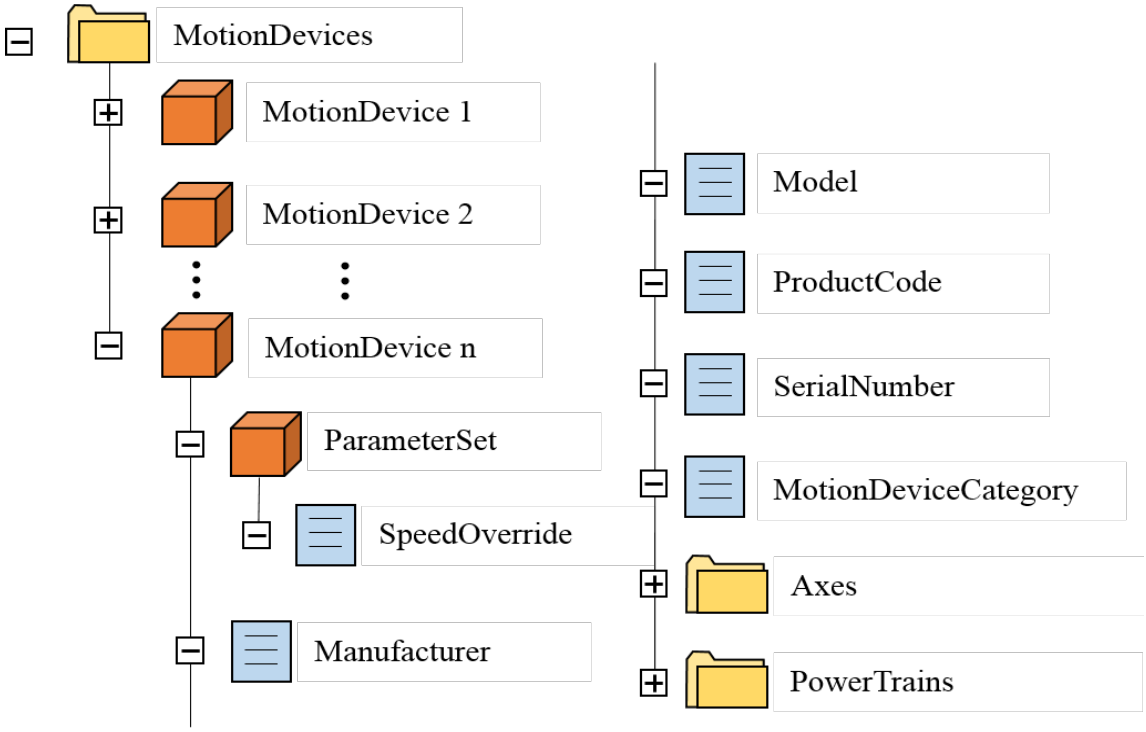
55.5.2 MotionDeviceSystem

The MotionDeviceSystem node is the node based on OPC UA for Robotics.
It has nodes for MotionDevices, Controllers, and SafetyStatus.



55.5.2.1 MotionDevices

The MotionDevices node provides information such as robot model, overrides, axes, and motors.
It has as many MotionDevice nodes as the number of robot groups.



SpeedOverride

The SpeedOverride node provides the robot speed override.
The value from 1 to 100 (%) is set to the node value.

Manufacturer

The Manufacturer node provides the name of the robot manufacturer.
“FANUC” is set to the node value.

Model

The Model node provides the robot model name.
e.g. R-2000iC/165F

ProductCode

The ProductCode node provides the product code of the robot.
Since there is no corresponding data, null is set to the node value.

SerialNumber

The SerialNumber node provides the serial number of the robot.
Since there is no corresponding data, null is set to the node value.

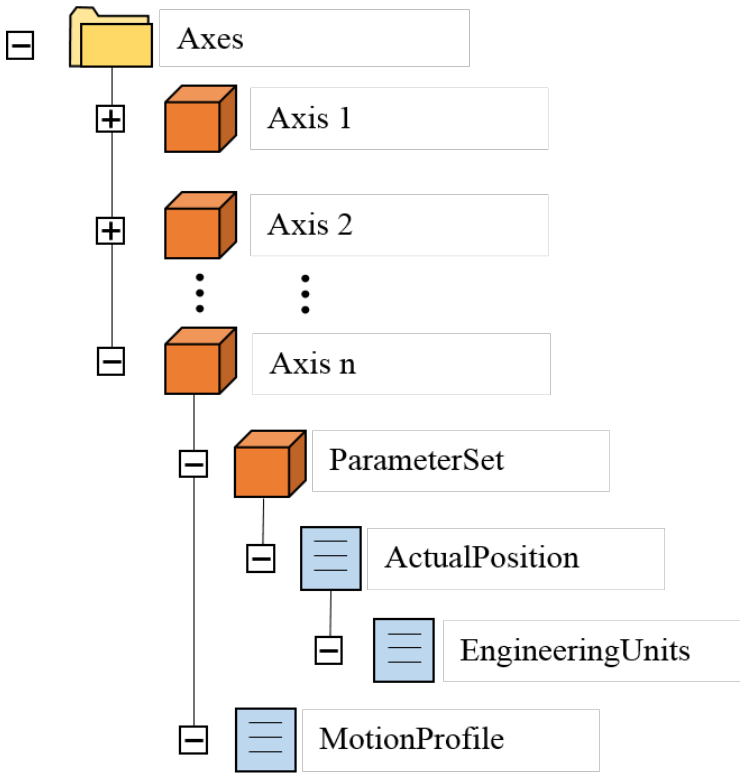
MotionDeviceCategory

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 which is not defined.
ARTICULATED_ROBOT	1	This robot design features 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	Robot has two parallel rotary joints to provide compliance in 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 prismatic joints deliver a linear motion along the 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 prismatic joint to connect the links. The rotary joint uses a rotational motion along the joint axis, while the prismatic joint moves in a linear motion. Cylindrical robots operate within a cylindrical-shaped work envelope.

Axes

The Axes node provides the axis information.
It has as many Axis nodes as the number of robot axes. This supports until 9 axes for extended axis and independent axis.



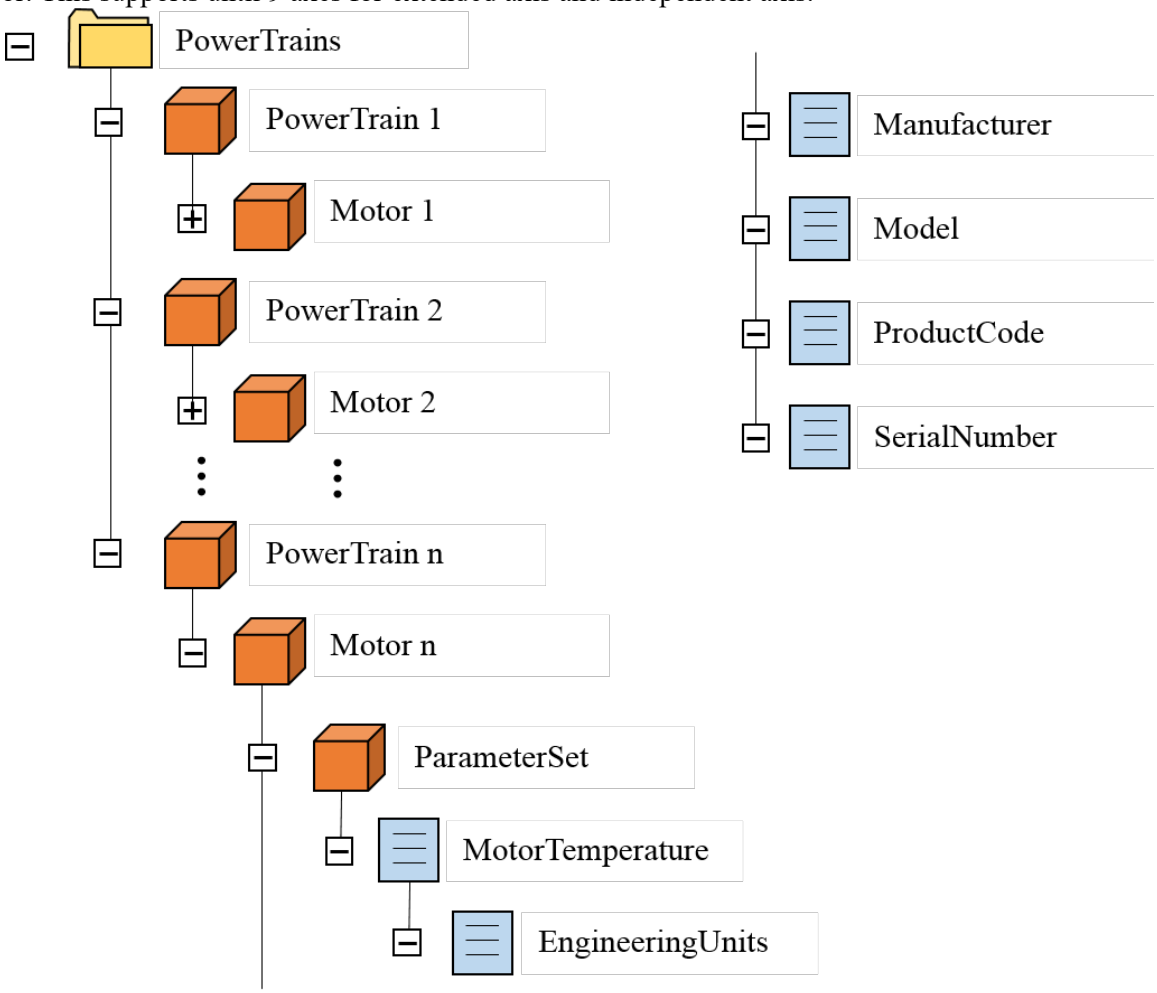
ActualPosition
The ActualPosition node provides the current position of the axis.

EngineeringUnits
The EngineeringUnits node provides the units for the ActualPosition node.
The unit of ActualPosition is “deg.” for rotary axis and “mm” for linear axis.

MotionProfile
The MotionProfile node provides whether the axis is linear or rotary and whether it rotates continuously.
The category value corresponding to the following table is set to the node value.

Category	Value	Description
OTHER	0	The axis that does 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.
It has as many PowerTrain nodes as the number of robot axes and as many Motor nodes as the number of motor. This supports until 9 axes for extended axis and independent axis.



MotorTemperature
The MotorTemperature node provides the motor temperature.
Since there is no corresponding data, null is set to the node value.

EngineeringUnit
The EngineeringUnit node provides the units for the MotorTemperature node.
Since there is no corresponding data, null is set to the node value.

Manufacturer
The Manufacturer node provides the name of the motor manufacturer.
“FANUC” is set to the node value.

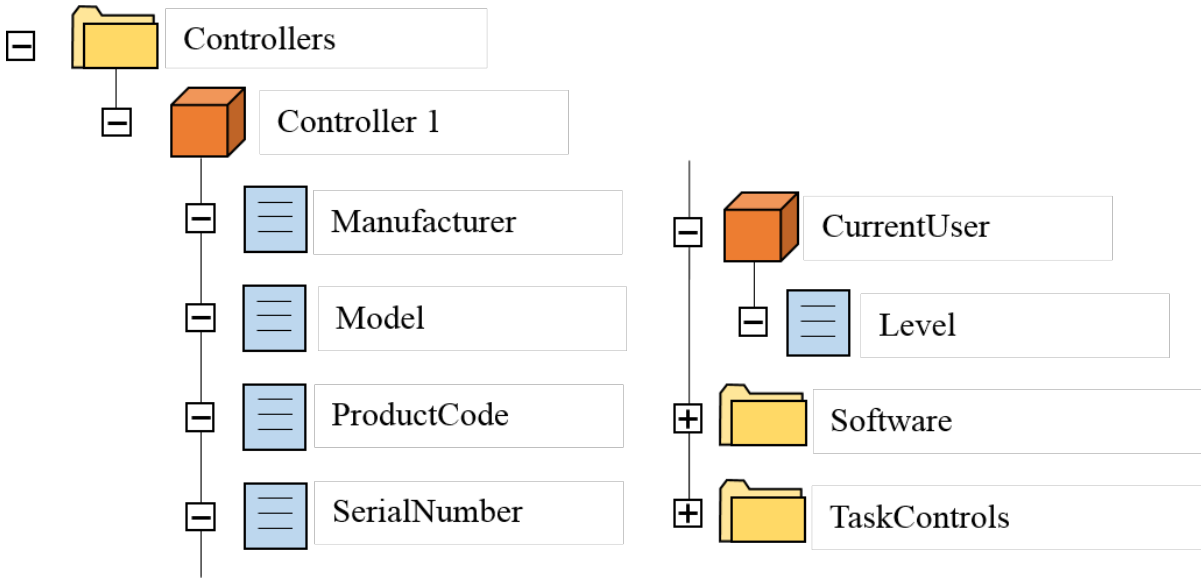
Model
The Model node provides the motor model name.
e.g. aiSR22/4000 80A

ProductCode
The ProductCode node provides the product code of the motor.
Since there is no corresponding data, null is set to the node value.

SerialNumber
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 Controller 1 node. Since the robot controller becomes the OPCUA server, there is only one Controller node.



Manufacturer
The Manufacturer node provides the name of the robot controller manufacturer.
“FANUC” is set to the node value.

Model
The Model node provides the robot controller model name.
e.g. R-30*i*B Plus

ProductCode
The ProductCode node provides the product code of the robot controller.

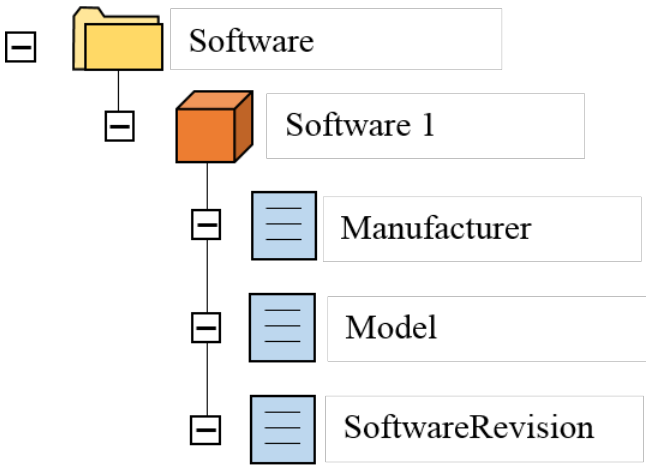
SerialNumber
The SerialNumber node provides the serial number of the robot controller.

Level
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
SETUP	Setup
PROGRAM	Program
OPERATOR	Operator

For the operation authority of each level, refer to the “Operator’s Manual (Basic Function)” (B-83284EN) “9.11 Password Function”.

Software
The Software node provides information on the software installed on the robot controller.
The Software node has the Software1 node.



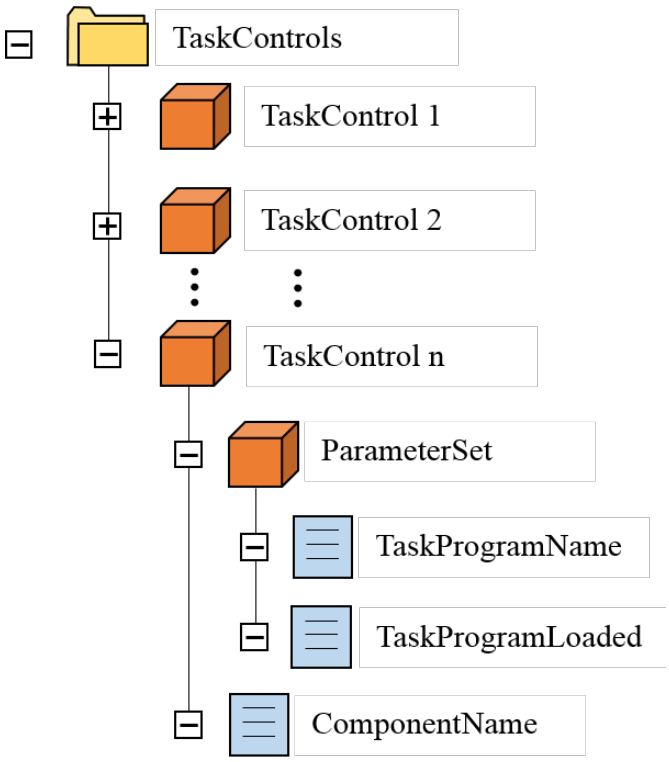
Manufacturer
The Manufacturer node provides the name of the software manufacturer.
“FANUC” is set to the node value.

Model
The Model node provides the software model name.
e.g. R-30*i*B Plus

SoftwareRevision
The SoftwareRevision node provides the version of the software.
e.g. 7DF1/19 (V9.10P/19)

TaskControls

The TaskControls node provides information on user tasks.
It has as many TaskControl nodes as the number of tasks set in the maximum number setting.



TaskProgramName

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

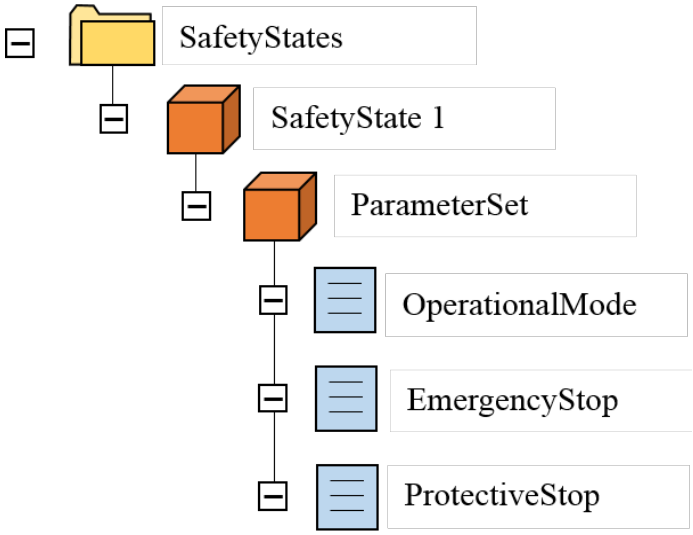
The TaskProgramLoaded node provides the loading status of the user program.
This node is set to 1 if the user program has been loaded and 0 if not.

ComponentName

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

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

The EmergencyStop node provides the status of the emergency stop signal.
This node is set to 1 when any one of the SOP E-STOP, TP E-STOP, and Ext E-STOP is input, and is set to 0 otherwise.

ProtectiveStop

The ProtectiveStop node provides the status of the stop signal.
This node is set to 1 when any one of the Fence Open, TP Deadman, TP Enable, Hand Broken, Overtravel, Low Air Alarm, Belt Broken, SVOFF Input, and Non Teacher Enb.Dev. is input, and is set to 0 otherwise.

56 FAULT & INCIDENT REPORT

FAULT & INCIDENT REPORT is a function to analyze alarms occurred in the past. It is possible to display the top5 faults about the number of incident, total time, and the longest time with graph.

This function is optional function. To use this function, “FAULT & INCIDENT REPORT option (R544) is required.

56.1 ALARMS REPORTING SCREEN

You can check the top 5 alarms about the number of incident, total time, and longest time of the alarms which are occurred at AUTO mode.
You can display Alarms Reporting Screen by selecting [MENU] key ->Alarm->F1[TYPE]->Reporting.

Alarms whose severity are NONE or WARN are not displayed on this screen.

INCIDENT

The top 5 faults by incident are displayed. Faults are located in order of the number occurred in the past. Each value and bar graph means the number of incident. This number is counted when the target fault is occurred at AUTO mode. (However, only alarms “SYST-038 Operation mode T1 Selected” and “SYST-039 Operation mode T2 Selected” are counted at T1mode, T2mode too)

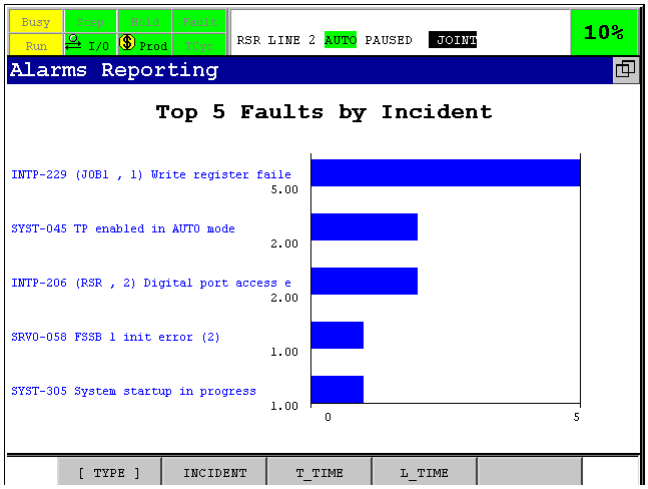


Fig. 56.1(a) Alarms Reporting INCEIDENT Screen

Total Time(T_TIME)

The top 5 faults by Total Time are displayed. Faults are located in order of the total time occurred in the past. Each value and bar graph means the total time (unit is 1/10 minute). The incident time means time after the target fault is occurred at AUTO mode before you press the fault reset key. And the total time means sum of the incident times. (However, only alarms “SYST-038 Operation mode T1 Selected” and “SYST-039 Operation mode T2 Selected” are recorded at T1mode, T2mode too)

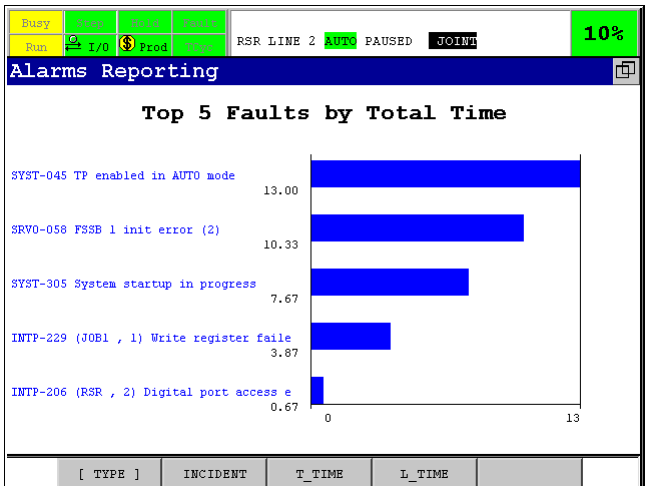


Fig. 56.1(b) Alarms Reporting Total Time Screen

Largest Time(L_TIME)

The top 5 faults by Largest Time are displayed. Faults are located in order of the largest time occurred in the past. Each value and bar graph means the largest time (unit is 1/10 minute). The incident time means time after the target fault is occurred at AUTO mode before you press the fault reset key. And the largest time means the largest time in the incident times. (However, only alarms “SYST-038 Operation mode T1 Selected” and “SYST-039 Operation mode T2 Selected” are recorded at T1mode, T2mode too)

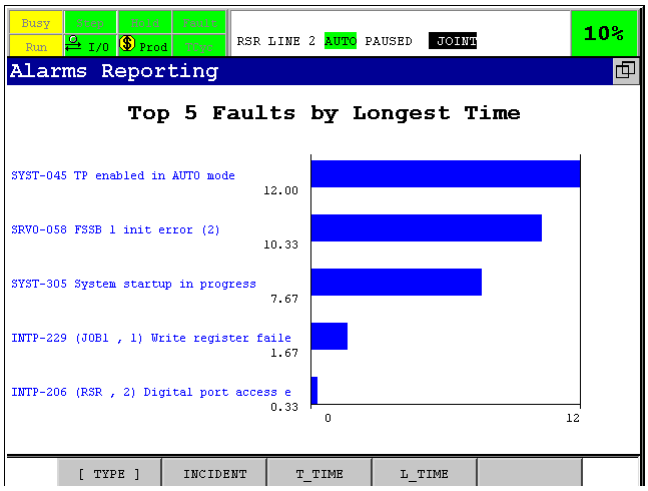


Fig. 56.1(c) Alarms Reporting Longest Time Screen

57 CORETEC BEAD INSPECTION INTERFACE FUNCTION

CORETEC bead inspection interface function is a function to connect robot with 3D WELD BEAD INSPECTION SYSTEM from CORETEC INC.
To use this function, software option "CORETEC bead inspection interface" (J714) and 7DC3 series or later are required. In 7DC3 series, P32 or later is necessary. In 7DF1 series, P04 or later is necessary. Series after 7DF1, there is no limitation.
Robot position data is sent to controller of 3D WELD BEAD INSPECTION SYSTEM by Ethernet.
Position data is position (X, Y, Z, W, P, R) of Tool Center Point (TCP) on world coordinate of robot.

Specified DO or RO can be ON only when robot speed is constant.
This signal is changed to ON only when override is 100% and this signal is changed to OFF when stopping by HOLD etc. or changed to not constant speed.
The function, which changes the signal to ON, become active only after CORE_START, which is KAREL program, is called.
The function, which changes the signal to ON, become inactive after CORE_END, which is KAREL program, is called. (This is default condition.)
Aforementioned condition is kept after power OFF/ON.
The case, that constant speed is kept during some motion command by connecting some motion command by CNT, isn't supported. If some motion command is done during the function, which changes the signal to ON, is enabled, the signal is changed to OFF at motion link point.

57.1 CABLE CONNECTION

It is recommended that CORETEC controller and robot controller are connected directly without switching hub etc. If switching hub exists during them, error may happen at CORETEC controller.
See "Ethernet Function OPERATOR'S MANUAL (B-82974EN) APPENDIX D CABLE CONNECTION" for other attention. Especially, do clamp (earthing) between cable and ground plate.
Only CD38A (Port 1) or CD38B (Port 2) can be used. CD38C can't communicate of this function.

⚠ CAUTION
1 Unshielded cable (UTP cable) is commercially available as 100BASE-TX twisted-pair cable: However, be sure to use shielded Category 5 twisted-pair cable (STP cable) to improve the resistance to electrical noise in an FA environment.
2 Error may happen if retransmission like general Ethernet communication is done because communication of this function requires fast response. Accordingly, severer anti-noise measures must be taken as compared with general Ethernet wiring.

57.2 SETTING (SYSTEM VARIABLES)

This function is set by system variables.
After changing system variables, cycle power is necessary.
System variables under $TCPIPCFG are saved to syshost.sv.
System variables under $CORT are saved to sysvars.sv.

$TCPIPCFG.$ARPSIZE

If this function uses Ethernet physical port 1 (CD38A) and if $TCPIPCFG.$ARPSIZE is 16, set 272 (up bit 8 from the original value). If this function uses Ethernet physical port 2 (CD38B) and if $TCPIPCFG.$ARPSIZE is 16, set 528 (up bit 9 from the original value). If this setting doesn't set, response takes long time once in every 15 minutes and alarm occurred in bead inspection system. If Ethernet port is used by other function, set back to original value (normally 16).

$CORT.$ENABLE (default 1:Enable)

Set CORETEC bead inspection interface function ENABLE (1) or DISABLE (0).

$CORT.$IO_TYPE（default 2:DO）

Set signal type which changes to ON when constant speed.
2:DO
9:RO
RO is more frequent update so if RO can be used, RO is recommended.

$CORT.$IO_NUM（default 0）

Set signal index which changes to ON when constant speed.
If this variable is set to invalid value like 0 etc., signal isn't changed to ON when constant speed.

$CORT.$PORT（default 502）

If this function is used with Modbus TCP server (R800), using port number is lapped over, so change listen port number of robot by this system variable and change destination port number in 3D welding bead inspection system.

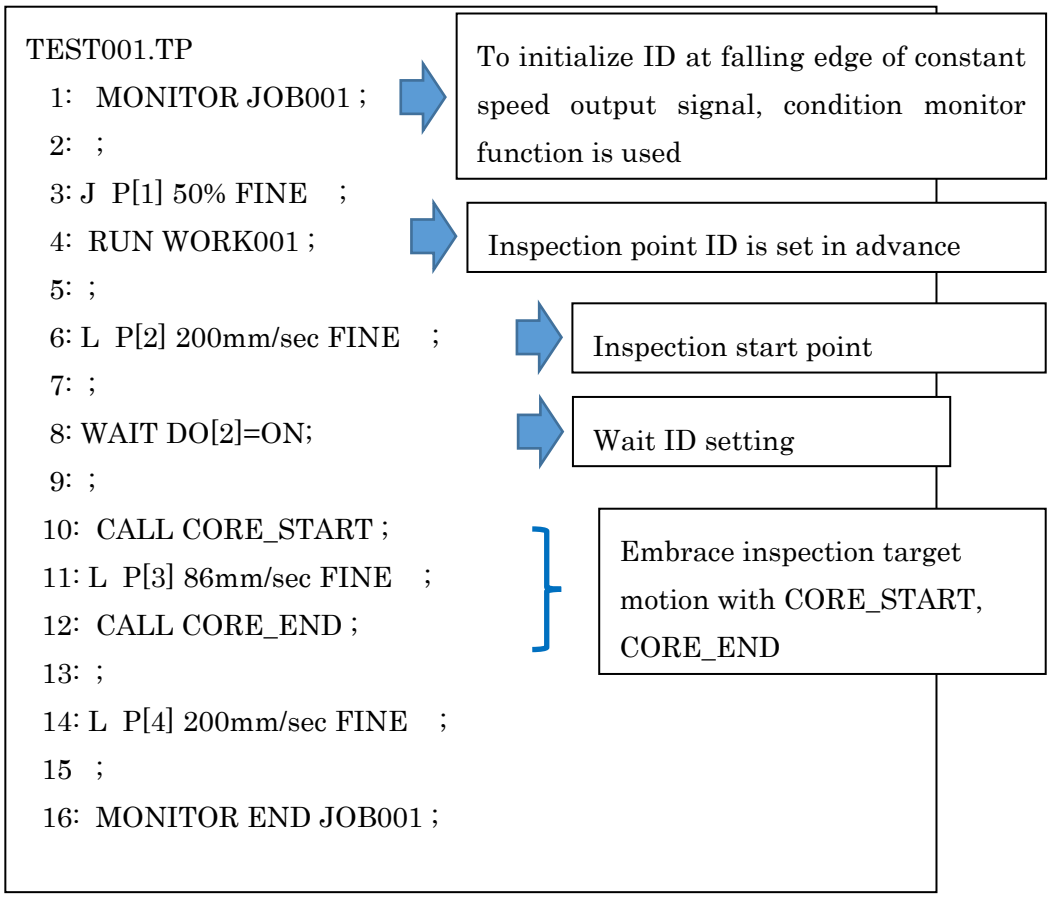
57.3 PROGRAM EXAMPLE

This section shows a program example.

Followings are supposed in example.

DO[1] is set to output when constant speed.
DO[2] is used to indicate that, inspection point ID is set already.
DO[3], DO[4] are used to set inspection point ID.
P[2] to P[3] is inspection point.

Main program : TEST001.TP



ID setting program: WORK001.TP
(Set all group mask to * from detail screen of program)

```
WORK001.TP
  1:  DO[3:id1]=ON ;
  2:  DO[4:id2]=OFF ;
  3:  WAIT    .10(sec) ;
  4:  DO[2:ids]=ON ;
```

Program for condition monitor function: JOB001.TP

Program for condition monitor function should be set to “Cond” in “Sub type” from F2 [DETAIL] when the program is made.

```
JOB001.TP
  1:  WHEN DO[1]=OFF-,CALL OFF234 ;
```

ID reset program: OFF234.TP (Set all group mask to * from detail screen of program.)

```
OFF234.TP
  1:  DO[2:ids]=OFF ;
  2:  DO[3:id1]=OFF ;
  3:  DO[4:id2]=OFF ;
```

58 RSI INTERFACE

"RSI" is abbreviated from "Remote Sensor Interface". This is a function to communicate with RSI device. "RSI device" is the device which can interpret "58.1 FORMAT OF THE TELEGRAMS SENT BY THE ROBOT" in this manual. This function sends position data etc. to device, which supports RSI, in real time. The order of this function is "RSI Interface" (J802).

There are following limitations to use this function.
- 7DC3 series or later is necessary.
- In 7DC3 series, P34 or later is necessary.
- In 7DF1 series, P04 or later is necessary. After 7DF1 series, there is no limitation.

This function support only 1st group.

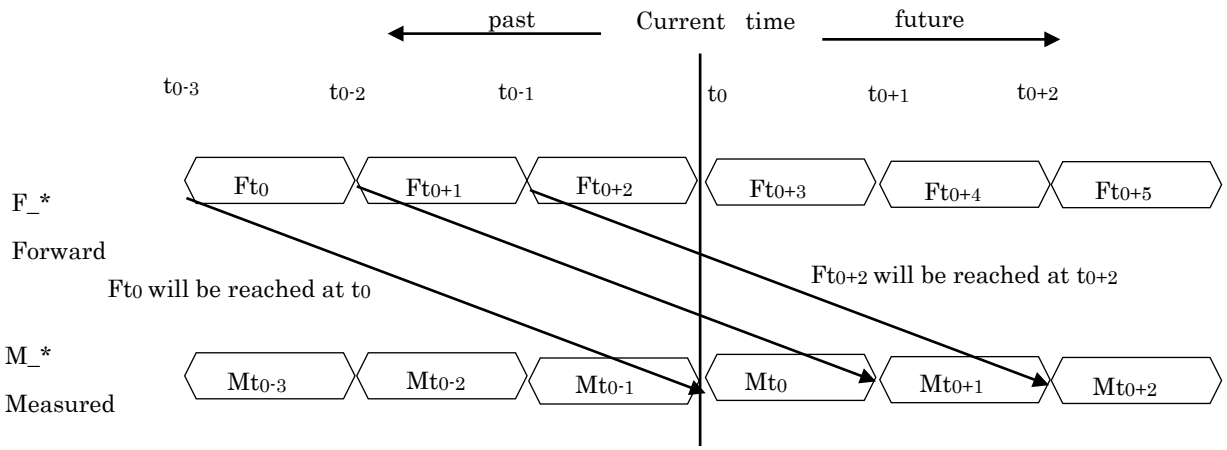
Data are sent by the robot controller through Ethernet port UDP communication to the external device every communication interval of this function. Current communication interval is set to system variable $RSI.$INTERVAL. Communication interval can't be changed.

There are 2 types of Cartesian positional data this option send to: Measured and Forward. Both are represented in XYZWPR format.

Measured Position is the actual position the robot moved to.
Measured Position is abbreviated as M_* where * are each of the XYZWPR positional elements.

Forward Position is the robot pre-planned position by default 3 interpolation times ahead of actual (or Measured) position.
Forward Position is abbreviated as F_* where * are each of the XYZWPR positional elements

Scheme below depicts relationship between Forward Position and Measured Position.



When the target position is planned in the robot controller, the target position data is immediately sent to external device via RSI but actual moving is waited for delay time ($AVC_GRP[1].$AVC_MODE[2]). Therefore there may be a delay when this function is enabled and delay time is set, hence cycle time becomes slightly longer, whose unit is communication interval, and the communication interval time is set to $RSI.$INTERVAL with millisecond unit.
Time Before and Distance Before functions would consequently shift by same amount too.
When this option is not ordered, the robot will move just after target position is internally planned by the robot controller.

58.1 FORMAT OF THE TELEGRAMS SENT BY THE ROBOT

The robot sends the XML data like following.

```
<Rob Type="FANUC">
    <M_X>-2851.123</M_X>
    <M_Y>1931.863</M_Y>
    <M_Z>1295.158</M_Z>
    <M_W>-91.728728</M_W>
    <M_P>17.543543</M_P>
    <M_R>177.519519</M_R>
    <F_X>-2856.183</F_X>
    <F_Y>1931.863</F_Y>
    <F_Z>1295.158</F_Z>
    <F_W>-91.728728</F_W>
    <F_P>17.543543</F_P>
    <F_R>177.519519</F_R>
    <IPOStat>0</IPOStat>
    <BMode>3</BMode>
    <TRG>0</TRG>
    <myRob>0</myRob>
    <IPOC>16</IPOC>
</Rob>
```

Wireshark EtherNet log:

```
0000    00 02 a2 2c d9 7d 90 1b 0e 3f d8 94 08 00 45 00  ...,.)...?....E.
0010    01 42 d7 f7 00 00 ff 11 99 08 c0 a7 64 2d c0 a7  .B..........d-..
0020    64 2e d0 cd 17 78 01 2e 18 36 3c 52 6f 62 20 54  d....x...6<Rob T
0030    79 70 65 3d 22 46 41 4e 55 43 22 3e 3c 4d 5f 58  ype="FANUC"><M_X
0040    3e 2d 32 38 35 31 2e 31 32 33 3c 2f 4d 5f 58 3e  >-2851.123</M_X>
0050    3c 4d 5f 59 3e 31 39 33 31 2e 38 36 33 3c 2f 4d  <M_Y>1931.863</M
0060    5f 59 3e 3c 4d 5f 5a 3e 31 32 39 35 2e 31 35 38  _Y><M_Z>1295.158
0070    3c 2f 45 5f 5a 3e 3c 4d 5f 57 3e 25 39 31 2e 37  </M_Z><M_W>-91.7
0080    32 38 37 32 38 3c 2f 4d 5f 57 3e 3c 4d 5f 50 3e  28728</M_W><M_P>
0090    31 37 2e 35 34 33 35 34 33 3c 2f 4d 5f 50 3e 3c  17.543543</M_P><
00a0    4d 5f 52 3e 31 37 37 2e 35 31 39 35 31 39 3c 2f  M_R>177.519519</
00b0    4d 5f 52 3e 3c 46 5f 58 3e 2d 32 38 35 36 2e 31  M_R><F_X>-2856.1
00c0    38 33 3c 2f 46 5f 58 3e 3c 46 5f 59 3e 31 39 33  83</F_X><F_Y>193
00d0    31 2e 38 36 33 3c 2f 46 5f 59 3e 3c 46 5f 5a 3e  1.863</F_Y><F_Z>
00f0    31 32 39 35 2e 31 35 38 3c 2f 46 5f 5a 3e 3c 46  1295.158</F_Z><F
0100    5f 57 3e 2d 39 31 2e 37 32 38 37 32 38 3c 2f 46  _W>-91.728728</F
0110    5f 57 3e 3c 46 5f 50 3e 31 37 2e 35 34 33 35 34  _W><F_P>17.54354
0120    33 3c 2f 46 5f 50 3e 3c 46 5f 52 3e 31 37 37 2e  3</F_P><F_R>177.
0130    35 31 39 35 31 39 3c 2f 46 5f 52 3e 3c 49 50 4f  519519</F_R><IPO
0140    53 74 61 74 3e 30 3c 2f 49 50 4f 53 74 61 74 3e  Stat>0</IPOStat>
0150    3c 42 4d 6f 64 65 3e 33 3c 2f 42 4d 6f 64 65 3e  <BMode>3</BMode>
0160    3c 54 52 47 3e 30 3c 2f 54 52 47 3e 3c 6d 79 52  <TRG>0</TRG><myR
0170    6f 62 3e 30 3c 2f 6d 79 52 6f 62 3e 3c 49 50 4f  ob>0</myRob><IPO
0180    43 3e 31 36 3c 2f 49 50 4f 43 3e 3c 2f 52 6f 62  C>16</IPOC></Rob
0190    3e 0a                                            >
```

Description of each tag

XML Tag	Description	unit	Value
Rob Type	Identifier of the robot manufacturer	-	FANUC
M_*	Measured position	X,Y,Z: mm W, P, R: degree	-
F_*	Forward position	X,Y,Z: mm W, P, R: degree	-
IPOStat	Status of the path interpolator. Currently always 0.		0
BMode	Describes the status of 3 mode switch of the robot.	-	1: T1 2: T2 3: AUTO
TRG	Signal "TRIGGER" for device. Static and as pulse.	-	0/1
myRob	Describes what the robot is handling.	-	0 : Workpiece 1 : RSI device
IPOC	This is incremented at every frame by communication interval time. Possibility to detect any missed frames.	ms	This value is 0 when 1st frame. If communication interval is 8ms, this value becomes following. 1st frame : 0 2nd frame : 8 3rd frame : 16 :

58.2 SETTING

Set the robot controller IP address and subnet mask etc. in the screen displayed by MENU key -> 6 SETUP -> Host Comm. -> TCP/IP.

Set the following system variables.
To enable the setting change, perform a COLD START or HOT START.

$AVC_GRP[1].$AVC_MODE[1] Default : 1
0: Disable this function (no robot data will be streamed to device)
1: Enable this function

$AVC_GRP[1].$AVC_MODE[2] Default : 0 (Max value is 20)
Delay time
F_* is 3 communication interval faster than M_* when this variable is set to 0.
When you set n to this variable, F_* becomes (n+3) communication interval faster than M_*.

$RSI.$INT_PORT Default:2
1: Using Main board Ethernet port 1 (CD38A)
2: Using Main board Ethernet port 2 (CD38B)

$RSI.$INTERVAL
Communication interval is showed to this variable with ms unit. Value is set at power-on time.
This system variable can't be changed.
e.g. 8ms.

$RSI.$PRIORITY Default:16
Priority of communication task in COMM CPU.
If sending time jitter performance is not enough, change this system variable to lower value.
1 is highest priority.

255 is lowest priority. (Not running.)
Setting high priority needlessly may influence to other functions. Please do minimum changing.

$RSI.$PORT_NUM Default:31439
Source port number of the robot in UDP communication.

58.2.1 Configuration of the Data Connection

Create on a PC by any text editor the following configuration XML file and copy it manually to controller at main root of FR:
Its file name can be selected by the argument of RSI_CREATE, which is KAREL program to set communication setting explained later.

e.g. RSI1.xml

```
<CONFIG>
<IP_NUMBER>136.230.168.112</IP_NUMBER>
<PORT>6008</PORT>
<SENTYPE>IPFO</SENTYPE>
<ONLYSEND>TRUE</ONLYSEND>
</CONFIG>
```

Remarks:
If file name is RSI*.xml, the file is saved by all backup and loaded by all load. * can be no character or any numbers or any characters. For example, if file name is RSI1.xml, the file is saved and loaded.
Start tag and end tag should be exactly same case sensitive.
IP_NUMBER = IP address of the RSI device
PORT = Port number of the RSI device.
SENTYPE = Any name describing external device.
ONLYSEND = Unidirectional (TRUE) or bidirectional (FALSE) data connection. Unidirectional (TRUE) is only supported. If ONLYSEND is set to other than TRUE, STOP global alarm "HOST-390 RSI: ONLYSEND in RSI configuration XML should be TRUE" will happen.

58.3 KAREL PROGRAMS AND KAREL BUILT-IN ROUTINES

If user ordered "RSI Interface" (J802), KAREL programs and KAREL built-in routines are installed to controller. This function can be used with executing following TP program etc. with using installed KAREL programs.

Example of TP program

```
1: CALL RSI_CREATE('fr:abc.xml') -> Set communication setting with setting XML file fr:abc.xml.
2: CALL RSI_ON(1, 1)       -> Start data sending with absolute correction and with active UTOOL.
3: L P[1] 100mm/sec FINE
4: CALL RSI_WRXML('myRob', '1')        -> Set myRob tag value to 1.
5: CALL RSI_WRXML('TRG', '1')          -> Set TRG tag value to 1.
6: L P[2] 100mm/sec FINE
7: CALL RSI_OFF()                      -> Exit data sending.
8: CALL RSI_DELETE()                   -> Delete communication setting.
```

Normally, KAREL built-in routines are not used but they are prepared to be called from KAREL programs. The names of KAREL programs are without "_I" from KAREL built-in routines name. For example, KAREL built-in routine RSI_CREATE_I becomes KAREL program RSI_CREATE. This is because same name between KAREL program and KAREL built-in routine's name isn't accepted. The arguments of

KAREL programs are same with KAREL built-in routines. There is no value returned from KAREL programs. If you want to use KAREL built-in routines, please include rsi.ev (environment file) by writing "%ENVIRONMENT RSI" in your KAREL program. Followings explain about KAREL built-in routines.

58.3.1 RSI_CREATE_I( )

This function sets communication setting.

Syntax *RET* = RSI_CREATE_I(*File:IN*)

Element	Description
RET	Type: INTEGER Return values: ■ OK(0): Function executed successfully ■ If alarm happens, KAREL program will stop with "INTP-103(RSI_CREATE, 22) Program error" and the detail (cause code) indicates the reason.
File:IN	Type: STRING Path and name of the configuration XML file: Ex .fr:RSI1.xml

58.3.2 RSI_DELETE_I( )

This function deletes communication setting.

Syntax *RET* = RSI_DELETE_I( )

Element	Description
RET	Type: INTEGER Return values: ■ OK(0): Function executed successfully ■ If function is not executed successfully the program is stopping by error.

58.3.3 RSI_RESET_I( )

This function do absolutely same thing with RSI_DELETE_I( ).

Syntax *RET* = RSI_RESET_I( )

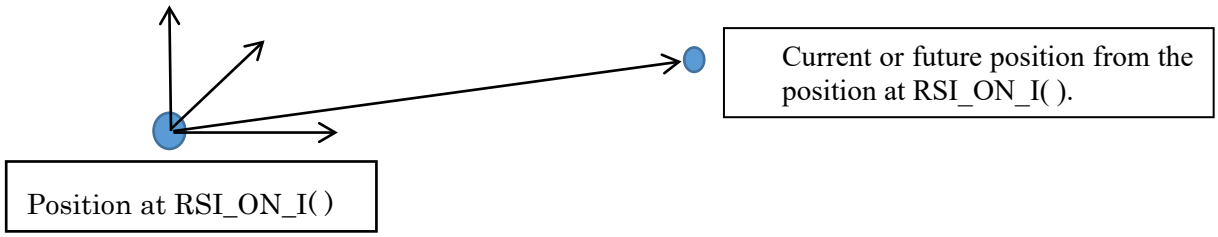
Element	Description
RET	Type: INTEGER Return values: ■ OK(0): Function executed successfully ■ If function is not executed successfully the program is stopping by error.

58.3.4 RSI_ON_I ( )

Data are sent every communication interval after this function is done.
The value of IPOC tag is reset to 0 by doing this function.

Syntax *RET* = RSI_ON_I(<*Correction mode:IN*>,<*Coordinate system:IN*>)

Element	Description
RET	Type: INTEGER Return values: ■ OK(0): Function executed successfully ■ If function is not executed successfully the program is stopping by error.
Correction mode:IN	Type: INTEGER Correction mode: ■ ABSOLUTE (1) : Absolute correction
Coordinate system: IN	Type: INTEGER Reference coordinate system ■ BASE (1) : active UTOOL is used ■ UFRAME (2) : active UFRAME and UTOOL is used ■ UTOOL (3) : active UTOOL is used, referenced from the time of activation RSI_ON_I(). See a following sketch example to show offset behavior.



58.3.5 RSI_OFF_I ( )

Data are not sent after this function is done.

Syntax *RET* = RSI_OFF_I( )

Element	Description
RET	Type: INTEGER Return values: ■ OK(0): Function executed successfully ■ If function is not executed successfully the program is stopping by error.

58.3.6 RSI_WRXML_I ( )

This function changes the value of XML tag.

Syntax *RET* = RSI_WRXML_I(<*XML attribute:IN*>, <*value:IN*>)

Element	Description
RET	Type: INTEGER Return values: ■ OK(0): Function executed successfully ■ If function is not executed successfully the program is stopping by error.
XML attribute:IN	Type: STRING This argument indicates which XML tag to write (e.g. 'TRG'). Case sensitive.
Value:IN	Type: STRING This argument indicates which value to write in the selected XML tag

This supports only "myRob" and "TRG" for XML attribute and value should be 1 character now.
The tag to change (write) is specified by argument 1. For changing multiple tags this program should be called n times with different 1st argument, which means XML attribute.

If not supporting tag is set to XML attribute (1st argument), STOP global alarm "HOST-391 RSI: 1st argument in RSI_WRXML_I is not supported" will happen.
If value (2nd argument) is larger than 1 character, STOP global alarm "HOST-392 RSI: 2nd argument in RSI_WRXML_I should be 1 character" will happen.

58.4 DETAILS

- Variables under $RSI are saved to rsi.sv.
- If this function J802 is ordered, $KAREL_ENB becomes 1 as default then karel can be used as default.
- If configuration XML file name is FR:\RSI*.xml, the file is saved and loaded by all backup or all load. At here, * can be nothing or any numbers or any characters.
- Karel programs, which are RSI_CREATE, RSI_DELETE, RSI_RESET, RSI_ON, RSI_OFF, and RSI_WRXML, are write protected.

59 SHAPE GENERATION FUNCTION

59.1 OVERVIEW

NOTE
This function is option, and requires Shape Generation Function Ⅲ （A05B-xxxx-R801）.

Shape Generation Function makes it easy to create machining programs aimed at cutting 2-dimensional shapes. You can reduce teaching time online by automatically generating a simple robot motion program using a graphic interface.
Using Shape Generation Function, you can generation robot motion program of following shape.

- Circle
- Slot
- Hexagon
- Rectangle
- Keyhole
- Pommel
- Custom

NOTE
Cutting Tool moves on only X-Y plane, so Pommel shape is not available

To perform machining operation using this function, the following operations are required.
1. Standard setting and shape data as basis of shape generation are set from setting screen.
2. Generate shape program from utility screen
3. Teach processing and correct program in the generated program
4. Run program

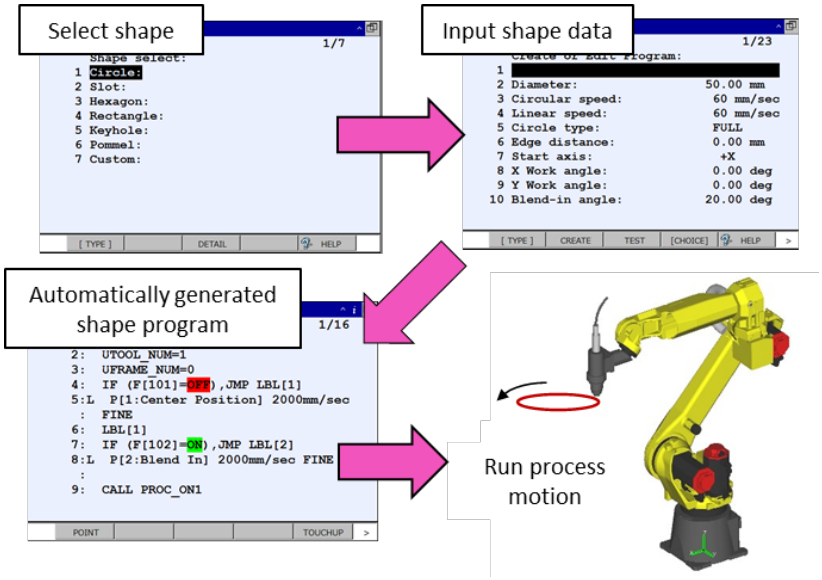


Fig. 59.1 (a) Function overview

> ⚠ **WARNING**
> When error occurs, check to stop machining process (e.g. Laser cutting or water jet cutting). Shape Generation function does not have the function to automatically stop machining process. Design system where machining process is properly stopped when error occurs. If machining process is not properly stooped, there is danger of harming worker and peripheral devices.

59.1.1 Program used for Shape Generation Function

For machining using Shape Generation Function, following TP program is required.

・Shape program

This is generated program based on input data in shape generation utility screen. In this program, path for shape generation is defined and it performs process defined in process program. This program includes only program instruction for these purposes.

・Production program

This is master program for calling shape program. Program instructions required except shape generation teach this program

> ⚠ **WARNING**
> Ensure that calling sequence of shape program for proper processing is performed correctly.
> When error occurs, check to stop machining process (e.g. Laser cutting or water jet cutting). Shape Generation function does not have the function to automatically stop machining process. Design system where machining process is properly stopped when error occurs. If machining process is not properly stooped, there is danger of harming worker and peripheral devices.

・Process program（optional）

This is program for defining machining process. When creating shape program other than custom, you can use PROC_ON#.TP and PROC_OF#.TP as Model programs (# is group number).
Without using these, you can create machining process to directly edit automatically generated shape program on user side.

59.2 SHAPE GENERATION SETUP

Basic setup of shape generation is done from the item "Shapes" in the submenu of the setup menu. Perform common setup in all figure shapes and standard setup of each shape generation.
In the Shape Generation Function, you can create robot motion programs of following shapes.

- ・Circle
- ・Slot
- ・Hexagon
- ・Rectangle
- ・Keyhole
- ・Pommel
- ・Custom

Standard setup can be set from the detail setup screen of each shape. The contents set on the setup screen are used as standard setup values on using the shape generation utility. Adjust the shape generation based on this content,

Procedure 59-1 Display of setup screen

Step

1 Press the [MENU] key.
2 Select SETUP.
3 Press the F1[TYPE].
4 Select SHAPES.
5 The following screen will be displayed.

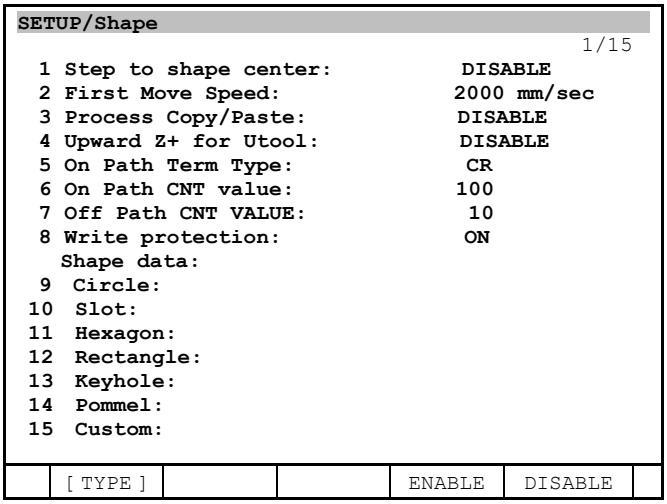


Fig. 59.2 (a) Setup screen

6 Select the desired shape type (Circle, Slot, Hexagon, Rectangle, Keyhole, Pommel, Custom), and press the F3[DETAIL].
7 After setup change, press the F2[Save] to apply the change content as the standard setting value.

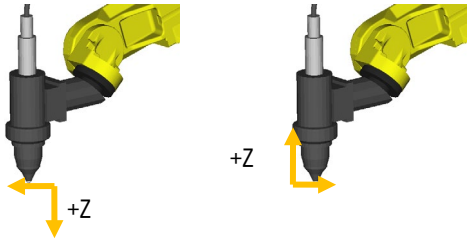
Refer to Table.59.2.1 for basic setup in shape generation, Table.59.2.2 for common setup for all shape data except for custom shapes, and from Table.59.2.3 to Table.59.2.9 for specific setup for individual shape.

59.2.1 Basic Setup

The basic setup items in figure creation are shown below.

Table 59.2.1 (a) Basic setup

ITME	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 indicates the move speed of the robot to the first position in the program.
Process Copy/Paste	When this is enabled, the information contained in the PROCESS_ONx.TP and PROCESS_OFx.TP programs is copied to the shape program. When it is disabled, the CALL to the process programs is used. For details, refer to "Process Copy/Paste " below.

ITME		DESCRIPITION
Upward Z+ for Utool		This is used when the standard ARC Tool Utool defines the tool Z direction to go into the TCP, instead of away from the TCP.  invers : disable invers : enable
On Path Term Type		This is used to control the term 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 is used to adjust the CNT value as the motion onto the end part of shape.
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 Slot.
	Hexagon	Selecting this item allows you to set up default shape information for Hexagon.
	Rectangle	Selecting this item allows you to set up default shape information for Rectangle.
	Keyhole	Selecting this item allows you to set up default shape information for Keyhole.
	Pommel	Selecting this item allows you to set up default shape information for Pommel.
	Custom	Selecting this item allows you to set up default shape information for Custom.

Process Copy/Paste

When [Process Copy/Paste] is disabled, the program calling PROC_ON1.TP and PROC_OF1.TP as follows is generated.

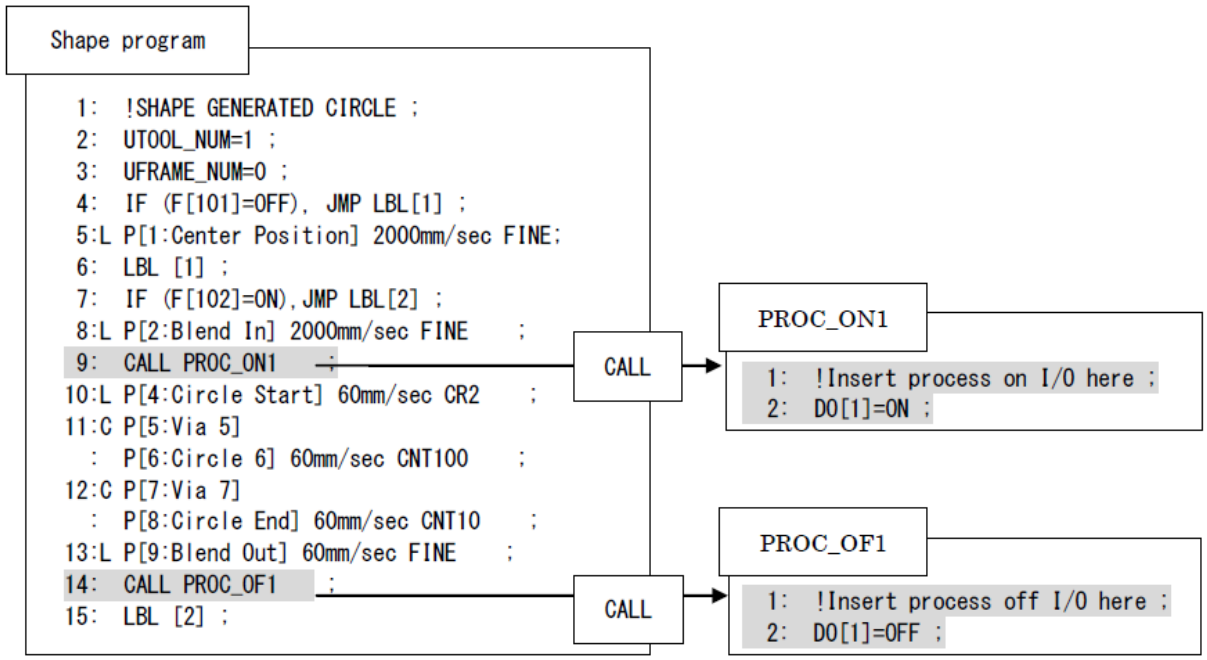


Fig. 59.2.1(a) Shape program sample (Process Copy/Paste is disabled)

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.

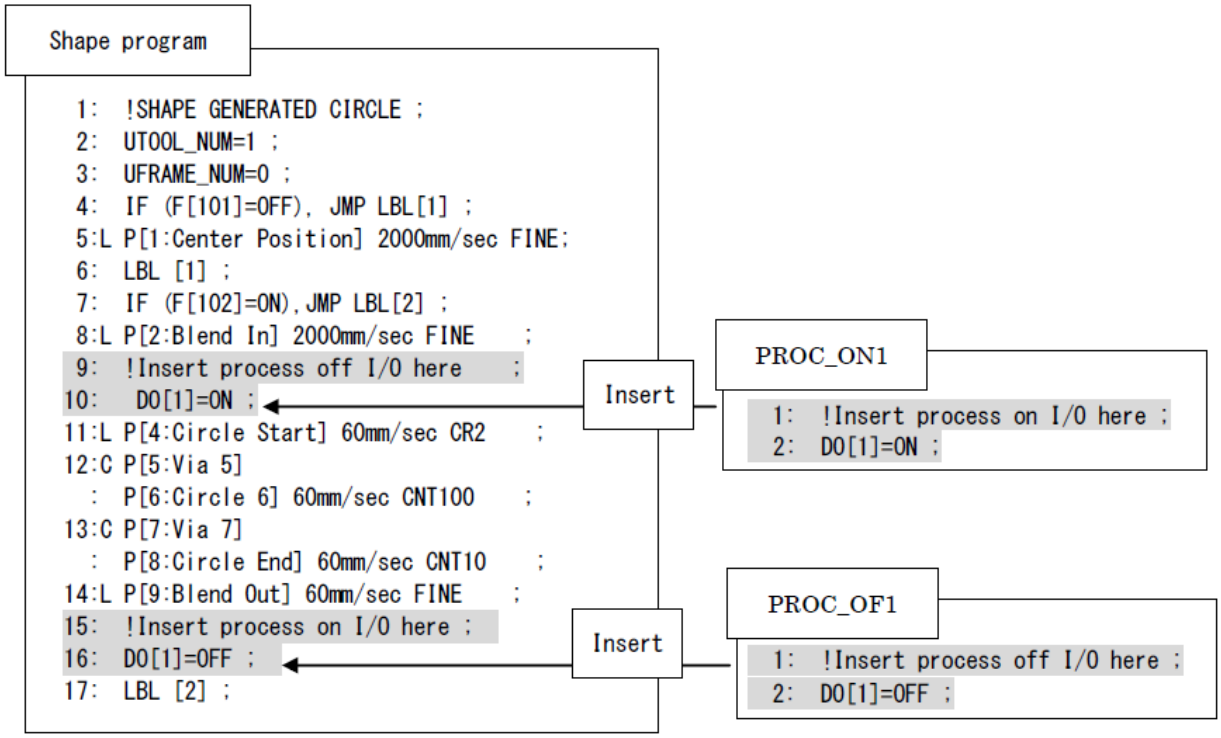


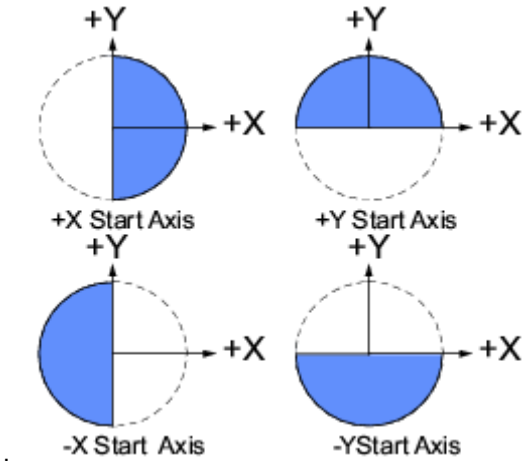
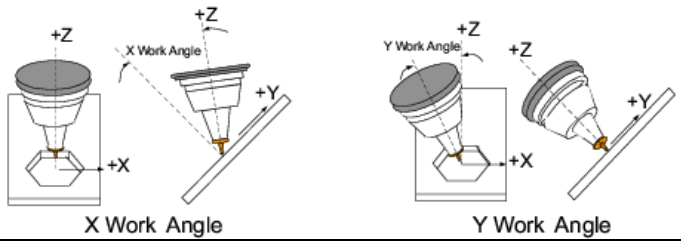
Fig. 59.2.1(b) Shape program sample (Process Copy/Paste is enabled)

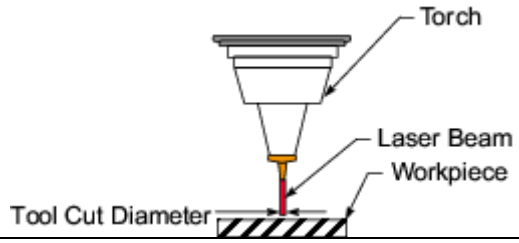
NOTE
Generated shape program automatically includes call instructions of PROC_ON and PROC_OF. When these programs are not exit, system software creates them automatically. After creating them Since they are not overwritten on the system software side once created, you can edit the instructions for process processing freely.

59.2.2 Common Setting of All Shapes

The setup items that are common to all the figure data except Custom are shown below.

Table 59.2.2 (a) Common setting

ITEM	DESCRIPTION
Start Axis (+/-X or +/-Y)	This item is defined as the axis on the shape at which that the robot will begin its cut. Each of the four (4) start axis choices are shown. Note: Half circle orientation depends on which start axis is implemented 
X-Work angle Range : ±45°	The X and Y Work angles are provided to 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 start position.
Overlap angle Range : ±45°	This item is the angle between the start axis and end position.
Blend-off angle Range : ±45°	This item is the angle from the start axis to the end position inside the shape.

ITEM	DESCRIPTION
Blend-off distance **Range : ±20 mm**	This item is the distance from the shape edge to the end position.
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 CR is used with a single point in the corner. If this is a non 0 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 TCP on 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

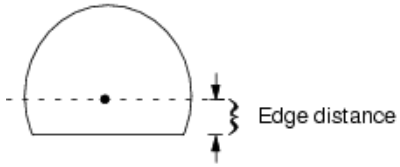
Cutting Tool moves on only X-Y plane, so the following items are not available;
- X-Work angle
- Y-Work angle
- Plunge distance
- Plunge speed
- Retract distance
- Retract speed

59.2.3 Specific Setup of Circular Shape

Specific setup items in creating circle shape are shown below.

Table 59.2.3 (a) Setup of circular shape

ITEM	DESCRIPTION
Diameter **Range : 1 - 500 mm**	This item is the diameter in mm of the current circle/half circle.
Circular speed **Range : 1 - 9999 mm/sec**	This item is the cutting speed of the circle/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.59.2.3 for the correspondence of the setup items in the generated shape.

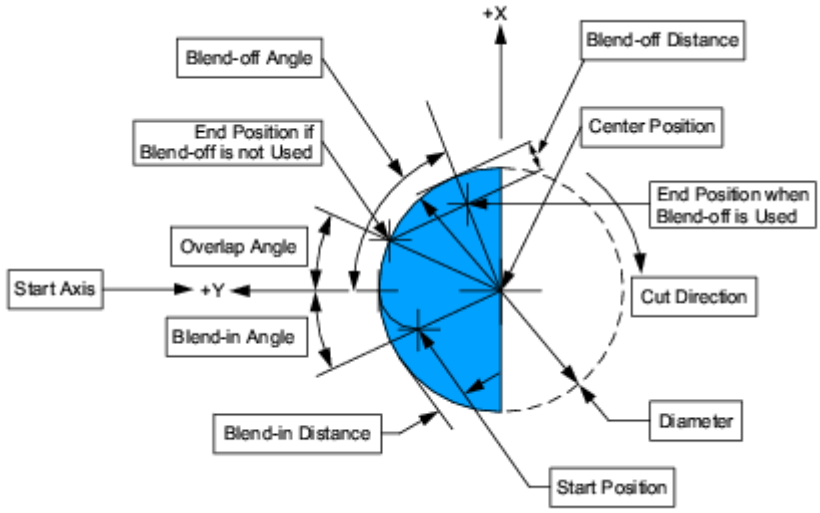


Fig.59.2.3 Setup of circle shape

59.2.4 Specific Setup of Slot Shape

Specific setup items in creating slot shape are shown below.

Table 59.2.4 (a) Setup of slot shape

ITEM	DISCRIPTION
Length **Range : （1 - 500 mm）**	This item is the length in mm across the longer side of the current slot.
Width **Range : （1 - 500 mm）**	This item is the width in mm 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. 59.2.4 for the correspondence of the setup items in the generated shape.

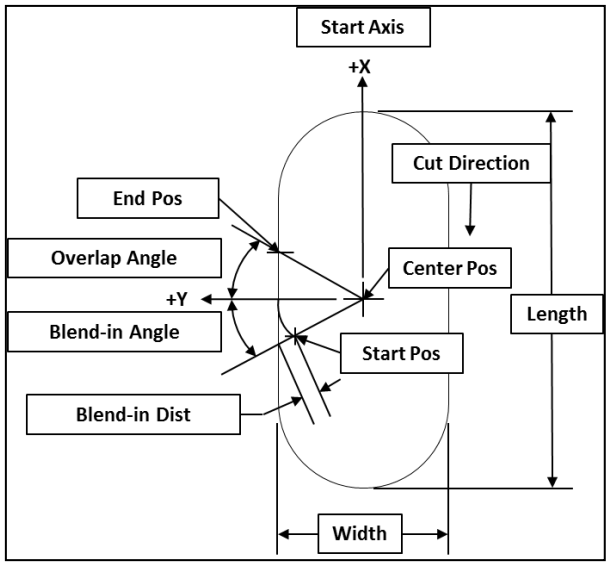


Fig.59.2.4 (a) Setup of slot shape

59.2.5 Specific Setup of Hexagon Shape

Specific setup items in creating hexagon shape are shown below.

Table 59.2.5 Setup of hexagon shape

ITEM	DISCRIPTION
Diameter **Range : (1 - 500 mm)**	This item is the diameter in mm 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.59.2.5 (a) for the correspondence of the setup items in the generated shape.

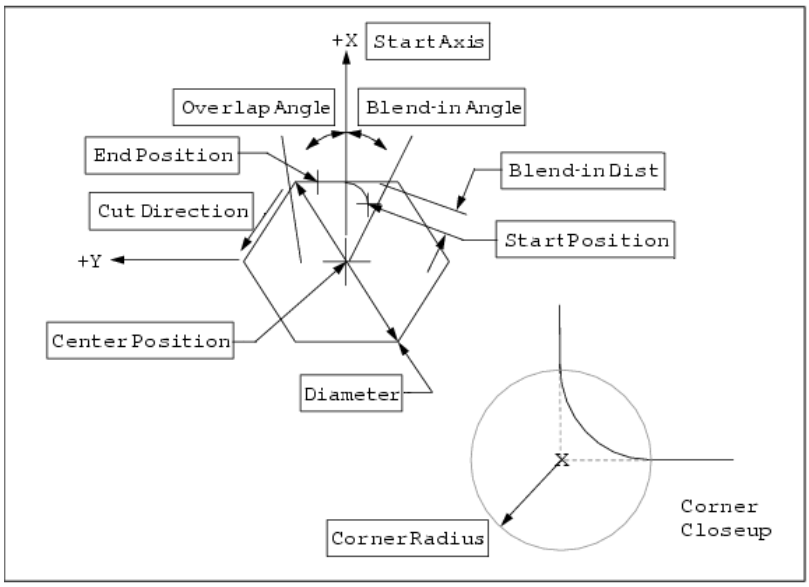


Fig.59.2.5 (a) Setup of hexagon shape

59.2.6 Specific Setup of Rectangle Shape

Specific setup items in creating rectangle shape are shown below.

Table 59.2.6 (a) Setup of rectangle shape

ITEM	DISCRIPTION
Length **Range : （1 - 500 mm）**	This item is the length in mm across the longer side of the current rectangle.
Width **Range : （1 - 500 mm）**	This item is the width in mm 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.59.2.6 (a) for the correspondence of the setup items in the generated shape.

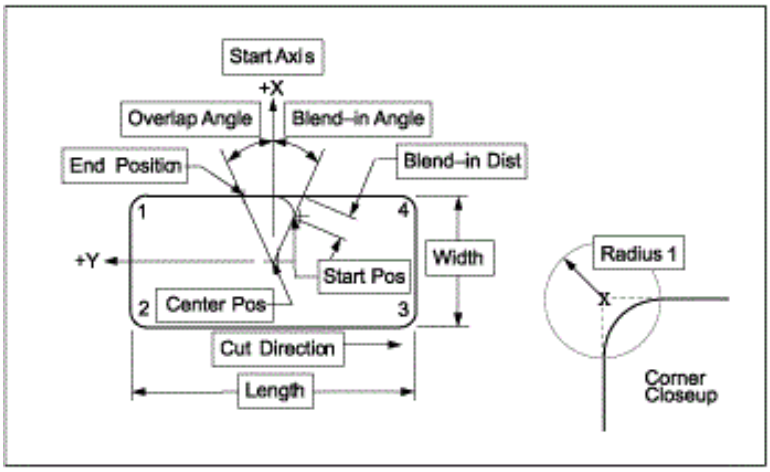


Fig.59.2.6 (a) Setup of rectangle shape

59.2.7 Specific Setup of Keyhole Shape

Specific setup items in creating keyhole shape are shown below.

Table 59.2.7 (a) Setup of keyhole shape

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 - 500mm）	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 linear portion of the current keyhole. You can also specify the speed units: mm/sec, cm/min, and in/min.

Refer to Fig.59.2.7 for the correspondence of the setup items in the generated shape.

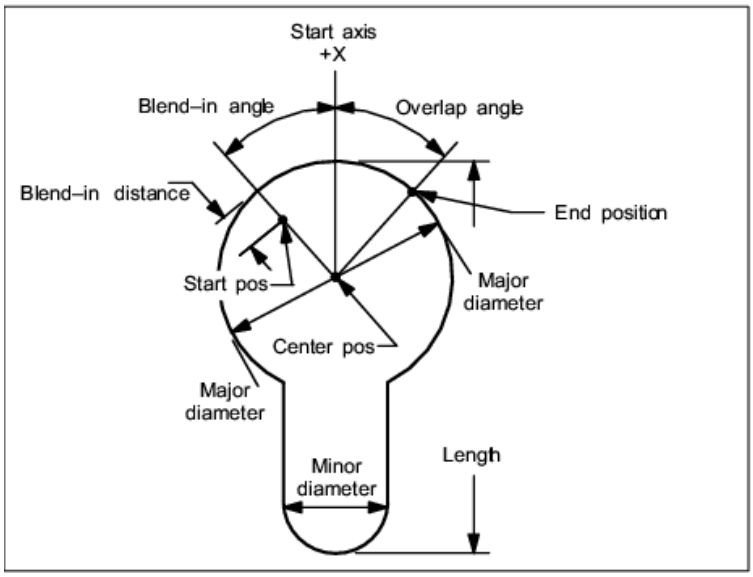
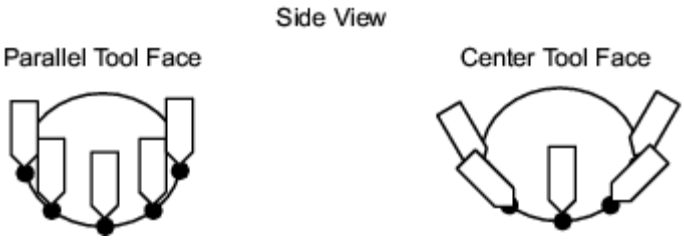


Fig.59.2.7 (a) Setup of slot shape

59.2.8 Specific Setup of Pommel Shape

Specific setup items in creating pommel shape (cut out of pipe) are shown below.

Table 59.2.8 (a) Setup of pommel shape

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.59.2.8 for the correspondence of the setup items in the generated shape.

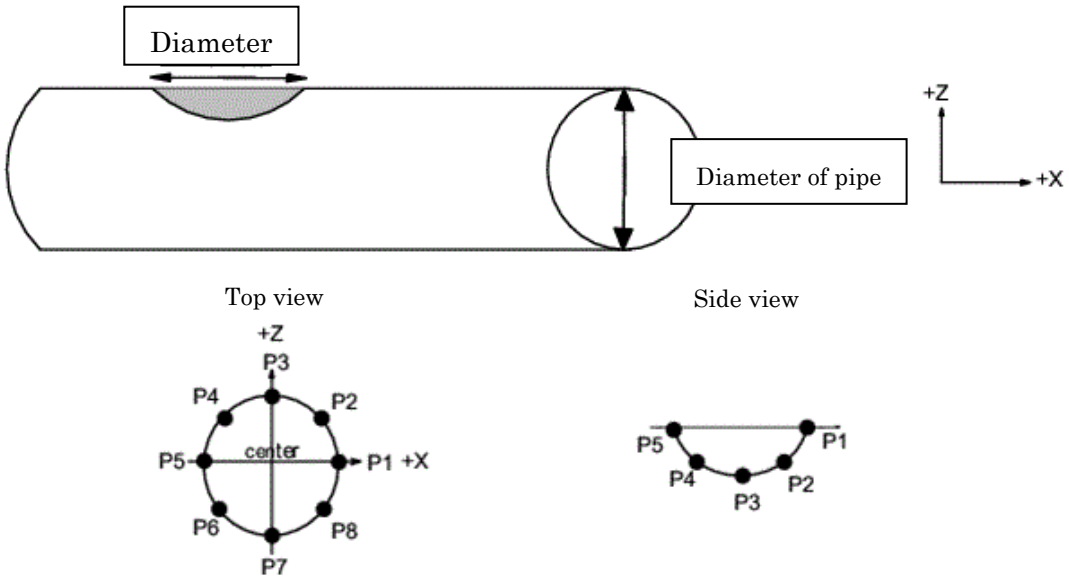


図 59.2.8 (a) Setup of slot shape

59.2.9 Custom

If select CUSTOM shape, it generates shape program with any shape template.
The following is the procedure to create shape program.

Step

1. Define the process path
   Process path is a position data of x,y,z coordinate system. This path needs reference position based motion, process start position and process end position. It can also include interpolation points connecting each point, if necessary, preliminary process at the start, and overlap at the end.

2. Create the custom shape template program
   Create a program based on the x, y, z coordinate system of the created machining pass. Define each coordinate system data as position data of the program. It is not necessary to specify attitude data (W, P, R).
   Also at this point.it does not need the robot TCP position. For shape data defined in template program, consider the robot TCP position when creating a shape program.
   And, template program can contain any TP instruction.

NOTE
1 The first point of the template program (position data with the smallest line number) is the reference position (TCP at the time of creation) when creating shape program. If it includes the reference position in process path, create a template program that skips the first point with a JMP LBL instruction.
2 Use FINE, CNT, and CR for the terminate type. Other terminate type cannot draw the shape with accuracy
3 Use Line and Circular for motion type. And posture data (W,P,R) is not used, then set these data to 0.
4 Define 64 point of maximum for path. Line number is not limited.

⚠ WARNING
The template program does not consider with the position data of the robot. Therefore, do not execute template program or modify directly position data that considering robot position on editor screen

For example, define a rhomboid shape with P1 as the reference position as shown below. (Unit is mm)

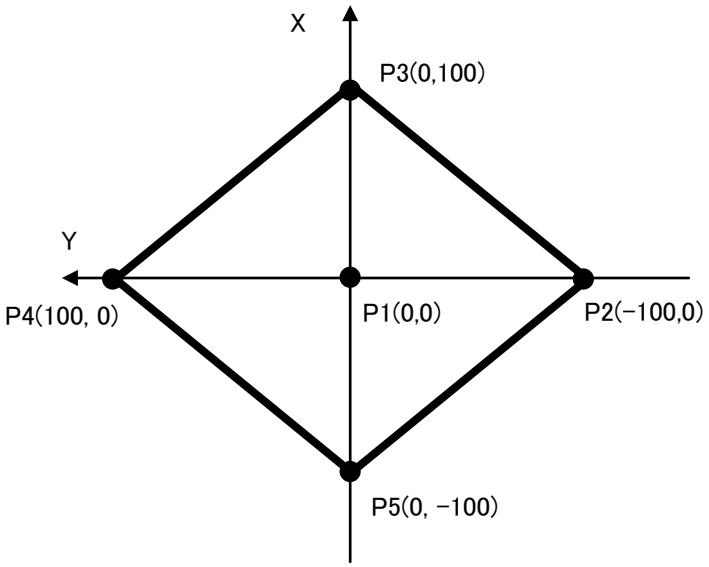


Fig. 59.2.9(a) Sample of template shape

In the case of such a shape, refer to the following as an example of creating a simple template program.

In case of passing through the reference position	In case of No 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.59.2.9(b) Sample program of template shape

3. Set up custom shape detail data.
   Table.59.2.9 shows specific setting items for creating custom shapes.

Table 59.2.9 (a) Setting items for CUSTOM shape

ITEM	DESCRIPTION
Template name	Specify the template program name to be used.
X Work angle	Specify bias angle for the X axis for diagonal cutting.
Y Work angle	Specify bias angle for the Y axis for diagonal cutting.
Rotation	Specify rotation angle of shapes about TCP.

59.3 SHAPE GENERATION UTILITY

59.3.1 Setting at UTILTIES screen

Specify detail setting for shape generation on UTILTILES screen. The standard value of the setting item on the UTILTILES screen is the content set on the setting screen. When it changes the setting contents on UTILTILES screen, the contents changed on UTILTILES screen at the time of shape generation are used preferentially.

NOTE
Modification of data on UTILILIES screen is not applied to standard value.

The setting items on UTILTILES screen are the setting items (common setting + setting per shape) on SETUP screen and the two items in Table.59.3.1 below. Refer to Table 59.2.2 (a) to Table 59.2.9 (a) for common setting and specific setting of each shape.

Table 59.3.1 (a) Setting items on UTILITIES screen (Additional items)

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 the F4, [CHOICE] to bring up 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. In case of an existing program, you can touch up the position and re-generate the program by pressing the F4, TOUCHUP. And you can also move the robot to the recorded center position by pressing the F5, MOVETO.

NOTE
In addition to the data you enter in the UTILITIES Shape Gen 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 GROUP that is selected. This will be the center point.

59.3.2 Create and Edit Shape program

The following is a procedure to create shape program at Shape Generation UTILITILE Screen.

STEP
1 Press the [MENU] key.
2 Select UTILITIES.
3 Press the F1[TYPE].
4 Select Shape Gen and press the [ENTER] key. The following screen will be displayed:

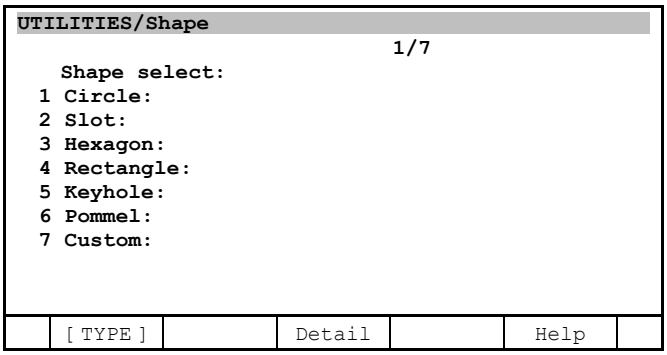


Fig.59.3.2(a) UTILITIES screen

5 Select the shape to create or modify and press the F3[DETAIL].
6 Start a new program, select an existing program to edit, or copy an existing program:
The following is a procedure **to start a new program,**
i. Move the cursor to Create or Edit Program and press the [ENTER] key.
ii. Type a name for the shape program that will be generated and press the [ENTER] key.

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 .

The following is a procedure **to edit an existing program**.
i. Move the cursor to Create or Edit Program and press the F4, [CHOICE]
ii. Select the name of the program to modify and press the [ENTER] key.

The following is a procedure to copy an existing program
i. Move the cursor to Create or Edit Program and press the F4, [CHOICE].
ii. Select the name of the program to modify and press the [ENTER] key.
The new program data will be copied from the existing one name. .
iii. Move the cursor to Create or Edit Program and enter the new program

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 section 59.2.1 –59.2.9 for each setting per shape

8. **To generate the program** when all the information is correct, press the CREATE. This will create the new shape program or re-create the existing edited program.

NOTE
When 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 TCP is correct when it creates shape program.

9. Add application and process-specific instructions to the PROC_ON#.TP and PROC_OFF#.TP if [Process Copy/Paste] is disabled

10. Make a test run of the program
i. Enable the Teach Pendant and set the program in the ready state.
ii. By pressing the F3[TEST] with holding the [SHIFT] key, generated program is run.

NOTE

Single step 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 which 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.

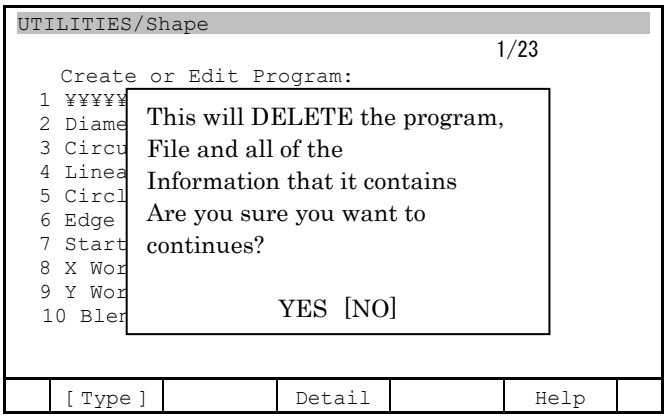


Fig.59.3.2(b) Massage at program re-creation

⚠ CAUTION

Pressing CREATE again will completely erase the current program and re-create it with the same name, using the data you set at the UTILITIES Shape Gen 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 press CREATE.

60 *i*RCalibration SIGNATURE

60.1 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 assembly 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 to achieve 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 Fig60.1(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 60.8.2). 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 just use the robot as standard robot and *i*RCalibration signature software make sure the robot reaches its taught position as accurately as possible.

Fig 60.1 (a) Label for high accuracy calibration

Custom Signature Calibration

To maintain the best accuracy of a robot, it is recommended to check its accuracy from time to time for 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 will 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.

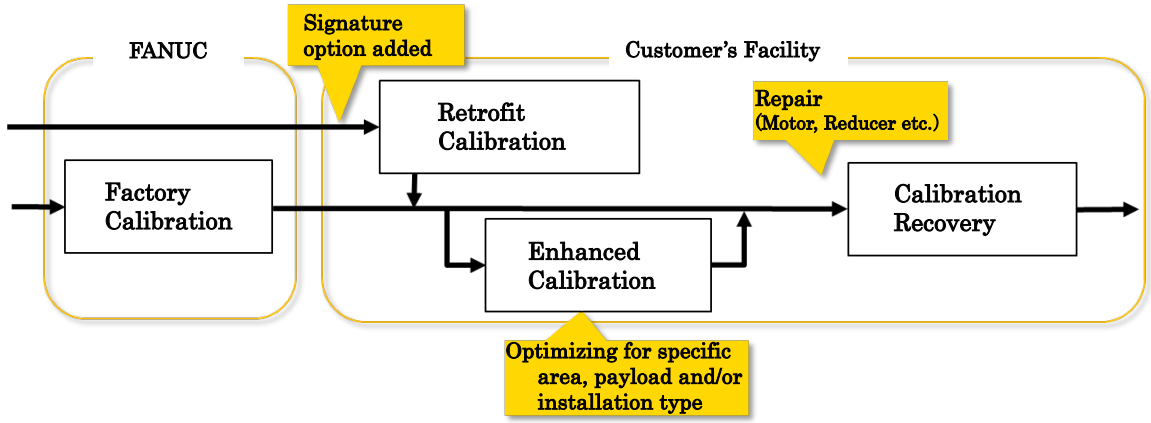


Fig 60.1 (b) Overview of custom calibration function

To do customer signature calibration, you will need 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. Similar to the factory calibration, this calibration requires measuring the reflector at various robot postures. Different calibration type has different requirement for number of measurement 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 notify you the calibration result. If the result is acceptable, the controller then loaded the new compensation parameters 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 Custom calibration function has to be conducted by FANUC personnel or personnel trained and certified by FANUC to ensure the correct procedure is followed. Please contact FANUC for the first time.

Signature Compensation

Once signature calibration parameters is 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 wide range of applications.

NOTE
Similar to gravity compensation, *i*RCalibration signature uses the payload information to compute the bending of the robot arm due to the gravity loading. Please refer to the Section 11.2 Motion Screen to set the accurate payload to take full benefits of the signature compensation function.

60.2 BENEFITS OF SIGNATURE CALIBRATED ROBOT

Simplifying off line programming

Signature calibrated robot makes Teach Pendant (TP) programs generated offline much easier to adopt in the real controller. The following example shows the steps that converts an offline generated TP program to a program in 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.

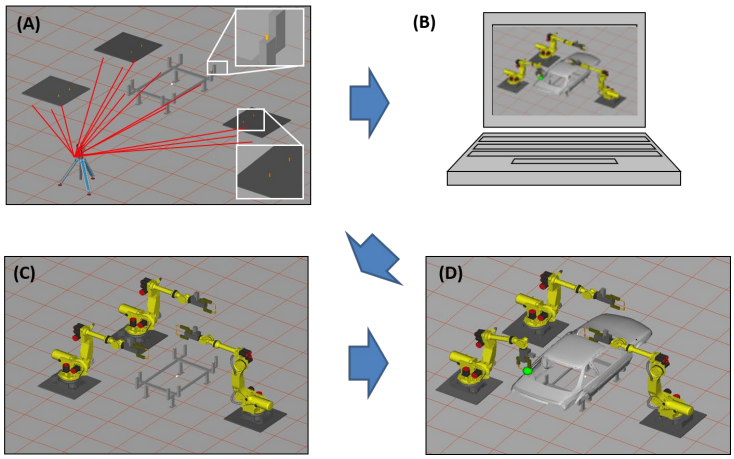


Fig 60.2(a) Example of robot system start-up

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 used in our Factory calibration as reference. Therefore, when replacing a signature calibrated robot with a new signature calibrated robot and using the location pin holes to place the new robot in correctly location, the new robot requires minimum touch ups. This "drop in replacement" can reduce down time of a production line significantly in case a robot is damage during production.

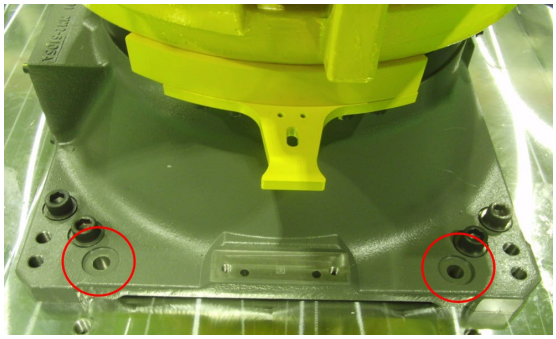


Fig 60.2(b) Locating pin holes on J1 base

60.3 CUSTOM SIGNATURE CALIBRATION

60.3.1 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 measured variety robot postures, a measurement data file from PC is loaded to robot controller via memory device (USB etc.). 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.

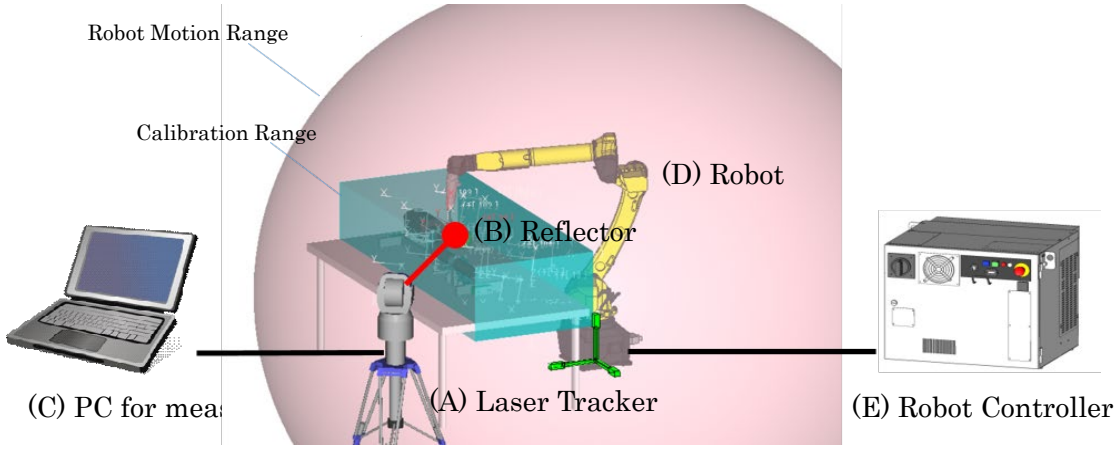


Fig 60.3.1(a) Calibration System

60.3.2 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 Not been 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 positon and measurement reflector position mounted on the robot's tool.

NOTE
1 Since the Retrofit Calibration is equivalent to the factory calibration, it has to be conducted by FANUC personnel or personnel trained and certified by FANUC to ensure the correct procedure is followed and an acceptable result can be obtained for this particular robot
2 Retrofit Calibration, Enhanced Calibration, Calibration Recovery are available in 7DC1/32 (V8.10P/32) , 7DC3/62 (V8.30P/62) , 7DF3/18 (V9.30P/18) , 7DF5/15 (V9.40P15) or later. Please contact FANUC for the status in 7DC2 (V8.20P) or 7DF1(V9.10P).
3 In 7DC3 (V8.30P) or later, calibration type “Signature” have same features with the Retrofit Calibration in the software version where Retrofit Calibration have not been supported. Functionality is improved in Retrofit Calibration. Therefore, it is recommended to update the software update to the latest version .

60.3.3 Custom Signature Calibration Preparation

Figure in below shows preparation work before custom signature calibration.

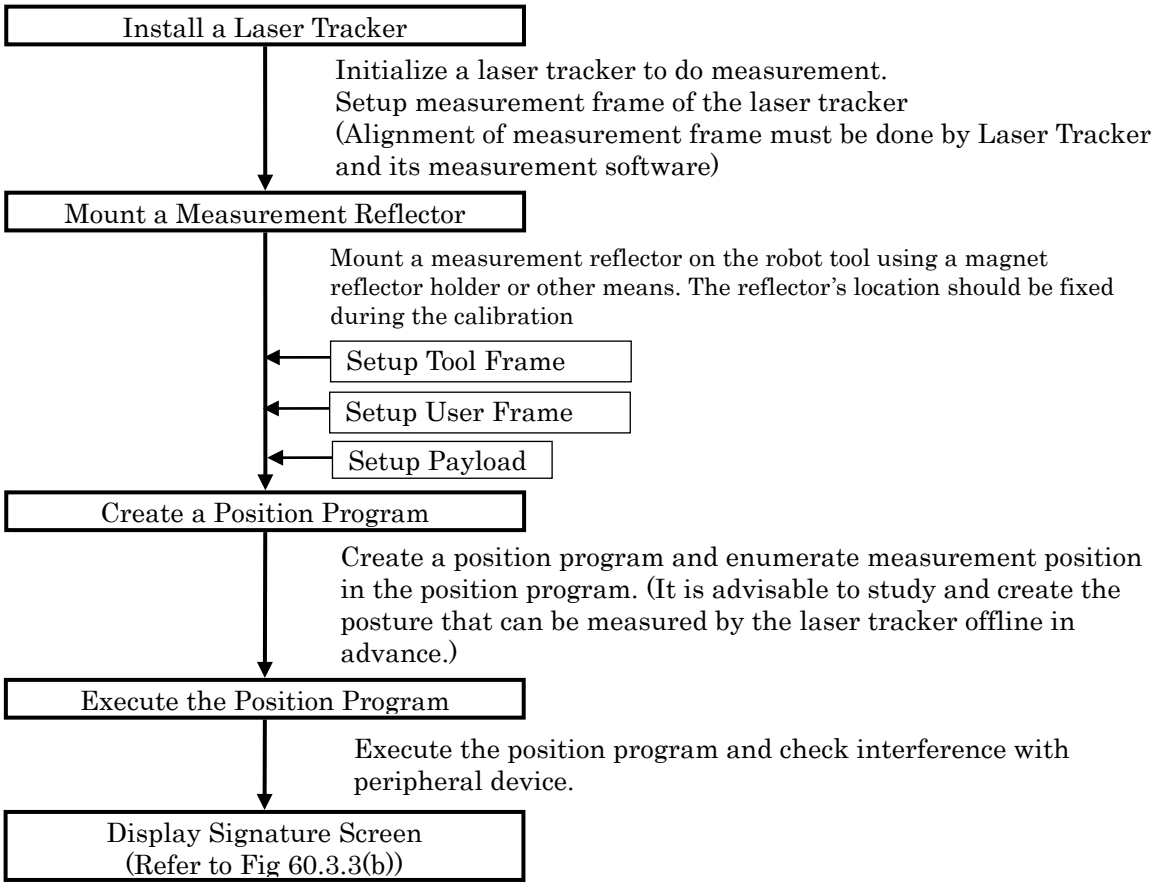


Fig. 60.3.3(a) Outline of custom signature calibration preparation

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 has to 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:

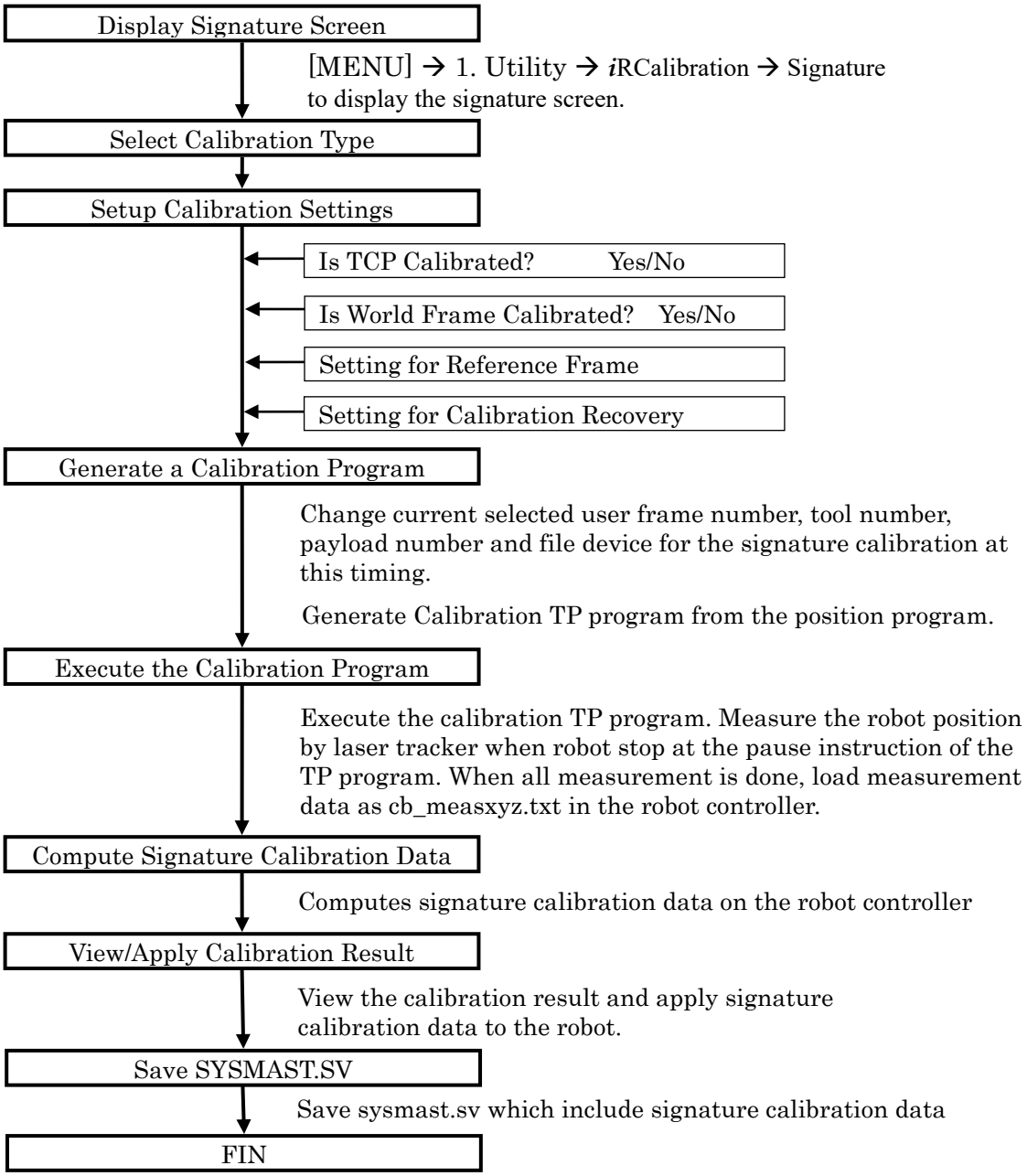


Fig 60.3.3(b) Operation flow of Signature screen

60.3.4 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

```
Program detail

 Program detail:
 1 CAL_POS
 2 Sub Type:  [None     ]
 3 Comment:   [        ]
 4 Group mask: [1,*,*,*,*,*,*,* ]

|  | END | PREV | NEXT | 1 | * |  |
```

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.

```
CAL_POS

P[1] UF:1 UT:1 ← The frame number to use
 J1   14.515 deg J4   0.000 deg
 J2   -5.787 deg J5  -88.723 deg
 J3   16.886 deg J6   0.000 deg
Position Detail
  1:J @P[1] 100% FINE
[END]

|  | END | PREV | NEXT | 1 | * |  |
```

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 point of 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 : 12, Recommended : 40-100 Better to include actual taught point where accuracy improvement is required.
Calibration Recovery	Minimum : 12, 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 using various posture (left) and actual taught points (right).

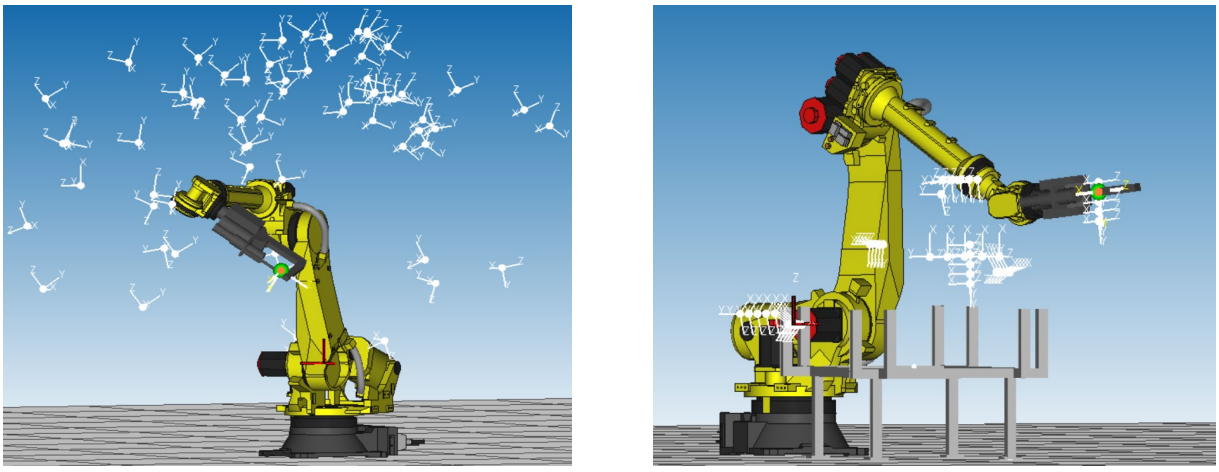


Fig 60.3.4(a) Example of position program

60.3.5 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.

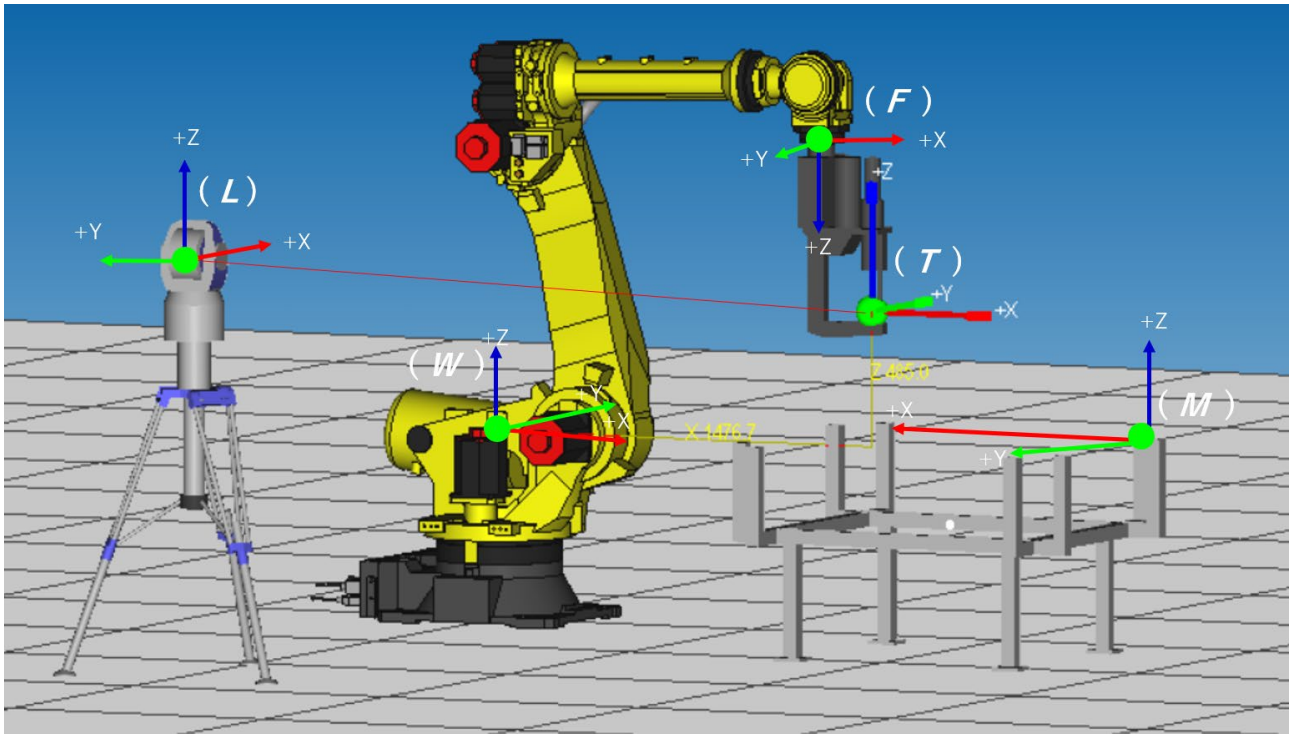


Fig 60.3.5(a) Coordinate systems in the signature calibration

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 to setup 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 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 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.

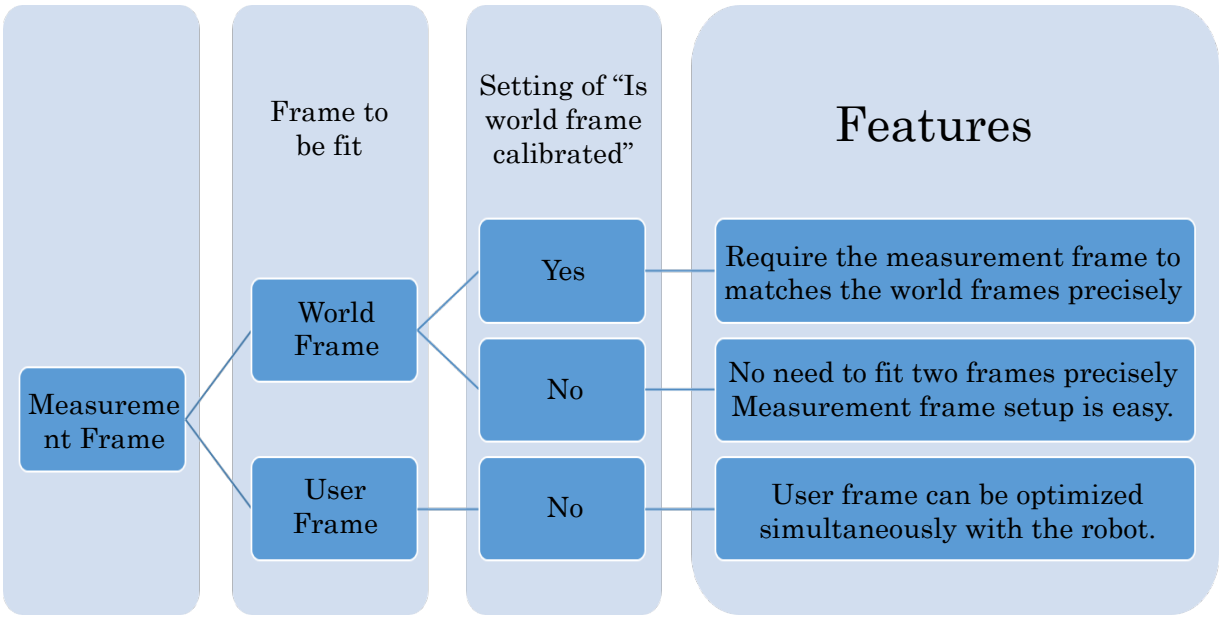


Fig. 60.3.5(b) Setup of measurement frame and its features

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 user frame defined on the 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 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 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.

60.4 BASIC FUNCTION

60.4.1 *i*RCalibration 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 term till you get the calibration result in the final screen.

***i*RCalibration Signature screen**

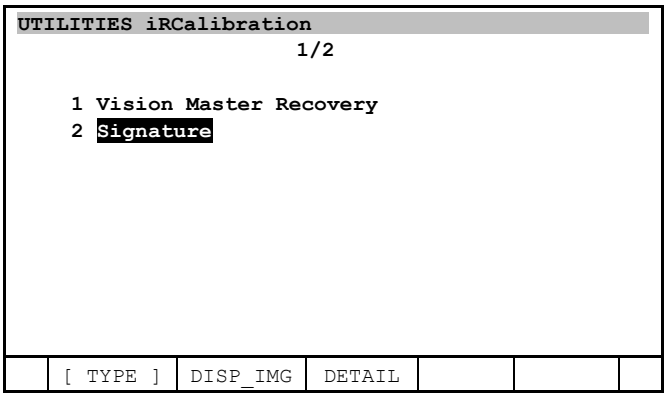
Items	Description
Group	Set the group number. Currently only Group 1 is supported.
Calibration Type	Select the calibration type.
Setup	Setup calibration settings.
Create Calibration Prog	Generate a calibration TP program from an existing position program.
Run Calibration Program	Execute the calibration 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 60-1 Display signature screen

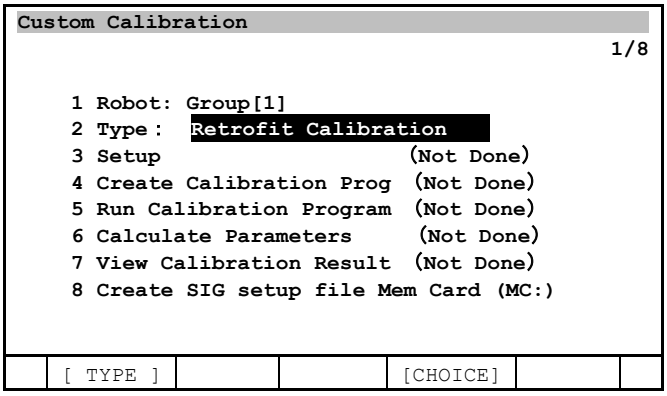
Step

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.



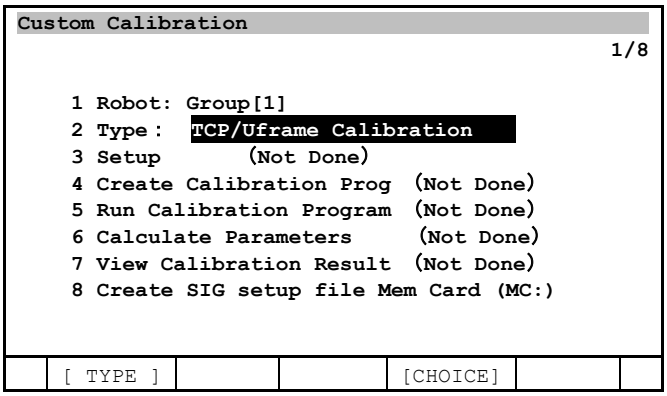
60.4.2 Calibration Type

On the item of Type on the signature screen, select a calibration type from 4 calibration types shown in the subsection 60.3.2.

Procedure 60-2 Change Calibration Type

Step

1 Display the signature screen. (Procedure 60-1)
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 “TCP/UFrame 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.

60.4.3 Setup Screen

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 60.5.1 Calibration Recovery. For the TCP/Uframe Calibration setup menu, please refer to Subsection 60.5.2.

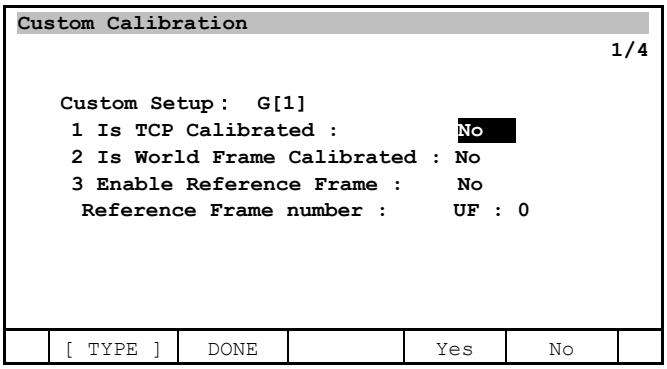
Items	Description
Is TCP Calibrated	Select Yes if the TCP is known, otherwise, select No.
Is World Frame Calibrated	Select Yes if the laser tracker's measurement frame is aligned to the robot's world frame. Otherwise, select No.
Reference Frame Enabled/Disabled	Select Yes if an known reference is used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

Please refer to Subsection 60.3.5 Coordinate Systems for the detail of the world frame calibration and the tool frame calibration. For the Reference Frame function, please refer to Subsection 60.5.3.

Procedure 60-3 Display setup screen

Step

1 Display the signature screen. (Procedure 60-1)
2 Switch the calibration type to Retrofit Calibration. (Procedure 60-2)
3 Move the cursor to the Setup item and press F4 "SELECT" to display a setup screen shown in below.
4 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".
5 If a message shows up ask you to recycle power, please cycle power.



60.4.4 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.

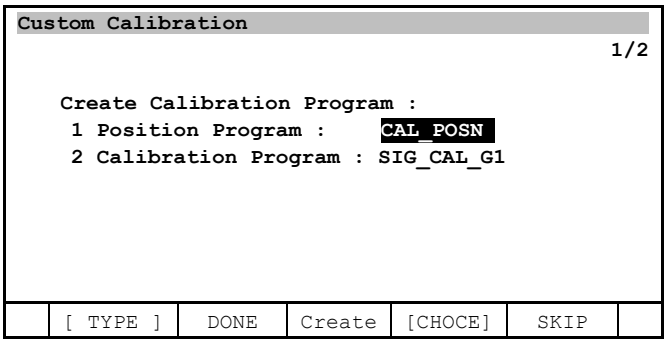
Procedure 60-4 Create Calibration TP Program

Condition

The status of the "Setup" item is "Done".

Step

1 Display the signature screen. (Procedure 60-1)
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.

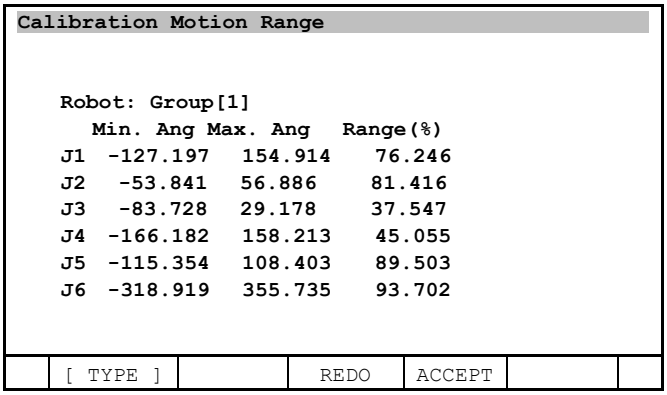
When setting the "Is World Frame Calibrated?" item is "No" on setup screen and user frame number used in position data of position program is 0 (world frame), the following message will be displayed to set Uframe number for the calibration program. Select the Uframe number for recording the measurement frame seen from the robot's world frame.

1 Target Uframe Number : 1

If the number of position is less than the minimum number of measurement point described in Subsection 60.3.4, following message will be displayed. Select “Yes” to continue as it is, select “No” to add some positions to the position program. If the number of position 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 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” .

60.4.5 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 attached and its location is properly set in the tool frame.
- Correct payload number is selected and its payload information is 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 60-4 “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 pause by PAUSE instruction at line 3. Measure the measurement reflector position by laser tracker at this timing. After one measurement, press the START button to move the robot to the next position.

	RECD_CAL_POS.TP	Description
1:	WAIT 2.00(sec)	Wait for robot complete stop
2:	CALL CB_PCDATA	Record internal data
3:	PAUSE	Measure by laser tracker

NOTE

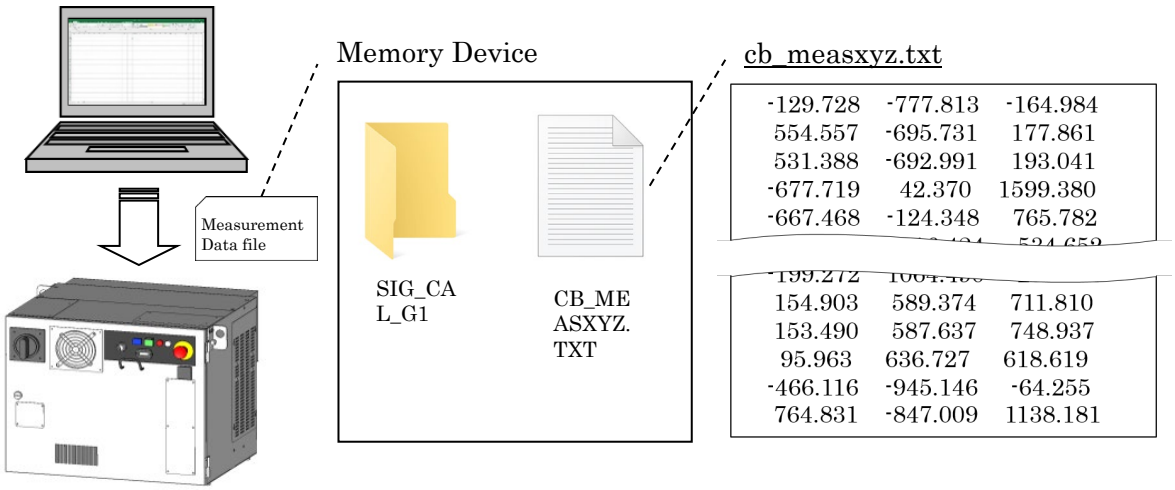
Recording internal data at line 2 and measurement by laser tracker at line 3 must be executed at same robot posture. ACAL-150 "Near Singularity, Skip Meas." could occur during the execution of line 2 when the measurement posture is around singularity position. In this case, recording internal data of 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 is running 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.



NOTE
Make sure to match the number of measurement data with the number of measurement points. 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.

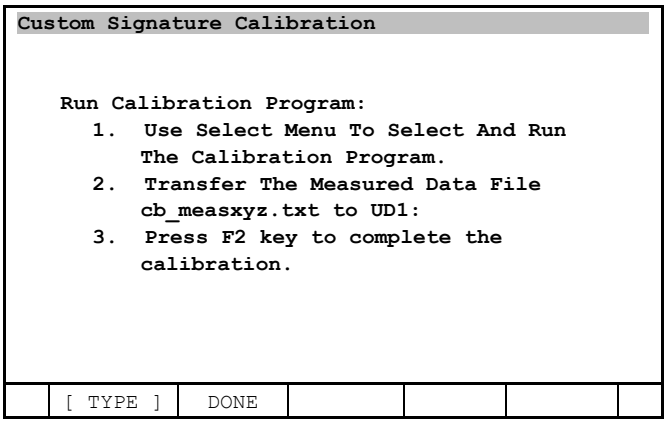
Procedure 60-5 Run Calibration TP Program

Condition

The status of "Create Calibration Prog" item is "DONE".

Step

1 Display the signature screen. (Procedure 60-1)
2 Move the cursor to the Run Calibration Program item and press the F4 "SELECT" key to display a screen shown in 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”.

60.4.6 Calculate Parameters Screen

On the calculate parameters screen, computes signature calibration data.

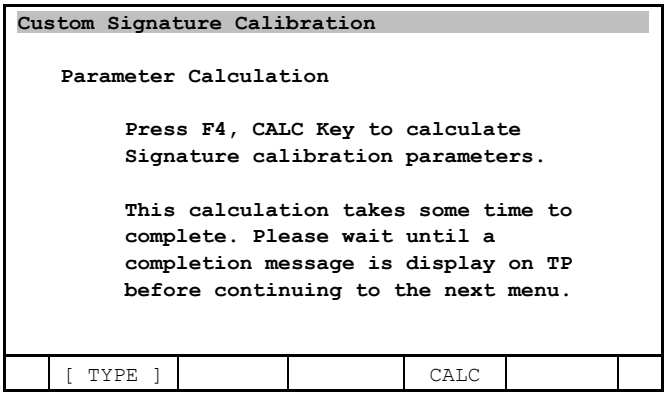
Procedure 60-6 Calculate Parameters

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 60-1)
2 Move the cursor to the Calculate Parameters item and press the F4 “SELECT” key to display a screen shown in 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”.

60.4.7 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 iteration to convergence. 10 or less than 10 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 60.5.2 for the detail.

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

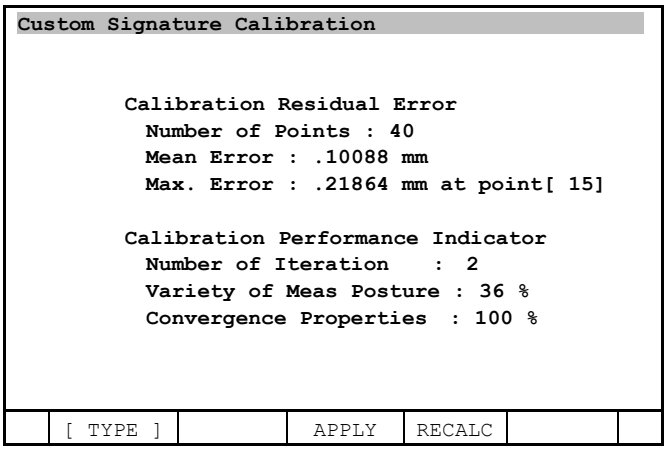
Procedure 60-7 View/Apply Calibration Result

Condition

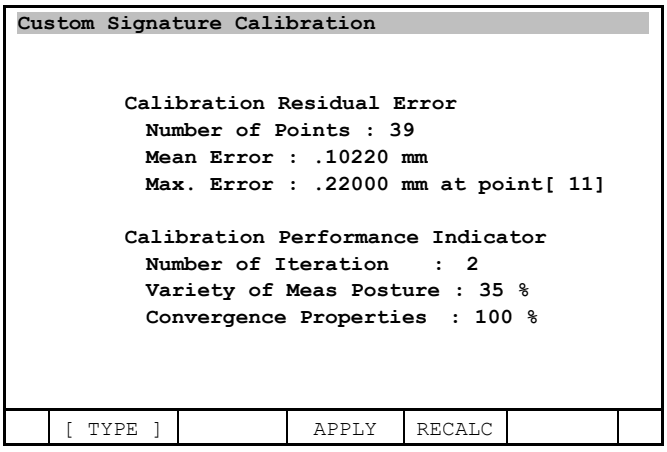
The status of "Calculate Parameters" item is "DONE".
Memory device including measurement data is inserted to the robot controller.

Step

1 Display the signature screen. (Procedure 60-1)
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 with removing measurement point where the maximum error is caused. 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 60-6 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.

60.4.8 Create Signature Calibration Parameters Setup File

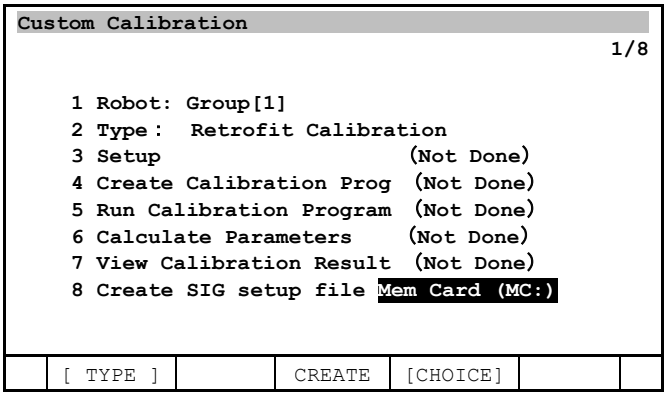
The signature calibration parameters is saved in a system file called SYSMAST.SV. In addition, it is possible to restore only the signature parameters only related by executing a command file. To create the command file, use the “Create SIG setup file” item in the signature menu of the retrofit calibration, 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 60-8 Create Signature Calibration Parameter File Setup File

Step

1 Display the signature screen. (Procedure 60-1)
2 Switch the calibration type to the retrofit calibration, the enhanced factory calibration or calibration recovery. (Procedure 60-2)
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.



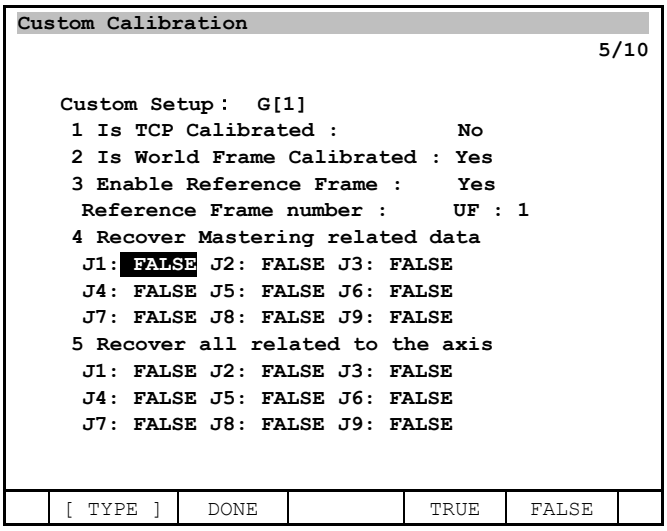
4 Press F3 “CREATE”. A message “MC:set_signature.cm is created” will be displayed when the file is successfully created.

60.5 Advanced Function

60.5.1 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 60.4.3, “Recover Mastering related Data” and “Recover all related axis” is selectable in each axis.



Please refer to the table below to setup the item to be recovered according to the parts been replaced. You can select multiple recovery items.

Selection of recovery items according to the replaced parts

Item	Description
Pulsecoder battery replacement Pulsecoder cable replacement	Perform Quick mastering. No need to perform the calibration recovery.
Pulsecoder replacement	Perform Jig position mastering first. In the calibration recovery, select “Recover mastering related data” for the axis number where parts replaced.
Secondary encoder’s sensor head detachment/replacement Secondary encoder’s scale detachment/replacement	
Motor replacement	
Reducer replacement	Perform Jig position mastering first. In the calibration recovery, select “Recover all related to the axis” for the axis number where parts replaced.
Robot arm replacement	
Wrist replacement	

Position Program

As described in Subsection 60.3.4, position program should include various postures that covers most motion range of all axes.

To accurately recover the mastering related parameters, the position program should include wide motion range of the axis to be recovered. For example, assuming J2 motor is 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 in order to compute an accurate tool frame.

Recovery of J1 signature calibration parameters

The world frame calibration must be done when recovering signature J1 calibration parameters after motor replacement or reducer replacement etc. Before falling into maintenance due to break, the measurement frame used for calibration recovery seen from the robot's world frame must be known by TCP/Uframe Calibration in advance and saved into user frame. 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 60.5.2 TCP/Uframe Calibration and 60.5.3 Reference Frame Function for the detail.

60.5.2 TCP/Uframe Calibration

TCP/Uframe Calibration computes robot installation positon and measurement reflector position mounted on the robot's tool. Fig 60.5.2(a) shows a schematic of TCP/Uframe Calibration.

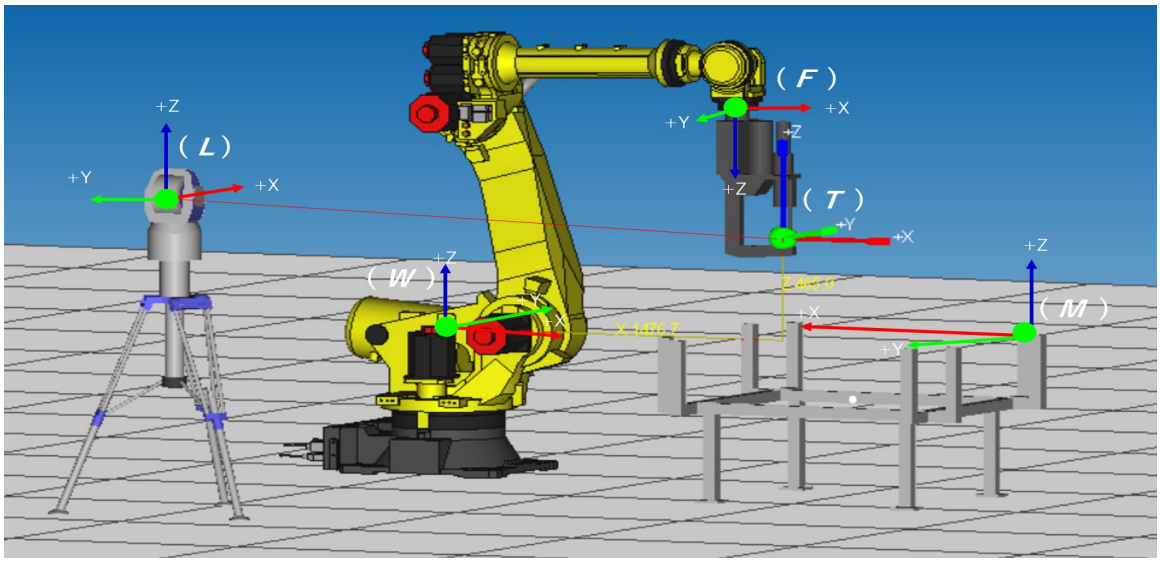


Fig 60.5.2(c) Schematic of TCP/Uframe Calibration

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.

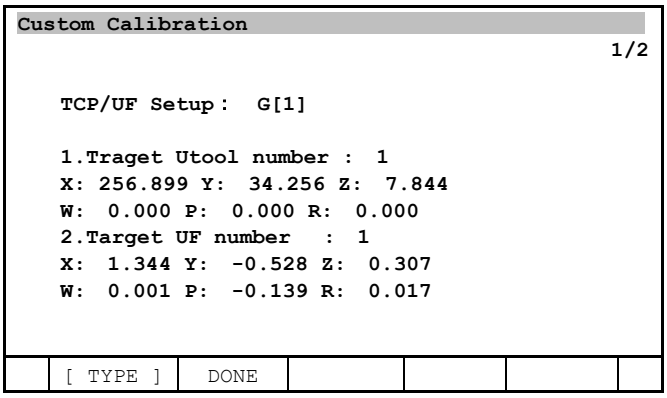
Setup screen and view calibration result screen for TCP/Uframe Calibration are different from another calibration type described in section 60.3 Basic Function.

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 60-9 Setup screen for TCP/UF Calibration

Procedure

1 Display the signature screen. (Procedure 60-1)
2 Switch the calibration type to the TCP/Uframe calibration. (Procedure 60-2)
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”.

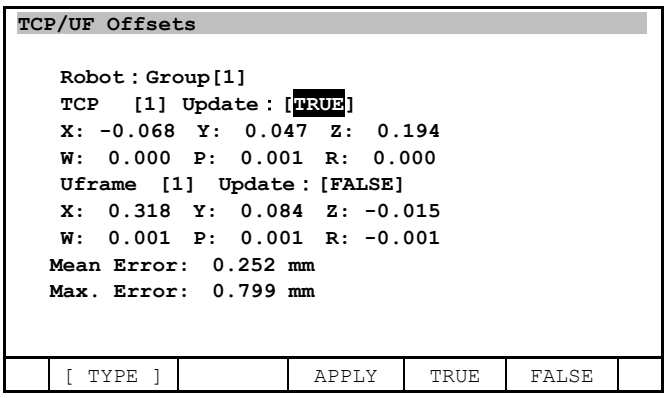
Procedure 60-10 Calibration Result Screen for TCP/Uframe Calibration

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 60-1)
2 Switch the calibration type to the TCP/Uframe calibration. (Procedure 60-2)
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, 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 is set to TRUE.

60.5.3 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 by setting the world frame calibration done.

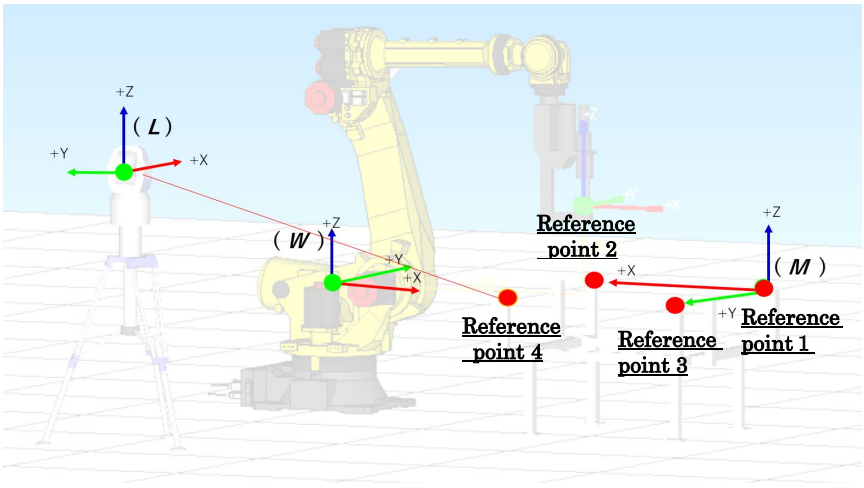


Fig 60.5.3(a) Schematic of the reference frame function

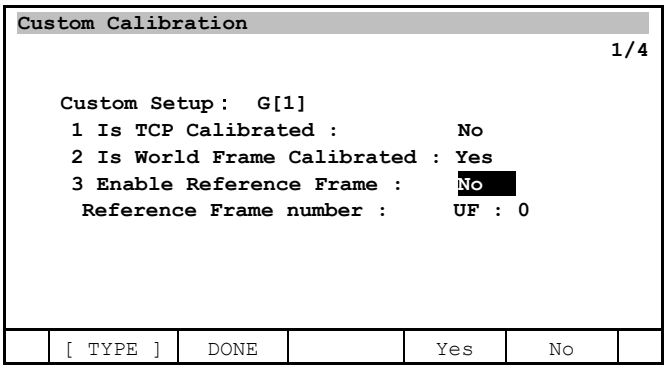
Procedure 60-11 Enabling Reference Frame Function

Condition

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

Step

1 Display the signature screen. (Procedure 60-1)
2 Switch the calibration type to the retrofit calibration, the enhanced factory calibration or calibration recovery. (Procedure 60-2)
3 Move the cursor to the Setup item and press F4 “SELECT” to display a setup screen shown in below.



3 Move the cursor to the Enable Reference Frame item and press F4 “Yes” to enable the reference frame function.
4 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.
5 In case “Is World Frame Calibrated” item is set to No, move the cursor to its item and press “Yes” to set world frame calibration done.

60.5.4 Enable/Disable iRCalibration Signature Compensation

For a given TP program position, the robot may reach a different physical position when it is signature compensated mode than when it is not in the signature compensated mode due to the applied compensation. If for some reason you need to put the robot in or out of 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 60-13.

60.6 iRCalibration Signature Status Screen

Signature Status Screen

This status screen shows current 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.

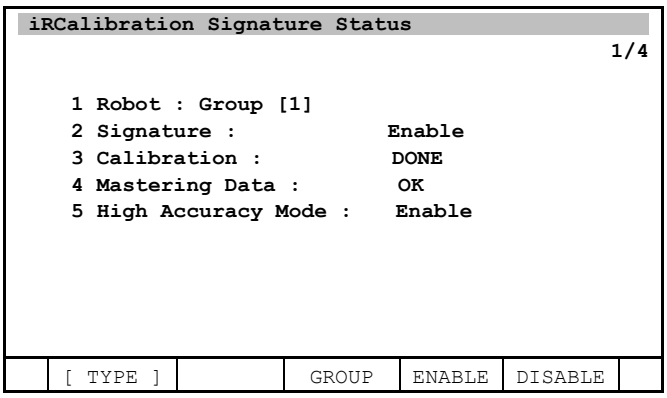
High Accuracy Mode

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.

Procedure 60-12 Display *i*RCalibration Signature Status Screen

Step

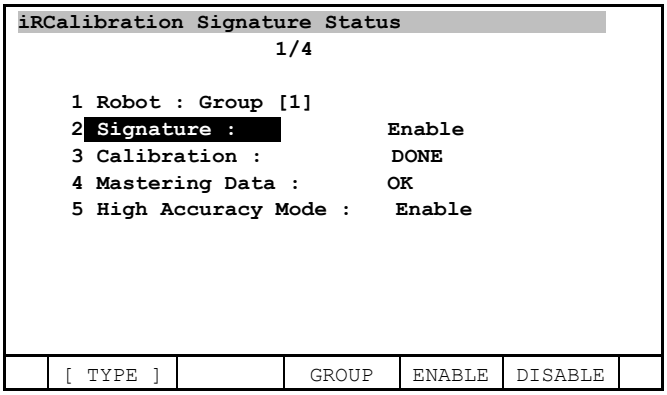
1 Press the [MENU] key to display the screen menu.
2 Select “0 NEXT → 4 STATUS”
3 Press F1, [TYPE] to display the screen selection menu.
4 Select Signature Status. Then the signature status screen will be displayed.



Procedure 60-13 Enable/Disable iRCalibration Signature

Step

1 Display *i*RCalibration Signature Status Screen. (Procedure 60-12)



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.

60.7 Mastering

60.7.1 Mastering for Signature Calibrated Robot

Our factory mastering is performed with disabled Signature compensation(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 put the robot out of signature compensated mode as well. Please refer to Procedure 60-13 to disable the Signature compensation.

NOTE
Attempting to master a robot when the robot is in signature compensated mode, a message will pop up to prompt you to put the robot out of signature compensated mode. Please follow the displayed message to disable the Signature compensation.

NOTE
When a signature calibration robot is not in signature compensated mode, all zero position marks match when the robot moves all its joints to zero degree.

After the mastering is correctly performed, please enable the signature compensated mode by Procedure 60-13 and validate accuracy of the robot. If accuracy is not acceptable, please refer to 60.5.2 to perform calibration recovery.

60.7.2 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 robot lost its pulse counts, it loses the number of revolutions value but keeps the pulse value within the revolution data. The 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.

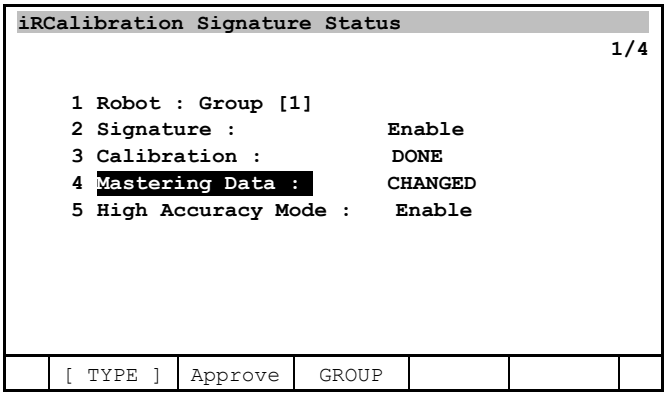
60.7.3 Monitoring the Change of Mastering Data

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 monitor the change of mastering data and notify with notification icon when mastering data is changed from the data at the signature calibration. If the change of mastering data is intentional, the change can be approved by procedure in below.

Procedure 60-14 Approving Mastering Data

Step

1 Display *i*RCalibration Signature Status Screen. (Procedure 60-12)



2 Set the cursor to the Mastering Data item and press F2 "Approve" to approve the change of mastering data. A pop up message as following is 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 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.

60.8 Cautions

60.8.1 Accuracy Around Singularity

In general, a robot in its singularity position means either the robot is fully extended or the robot's two or more axes are in a straight line .When a robot is close to its singularity position, the robot's accuracy is deteriorated comparing to 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 closed to robot's singularity. The following pictures shows the four singularity areas that should be avoided.

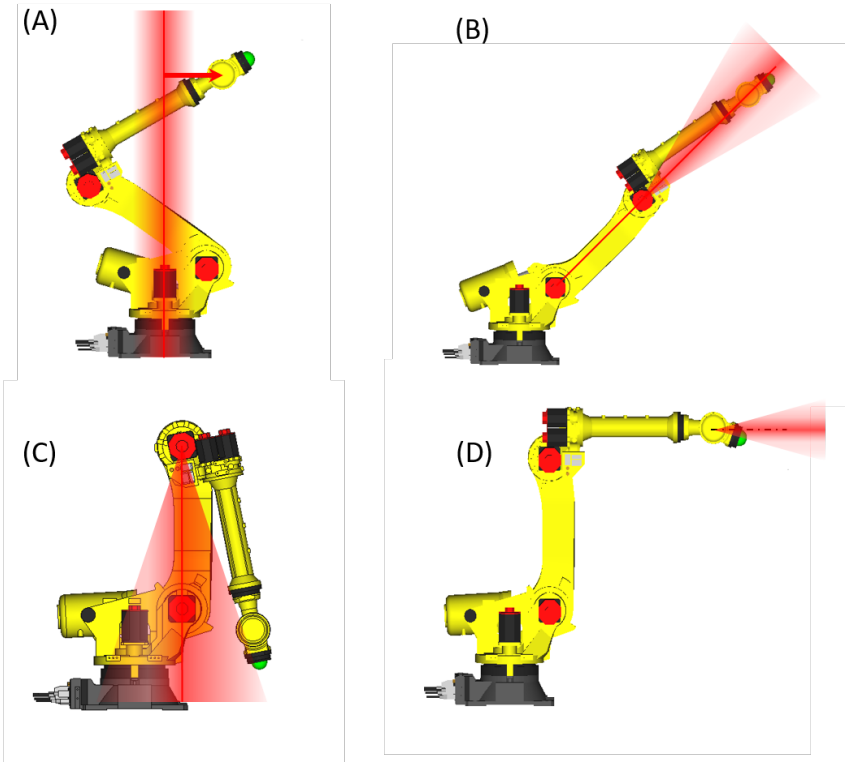


Fig 60.8.1(a) Example of posture around singularity

(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.

60.8.2 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.

61 NC PROGRAM CONVERSION

61.1 OVERVIEW

The NC Program Conversion feature allows you to load a robot program written in the NC program language (also known as G-code programs) into a robot controller, and convert it to the TP programs, thus enabling the robot to execute a user program that is written in the NC program language.
"NC Program Conversion (R775)" is necessary to use this function.

61.1.1 NC program syntax

As is always the case, an NC program is written specifically for the target machine tool. For example, an NC program for a lathe is different from that 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 is a feature that can only convert an NC program specifically written for a robot into a robot TP program.
One example is the position representation. The positions must be for 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).
The M-codes (miscellaneous codes) are, by definition, provided for each application. With limited configurability, the feature is able to convert a set of pre-defined M-codes into TP program instructions, or CALL to a sub-program that does the processing specific to that M-code. A list of supported M-codes is shown later.

61.1.2 Functional Specifications

The following is a list of specifications and limitations of this feature.

1. This is a feature used to convert an NC program that is written in ASCII format into 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 a name: "O0081"; for an NC program: "%O0081"). The source file name has to be the same as the program name, e.g. O0081.NC, etc.
3. There will be line-by-line correspondence between the source NC program and the output TP program with a few exceptions
   - Inline comments are not supported on a TP program.
   - The maximum length of a comment is limited by the TP program specifications, and will be truncated if too long (up to 32 characters for a comment line, up to 16 characters for a comment on a position, etc.).
4. The source NC program must be in compliance 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 program proceeds. No duplicate numbers are allowed
   - The N-code number does not have to be contiguous. Blank lines will be inserted in the output TP program if the number is not contiguous.
   - An N-code indicates a new line. In other words, one NC program instructions can spread over several lines (in 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 the O-code, which becomes the program comment.

6. The position data in the NC program must be in XYZWPR (or XYZWPRE for a 7axis robot) representation, with a config data, as is used in the robot programming. The format of config string is (N L U T,0,0,0). Please note a space between letters and commas are required within the config data. Refer to the ”4.3.2 Position Data” section in “OPERATOR’S MANUAL (Basic Function)” (B-83284EN) for the explanation of the 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 run-time 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 one previous motion instruction. The omission of position data is not allowed in the first motion instruction.
9. A “motion type” 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 TP program back to NC program is not supported.
11. Circular motion is not supported.
12. M-code instructions are implemented as a sub-program “Mnn,” with optional numerical parameters specified by the T-code. The sub-program 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 are supported up to three parameters.
13. An M-code can be specified as a stand-alone 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 binary format of a 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. The RAM drive size can be expanded to accommodate a large program by $FILE_MAXSEC. (The absolute value of $FILE_MAXSEC) x (512 byte) is the size of RD. If $FILE_MAXSEC is positive value then RD is in SRAM permanent pool, if it is negative value then RD is in DRAM.
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.

61.2 OPERATION

61.2.1 Conversion Operation

There are two ways to use the conversion feature.
The FILE menu allows you to load an NC program, converting it to a TP program. If you perform a load operation on an NC program file, it will convert the file into a TP program.
Another way to start conversion is to copy the NC program file into the MD device. The MD device is a virtual device, and it will automatically convert the NC program file into a TP program file.

61.2.1.1 Load NC program in FILE menu

Performing a LOAD operation within the FILE menu will start conversion of the NC program into a TP program. See following procedure for the operation.

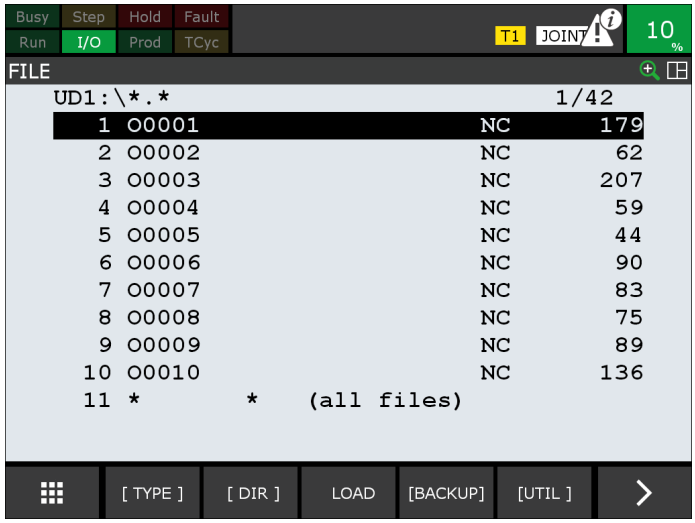
Procedure to load NC program

Conditions

An NC program file is listed on the FILE screen.

Step

1 In the FILE menu, display the contents of the device that contains the NC programs.



2 Set the cursor to the file you want to load, using the up or the down arrow keys.
3 Press the F3 (LOAD) key, followed by 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 in the program SELECT screen.

61.2.1.2 COPY NC program to MD drive

Performing the file Copy operation of an NC program to the MD drive will cause the system to convert the NC program into a TP program. The COPY operation can be performed either on the FILE menu, KCL command, or via the network using the FTP (File Transfer Protocol) operation.
Please note that the MD drive is a special drive that automatically performs the conversion operation. The source NC program itself will not be found in the MD drive after the copy operation.

61.3 NC PROGRAM SYNTAX

Due to the special nature of using Robot as the target machine, there are some modifications/additions to the standard NC programming language.
Please also note that the following describes the default setting of the system, as the syntax can be configurable.

61.3.1 O code – Program Name/Number

All programs must start the O-code to specify the program number.

Syntax:

Onnn

Explanation:

1. It must start with a letter O, followed by numbers. (up to 9999)

2. This program name (Onnnn) must match the file name.
3. A comment line (without N-code) immediately following O-code becomes the “program comment”. Maximum 16 characters.

Example:
%O6781
(first line comment becomes the program comment)

61.3.2 N code – Line numbers

Syntax:
Nnnn

Explanation:
At the start of the block in a program an 8 digit sequence number may be specified.

1. The line number is specified after “N” letter
2. The N-code must be the first instruction within each program line.
3. The numbers become the line number in the output TP program.
4. The number may skip. However, the blank lines are inserted in the robot TP program for those skipped lines. Therefore, it may be a good idea to use contiguous line numbers to avoid number of blank lines.

Example:
N001
N7975

61.3.3 G code – Motion modifiers

Syntax:
Gnnn

Explanation:
Only the following G-codes are supported in this implementation. G-code is usually followed by X, Y, Z, etc., because it is usually attached to a motion instruction. G-code is used to modify the motion instruction.

Code	Description	Modal	Explanation
G00	Rapid traverse mode	Yes	The machine moves at the fastest motion. The machine might not move on a straight line. It is converted to J (Joint) motype in the output TP program.
G01	Linear Interpolation	No	The machine moves on a straight line. This is the default setting. It is converted to L (Linear) motype in the output TP program.
G09	Exact Stop Check	No	Same as FINE term type on 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) Note: When the unit is changed, all positional values have to be specified in the next motion, or uninitialized position error will be posted.
G21	Positions in millimeters	Yes	Positional values are in millimeters. Note: When the unit is changed, the all positional values have to be specified in the next motion, or uninitialized position error will be posted.

Code	Description	Modal	Explanation
G61	Exact Stop Check (Modal)	Yes	Same as FINE term type on robot. Unlike G09, the effect remains until it is cancelled by G64.
G64	Cancels G61	Yes	This cancels the effect of G61, and set the following motions to regular “Continuous” term type. It is converted to CNT100 termtype in the output TP program.
G90	Absolute programming	Yes	This is the default setting. The positional value is absolute. When this setting is changed, the all positional values have to be specified in the next motion, or uninitialized position error will be posted.
G91	Incremental programming	Yes	The positional value is in reference to the previous position (translational moves only). When this setting is changed, the all positional values have to be specified in the next motion, or uninitialized position error will be posted.

61.3.4 F code – Feed Rate / Speed

F-code is to specify the speed of the robot.

Syntax:

Fnnn

Explanation:

For Linear motions (M210):

1. When G21 (positions in millimeters) is effective: The unit is in mm/min. In the TP program, the unit is converted to cm/min (the value is rounded to nearest integer).
2. When G20 (positions in inches) is effective: The unit is in inch/min. In the TP program, the unit is in inch/min (no conversion of the value).

For Joint motions (M211): The unit is 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 specified in the system variable is used.

61.3.5 Position data

Syntax:

Xnnn - value of 1st component of a position (X value)
Ynnn - value of 2nd component of a position (Y value)
Znnn - value of 3rd component of a position (Z value)
Wnnn - value of 4th component of a position (W value)
Pnnn - value of 5th component of a position (P value)
Rnnn - value of 6th component of a position (R value)
Hnnn - value of J7 axis (if applicable)
Innn - value of J8 axis (if applicable)
Jnnn - value of J9 axis (if applicable)

Explanation:

1. Refer to the ”Positional Information” section of your application-specific *Setup and Operations Manual* for the explanation of the position representation.
2. All the positional data have to be specified in the first motion of a program, or “Uninitialized position” error will be posted.
3. From the second motion and after, some position components can be omitted. The previous value will be used for the omitted components.

61.3.6 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 within the parentheses. Refer to the application specific *Setup and Operations Manual* for the meaning of the config string.
2. If 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)

61.3.7 M code – Miscellaneous codes

M-codes are generally executed by calling a TP sub-program. Because the M-codes can be defined by the user depending on the machine tool being used, the TP sub-program is to be provided and customized by the Machine Tool Builder or the user. Some of the 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 optional T-code with numbers.

Run-time processing:

1. For some pre-defined M-codes, the equivalent TP instruction will be generated.
2. Other M-codes will be converted to a CALL instruction to a specific TP sub-program.
   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 run-time.
   In this example, the TP program “Mpp” has the program name with M-code number as part of it. The T-code number (qq) is passed as the argument for the program.
3. Since the M-codes are to 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 pre-defined M-codes.
4. The number portions of the T-code are passed to the TP sub-program as the argument. If no T-code is specified, no argument will be passed.

> ⚠ **CAUTION**
> You can only specify up to one M-code and up to three T-code within one line (block) of the program instruction.

61.3.7.1 Pre-defined M-codes

The following M-codes are pre-defined, and therefore the user may not change its meaning.

Code	Description	Modal	Explanation
M00	Compulsory stop	No	It is converted to PAUSE TP instruction.
M01	Optional stop	No	Machine will only stop at M01 if operator has pushed the optional stop button. (generates “CALL M01” instruction)
M02	End of program	No	It is converted to ABORT TP instruction.
M30	End of program with return to program top	No	It is converted to END TP instruction.

61.3.7.2 Special M-codes

There are some M-codes that are specific to the robot application, or require special processing during the program conversion. Please note the M-code number may be different depending on the system configuration, and therefore, these M-code numbers are configurable (by the system variable setting).

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 in Linear mo-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 in Joint mo-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
“Offset” clause could be either OFFSET, VOFFSET, TOOL_OFFSET, depending on the configuration specified.

Other M-codes may be defined by the Machine Tool Builder or the user, by creating the corresponding TP sub-programs.

61.3.7.3 Predefined M-codes

The default conversion for the pre-defined M-codes are as follows.

M00 (compulsory stop)

The M-code “M00” will be converted to the PAUSE command in TP program.

Example: TP instruction for “M00” M-code

```
PAUSE
```

The functionality of 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 the ABORT command in 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 the END command in 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, END instruction works as the RETURN instruction to go back to the caller program and continue. Please note the functionality of END statement may not be exactly equivalent to M30 code in the typical NC machines.

M221 (User Frame Selection)

Syntax:

M221Tnn - Selection of User Frame number

Explanation:

It generates “UFRAME_NUM=nn” instruction.

It is not a standard M-code definition. This is a new M-code specifically for robot, due to the requirement for the way how positions are specified on robots. The functionality of M221 is as follows.

- For program convertor – This will set the user frame number for all the positions thereafter (user frame number is part of the position data)
- For program execution – This will generate a code to set the user frame number at run-time

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 User Tool Frame number

Explanation:
It generates “UTOOL_NUM=nn” instruction.

It is not a standard M-code definition. This is a new M-code specifically for robot, due to the requirement for the way 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 the positions thereafter (user tool number is part of the position data)
- For program execution – This will generate a code to set the user tool number for run-time

Example: TP instruction for “M222” M-code
```
utool_num = nn
```

The TOOL number should be specified by a T-code that follows.

61.3.7.4 Other (user defined) M-codes

M-codes with no associated T-code
Because such an M-code has no T-code associated, the TP sub-program will be called with no argument.

Example: TP instruction for “M25” M-code
CALL M25

Because the functionality of M25 is machine dependent, the TP program M25 will have to be provided by the Machine Tool Builder. In many cases, the program will communicate with the peripheral equipment via process I/O interface.

Please note 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-code
The TP sub-program for such an M-code will be called with the argument that is specified as T-code. Up to three argument may 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 M56 will have to be defined by the Machine Tool Builder. In most cases, the program will communicate with the peripheral equipment via process I/O interface.

61.4 SYSTEM VARIABLES

61.4.1 $NC_SYTX_TBL

This variable is to define command letters of NC language with their meaning.

- $NC_SYTX_TBL[].$nc_inst_str : STRING - To hold the command letter for the NC instruction
- $NC_SYTX_TBL[].$nc_inst_sfx : STRING - Optional suffix character for NC instruction
- $NC_SYTX_TBL[].$nc_inst_typ : INTEGER - Instruction type for the NC instruction
- $NC_SYTX_TBL[].$nc_elem_idx : INTEGER - To specify the field index within the pos data structure._For example, X is the first index (0), Y being the next index (1), etc.

The meaning of $NC_SYTX_TBL[].$nc_inst_typ is as follows.
1: O-code
2: N-code
3: comment line
4: M-code, further classified in M-code syntax table
5: G-code, further classified in G-code syntax table
6: T-code, as an argument for M-code
7: F-code, feed rate
8: Config string (K-code)
9: position field (X, Y, Z, W, P, R or like)

For example, the following will define the command "Y" to indicate the position value for the 2nd positional field.
- $NC_SYTX_TBL[1].$nc_inst_str = "Y"
- $NC_SYTX_TBL[1].$nc_inst_sfx = ""
- $NC_SYTX_TBL[1].$nc_inst_typ = 9
- $NC_SYTX_TBL[1].$nc_elem_idx = 1

61.4.2 $NC_GCOD_TBL : To define G-code numbers with their meaning.

- $NC_GCOD_TBL[].$nc_gvalue : STRING – To hold the numerical value for a G-code
- $NC_GCOD_TBL[].$nc_gtype : INTEGER - Code type for this G-code
- $NC_GCOD_TBL[].$nc_gmodal : BOOLEAN - Whether this G-code is modal or not

The meaning of $NC_GCOD_TBL[].$nc_gtype is as follows.
1: Rapid traverse motion (default: G00)
2: Linear interpolation (default: G01)
3: Exact stop check (default: G09)
4: Positions in inches (default: G20)
5: Positions in millimeters (default: G21)
6: Exact stop check (modal) (default: G61)
7: Cancels FINEMODE (default: G64)
8: Absolute programming (default: G90)
9: Incremental programming (default: G91)
10: Reposition origin point (default: G92)
11: Preset origin point (default: G92.1)

For example, the following defines the G09 code.
- $NC_GCOD_TBL[1].$nc_gvalue = "9"
- $NC_GCOD_TBL[1].$nc_gtype = 3
- $NC_GCOD_TBL[1].$nc_gmodal = FALSE

61.4.3 $NC_MCOD_TBL : To define M-code numbers with their meaning.

- $NC_MCOD_TBL[].$nc_mvalue : STRING - Numerical value for an M-code
- $NC_MCOD_TBL[].$nc_mtype : INTEGER - Instruction type for this M-code
- $NC_MCOD_TBL[].$nc_call_sub : BOOLEAN - Whether sub-prog is called
- $NC_MCOD_TBL[].$nc_ofst_typ : specifies the type of offset (bit mask)
  1: OFFSET
  2: TOOL_OFFSET

4: VOFFSET
- $NC_MCOD_TBL[].$nc_mmodal : BOOLEAN - Whether this M-code is modal or not

The meaning of $NC_MCOD_TBL[].$nc_mtype is as follows.
1: PAUSE : for M00
2: ABORT : for M02
3: END : for M30
4: UFRAME : for M221
5: UTOOL : for M222
6: LIN : Enable Linear motion M210
7: JNT : Enable Joint motion M211
8: OFST : Enable offset (type of offset specified by $NC_MCOD_TBL[].$nc_ofst_typ)
9: CLR : Clear offset

For example, the following will define M31 (Vision resync) command:
- $NC_MCOD_TBL[1].$nc_mvalue = “31”
- $NC_MCOD_TBL[1].$nc_mtype = 8
- $NC_MCOD_TBL[1].$nc_call_sub = TRUE
- $NC_MCOD_TBL[1].$nc_ofst_typ = 1 (offset)
- $NC_MCOD_TBL[1].$nc_mmodal = TRUE

61.4.4 $NC_DEFMODAL

This variable is to define the default value of the modal values, and is used every time a new NC program is being read.

The following is the default value of this variable.
- $NC_DEFMODAL.$INST.$posid = 0
- $NC_DEFMODAL.$INST.$motype = 1 (Linear)
- $NC_DEFMODAL.$INST.$termtype = 1 (CNT)
- $NC_DEFMODAL.$INST.$termpct = 100 (100%)
- $NC_DEFMODAL.$INST.$spdunit = 0 (mm/min)
- $NC_DEFMODAL.$INST.$lin_speed = 6000 (6000 mm/min = 100 mm/sec)
- $NC_DEFMODAL.$INST.$jnt_speed = 100 (100%)
- $NC_DEFMODAL.$INST.$incremental = 0 (false)
- $NC_DEFMODAL.$INST.$use_offset = 0 (bit mask)
- $NC_DEFMODAL.$POS.$posid = 0
- $NC_DEFMODAL.$POS.$uframenum = 0($uframe_num)
- $NC_DEFMODAL.$POS.$utoolnum = 1 ($utool_num)
- $NC_DEFMODAL.$POS.$numposelem = 6 (number of pos data elements)
- $NC_DEFMODAL.$POS.$posrep = 0 (XYZWPR)
- $NC_DEFMODAL.$POS.$posunit = 0 (mm)
- $NC_DEFMODAL.$POS.$comment = ""
- $NC_DEFMODAL.$POS.$config = "N L U T, 0, 0, 0 "
- $NC_DEFMODAL.$POS.$pos1_value = uninit
- $NC_DEFMODAL.$POS.$pos2_value = uninit
- $NC_DEFMODAL.$POS.$pos3_value = uninit
- $NC_DEFMODAL.$POS.$pos4_value = uninit
- $NC_DEFMODAL.$POS.$pos5_value = uninit
- $NC_DEFMODAL.$POS.$pos6_value = uninit
- $NC_DEFMODAL.$POS.$pos7_value = uninit
- $NC_DEFMODAL.$POS.$pos8_value = uninit
- $NC_DEFMODAL.$POS.$pos9_value = uninit
- $NC_DEFMODAL.$ATTR.$protect = 1
- $NC_DEFMODAL.$ATTR.$stacksize = 0

- $NC_DEFMODAL.$ATTR.$priority = 50
- $NC_DEFMODAL.$ATTR.$timeslice = 0
- $NC_DEFMODAL.$ATTR.$busylampoff = 0
- $NC_DEFMODAL.$ATTR.$abortreq = 0
- $NC_DEFMODAL.$ATTR.$pausereq = 0
- $NC_DEFMODAL.$ATTR.$defaultgrp = 1
- $NC_DEFMODAL.$ATTR.$controlcode = 0

61.4.5 $NC_OFST_CMD

This variable indicates the command string to be used for Offset command (when the mask 1 is set in $NC_MCOD_TBL[].$nc_ofst_typ system variable)

Deafult value: “OFFSET”

For example, when the vision sensor is to be used, the motion instruction will add the specified Offset clause to its statements after a vision resync command (e.g. M31) is executed until a vision clear command (e.g. M35) is executed. The user can specify the actual offset clause in a system variable ($NC_OFST_CMD) as a string data (The conversion software just copies the string as the “offset” clause).

61.4.6 $NC_TOFS_CMD

This variable indicates the command string to be used for Tool_Offset command (when the mask 2 is set in $NC_MCOD_TBL[].$nc_ofst_typ system variable)

Deafult value: “TOOL_OFFSET”

For example, if the proximity sensor is enabled, the motion instruction will add the specified TOOL_OFFSET clause to its statements after a sensor resync command (e.g. M39/M60/M61) is executed until a sensor clear command (e.g. M26) is executed. The user can specify the actual tool_offset clause in a system variable ($NC_TOFST_CMD) as a string data (The conversion software just copies the string as the “tool_offset” clause).

61.4.7 $NC_VOFS_CMD : Command string to be used for Vision offset command

This variable indicates the command string to be used for Vision Offset command (when the mask 4 is set in $NC_MCOD_TBL[].$nc_ofst_typ system variable)

Deafult value: uninitialized

61.4.8 $NC_DEST_DRV

The device name for the intermediate file. This value should be always “MD”

61.5 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
```

62 FANUC FIBER LASER INTERFACE

62.1 OVERVIEW

This is the function to control fiber laser oscillator made by FANUC from robot.

62.1.1 Necessary Software Option

The application tool supported by this function is only Handling Tool or LR Handling Tool.

Table 62.1.1 (a) Application tool of this function

Name	Drawing number
Handling Tool	A05B-2600-H552
LR Handling Tool	A05B-2680-H551

The following software option is necessary to use this function.

Table 62.1.1 (b) Necessary software option

Name	Drawing number
FANUC Fiber Laser Interface（This Function）	A05B-2600-J922
EtherNet/IP scanner	A05B-2600-R785

62.1.2 Caution

62.1.2.1 Restriction of the number of group

⚠ WARNING
As a principle, this function cannot be used in the system of more than three groups. If you would like to use this function in the system of more than three groups, please be sure to contact your local FANUC representative.

62.1.2.2 For the safety

If the teach pendant is enabled, this function turns “Interlock OFF rq” signal OFF and takes interlocks beam power.

NOTE
In addition, please refer to Beam interlock OFF request signal in Subsection 62.5.1.1 “Laser Oscillator Output Screen”.

⚠ WARNING
Class 4 laser is applied. If you use this function, it is necessary to prepare the interlock with sufficient consideration for user’s safety.

62.1.2.3 For multi-task

NOTE
This function is not compatible with multitasking (Laser processing cannot be performed simultaneously by multiple programs.).

62.2 CONNECTION OF LASER AND ROBOT

This function controls laser oscillator by I/O and data communication on Ethernet.

- EtherNet/IP is used for I/O communication.
- Ethernet is used for data communication.
- It is possible to connect at most six robots to one laser oscillator.
- The basic configuration of this function is as follows.

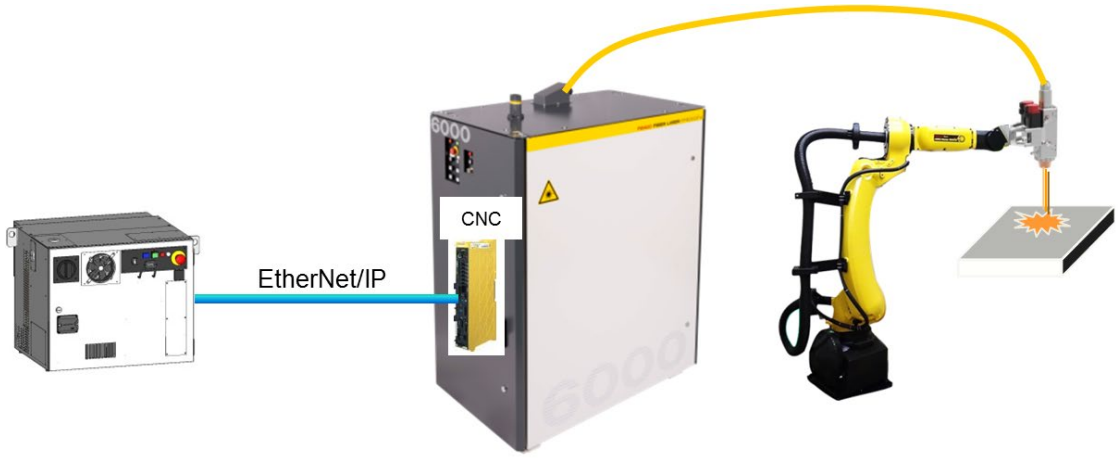


Fig. 62.2 (a) Basic configuration of laser oscillator and robot

62.2.1 Ethernet cable connection to robot

This function controls laser oscillator by I/O and data communication on Ethernet. Please connect Ethernet cable to Ethernet connecter (CD38B) of the main board on the robot controller.

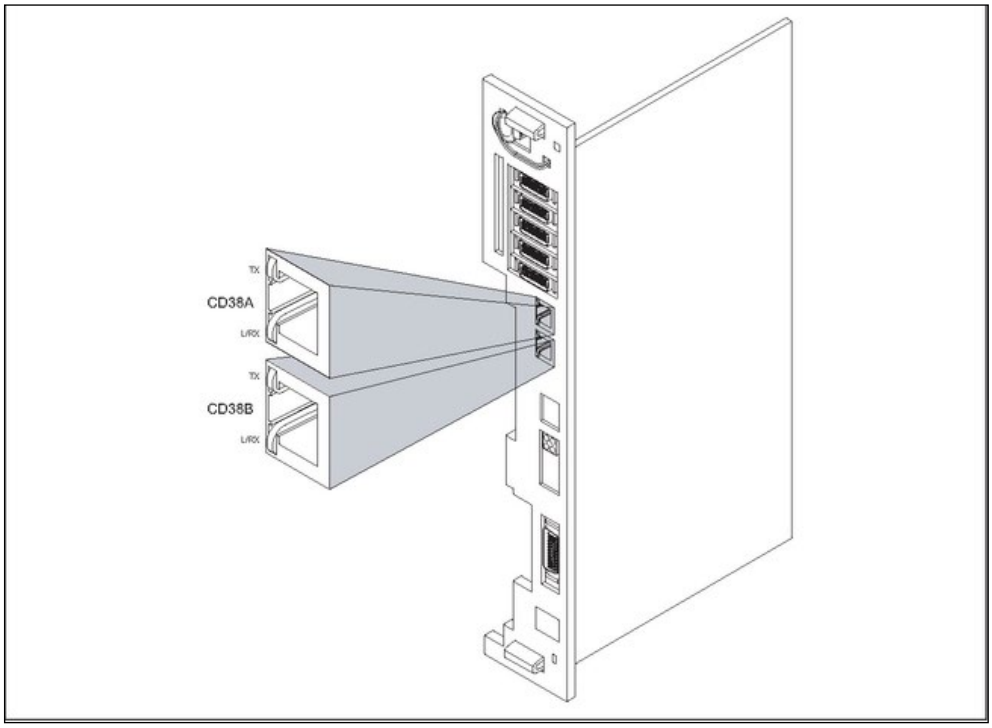
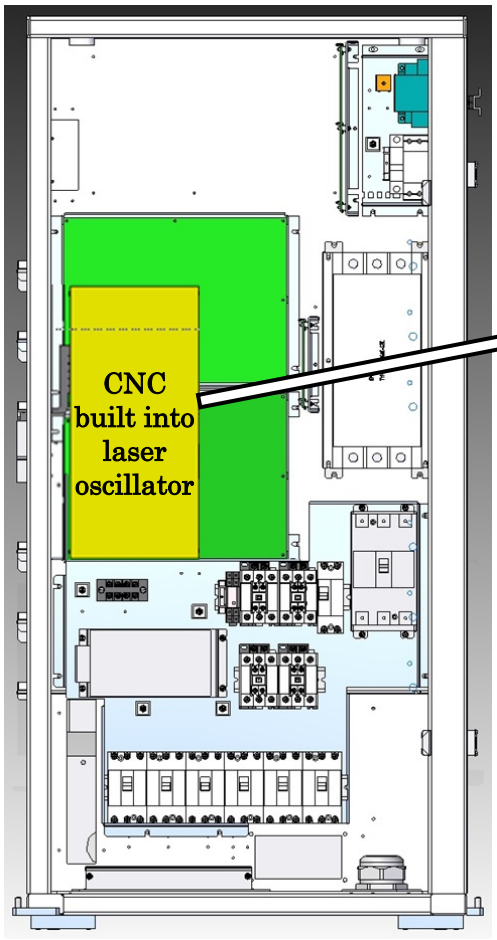
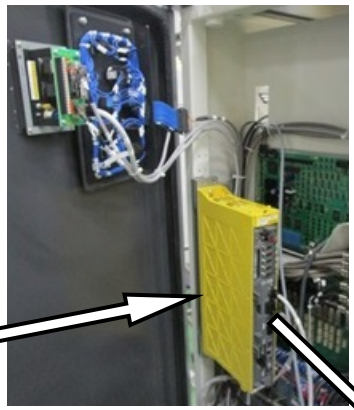


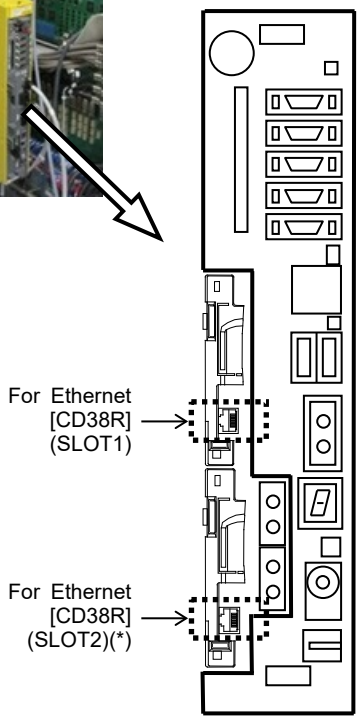
Fig. 62.2.1 (a) Ethernet connecter on main board

62.2.2 Ethernet Cable Connection to Laser Oscillator

Ethernet cable is connected from CD38B of robot to CD38R of CNC built into laser oscillator. (Robots from 1st to 3rd are connected to SLOT1, and robots from 4th to 6th are connected to SLOT2.) Please refer to Fig. 62.2.2 (a) about features of Ethernet connector.
Ethernet cable is connected directly or through the Ethernet hub with Ethernet connector.
If a robot is connected, Ethernet cable (Cross) must be used.
If two or more robots are connected, Ethernet cable (Cross) must be used.
Please refer to Fig. 62.2.2 (b) about method of the connection.







Fiber laser oscillator (input unit)

Main board of CNC built into laser oscillator

(*) When the robots are three or less, there is no Ethernet connector SLOT2.

Fig. 62.2.2(a) Ethernet connecter for fiber laser oscillator

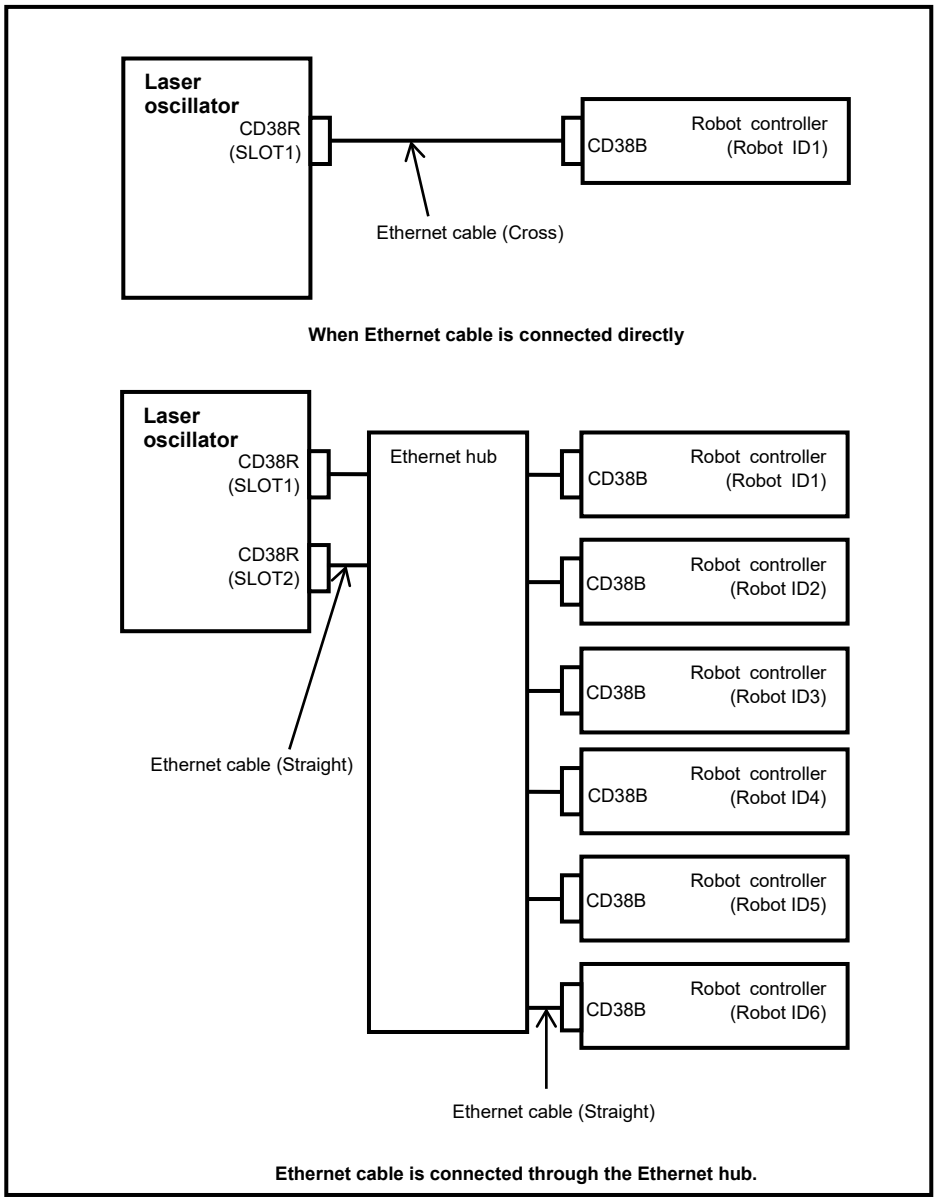


Fig. 62.2.2(b) Method of connecting from robot to laser oscillator

62.2.3 I/O and Data Communication Setting

For controlling laser oscillator with this function, I/O and data communication with laser oscillator needs to be established.

62.2.3.1 Robot ID setting (in case of connecting several robots)

It is necessary to set robot ID before setting of I/O and data communication with laser oscillator.
If fiber selector is installed to laser oscillator and is connected with several robots, robot ID is required to classify each robot from laser oscillator.
For robot ID, please set same number as CH (channels) of laser control request and shutter number used for this robot.

NOTE
In case of connecting several robots, enable the time sharing control on the laser initial setup screen.

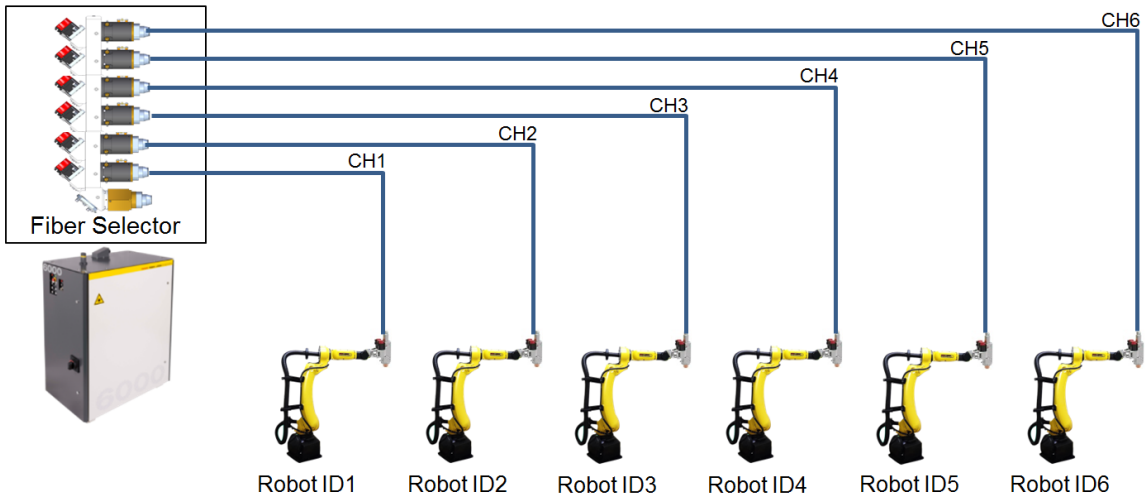


Fig. 62.2.3.1 (a) Definition of robot ID

If laser oscillator is connected with one robot, robot ID is always 1.

Procedure 62-1 Change robot ID

It is possible to change robot ID on laser initial setup screen of controlled start menu after controlled start. To change robot ID, it is necessary to change before communication setting with laser oscillator.

Step

1 Perform controlled start.
2 After controlled start, press the [MENU] key. Controlled start menu will be displayed.
3 Select [0 – NEXT --]. Select [4 Laser Initial Setup].

```
Laser Initialization                                  1/10

  1  Laser Oscillator Comm:                      INCOMP
  2  Laser Oscillator Model:                   FF6000i-A
  3  Laser robot group number:                         1
  4  My robot ID:                                      1
  5  Time sharing control:                       DISABLE
  6  CH# Laser Ctrl rq switching:                DISABLE
  7  Processing speed unit:                       in/min
  8  Processing speed default:                         0
  9  Stop/NonStop on LE instr:                   NonStop
 10  Direct power override mode:                 DISABLE

[ TYPE ]                          [ SETUP ]
```

4 Move cursor on “Robot ID” and change robot ID. Set robot ID to the same number as CH (channels) of laser control request and shutter number which is used in this robot.
5 In case of connecting several robots, enable the time sharing control on this screen.

62.2.3.2 Communication setting

To establish I/O and data communication with laser oscillator, it is necessary to set communication.

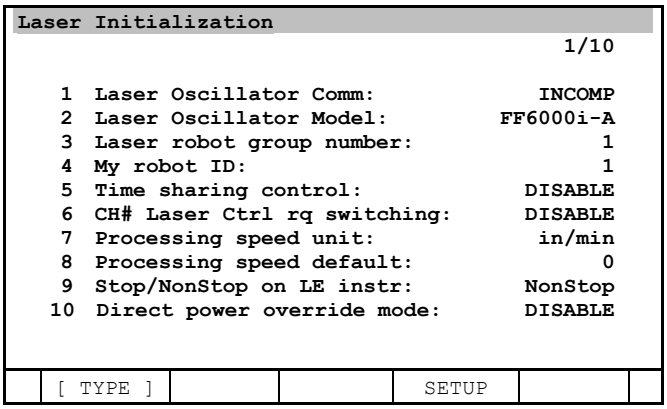
NOTE
Normally, EtherNet/IP of this function uses slot 1. Do not assign I/O which is used for other device to slot 1.

Procedure 62-2 Communication Setting

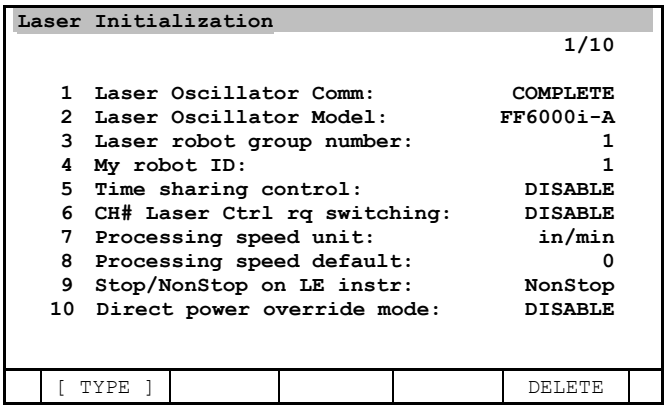
It is possible to set communication on laser initial setup screen of controlled start menu after controlled start.

Step

1 Perform controlled start.
2 After controlled start, press the [MENU] key. Controlled start menu will be displayed.
3 Select [0 – NEXT --]. Select [4 Laser Initial Setup].



4 Move cursor on “Laser Oscillator Comm” and press F4 [SETUP]. If communication setting is completed, “Laser Oscillator Comm” is changed to “COMPLETE”.



5 If you would like to delete communication setting, please press F5[DELETE]. After deleting, “Laser Oscillator Comm” is changed to “INCOMP”.

NOTE
Be sure to perform communication setting and delete it on laser initial setup screen.

6 Perform Cold Start.

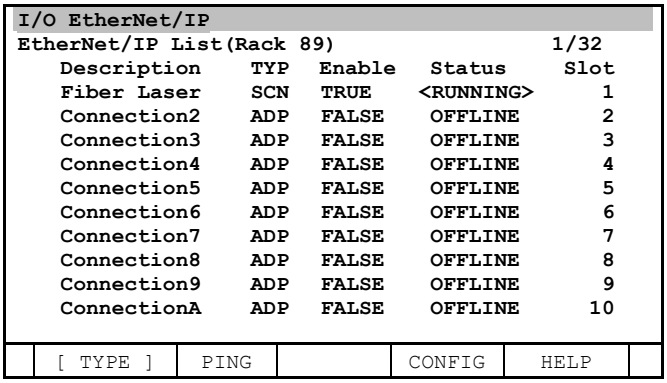
62.2.3.3 Confirm communication setting

It is possible to confirm communication setting on EtherNet/IP status screen, client setup screen and host communication setup screen.

Procedure 62-3 Confirm EtherNet/IP status

Step

1 Press the [MENU] key then screen menu will be displayed.
2 Select [5 I/O].
3 Show Menu by pressed F1 [TYPE].
4 Select “EtherNet/IP”.



5 If communication setting is properly performed, “Fiber Laser” is set to comment of slot 1.

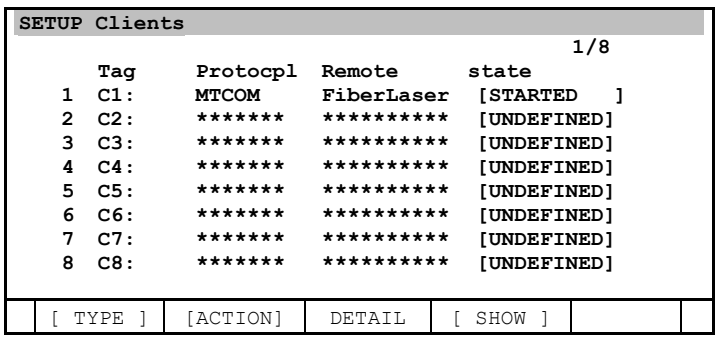
NOTE
Be sure to perform communication setting and delete it on laser initial setup screen.

NOTE
Normally, EtherNet/IP of this function uses slot 1. Do not assign I/O which is used for other device to slot 1.

Procedure 62-4 Confirm client tag setting

Step

1 Press the [MENU] key then screen menu will be displayed.
2 Select [6 SETUP].
3 Press F1 [TYPE].
4 Select “Host Comm”.
5 Press F4 [SHOW] and select [2 Clients].



6 If communication setting is properly performed, “Fiber Laser” is set to any one port of C1: to C8:.

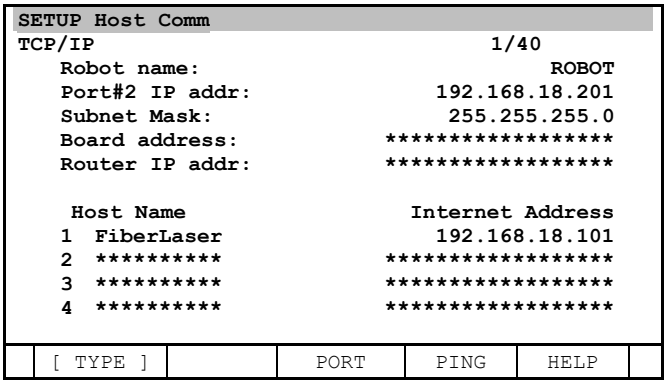
NOTE
Be sure to perform communication setting and delete it on laser initial setup screen.

NOTE
Client tag of this function is automatically set after finding tag whose status is undefined.

Procedure 62-5 Confirm host setting

Step

1 Press the [MENU] key then screen menu will be displayed.
2 Select [6 SETUP].
3 Show Menu by pressed F1 [TYPE].
4 Select “Host Comm”.
5 Press F3 [DETAIL] then host communication setup screen will be displayed..
6 Press F3 [PORT] then IP address of port #2 will be displayed.



7 Confirm whether IP address which is used in this function is set to port #2 IP address. IP address which is set will vary depending on robot ID.

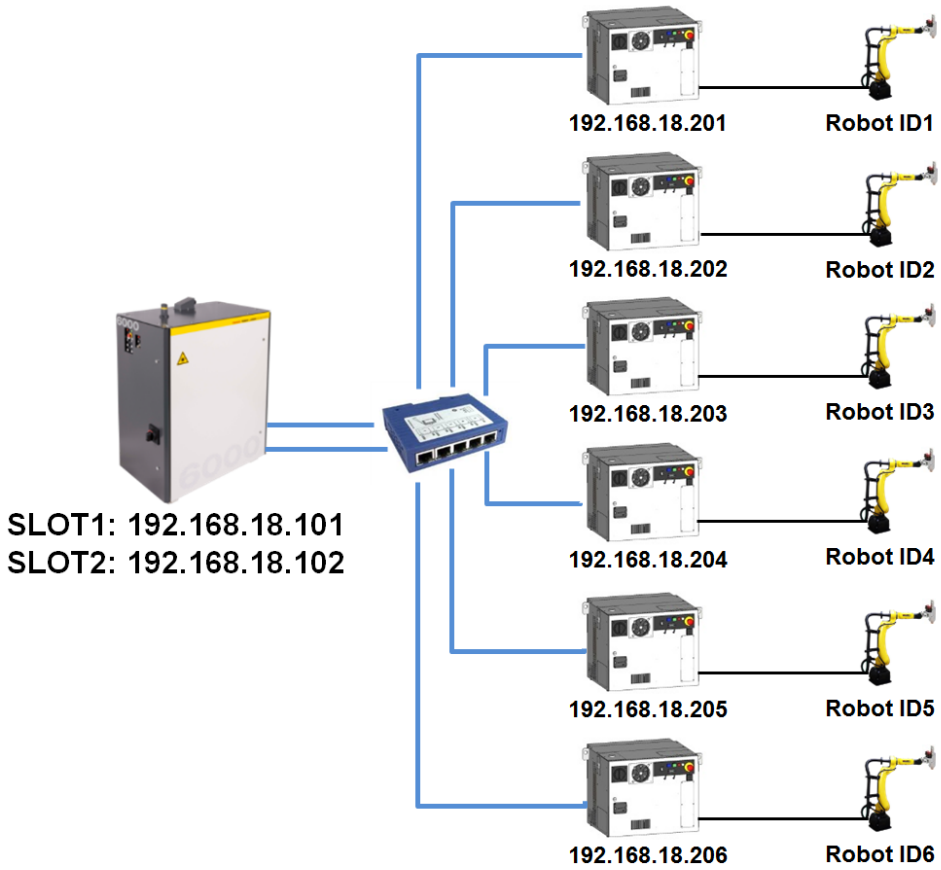


Fig. 62.2.3.3 (a) Relationship between IP address and robot ID

8 If communication setting is properly performed, “Fiber Laser” is set to “Host Name”. IP address of laser oscillator side is set to “Internet Address”.

NOTE
Be sure to perform communication setting and delete it on laser initial setup screen.

NOTE
Even if you delete communication setting, IP address of port #2 is not deleted. If you would like to delete it, please delete manually.

NOTE
Host setting of this function is automatically set after finding line whose column of “Host Name” is blank.

62.2.3.4 Communication setting on the laser oscillator side

This communication setting is completed at the factory shipment, so there is no need to change it.
For details on how to check the above setting, refer to the subsection “Confirm Communication Setting for Laser Oscillator” below.

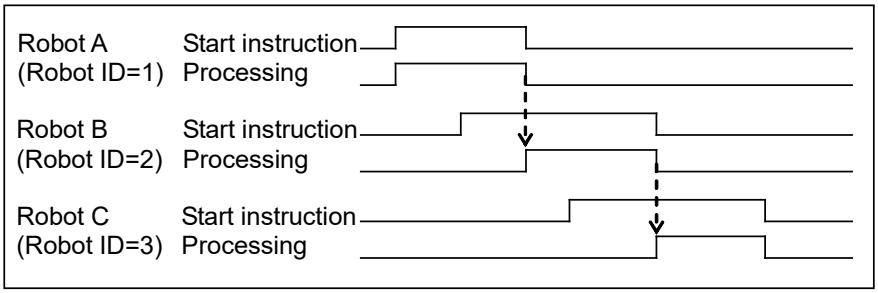
62.3 SETTING OF LASER

This function has three settings. These are laser initial setup, laser setup and time sharing control setup.

62.3.1 Laser Initial Setup

Laser initial setup is the setting which is the base of this function. For safety, you can only change laser initial setup on the laser initial setup screen of the controlled start menu. After the normal start, you cannot change the laser initial setup, you can check only the setting state.

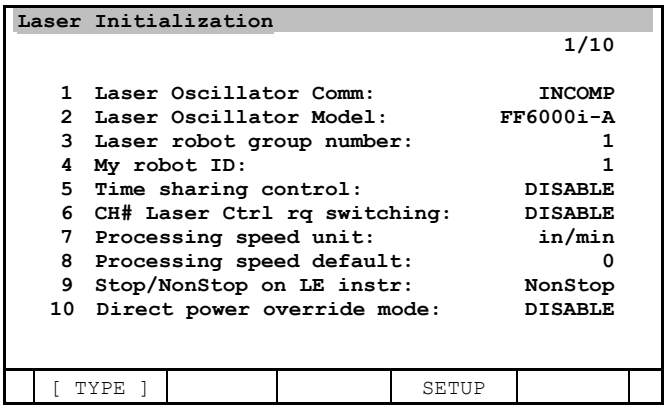
Table 62.3.1 (a) Laser initial setup screen

Setting item	Explanation
Laser Oscillator Comm	Communication with laser oscillator is set. In addition, please refer to Subsection 62.2.3.2 “Communication Setting”.
Laser Oscillator Model	Please select model of laser oscillator which is used. Maximum and minimum value of peak power and pulse bottom power by this setting.
Laser robot group number	Please specify the number of group of robot which performs laser processing with torch and so on.
My robot ID （In case of connecting several robots）	If laser oscillator is connected with several robots, robot ID is required to classify each robot from laser oscillator. Please set robot ID to the same number as CH (channels) of laser control request and shutter number (fiber selector number). If laser oscillator is connected with one robot, robot ID is always 1. To change robot ID, it is necessary to change before communication setting with laser oscillator. In addition, please refer to Subsection 62.2.3.1 “Setting Robot ID (In Case of Connecting Several Robots”.
Time sharing control （In case of connecting several robots）	If fiber selector is installed to laser oscillator, please set this item to enabled When this item is enabled, it is possible to use one laser oscillator on several robots while time sharing. If fiber selector is not installed, this item remains disabled.  When this item is enabled, it is possible to perform processing in order with several robots. In addition, please refer to Subsection 62.3.3 “TIME SHARING CONTROL SETUP”.
CH# Laser Ctrl rq switching	When this item is enabled, you can switch the CH (channel) of laser control request within one robot controller. In addition, please refer to Section 62.7 “SWITCHING CH# FOR LASER CTRL RQ”.
Processing speed unit	The processing speed of LS instruction is set. If you use speed tracking function, for example, power control function, wire control function and so on, please set this item. If you do not use speed tracking function, it is unnecessary to set this item.
Processing speed default	Initial value of the processing speed of LS instruction is set. If you use speed tracking function, for example, power control function, wire control function and so on, please set this item. If you do not use speed tracking function, it is unnecessary to set this item.
Stop/NonStop on LE instr	Stop: Robot stops even when the positioning path is CNT. NonStop: Robot doesn't stop when the positioning path is CNT.
Direct power override mode	Here you can choose whether to use the direct power override mode. Refer also to the section “Direct Power Override Mode” below.

Procedure 62-6 Display laser initial setup screen after controlled start

Step

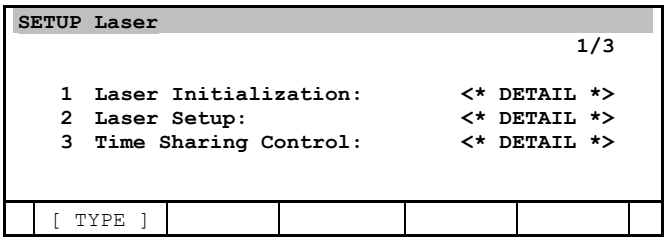
1 Perform controlled start.
2 After controlled start, press the [MENU] key. Controlled start menu will be displayed.
3 Select [0 – NEXT --]. Select [4 Laser Initial Setup].



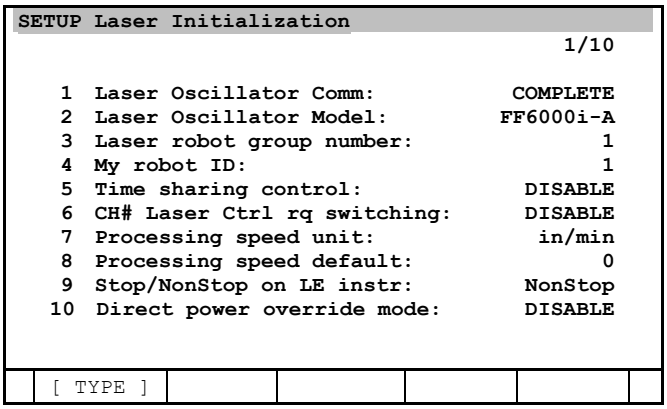
Procedure 62-7 Display laser initial setup screen after Normal Start

Step

1 Press the [MENU] key then screen menu will be displayed.
2 Select [6 SETUP].
3 Show Menu by pressed F1 [TYPE].
4 Select “Laser Setup”.



5 Move cursor on <* DETAIL *> of “Laser Initialization” and press [ENTER] key.



62.3.2 Laser Setup

In laser setup, item related to the laser processing is mainly set. Please set, if necessary.

Table 62.3.2 (a) Laser setup screen

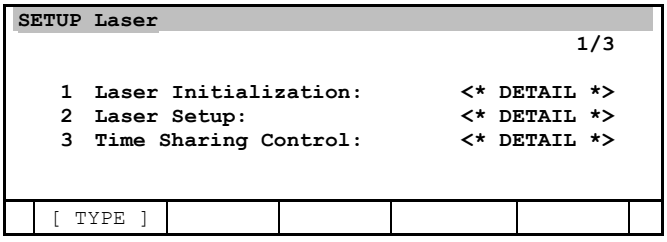
Setting item	Explanation
Laser ready latency (ms)	When PRE_LS or LS instruction is executed, “Laser ready st” signal needs to be ON. If this signal is OFF, robot waits until this signal is turned ON before processing is started. If this signal is not turned ON during the time which is set with this item, an alarm of “FFL-038” generates. If this item is set to 0, robot continues to wait until this signal is turned ON. However, if the status of beam is interlock, robot does not wait. And if processing is restart when “Process disable for restart” remains disable, robot does not wait.
Stand-by ON latency (ms)	When PRE_LS or LS instruction is executed, of “Stand-by st” signal needs to be ON. If this signal is OFF, robot waits until this signal is turned ON before processing is started. If this signal is not turned ON during the time which is set with this item, an alarm of “FFL-045” generates. If this item is set to 0, robot continues to wait until this signal is turned ON. However, if the beam is interlocked, robot does not wait. And if processing is restart when “Process disable for restart” remains disable, robot does not wait.
Delay time for beam ON (ms)	This function delays start of laser oscillator. When standalone LS instruction is executed to robot and laser oscillator at the same time, start of robot is delayed. To start robot and laser oscillator at the same time, you need to execute standalone LS instruction to laser oscillator in consideration of delay time for beam ON. By this operation, it is possible to delay start of laser operator. It is necessary to evaluate delay time for beam ON from experiment when set it. It is not necessary to change this item when using the motion option LS instruction.
Process disable for restart	If this item is enabled, it is not possible to restart laser processing when program is restarted after being paused in the processing section (including processing preparation). Therefore, “Assist gas start” signal, “Laser ready rq” signal, and “Wire feed ON” and “Wire feed speed” signal are not turned ON. In addition, it is not confirmed that “Assist gas ready” signal, “Stand-by st” signal, “Laser ready st” signal, and “Wire feed OK” signal are turned ON. If you would not like to restart laser processing, please set this item to enable.
Assist gas start/ready type	Please specify port type of “Assist gas start” signal and “Assist gas ready” signal. It is possible to select DO/DI or RO/RI.
Assist gas settling time (ms)	After outputting the assist gas, specify the time to reach the desired gas flow rate / gas pressure. Proccesing will start after this time has elapsed. However, when starting the assist gas using the PRE_LS instruction, if the assist gas sufficiently reaches the desired gas flow rate / gas pressure during the move from the PRE_LS instruction to the LS instruction, setting of this item is not necessary. Also, when using the "Assist gas ready" signal of the peripheral equipment input screen, setting of this item is not necessary.
Assist gas ready latency (ms)	When “Assist gas ready” signal is set, robot waits on LS instruction until Assist gas ready” signal is turned ON. When “Assist gas ready” signal is not turned ON after the fixed time by this item passes, “FFL-002 Assist gas ready time out” alarm is posted.
Wire feed ON/OK type	Please specify port type of “Wire feed ON” signal and “Wire feed OK” signal. It is possible to select DO/DI or RO/RI.
Delay time for wire feed (ms)	This function delays start of wire feeder. When LS instruction is executed to robot and wire feeder at the same time, start of robot is delayed. To start robot and wire feeder at the same time, you need to execute LS instruction to wire feeder in consideration of delay time for wire feed. By this operation, it is possible to delay start of wire feeder. It is necessary to evaluate delay time for wire feed from experiment when set it.
Wire feed speed output type	Please specify output type of “Wire feed speed” signal. It is possible to select group output or analog output. Refer also to the section “WIRE CONTROL” below.

Setting item	Explanation
Max val for wire feed speed AOUT	Analog output signal of max value for wire feed represents a speed range between 0 and “Max val for wire speed AOUT” using a voltage range between 0 and 10-15V. For example, if you want to represent the speed range between 0 and 200 cm/min as range between 0 and 10-15 V, please set 200 to this item. Refer also to the section “WIRE CONTROL” below.
Wire feed OK latency (ms)	When “Wire feed OK” signal is set, robot confirms after LS instruction that “Wire feed OK” signal is turned ON. When “Wire feed OK” signal is not turned ON after the fixed time by this item passes, “FFL-022 Wire feed fault” alarm is posted.
Processing state record	When this item is ENABLE, record of the laser processing state begins. Refer also to the section “PROCESSING STATE RECORD FUNCTION” below.
Guide light auto OFF timer(s)	Guide light automatic OFF is a function for saving the guide light life. “Guide light ON rq” signal turns OFF automatically after the time set here has elapsed.
LS start time adjustment (ms)	Adjust the start timing of the motion option LS instruction. Do not change this item basically.
LE start time adjustment (ms)	Adjust the start timing of the motion option LE instruction. Do not change this item basically.
LSTA start time adjustment (ms)	Adjust the start timing of the LS_TA instruction. Refer also to the section “LS_TA INSTRUCTION WITH TIME ADJUSTMENT” below.

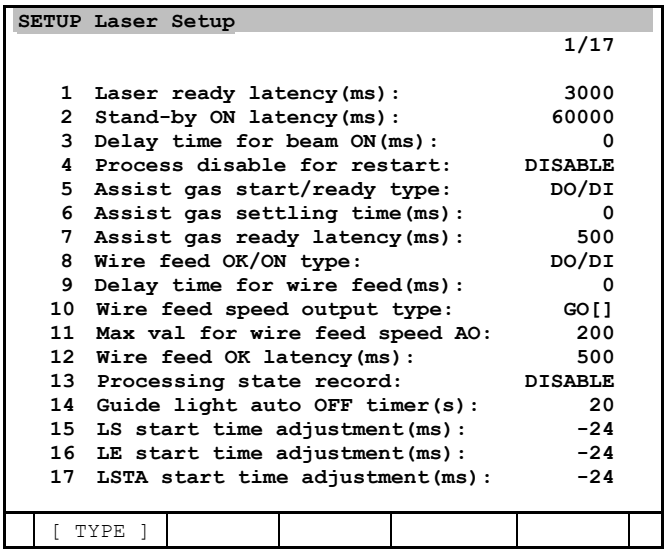
Procedure 62-8 Display laser setup screen

Step

1 Press the [MENU] key then screen menu will be displayed.
2 Select [6 SETUP].
3 Show Menu by pressed F1 [TYPE].
4 Select “Laser Setup”.



5 Move cursor on <* DETAIL *> of “Laser setup” and press [ENTER] key.



62.3.3 Time Sharing Control Setup

In time sharing control setup, item related time sharing control function of laser oscillator is mainly set. With regard to the function of time sharing control of laser oscillator, please refer to Subsection 62.3.1 “LASER INITIAL SETUP” in addition.
Please set if necessary.

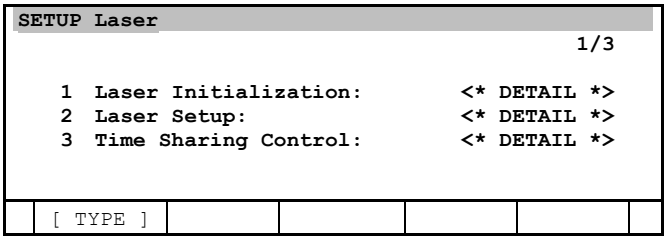
Table 62.3.3 (a) Time sharing control setup screen

Setting item	Explanation
Laser Ctrl st ON latency(ms)	**NOTE** # is the same number as the Robot ID. In order for this robot to perform laser processing, other robots must not use laser oscillators. When PRE_LS or LS instruction is performed during laser processing of other robots, robot waits until process of other robots is finished (robot waits until “CH# Laser Ctrl st” signal is turned ON.) When laser processing does not begin (“CH# Laser Ctrl st” signal is not turned ON) after fixed time by this item, “FFL-004 Las control stat ON time out” alarm is posted. If this item is set to 0, robot continues to wait until process of other robots is finished.

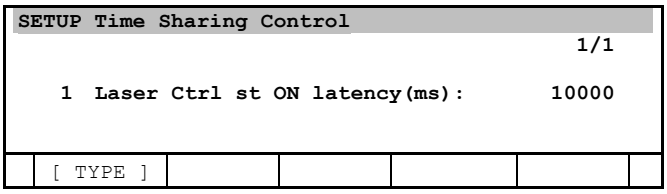
Procedure 62-9 Display time sharing control setup screen

Step

1 Press the [MENU] key then screen menu will be displayed.
2 Select [6 SETUP].
3 Show Menu by pressed F1 [TYPE].
4 Select “Laser Setup”.



5 Move cursor on <* DETAIL *> of “Time Sharing Control” and press [ENTER] key.



62.4 STARTING OF LASER OSCILLATOR

Immediately after power of laser oscillator is turned ON, laser oscillator is initial state. It is not possible to perform laser processing in this state. To perform laser processing, it is necessary to change state of laser oscillation.

States of laser oscillator are shown below.

- Initial state : When laser oscillator is turned ON power, it becomes initial state. Chiller does not run.
- Laser ready state : When “Laser ready st” signal is turned ON after “Laser ready rq” signal is turned ON, it becomes laser ready state. In this state, chiller is activated and it is started to monitor laser alarm. After that, stand-by instruction is executed automatically and it becomes stand-by state if conditions are met.
- Stand-by state : When “Stand-by st” signal is turned ON, LD power supply is energized and it becomes stand-by state. It means that it is possible to perform laser processing. When laser alarm occurs in stand-by state, it returns to initial state.

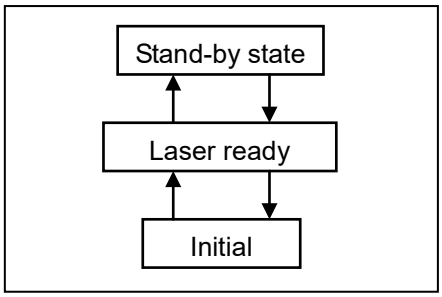


Fig. 62.4(a) Laser state

When laser alarm occurs, laser oscillator returns to initial state. If you would like to raise the state again after it returns to initial state, it is necessary to turn “Laser ready rq” signal off and turn it ON again.

NOTE
- For “Laser reset rq” and “Laser ready rq” signal, please refer to Subsection 62.5.1.1 “Laser Oscillator Output Screen”.
- For “Laser alarm st”, “Laser ready st” and “Stand-by st” signal, please refer to Subsection 62.5.1.2 “Laser Oscillator Input Screen” in addition.

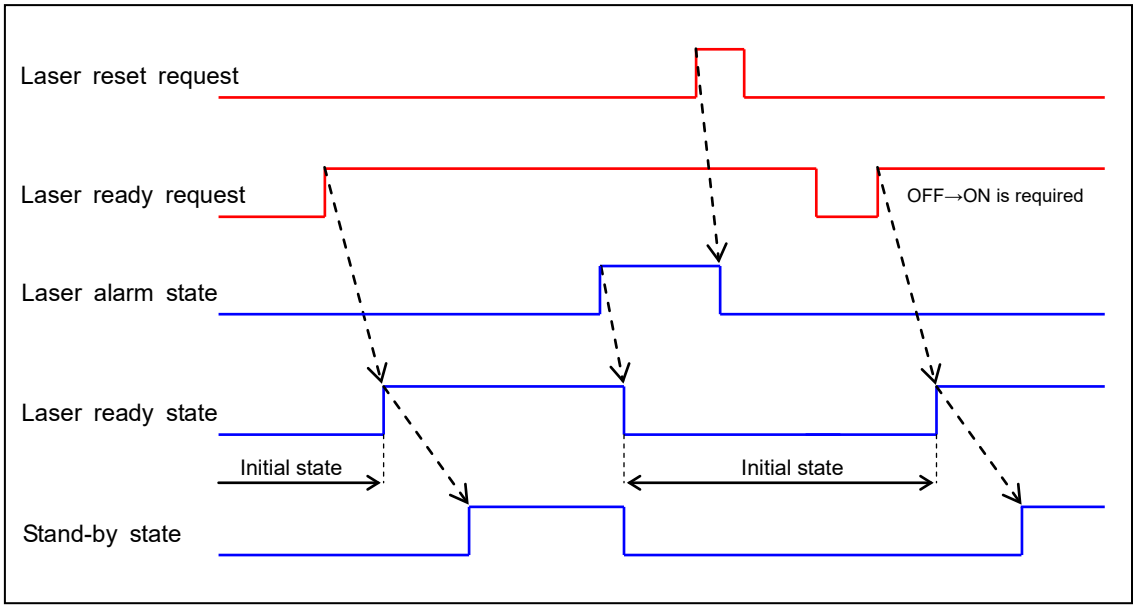


Fig. 62.4(b) Laser sequence

62.5 LASER I/O

This function has laser oscillator I/O and peripheral equipment I/O.

62.5.1 Laser Oscillator I/O

Laser oscillator I/O is set correctly by performing Procedure 62-2 of Subsection 62.2.3.2 "Communication Setting".
Please perform Procedure 62-2 again if I/O assignment is changed by the addition of other I/O device (DeviceNet board, CC-link board etc.) or the re-arrangement of I/O port numbers.

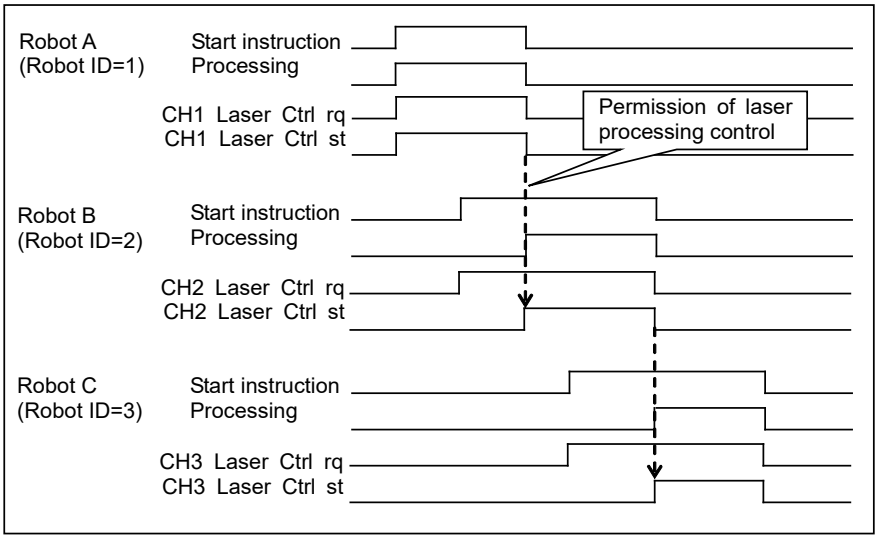
NOTE
If laser oscillator I/O assignment change is required by the addition of I/O device or the reset of I/O port numbers, please delete communication setting and reset on laser initialization screen.

62.5.1.1 Laser oscillator output screen

Table 62.5.1.1(a) Laser oscillator output screen (robot➔laser oscillator)

Name of signal	Explanation
S Peak power **Unit : W**	While LS instruction is executed, value of "S" of LS instruction is sent to laser oscillator. Value which is set is displayed here. Normally, it is not necessary to be conscious of this signal.
L Pulse bottom pwr **Unit : W**	While LS instruction is executed, value of "L" of LS instruction is sent to laser oscillator. Value which is set is displayed here. Normally, it is not necessary to be conscious of this signal.
P Pulse frequency **Unit : Hz**	While LS instruction is executed, value of "P" of LS instruction is sent to laser oscillator. Value which is set is displayed here. Normally, it is not necessary to be conscious of this signal.
Q Pulse duty **Unit : %**	While LS instruction is executed, value of "Q" of LS instruction is sent to laser oscillator. Value which is set is displayed here. Normally, it is not necessary to be conscious of this signal.

Name of signal	Explanation
Laser ready rq	This signal requests that “Laser ready st” signal and “Stand-by st” signal are turned ON. In addition, please refer to Laser ready request signal in Subsection 62.5.1.1 “Laser Oscillator Output Screen”.
Interlock OFF rq	This signal requests the permission of beam output. When this signal is off, interlock of beam output is requested from laser oscillator. When this signal and “Interlock OFF st” signal are ON, interlock of beam is canceled. When this signal is OFF, PRE_LS and LS instructions do not turn “Assist gas start”, “Laser ready rq”, “Wire feed ON” and “Wire feed speed” signal ON. In addition, it is not confirmed that “Assist gas ready”, “Laser ready st”, “Stand-by st” and “Wire feed OK” signal are turned ON. In addition, please refer to Beam interlock OFF request signal in Subsection 62.5.1.1 “Laser Oscillator Output Screen”..
Guide light ON rq	This signal requests the lighting of guide light. When this signal and “Guide light ON st” signal are ON, guide light lights. However, guide light does not light when interlock of beam has been canceled. Please use this signal for teaching.
Laser reset rq	This signal resets the laser alarm of laser oscillator (this signal turns “Laser alarm st” signal OFF). When [RESET] key is pressed or FAULT_RESET UI [5] is input, pulse is output.
Minute output rq	This signal requests that the laser oscillator become the minute output mode. When this signal is on, the robot requests a minute output mode to the laser oscillator. In addition, please refer to Section 62.11 “MINUTE OUTPUT MODE”.

Name of signal	Explanation
CH# Laser Ctrl rq （In case of connecting several robots）	**NOTE** ・ CH stands for channel. ・ # is the same number as the Robot ID. This signal requests the permission of laser processing control from laser oscillator. Robot turns this signal ON when PRE_LS or LS instruction is executed. Then, robot waits on LS instruction until the control of laser is provided. Laser processing is started after the permission of laser processing control is provided. When LE instruction is executed, the permission of laser processing control is moved to other robots after this signal is turned OFF and the permission of laser processing control is canceled. When “Time Sharing Control” is disabled, “CH# Laser Ctrl rq” signal is always ON.  As above, “CH# Laser Ctrl rq” signal is switched automatically, so normally it is not necessary to be conscious of this signal. In addition, “CH# Laser Ctrl rq” signal is switched automatically in the following pattern. <u>Teach pendant is enabled:</u> Guide light is used for teaching the laser processing part. In addition, interlock is applied to beam output during teaching. Therefore, "CH# Laser Ctrl rq” signal is turned on while the teach pendant is enabled. When the teach pendant is switched to disabled, "CH# Laser Ctrl rq” signal is turned OFF. <u>Program is finished:</u> When program is finished during laser processing, “CH# Laser Ctrl rq” signal is turned OFF.

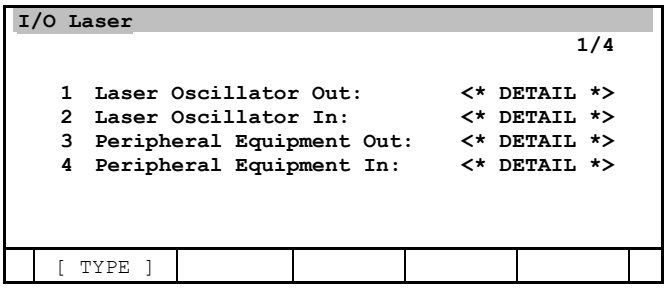
Name of signal	Explanation
Shutter# open rq （In case of connecting several robots）	**NOTE** # is the same number as the Robot ID. This signal requests laser oscillator to open shutter. “Shutter# open rq” signal whose number is same as robot ID is turned ON when PRE_LS or LS instruction is executed. Then, it is turned OFF when LE instruction is executed or alarm occurs or program is finished or stopped. If fiber selector is not installed, the state of “Shutter# open rq” signal is always same as the state of “Shutter# open st” signal. Normally, it is not necessary to be conscious of this signal.

NOTE
- Only when “CH# Laser Ctrl st” signal is ON, laser oscillator receives “Laser ready rq”, “Interlock OFF rq”, “Guide light ON rq” and “Laser reset rq” signal. The state signals are not turned ON after these request signals are turned ON when “CH# Laser Ctrl st” signal is OFF.
- If the “CH# Laser Ctrl st” signal is ON, even if these request signals are turned ON but the status signal does not turn ON, some laser alarm may be occurring. Check if there is a laser alarm on the alarm history screen, and if it is occurring, remove the cause.

Procedure 62-10 Display laser oscillator output screen

Step

1 Press the [MENU] key then screen menu will be displayed.
2 Select [5 I/O].
3 Show Menu by pressed F1 [TYPE].
4 Select “Laser I/O”.



5 Move cursor on <* DETAIL *> of “Laser Oscillator Out” and press [ENTER] key.

Laser oscillator I/O output screen [Robot➔Laser]

I/O Laser Oscillator Out

1/21

	NAME	OUT	PT	SIM	STATE
1	S Peal power:	GO[	0]	U	*****
2	L Pulse bottom pwr:	GO[	0]	U	*****
3	P Pulse frequency:	GO[	0]	U	*****
4	Q pulse duty:	GO[	0]	U	*****
5	Laser ready rq:	DO[	0]	U	***
6	Interlock OFF rq:	DO[	0]	U	***
7	Guide light ON rq:	DO[	0]	U	***
8	Laser reset rq:	DO[	0]	U	***
9	Minute output rq:	DO[	0]	U	***
10	CH1 Laser Ctrl rq:	DO[	0]	U	***
11	CH2 Laser Ctrl rq:	DO[	0]	U	***
12	CH3 Laser Ctrl rq:	DO[	0]	U	***
13	CH4 Laser Ctrl rq:	DO[	0]	U	***
14	CH5 Laser Ctrl rq:	DO[	0]	U	***
15	CH6 Laser Ctrl rq:	DO[	0]	U	***
16	Shutter1 open rq:	DO[	0]	U	***
17	Shutter2 open rq:	DO[	0]	U	***
18	Shutter3 open rq:	DO[	0]	U	***
19	Shutter4 open rq:	DO[	0]	U	***
20	Shutter5 open rq:	DO[	0]	U	***
21	Shutter6 open rq:	DO[	0]	U	***

[TYPE] IN/OUT

Laser ready request signal

It is necessary to turn “Laser ready rq” signal ON to change the state of laser oscillator to laser ready state and stand-by state from initial state. There are two methods to control “Laser ready rq” signal. One is the method of using “Laser ready rq” signal of peripheral equipment I/O and the other is the method of using PRE_LS or LS instruction.

In case of using “Laser ready rq” signal of peripheral equipment I/O

- “Laser ready rq” signal is changed according to “Laser ready rq” signal on peripheral equipment input screen.
- If you would like to raise the state again after it returns to initial state, it is necessary to turn “Laser ready rq” signal on peripheral equipment input screen OFF and turn it ON again.

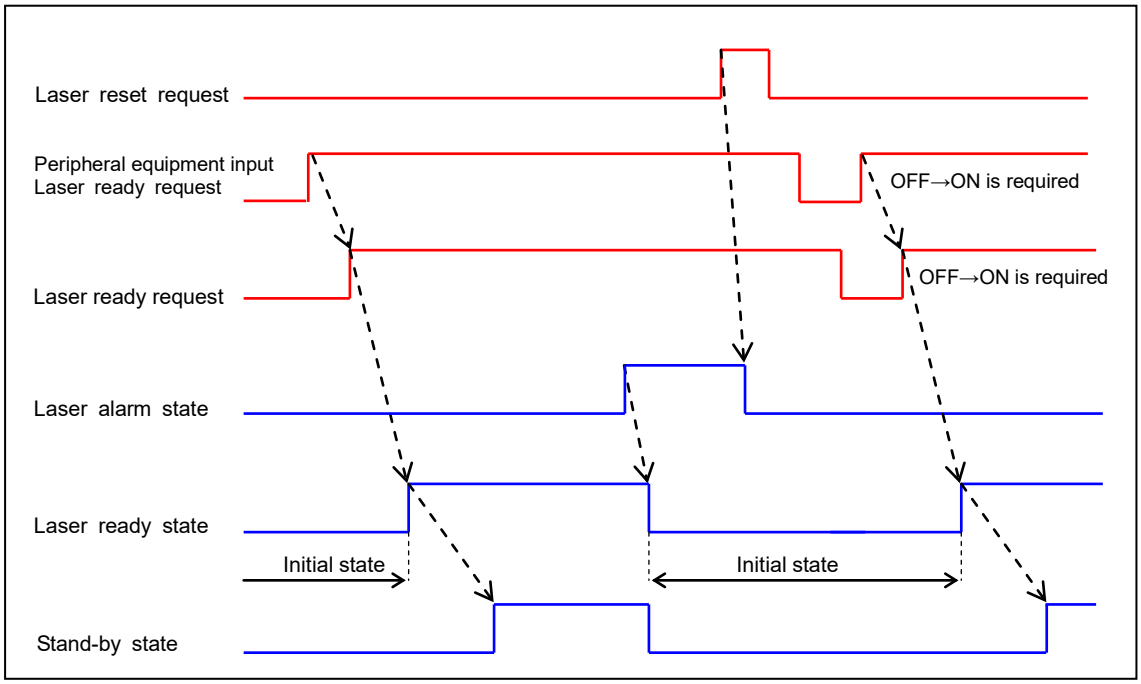


Fig. 62.5.1.1(a) “Laser ready rq” signal

In case of using PRE_LS or LS

- When PRE_LS or LS signal is executed, “Laser ready rq” signal is turned ON.
  However, “Laser ready rq” signal is not turned ON when “Interlock OFF rq” signal is OFF.
  In addition, “Laser ready rq” signal is also not turned ON if program is restarted when “Process disable for start” is enabled.
- Executing the LE instruction while the argument SOF of the LE instruction is 0 does not turn OFF “Laser ready rq” signal. If the LE instrucion is executed while the argument SOF of the LE instruction is 1, “Laser ready rq” signal will be turned OFF.
- When laser alarm is received, “Laser ready rq” signal is automatically turned OFF. Therefore, if you would like to raise the state again after it returns to initial state, it is not necessary to turn “Laser ready rq” signal OFF and turn it ON again.
- If you would like to using PRE_LS or LS instruction to control “Laser ready rq” signal, you must not set “Laser ready rq” signal of peripheral equipment I/O.

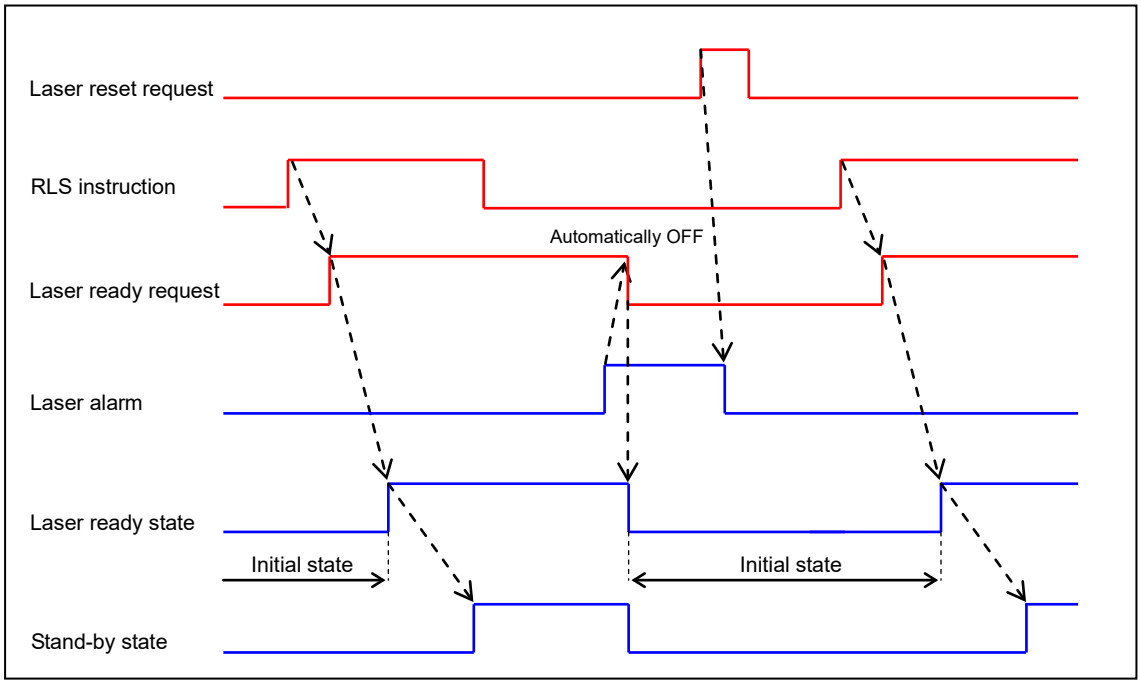


Fig. 62.5.1.1(b) Laser ready rq signal

Beam interlock OFF request signal

As a means to control the beam interlock is prepared “Interlock OFF rq” signal on peripheral equipment input screen. However, only when teach pendant is disabled, it is possible to control “Interlock OFF rq” signal with “Interlock OFF rq” signal on peripheral equipment input screen. When teach pendant is enabled, “Interlock OFF rq” signal is turned OFF forcibly because of safety. If “Guide light ON rq” signal is ON while “Interlock OFF rq” signal is OFF, guide light lights.

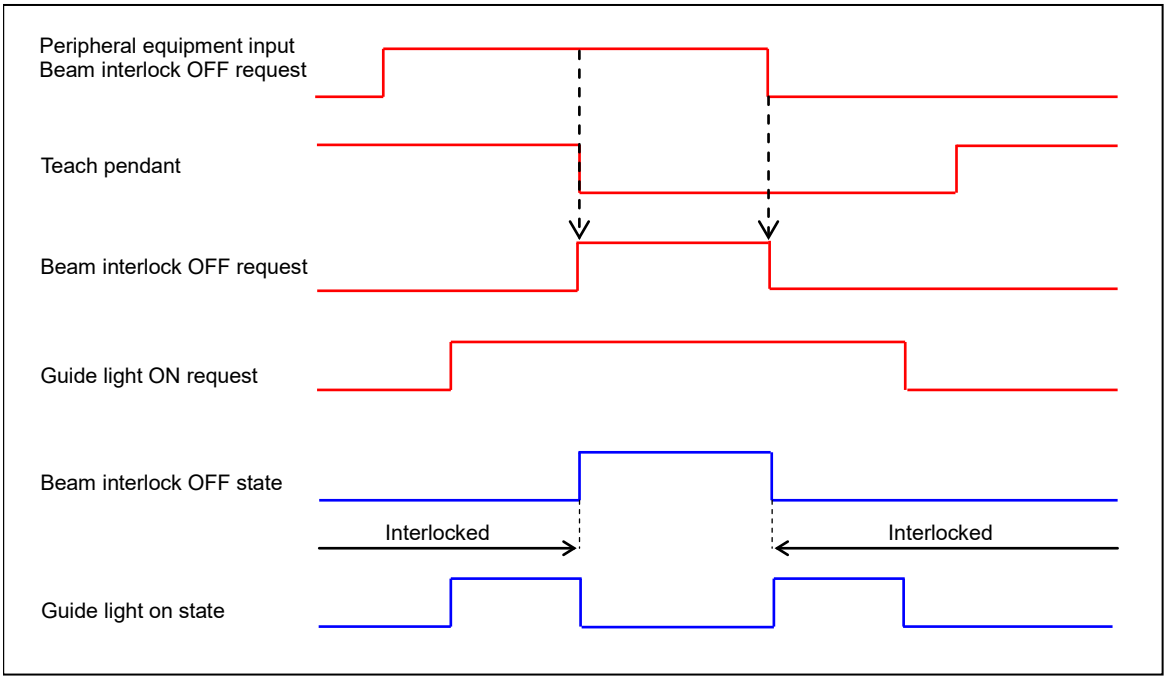
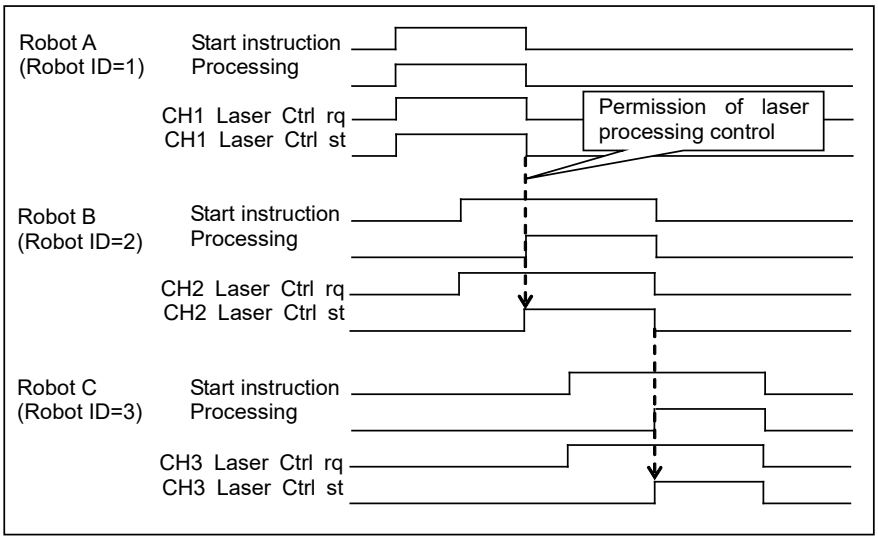


Fig. 62.5.1.1(c) "Interlock OFF rq" signal

62.5.1.2 Laser oscillator input screen

Table 62.5.1.2 (a) Laser oscillator input screen (laser oscillator➔robot)

Name of signal	Explanation
Actual power **Unit : W**	Actual output value of beam is displayed.
S Peak power **Unit : W**	Output result of "S Peak power" signal is displayed. Normally, it is not necessary to be conscious of this signal.
L Pulse bottom pwr **Unit : W**	Output result of "L Pulse bottom pwr" signal is displayed. Normally, it is not necessary to be conscious of this signal.
P Pulse frequency **Unit : Hz**	Output result of "P Pulse frequency" signal is displayed. Normally, it is not necessary to be conscious of this signal.
Q Pulse duty **Unit : %**	Output result of "Q Pulse duty" signal is displayed. Normally, it is not necessary to be conscious of this signal.
Stand-by st	Normally, this signal is turned ON when "Laser ready rq" signal is turned ON. It means that laser oscillator is stand-by state.
Laser ready st	Normally, this signal is turned ON when "Laser ready rq" signal is turned ON. It means that laser oscillator is laser ready state.
Beam ON st	This signal is turned ON after beam is output actually.
Interlock OFF st	Normally, this signal is turned ON when "Interlock OFF rq" signal is turned ON. When this signal is ON, beam is output. However, when "Stand-by st" signal is OFF, this signal is remains OFF. When this signal is OFF, beam is interlocked and "Assit gas start", "Laser ready rq", "Wire feed ON" and "Wire feed speed" signal is not turned ON in PRE_LS and LS instruction. In addition, it is not confirmed that "Assist gas ready", "Laser ready st", "Stand-by st" and "Wire feed OK" signal are turned ON.
Guide light ON st	Normally, this signal is turned ON when "Guide light ON rq" signal is turned ON. When this signal is ON, guide light lights. However, when "Interlock OFF st" signal is ON, this signal is remains OFF. Please use this signal for teaching.

Name of signal	Explanation
Laser alarm st	When laser alarm from laser oscillator is detected, this signal is turned ON. Please confirm laser alarm on alarm history screen and remove the cause if alarm occurs. When “Laser reset rq” signal is turned ON, this signal is turned OFF. Refer also to the section “Laser Alarm” below.
Minute output rq	Normally, this signal is turned ON when “Minute output rq” signal is turned ON. It means that laser oscillator is the minute output mode.
CH# Laser Ctrl st （In case of connecting several robots）	**NOTE** ・ CH stands for channel. ・ # is the same number as the Robot ID. It is not possible to perform laser processing with some robots at the same time by using one laser oscillator. Therefore, when laser oscillator is received “CH# Laser Ctrl rq” signal during laser processing of other robot, laser oscillator waits until processing of other robot is finished. After laser processing of other robot, laser oscillator turns this signal ON to provide the permission of laser processing control.  If “Time Sharing Control” on laser initial setup screen is disabled, “CH# Laser Ctrl rq” signal is always ON. Therefore, “CH# Laser Ctrl st” signal is also ON. Normally, it is not necessary to be conscious of this signal.
Shutter# open st （In case of connecting several robots）	**NOTE** # is the same number as the Robot ID. This signal indicates which shutter is currently open. Normally, the “Shutter# open rq” signal is ON, and this signal turns ON. However, if you do not have the laser control, this signal will remain OFF even if the “Shutter# open rq” signal is ON. When the fiber selector is not installed, the state of the “Shutter# open rq” signal and the state of the “Shutter# open st” signal are always in agreement. Normally, you do not need to be conscious of this signal.

NOTE
- Only when “CH# Laser Ctrl st” signal is ON, laser oscillator receives “Laser ready rq”, “Interlock OFF rq”, “Guide light ON rq” and “Laser reset rq” signal. The state signals are not turned ON after these request signals are turned ON when “CH# Laser Ctrl st” signal is OFF.
- If the “CH# Laser Ctrl st” signal is ON, even if these request signals are turned ON but the status signal does not turn ON, some laser alarm may be occurring. Check if there is a laser alarm on the alarm history screen, and if it is occurring, remove the cause.

Procedure 62-11 Display laser oscillator input screen

Step
1 Press the [MENU] key then screen menu will be displayed.
2 Select [5 I/O].
3 Show Menu by pressed F1 [TYPE].
4 Select “Laser I/O”.

```
I/O Laser
                                                    1/4

   1  Laser Oscillator Out:          <* DETAIL *>
   2  Laser Oscillator In:           <* DETAIL *>
   3  Peripheral Equipment Out:      <* DETAIL *>
   4  Peripheral Equipment In:       <* DETAIL *>

 [ TYPE ]
```

5 Move cursor on <* DETAIL *> of “Laser Oscillator In” and press [ENTER] key.

Laser oscillator I/O input screen [Laser➔Robot]

```
I/O Laser Oscillator In
                                                    1/24
     NAME                  IN PT      SIM      STATE
 1 Actual power:           GI[  0]    U        *****
 2 S Peal power:           GI[  0]    U        *****
 3 L Pulse bottom pwr:     GI[  0]    U        *****
 4 P Pulse frequency:      GI[  0]    U        *****
 5 Q pulse duty:           GI[  0]    U        *****
 6 Stand-by st:            DI[  0]    U        ***
 7 Laser ready st:         DI[  0]    U        ***
 8 Beam ON st:             DI[  0]    U        ***
 9 Interlock OFF st:       DI[  0]    U        ***
10 Guide light ON st:      DI[  0]    U        ***
11 Laser alarm st:         DI[  0]    U        ***
12 Minute output rq:       DI[  0]    U        ***
13 CH1 Laser Ctrl st:      DI[  0]    U        ***
14 CH2 Laser Ctrl st:      DI[  0]    U        ***
15 CH3 Laser Ctrl st:      DI[  0]    U        ***
16 CH4 Laser Ctrl st:      DI[  0]    U        ***
17 CH5 Laser Ctrl st:      DI[  0]    U        ***
18 CH6 Laser Ctrl st:      DI[  0]    U        ***
19 Shutter1 open st:       DI[  0]    U        ***
20 Shutter2 open st:       DI[  0]    U        ***
21 Shutter3 open st:       DI[  0]    U        ***
22 Shutter4 open st:       DI[  0]    U        ***
23 Shutter5 open st:       DI[  0]    U        ***
24 Shutter6 open st:       DI[  0]    U        ***

 [ TYPE ]          IN/OUT
```

62.5.2 Peripheral Equipment I/O

Peripheral equipment I/O is used for I/O communication between robot and peripheral equipment (PLC, etc.). Please set, if necessary.

62.5.2.1 Peripheral equipment output screen

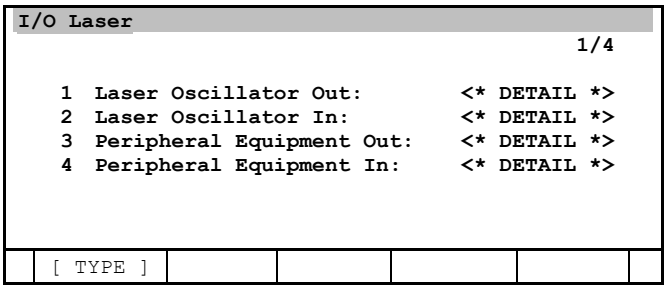
Table 62.5.2.1 (a) Peripheral equipment output screen (robot➔peripheral equipment)

Name of signal	Explanation
Stand-by st	This signal is used to output the status of "Stand-by st" signal on the laser oscillator input screen to the peripheral equipment.
Laser ready st	This signal is used to output the status of "Laser ready st" signal on the laser oscillator input screen to the peripheral equipment.
Interlock OFF st	This signal is used to output the status of "Interlock OFF st" signal on the laser oscillator input screen to the peripheral equipment.
Guide light ON st	This signal is used to output the status of "Guide light ON st" signal on the laser oscillator input screen to the peripheral equipment.
Assist gas start	This signal instructs the assist gas supply unit to turn ON/OFF the assist gas supply during PRE_LS, LS instruction execution.
Wire feed ON	This signal commands ON/OFF of wire feed to wire feeder while wire control instruction is executed. In addition, please refer to Section 62.10 “WIRE CONTROL”.
Wire feed speed	While wire control instruction is executed, value of “Sp=” of wire control instruction is output. It is possible to select output type either group output or analog output. Output type is set on laser setup screen. In addition, please refer to Section 62.10 “WIRE CONTROL”.

Procedure 62-12 Display peripheral equipment output screen

Step

1 Press the [MENU] key then screen menu will be displayed.
2 Select [5 I/O].
3 Show Menu by pressed F1 [TYPE].
4 Select “Laser I/O”.



5 Move cursor on <* DETAIL *> of “Peripheral equipment Out” and press [ENTER] key.

Peripheral equipment out screen
[Robot➔Peripheral equipment]

```
I/O Peripheral Equipment Out
                                                    1/7
     NAME                  OUT PT    SIM      STATE
  1 Stand-by st:           DO[  0]   U        ***
  2 Laser ready st:        DO[  0]   U        ***
  3 Interlock OFF st:      DO[  0]   U        ***
  4 Guide light ON st:     DO[  0]   U        ***
  5 Assist gas start:      DO[  0]   U        ***
  6 Wire feed ON:          DO[  0]   U        ***
  7 Wire feed speed:       GO[  0]   U        *****

 [ TYPE ]  CH +  IN/OUT
```

62.5.2.2 Peripheral equipment input screen

Table 62.5.2.2 (a) Peripheral equipment screen (peripheral equipment➔robot)

Name of signal	Explanation
Laser ready rq	This signal is used for controlling ON/OFF of “Laser ready rq” signal on laser oscillator output screen from peripheral equipment. This signal is available only when the teach pendant is disabled. In addition, please refer to Laser ready request signal in Subsection 62.5.1.1 “Laser Oscillator Output Screen”.
Interlock OFF rq	This signal is used for controlling ON/OFF of “Interlock OFF rq” signal on laser oscillator output screen from peripheral equipment. This signal is available only when the teach pendant is disabled. In addition, please refer to Beam interlock OFF request signal in Subsection 62.5.1.1 “Laser Oscillator Output Screen”.
Guide light ON rq	This signal is used for controlling ON/OFF of “Guide light ON rq” signal on laser oscillator output screen from peripheral equipment. This signal is available only when the teach pendant is disabled.
Assist gas ready	Use this signal when you want to check if assist gas is being supplied. Please turn ON if assist gas is supplied from the assist gas supply unit and OFF to this signal when not supplied. If this signal is ON when LS instruction is executed, laser processing is started.
Wire feed ON	This signal is used for confirming whether wire is fed or not. Please control wire feeder to output ON when wire is fed and to output OFF when wire is not fed. If this signal is ON during laser processing, laser processing is continued.

Procedure 62-13 Display peripheral equipment input screen

Step

1 Press the [MENU] key then screen menu will be displayed.
2 Select [5 I/O].
3 Show Menu by pressed F1 [TYPE].
4 Select “Laser I/O”.

```
I/O Laser
                                                    1/4

   1  Laser Oscillator Out:          <* DETAIL *>
   2  Laser Oscillator In:           <* DETAIL *>
   3  Peripheral Equipment Out:      <* DETAIL *>
   4  Peripheral Equipment In:       <* DETAIL *>

 [ 画面 ]
```

5 Move cursor on <* DETAIL *> of “Peripheral equipment In” and press [ENTER] key.

Peripheral equipment in screen
[Peripheral equipment→Robot]

```
I/O Peripheral Equipment In
                                                    1/5
     NAME                    IN PT     SIM      STATE
  1 Laser ready rq:          DI[  0]   U        ***
  2 Interlock OFF rq:        DI[  0]   U        ***
  3 Guide light ON rq:       DI[  0]   U        ***
  4 Assist gas ready:        DI[  0]   U        ***
  5 Wire feed OK:            DI[  0]   U        ***

 [ TYPE ]  CH +  IN/OUT
```

62.6 LASER INSTRUCTION

This function has laser and sub laser instruction which are necessary for laser processing.

62.6.1 Laser Instruction

Laser instruction commands to robot when and how laser processing is performed. Laser processing is performed while motion instruction lines between LS and LE instruction are executed.

- PRE_LS instruction (preparation instruction for LS instruction) : Preparation of laser processing is commanded.
- LS instruction (laser processing start instruction) : Start of laser processing is commanded.
- LE instruction (laser processing end instruction) : End of laser processing is commanded.
- LP instruction (piercing instruction) : Piercing (drilling) is commanded.

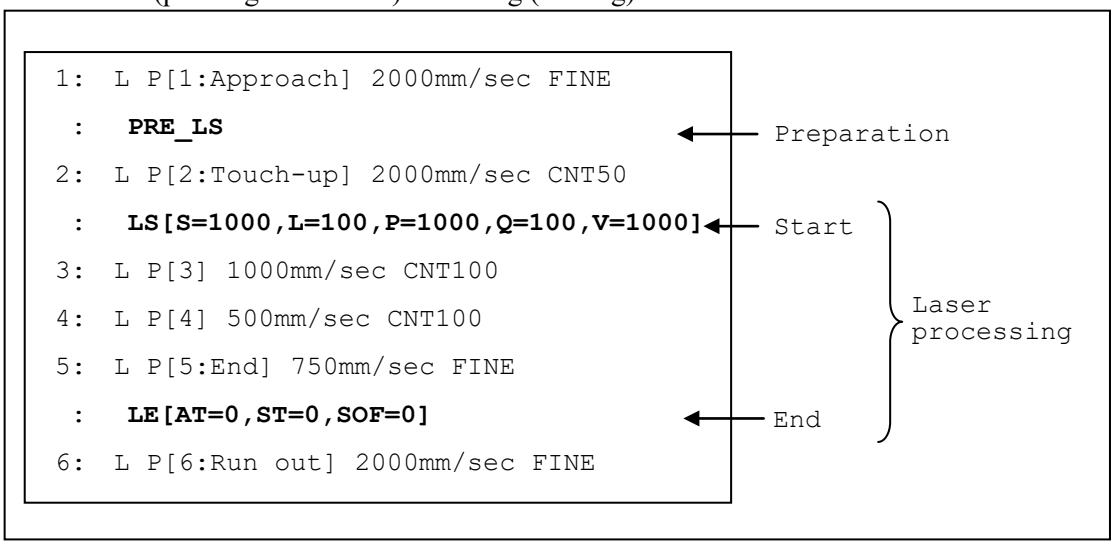


Fig. 62.6.1 Laser processing by laser instruction

62.6.1.1 PRE_LS instruction (preparation instruction for ls instruction)

- PRE_LS instruction is a preparation instruction for LS instruction. Teach PRE_LS instruction before LS instruction.
- PRE_LS instruction turns “Assist gas start” signal ON. From this, assist gas is already output when LS instruction is executed.
- If “Laser ready rq” signal on peripheral equipment screen is not set, PRE_LS instruction turns “Laser ready rq” signal ON.

- Robot waits for “Laser ready st” signal to be turned ON at PRE_LS instruction. If “Laser ready st” signal is not turned ON during the time which is set in advance, an alarm occurs.
- Robot waits for “Stand-by st” signal to be turned ON at PRE_LS instruction. If “Stand-by st” signal is not turned ON during the time which is set in advance, an alarm occurs.

When time sharing control is disabled

- “CH# Laser Ctrl rq” signal and “CH# Laser Ctrl st” signal are always ON.
- However “Shutter# open rq” signal is turned ON, “Shutter# open rq” signal always same as “Shutter open# st” signal because fiber selector is not installed.

NOTE
is the same number as the Robot ID.

When time sharing control is enabled

- In order to acquire the laser control and open the shutter, turn ON the “CH# Laser Ctrl rq” signal and “Shutter# open rq” signal.
- Robot waits for “CH# Laser Ctrl st” signal to be turned ON at PRE_LS instruction. If “CH# Laser Ctrl st” signal is not turned ON during the time which is set in advance, an alarm occurs.
- Robot waits for “Shutter# open st” signal to be turned ON at PRE_LS instruction. If “Shutter# open st” signal is not turned ON during the time which is set in advance, an alarm occurs.

NOTE
is the same number as the Robot ID.

When beam is interlocked
When “Process disable for restart” is enabled

- “Assist gas start” signal is not turned ON.
- “Laser ready rq” signal is not turned ON.
- Robot does not wait for “Laser ready st” and “Stand-by st” signal to be turned ON.

62.6.1.2 LS instruction (laser processing start instruction)

- The format of LS instruction is shown below.

LS[S=*, L=*, P=*, Q=*, V=*]

- S : Peak power(W)
- L : Pulse bottom power(W)
- P : Pulse frequency(Hz)
- Q : Pulse duty(%)
- V : Processing speed(mm/sec, cm/min, inch/min)

Fig. 62.6.1.2(a) LS instruction

- Meanings of processing schedules (argument) are shown below.

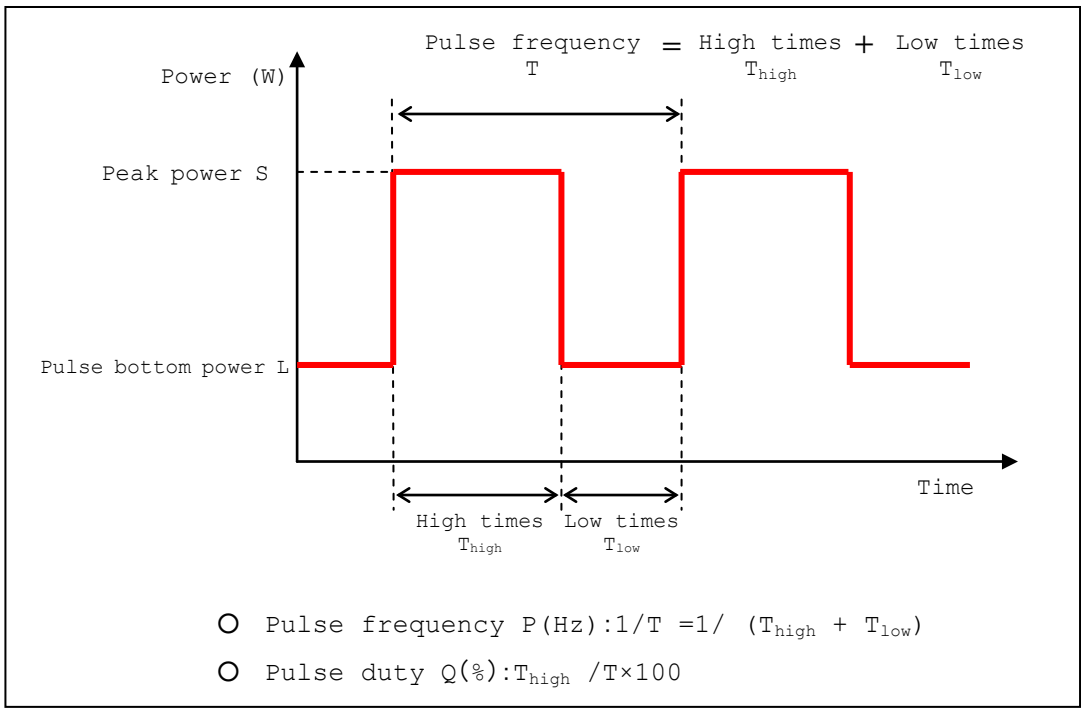


Fig. 62.6.1.2(b) Processing schedule of LS instruction

- LS instruction is laser processing start instruction.
- If LS instruction is executed when conditions of laser output are met, processing schedules set to LS instruction are output to laser oscillator.

NOTE
In this function, beam is not output when teach pendant is enabled because of safety. To output beam, it is necessary that teach pendant is disabled.

- LS instruction turns “Wire feed” and “Wire feed speed” signal ON.
- LS instruction starts to confirm “Wire feed OK” signal. This confirmation continues during laser processing.
- Processing speed V is used for speed tracking function (power control function, wire control function and etc.). Output in accordance with change of speed or wire feed speed is calculated based on the value which is set to processing speed V.

NOTE
If speed tracking function is not used, it is not necessary to consider processing speed V.

- Maximum and minimum value of processing schedule depends on “Laser Oscillator Model” on laser initial setup screen. Please change “Laser Oscillator Mode”, if necessary.
- Unit, initial value, maximum and minimum value of processing speed V depends on “Processing speed unit” and “Processing speed default” on laser initial setup screen. Please change “Processing speed unit” and “Processing speed default”, if necessary.
- It is possible to change processing schedules during laser processing.

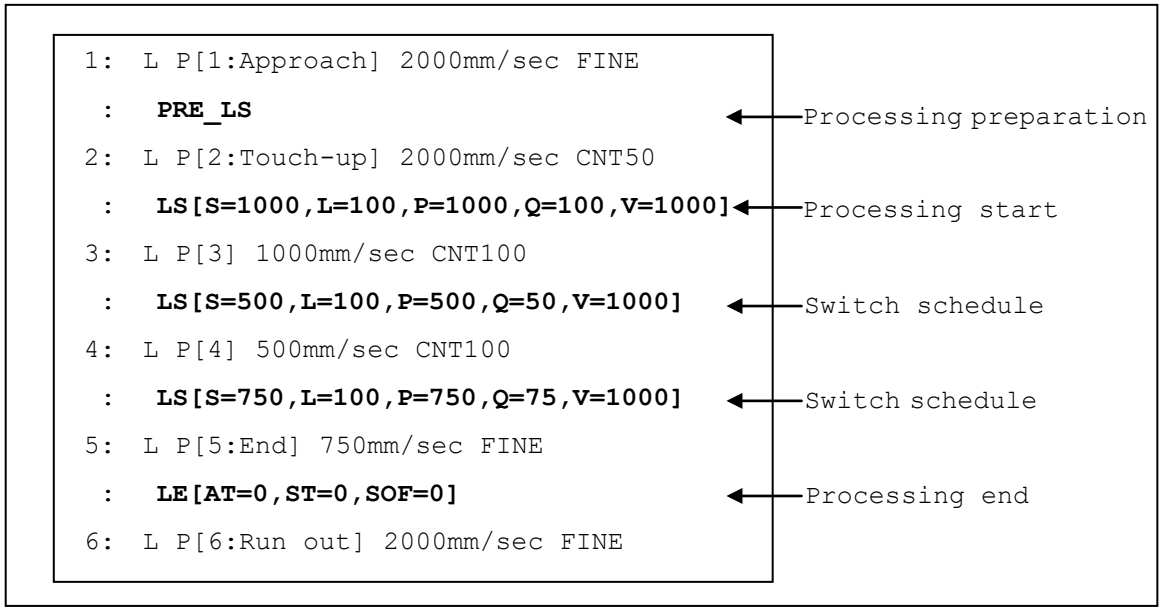


Fig. 62.6.1.2(c) Switch processing schedule

When LS instruction is executed after PRE_LS instruction (Normal use)

- When LS instruction is executed, it is necessary that the following are already satisfied by PRE_LS instruction.
  - Robot is laser ready state and stand-by state.
  - Permission of laser processing is provided.
  - Assist gas is output.
  - Shutter is opened.

  Therefore, the robot doesn't stop if the positioning path is CNT.

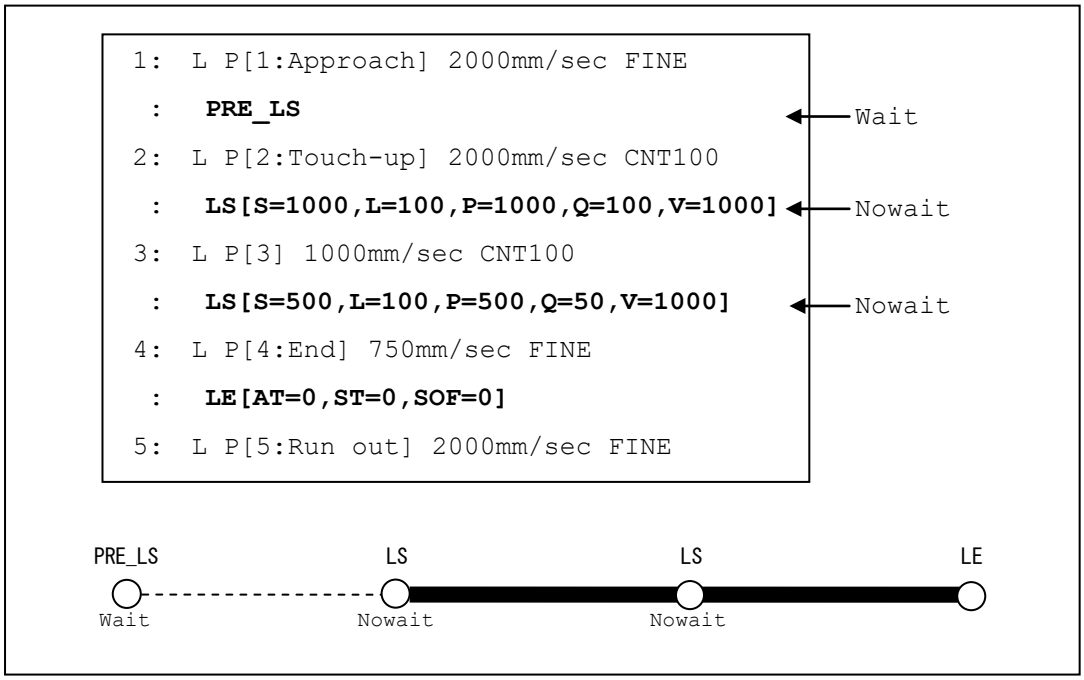


Fig. 62.6.1.2(d) When PRE_LS instruction exists

- When "Assist gas ready" signal is set, LS instruction confirms whether "Assist gas ready" signal has been turned ON or not. However, when the positioning path is CNT, robot confirms that "Assist gas ready" signal is turned ON while the robot is moving because robot does not decelerate. When "Assist

gas ready" signal is not turned ON during fixed time after LS instruction is executed, the alarm occurs during LS instruction sequence.

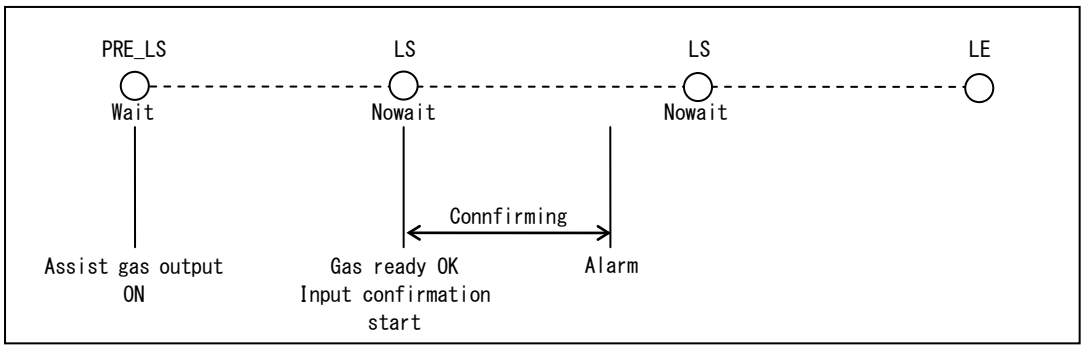


Fig. 62.6.1.2(e) Confirming "Gas ready" signal

When beam is interlocked
When [Process disable for restart] is enabled

- "Assist gas ready" signal is not confirmed.
- "Wire feed ON" and "Wire feed speed" signal are not turned ON.
- "Wire feed OK" signal is not confirmed.

When LS instruction is executed without PRE_LS instruction

- Robot waits for "Laser ready st" signal to be turned ON at LS instruction. If "Laser ready st" signal is not turned ON during the time which is set in advance, the alarm occurs.
- Robot waits for "Stand-by st" signal to be turned ON at LS instruction. If "Stand-by st" signal is not turned ON during the time which is set in advance, an alarm occurs.
- "Assist gas start" signal are turned ON when LS instruction is executed.
- When "Assist gas ready" signal is set, robot waits at LS instruction until "Assist gas ready" signal is turned ON. Therefore, robot stops even when the positioning path is CNT. When "Assist gas ready" signal is not turned ON during the fixed time, the alarm occurs.

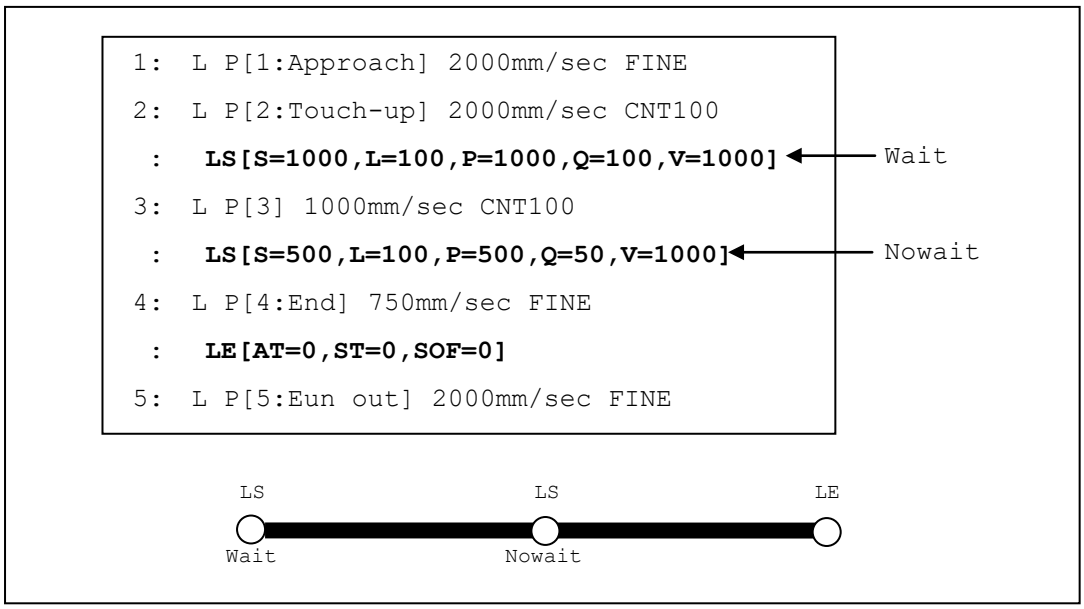


Fig. 62.6.1.2(f) PRE_LS instruction does not exist

When time sharing control is disabled

- "CH# Laser Ctrl rq" signal and "CH# Laser Ctrl st" signal are always ON.
- However "Shutter# open rq" signal is turned ON, "Shutter# open rq" signal always same as "Shutter# open st" signal because fiber selector is not installed.

NOTE
is the same number as the Robot ID.

When time sharing control is enabled

- "CH# Laser Ctrl rq" signal and "Shutter# open rq" signal are turned ON. From this, control of laser is already provided and shutter is already opened when LS instruction is executed.
- Robot waits for "CH# Laser Ctrl st" signal to be turned ON at LS instruction. If "CH# Laser Ctrl st" signal is not turned ON during the time which is set in advance, an alarm occurs.
- Robot waits for "Shutter# open st" signal to be turned ON at LS instruction. If "Shutter #open st" signal is not turned ON during the time which is set in advance, an alarm occurs.

NOTE
is the same number as the Robot ID.

When beam is interlocked
When "Process disable for restart" is enabled

- "Assist gas start" signal is not turned ON.
- Robot does not wait for "Assist gas ready" signal to be turned ON.
- "Laser ready rq" signal is not turned ON.
- Robot does not wait for "Laser ready st" and "Stand-by st" signal to be turned ON.
- "Wire feed ON" and "Wire feed speed" signal are not turned ON.
- "Wire feed OK" signal is not confirmed.

62.6.1.3 LE instruction (laser processing end instruction)

- The format of LS instruction is shown below.

LE[AT=*, ST=*, SOF=*]

- AT：Afterflow time (0ms-9999ms)
- ST：Shutter OFF delay time (0ms-9999ms)
- SOF：Stand-by OFF (0-1)

Fig. 62.6.1.3(a) LE instruction

- LE instruction is laser processing end instruction.
- Stop the output of processing schedule.
- "Assist gas start" signal are turned OFF.
- Executing the LE instruction while the argument SOF is 0 does not turn OFF "Laser ready rq" signal. If the LE instrucion is executed while the argument SOF is 1, "Laser ready rq" signal will be turned OFF.
- "Wire feed ON" and "Wire feed speed" signal are turned OFF.
- If afterflow time (AT=) is set to LE instruction, "Assist gas start" signal are turned OFF after exceeding afterflow time. If next PRE_LS or LS instruction is executed before afterflow time passes, "Assist gas start" signal are not turned OFF by LE instruction. This function has the effect of afterflow, and the effect of saving the assist gas ON /OFF time in the repeat of shot processing.

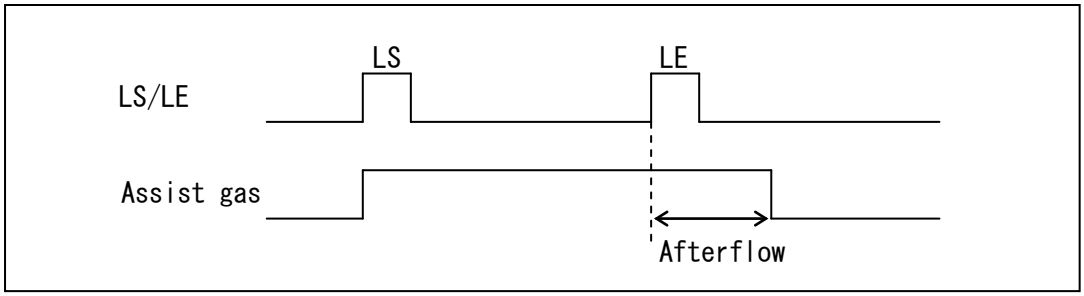


Fig. 62.6.1.3(b) Afterflow time

- When LE instruction is executed, “Shutter open rq” signal is turned OFF. If shutter OFF delay time (ST=) is set to LE instruction, “Shutter open rq” signal is turned OFF after exceeding shutter OFF delay time. If next PRE_LS or LS instruction is executed before shutter OFF delay time passes, “Shutter open rq” signal is not turned OFF by LE instruction. This function has the effect of saving the shutter open/close time in the repeat of short processing.

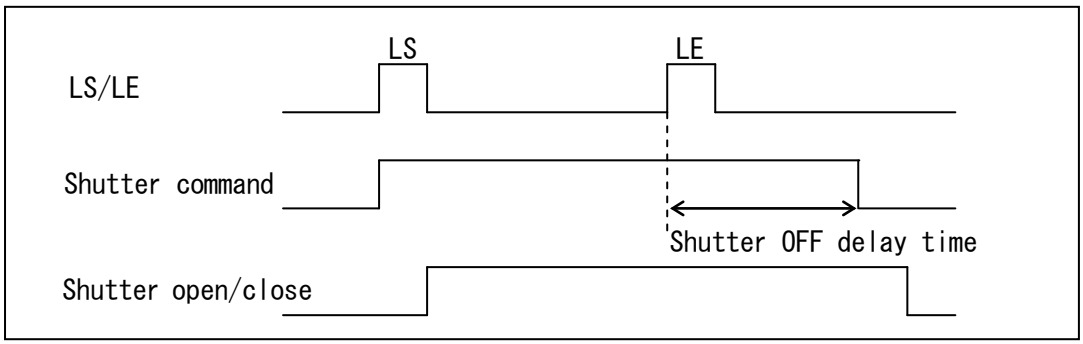


Fig. 62.6.1.3(c) Shutter OFF delay time

When time sharing control is disabled

- “CH# Laser Ctrl rq” signal is not turned OFF and permission of laser processing control is not released.

NOTE
is the same number as the Robot ID.

When time sharing control is enabled

- “CH# Laser Ctrl rq” signal and “Shutter# open rq” signal are turned OFF, and the permission of laser processing control is released and shutter close command is performed. It is not confirmed whether these request is finished or not.
- Because the permission of laser processing control is released after the shutter close command is performed, if shutter delay OFF time (ST=) is set, the timing to turn “CH# Laser Ctrl rq” signal OFF is delayed according to the delay of the shutter close command. If you would not like to transfer the permission of laser processing control to other robots, please set shutter OFF delay time. If the program is finished, the permission of laser processing control is released.

NOTE
is the same number as the Robot ID.

When Stop/NonStop on LE instr is Stop

- Robot stops even when the positioning path is CNT.

When Stop/NonStop on LE instr is NonStop

- Robot doesn't stop when the positioning path is CNT.

62.6.1.4 LP instruction (piercing instruction)

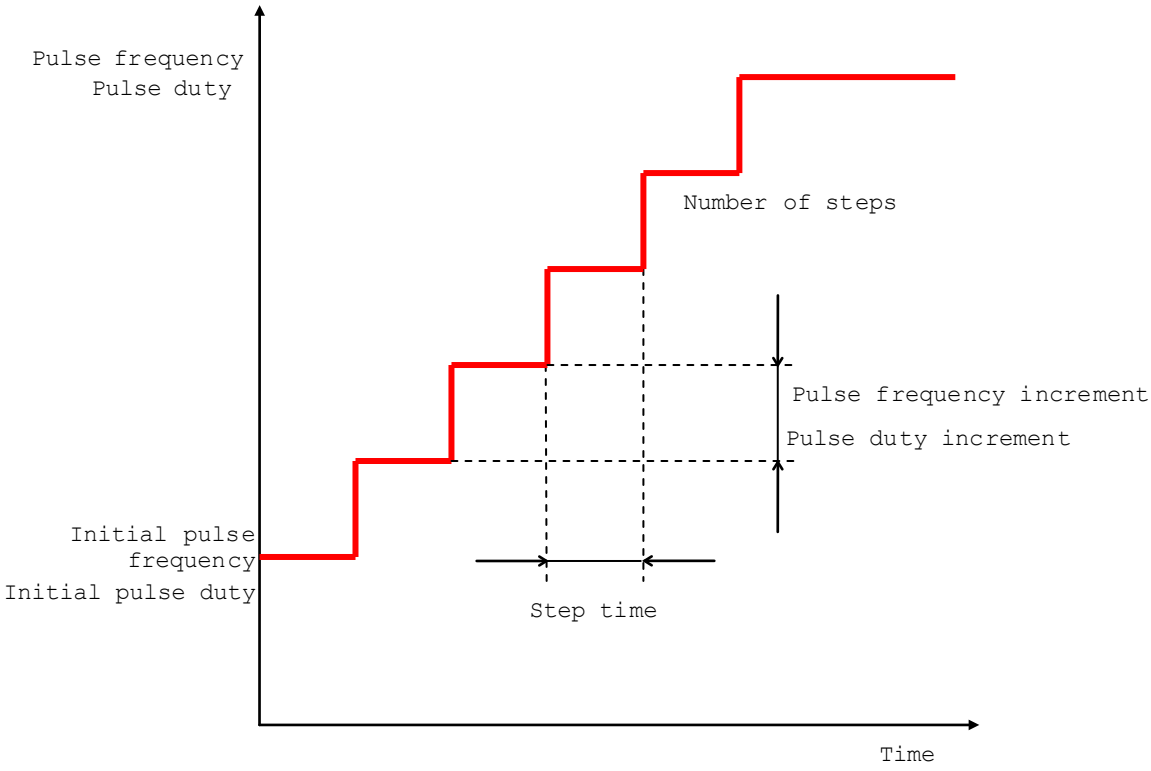
- The format of LP instruction is shown below.

LP[S=*, L=*, P=*, Q=*, I =*, J=*, K=*, H=*, R=*]

- S : Peal power (W)
- L : Pulse bottom power (W)
- P : Initial pulse frequency (Hz)
- Q : Initial pulse duty (%)
- I : Pulse frequency increment (Hz)
- J : Pulse duty increment (%)
- K : Step time (msec)
- H : Number of steps
- R : Piercing time (msec)

Fig. 62.6.1.4(a) LP instruction

- Meanings of processing schedules (argument) are shown below.



- Initial pulse frequency: Specifies the initial pulse frequency during piercing.
- Initial pulse duty: Specifies the initial pulse duty during piercing.
- Pulse frequency increment: Specifies the pulse frequency increment per step during piercing.
- Pulse duty increment: Specifies the pulse duty increment per step during piercing.
- Step time: Specifies the duration of a single step during piercing.
- Number of steps: Specifies the total number of steps during piercing.
- Piercing time: Specifies the duration of piercing. The duration must be longer than the time which is the result of multiplying the step time and the number of steps. If it is shorter, piercing is finished after piercing time passes even when the specified number of steps is executed.

Fig. 62.6.1.4(b) Processing schedules of LP instruction

- LP instruction is used to achieve the optimum power command by changing the output during piercing (drilling) stepwise. From this, it is possible to perform stable piercing in the shortest time. LP instruction is used as before processing to let beam penetrate the work.
- LP instruction increases average output by increasing pulse frequency and pulse duty for every unit time while keeping the power command of the laser beam.
- Output requirement of each signal and specification for confirming states are same as LS instruction basically.
- When LP instruction is taught, LS instruction is added automatically. This LS instruction is used for cut processing after LP instruction ends (After pre-processing ends).

NOTE
If the value specified for K, H or R is 0, the LP instruction is not executed.

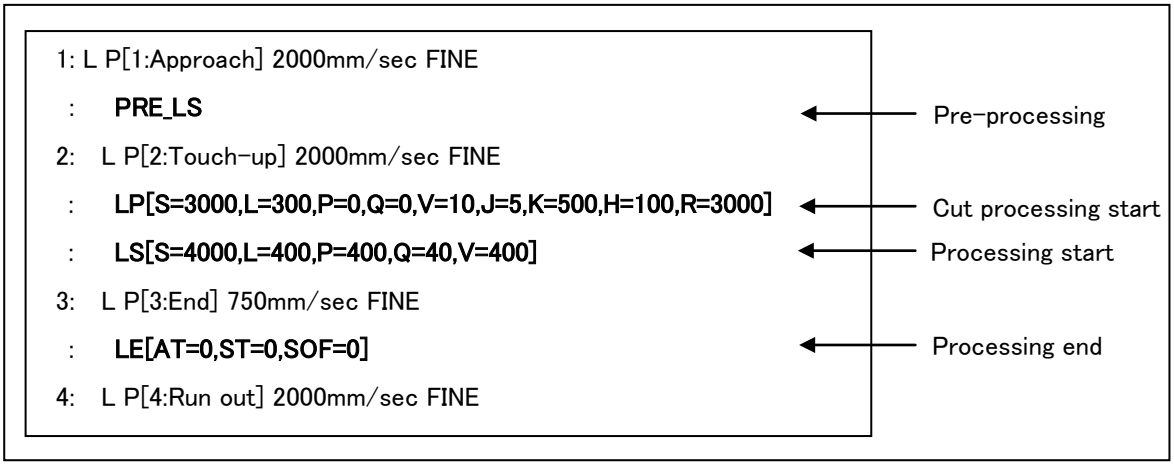


Fig. 62.6.1.4(c) Performing laser processing by LS instruction

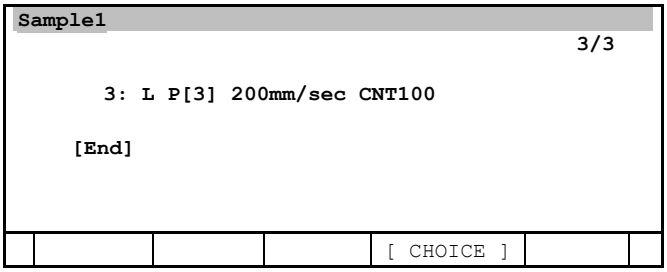
62.6.1.5 Teaching laser instruction

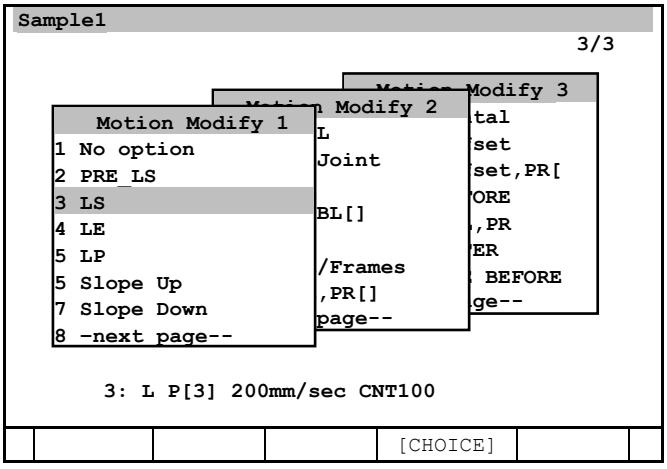
Laser instructions have PRE_LS, LS, LE and LP instruction. The teaching procedure is the same. Therefore, only the LS instruction is explained.

NOTE
Additional motion instruction and single instruction are provided for laser instruction. Normally, additional motion instruction is used.

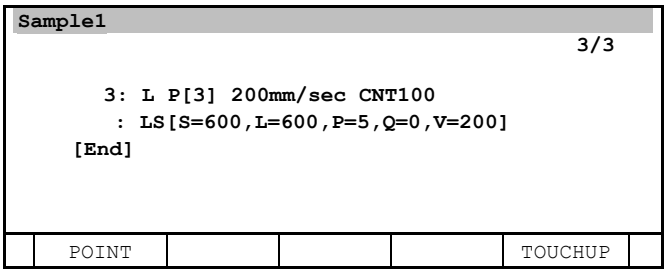
Procedure 62-14 Teaching laser instruction

1 Place the cursor behind the motion instruction and press F4 [CHOICE] to display the additional motion instruction menu.





2 Select “LS” from the menu.



62.6.2 Sub Laser Instruction

Sub laser instruction is the instruction to support laser processing.
For sub laser instruction, the following instructions are available. Please use, if necessary.

- Power control instruction
- Slope up instruction
- Slope down instruction
- Wire control instruction.
- Minute output request instruction.

62.6.2.1 Power control instruction

Please refer to the Chapter 8 “POWER CONTROL”.

62.6.2.2 Slope up instruction

Please refer to the Chapter 9 “SLOPE CONTROL”.

62.6.2.3 Slope down instruction

Please refer to the Chapter 9 “SLOPE CONTROL”.

62.6.2.4 Wire control instruction

Please refer to the Chapter 10 “WIRE CONTROL”.

62.6.2.5 Minute output request instruction

Please refer to Section 62.11 “MINUTE OUTPUT MODE”.

62.7 SWITCHING CH# FOR LASER CTRL RQ

You can switch the CH (channels) for laser control request within one robot controller. With this function, multiple heads can be switched with one robot controller without plugging or unplugging the fiber cable.

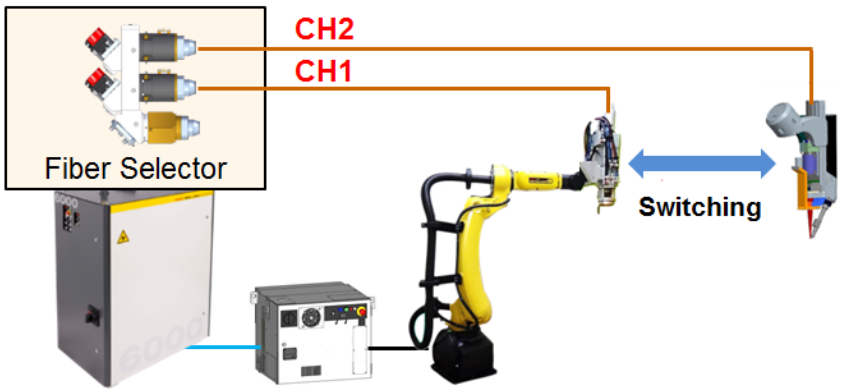


Fig. 62.7 (a) Head switching within one robot controller

62.7.1 Enable/Disable Ch# Laser Ctrl Rq Switching

To switch the CH (channels), please enable “CH# Laser Ctrl rq switching” on the laser initial setup screen.

Procedure 62-15 Enable/Disable CH# Laser Ctrl rq switching

Step

1 Perform controlled start.
2 After controlled start, press the [MENU] key. Controlled start menu will be displayed.
3 Select [0 – NEXT --]. Select [4 Laser Initial Setup].

```
Laser Initialization                                  1/10

   1  Laser Oscillator Comm:                      INCOMP
   2  Laser Oscillator Model:                  FF6000i-A
   3  Laser robot group number:                        1
   4  My robot ID:                                     1
   5  Time sharing control:                      DISABLE
   6  CH# Laser Ctrl rq switching:               DISABLE
   7  Processing speed unit:                       in/min
   8  Processing speed default:                        0
   9  Stop/NonStop on LE instr:                  NonStop
  10  Direct power override mode:                DISABLE

[ TYPE ]                          SETUP
```

4 Enable “Time sharing control”.
5 Enable “CH# Laser Ctrl rq switching”.

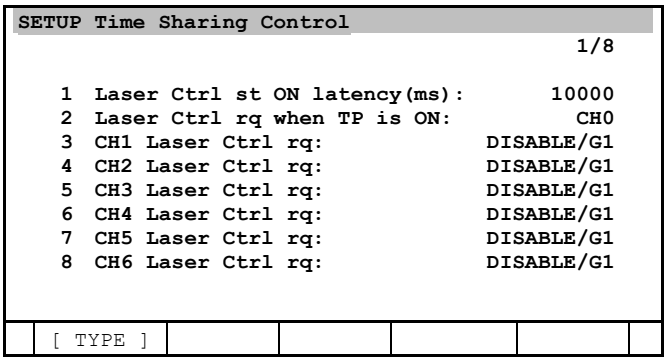
62.7.2 Enable/Disable Ch# Laser Ctrl Rq

Enable all CH (channels) to be used within this robot controller.

Procedure 62-16 Enable/Disable CH# Laser Ctrl rq

Step

1 Press the [MENU] key then screen menu will be displayed.
2 Select [6 SETUP].
3 Show Menu by pressed F1 [TYPE].
4 Select “Laser Setup”.
5 Move cursor on <* DETAIL *> of “Time Sharing Control” and press [ENTER] key.



6 Move the cursor to “CH1-6 Laser Ctrl rq“ and enable all CH (channels) planned to be used.
7 For multi-group system, specify the group number of the robot that uses the enabled CH (channels).

62.7.3 Expansion of Peripheral Equipment I/O

“Assist gas start” signal, “Wire feed ON” signal, “Wire feed speed” signal of peripheral equipment output are extended for each CH (channels).
“Assist gas ready” signal, “Wire feed OK” signal of peripheral equipment input are extended for each CH (channels).

Procedure 62-17 Expansion of peripheral equipment I/O

Step

1 Press the [MENU] key then screen menu will be displayed.
2 Select [5 I/O].
3 Show Menu by pressed F1 [TYPE].
4 Select “Laser I/O”.
5 Move cursor on <* DETAIL *> of “Peripheral equipment Out/In” and press [ENTER] key.

Peripheral equipment out screen
[Robot➔Peripheral equipment]

```
I/O Peripheral Equipment Out
                                                    1/7
     NAME                  OUT PT    SIM      STATE
  1 Stand-by st:           DO[  0]   U        ***
  2 Laser ready st:        DO[  0]   U        ***
  3 Interlock OFF st:      DO[  0]   U        ***
  4 Guide light ON st:     DO[  0]   U        ***

CH1:
  5 Assist gas start:      DO[  0]   U        ***
  6 Wire feed ON:          DO[  0]   U        ***
  7 Wire feed speed:       GO[  0]   U        *****

 [ TYPE ]  CH +  IN/OUT
```

Peripheral equipment in screen
[Peripheral equipment➔Robot]

```
I/O Peripheral Equipment In
                                                    1/5
     NAME                  IN PT     SIM      STATE
  1 Laser ready rq:        DI[  0]   U        ***
  2 Interlock OFF rq:      DI[  0]   U        ***
  3 Guide light ON rq:     DI[  0]   U        ***

CH1:
  4 Assist gas ready:      DI[  0]   U        ***
  5 Wire feed OK:          DI[  0]   U        ***

 [ TYPE ]  CH +  IN/OUT
```

6 Assign each signal as necessary.
7 Press F2 "CH +" to switch to the display of another CH (channel).

62.7.4 Ch (Channels) Switching by Pre_Ls Instruction

- Switching of CH (channels) is performed with the PRE_LS instruction.
- The format of the PRE_LS instruction is shown below.

PRE_LS[CH=*]

– CH : Channel of laser control request

Fig. 62.7.4(a) PRE_LS instruction

- The PRE_LS instruction turns ON the “CH# Laser Ctrl rq” signal with the number specified in CH=.
- The PRE_LS instruction waits until the “CH# Laser Ctrl st” signal with the number specified in CH= turns ON.

NOTE
If “CH# Laser Ctrl rq switching” on the laser initial setup screen is enabled, PRE_LS [CH = #] instruction must be executed before executing LS instruction.

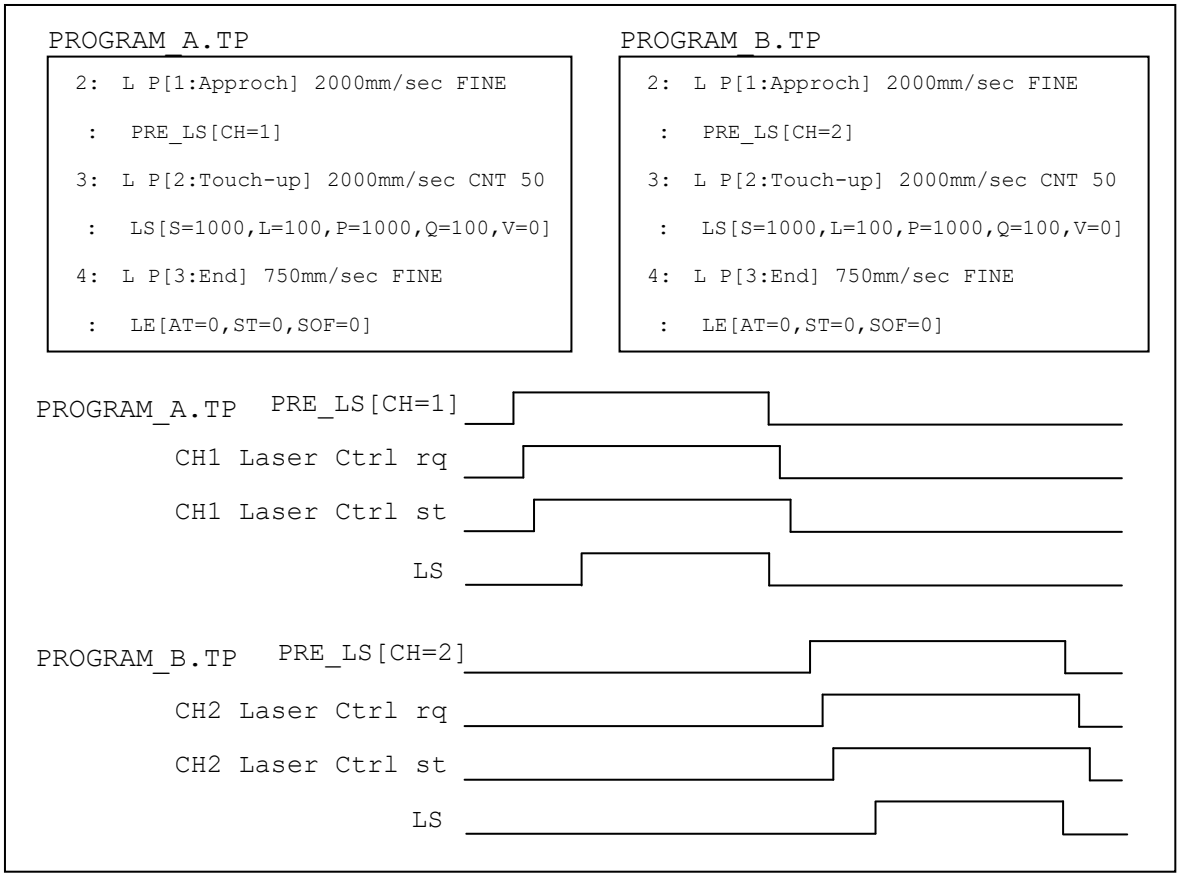
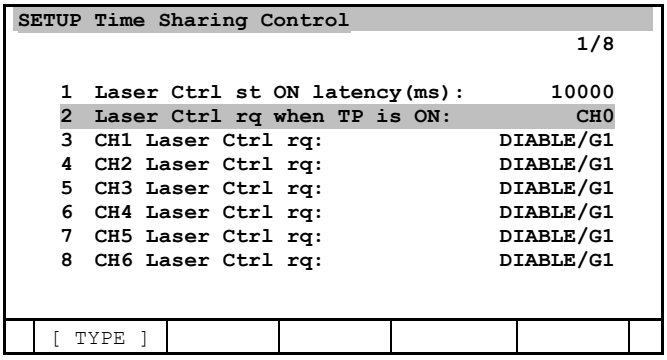


Fig. 62.7.4(b) Switching of CH of laser control request

- When switching the CH (channels) manually when the teach pendant is enabled, please set “Laser Ctrl rq when TP is ON” on the time sharing control setup screen.



NOTE

When executing the PRE_LS[CH=#] instruction while the teach pendant is enabled, the CH=# of PRE_LS instruction and the CH# of "Laser Ctrl rq when TP is ON" must be the same.

62.8 POWER CONTROL

The laser power control ensures uniform processing by controlling the peak power, pulse bottom power, pulse frequency, and pulse duty command when an actual processing speed changes in the corner part etc. compared with the processing speed of LS instruction.

62.8.1 Details of Power Control

According to power control schedule defined beforehand by power control instruction, LS instruction turns ON power control mode.
When power control mode turns ON, power control functions during the laser output.
During power control, the output of each processing schedule is changed depending on the ratio of actual processing speed and specified processing speed within the range of each processing schedule and power control schedule set by LS instruction.

Table 62.8.1 (a) Power control schedule

	Power control schedule	
	Power control schedule ≦ LS instruction data	Power control schedule ＞ LS instruction data
Minimum peak power	Peak power	0 (Clamp value)
Minimum pulse bottom power	Pulse bottom power	0 (Clamp value)
Minimum pulse frequency	Pulse frequency	5 (Clamp value)
Minimum pulse duty	Pulse duty	0 (Clamp value)
Processing speed 0 peak power	0 (Clamp value)	Peak power
Processing speed 0 pulse bottom power	0 (Clamp value)	Pulse bottom power
Processing speed 0 pulse frequency	5 (Clamp value)	Pulse frequency
Processing speed 0 pulse duty	0 (Clamp value)	Pulse duty
Processing speed allowable variation	Processing speed allowable variation	Processing speed allowable variation

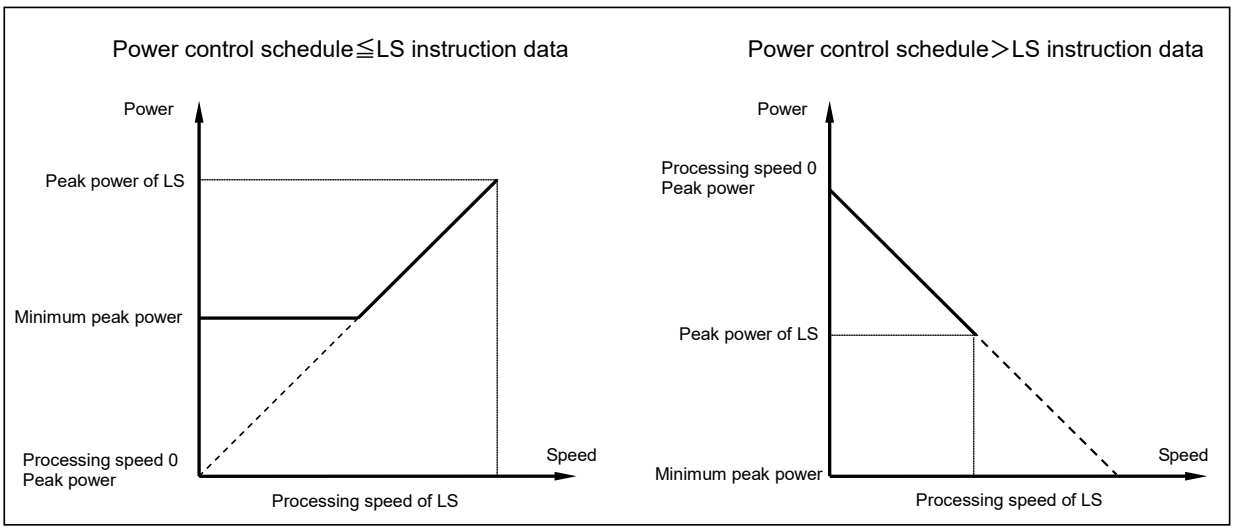


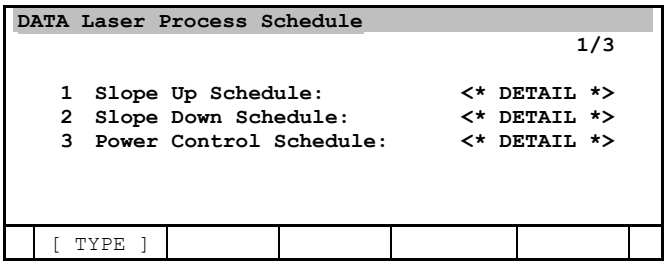
Fig. 62.8.1 (a) Details of power control

62.8.2 Power Control Schedule

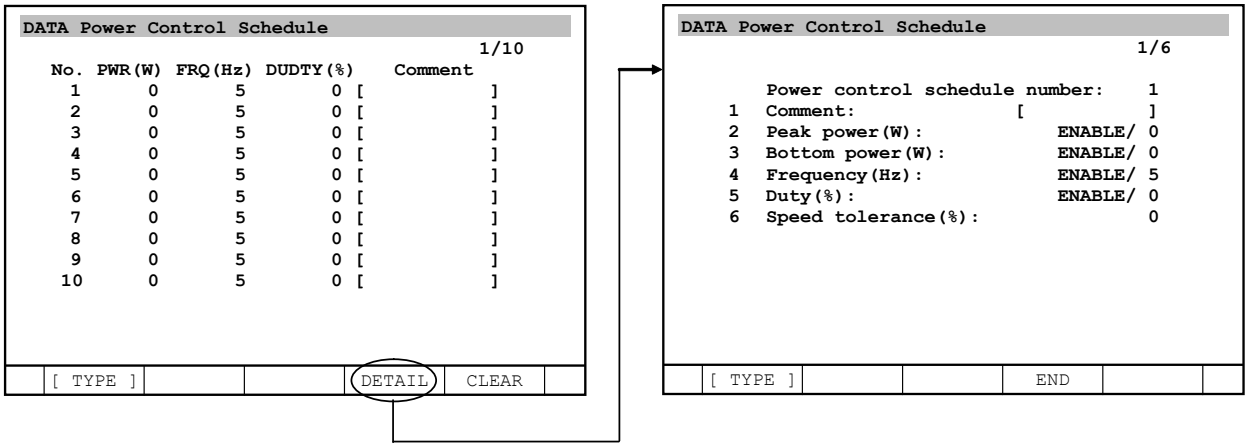
10 power control schedules are provided.

Procedure 62-18 Display power control schedule screen

1 Press the [MENU] key and display the screen menu.
2 Press [0. -- NEXT --] and select [3. DATA].
3 Press F1[Type] and display the screen menu.
4 Select “Laser Schedule”.



5 Move the cursor to <*DETAIL*> of Power Control Schedule, and press [ENTER] key.



62.8.3 Power Control Instruction

- The format of power control instruction is shown below.

POWER_CONTROL[*]

– * : power control schedule number (0 - 10)

Fig. 62.8.3(a) Power control instruction

- Please set assist gas instruction preliminary to LS instructions.
- If the power control schedule has been specified in range of 1 to 10 by the power control instruction before the LS instruction is executed, the power control mode becomes active. And, when 0 is specified to it, the power control mode is canceled.
- In the example below, the power control mode becomes active by POWER CONTROL [1] of the second line, and the power control mode is canceled by POWER CONTROL [0] of the 24th line.

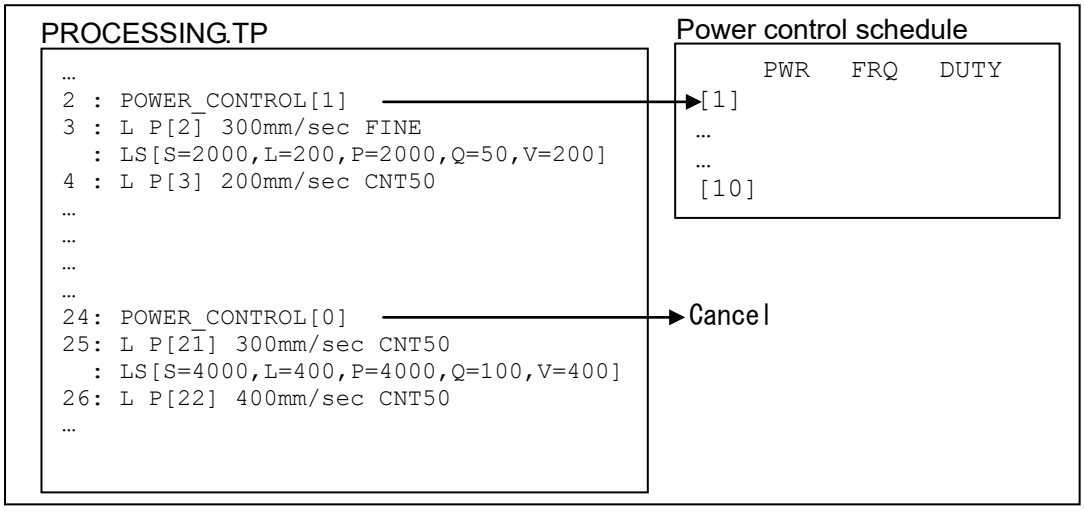


Fig. 62.8.3(b) Example of power control instruction execution

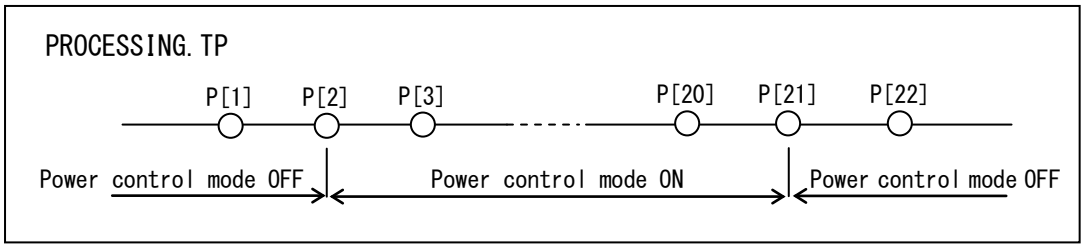
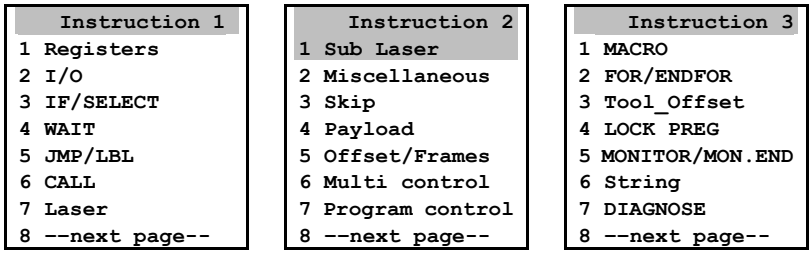


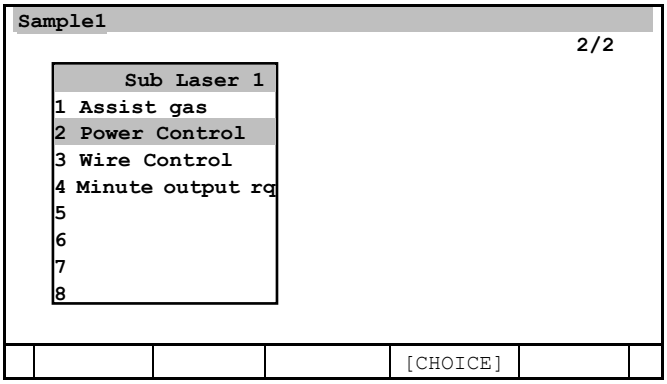
Fig. 62.8.3(c) Example of power control instruction execution

Procedure 62-19 Teaching power control instruction

1 To teach this instruction, press F1[INST], display sub menu, and select “Sub Laser”.



2 Select “Power Control” from Sub Laser menu.



```
Sample1                                              2/2

      2: POWER_CONTROL[1]

    [End]

 [ INST ]                                        [EDCMD]
```

62.8.4 Processing Speed Allowable Variation

By setting the processing speed allowable variation, when robot speed changes slightly, if the change is smaller than the processing speed variation, it is considered that the robot is moving at the specified motion speed, and each processing schedule output calculated from the ratio of actual processing speed and specified processing speed doesn't change.

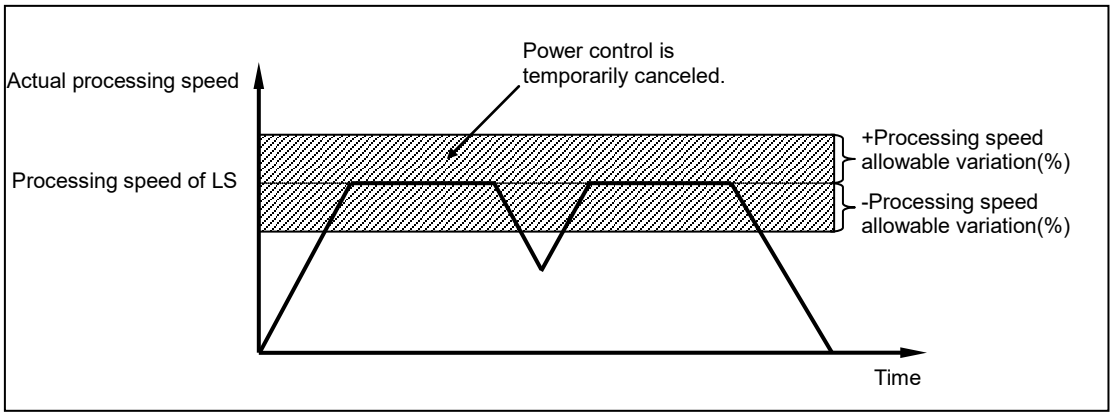
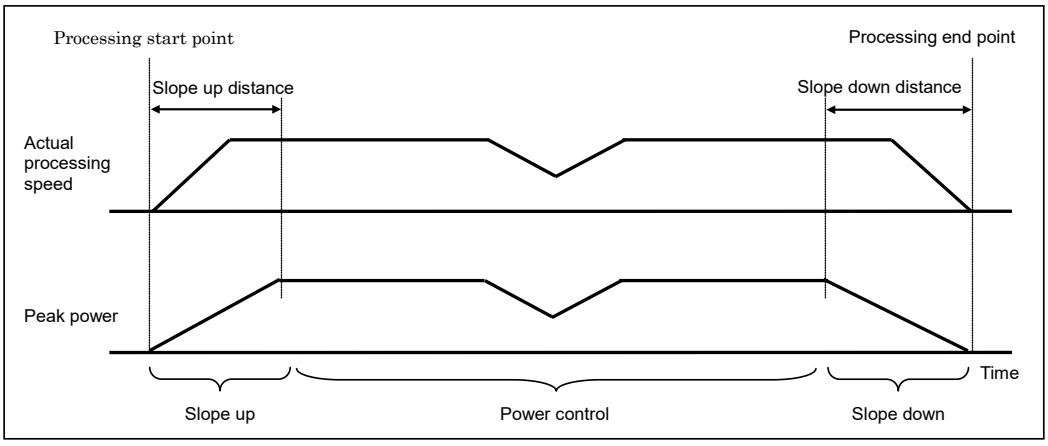


Fig. 62.8.4 (a) Processing speed allowable variation

62.8.5 Notifications For Power Control

NOTE

When both the power control schedule and the slope schedule are specified to one program, the slope control is given to priority in the vicinity of the processing start point or the processing end point, and the power control becomes active in other parts.



NOTE
Specify either of the linear motion or the circular motion to the motion format of the processing path if you want to use power control. You can't use power control when the joint motion is specified to the motion format.
When the program is terminated or another line is selected, power control schedule registered by power control instruction is cleared.

62.9 SLOPE CONTROL

Slope control can gradually increase or decrease a processing data (peak power, pulse bottom power, pulse frequency, and pulse duty) of LS instruction while processing the specified section. It is possible to increase / decrease a processing data smoothly.
Following 2 type of slope control are provided.

- Slope up control: This can be used with LS instruction.
- Slope down control : This can be used with LS, LE instruction.

62.9.1 Slope up Control

According to slope up schedule defined beforehand by slope up instruction, LS instruction turns ON slope up control.
When slope up control turns ON, peak power, pulse bottom power, pulse frequency, and pulse duty of LS instruction are increased gradually during the processing between the position with slope up instruction and the point where slope up schedule is assigned. So, processing data rises up smoothly.

62.9.1.1 Details of slope up control

The robot outputs the laser by using "Start peak power" until processing time passes "Stop time" while pausing at P[2], and changes the output of the laser toward peak power of LS after the processing time passes "Stop time". An actual laser output is changed in proportion to moved distance so that "Start peak power" may reach peak power of LS. An actual laser output is decreased in proportion to moved distance if peak power of LS is lower than "Start peak power".
Start peak power, Stop time and Slope up distance can be assigned by slope up schedule.

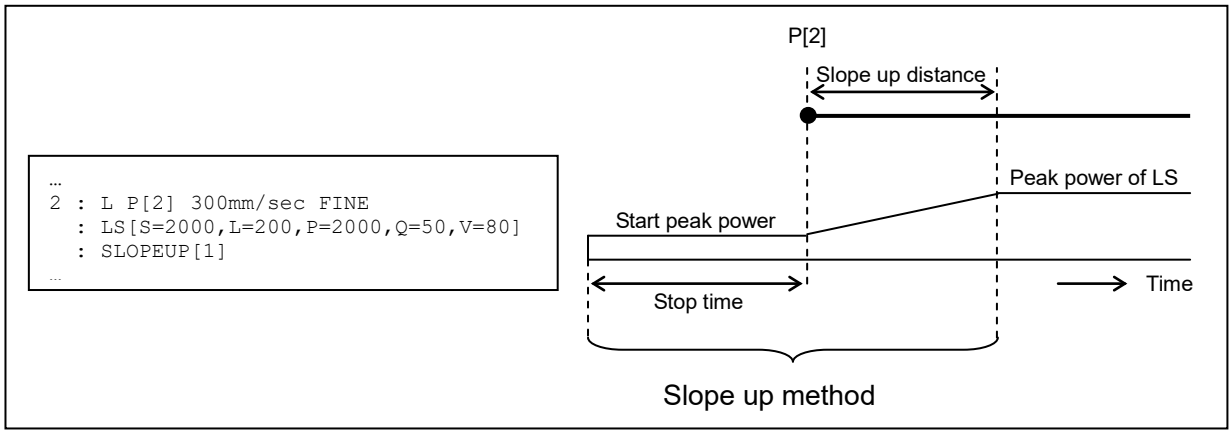


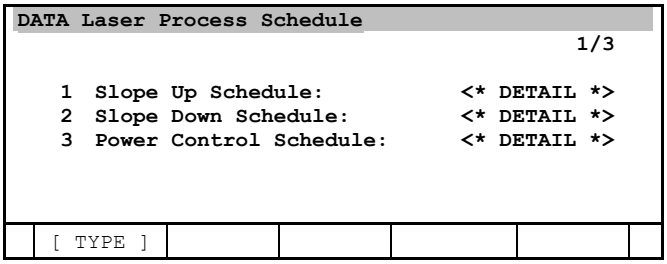
Fig. 62.9.1.1 (a) Details of slope up control

62.9.1.2 Slope up schedule

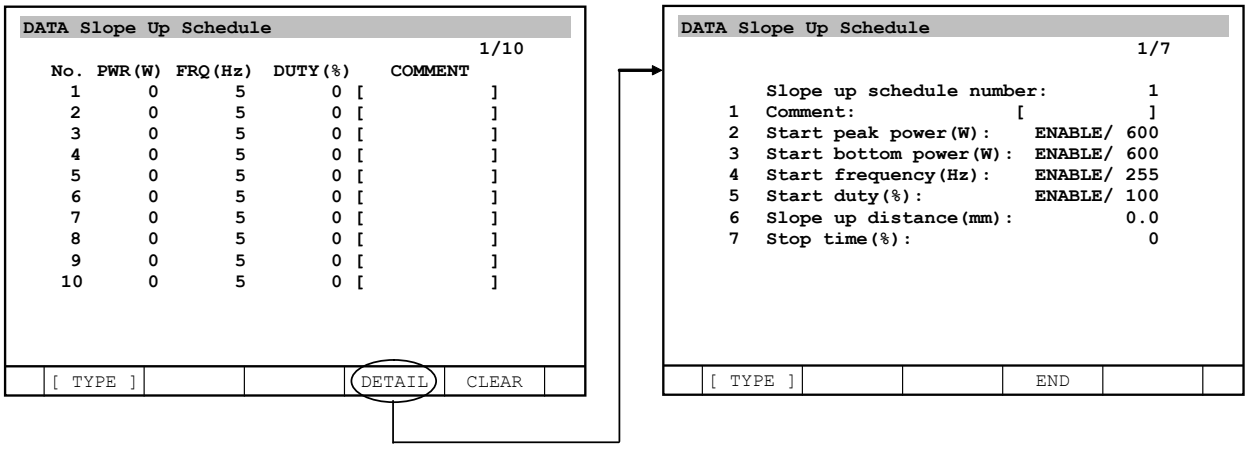
10 slopes up schedules are provided.

Procedure 62-20 Display slope up schedule screen

1 Press the [MENU] key and display the screen menu.
2 Press [0. -- NEXT --] and select [3. DATA].
3 Press F1[TYPE] and display the screen menu.
4 Select “Laser Schedule”.



5 Move the cursor to <*DETAIL*> of “Slope Up Schedule”, and press [ENTER] key.



62.9.1.3 Slope up instruction

- The format of slope up instruction is shown below.

SLOPEUP[*]

– * : slope up schedule number (1 - 10)

Fig. 62.9.1.3(a) Slope up instruction

- Slope up instruction is provided as the supplementary motion instruction.
- Slope up instruction can only be added to the same line as LS instruction.
- If LS instruction and slope up instruction have been added to motion instruction, the slope up mode becomes active according to the schedule number specified for the slope up instruction.
- In the example below, the slope up mode becomes active by SLOPEUP [1] of the second line, and the slope up mode continues until the robot reaches the distance in the slope up schedule. There is no slope up mode in the 25th line because the 25th line LS instruction doesn't have the slope up instruction.

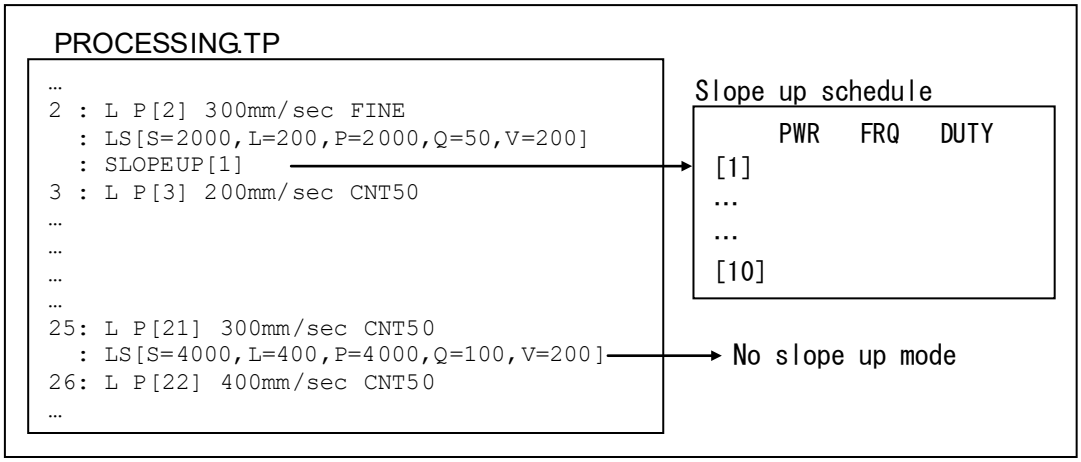


Fig. 62.9.1.3(b) Example of slope up instruction execution

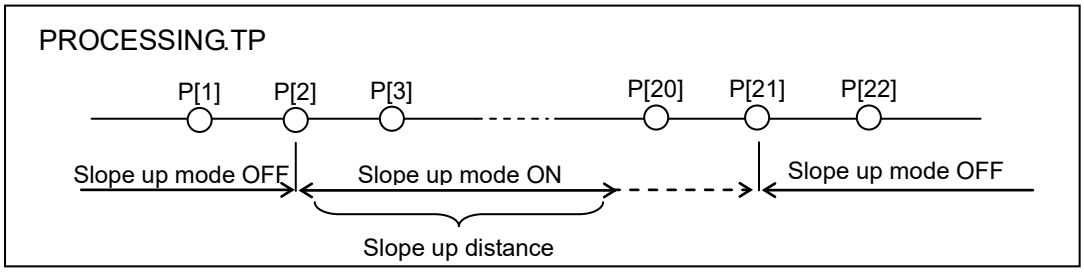
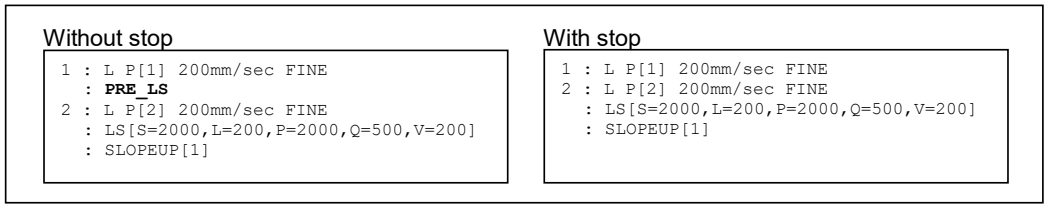


Fig. 62.9.1.3(c) Example of slope up instruction execution

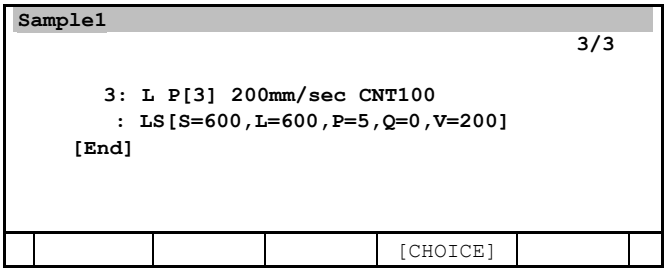
NOTE

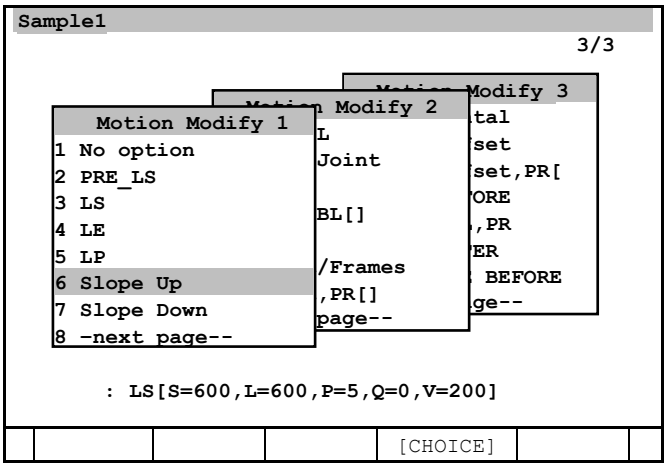
When PRE_LS instruction is assigned before LS instruction, robot doesn't stop even if "Stop time"of slope up schedule is assigned. To stop robot by slope up, please don't use PLE_LS instruction.



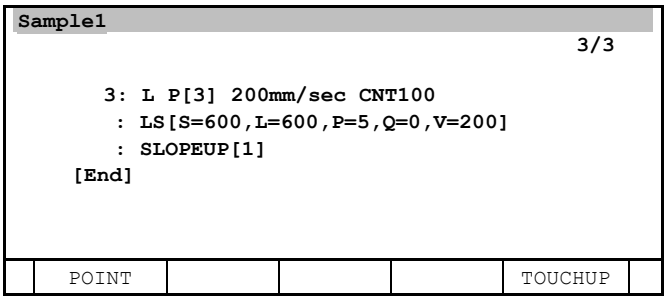
Procedure 62-21 Teaching slope up instruction

1 To teach this instruction, place the cursor behind the LS instruction and press F4[CHOICE] to display the supplementary motion instruction menu.





2 Select “Slope Up” from the menu.



62.9.2 Slope Down Control

According to slope down schedule defined beforehand by slope down instruction, LS, LE instruction turns ON slope down control.
When slope down control turns ON, peak power, pulse bottom power, pulse frequency, and pulse duty of LS instruction are decreased gradually during the processing between the point where slope up schedule is assigned and the position with slope down instruction. So, processing data descend smoothly.

62.9.2.1 Details of slope down control

When robot is at the “Slope down distance” front from P[4], the robot changes the peak power of LS toward "End peak power", and an actual laser output is changed in proportion to moved distance so that peak power of LS may reach "End peak power" at P[4]. An actual laser output is increased in proportion to moved distance if "End peak power" is higher than peak power of LS. And, the robot outputs the laser by using "End peak power" until processing time passes "Stop time" while pausing at P[4].
End peak power, Stop time and Slope down distance can be assigned by slope down schedule.

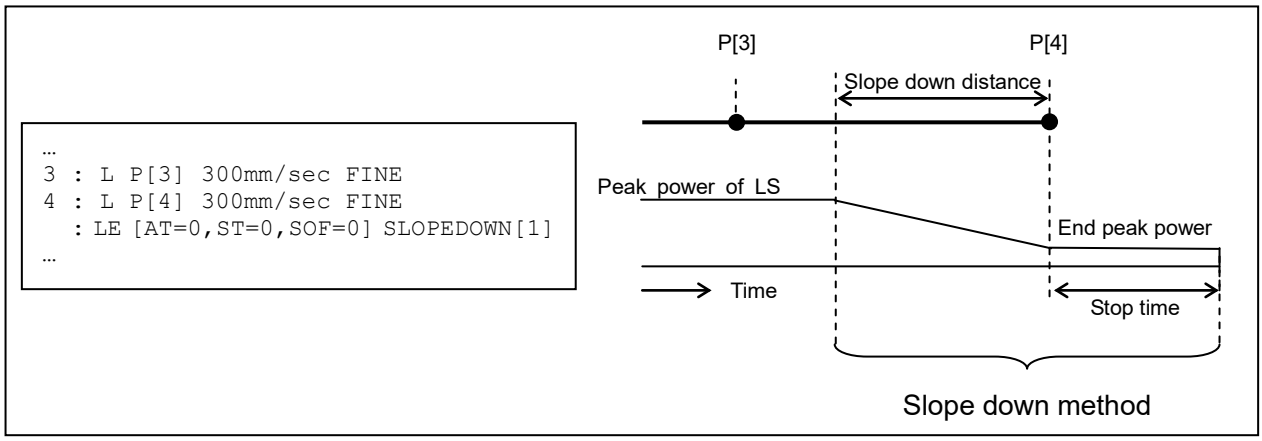


Fig. 62.9.2.1 (a) Details of slope down control

NOTE
In the case of above figure, slope down distance should be smaller than the distance between P[3] and P[4].

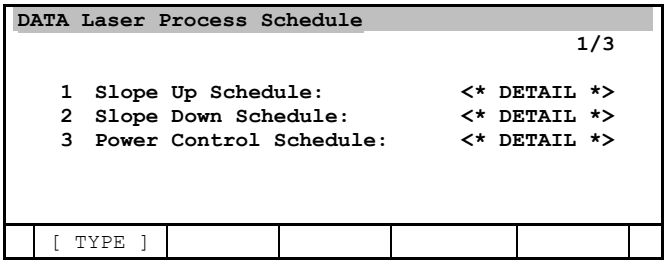
NOTE
When Stop/NonStop on LE instr is specified to NonStop, Stop time cannot be used. Specify Stop/NonStop on LE instr to Stop if you want to use Stop time.

62.9.2.2 Slope down schedule

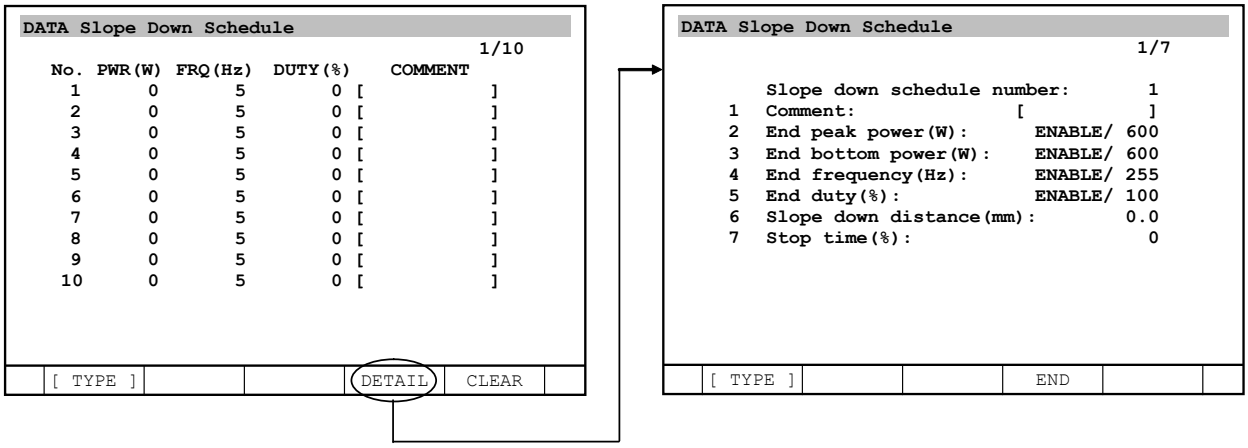
10 slopes down schedules are provided.

Procedure 62-22 Display slope down schedule screen

1 Press the [MENU] key and display the screen menu.
2 Press [0. -- NEXT --] and select [3. DATA].
3 Press F1[TYPE] and display the screen menu.
4 Select “Laser Schedule”.



5 Move the cursor to <*DETAIL*> of “Slope Down Schedule”, and press [ENTER] key.



62.9.2.3 Slope down instruction

- The format of slope up instruction is shown below.

SLOPEDOWN[*]

– * : slope down schedule number (1 - 10)

Fig. 62.9.2.3(a) Slope down instruction

- Slope down instruction is provided as the supplementary motion instruction.
- Slope down instruction can only be added to the same line as LE instruction.
- If LS, LE instruction and slope down instruction have been added to motion instruction, the slope down mode becomes active according to the schedule number specified for the slope down instruction.
- In the example below, when robot is in slope down distance, the slope down mode becomes active by SLOPEDOWN [1] of the forth line until P[4].

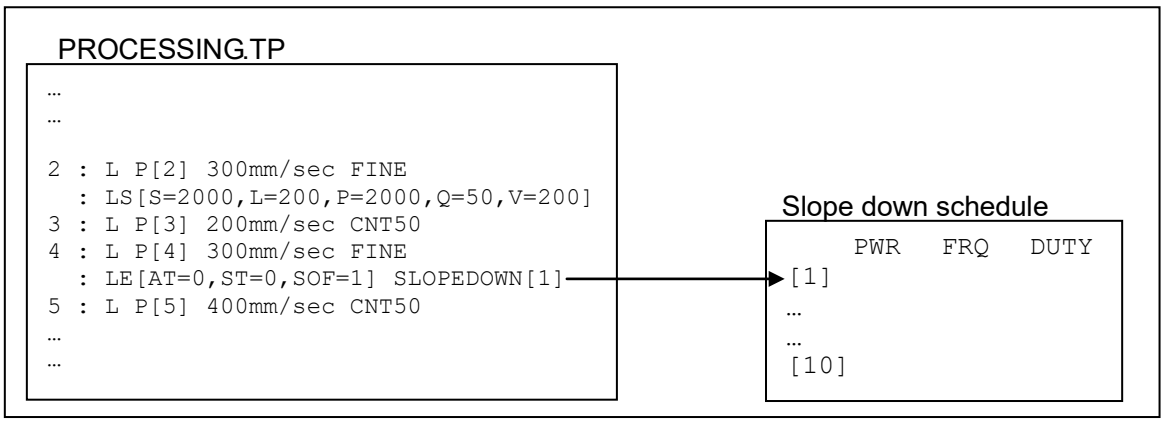


Fig. 62.9.2.3(b) Example of slope down instruction execution

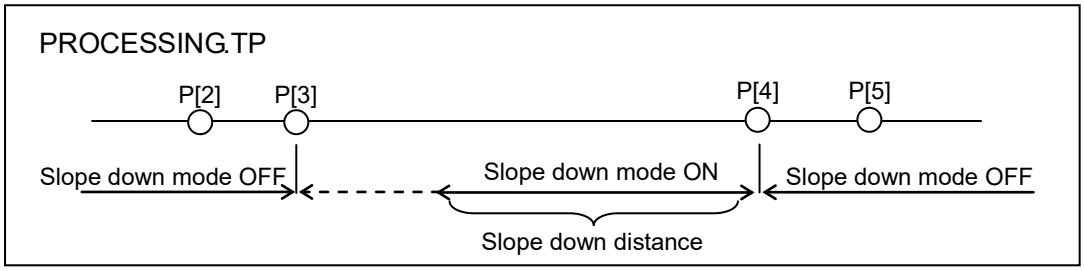
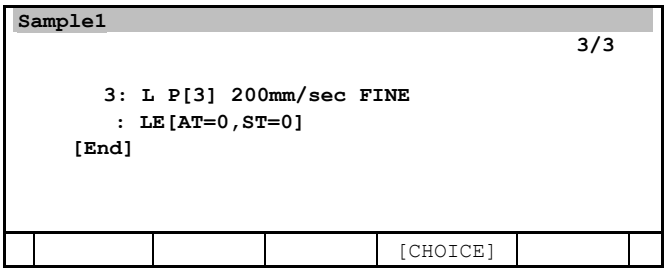
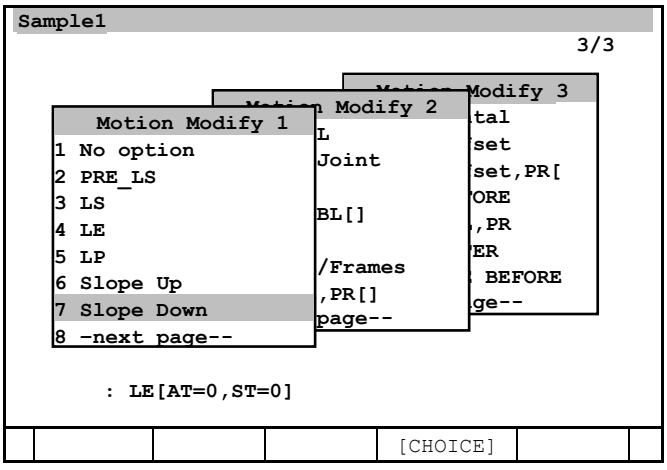


Fig. 62.9.2.3(c) Example of slope down instruction execution

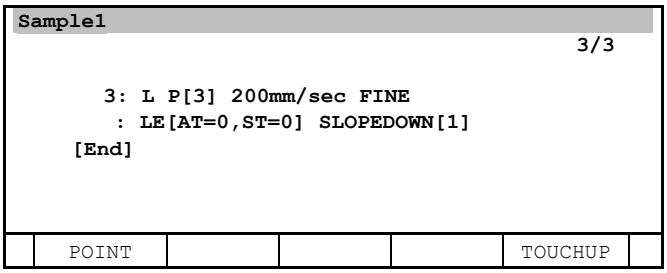
Procedure 62-23 Teaching slope down instruction

1 To teach this instruction, place the cursor behind the LS, LE instruction and press F4[CHOICE] to display the supplementary motion instruction menu.





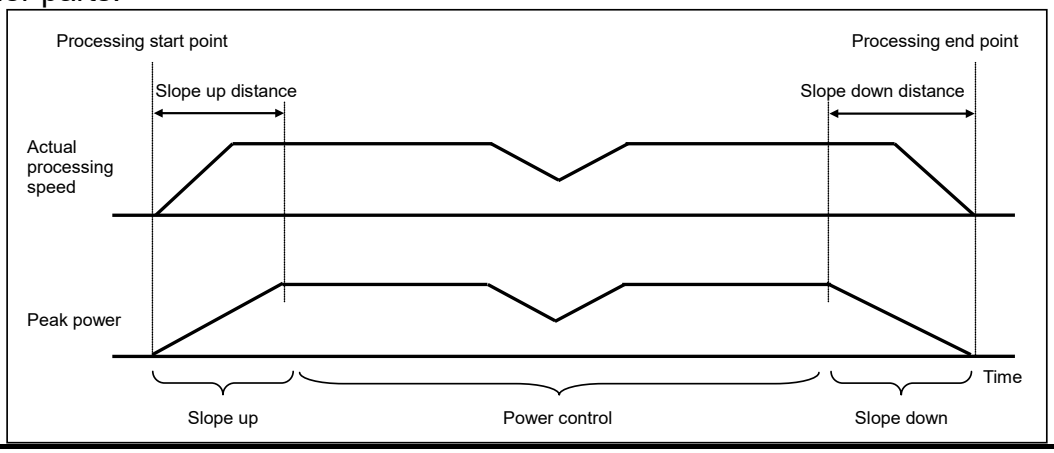
2 Select “Slope Down” from the menu.



62.9.3 Notifications for Slope Control

NOTE

When both the slope schedule and the power control schedule are specified to one program, the slope control is given to priority in the vicinity of the processing start point or the processing end point, and the power control becomes active in other parts.



NOTE

Specify either of the linear motion or the circular motion to the motion format of the processing path if you want to use slope control. You can't use slope control when the joint motion is specified to the motion format.

When the program is terminated or another line is selected, slope up/down schedule registered by slope up/down control instruction is cleared.

62.10 WIRE CONTROL

This function controls ON/OFF of wire feeder from robot. According to wire feed schedule defined beforehand by wire control instruction, wire control is instructed to turn ON by LS instruction and turn OFF by LE instruction or PAUSE/END of the program.

When "Wire feed OK" signal is assigned, "Wire feed OK" signal is start to be checked with LS instruction. The check continues during laser processing.

In case that beam is interlocked or that "Process disable for restart" is disabled, wire control doesn't function.

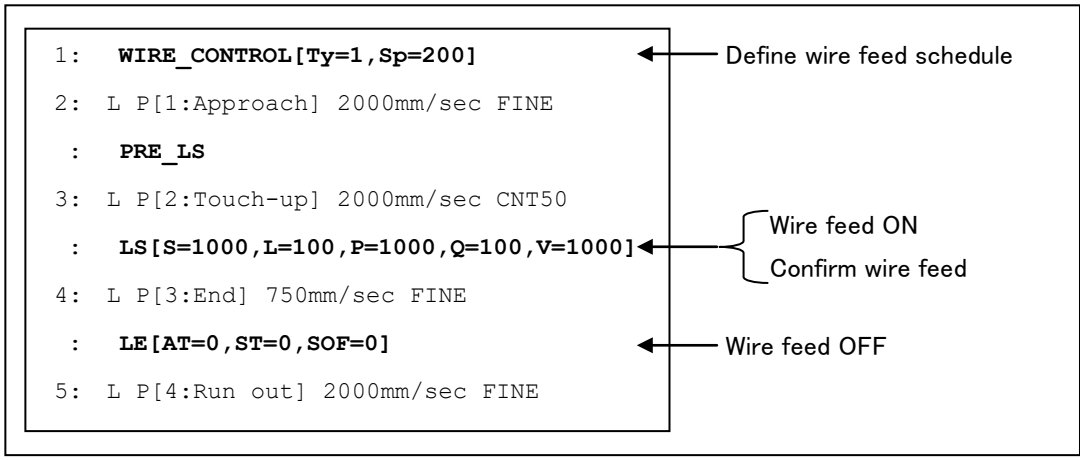


Fig. 62.10 Sample program of wire control instruction

62.10.1 Wire Control Instruction

- The format of wire control instruction is shown below.

WIRE_CONTROL[Ty=*, Sp=*]

– Ty : Wire feed type (0 - 2)

– Sp : Wire feed speed (1 - 32767)

Fig. 62.10.1(a) Wire control instruction

- Teach wire control instruction before LS instruction.
- The way to feed wire is assigned to wire feed type (Ty=).
  Ty=0 **:** OFF is output to "Wire feed ON" signal, and 0 is output to "Wire feed speed" signal.
  Ty=1 **:** It's feeding with constant speed type. "Wire feed ON" signal is turned ON, and the value assigned to wire feed speed (Sp=) is output as "Wire feed speed" signal. Even if actual processing speed changes, wire feed speed doesn't change.
  Ty=2 **:** It's feeding with speed follow type. "Wire feed ON" signal is turned ON, and the value assigned to wire feed speed (Sp=) changes with proportion to the ratio of processing speed of LS instruction and actual processing speed, and it is output as "Wire feed speed" signal.

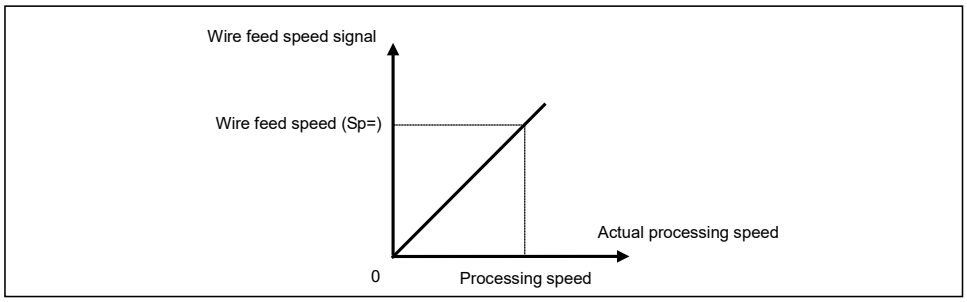
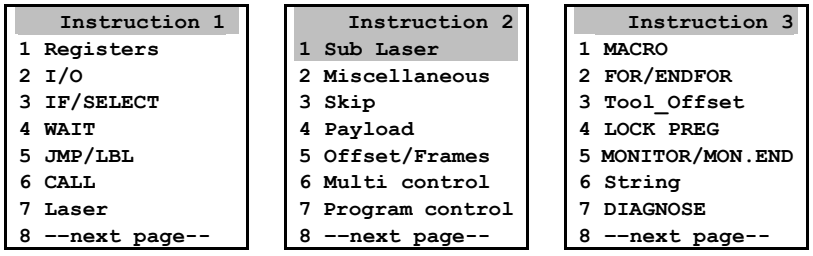


Fig. 62.10.1(b) Speed follow type

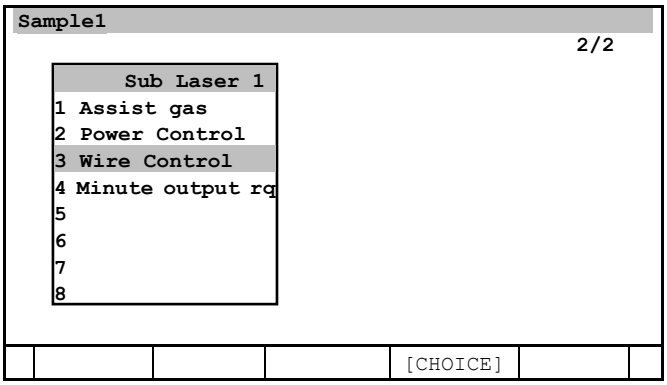
- Even if LS instruction is sent simultaneously to robot and wire feeder, robot starts to move later. To match the movement start of robot and wire feeder, please add feeding latency time to LS instruction and sent to wire feeder. By this, movement start of wire feeder can be delayed. "Delay time for wire feed" can be configured on Laser Setup screen.
  Wire feeding time should be configured on a trial basis.

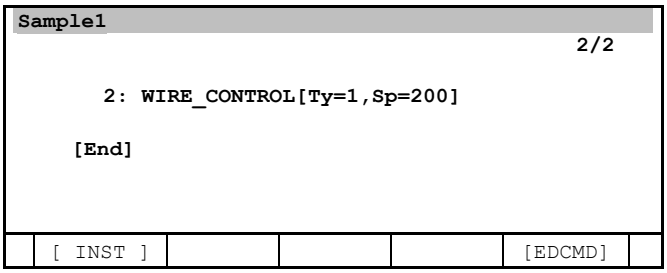
Procedure 62-24 Teaching wire control instruction

1 To teach this instruction, press F1[INST], display sub menu, and select "Sub Laser".



2 Select “Wire Control” from Sub Laser menu.





62.10.2 Wire Contol Signal

62.10.2.1 Wire feed on signal

When wire feed type (Ty=) is assigned by wire control instruction, wire feed signal turns ON with the execution of LS instruction. Wire feed signal can be assigned on Peripheral Equipment Out screen.

62.10.2.2 Wire feed speed signal

- When wire feed speed (Sp=) is assigned by wire control instruction, wire feed speed is output by “Wire feed speed” signal as Group Output with the execution of LS instruction. By receiving this signal outside, wire feeder can adjust wire feed speed automatically.
- “Wire feed speed” signal can be assigned on Peripheral Equipment Out screen.
- It is possible to output “Wire feed speed” signal as Analog Output. By setting “Wire feed speed output type” on Laser Setup screen, the output type can be changed to AOUT.
- Output range of wire feed speed analog signal is 0 to “Max val for wire feed speed AO” as output from 0 to 10-15V. For example, if you want to output wire feed speed the range of which is 0 to 200 inch/min as 0 to 10-15V, please set 200 to “Max val for wire feed speed AO” and set the value of 0~200 to Sp. “Max val for wire feed speed AO” can be changed on Laser Setup screen.

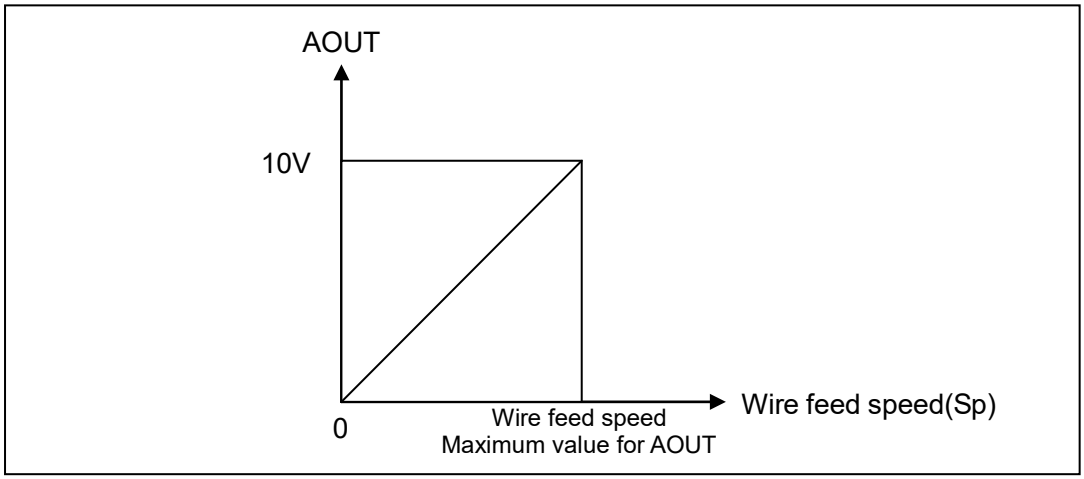


Fig. 62.10.2.2 Relationship between wire feed speed and analog signal

62.10.2.3 Wire feed OK signal

Wire shortage detector is loaded to wire feeder, and the wire shortage during feeding can be detected. Please configure this signal so that ON is input when wire is being fed correctly, and OFF is output when wire shortage is occurring. If this signal is ON during laser processing, the process continues.
"Wire feed OK" signal can be assigned on Peripheral Equipment Input screen.

62.11 MINUTE OUTPUT MODE

NOTE
Depending on the type of laser oscillator, it is possible to command 500W or less in normal output mode. Contact the service whether your laser oscillator can command 500W or less in normal output mode.

There are a normal output mode and a minute output mode in the laser oscillator.
When the laser oscillator is a normal output mode, the power range of the laser oscillator is "10% of declared power - declared power". For instance, when the laser oscillator is FF5000*i*-A, the power range is "500W - 5000W". The laser oscillator is normal output mode when the minute output state signal is OFF.
When the laser oscillator is a minute output mode, the power range of the laser oscillator is "0W - 500W". Even when using a high power laser, the minute output mode enables stable output of low laser power, suitable for laser marking. When the minute output request signal is ON, the laser oscillator becomes a minute output mode. Use the minute output request instruction to turn ON the minute output request signal. The laser oscillator is minute output mode when the minute output state signal is ON.

NOTE
When the laser oscillator is FF500 *i* -A, the laser oscillator is always in the normal output mode (minor output mode does not exist). Therefore, when you use the FF500 *i* -A, you do not need to request a minute output mode.

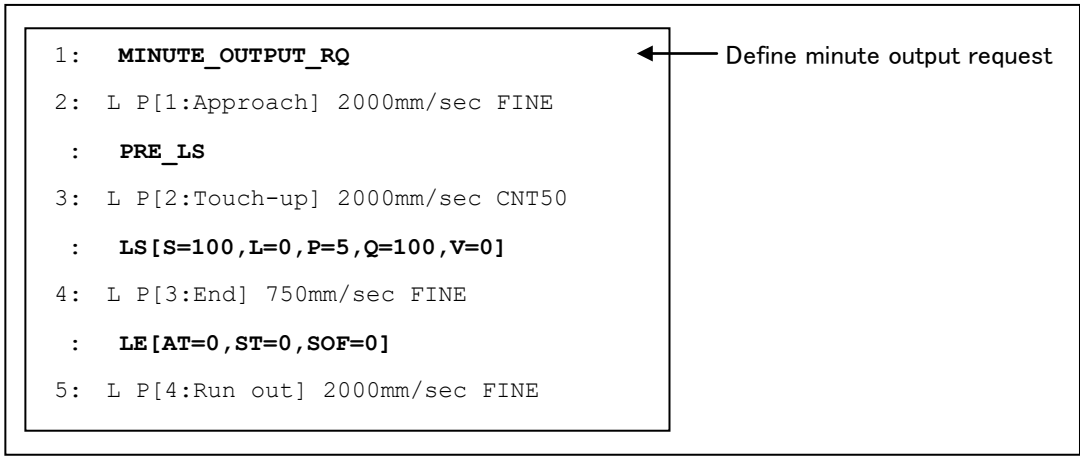


Fig. 62.11 (a) Sample program of minute output request instruction

62.11.1 Minute Output Request Instruction

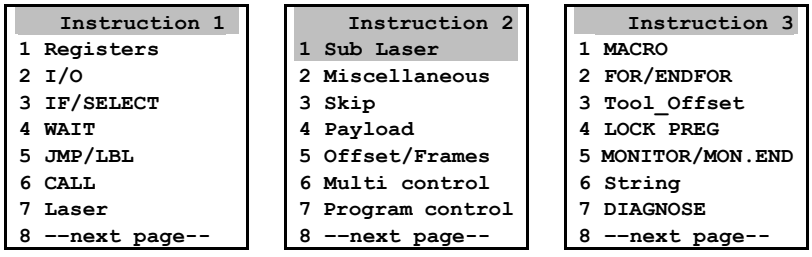
- The format of the minute output request instruction is shown below.

Fig. 62.11.1 (a) Minute output request instruction

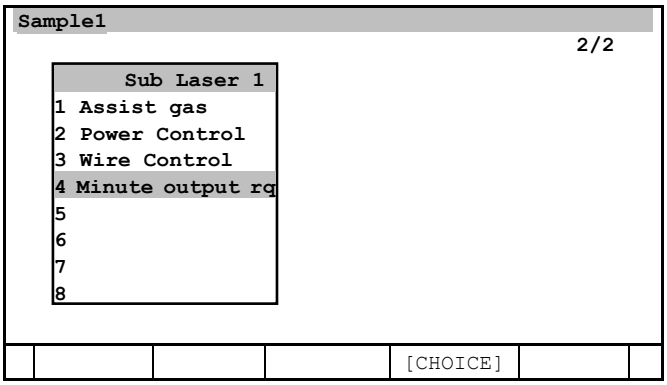
- Teach PRE_LS instruction before LS instruction.
- Minute output request instruction turns ON the minute output request signal. From this, laser oscillator is already minute output mode when LS instruction is executed.
- The minute output request signal is automatically turned OFF when the LE instruction is executed, and the laser oscillator returns to the normal output mode.
- When the program is terminated or another line is selected, the minute output request signal is automatically turned OFF, and the laser oscillator returns to the normal output mode.

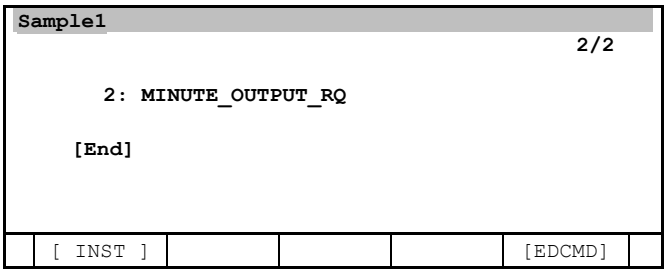
Procedure 62-25 Teaching minute output request instruction

1 To teach this instruction, press F1[INST], display sub menu, and select “Sub Laser”.



2 Select “Minute output rq” from Sub Laser menu.





62.12 DIRECT POWER OVERRIDE MODE

62.12.1 Direct Power Override Mode in Laser Oscillator

When the laser oscillator is in the direct power override mode, the peak power commanded by the robot changes as shown in the figure below according to the analog voltage input from the scanner controller to the laser oscillator.

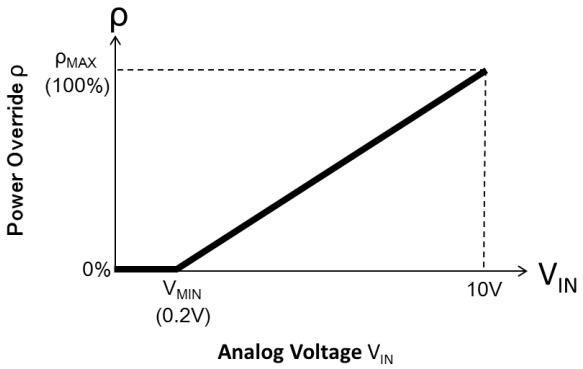


Fig. 62.12.1(a) Relationship between power override and analog voltage

NOTE
The V_{MIN} margin (0.2V) is provided to prevent unintentional laser output due to noise at the 0V input.

For example, when V_{IN} is 5V and original peak power command is 4000W, power override ρ is 49% and final peak power command is 1960W.

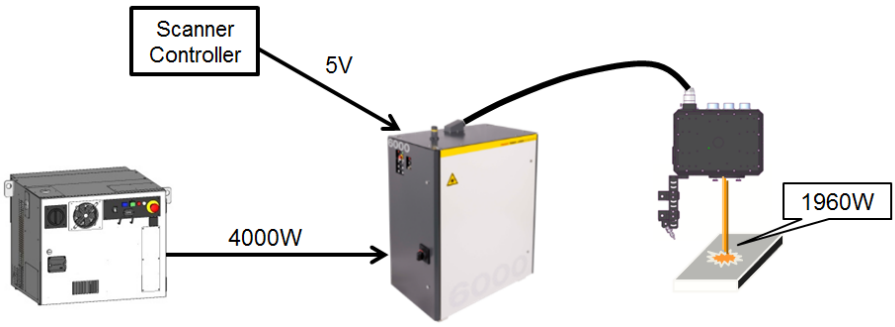


Fig. 62.12.1(b) Direct power override mode in laser oscillator

For details on the direct power override mode of the laser oscillator, refer to the instruction manual for the laser oscillator.

62.12.2 Direct Power Override Mode in Robot

When the robot is in the direct power override mode, the scanner ON signal and scanner program number signal can be output to the scanner controller in addition to outputting the processing schedules to the laser oscillator.
Use the LS_PO instruction to output the scanner ON signal and scanner program number signal. Refer to the subsection "LS_PO Instruction" below.
The scanner ON signal and scanner program number signal can be assigned on the peripheral equipment output screen.

Peripheral equipment out screen
[Robot➔Peripheral equipment]

```
I/O Peripheral Equipment Out
                                                    1/9
     NAME                  OUT PT    SIM      STATE
  1 Stand-by st:           DO[  0]   U        ***
  2 Laser ready st:        DO[  0]   U        ***
  3 Interlock OFF st:      DO[  0]   U        ***
  4 Guide light ON st:     DO[  0]   U        ***
  5 Assist gas start:      DO[  0]   U        ***
  6 Wire feed ON:          DO[  0]   U        ***
  7 Wire feed speed:       GO[  0]   U        *****
  8 Scanner ON:            DO[  0]   U        ***
  9 Scanner prog num:      GO[  0]   U        *****

[ TYPE ]  CH +  IN/OUT
```

Table 62.12.2 (a) Peripheral equipment output screen (robot➔peripheral equipment)

Name of signal	Explanation
Scanner ON	This signal instructs the scanner controller to start scanning. It turns ON when the LS_PO instruction is executed.
Scanner prog num	This signal is used to select the scanner program held by the scanner controller from the robot. When the LS_PO instruction is executed, the scanner program number specified in the argument SPN is output.

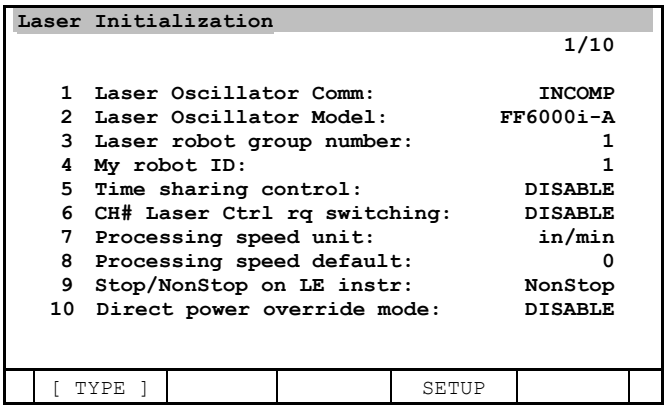
62.12.3 Enable/Disable Direct Power Override Mode

When using the direct power override mode, enable the direct power override mode on the laser initial setup screen.

Procedure 62-26 Enable/Disable direct power override mode

Step

1 Perform controlled start.
2 After controlled start, press the [MENU] key. Controlled start menu will be displayed.
3 Select [0 – NEXT --]. Select [4 Laser Initial Setup].



4 Enable “Direct power override mode”. When the direct power override mode is enabled, the following functions are available.

- The scanner ON signal and the scanner program number signal are displayed on the peripheral equipment output screen.
- The argument PO is displayed in the PRE_LS instruction.
- The LS_PO instruction becomes available.

62.12.4 Switching to Direct Power Override Mode by PRE_LS

- The format of the PRE_LS instruction is shown below.

PRE_LS[PO=*]

– PO : Direct Power Override Mode (0 - 1)

Fig. 62.12.4(a) PRE_LS instruction

- When the PRE_LS[PO=0] instruction is executed, the robot and laser oscillator switch to the normal mode.
- When the PRE_LS[PO=1] instruction is executed, the robot and laser oscillator switch to the direct power override mode.
- Execute the PRE_LS[PO=1] instruction before executing the LS_PO instruction.
- When the LE instruction is executed, the robot and laser oscillator switch to normal mode.

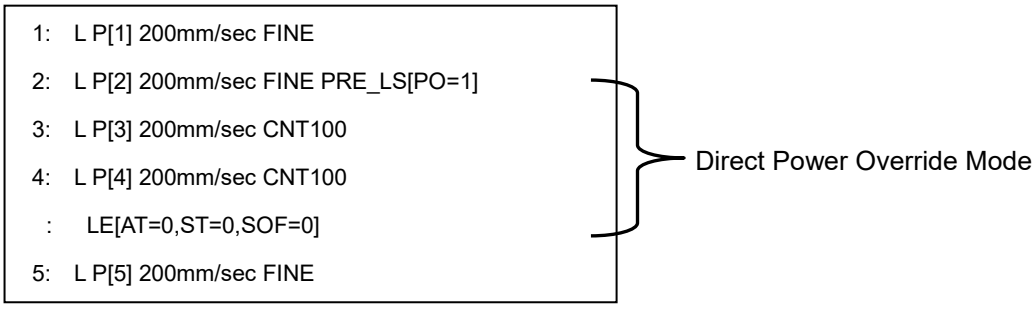


Fig. 62.12.4(b) Sample program for PRE_LS[PO=1] instruction

62.12.5 LS_PO Instruction

- The format of the LS_PO instruction is shown below.

LS_PO[S=*, L=*, P=*, Q=*, SPN=*, TA=*]

- S : Peak power(W)
- L : Pulse bottom power(W)
- P : Pulse frequency(Hz)
- Q : Pulse duty(%)
- SPN : Scanner program number (0 - 65535)
- TA : Time adjustment(-99999msec – 99999msec)

Fig. 62.12.5(a) LS_PO instruction

- The LS_PO instruction can be executed only in the direct power override mode. Execute the PRE_LS [PO=1] instruction before executing the LS_PO instruction.
- The LS_PO instruction is executed using a CALL instruction or a PR_CALL instruction.
- When the LS_PO instruction is executed using the CALL instruction, the scanner ON signal and scanner program number signal are output at the same time as the processing schedules (S, L, P and Q). The time adjustment of the scanner ON signal and scanner program number signal by the argument TA cannot be performed.

```
1:  L P[1] 200mm/sec FINE
2:  L P[2] 200mm/sec FINE PRE_LS[PO=1]
3:  L P[3] 200mm/sec CNT100
4:  CALL LS_PO[S=4000,L=0,P=5,Q=100,SPN=55,TA=0]
5:  L P[4] 200mm/sec CNT100
6:  L P[5] 200mm/sec CNT100
 :   LE[AT=0,ST=0,SOF=0]
```

Fig. 62.12.5(b) Sample program for CALL LS_PO instruction

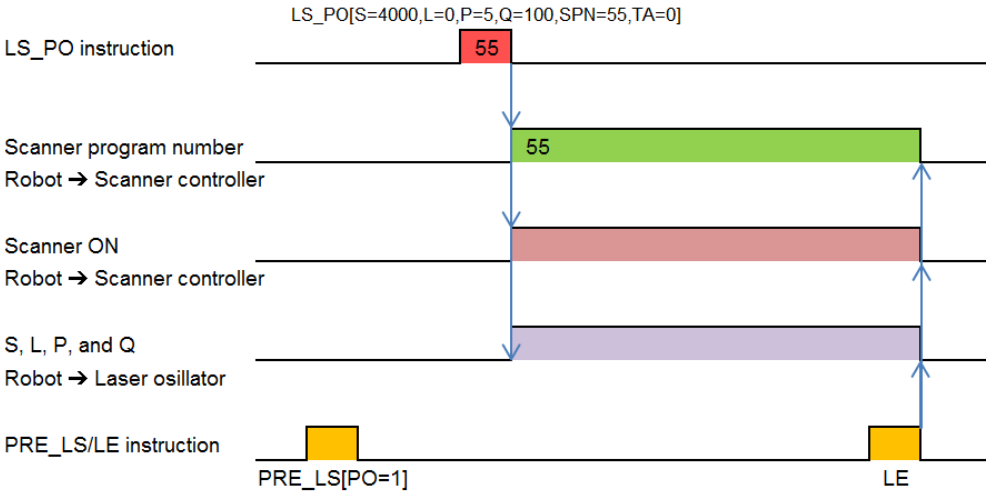


Fig. 62.12.5(c) Time chart when using CALL LS_PO instruction

- When the LS_PO instruction is executed using the PR_CALL instruction, the output timing of the scanner ON signal and scanner program number signal can be adjusted by the argument TA in addition to outputting the processing schedules (S, L, P and Q) to the laser oscillator.
  If the argument TA is a negative value, the scanner ON signal and scanner program number signal will be output in advance for the specified time based on the teaching position or the closest point on the motion path. If the argument TA is a positive value, the scanner ON signal and scanner program number signal will be output delayed by the specified time, based on the teaching position or the closest point on the motion path.

```
1:  L P[1] 200mm/sec FINE PRE_LS[PO=1]
2:  L P[2] 200mm/sec CNT100
3:  L P[3] 200mm/sec CNT100
 :    PR_CALL LS_PO[S=5000,L=0,P=5,Q=100,SPN=10,TA=-200]
4:  L P[4] 200mm/sec CNT100
 :    PR_CALL LS_PO[S=6000,L=0,P=5,Q=100,SPN=25,TA=150]
5:  L P[5] 200mm/sec CNT100
 :    LE[AT=0,ST=0,SOF=0]
```

Fig. 62.12.5(d) Sample program for PR_CALL LS_PO instruction

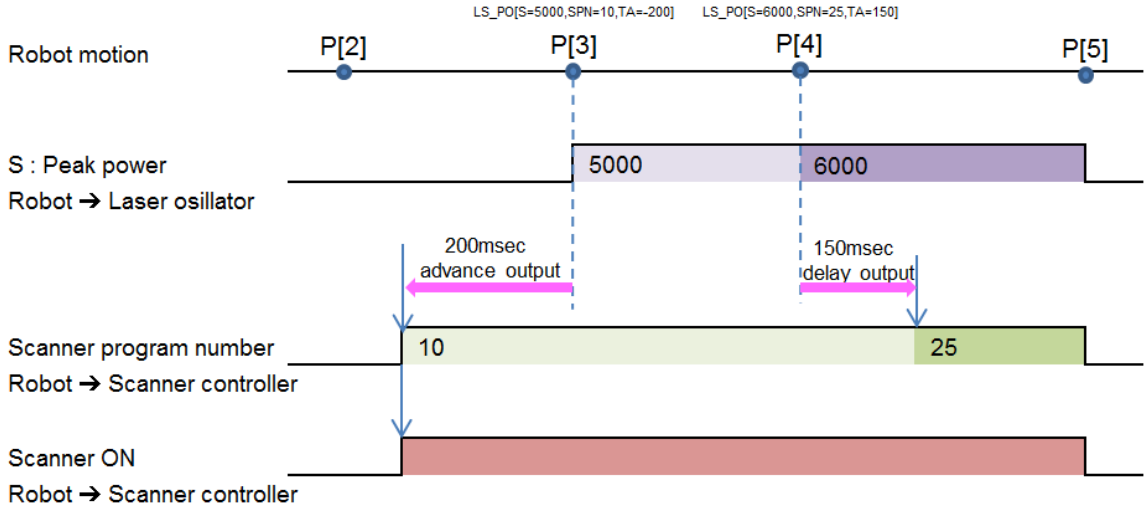
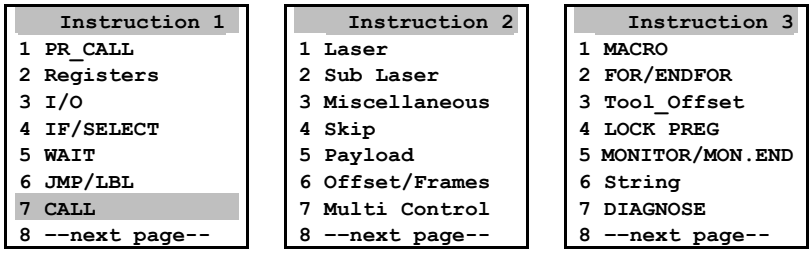


Fig. 62.12.5(e) Time chart when using PR_CALL LS_PO instruction

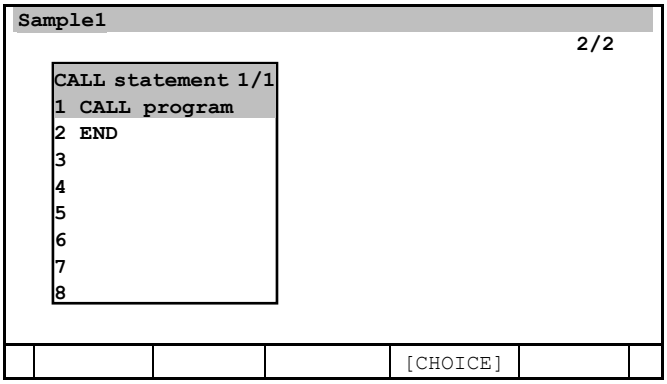
- When the LE instruction is executed, the scanner ON signal and scanner program number signal are turned off.

Procedure 62-27 Teaching CALL LS_PO instruction

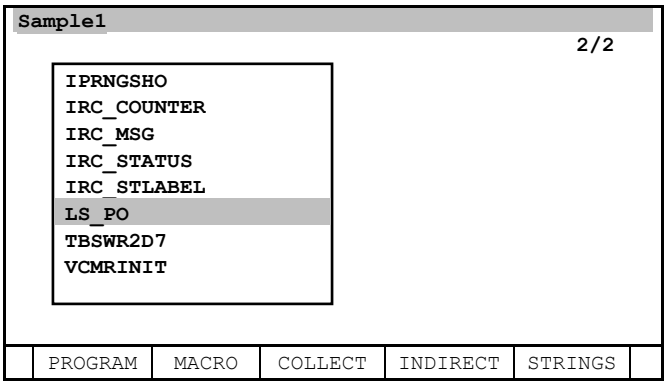
1 To teach this instruction, press F1[INST], display sub menu, and select “CALL”.

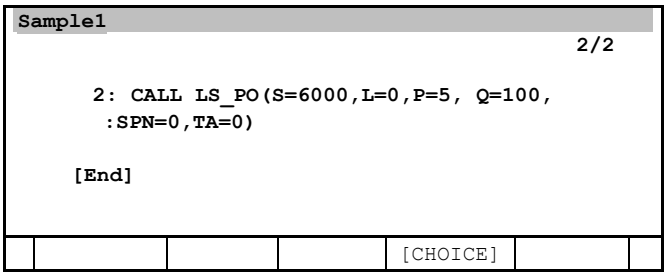


2 Select “CALL program” from CALL statement menu.



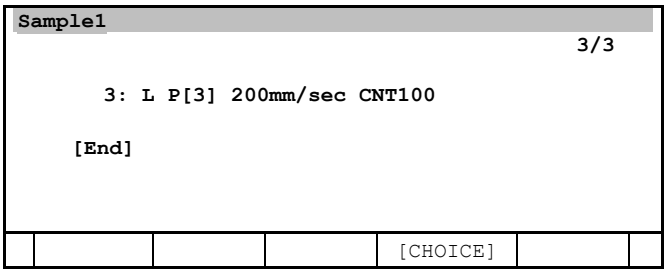
3 Press F3 ”COLLECT” and select ”KAREL Progs”. A list will be displayed. Select LS_PO.

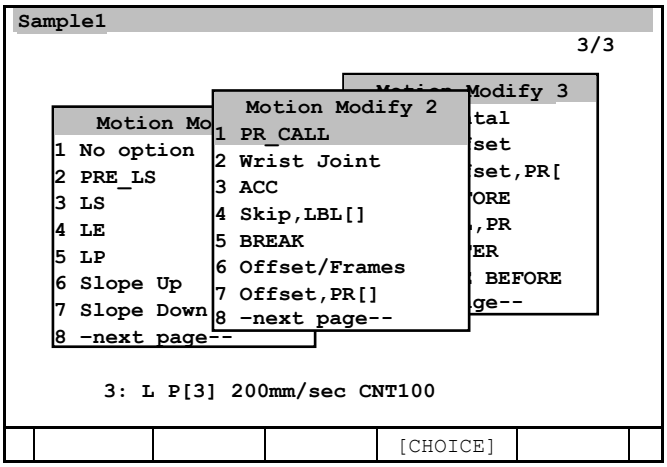




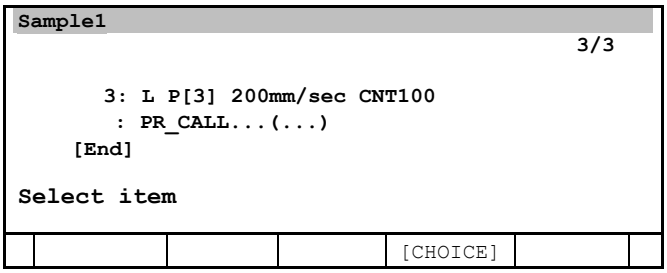
Procedure 62-28 Teaching PR_CALL LS_PO instruction

1 Place the cursor behind the motion instruction and press F4 [CHOICE] to display the additional motion instruction menu.

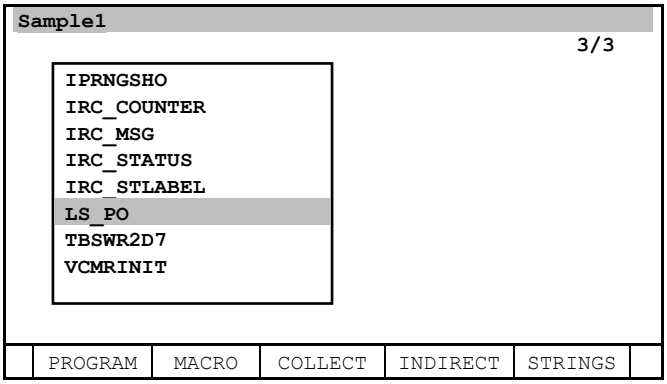


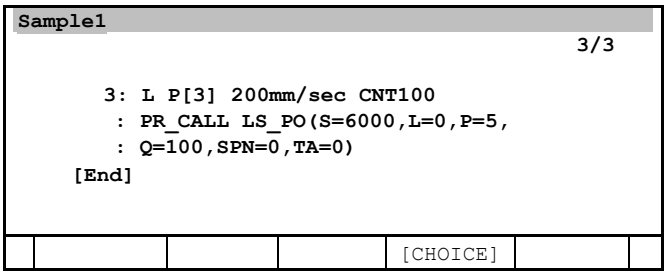


2 Select “PR_CALL” from the motion instruction menu.



3 Press F4 [CHOICE] then F3 ”COLLECT” and select ”KAREL Progs”. A list will be displayed. Select LS_PO.





62.13 LS_TA INSTRUCTION WITH TIME ADJUSTMENT

62.13.1 LS_TA Instruction

- The format of the LS_TA instruction is shown below.

LS_TA[S=*, L=*, P=*, Q=*, V=*, TA=*]

- S : Peak power(W)
- L : Pulse bottom power(W)
- P : Pulse frequency(Hz)
- Q : Pulse duty(%)
- V : Processing speed(mm/sec, cm/min, inch/min)
- TA : Time adjustment(-99999msec – 99999msec)

Fig. 62.13.1(a) LS_TA instruction

- Execute the PRE_LS instruction before executing the LS_TA instruction.
- The LS_TA instruction is executed using a PR_CALL instruction.

NOTE
- If there is no PRE_LS instruction before the LS_TA instruction, the LS_TA instruction will not be executed.
- The LS_TA instruction cannot be executed when the direct power override mode is enabled.
- You cannot use the CALL instruction to execute the LS_TA instruction.

- When the PR_CALL LS_TA instruction is executed, the output timing of the processing schedules (S, L, P, Q and V) can be adjusted by the argument TA.
  If the argument TA is a negative value, the processing schedules (S, L, P, Q and V) will be output in advance for the specified time based on the teaching position or the closest point on the motion path.
  If the argument TA is a positive value, the processing schedules (S, L, P, Q and V) will be output delayed by the specified time, based on the teaching position or the closest point on the motion path.

```
2:  L P[2] 200mm/sec FINE PRE_LS
3:  L P[3] 200mm/sec CNT100
 :    PR_CALL LS_TA[S=5000,L=0,P=5,Q=100,V=200,TA=-20]
4:  L P[4] 200mm/sec CNT100
 :    PR_CALL LS_TA[S=6000,L=0,P=5,Q=100, V=200,TA=15]
5:  L P[5] 200mm/sec CNT100
```

Fig. 62.13.1(b) Sample program for PR_CALL LS_TA instruction

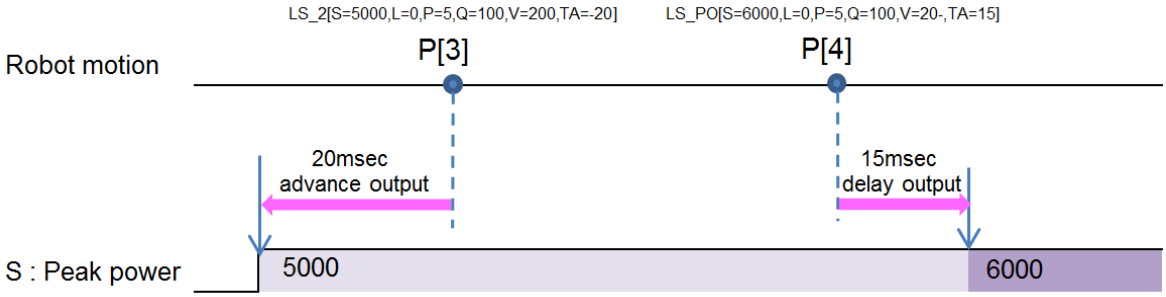


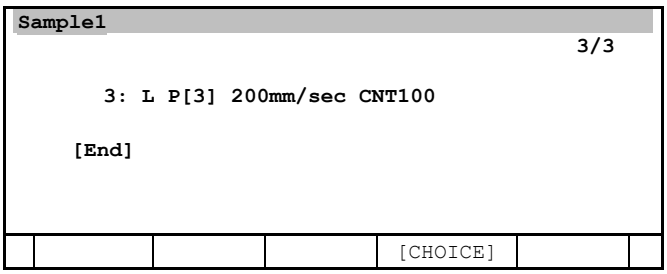
Fig. 62.13.1(c) Time chart when using PR_CALL LS_TA instruction

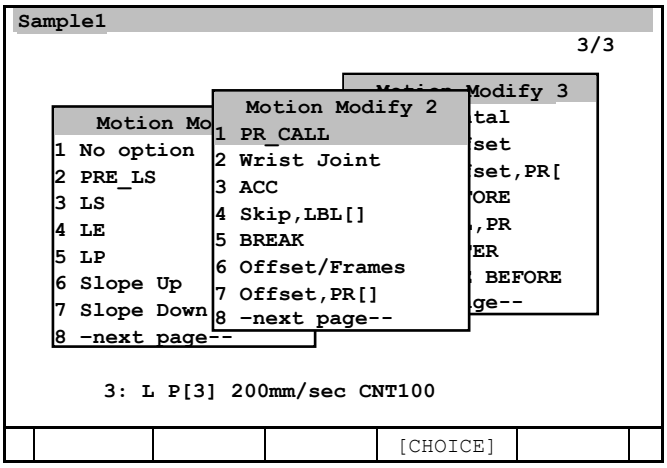
- When the “Delay time for beam ON” on the laser setup screen is set, the output timing of the processing schedules (S, L, P, Q and V) is the sum of the “Delay time for beam ON” and the argument TA. For example, if the “Delay time for beam ON” is 15 msec and the argument TA is -20 msec, the processing schedules (S, L, P, Q and V) will be output -5 msec ahead.
- When the “LSTA start time adjustment” on the laser setup screen is set, the output timing of the processing schedules (S, L, P, Q and V) is the sum of the “LSTA start time adjustment” and the argument TA. For example, if the “LSTA start time adjustment” is -24 msec and the argument TA is -6 msec, the processing schedules (S, L, P, Q and V) will be output -30 msec ahead.

NOTE
- Slope-up and slope-down instructions cannot be used with the PR_CALL LS_TA instruction.
- PR_CALL LS_TA instruction supports only G1. If there are multiple groups in the TP program, PR_CALL LS_TA instruction works only for G1. If G1 does not exist in the TP program, PR_CALL LS_TA instruction will have no work.

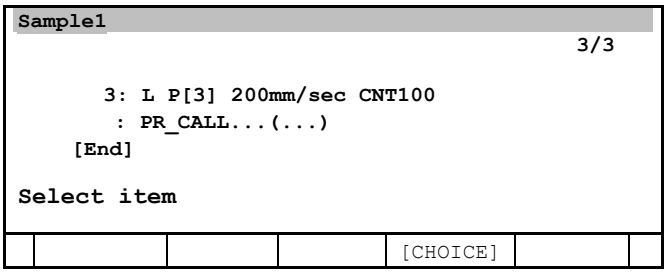
Procedure 62-29 Teaching PR_CALL LS_TA instruction

1 Place the cursor behind the motion instruction and press F4 [CHOICE] to display the additional motion instruction menu.

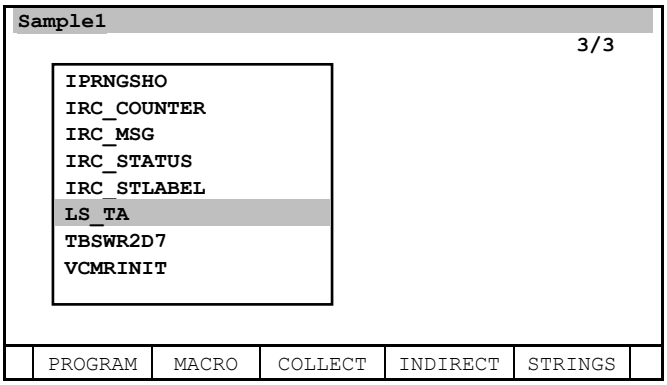


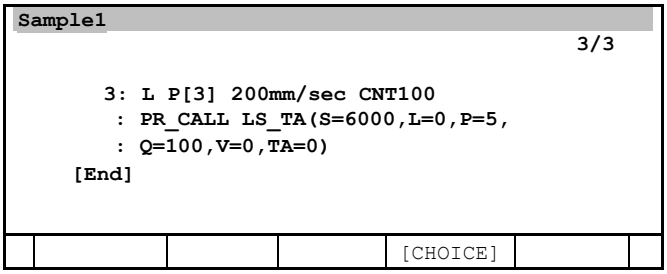


2 Select “PR_CALL” from the motion instruction menu.



3 Press F4 [CHOICE] then F3 ”COLLECT” and select ”KAREL Progs”. A list will be displayed. Select LS_TA.





62.14 LASER STATUS SCREEN

This function provides the graphical display of the laser processing status and laser oscillator status on teach pendant screen of robot.

Procedure 62-30 Display laser status screen

1 Press the [MENU] key and display the screen menu.
2 Press [0. -- NEXT --] and select [4. STATUS].
3 Press F1[TYPE] and display the screen menu.
4 Select “LASER STATUS”.

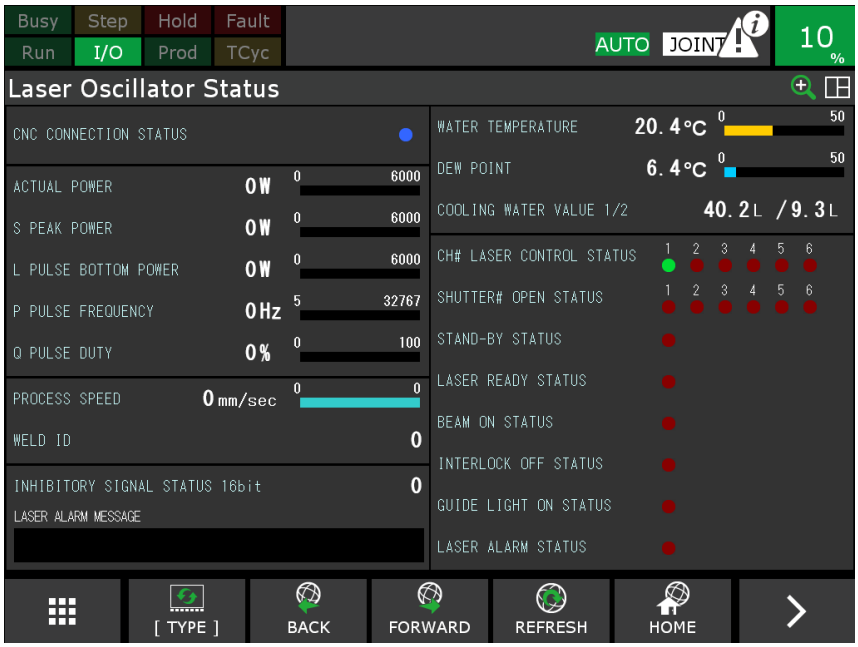
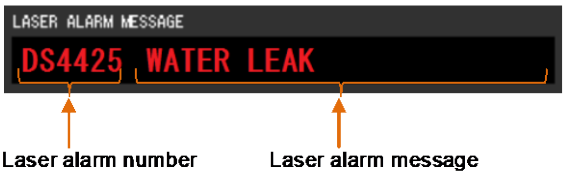
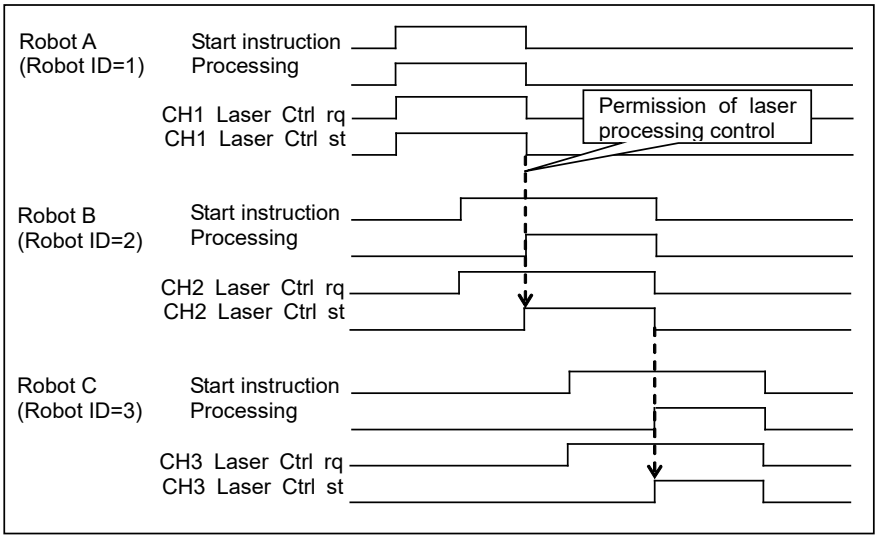


Table 62.14 Laser status screen

Items	Descriptions
CNC CONNECTION STATUS	Show the connection status with laser oscillator ● : Connected. ● : Not connected. Please reconfigure the transmission setting.
ACTUAL POWER	Show actual power of beam output.
S PEAK POWER	Show the result of S peak power request.
L PULSE BOTTOM POWER	Show the result of L pulse bottom power request.
P PULSE FREQUENCY	Show the result of P pulse frequency request.
Q PULSE DUTY	Show the result of Q pulse duty request.
PROCESS SPEED	Show actual speed during laser processing
WELD ID	Not in use.
INHIBITORY SIGNAL STATUS 16bit	Not in use.

Items	Descriptions
LASER ALARM MESSAGE	If the laser alarm from laser oscillator is detected, laser alarm number and laser alarm message is displayed here. Laser alarm number and laser alarm message is got from laser oscillator. Please refer to the laser oscillator manual for the details of the laser alarm. 
WATER TEMPERATURE	Show the temperature of cooling water. : Low temperature : Medium temperature : High temperature Please refer to the laser oscillator manual for the details of water temperature.
DEW POINT	Show the dew point when the inside air is exposed to the cooling water. : Low temperature : Medium temperature : High temperature Please refer to the laser oscillator manual for the details of dew point.
COOLING WATER VALUE 1/2	Show the quantity of cooling water. Please refer to the laser oscillator manual for the details of cooling water value 1/2.

Items	Descriptions
CH# LASER CONTROL STATUS （In case of connecting several robots）	**NOTE** ・ CH stands for channel. ・ # is the same number as the Robot ID. It is not possible to perform laser processing with some robots at the same time by using one laser oscillator. Therefore, when laser oscillator is received “CH# Laser Ctrl rq” signal during laser processing of other robot, laser oscillator waits until processing of other robot is finished. After laser processing of other robot, laser oscillator turns this signal ON to provide the permission of laser processing control.  If “Time Sharing Control” on laser initial setup screen is disabled, “CH# Laser Ctrl rq” signal is always ON. Therefore, “CH# Laser Ctrl st” signal is also ON. Normally, it is not necessary to be conscious of this signal.
SHUTTER# OPEN STATUS （In case of connecting several robots）	**NOTE** # is the same number as the Robot ID. This signal indicates which shutter is currently open. Normally, the “Shutter# open rq” signal is ON, and this signal turns ON. However, if you do not have the laser control, this signal will remain OFF even if the “Shutter# open rq” signal is ON. When the fiber selector is not installed, the state of the “Shutter# open rq” signal and the state of the “Shutter# open st” signal are always in agreement. Normally, you do not need to be conscious of this signal.
STAND-BY STATUS	Normally, this signal turns ON by “Laser ready rq” signal turning ON. This signal indicates that the status of laser oscillator is Stand-by status.
LASER READY STATUS	Normally, this signal turns ON by “Laser ready rq” signal turning ON. This signal indicates that the status of laser oscillator is Laser Ready status.
BEAM ON STATUS	This signal become ON when the beam is output actually.
INTERLOCK OFF STATUS	Normally, this signal is turned ON when “Interlock OFF rq” signal is turned ON. When this signal is ON, beam is output. However, when “Stand-by st” signal is OFF, this signal is remains OFF. When this signal is OFF, beam is interlocked and “Assist gas start”, ”Laser ready rq”, “Wire feed ON” and “Wire feed speed” signal is not turned ON in PRE_LS and LS instruction. In addition, it is not confirmed that “Assist gas ready”, “Laser ready st”, “Stand-by st” and “Wire feed OK” signal are turned ON.

Items	Descriptions
GUIDE LIGHT ON STATUS	Normally, this signal turns ON by “Guide light ON rq” signal turning ON. When this signal is ON, guide light turns on. However, when “Interlock OFF st” signal is ON, this signal remains OFF. Please use this during teaching.
LASER ALARM STATUS	Indicate that the laser alarm from laser oscillator is detected. Please confirm the content of alarm on alarm history screen, and remove the cause. You can turn OFF this signal by turning ON “Laser reset rq” signal. Refer also to the section “Laser Alarm” below.

62.15 PROCESSING STATE RECORD FUNCTION

Processing status record function is the function that records the following data during laser processing.

Table 62.15 (a) Recorded data

Items	Descriptions
TIME(ms)	Time
POWER(W)	Peak power sent to laser oscillator
BOTTOM_POWER(W)	Pulse bottom power sent to laser oscillator
FREQUENCY(Hz)	Frequency sent to laser oscillator
DUTY(%)	Duty sent to laser oscillator
ACTUAL_POWER(W)	Actual power of beam output received from laser oscillator
PROCESS_SPEED	Actual processing speed
FEED_SPEED	Actual wire feed speed

- These data are recorded every ITP period. And they are output to internal device of robot controller when LE instruction is executed.
- The output file is made with the format below.

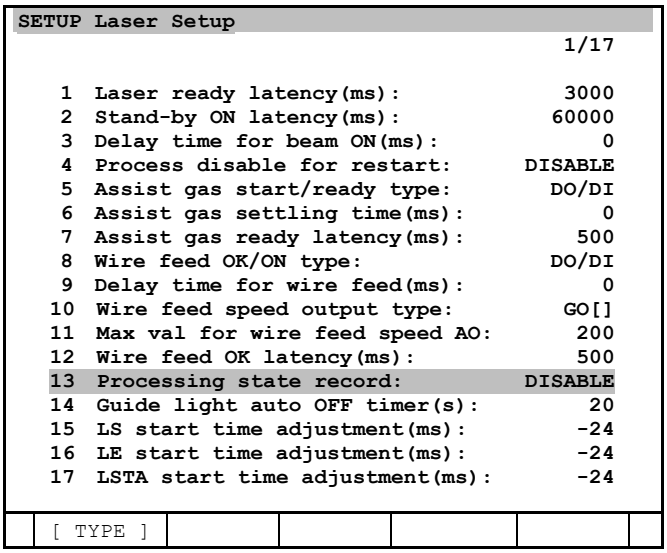
```
TIME(ms), POWER(W), BOTTOM_POWER(W), FREQUENCY(Hz), DUTY(%), ACTUAL_POWER(W), PROCESS_SPEED, FEED_SPEED
8, 2000, 1000, 500, 100, 1999,    200.00, 0
16, 2000, 1000, 500, 100, 1998,    200.00, 0
24, 2000, 1000, 500, 100, 2000,    200.00, 0
```

Fig. 62.15 Format

Procedure 62-31 Processing state record function

Step

1 Press the [MENU] key then screen menu will be displayed.
2 Select [6 SETUP].
3 Show Menu by pressed F1 [TYPE].
4 Select “Laser Setup”.
5 Move cursor on <* DETAIL *> of “Laser setup” and press [ENTER] key.



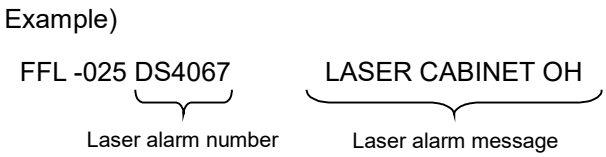
6 Move the cursor to the item of the processing state record, and change it to ENABLE.
7 Execute the laser processing. When the laser processing is finished, the file is output to an external device, and the item of the processing state record become FALSE automatically.
- The file is output to the external device directory that is selected on file screen.
- The file name that is restored to the external device is shown below.

FLPRCLOGaabbcc.DT (aa: hour, bb: minute, cc: second)

62.16 ALARM

62.16.1 Laser Alarm

If “Laser alarm st” signal turns ON, this function immediately causes “FFL-025 %s”. Here, the laser alarm sent by laser oscillator is displayed at %s. Laser alarm consists of laser alarm number and laser alarm message.



If “FFL-025 %s” occurs during laser processing, the alarm severity is PAUSE. In other cases, the alarm severity is WARN.

62.16.1.1 Display in status window

Laser alarm can be confirmed on status window of the teach pendant.

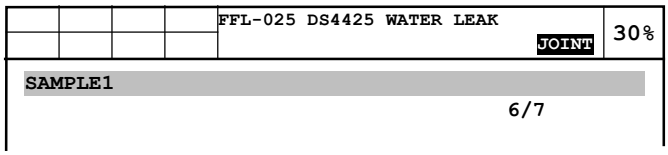


Fig. 62.16.1.1 (a) Display of laser alarm

62.16.1.2 Display in the active alarm screen

Laser alarm can be confirmed on the active alarm screen. “FFL-025 %s” occurs together with “FFL-008 Received laser alarm”.

Procedure 62-32 Display the active alarm screen

1 Press the [MENU] key and display the screen menu.
2 Select [4. ALARM], and the active alarm screen is displayed.
When the alarm occurs, the active alarm screen is displayed automatically.

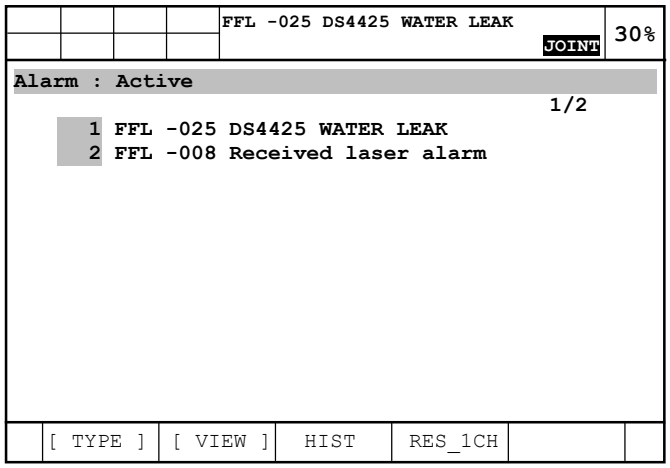


Fig. 62.16.1.2 Display of laser alarme

62.16.1.3 Display in laser status screen

Laser alarm can be confirmed on LASER STATUS screen.
Refer to the section “LASET STATUS SCREEN” above.

62.16.1.4 Release and recover of laser alarm

- When laser alarm occurs, please remove the cause of the alarm of laser oscillator side with reference to laser alarm number and laser alarm message.
- To release laser alarm, please turn ON the “Laser reset rq” signal. “Laser reset rq” signal is pulsed output by the pressing RESET key of teach pendant or the input of FAULT_RESET UI[5].
- Once laser alarm occurs, the status of laser oscillator becomes back to the initial status (“Laser ready rq” signal and “Stand-by st” signal are OFF). To restart laser processing, it is necessary to release the laser alarm, and then turn OFF→ON “Laser ready rq” signal.

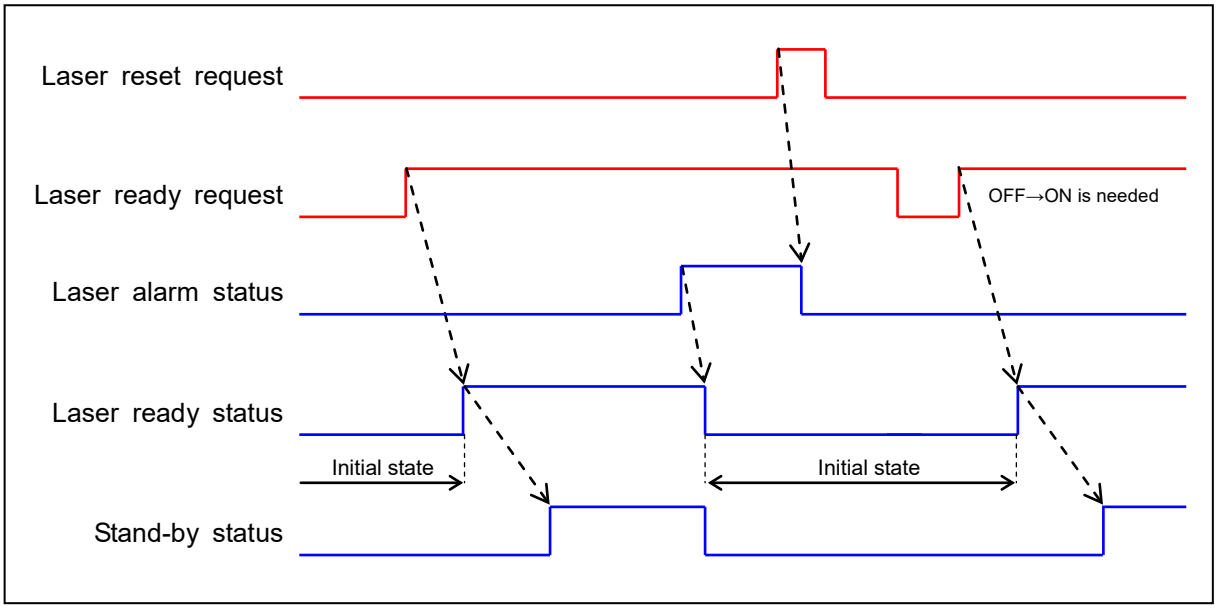


Fig. 62.16.1.4 (a) Laser sequence

NOTE
- As for “Laser reset rq” signal and “Laser ready rq” signal, please additionally refer to Subsection 62.5.1.1 “Laser Oscillator Output Screen” and Beam interlock OFF request signal in Subsection 62.5.1.1 “Laser Oscillator Output Screen”.
- As for “Laser alarm st” signal, “Laser ready st” signal and “Stand-by st” signal, please additionally refer to Subsection 62.5.1.2 “Laser Oscillator Input Screen”.

62.16.2 Trouble Shooting About This Function’S Alarm

The cause and remedy of alarm/message displayed related to this function are described below.

NOTE
In the alarm code, following signs are used.
%d : decimal number
%f : floating-point number
%x : hex number
%s : character string
When the number is added after %, the number means a display width.

FFL-001 Internal error (%s)
Cause: Internal error is occurred.
Remedy: Execute Diagnostic log function [FNCT menu / Diagnostic log] before you do other operation such as power off or jogging, and get the image backup. And contact your FANUC technical representative. If you cannot execute Diagnostic log function, document the events that led to the error and get the image backup. And contact your FANUC representative.

FFL-002 Assist gas ready time out
Cause: Even after “Assist gas ready latency” time on Laser Setup screen, “Gas ready” signal didn’t become ON.
Remedy: Please check assist gas and gas valve unit (sensor).

FFL-003 Las control stat OFF time out

Cause: Even after “Laser Ctrl st ON latency” time on Time Sharing Control screen, “CH# Laser Ctrl st” signal didn’t become OFF.

Remedy: Please check if “Laser Ctrl st ON latency” is appropriate. Please set appropriate value as system.

FFL-004 Las control stat ON time out

Cause: Even after “Laser Ctrl st ON latency” time on Time Sharing Control screen, “CH# Laser Ctrl st” signal didn’t become ON.

Remedy: Please check if “Laser Ctrl st ON latency” is appropriate. Please set appropriate value as system.
Check the status of the laser oscillator. The emergency stop may not be released or the automatic test may not be completed.

FFL-005 Shutter close time out

Cause: Even after “Shutter open/close latency” time on Time Sharing Control screen, “Shutter# open st” signal didn’t become OFF.

Remedy: Please check if Shutter open/close latency time is appropriate.
Fiber selector may be broken down. Please refer to the manual of laser oscillator, and confirm the status of fiber selector.

FFL-006 Shutter open time out

Cause: Even after “Shutter open/close latency” time on Time Sharing Control screen, “Shutter# open st” signal didn’t become ON.

Remedy: Please check if Shutter open/close latency time is appropriate.
Fiber selector may be broken down. Please refer to the manual of laser oscillator, and confirm the status of fiber selector.
The time sharing control on the laser initial setup screen may be disabled even though the fiber selector is installed. If a fiber selector is installed, specify the time sharing control as enabled.
This alarm may occur when consecutive standalone PRE_LS, LS, and LE instructions are executed. In that case, insert an about 3-second WAIT instruction between each instruction.

FFL-007 Shutter status mismatch

Cause: Even after “Shutter open/close latency” time on Time Sharing Control screen, “Shutter# open rq” signal and “Shutter #open st” signal didn’t correspond.

Remedy: Please check if Shutter open/close latency time is appropriate.
Fiber selector may be broken down. Please refer to the manual of laser oscillator, and confirm the status of fiber selector.

FFL-008 Received laser alarm

Cause: Robot received the laser alarm from laser oscillator.

Remedy: Please confirm the co-occurring FFL-025 alarm, and take measures with reference to the manual of laser oscillator. If all else fails, execute Diagnostic log function [FNCT menu / Diagnostic log] before you do other operation such as power off or jogging, and get the image backup. And contact your FANUC technical representative. If you cannot execute Diagnostic log function, document the events that led to the error and get the image backup. And contact your FANUC representative.

FFL-009 Laser control status mismatch

Cause: **"**CH# Laser Ctrl rq" signal was turned ON, but "CH# Laser Ctrl st" signal didn't become ON. Otherwise, "CH# Laser Ctrl rq" signal was turned OFF, but "CH# Laser Ctrl st" signal didn't become OFF.

Remedy: Please check if laser oscillator is ready to provide the control of laser.

FFL-010 Stand-by status is OFF

Cause: During the processing, "Laser ready st" signal is ON, but "Stand-by st" signal become OFF.

Remedy: Laser oscillator may be broken down. Please take measures with reference to the manual of laser oscillator. If all else fails, execute Diagnostic log function [FNCT menu / Diagnostic log] before you do other operation such as power off or jogging, and get the image backup. And contact your FANUC technical representative. If you cannot execute Diagnostic log function, document the events that led to the error and get the image backup. And contact your FANUC representative.

FFL-011 No Beam ON

Cause: **"**Beam ON st" signal did not turn on at the start of processing. Or **"**Beam ON st" signal turned OFF during processing.

Remedy: Please check if laser oscillator is ready to output beam.

FFL-012 Beam is NOT interlocked

Cause: **"**Interlock OFF rq" signal is OFF, but "Interlock OFF st" signal is ON.

Remedy: This is illegal case. Please check the laser oscillator. If all else fails, execute Diagnostic log function [FNCT menu / Diagnostic log] before you do other operation such as power off or jogging, and get the image backup. And contact your FANUC technical representative. If you cannot execute Diagnostic log function, document the events that led to the error and get the image backup. And contact your FANUC representative.

FFL-013 Guide light ON status mismatch

Cause: **"**Guide light ON rq" signal was turned ON, but even after constant time, "Guide light ON st" signal didn't become ON.

Remedy: Please check the laser oscillator.

FFL-014 Command Type status mismatch

Cause: **"**Command type rq" signal was turned ON, but even after constant time, "Command type st" signal didn't become ON.

Remedy: Please check the laser oscillator.

FFL-015 Stand-by status (%s)

Cause: **"**Stand-by st" signal was changed. (%s) shows the transition.

Remedy: This message is just for notification and information.

FFL-016 Laser ready status (%s)

Cause: **"**Laser ready st" signal was changed. (%s) shows the transition.

Remedy: This message is just for notification and information.

FFL-017 Beam interlocked

Cause: **"**Interlock OFF st" signal is changed from ON to OFF. It means that beam was interlocked.

Remedy: This message is just for notification and information.

FFL-018 Guide light ON status (%s)

Cause: **"**Guide light ON st" signal was changed. (%s) shows the transition.

Remedy: This message is just for notification and information.

FFL-019 Command type status (%s)

Cause: **“**Command type st” signal was changed. (%s) shows the transition.

Remedy: This message is just for notification and information.

FFL-020 Laser ready status mismatch

Cause: **“**Laser ready rq” signal was turned ON, but even after “Laser ready latency” time on Laser Setup screen, “Laser ready st” signal didn’t become ON.

Remedy: It takes a few second to become laser ready state from initial state. Please check if Laser ready latency is correct.
Please check if “Stand-by st” signal is ON, and if “Stand-by st” signal is OFF, check the chiller or peripherals.
There is a possibility that laser alarm is co-occurring. Please check if FFL-025 is occurring on active alarm log screen.

FFL-021 There is no (%s)

Cause: Slope control instruction was tried to be added to the motion instruction without LS, LE instruction.

Remedy: Slope control instructions can be added to the motion instructions with LS, LE instructions only.

FFL-022 Wire feed fault

Cause: Even after “Wire feed OK latency” time on Laser Setup screen, “Wire feed OK” signal didn’t become ON.

Remedy: Please confirm the wire feeder and wire shortage detector.

FFL-023 Present ITP_TIME is invalid

Cause: When this function is used, ITP time must be 8-16 msec. (Normally, ITP is less than 8 msec.)

Remedy: Please review the system so that the ITP will be 8 -16 msec. You can confirm the temporary ITP by $SCR.$ITP_TIME of System variables.

FFL-024 Connection to CNC fails

Cause: The connection with CNC integrated in laser oscillator was lost.

Remedy: Please confirm that laser oscillator power is turn on.
Please check the hard-wiring between the robot and laser oscillator.
Please reconfigure the transmission setting on Laser Initialization screen.

FFL-025 %s

Cause: Robot received the laser alarm message from laser oscillator.

Remedy: Laser alarm received from laser oscillator is displayed at %s. The laser alarm consists of laser alarm number and laser alarm message. Please confirm laser alarm number and laser alarm message, and take measures with reference to the manual of laser oscillator. If all else fails, execute Diagnostic log function [FNCT menu / Diagnostic log] before you do other operation such as power off or jogging, and get the image backup. And contact your FANUC technical representative. If you cannot execute Diagnostic log function, document the events that led to the error and get the image backup. And contact your FANUC representative.

FFL-026 Laser control status is OFF

Cause: Even though “CH# Laser Ctrl st” signal is OFF, “Laser ready rq” signal, “Beam ON rq” signal, “Interlock OFF rq” signal, “Guide light ON rq” signal or “Command type rq” signal became ON.

Remedy: When “CH# Laser Ctrl st” signal is OFF, status signals do not become ON by request signals’ ON.

FFL-028 Laser alarm is ON

Cause: Robot received the laser alarm from laser oscillator.

Remedy: Please confirm the co-occurring FFL-025 alarm, and take measures with reference to the manual of laser oscillator. If all else fails, execute Diagnostic log function [FNCT menu / Diagnostic log] before you do other operation such as power off or jogging, and get the image backup. And contact your FANUC technical representative. If you cannot

execute Diagnostic log function, document the events that led to the error and get the image backup. And contact your FANUC representative.

FFL-029 Out of range(%s)

Cause: The specified parameter is outside the range.
Remedy: Please consult to the string displayed at %s, and configure the suitable setting for the laser oscillator model on Laser Initialization screen.

FFL-030 Laser instruction error

Cause: The parameter of laser instruction is not correct.
Remedy: Please configure the suitable setting for the laser oscillator model on Laser Initialization screen.

FFL-031 Power control schedule error

Cause: The parameter of power control schedule is not correct.
Remedy: Please confirm the co-occurring FFL-025 alarm, and configure the suitable setting for the laser oscillator model on Laser Initialization screen.

FFL-032 Slope up schedule error

Cause: The parameter of slope up schedule is not correct.
Remedy: Please confirm the co-occurring FFL-025 alarm, and configure the suitable setting for the laser oscillator model on Laser Initialization screen.

FFL-033 Slope down schedule error

Cause: The parameter of slope down schedule is not correct.
Remedy: Please confirm the co-occurring FFL-025 alarm, and configure the suitable setting for the laser oscillator model on Laser Initialization screen.

FFL-034 Processing speed is too large

Cause: The processing speed in LS instruction is too large.
Remedy: Please correct the processing speed in LS instruction to appropriate value.

FFL-035 CNC connection error

Cause: The connection with CNC integrated in laser oscillator was lost.
Remedy: Please confirm that laser oscillator power is turn on.
Please check the hard-wiring between the robot and laser oscillator.
Please reconfigure the transmission setting on Laser Initialization screen.

FFL-037 Assist gas ready OFF during LS

Cause: Assist “Gas ready” signal is turned off during processing.
Remedy: Please check assist gas and gas valve unit (sensor).

FFL-038 Laser ready status ON time out

Cause: Even after “Laser ready latency” time on Laser Setup screen, “Laser ready st” signal didn’t become ON.
Remedy: Please check if laser alarm is occurring.
Please check if Laser ready latency time is appropriate. Please set appropriate value as system.

FFL-039 Output of ITP signal failed

Cause: Output of ITP signal is failed.
Remedy: Please confirm that laser oscillator power is turn on.
Please check the hard-wiring between the robot and laser oscillator.
Please reconfigure the transmission setting on Laser Initialization screen.

FFL-041 Teach pendant is enabled

Cause: Teach pendant is enabled.
Remedy: This message is just for notification and information.

FFL-042 Teach pendant is disabled

Cause: Teach pendant is disabled.
Remedy: This message is just for notification and information.

FFL-043 Controlled start is necessary

Cause: ITP_TIME was changed.
Remedy: Controlled start is necessary to enable this ITP_TIME.

FFL-044 Release beam interlock

Cause: “Interlock OFF st” signal is changed from OFF to ON. It means that the interlock of beam is released.
Remedy: This message is just for notification and information.

FFL-045 Stand-by ON time out

Cause: Even after “Stand-by ON latency” time on Laser Setup screen, “Stand-by st” signal didn’t become ON.
Remedy: Please check if laser alarm is occurring.
Please check if Stand-by ON latency time is appropriate. Please set appropriate value as system.

FFL-047 Beam is interlocked

Cause: Beam is interlocked.
Remedy: Beam is not output.
This message is just for notification and information.

FFL-048 Beam is not interlocked

Cause: Beam is not interlocked and “Guide light rq” signal became ON.
Remedy: To turn on guide light, beam should be interlocked.

FFL-049 No Beam ON for restart

Cause: Because “Process disable for restart” on Laser Setup screen is enabled, beam isn’t output on the restart during laser processing.
Remedy: This message is just for notification and information.

FFL-050 There is no %s

Cause: The function displayed to %s is not loaded.
Remedy: Please confirm your order list.

FFL-051 Minute output mode is OFF

Cause: The value of S is too low.
Remedy: Please execute MINUTE_OUTPUT_RQ, and enable minute output mode.

FFL-052 Minute output mode is ON

Cause: The value of S is too large.
Remedy: Please remove MINUTE_OUTPUT_RQ, and disable minute output mode.

FFL-053 Minute output status mismatch

Cause: State of minute output request signal and minute output status signal is the mismatch.
Remedy: Please check the laser oscillator.

FFL-054 Guide light off automatically

Cause: The guide light turned off automatically to save the life of guide light.
Remedy: Adjust the guide light auto OFF timer on laser setup screen

FFL-055 Interlock OFF status mismatch

Cause: **"**Interlock OFF rq" signal was turned ON, but even after constant time, "Interlock OFF st" signal didn't become ON.
Remedy: Please make sure **"**Stand-by st" signal is ON.
There is another line "Interlock OFF rq" signal (between peripheral equipment and laser oscillator). Another line "Interlock OFF rq" signal may remain OFF. Please check the laser oscillator.

FFL-056 Invalid laser control request

Cause: CH (channel) of this laser control request is disabled.
Remedy: Enable the "CH # Laser Ctrl rq" on the time sharing control setup screen.

FFL-057 CNC battery low

Cause: The battery level of the CNC built into the laser oscillator is low.
Remedy: If you turn off the laser oscillator while the CNC battery is empty, the settings will be lost. Replace the CNC battery.

FFL-058 Direct power override is ON

Cause: When the direct power override mode is enabled, the LS and LP instructions cannot be used.
Remedy: Specify 0 for the argument PO in the PRE_LS instruction.

FFL-059 Direct power override is OFF

Cause: When the direct power override mode is disabled, the LS_PO instruction cannot be used.
Remedy: Specify 1 for the argument PO in the PRE_LS instruction.

FFL-060 Argument %s is incorrect

Cause: The argument specified for this program is incorrect.
Remedy: Check the type and range of the argument.

FFL-061 Standalone PR_CALL cannot use

Cause: This instruction cannot be called from a standalone PR_CALL.
Remedy: Call this instruction from motion option PR_CALL.

FFL-062 Execute PRE_LS[CH=#] first

Cause: LS instruction was executed before PRE_LS[CH=#] instruction was executed.
Remedy: When "CH# Laser Ctrl rq switching" on Laser Initialization screen is enabled, PRE_LS[CH=#] instruction must be executed before LS instruction."

FFL-063 PRE_LS[CH=#] and CH# mismatch

Cause: CH# of “Laser Ctrl rq when TP is ON” on the time sharing control screen does not match the CH=# of PRE_LS.

Remedy: When executing the PRE_LS[CH=#] instruction while the teach pendant is enabled, the CH=# of PRE_LS instruction and the CH# of "Laser Ctrl rq when TP is ON" must be the same.

62.17 APPENDIX

62.17.1 Confirm Communication Setting for Laser Oscillator

This communication setting is completed at the factory shipment, so there is no need to change it.
Please confirm following items if communication is not established in spite of the fact that there are correct with communication setting for robot controller and connection of Ethernet cable.
For confirm the communication setting for laser oscillator by connecting *i*Pendant for maintenance of laser oscillator.

NOTE
Changing these setting may not connection to the robot.

Confirm Communication Setting for Laser Oscillator

(1) Confirm valid setting of Ethernet

A) Connect *i*Pendant to laser oscillator, and turn on the power of laser oscillator.

B) Press the function key [SYSTEM] after start completion to display the parameter screen.
C) Key in “970”, and press the soft key [NO. search].
D) Confirm the same as the settings in the table below.

Parameter	When 1 to 3 robots are connected	When 4 to 6 robots are connected
No.970	3	3
No.971	-1	-1
No.972	-1	-1
No.973	-1	-1
No.974	-1	-1
No.975	20	20
No.976	-1	24

(2) Confirm that necessary functions are available
After confirming (1), press the function key [SYSTEM] to display the diagnosis screen.
Key in “1272”, and press the soft key [NO. search].
Confirm diagnosis No.1272#0=1
If 4 to 6 robots are connected, key in “1181”, and press the soft key [NO. search].
Confirm diagnosis No.1181#3=1

Function	Drawing number	Diagnosis number
EtherNet/IP adaptor function	A02B-0327-R967	No.1272#0
EtherNet/IP adaptor PORT2function	A02B-0327-R974	No.1181#3

(3) Confirm setting of Ethernet connector (SLOT 1)
A) After confirming (2), press the function key [SYSTEM].
B) Press the soft key [+] several times, and press the soft key [ETHERNET].
C) Confirm the same as settings in the table below.

(4) Confirm setting of Ethernet connector (SLOT 2) (When 4 to 6 robots are connected)
After confirming (3), press the function key [SYSTEM].
Press the soft key [+] several times, and press the soft key [ETHERNET/IP].
Confirm the same as setting in the table below.

Setting of Ethernet

	Contents	Ethernet connector (SLOT 1)	Ethernet connector (SLOT 2)
Common	IP ADDRESS	192.168.18.101	192.168.18.102
	SUBNET MASK	255.255.255.0	255.255.255.0
	ROUTER IP ADDRESS	None	None
	DHCP CLIENT	0	0
	OPTION1	00000000	00000000
	OPTION2	00000000	00000000
FOCAS2	TCP port	8193	None

(5) Confirm setting of Ethernet connector (SLOT 1) : EtherNet/IP adaptor function
After confirming (4), press the function key [SYSTEM].
Press the soft key [+] several times, and press the soft key [ETHERNET].
Press the soft key [+] several times, and press the soft key [EIP A SET].
Press the cursor keys, and confirm the settings of ALLOCATION 1 to 3 for Fast Ethernet board 1(SLOT1) (*).

(6) Confirm setting of Ethernet connector (SLOT 2) : EtherNet/IP adaptor function (When 4 to 6 robots are connected)
After confirming (5), Press the cursor keys, and confirm the settings of ALLOCATION 1 to 3 for Fast Ethernet board 2(SLOT2)(*).

(*) "Fast Ethernet board 1 (SLOT 1)","Fast Ethernet board 2 (SLOT 2)" is displayed in screen title.

Setting of EtherNet/IP Adapter function

		Ethernet connector (SLOT 1) (Fast Ethernet board 1(SLOT1))				**Ethernet connector (SLOT 2) (Fast Ethernet board 2(SLOT2))**			
		ALLOCATION 1 (ROBOT1)	ALLOCATION 2 (ROBOT2)	ALLOCATION 3 (ROBOT3)	ALLOCATION 4	ALLOCATION 1 (ROBOT4)	ALLOCATION 2 (ROBOT5)	ALLOCATION 3 (ROBOT6)	ALLOCATION 4
STATUS ADDRESS		R0246				R0250			
STATUS SIZE		3				3			
DI DATA ON ABNORMAL		CLEAR				CLEAR			
DI	TYPE	1	1	1	-	1	1	1	-
	ADDRESS	R0256	R0768	R1280	-	R1792	R2304	R2816	-
	SIZE	152	152	152	-	152	152	152	-
DO	TYPE	1	1	1	-	1	1	1	-
	ADDRESS	R0512	R1024	R1536	-	R2048	R2560	R3072	-
	SIZE	18	18	18	-	18	18	18	-
OPTION		00000101	00000101	00000101	-	00000101	00000101	00000101	-

ALLOCATION 4 is not used.

63 VIA INSTRUCTION

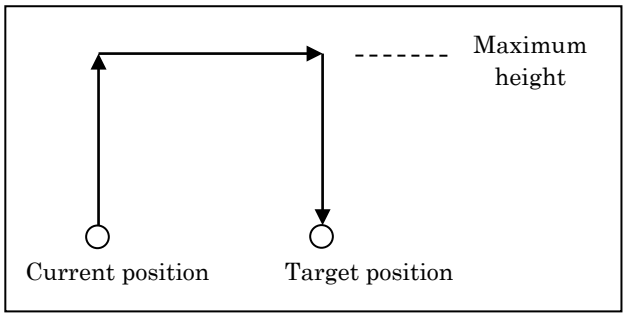
VIA instruction is one of additional motion instructions. This instruction makes it easy for user to teach the motion that goes up to the intended height, go horizontally, and go down to the target position without directly going to the target position.
(This instruction is for SCARA robot. It has been supported since software version 7DF1/20.)

63.1 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.

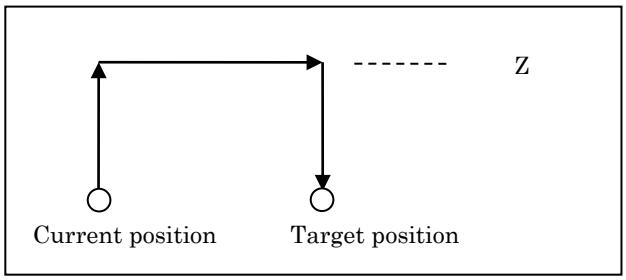
63.1.1 DEFAULT

VIA(DEFAULT)
Robot goes up to the maximum height. This is FINE motion.



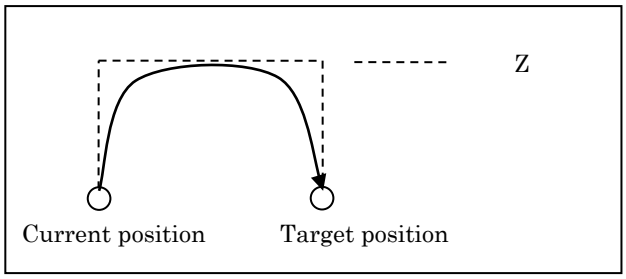
63.1.2 FINE

VIA(FINE,Z=n)
Robot goes up to the height that is specified by Z value. This is FINE motion.

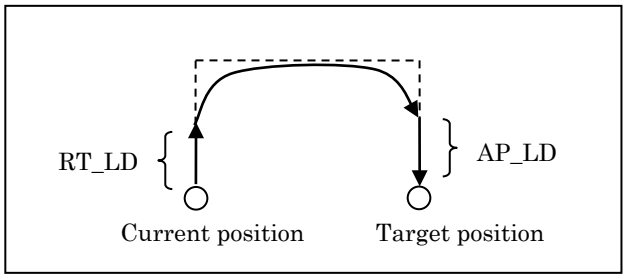


63.1.3 CNT

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.



63.2 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.)

64 CRW SHAPE GENERATION FUNCTION

64.1 OVERVIEW

This function makes it easy to create machining programs aimed at welding 2-dimensional shapes (hereinafter called welding shape program) using a Compact Remote Welder (hereinafter called CRW).
The guide screen of the CRW Shape Generation Function is added to ***i***HMI, and it enables creating a welding shape program via a step-by-step sequence. (For details about the function and the operation of ***i***HMI, please refer to Section 64.2 “WELDING SHAPE PROGRAM CREATION BY ***i***HMI” and the ***i***HMI chapter of operator’s manual (Basic Operation) (B-83284EN).

64.1.1 Necessary Software Option

The following software options are necessary to use this function.

Table 64.1.1 (a) Necessary software options

Name	Option number
CRW Shape Generation Function (This Function)	A05B-2670-S514
LaserMech FiberSCAN CRW	A05B-2600-H872
FANUC FIBER LASER Interface Function	A05B-2600-J922
EtherNet/IP Scanner Function	A05B-2600-R785
Continuous Turn Function	A05B-2600-J613

64.1.2 Supported Shapes

The CRW Shape Generation function generates a welding shape program for point welding (the robot stops during welding) and path welding (the robot moves during welding).

The following six shapes are supported for a **point** welding shape program.

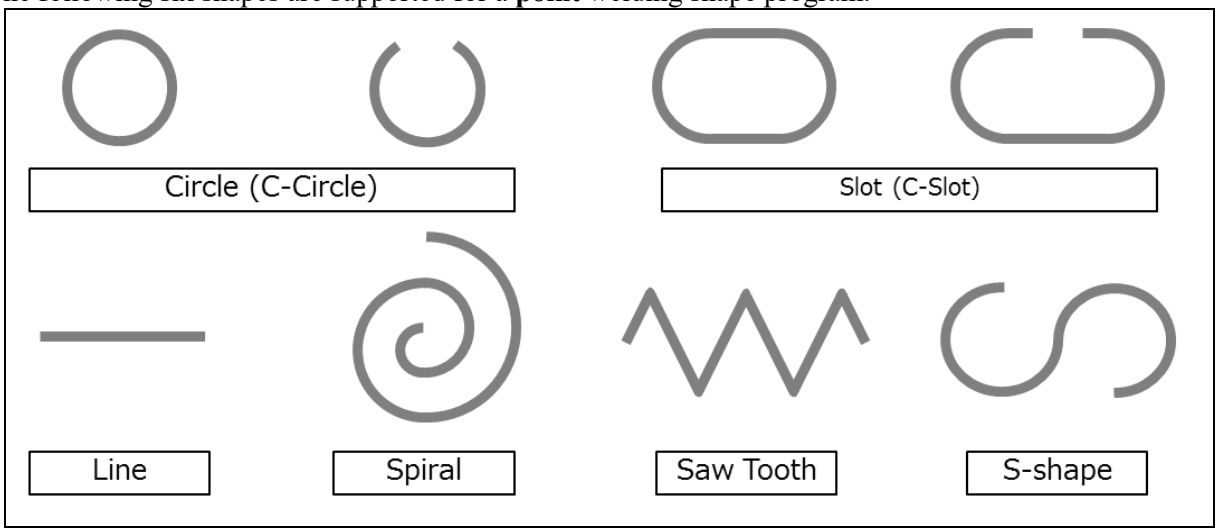


Fig. 64.1.2(a) Point Welding Shapes

The following three shapes are supported for a **path** welding shape program.

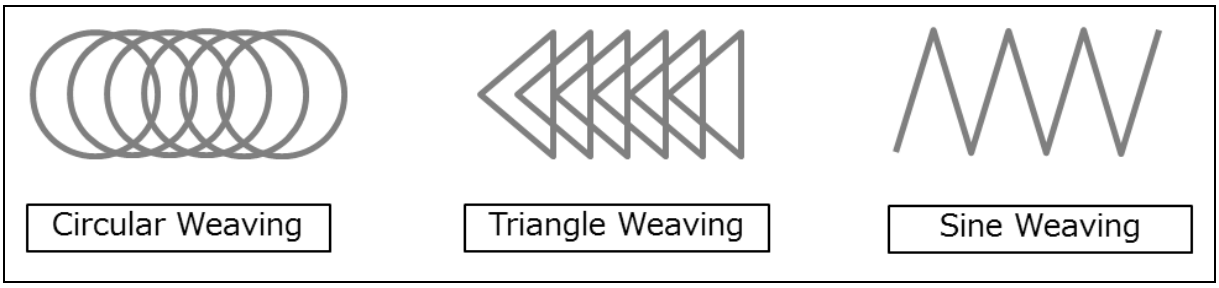


Fig. 64.1.2(b) Path Welding Shapes

64.1.3 Common Settings

The following items can be configured as common settings for all welding shape programs.

- Insert Laser instruction
- Laser instruction setup
- Write Protection
- Advanced Types (Program Subtype)
- Select unit

64.2 WELDING SHAPE PROGRAM CREATION BY *i*HMI

64.2.1 CRW Shape Generation Screen

Tap the [BASIC] icon on the HOME screen and display the Basic Setup Selection screen.
On the Basic Setup Selection screen, select Compact Laser Shape Generation screen.
The screen display flow is shown below.

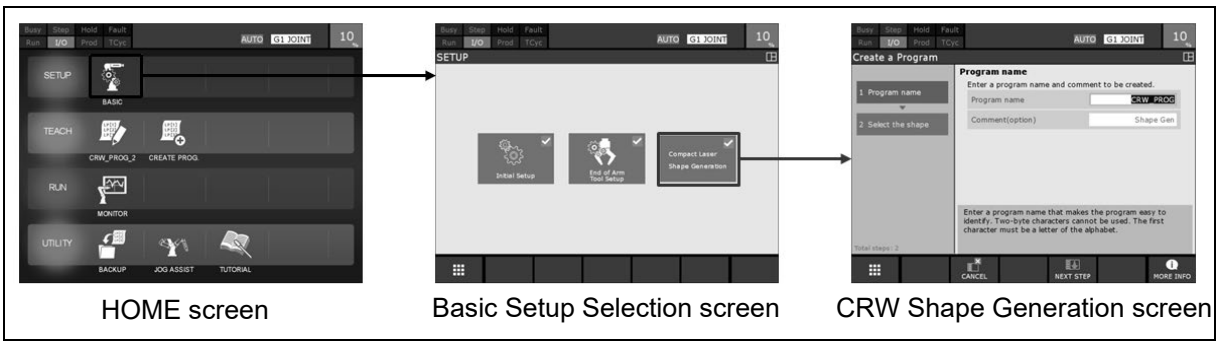


Fig. 64.2.1 Screen display flow

64.2.2 Creation Of Welding Shape Program

On the CRW Shape Generation screen, you can create a welding shape program via a step-by-step sequence.
Please follow the on-screen guidance and select items or input values at each step, and then tap the [NEXT STEP] in the function key menu display area. When you proceed to the “Create TP program” step, you can generate a welding shape program, and the program is added to program selection screen.

The flow of the welding shape program creation is shown below.

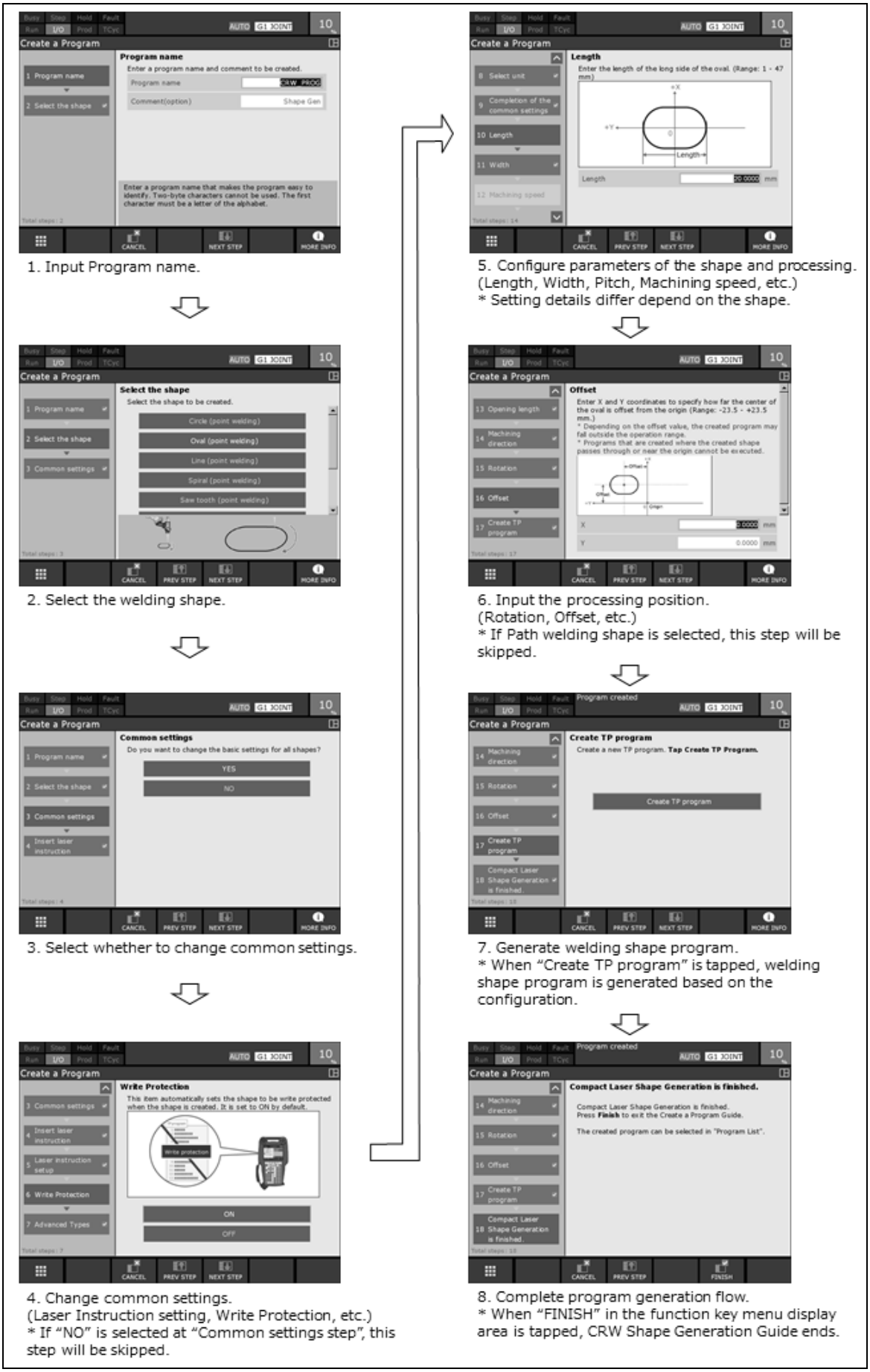


Fig. 64.2.2 Welding shape program creation flow

64.3 EXECUTION OF WELDING SHAPE PROGRAM

Welding shape programs generated by CRW Shape Generation function have a motion group of CRW only. So, welding shape programs need to be executed by a production program (main program) with a CALL instruction or a RUN instruction, depending on the application.

> ⚠ **WARNING**
> Ensure that the calling sequence of the welding shape program for proper processing is performed correctly.
> When an error occurs, check to stop the welding process. The CRW Shape Generation function does not automatically stop the machining process. Design the system such that the machining process is properly stopped when an error occurs. Otherwise, machining processes that are not stopped properly can harm workers or peripheral devices.

64.3.1 Execution Of A Point Welding Shape Program

When a **point** welding shape is selected and a welding shape program is generated, laser welding is performed without robot motion.
Therefore, the welding shape program must be executed by the main program with a CALL instruction as shown below.

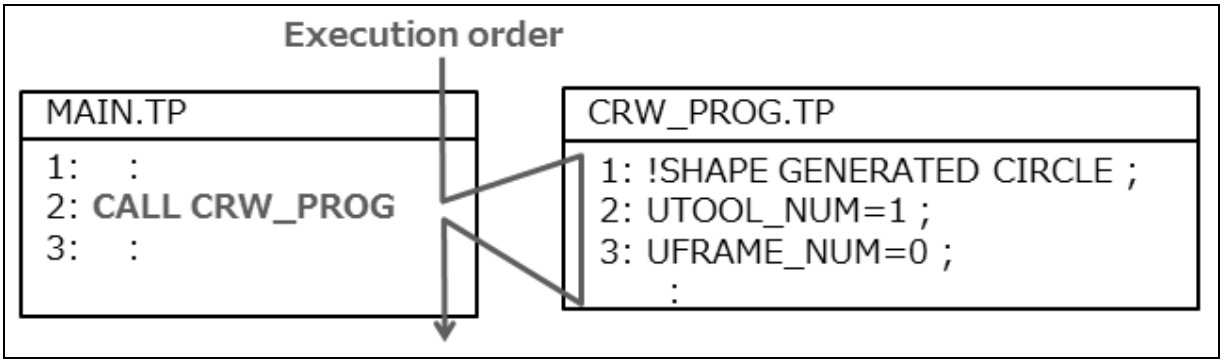


Fig. 64.3.1 Execution of point welding shape program

64.3.2 Execution Of A Path Welding Shape Program

When a **path** welding shape is selected and a welding shape program is generated, laser welding is performed with robot motion.
Therefore, a welding shape program must be executed by the main program with a RUN instruction as shown below.

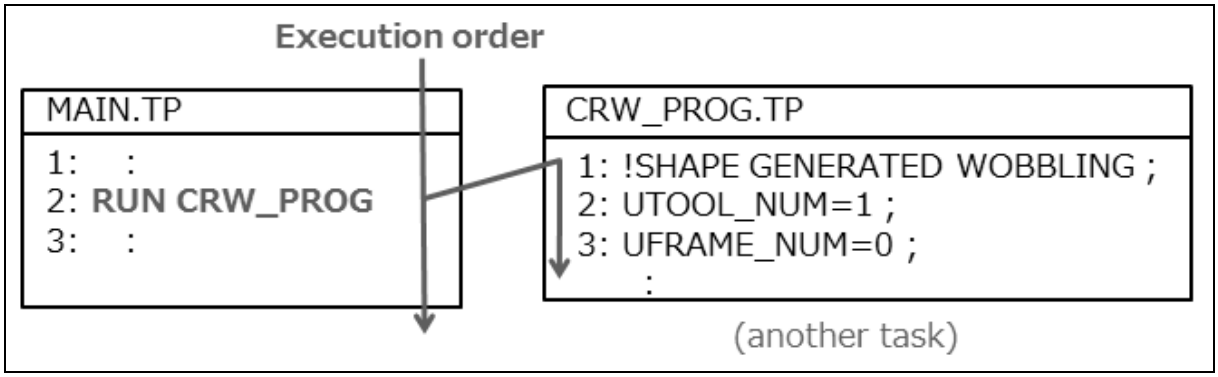


Fig. 64.3.2 Execution of path welding shape program

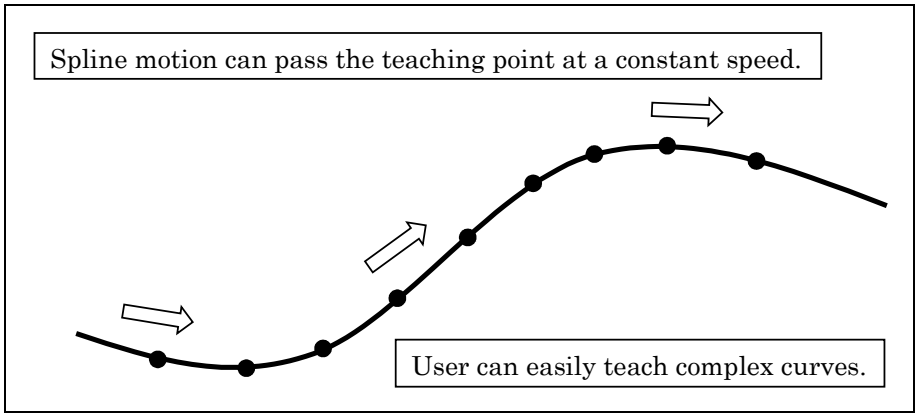
NOTE

1 When a path welding shape is selected, the generated welding shape program continues welding until R[1] becomes 1. Therefore, please modify the welding shape program and change the end condition of welding appropriately.
(It is assumed that the main program executes and stops the execution of the path welding shape program. Therefore, please put the instruction that matches the end condition in the main program.)

2 When you run the welding shape program with RUN instruction, the timing of program start may not match the robot motion. In such cases you can insert WAIT instruction (e.g. WAIT 0.2(sec)) right after the RUN instruction to synchronize them.

65 SPLINE MOTION INSTRUCTION

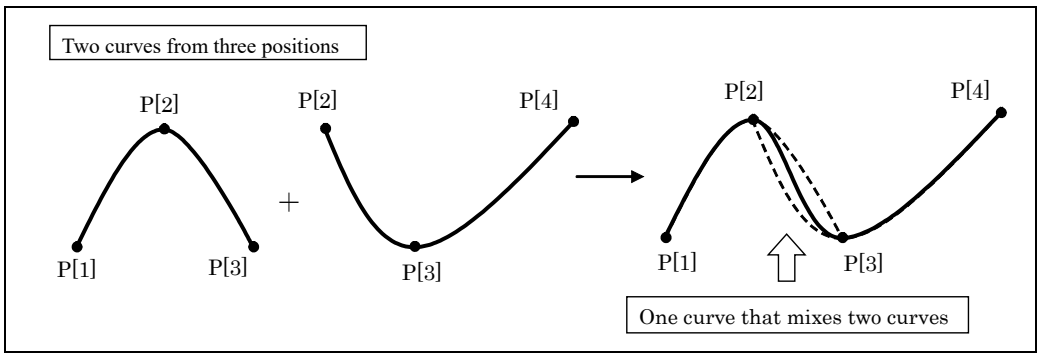
Spline (type S) motion moves on a curve that passes through many positions. One row has one position. The controller calculates a curve through all positions from a continuous spline motion line. If the teaching speed is slow, it moves at a constant speed.
To use this function, Spline motion (R904) is required.



NOTE
If the distance between the teaching points is very short, or if the robot draws a path that curves sharply, the robot will automatically decelerate because it cannot pass the path at the teaching speed. MOTN-632 SPL:Speed Reduced (G: , L:) will be posted if the robot cannot pass at teaching speed. Also, if the robot slows down but cannot pass the path, MOTN-633 SPL:VAJ Limit (G: , L:) will be displayed.

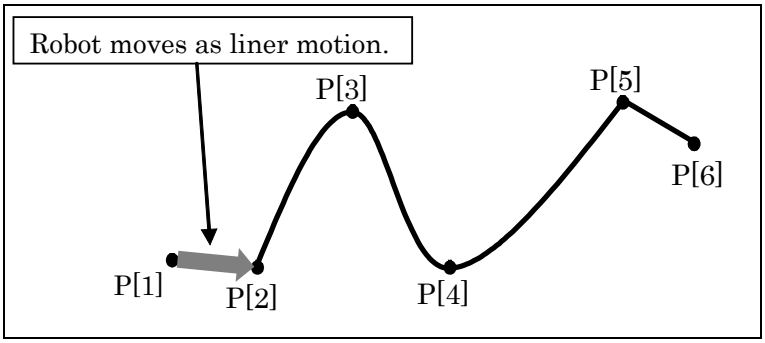
65.1 NORMAL MOTION

If the spline motion is 3 lines, draw a curve that smoothly passes through the 3 positions. If the spline motion is more than 4 lines, draw a curve that mixes the two curves between the two positions.



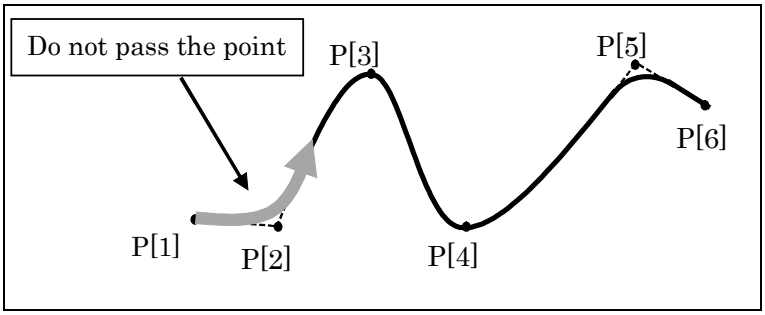
The starting point of the curve is taught by spline motion. The first Spline motion line moves in the same way as L motion. If L motion to the starting position is not required, set the first line of the spline motion to the same position as the previous motion line.

First spline motion instruction



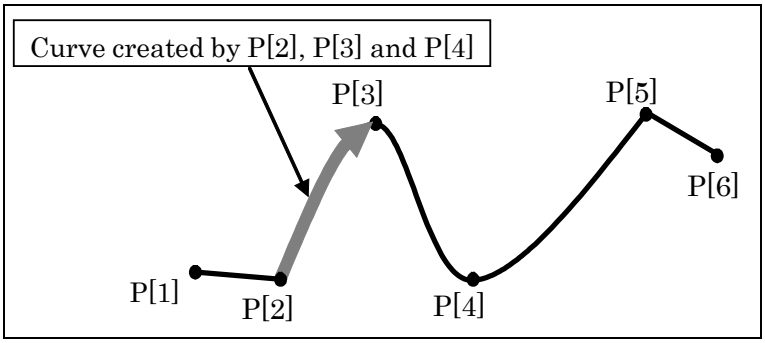
If you set CNT in the first S motion, the connection will be as smooth as L motion. At this time, near the start point, L motion and Spline motion are blended by CNT, so they do not pass through the teaching point.

CNT from line motion to spline motion



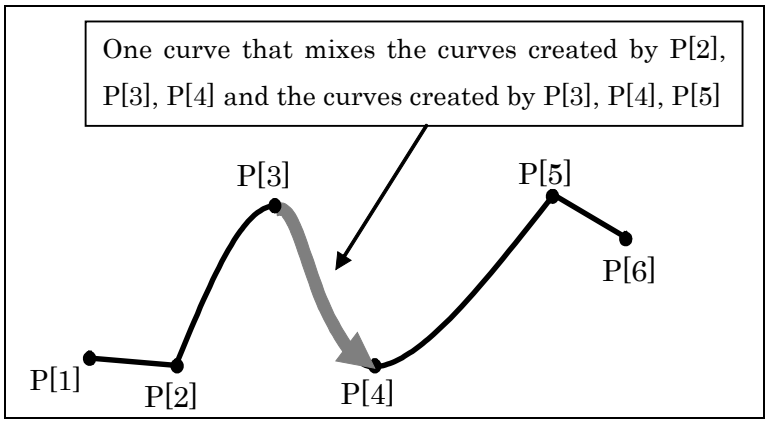
In the second spline motion instruction, it moves on a curve created by three points. Current position, destination position of this spline motion instruction and that of the next spline motion instruction.

Second spline motion instruction



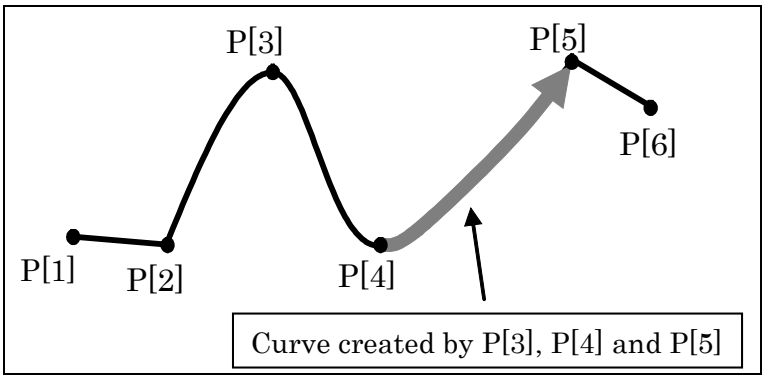
In the third and subsequent spline motion instructions, it moves on the curve calculated by the four positions.

Third and subsequent spline motion instructions



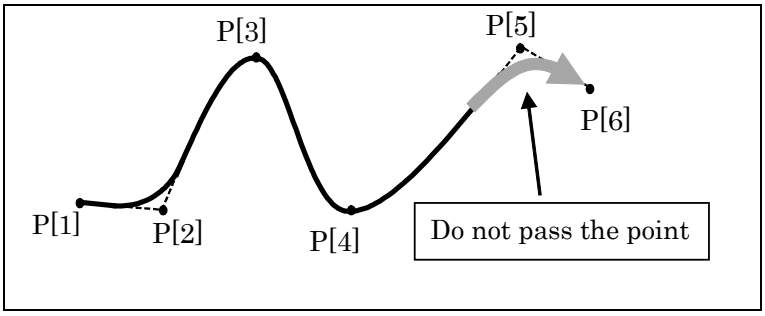
When the next motion instruction is not spline motion instruction, the last spline motion instruction is regarded as the end position of the curve. In this case, robot moves on the curve that is calculated by 3 positions, start position of the previous motion instruction, current position, and destination position of this motion instruction.

Last spline motion



If CNT is selected for the last spline motion, it will connect smoothly to other motion types. At this time, the teaching point does not pass through the end point of the spline motion.

CNT from spline motion to other types motion

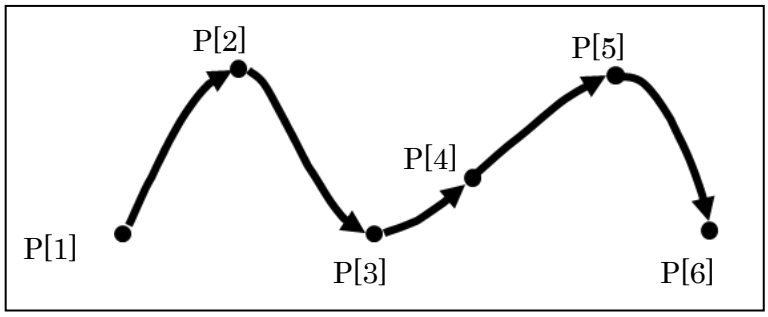


65.2 POSITIONING PATH OF THE SPLINE MOTION

In contrast to L motion and J motion, S motion passes through points even if set to CNT 1-100. If the CNT100 is set by connecting two S-motions with the same speed, it will move at a constant speed. If you use FINE or CNT0-99, it will slow down near the point. In particular, when FINE or CNT0 is used, it pauses on a point. At this time, the path does not change. When passing the teaching point, it passes at the attitude close to the teaching posture. If CNT1-100 is set for connection with other motion types, the connection will be as smooth as the connection between other motion instructions.

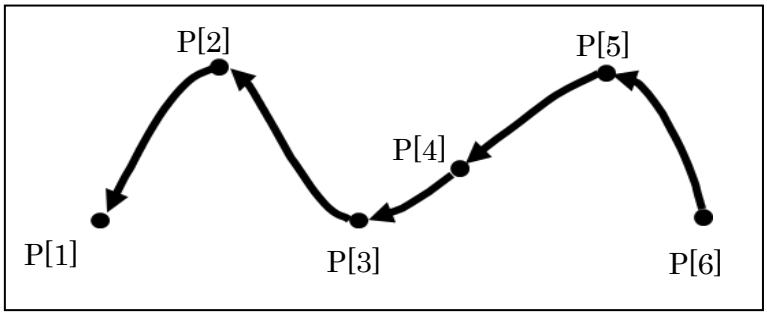
65.3 SINGLE STEP EXECUTION

If the spline instruction is executed by single step, program is paused on every taught point like other motion instruction.



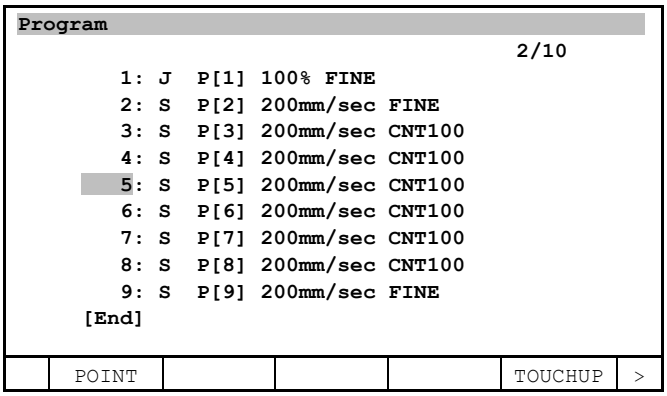
65.4 BACKWARD EXECUTION

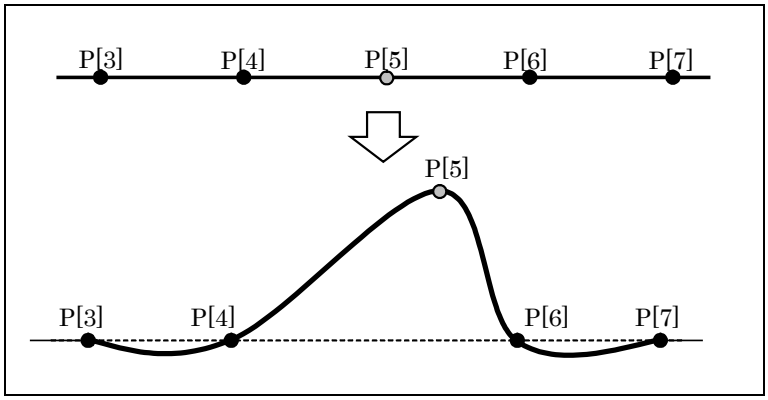
If the spline motion instruction is executed by backward, robot moves as backward on the same path as forward execution, and the program is paused on every taught point.



65.5 THE EFFECT OF ONE TEACHING 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. It should be noted that one point affects many segments compared to other motion types.



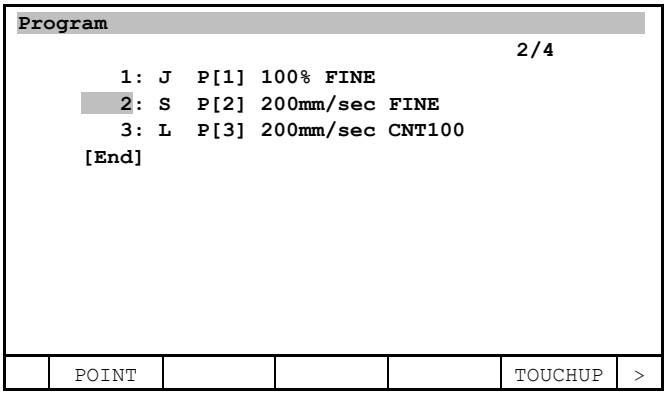


65.6 CASES SPLINE CANNOT BE PLANNED

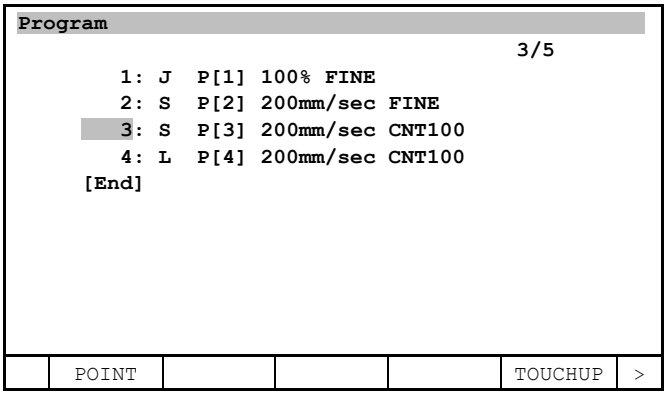
65.6.1 Shortage of the Spline Motion Instruction

You must teach successive three or more spline motion instructions.
If you don't, "INTP-737 SPLIN:S needs 3 points" will be displayed.

Ex.1) The spline motion instruction is isolated. "INTP-737 SPLIN:S needs 3 points (1)" will be displayed.

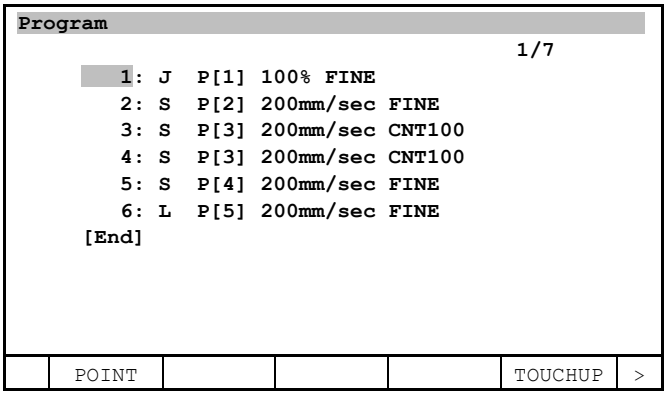


Ex.2) There are only two spline motion instructions. “INTP-737 SPLIN:S needs 3 points (2)” will be displayed.



65.6.2 Teaching of Same Position

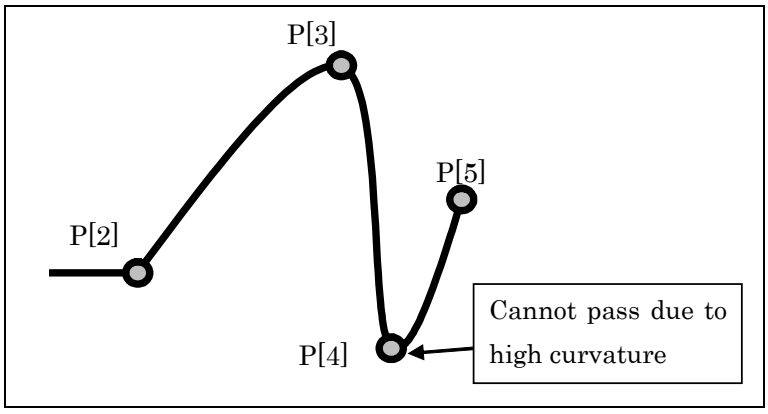
If two consecutive points are at the same position, “MOTN-635 SPL: Zero move” will be displayed.



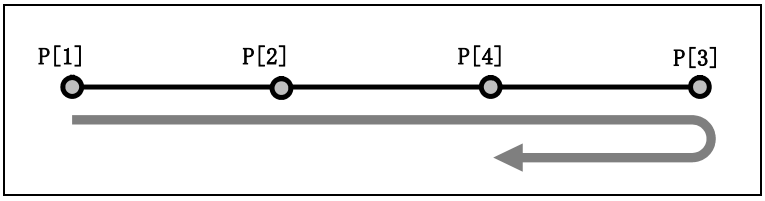
65.6.3 Case the Path Has High Curvature

If the robot cannot move on the curve at the teaching speed, it will automatically decelerate. In addition, if the automatic deceleration is not enough, an alarm will be generated. You need to change the position of the point, reduce the teaching speed, or change the positioning path.

Ex.1) If automatic deceleration is not in time, “MOTN-633 SPL:VAJ Limit” will be displayed.

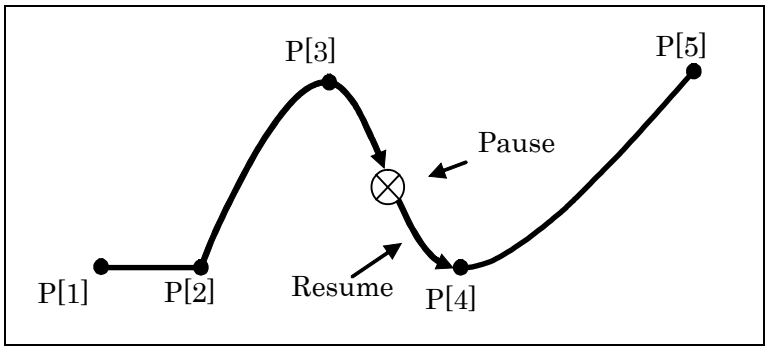


Ex.2) When moving in the reverse direction, “MOTN-637 SPL:Invalid Set of Points” will be displayed.



65.7 RESUME AFTER PAUSE

In case robot motion is paused and resumed without change, resumed path is just as usual.

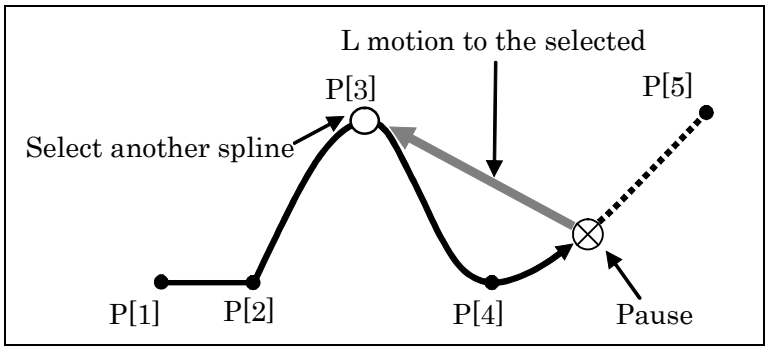


65.7.1 Resume After JOG

Suppose that a program is paused during spline motion. Then robot is jogged and the program is resumed. In this case, if original path resume is enabled, robot returns to the paused position, and continue the motion. If original path resume is disabled, robot does not resume. After moving to a safe position by jog and executing on a line different from the paused line, the robot can return to the original path after moving to that point with L motion.

65.7.2 Pause and Resume from Another Spline Motion Instruction

The robot moves to the target position on the cursor line in L motion. When the robot reaches that point, it draws a curve from that point. When single step or backward execution is used, only the L motion to that position is executed.



65.7.3 Resume 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 displayed. This alarm will not be displayed when restarting from another motion instruction.

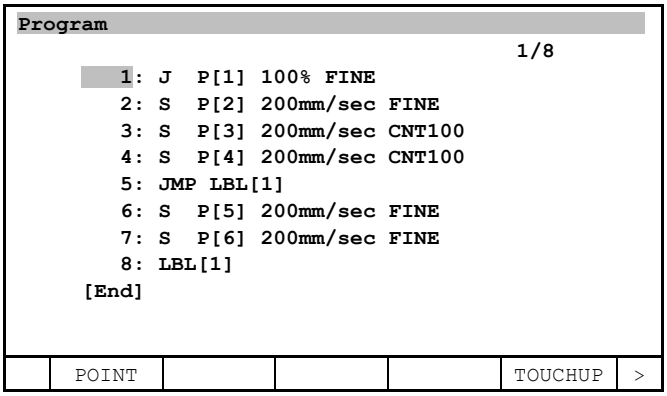
65.7.4 Start Program from Spline Motion Instruction

This case is same as "Pause and resume from another Spline Motion instruction"

65.8 LOGIC INSTRUCTIONS BETWEEN SPLINE MOTION

Logic instructions can be taught between spline motion instructions.
Spline motion instruction uses positions of successive Spline motion instructions to calculate curve path. Logic instructions that change the order of execution such as JMP, LBL cannot be executed between spline motions. Logic instructions that effect on position data such as position register instruction neither.

For example, execution of the following programs causes error at line 4.



Available logic instructions 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 error. When used between S motion instructions connected with CNT1 to CNT100, stop near the teaching point on the path. If you want to wait on the correct teaching point, set the positioning path to FINE or CNT0.
Call	CALL (Prg)	If called program has motion instruction or unavailable logic instruction between spline motion instructions, error occurs.
Macro	HAND1 OPEN	If macro program has motion instruction or unavailable logic instruction between spline motion instructions, error occurs.
If / Call	IF DI[1]=ON CALL (Prg)	IF / JMP / LBL causes error.
Select / Call	SELECT R[1]=1 CALL (Prg1) =1 CALL (Prg2) OTHERS CALL(Prg3)	SELECT / JMP / LBL causes error.

Instructions	Examples	Notes
RSR	RSR[1]	
User alarm	UARM[1]	
Override	OVERRIDE=50%	
Timer	TIMER[1]=START	
Parameter	$(Parameter)=100	
Remark Multi language remark	! (Remark) -- (Remark)	
TP line remark	// J P[1] 100% FINE	
Message	Message[message statement]	
Positional register lock / unlock	LOCK PREG UNLOCK PREG	
Maximum speed	JOINT_MAX_SPEED[1]=50	
Pause	PAUSE	
Program execution instruction	RUN (program name)	
Torque limit	TORQ_LIMIT 50.0%	
Monitor instruction	MONITER (Prg) MONITER END (Prg)	Contents of the program specified by monitor instruction are not checked.
Condition	CONDITION R[1]=1 CALL(Prg)	Condition instruction isn't checked because it is taught only in program specified by monitor instruction.
InterBus-S	IBS ATTACH	
InterBus PCI	IB attach	
Path switch	PS -10mm,1.0sec DO[1]=ON	
Collision detect	COL DETECT ON	
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 following instructions between spline motion instructions causes error. But they are available in programs called between the circle arc motion instructions.

Instructions	Examples	Notes
End	END	
Jump	JMP LBL[1]	
Label	LBL[1]	

Instructions unavailable between spline motion instructions

Instructions not given above examples are unavailable between spline motion instructions. Major unavailable instructions are as follows.

Instructions	Examples	Notes
Position register	PREG[1]=LPOS PREG[1,1]=100	
Abort	ABORT	
Offset	OFFSET_CONDITION PR[1]	
Tool offset	TOOL_OFFSET_CONDITION PR[1]	
Vision offset	VOFFSET VR[1]	
User frame User frame number Tool frame Tool frame number	UFRAME[1]=PR[1] UFRAME_NUM=1 UTOOL[1]=PR[1] UTOOL_NUM=1	
Spot tool	SPOT[1]	
Palletizing	PALLETIZING-B_1	
Simultaneous group instruction Independent group instruction	Simultaneous GP Independent GP	

65.9 AVAILABLE MOTION OPTIONS

Some motion options cannot be used with spline motion instruction. Unavailable motion options aren't displayed in motion option sub-menu.

Available Motion Options

Instructions	Example	Comment
Wrist joint	WJNT	
Acceleration override	ACC 80	
Path	PTH	
Offset	Offset PR[1]	
Tool Offset	Tool_Offset PR[1]	
Remote TCP	RTCP	
Time Before Time After Distance Before	TB 2.00sec CALL PRG TA 0.30sec CALL PRG DB10.0mm CALL PRG	In case there is a logic instruction forbidden between spline motion instructions in called program, error is posted when executed.
Arc Welding	Arc Start[1]	
Dispense start/end	SS[1]/SE	
Coordinated motion	COORD	
Process speed	PSPD 50	
Vision offset	VOFFSET	

Major unavailable Motion Options

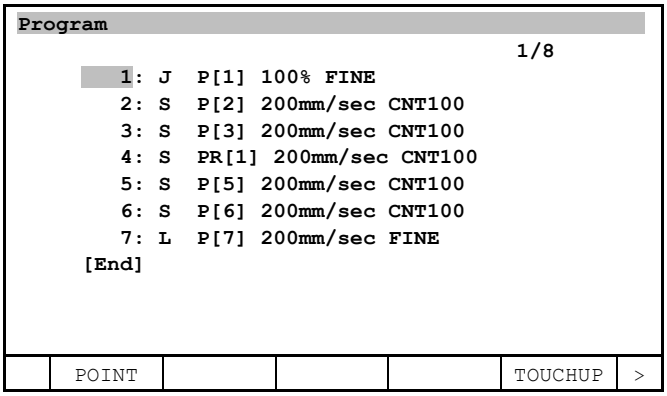
Motion Options not in above table cannot be added to the circle arc motion instruction. Major unavailable Motion Options are as below.

Instructions	Example	Comment
Incremental	INC	
Skip	Skip LBL[1]	
High Speed Skip	Skip PR[1] LBL[1]	
Spot Welding	SPOT[1]	
Weave	Weave Sine[1]	
Touch Sensing	Search [X]	
Corner Region	CR50	
Continuous Rotation	CTV100	
Minimal rotation	WJNT MROT	

65.10 CHANGE POSITION DATA DURING S MOTION EXECUTION

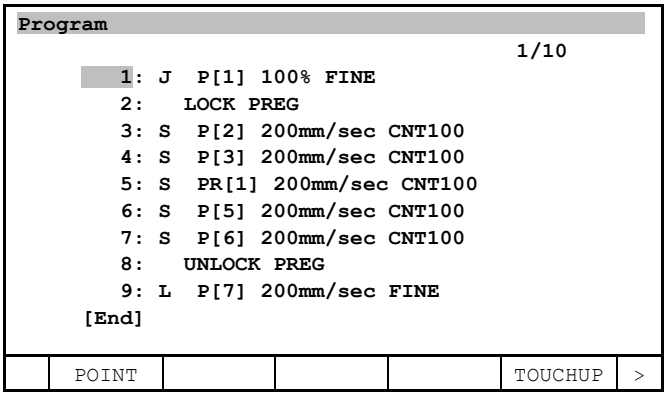
In program with spline motion instructions, one position data is accessed by multiple motion instructions. So in case the position data is changed while the data is being used for calculation of curve path it causes contradiction between calculated paths. Robot can move along unexpected path. To avoid this, the circle arc motion instruction can detect inconsistent position during execution. If detected, alarm is posted.

Ex.1) If the value of the PR[1] is changed in position register screen during execution of line 5, “INTP-755 SPLIN: Inconsistent position” is posted when line 6 is executed.



To avoid stopping production by changing position register, you can use “LOCK PREG”/“UNLOCK PREG” instructions. Teach “LOCK PREG”/“UNLOCK PREG” before/after spline motion instructions that access position register.

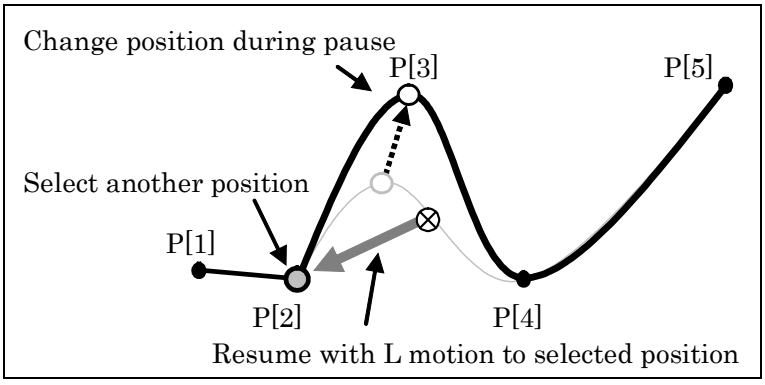
Ex.2) This makes it is impossible to change the value of position register by position resister screen when instructions between “LOCK PREG” and “UNLOCK PREG” are executed.



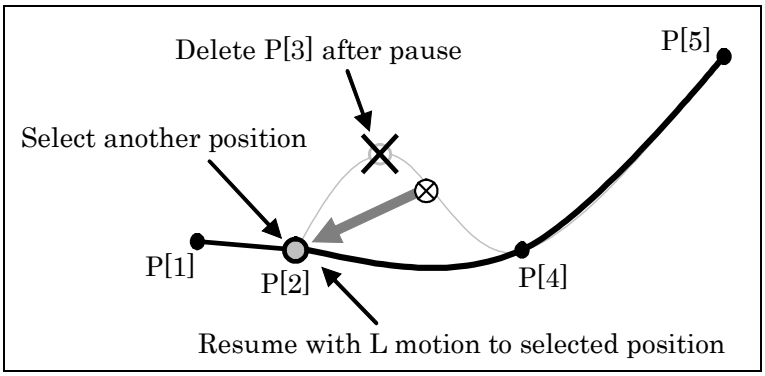
65.11 RESTART AFTER CHANGING PROGRAM

If the path of the current instruction changes while paused in spline motion, the current position is not on the path. At this time, the robot cannot be restarted and an alarm will be displayed. After performing JOG, check the safety and resume from another Spline motion instruction to resume. At this time, it moves to the selected point with L motion.

Ex.1) After pausing in S motion instruction, if you change the position of a point used in the current path, “INTP-731 SPLIN:path change” will be displayed.



Ex.2) After pausing in S motion instruction, if you delete the position of a point used in the current path, "INTP-731 SPLIN:path change" will be displayed.



65.12 RESTRICTIONS

- Backward execution from an interrupted program is not supported
- In case PAL_*[ ] is used as the position of a type S motion instruction, an error is posted when it is executed.
- In case multiple robots are set in the motion group of the TP program, an error is posted when it is executed.
- In case the position register is used as the position type S motion instruction and its index is specified indirectly by register, an error is posted when it is executed.
- In case Offset or Tool_Offset is attached to a type S motion instruction and an index of PR is indirectly specified, an error is posted when it is executed.
- Singularity Avoidance is not supported.
- LVC function is not supported.
- Line tracking is not supported.

66 LEARNING PATH CONTROL

66.1 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 (A05B-2600-R816).
- When LPC is used to improve small shapes performance, Small Circle Accuracy Improvement (A05B-2600-J644) is recommended to be installed first.
- LPC recognizes the TP program generated by ShapeGen software. LPC's setup will be completed automatically if learning targeted program is ShapeGen created TP program.

NOTE
Model specific, contact your local FANUC representative for robot support list

66.1.1 Procedure

To set up this function, please follow this sequence:

1. Understand necessary hardware, necessary software, and function limitations. (Detailed information in Ch.66.2)
2. Set up hardware and software. (Detailed information in Ch.66.3)
3. Perform learning procedure. (Detailed information in Ch.66.4)
4. Backup LPC schedule. (Detailed information in Ch. 66.5)

66.2 HARDWARE AND SOFTWARE

This chapter explains necessary hardware, necessary software, and software limitations.

66.2.1 Hardware

LPC requires the following hardware:

- 64MB D-RAM is necessary. LPC requires around 3MB available memory in D-RAM.

66.2.2 Software

Required Software Options

- Learning Path Control (LPC) (A05B-2500-R816) can be used in the R-30iB controller with version v8.20/P12 or later.
- Using LPC to improve small shape path performance, Small Circle Accuracy (J644) is recommended to be installed and ShapeGen software utility is recommended to be used to generate small shapes TP program.

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.

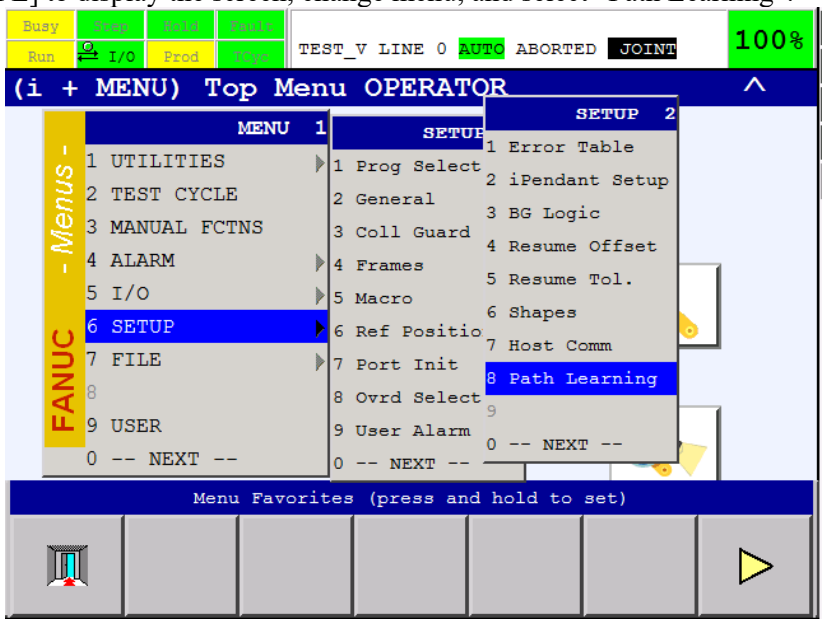
- LPC is currently designed and validated for circles only. Other geometric types will be supported in the future release.

66.3 LPC SETUP MENU

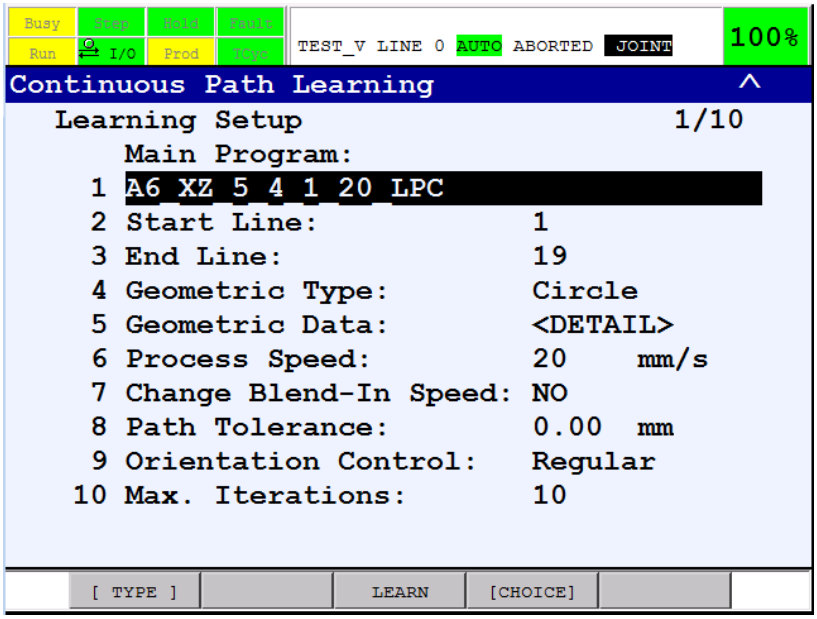
This chapter explains how to set up LPC for learning. As the first step, LPC must be configured in the setup menu. The setup menu can be launched from TP editor's related view for easy access and better display. The following procedure is used to set up LPC.

Procedure 66-3-1 Displaying the LPC setup menu

1 Press the [MENU] 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.



66.3.1 “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.

66.3.2 “Start Line” ITEM

This menu item specifies the start TP line for learning process. LRN motion modifier will be added starting from this line.

66.3.3 “End Line” ITEM

This menu item specifies the end TP line for learning process. LRN motion modifier will be added before from this line.

66.3.4 “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.

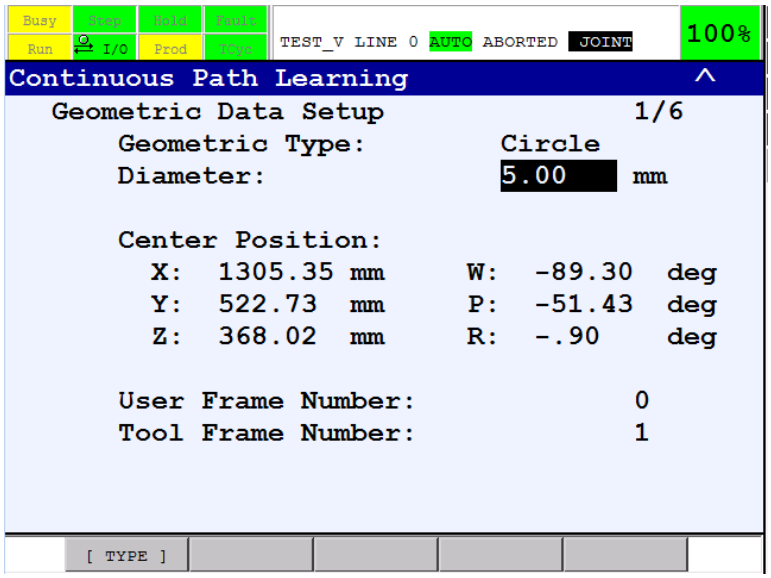
66.3.5 “Geometric Data” ITEM

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



66.3.6 “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.

66.3.7 “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.

66.3.8 "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.

66.3.9 "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 set $PTHLRN_CFG.$FLAG to 2.

66.3.10 "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.

66.3.11 "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.

66.3.12 "Auto Add TP Points" ITEM

This menu item is effective only when circular motion are learned. When this function is effective, teach point is added automatically. This may improve path accuracy.

66.3.13 "No. CIR Lines" ITEM

This menu item specifies the number of teach points to be added. It can be selected only when "Auto Add TP Points" item is valid.

66.3.14 "LEARN" FUNCTION KEY

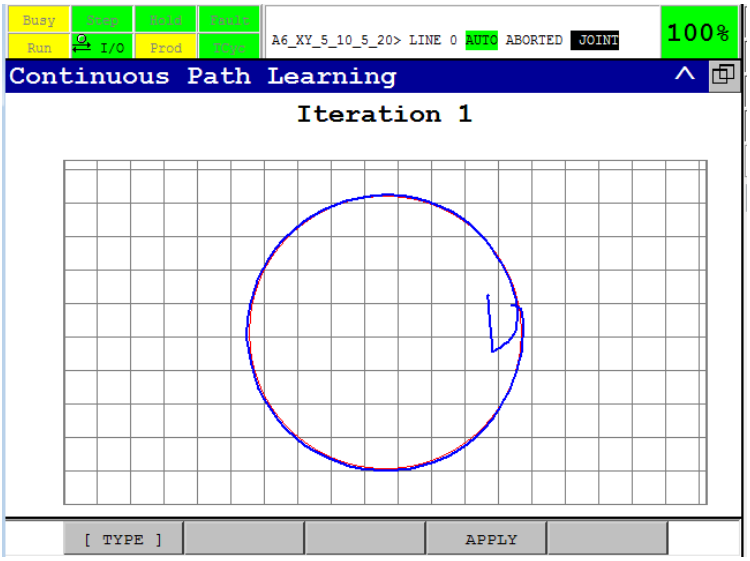
The "LEARN" function key begins the learning sequence.

- 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.

66.3.15 “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.



66.3.16 “APPLY” FUNCTION KEY

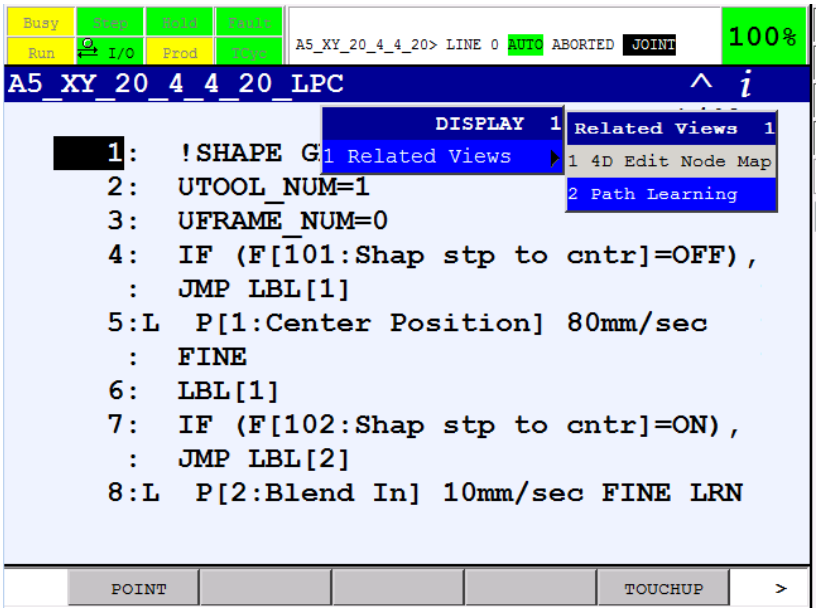
In the display window, “APPLY” function key will be available to use the displayed TCP trace as a final result and overwrite current learned data.

66.3.17 RELATED VIEW

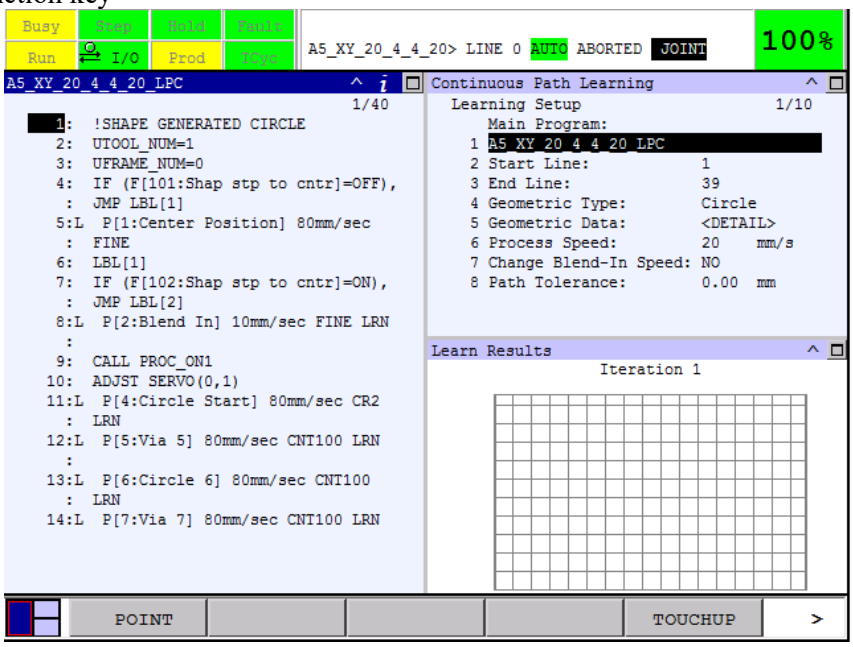
The setup menu can be launched easily when targeted TP program is opened in TP editor using related view. The TCP trace will display in the lower right pane real time during learning.

Procedure 66-3-2 Displaying the LPC related view menu

1 Open targeted TP program in editor.
2 Touch *i* key to bring up related view menu or press *i* + Fctn key.
3 Select “Path Learning” in related view menu.



4 After Path Learning related view windows opened, move the focus to upper right pane for setup and LEARN function key



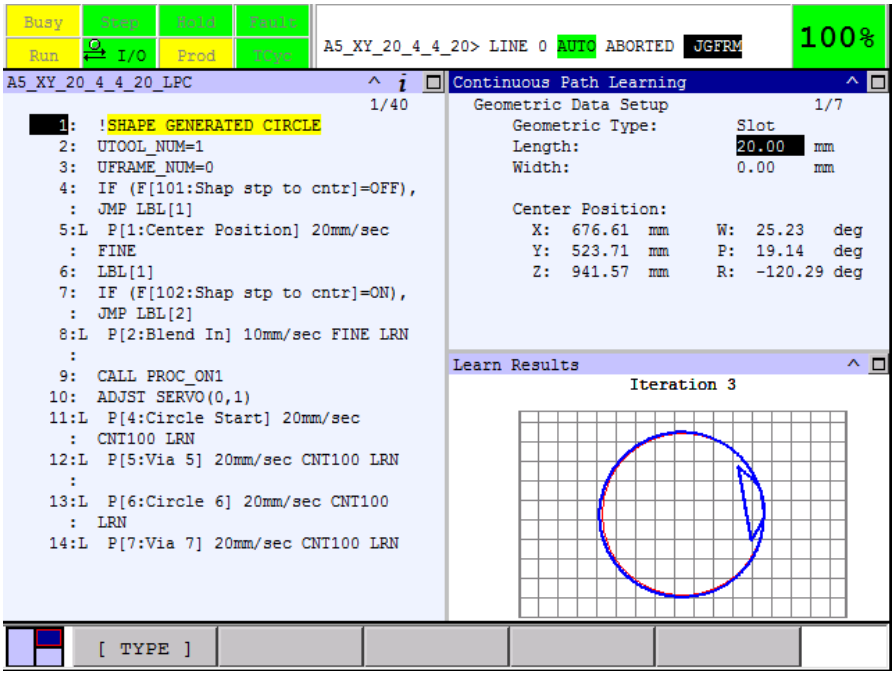
66.4 LPC LEARNING AND PLAYBACK

This chapter explains LPC learning and playback.

66.4.1 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.

66.4.2 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.

66.4.3 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.

66.5 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.

67 MQTT

67.1 OVERVIEW

To use this function, you must order “MQTT (S607)” with 7DF5P/20 or later or all versions after 7DF5 series.

MQTT is publish-subscribe network protocol that transports messages between devices.
Robot controller becomes only publisher client. Robot controller can’t become subscriber or broker or server. Only MQTT version 5.0 is supported.
FANUC will prepare KAREL built-in routines to support MQTT.
Only following MQTT Control Packets and functions are supported. Others are not supported.
(1) CONNECT
  (a) Only Clean Start is supported.
(2) CONNACK
(3) PUBLISH
  (a) QoS 0, 1 and 2 are supported.
(4) PUBACK
(5) PUBREC
(6) PUBREL
(7) PUBCOMP
(8) DISCONNECT

67.2 KAREL BUILT-IN ROUTINES

67.2.1 MQTT_CONNECT

Purpose:
Connecting to MQTT server (broker)

Syntax:
MQTT_CONNECT (ip_address, sockfd, reason_code, status)
Input/Output Parameters:
[in] ip_address: STRING
[out] sockfd: INTEGER
[out] reason_code: INTEGER
[out] status: INTEGER
%ENVIRONMENT Group: MQTT

Details:
- ip_address is the MQTT server ip address to be connected from robot controller (client).
- sockfd is the handler for this connection.
- reason_code explains the status of connect. The values are shown below.
- status explains the status of the connect operation. If not equal to 0, then an error occurred.

Table 67.2.1 (a) Reason Code returned from MQTT_CONNECT

Value	Hex	Reason Code name	Description
0	0x00	Success	The Connection is accepted.
128	0x80	Unspecified error	The Server does not wish to reveal the reason for the failure, or none of the other Reason Codes apply.
129	0x81	Malformed Packet	Data within the CONNECT packet could not be correctly parsed.
130	0x82	Protocol Error	Data in the CONNECT packet does not correct.

Value	Hex	Reason Code name	Description
131	0x83	Implementation specific error	The CONNECT is valid but is not accepted by this Server.
132	0x84	Unsupported Protocol Version	The Server does not support the version of the MQTT protocol requested by the Client.
133	0x85	Client Identifier not valid	The Client Identifier is a valid string but is not allowed by the Server.
134	0x86	Bad User Name or Password	The Server does not accept the User Name or Password specified by the Client
135	0x87	Not authorized	The Client is not authorized to connect.
136	0x88	Server unavailable	The MQTT Server is not available.
137	0x89	Server busy	The Server is busy. Try again later.
138	0x8A	Banned	This Client has been banned by administrative action. Contact the server administrator.
140	0x8C	Bad authentication method	The authentication method is not supported or does not match the authentication method currently in use.
144	0x90	Topic Name invalid	The Will Topic Name is not malformed, but is not accepted by this Server.
149	0x95	Packet too large	The CONNECT packet exceeded the maximum permissible size.
151	0x97	Quota exceeded	An implementation or administrative imposed limit has been exceeded.
153	0x99	Payload format invalid	The Will Payload does not match the specified Payload Format Indicator.
154	0x9A	Retain not supported	The Server does not support retained messages, and Will Retain was set to 1.
155	0x9B	QoS not supported	The Server does not support the QoS set in Will QoS.
156	0x9C	Use another server	The Client should temporarily use another server.
157	0x9D	Server moved	The Client should permanently use another server.
159	0x9F	Connection rate exceeded	The connection rate limit has been exceeded.

67.2.2 MQTT_PUBLISH

Purpose:

MQTT_PUBLISH is used to publish message.

Syntax:

MQTT_PUBLISH (sockfd, qos, topic_name, payload_pip, reason_code, status)
Input/Output Parameters:
[in] sockfd: INTEGER
[in] qos: INTEGER
0: At most once delivery
1: At least once delivery
2: Exactly once delivery
[in] topic_name: STRING
[in] payload_pip: STRING
[out] reason_code: INTEGER
[out] status: INTEGER
%ENVIRONMENT Group: MQTT

Details:

- sockfd is the handler output from MQTT_CONNECT KAREL built-in function.
- qos indicates the level of assurance for delivery of an Application Message (payload).
- topic_name identifies the information channel to which payload data is published.
- payload_pip identifies the pipe file name to which the payload is contained.
  Payload in PUBLISH packet contains the Application Message that is being published. The content and format of the data is application specific. To set payload, pipe mechanism is used. Written data to PIP:/* is sent as payload of PUBLISH.

A typical operation sequence includes:

```
…
VAR
   mqttfile       : FILE
BEGIN
   …
   OPEN FILE mqttfile (‘RW’, ‘PIP:/mqtt1’) -- open_mode = 'W' is not supported. open_mode should be 'RW'
   Write mqttfile (‘This is payload.’)
   MQTT_PUBLISH (sockfd, qos, topic_name, “PIP:/mqtt1”, status)
END SAMPLE
```

- reason_code explains the status of publish. The values are shown below.
- status explains the status of the connect operation. If not equal to 0, then an error occurred.

Table 62.2.2 (a) Reason Code for MQTT_PUBLISH

Value	Hex	Reason Code name	Description
0	0x00	Success	The message is accepted. Publication of the QoS 1 or 2 message proceeds. or Packet Identifier released. Publication of QoS 2 message is complete.
16	0x10	No matching subscribers	The message is accepted but there are no subscribers. This is sent only by the Server. If the Server knows that there are no matching subscribers, it MAY use this Reason Code instead of 0x00 (Success).
128	0x80	Unspecified error	The receiver does not accept the publish but either does not want to reveal the reason, or it does not match one of the other values.
131	0x83	Implementation specific error	The PUBLISH is valid but the receiver is not willing to accept it.
135	0x87	Not authorized	The PUBLISH is not authorized.
144	0x90	Topic Name invalid	The Topic Name is not malformed, but is not accepted by this Server.
145	0x91	Packet identifier in use	The Packet Identifier is already in use. This might indicate a mismatch in the Session State between the Client and Server.
146	0x92	Packet Identifier not found	The Packet Identifier is not known. This is not an error during recovery, but at other times indicates a mismatch between the Session State on the Client and Server.
151	0x97	Quota exceeded	An implementation or administrative imposed limit has been exceeded.
153	0x99	Payload format invalid	The payload format does not match the specified Payload Format Indicator.

67.2.3 MQTT_DISCON

Purpose:

MQTT_DISCON is used to disconnect connection.

Syntax:

MQTT_DISCON (sockfd, reason_code, status)
Input/Output Parameters:
[in] sockfd: INTEGER
[in, out] reason_code: INTEGER
[out] status: INTEGER
%ENVIRONMENT Group: MQTT

Details:

- sockfd is the handler output from MQTT_CONNECT KAREL built-in function.
- reason_code is sent and receive to/from the server. This explains the status of the disconnection. The values are shown below.
- status explains the status of the disconnect operation. If not equal to 0, then an error occurred.

Table 62.2.3 (a) Reason Code for MQTT_DISCON

Value	Hex	Reason Code name	In or Out	Description
0	0x00	Normal disconnection	In or Out	Close the connection normally.
128	0x80	Unspecified error	In or Out	The Connection is closed but the sender either does not wish to reveal the reason, or none of the other Reason Codes apply.
129	0x81	Malformed Packet	In or Out	The received packet does not conform the specification.
130	0x82	Protocol Error	In or Out	An unexpected or out of order packet was received.
131	0x83	Implementation specific error	In or Out	The packet received is valid but cannot be processed by this implementation.
135	0x87	Not authorized	Out	The request is not authorized.
137	0x89	Server busy	Out	The Server is busy and cannot continue processing requests from this Client.
139	0x8B	Server shutting down	Out	The Server is shutting down.
141	0x8D	Keep Alive timeout	Out	The Connection is closed because no packet has been received for 1.5 times the Keepalive time.
142	0x8E	Session taken over	Out	Another Connection using the same ClientID has connected causing this Connection to be closed.
144	0x90	Topic Name invalid	In or Out	The Topic Name is correctly formed, but is not accepted by this Client or Server.
147	0x93	Receive Maximum exceeded	In or Out	The Client or Server has received more than Receive Maximum publication for which it has not sent PUBACK or PUBCOMP.
148	0x94	Topic Alias invalid	In or Out	The Client or Server has received a PUBLISH packet containing a Topic Alias which is greater than the Maximum Topic Alias it sent in the CONNECT or CONNACK packet.
149	0x95	Packet too large	In or Out	The packet size is greater than Maximum Packet Size for this Client or Server.
150	0x96	Message rate too high	In or Out	The received data rate is too high.
151	0x97	Quota exceeded	In or Out	An implementation or administrative imposed limit has been exceeded.
152	0x98	Administrative action	In or Out	The Connection is closed due to an administrative action.
153	0x99	Payload format invalid	In or Out	The payload format does not match the one specified by the Payload Format Indicator.
155	0x9B	QoS not supported	Out	The Client specified a QoS greater than the QoS specified in a Maximum QoS in the CONNACK.
156	0x9C	Use another server	Out	The Client should temporarily change its Server.
157	0x9D	Server moved	Out	The Server is moved and the Client should permanently change its server location.
159	0x9F	Connection rate exceeded	Out	This connection is closed because the connection rate is too high.
160	0xA0	Maximum connect time	Out	The maximum connection time authorized for this connection has been exceeded.

67.3 KAREL PROGRAM SAMPLE

```
PROGRAM SAMPLE

%NOLOCKGROUP
%ENVIRONMENT MQTT

VAR
   sockfd      : INTEGER
   reason_code : INTEGER
   status      : INTEGER
   qos         : INTEGER
```

```
   count          : INTEGER
   console        : FILE
   mqttfile       : FILE
   ip_address     : STRING[15]

------------------------------------------------------------
---------- MAIN FUNCTION ----------
------------------------------------------------------------
BEGIN
   IF uninit(count) THEN
      count = 0
   ENDIF

   IF uninit(qos) THEN
      qos = 1
   ENDIF

   OPEN FILE console('RW','CONS:')

   REPEAT
      ip_address = '10.18.57.48'
      MQTT_CONNECT(ip_address, sockfd, reason_code, status)
      IF (reason_code <> 0) OR (status <> 0) THEN
         write console('MQTT_CONNECT reason_code = ', reason_code, ' status = ', status, cr)
         RETURN
      ENDIF

      OPEN FILE mqttfile ('RW', 'PIP:mqtt1') -- open_mode = 'W' is not supported. open_mode should be 'RW'

      status = IO_STATUS(mqttfile)
      IF status <> 0 THEN
         WRITE CONSOLE('open ', status, cr)
         RETURN
      ENDIF

      Write mqttfile ('This is payload. count = ', count)
      count = count + 1

      status = IO_STATUS(mqttfile)
      IF status   <> 0THEN
         WRITE CONSOLE('write ', status, cr)
         RETURN
      ENDIF

      CLOSE FILE mqttfile

      MQTT_PUBLISH(sockfd, qos, 'topic_name1/topic_name2', 'PIP:mqtt1', reason_code, status)
      IF (reason_code <> 0) OR (status <> 0) THEN
         write console('MQTT_PUBLISH reason_code = ', reason_code, ' status = ', status, cr)
         RETURN
      ENDIF

      reason_code = 0
      MQTT_DISCON(sockfd, reason_code, status)
      IF (reason_code <> 0) OR (status <> 0) THEN
         write console('MQTT_DISCON reason_code = ', reason_code, ' status = ', status, cr)
         RETURN
      ENDIF
```

```
    DELAY 100
  UNTIL FALSE -- If this is changed to FALSE then error may happen after a while
              -- especially when DELAY time is low but this is not bug.
              -- This is a limitation of TCP/IP to take a while to close socket.

  CLOSE FILE console
END SAMPLE
```

68 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. The maximum output cycle can be set to 2 msec.
To use this function, R891 must be ordered. This function uses one of the controller's regular Ethernet communication ports (port 1 or port 2) or the vision communication port (port 3).

NOTE
The software that runs on the receiving PC used to receive the packets must be provided by the system designer.

NOTE
For high-speed communication such as 2msec cycle, it is necessary to use the vision communication port (port 3), and only binary data format can be used. If this feature does not use port 3, it will usually display a warning and will not enable the feature if it is set faster than 8msec. If it is a system configuration such as multi-group, it may be necessary to set a cycle longer than 8 msec.

68.1 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 of 2msec, please select one data type only to minimize CPU loading. For a slower data rate, more than one data type can be output.

68.1.1 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

68.1.1.1 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)]

68.1.1.2 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)]

68.1.1.3 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)

68.1.2 XML

68.1.2.1 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>
```

68.1.2.2 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><J7>*</J7><J8>*</J8><J9>*</J9></G1></ANG><CYCLE>*</CYCLE><IO>*</IO>
```

68.1.2.3 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>
```

68.2 SETUP

High Speed Position Output is set up by changing the system variables. After setting the $MCGR_CFG and $MCGR_SET variables as described below, change $MCGR_CFG.$ENABLE from FALSE to TRUE to start communication.

NOTE
This feature must be disabled in order to change the settings.

68.2.1 Destination Information Settings

Change the following system variables to register the destination IP address and port number. After making these changes, restart the controller.

System variables used to set destination information

System variables	Description
$MCGR_CFG.$SERVER_NAME	External process IP address
$MCGR_CFG.$PORT	External process port number

68.2.2 Communication Method

Change the following system variables to set the communication method.

System variables used to set the communication method

System variables	Description
$MCGR_CFG.$PROTOCOL	Select a protocol from the following. 0: UDP 1: TCP
$MCGR_CFG.$RATE	Data sending rate in increment of 2 msec. The fastest rate is 2msec. For 2msec, set to 2. For 8msec, set to 8, etc. (default: 8)
$MCGR_CFG.$MODE	0 : Binary 1: XML
$MCGR_CFG.$OUT_TYPE	Select the data type to output. If you select multiple types, set the sum of each value. 1 : TCP Cartesian position 4 : Joint angles 16: Variables
$MCGR_CFG.$GROUP_MSK	Set the motion group of the output data. If multiple groups are output, set the sum of each value. 1 : G1 16 : G5 2 : G2 32 : G6 4 : G3 64 : G7 8 : G4 128 : G8
$MCGR_CFG.$ENDIAN	0 : Big Endian 1 : Little Endian
$MCGR_CFG.$PROC_TYPE	0 : Using port 3 1 : Using port 1 or 2

68.2.3 Output Data Setup

When TCP Cartesian position or Joint angles is output, the following system variables can be set.

System variables when TCP Cartesian Position or Joint Angles is output

System variables	Description
$MCGR_SET.$CYCLE_TYPE	The I/O port type available for sending the program status. By default, this is 21, which is the type value for UO signals. This is the port type for the I/O point sent as STATUS or CYCLE in the packet definitions.
$MCGR_SET.$CYCLE_PORT	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 $MCGR_SET.$CYCLE_TYPE.
$MCGR_SET.$IO_TYPE	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. 1 : DI [ ] 2 : DO [ ] 3 : AI [ ] 4 : AO [ ] 8 : RI [ ] 9 : RO [ ] 18 : GI [ ] 19 : GO [ ] 20 : UI [ ] 21 : UO [ ] 35 : F [ ]
$MCGR_SET.$IO_PORT	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.
$MCGR_SET.$FRAME_USE	This value is the User Frame number to use when outputting Cartesian positions. The default of 0 is for the World frame.

When Variables is output, changing the following system variables will select the variables to output.

System variables used to output variables

System variables	Description
$MCGR_SET.$VAR_MASK	This mask indicates which of the variables configured in $VAR_DATA[ ] are defined. When sending multiple values, set the sum of each value. (For example, if $VAR_MASK = 1023, then all ten variables have been defined) 1 : $VAR_DATA[1] 32 : $VAR_DATA[6] 2 : $VAR_DATA[2] 64 : $VAR_DATA[7] 4 : $VAR_DATA[3] 128 : $VAR_DATA[8] 8 : $VAR_DATA[4] 256 : $VAR_DATA[9] 16 : $VAR_DATA[5] 512 : $VAR_DATA[10]
$MCGR_SET.$VAR_USED (read-only)	This mask indicates which variables configured in $VAR_DATA[ ] are valid
$MCGR_SET.$VAR_DATA[1-10].$PROG_NAME	The name of the program to which the variable belongs. This can be a KAREL program name or "*system*" for system variables.
$MCGR_SET.$VAR_DATA[1-10].$VAR_NAME	The name of the variable.

68.2.4 Start Output

When $ MCGR_CFG $ ENABLE is changed from FALSE to TRUE after all the settings are completed, it connects to the destination and starts data output. To end the communication, change $MCGR_CFG.$ENABLE to FALSE.

69 DIGITAL I/O JOG FUNCTION

69.1 OVERVIEW

Digital I/O Jog Function (J543) is the function that can do jog operation according to external digital signal input.

69.2 SPECIFICATION

69.2.1 Version

R-30*i*B、R-30*i*B Mate : after 7DC3/36
R-30*i*B plus、R-30*i*B Mate plus : after 7DF1/05

69.2.2 Setup

Below table shows the system variables for setup and those meanings.

Table 69.2.2 (a) System variables for setup

item	explaination
$MTSI.$ENABLE	1：enable function, 0：disable function (No need cycle power)
$MTSI.$AUX_MODE	0: Specify to use axis 7 to J9 as J7 to J9 of Group1. 1: Specify to use axis 7 to J9 as J1 to J3 of Group2. 2: Enable to specify the jog group by using Group Signal Input (GI). (Supported for multi arm)
$MTSI. $DI_INDEX[1]～[26]	System variable to specify the index of digital signal input (DI). (For actual usage, please refer to table2 in the next page)
$MTSI. $MULTGRP_GIN	The system variable to specify the index of GI which is used to specify group number in case of $MTSI.$AUX_MODE = 2.

Below table shows the system variables to specify index of DI, and those meanings.
The setup in below table2 shows the example in case that specify DI[41] to DI[66] for DI jog.

Table 69.2.2(b) System variables to specify index of DI

The role of DI signals	DI index to use DI jog	System variables to specify the index of DI	The value of system variables
Enable	DI[41]	$MTSI.$DI_INDEX[1]	41
Specify JOINT Frame	DI[42]	$MTSI.$DI_INDEX[2]	42
Specify JOG Frame	DI[43]	$MTSI.$DI_INDEX[3]	43
Specify JOG Frame 1	DI[44]	$MTSI.$DI_INDEX[4]	44
Specify JOG Frame 2	DI[45]	$MTSI.$DI_INDEX[5]	45
Specify JOG Frame 3	DI[46]	$MTSI.$DI_INDEX[6]	46
Specify JOG Frame 4	DI[47]	$MTSI.$DI_INDEX[7]	47
Specify JOG Frame 5	DI[48]	$MTSI.$DI_INDEX[8]	48
J1+/X+	DI[49]	$MTSI.$DI_INDEX[9]	49
J1-/X-	DI[50]	$MTSI.$DI_INDEX[10]	50
J2+/Y+	DI[51]	$MTSI.$DI_INDEX[11]	51
J2-/Y-	DI[52]	$MTSI.$DI_INDEX[12]	52
J3+/Z+	DI[53]	$MTSI.$DI_INDEX[13]	53

The role of DI signals	DI index to use DI jog	System variables to specify the index of DI	The value of system variables
J3-/Z-	DI[54]	$MTSI.$DI_INDEX[14]	54
J4+/W+	DI[55]	$MTSI.$DI_INDEX[15]	55
J4-/W-	DI[56]	$MTSI.$DI_INDEX[16]	56
J5+/P+	DI[57]	$MTSI.$DI_INDEX[17]	57
J5-/P-	DI[58]	$MTSI.$DI_INDEX[18]	58
J6+/R+	DI[59]	$MTSI.$DI_INDEX[19]	59
J6-/R-	DI[60]	$MTSI.$DI_INDEX[20]	60
Axis 7+	DI[61]	$MTSI.$DI_INDEX[21]	61
Axis 7-	DI[62]	$MTSI.$DI_INDEX[22]	62
Axis 8+	DI[63]	$MTSI.$DI_INDEX[23]	63
Axis 8-	DI[64]	$MTSI.$DI_INDEX[24]	64
Axis 9+	DI[65]	$MTSI.$DI_INDEX[25]	65
Axis 9-	DI[66]	$MTSI.$DI_INDEX[26]	66

69.2.3 Limitations

- Digital I/O jog enables jog when below conditions are all satisfied.
  - Enable signal ON
  - Remote condition is valid
  - $MTSI.$ENABLE = TRUE
- When “Specify JOINT Frame DI” and “Specify JOG Frame DI” are both ON or both OFF simultaneously, jog will be stopped

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

- “Specify JOG Frame” signal can specify JOG Frame only in below cases. In other cases than below, jog will be stopped. For example, JOG Frame1 and JOG Frame 2 were selected simultaneously.

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 same time, robot will move to all specified directions.
  Example: If +X direction and +Y direction were input at same time, robot will move to +X and +Y at same time.
- If JOG key was pressed during jog motion, 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.

70 LASER INTERFACE FUNCTION

70.1 OVERVIEW

This function is for controlling the 3rd party laser oscillator from the robot.

70.1.1 Necessary Software Option

The application tools that support this function are the Handling Tool, LR Handling Tool, Arc Tool, and LR Arc Tool.

The following software options are required for this function.

Table 70.1.1 (a) Necessary software option

Name	Option number
Laser Interface Function（This Function）	A05B-2670-S555

70.1.2 Caution

70.1.2.1 For the safety

This function does not output a laser command when teach pendant is enabled or when servo power is turned off due to an emergency stop etc. This is to reassure the programmers or maintenance technicians. For details, refer to the “LASER ENABLE/DISABLE” section.

⚠ WARNING
If you use this function, it is necessary to prepare the interlock with sufficient consideration for user’s safety.

70.1.2.2 For multi-task

NOTE
This function is not compatible with multitasking (Laser processing cannot be performed simultaneously by multiple programs.).

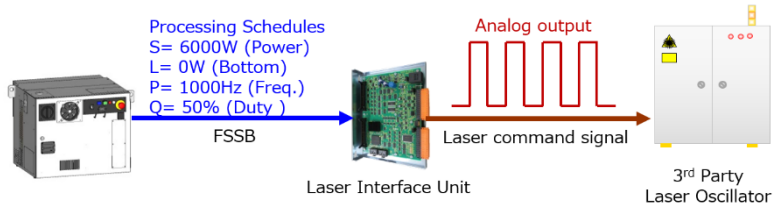
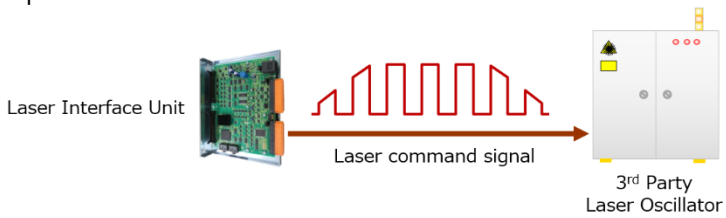
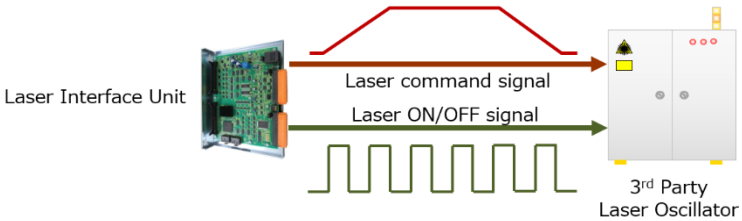
70.2 SETUP

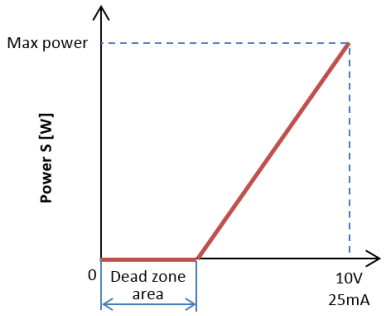
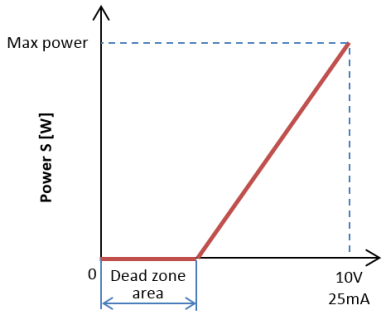
The setup for this function includes laser initial setup and laser setup.

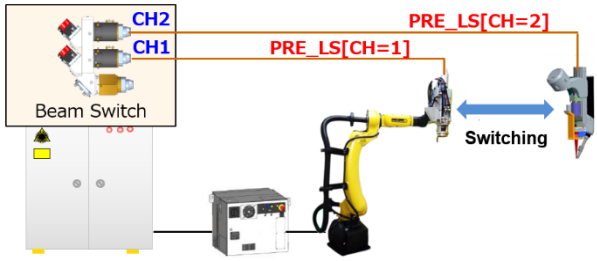
70.2.1 Laser Initial Setup

Laser initial setup is the setting which is the base of this function. For safety, you can only change laser initial setup on the laser initial setup screen of the controlled start menu. After the normal start, you cannot change the laser initial setup, you can check only the setting state.

Table 70.2.1 Laser initial setup screen

Items	Descriptions
Manufacturer	Currently, you can only select IPG or FANUC. Select IPG to control the IPG laser oscillator with this function. If FANUC is selected here, the screen will switch to the laser initial setup screen for the “FANUC Fiber Laser Interface Function (A05B-2600-J922)”. See also the section “Switching to FANUC Fiber Laser Interface Function” below.
Model	Specify the model of the laser oscillator. Currently, you can only select YLS or YLR.
Laser command interface	(Currently, you can only select FSSB.) When FSSB is specified to this item, the processing schedules (S, L, P, Q) are transmitted to the laser interface unit via the FSSB cable, and the laser interface unit outputs a laser command to the laser oscillator as analog output.  For more information on the laser interface unit, see the “LASER INTERFACE UNIT” section at the end of this chapter.
Analog output type	You can select whether the analog output of the laser command signal is pulse or continuous. If PULSE is specified in this item, the analog output of the laser command signal will be a pulse.  If CONT. is specified for this item, the analog output of the laser command signal will be continuous. If you want the laser to be a pulse wave, combine the analog output of the laser command signal with the digital output of the laser ON/OFF signal.  For more information on the laser command signal and laser ON/OFF signale, see the “LASER INTERFACE UNIT” section at the end of this chapter.

Items	Descriptions
Dead zone area(V)	Even if the robot commands 0V analog voltage (0W power), the laser may be unintentionally emitted because the analog voltage command is affected by noise depending on the usage environment. To avoid this situation, the laser oscillator may have the ability to make a dead zone area. The same value as the dead zone area of the laser oscillator used must be set to here. The value specified here is used to define the relationship between power S and laser command signal analog output.  **Laser command signal analog output (VOUT0, IOUT)**
Maximum power(W)	The maximum value of power. If a power exceeding this value is commanded, the power is internally clamped to this value and output. The same value as the maximum power of the laser oscillator used must be set to here. The value specified here is used to define the relationship between power S and laser command signal analog output.  **Laser command signal analog output (VOUT0, IOUT)**
Maximum frequency(Hz)	The maximum value of frequency. When a frequency exceeding this value is commanded, the frequency is internally clamped to this value and output.
Processing speed unit	Set the unit of processing speed of argument V on the LS instruction. Set this item when using the speed tracking function such as the power control function and the wire control function where the processing speed is used. If you do not use the speed following function, you do not need to set this item.

Items	Descriptions
Time-sharing	If the laser oscillator has a beam switch and one laser oscillator is shared by multiple robot controllers, specify this item to ENABLE. If this item is ENABLE, "Laser Request" output[b] turns on when PRE_LS (LS[a]) instruction is executed, and "Laser Request" output turns off when LE instruction is executed. If this item is DISABLE, "Laser Request" output will always be on.
Beam switch CH switching	When this item is ENABLE, beam switch CH number can be switched with the PRE_LS instruction. Enable this if you want to dynamically switch between multiple processing heads with one robot controller. 

NOTE
a. When the LS instruction is executed without executing the PRE_LS instruction.
b. This is YLS laser oscillator I/O. See the section "YLS Laser Oscillator I/O".

Procedure 70-1 Display laser initial setup screen after controlled start

Step

1 Perform controlled start.
2 After controlled start, press the [MENU] key. Controlled start menu will be displayed.
3 Select [0 – NEXT --]. Select [4 Laser Initial Setup].

```
Laser Initial Setup
                                                    1/10

   1  Manufacturer:                                 IPG
   2  Model:                                        YLS
   3  Laser command interface:                     FSSB
   4  Analog output type:                         PULSE
   5  Dead zone area(V):                            0.0
   6  Maximum power(W):                            6000
   7  Maximum frequency(Hz):                      10000
   8  Processing speed unit:                      mm/sec
   9  Time-sharing:                              DISABLE
  10  Beam switch CH switching:                  DISABLE

[ TYPE ]                          [CHOICE]
```

Procedure 70-2 Display laser initial setup screen after Normal Start

Step

1 Press the [MENU] key then screen menu will be displayed.
2 Select [6 SETUP].
3 Show Menu by pressed F1 [TYPE].
4 Select “Laser Setup”.

```
SETUP Laser
                                                    1/3

   1  Laser Initial Setup:          <* DETAIL *>
   2  Laser Setup:                  <* DETAIL *>
   3  Wire Control Setup:           <* DETAIL *>

[ TYPE ]
```

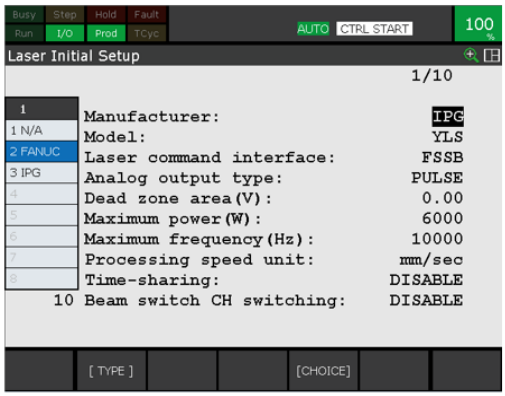
5 Move cursor on <* DETAIL *> of “Laser Initial Setup” and press [ENTER] key.

```
SETUP Laser Initial Setup
                                                   1/10

   1  Manufacturer:                              IPG
   2  Model:                                     YLS
   3  Laser command interface:                  FSSB
   4  Analog output type:                      PULSE
   5  Dead zone area(V):                         0.0
   6  Maximum power(W):                         6000
   7  Maximum frequency(Hz):                   10000
   8  Processing speed unit:                   mm/sec
   9  Time-sharing:                          DISABLE
  10  Beam switch CH switching:              DISABLE

[ TYPE ]
```

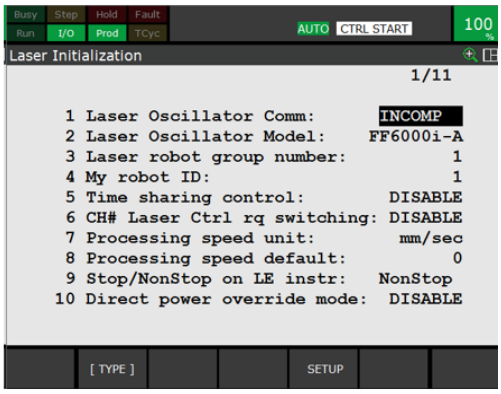
70.2.1.1 Switching to FANUC Fiber Laser Interface Function

When connecting a FANUC fiber laser oscillator, FANUC Fiber Laser Interface Function is required. When this function is installed, FANUC Fiber Laser Interface Function is also installed at the same time.
If you want to switch from this function to FANUC Fiber Laser Interface Function, specify the Manufacturer on the laser initial setup screen to FANUC.

Laser Interface Function

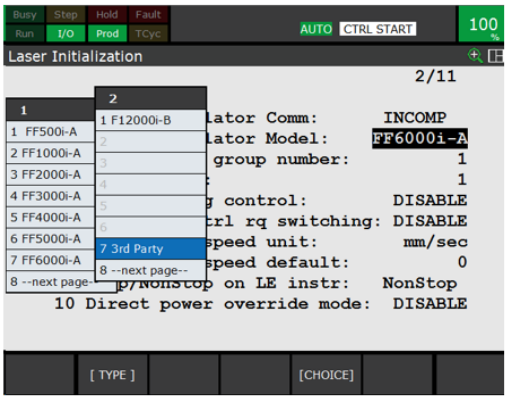


FANUC Fiber Laser Interface Function

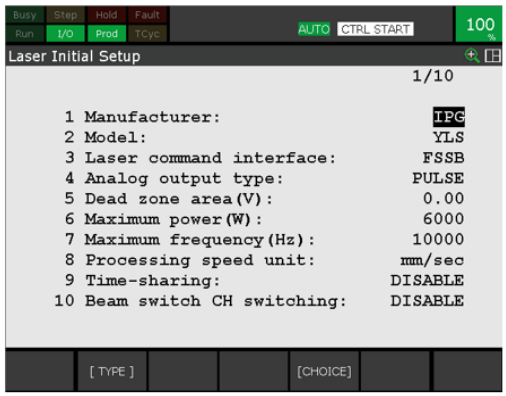


You can return to this function by specifying to 3rd Party in the Laser Oscillator Model on the laser initial setup screen of FANUC Fiber Laser Interface Function.

FANUC Fiber Laser Interface Function



Laser Interface Function



70.2.2 Laser Setup

Laser Setup are mainly related to laser processing. Make the settings as necessary.

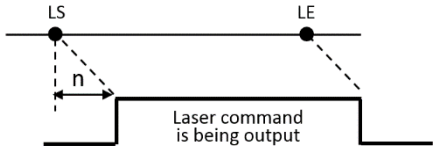
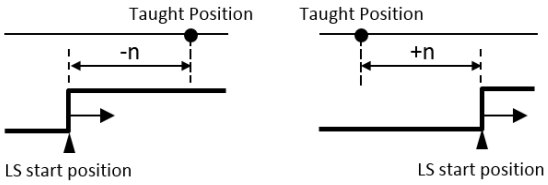
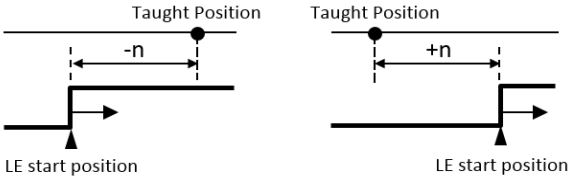
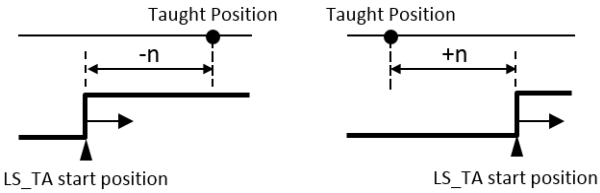
NOTE
The contents of Laser Setup differ depending on the model settings on the laser initial setup screen.

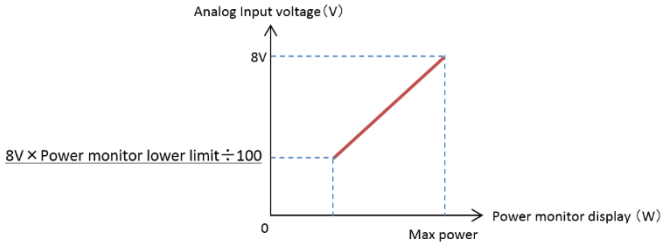
70.2.2.1 YLS Laser setup

When the model of the laser initial setup screen is specified as "YLS", the following items are displayed on the laser setup screen.

Table 70.2.2.1 YLS Laser setup screen

Items	Descriptions
Laser robot group	Specify the group number of the robot that has the laser processing head and processes the laser. If you want to change the Laser robot group in the TP program, execute the KAREL program CHG_LSR_GRP before the PRE_LS instruction. Group number to change 10: CALL CHG_LSR_GRP(2) 11: L P[5]200mm/sec FINE PRE_LS
Beam switch channel number	Specify the channel number of the beam switch used by this robot. The value set here is always output from “Beam Switch CH” output[b]. If PRE_LS[CH=*] instruction is executed, this item will be overwritten to the value of argument CH.
Process disable for restart	If this item is enabled, laser processing will not restart when program is restarted after being paused in the laser processing section (including processing preparation). Therefore, “Laser ON” output[b], “Analog Control ON” output[b], “Program Start” output[b], “Air/Gas” output[c], “Wire Feed” output[c] and “Wire Feed Speed” output[c] will not be turned on when restarting. Also, it does not check whether “Laser is ON” input[b], “Analog Ctrl is ON” input[b], “Laser Ready” input[b], “Emission is ON” input[b], “Air/Gas” input[c], and “Wire Feed” input[c] are on. If you do not want to restart laser processing, enable this item.

Items	Descriptions
Air/Gas OK latency(s)	If "Air/Gas" input[c] is assigned, check whether "Air/Gas" input is on after executing the LS instruction. If "Air/Gas" input does not turn on after the time set here, an alarm is occurred.
Processing state record	When this item is ENABLE, record of the laser processing state begins. Refer also to the section "PROCESSING STATE RECORD FUNCTION" below.
Guide laser auto OFF timer(s)	This item is a function for saving the guide laser life. "Guide Laser ON" output[b] turns off automatically after the time set here has elapsed.If this item is set to 0, this function is disabled and "Guide Laser ON" output is not turned off.
Laser command delay time(ms)	This item can delay the output of the laser command for a certain period of time.  The minimum unit for this item is ITP (it cannot be fine-tuned than ITP).
LS start time adjustment(ms)	Adjust the start timing of the motion option LS instruction.  The minimum unit for this item is ITP (it cannot be fine-tuned than ITP).
LE start time adjustment(ms)	Adjust the start timing of the motion option LE instruction.  The minimum unit for this item is ITP (it cannot be fine-tuned than ITP).
LSTA start time adjustment(ms)	Adjust the start timing of LS_TA (LS_SCH_TA) instruction. Refer also to the section "LS_TA INSTRUCTION WITH TIME ADJUSTMENT" below.  The minimum unit for this item is 1mec.

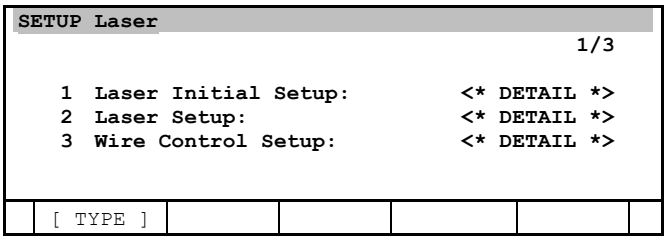
Items	Descriptions
Laser assigned DI latency(s)	In a system where the laser oscillator is equipped with a beam switch, laser processing cannot be started even if this robot executes the PRE_LS (LS[a]) instruction while another robot is performing laser processing. Therefore, it waits on the execution line of the PRE_LS (LS) instruction until the laser processing of other robots is completed (waits until "Laser is Assigned" input[b] turns on). This item is the limit value of the waiting time until "Laser is Assigned" input turns on. If "Laser is Assigned" input does not turn on after the time set here has elapsed, an alarm is occurred. If this item is set to 0, it will continue to wait until "Laser is Assigned" input turns on.
Laser ON DI latency(s)	This item is the limit value of the waiting time until "Laser is ON" input[b] is turned on. The robot waits on the execution line of the PRE_LS (LS[a]) instruction until "Laser is ON" input is turned on. If "Laser is ON" input does not turn on after the time set here has elapsed, an alarm is occurred. If this item is set to 0, it will continue to wait until "Laser is ON" input turns on.
Laser Ready DI latency(s)	This item is the limit value of the waiting time until "Laser Ready" input[b] is turned on. The robot waits on the execution line of the PRE_LS (LS[a]) instruction until "Laser Ready" input is turned on. If "Laser Ready" input does not turn on after the time set here has elapsed, an alarm is occurred. If this item is set to 0, it will continue to wait until "Laser Ready" input turns on.
Power monitor lower limit(%)	This item is lower limit of analog input voltage for power monitor input[b]. If analog input voltage falls below lower limit set here, power monitor will display 0W. 

NOTE
a. When the LS instruction is executed without executing the PRE_LS instruction.
b. This is YLS laser oscillator I/O. See the section "YLS Laser Oscillator I/O".
c. This is peripheral equipment I/O. See the section "Peripheral Equipment I/O".

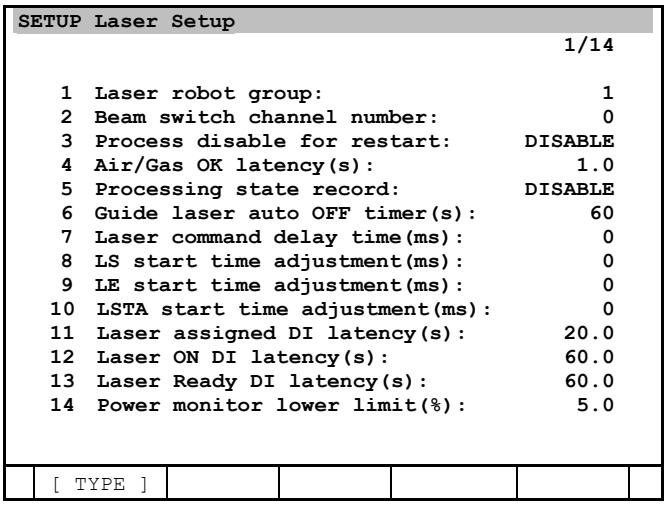
Procedure 70-3 Display YLS laser setup screen

Step

1 Press the [MENU] key then screen menu will be displayed.
2 Select [6 SETUP].
3 Show Menu by pressed F1 [TYPE].
4 Select "Laser Setup".



5 Move cursor on <* DETAIL *> of “Laser Setup” and press [ENTER] key.

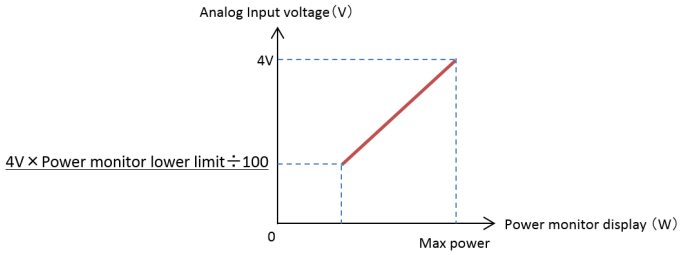


70.2.2.2 YLR Laser setup

When the model of the laser initial setup screen is specified as "YLR", the following items are displayed on the laser setup screen.

Table 70.2.2.2 YLR Laser setup screen

Items	Descriptions
Laser robot group	It is common with YLS. See the section "YLS Laser setup".
Beam switch channel number	It is common with YLS. See the section "YLS Laser setup".
Process disable for restart	If this item is enabled, laser processing will not restart when program is restarted after being paused in the laser processing section (including processing preparation). Therefore, “Interlock Ch1” output[b], “Interlock Ch2” output[b], “Remote Start” output[b], “Emission Enable” output[b], “Modulation” output[b], “Air/Gas” output[c], “Wire Feed” output[c] and “Wire Feed Speed” output[c] will not be turned on when restarting. Also, it does not check whether “Power On” input[b], “P_S_Active” input[b], “Emission ON” input[b], “Air/Gas” input[c], and “Wire Feed” input[c] are on. If you do not want to restart laser processing, enable this item.
Air/Gas OK latency(s)	It is common with YLS. See the section "YLS Laser setup".
Processing state record	It is common with YLS. See the section "YLS Laser setup".
Guide laser auto OFF timer(s)	It is common with YLS. See the section "YLS Laser setup".
Laser command delay time(ms)	It is common with YLS. See the section "YLS Laser setup".
LS start time adjustment(ms)	It is common with YLS. See the section "YLS Laser setup".
LE start time adjustment(ms)	It is common with YLS. See the section "YLS Laser setup".
LSTA start time adjustment(ms)	It is common with YLS. See the section "YLS Laser setup".

Items	Descriptions
P_S_Active DI latency(s)	This item is the limit value of the waiting time until “P_S_Active” input[b] is turned on. The robot waits on the execution line of the PRE_LS (LS[a]) instruction until “P_S_Active” input is turned on. If “P_S_Active” input does not turn on after the time set here has elapsed, an alarm is occurred. If this item is set to 0, it will continue to wait until “P_S_Active” input turns on.
Power monitor lower limit(%)	This item is lower limit of analog input voltage for power monitor input[b]. If analog input voltage falls below lower limit set here, power monitor will display 0W. 

NOTE
a. When the LS instruction is executed without executing the PRE_LS instruction.
b. This is YLR laser oscillator I/O. See the section “YLR Laser Oscillator I/O”.

```
SETUP Laser Setup
                                                    1/12

   1  Laser robot group:                              1
   2  Beam switch channel number:                     0
   3  Process disable for restart:              DISABLE
   4  Air/Gas OK latency(s):                        1.0
   5  Processing state record:                  DISABLE
   6  Guide laser auto OFF timer(s):                 60
   7  Laser command delay time(ms):                   0
   8  LS start time adjustment(ms):                   0
   9  LE start time adjustment(ms):                   0
  10  LSTA start time adjustment(ms):                 0
  11  P_S_Active DI latency(s):                    60.0
  12  Power monitor lower limit(%):                 5.0

 [ TYPE ]
```

The procedure for displaying the YLR laser setup screen is the same as for YLS. See the section "YLS Laser Setup".

70.3 LASER I/O

This function has laser oscillator I/O and peripheral equipment I/O.

NOTE
The contents of Laser oscillator I/O differ depending on the model settings on the laser initial setup screen.

70.3.1 YLS Laser Oscillator I/O

By assigning YLS laser oscillator I/O, it becomes easy to control YLS from the robot.

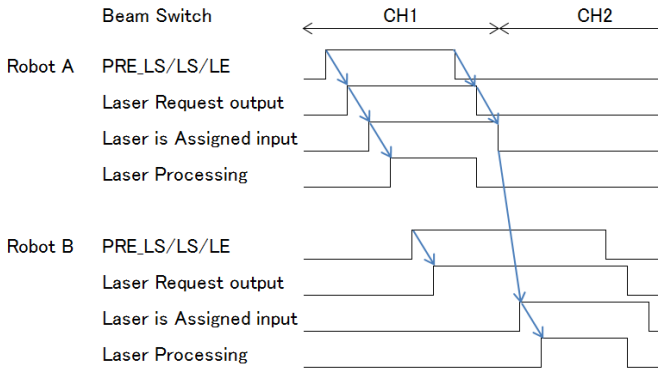
NOTE
- It is not necessary to assign all signals. Assign only the signals you need.
- Even if all signals are not assigned, if the Laser Enable/Disable[a] is Enable, the robot can output the laser command.

70.3.1.1 YLS laser oscillator output screen

When the model of the laser initial setup screen is specified as "YLS", the following items are displayed on the laser oscillator output screen.

Table 70.3.1.1 YLS laser oscillator output screen (Robot➔YLS)

Items	Descriptions
S/Power(W)	Here you can see the power being sent from the robot to the laser interface unit via FSSB. When LS instruction is executed, the value specified for S= in the LS instruction is sent to the laser interface unit.
L/Bottom Power(W)	Here you can see the bottom power being sent from the robot to the laser interface unit via FSSB. When LS instruction is executed, the value specified for L= in the LS instruction is sent to the laser interface unit.
P/Frequency(Hz)	Here you can see the frequency being sent from the robot to the laser interface unit via FSSB. When LS instruction is executed, the value specified for P= in LS instruction is sent to the laser interface unit.
Q/Duty(%)	Here you can see the duty being sent from the robot to the laser interface unit via the FSSB. When LS instruction is executed, the value specified for Q= in the LS instruction is sent to the laser interface unit.
Laser ENBL/DSBL	This signal is used to notify the laser oscillator of the robot's Laser Enable/Disable state[a]. When this signal is on, the robot is ready to output laser commands. Instead of “Laser ON” output, this signal can be input to the laser oscillator to control the laser oscillator state (Active or Inactive).

Items	Descriptions
Laser Request	If “Time-sharing” on the laser initial setup screen is set to DISABLE, this signal will always be on. If “Beam switch” on the laser initial setup screen is specified as ENABLE, this signal is turned on when PRE_LS (LS[b]) instruction is executed. Then, it waits on the execution line of the PRE_LS (LS) instruction until “Laser is Assigned” input turns on, and starts laser processing when “Laser is Assigned” input turns on. When LE instruction is executed, this signal is turned off so that other robots can perform laser processing.  In this way, “Laser Request” output is automatically turned on/off by laser instruction, so normally the customer does not need to be aware of this signal. In addition, “Laser Request” output is switched automatically in the following pattern. Teach pendant is enabled: “Laser Request” output is turned on while teach pendant is enabled. This is to turn off “Laser is ON” output and make the laser oscillator state idle. This is also because “Guide Laser ON” output is used to teach the laser processing part. When teach pendant is switched to disabled, “Laser Request” output signal is turned off. Forcibly terminate the program: If program is forcibly terminated during laser processing, “Laser Request” output signal will be turned off.
Laser ON	This signal controls the laser oscillator state (Active or Inactive). When Laser Enable/Disable state[a] is Enable, this signal is turned on by the PRE_LS (LS[b]) instruction. If Laser Enable/Disable state is Disable, this signal will not be turned on by the PRE_LS (LS) instruction. If Laser Enable/Disable state is Disable, this signal will be forcibly turned off.
Reset	This signal resets the error of the laser oscillator (the signal for turning off the “Laser Error” input). A pulse is output by the reset key input of the teach pendant or the input of the dedicated external signal FAULT_RESET UI[5].
Guide Laser ON	This signal controls the turning on/off of the guide laser. When this signal is assigned, the “Guide laser auto OFF timer” on YLS laser setup screen will function. This signal is forcibly turned off when the Laser Enable/Disable state[a] is Enable.
Analog Control ON	This signal is used to enable analog control in the laser oscillator. When Laser Enable/Disable state[a] is Enable, this signal is turned on by PRE_LS (LS[b]) instruction and turned off by LE instruction. If Laser Enable/Disable state is Disable, this signal will not be turned on by PRE_LS (LS) instruction.
Program Start	This is a signal to allow the laser output to the laser oscillator. When Laser Enable/Disable state[a] is Enable, this signal is turned on by PRE_LS (LS[b]) instruction and turned off by LE instruction. If Laser Enable/Disable state is Disable, this signal will not be turned on by PRE_LS (LS) instruction.

Items	Descriptions
Beam Switch CH	The value specified in “Beam switch channel number” on YLS laser setup screen is output from this signal.

NOTE
a. For details, refer to the “LASER ENABLE/DISABLE” section.
b. When the LS instruction is executed without executing the PRE_LS instruction.

Procedure 70-4 Display YLS laser oscillator output screen

Step

1 Press the [MENU] key then screen menu will be displayed.
2 Select [5 I/O].
3 Show Menu by pressed F1 [TYPE].
4 Select “Laser I/O”.

```
I/O Laser
                                                    1/4

   1  Laser Oscillator Out:          <* DETAIL *>
   2  Laser Oscillator In:           <* DETAIL *>
   3  Peripheral Equipment Out:      <* DETAIL *>
   4  Peripheral Equipment In:       <* DETAIL *>

 [ TYPE ]
```

5 Move cursor on <* DETAIL *> of “Laser Oscillator Out” and press [ENTER] key.

YLS laser oscillator output screen [Robot→YLS]

```
I/O Laser Oscillator Out
                                                    1/8
     NAME                  OUT PT     SIM      STATE
   S/Power(W):             FSSB                   0
   L/Bottom Power(W):      FSSB                   0
   P/Frequency(Hz):        FSSB                   0
   Q/Duty(%):              FSSB                   0
 1 Laser ENBL/DSBL:        DO[  0]    U        ***
 2 Laser Request:          DO[  0]    U        ***
 3 Laser ON:               DO[  0]    U        ***
 4 Reset:                  DO[  0]    U        ***
 5 Guide Laser ON:         DO[  0]    U        ***
 6 Analog Control ON:      DO[  0]    U        ***
 7 Program Start:          DO[  0]    U        ***
 8 Beam Switch CH:         GO[  0]    U        *****

 [ TYPE ]          IN/OUT
```

70.3.1.2 YLS laser oscillator input screen

When the model of the laser initial setup screen is specified as "YLS", the following items are displayed on the laser oscillator input screen.

Table 70.3.1.2 YLS laser oscillator input screen (YLS➔Robot)

Items	Descriptions
Power Monitor(W)	Assign the signal for the power monitor to here. Input type can be selected from AI or GI. When AI is selected as the input type, the value obtained by converting the analog input voltage (V) to power (W) is displayed. (0-8V = 0W-Maximum power) If GI is selected as the input type, the value of the group input signal will be displayed as is.
Laser is Assigned	This signal is an acknowledgement signal for “Laser Request” output. When Laser Enable/Disable state[a] is Enable, it keeps waiting for this signal to turn on with PRE_LS (LS[b]) instruction during “Laser assigned DI latency” on YLS laser setup screen. When Laser Enable/Disable state is Disable, this signal will not be checked.
Laser is ON	This signal is an acknowledgement signal for “Laser ON” output. Assign a signal that is turned on when the laser oscillator status becomes active. When Laser Enable/Disable state[a] is Enable, it keeps waiting for this signal to turn on with PRE_LS (LS[b]) instruction during “Laser ON DI latency” on YLS laser setup screen. When Laser Enable/Disable state is Disable, this signal will not be checked.
Laser Error	Assign signal to turn on when an error occurs in the laser oscillator. If you assign this signal, the robot will always check the status of this signal. When this signal is turned on during laser processing, an alarm is occurred and laser processing is stopped. In other cases, a warning is occurred when this signal turns on.
Guide Laser is ON	This signal is an acknowledgement signal for “Guide Laser ON” output. Assign a signal that will be turned on when the guide laser is turned on.
Analog Ctrl is ON	This signal is an acknowledgement signal for “Analog Control ON” output. Assign a signal that will be turned on when analog control in the laser oscillator is enabled. When the Laser Enable/Disable state[a] is Enable, it waits for a certain period of time for this signal to turn ON with the PRE_LS (LS[b]) instruction. When Laser Enable/Disable state is Disable, this signal will not be checked.
Laser Ready	Assign a signal that is turned on when the laser oscillator is ready to output the laser. When Laser Enable/Disable state[a] is Enable, it keeps waiting for this signal to turn on with PRE_LS (LS[b]) instruction during “Laser Ready DI latency” on YLS laser setup screen. When Laser Enable/Disable state is Disable, this signal will not be checked.
Emission is ON	Assign a signal that will be turned on when the laser is actually outputed. If this signal is turned off for a certain period of time during laser processing, an alarm will be occurred.
Beam Switch CH	This signal is an acknowledgement signal for “Beam Switch CH” output. When Laser Enable/Disable state[a] is Enable, PRE_LS (LS[b]) instruction continues to wait for a certain period of time until this signal matches “Beam Switch CH” output. When Laser Enable/Disable state is Disable, this signal will not be checked.

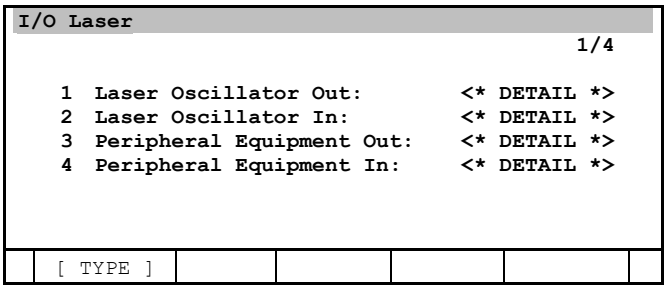
NOTE
a. For details, refer to the “LASER ENABLE/DISABLE” section.
b. When the LS instruction is executed without executing the PRE_LS instruction.

Procedure 70-5 Display YLS laser oscillator input screen

Step

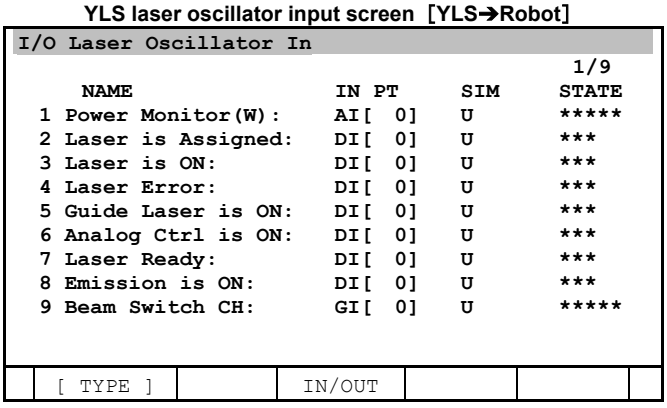
1 Press the [MENU] key then screen menu will be displayed.
2 Select [5 I/O].
3 Show Menu by pressed F1 [TYPE].

4 Select “Laser I/O”.



5 Move cursor on <* DETAIL *> of “Laser Oscillator In” and press [ENTER] key.

YLS laser oscillator input screen [YLS→Robot]



70.3.2 YLR Laser Oscillator I/O

By assigning YLR laser oscillator I/O, it becomes easy to control YLR from the robot.

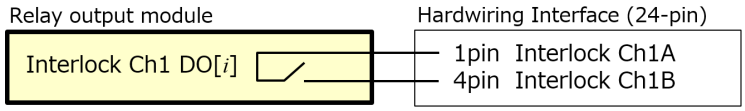
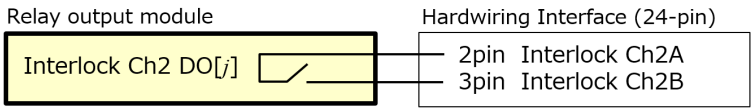
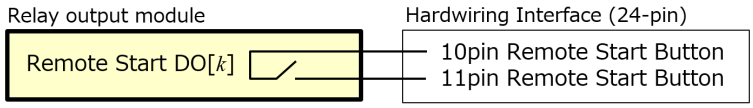
NOTE
- It is not necessary to assign all signals. Assign only the signals you need.
- Even if all signals are not assigned, if the Laser Enable/Disable[a] is Enable, the robot can output the laser command.

70.3.2.1 YLR laser oscillator output screen

When the model of the laser initial setup screen is specified as "YLR", the following items are displayed on the laser oscillator output screen.

Table 70.3.2.1 YLR laser oscillator output screen (Robot→YLR)

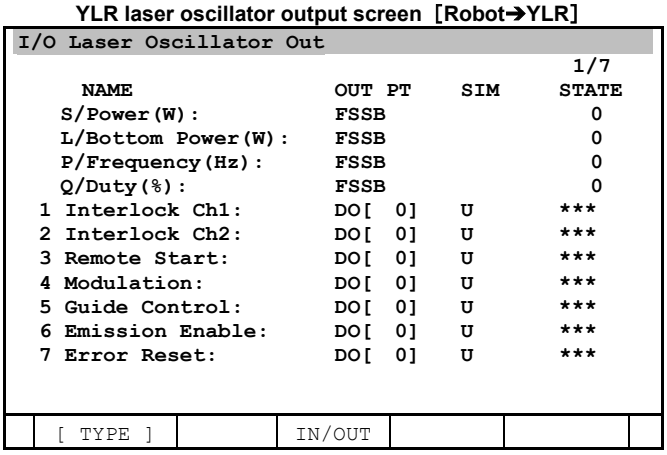
Items	Descriptions
S/Power(W)	It is common with YLS. See the section "YLS laser oscillator output screen".
L/Bottom Power(W)	It is common with YLS. See the section "YLS laser oscillator output screen".
P/Frequency(Hz)	It is common with YLS. See the section "YLS laser oscillator output screen".
Q/Duty(%)	It is common with YLS. See the section "YLS laser oscillator output screen".

Items	Descriptions
Interlock Ch1	This is a signal to control Interlock Ch1A and Interlock Ch1B in YLR. Assign this signal to a relay output module (AOR16G, etc.) for use.  When Laser Enable/Disable state[a] is Enable, this signal is turned on by the PRE_LS (LS[b]) instruction. If Laser Enable/Disable state is Disable, this signal will not be turned on by the PRE_LS (LS) instruction. If Laser Enable/Disable state is Disable, this signal will be forcibly turned off.
Interlock Ch2	This is a signal to control Interlock Ch2A and Interlock Ch2B in YLR. Assign this signal to a relay output module (AOR16G, etc.) for use.  When Laser Enable/Disable state[a] is Enable, this signal is turned on by the PRE_LS (LS[b]) instruction. If Laser Enable/Disable state is Disable, this signal will not be turned on by the PRE_LS (LS) instruction. If Laser Enable/Disable state is Disable, this signal will be forcibly turned off.
Remote Start	This is a signal to control Remote Start Button in YLR. Assign this signal to a relay output module (AOR16G, etc.) for use.  When Laser Enable/Disable state[a] is Enable, this signal is turned on by the PRE_LS (LS[b]) instruction, and turned off when the “P_S_Active” input signal is turned on. If Laser Enable/Disable state is Disable, this signal will not be turned on by the PRE_LS (LS) instruction. If Laser Enable/Disable state is Disable, this signal will be forcibly turned off.
Modulation	This is a signal to control Modulation in YLR. When Laser Enable/Disable state[a] is Enable, this signal is turned on by PRE_LS (LS[b]) instruction and turned off by LE instruction. If Laser Enable/Disable state is Disable, this signal will not be turned on by PRE_LS (LS) instruction. **NOTE:** This signal does not become a pulse even if a pulse wave (for example, Q = 50%) is specified by the LS instruction. If you want to assign a pulse signal to Modulation in YLR, do not use this signal, but use the laser ON/OFF signal output from the laser interface unit. For more information on the laser ON/OFF signale, see the “LASER INTERFACE UNIT” section at the end of this chapter.
Guide Control	This is a signal to control Guide Control in YLR (This signal is used to control the turning on/off of the guide laser.). When this signal is assigned, the “Guide laser auto OFF timer” on YLR laser setup screen will function. If Laser Enable/Disable state[a] is Enable or “Power On” input is off, this signal is forcibly turned off.
Emission Enable	This is a signal to control Emission Enable in YLR (a signal to allow the laser output to the laser oscillator.). When Laser Enable/Disable state[a] is Enable, this signal is turned on by PRE_LS (LS[b]) instruction and turned off by LE instruction. If Laser Enable/Disable state is Disable, this signal will not be turned on by PRE_LS (LS) instruction.

Items	Descriptions
Error Reset	This is a signal to control Error Reset in YLR (This signal resets the error of the laser oscillator.). A pulse is output by the reset key input of the teach pendant or the input of the dedicated external signal FAULT_RESET UI[5].

NOTE
a. For details, refer to the “LASER ENABLE/DISABLE” section.
b. When the LS instruction is executed without executing the PRE_LS instruction.

YLR laser oscillator output screen [Robot➔YLR]



The procedure for displaying the YLR laser oscillator output screen is the same as for YLS. See the section "YLS laser oscillator output screen".

70.3.2.2 YLR laser oscillator input screen

When the model of the laser initial setup screen is specified as "YLR", the following items are displayed on the laser oscillator input screen.

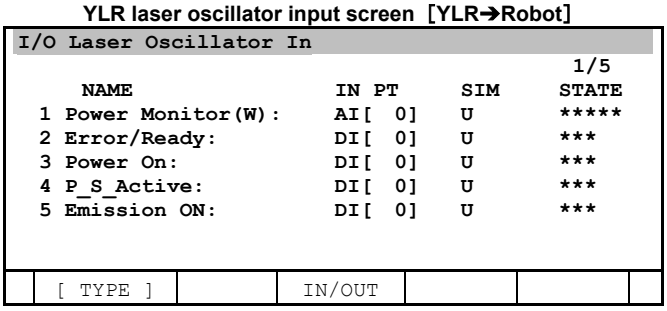
Table 70.3.2.2 YLR laser oscillator input screen (YLR➔Robot)

Items	Descriptions
Power Monitor(W)	Assign the signal for the power monitor to here. Input type can be selected from AI or GI. When AI is selected as the input type, the value obtained by converting the analog input voltage (V) to power (W) is displayed. (0-4V = 0W-Maximum power) If GI is selected as the input type, the value of the group input signal will be displayed as is.
Error/Ready	This is a signal to check the Error/Ready status in YLR. If you assign this signal, the robot will always check the status of this signal. When this signal is turned off during laser processing, an alarm is occurred and laser processing is stopped. In other cases, a warning is occurred when this signal turns off.
Power On	This is a signal to check the Power On status in YLR. When Laser Enable/Disable state[a] is Enable, PRE_LS (LS[b]) instruction checks if this signal is on. When Laser Enable/Disable state is Disable, this signal will not be checked. If this signal is off, “Guide Control” output is forcibly turned off.
P_S_Active	This is a signal to check the Power Supply Active status in YLR. When Laser Enable/Disable state[a] is Enable, it keeps waiting for this signal to turn on with PRE_LS (LS[b]) instruction during “P_S_Active DI latency” on YLR laser setup screen. When Laser Enable/Disable state is Disable, this signal will not be checked.

Items	Descriptions
Emission ON	This is a signal to check the Emission ON status in YLR. When Laser Enable/Disable state[a] is Enable, PRE_LS (LS[b]) instruction checks if this signal is on. When Laser Enable/Disable state is Disable, this signal will not be checked.

NOTE
a. For details, refer to the “LASER ENABLE/DISABLE” section.
b. When the LS instruction is executed without executing the PRE_LS instruction.

YLR laser oscillator input screen [YLR→Robot]



The procedure for displaying the YLR laser oscillator input screen is the same as for YLS. See the section "YLS laser oscillator input screen".

70.3.3 Peripheral Equipment I/O

By assigning peripheral equipment I/O, it becomes easy to control peripheral equipment from the robot or control the robot from the peripheral equipment.

NOTE
- It is not necessary to assign all signals. Assign only the signals you need.
- Even if all signals are not assigned, if the Laser Enable/Disable[a] is Enable, the robot can output the laser command.

70.3.3.1 Peripheral equipment output screen

Table 70.3.3.1 Peripheral equipment output screen (Robot→Peripheral equipment)

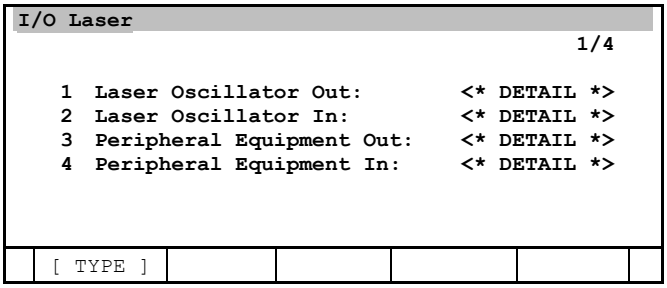
Items	Descriptions
Laser ENBL/DSBL	This signal is used to notify the peripheral device of the robot's Laser Enable/Disable state[a]. When this signal is on, the robot is ready to output laser commands. Use this signal to control warning lights etc. Output type can be selected from DO or RO.
Air/Gas	This signal commands the air/gas supply unit to turn on/off the air/gas during laser processing. Output type can be selected from DO or RO. This signal is output only when Laser Enable/Disable state[a] is Enable.
Wire Feed	This signal commands the wire feeder to turn the wire feed ON/OFF during laser processing when the wire control instruction is executed. Output type can be selected from DO or RO. This signal is output only when Laser Enable/Disable state[a] is Enable.
Wire Feed Speed	When the wire control instruction is executed, this signal outputs the argument Sp of the wire control instruction during laser processing. Output type can be selected from GO or AO. This signal is output only when Laser Enable/Disable state[a] is Enable.

NOTE
a. For details, refer to the “LASER ENABLE/DISABLE” section.

Procedure 70-6 Display peripheral equipment output screen

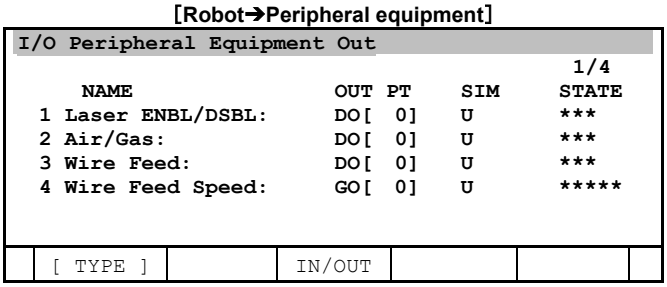
Step

1 Press the [MENU] key then screen menu will be displayed.
2 Select [5 I/O].
3 Show Menu by pressed F1 [TYPE].
4 Select “Laser I/O”.



5 Move cursor on <* DETAIL *> of “Peripheral equipment Out” and press [ENTER] key.

Peripheral equipment out screen
[Robot➔Peripheral equipment]



70.3.3.2 Peripheral equipment input screen

Table 70.3.3.2 Peripheral equipment screen (Peripheral equipment➔Robot)

Items	Descriptions
Laser ENBL/DSBL	This signal is used to control the Laser Enable/Disable request[a] from peripheral equipment. Input type can be selected from DI or RI.
Guide Laser	This signal is used to control on/off of guide laser from peripheral equipment. However, guide laser can be controlled from this signal only when the teach pendant is disabled and the Laser Enable/Disable statea is Disable. When guide laser is turned off by “Guide laser auto OFF timer”, guide laser can be turned on again by turning this signal off and on once. Input type can be selected from DI or RI.
Air/Gas	Use this signal when you want to check if air/gas is being supplied. If air/gas is being supplied, the air/gas supply unit must input on to this signal. If this signal is on when LS instruction is executed, laser processing is started. Input type can be selected from DI or RI. When Laser Enable/Disable state[a] is Disable, this signal will not be checked.

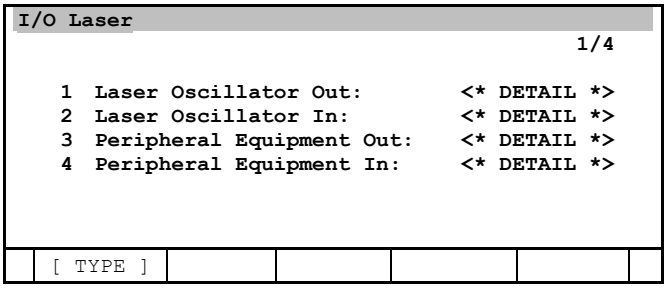
Items	Descriptions
Wire Feed	Use this signal when you want to check if wire is being fed. If the wire is being fed, the wire feeder must input on to this signal. If this signal is on during laser processing, laser processing is continued. Input type can be selected from DI or RI. When Laser Enable/Disable state[a] is Disable, this signal will not be checked.

NOTE
a. For details, refer to the “LASER ENABLE/DISABLE” section.

Procedure 70-7 Display peripheral equipment input screen

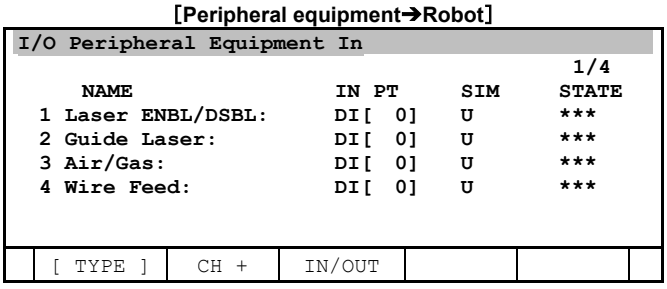
Step

1 Press the [MENU] key then screen menu will be displayed.
2 Select [5 I/O].
3 Show Menu by pressed F1 [TYPE].
4 Select “Laser I/O”.



5 Move cursor on <* DETAIL *> of “Peripheral equipment In” and press [ENTER] key.

Peripheral equipment in screen
[Peripheral equipment➔Robot]



70.4 LASER ENABLE/DISABLE

There are Laser Enable/Disable request and Laser Enable/Disable state in Laser Enable/Disable.
Laser Enable/Disable state is switched according to the Laser Enable/Disable request.
If the LS instruction is executed while the Laser Enable/Disable state is Enable, the laser command is output.
Even if the LS instruction is executed when the Laser Enable/Disable state is Disable, the laser command is not output.

70.4.1 Laser Enalbe/Disable Screen

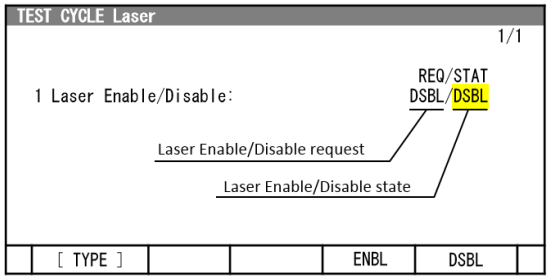
Table 70.4.1 Laser Enable/Disable screen

Items	Descriptions
Laser Enable/Disable request	This item allows you to switch the Laser Enable/Disable state. If “Laser ENBL/DSBL” input[a] is not assigned, this item can be operated manually. This item cannot be operated manually if “Laser ENBL/DSBL” input is assigned. Operated according to the “Laser ENBL/DSBL” input. **However, even if this item is ENBL (Enable), Laser Enable/Disable state will be forcibly DSBL (Disable) while teach pendant is enabled or while servo power is turned off due to an emergency stop etc.**
Laser Enable/Disable state	This item can be switched by operating Laser Enable/Disable request. (This item cannot be operated directly.) **However, even if Laser Enable/Disable request is ENBL (Enable), this item will be forcibly DSBL (Disable) while teach pendant is enabled or while servo power is turned off due to an emergency stop etc.** If DSBL (Disable) is displayed here, ・Laser command is NOT output even if LS instruction is executed. For YLS ・”Guide Laser ON” output[b] can be turned on. ・”Laser ON” output[b] is forcibly turned off. For YLR ・”Guide Control” output[c] can be turned on. ・”Interlock Ch1” and “Interlock Ch2” outputs[c] are forcibly turned off. ・”Remote Start” output[c] is forcibly turned off. If ENBL (Enable) is displayed here, ・When LS instruction is executed, laser command is output. For YLS ・”Guide Laser ON” output[b] is forcibly turned off. ・”Laser ON” output[b] can be turned on. For YLR ・”Guide Control” output[c] is forcibly turned off. ・”Interlock Ch1” and “Interlock Ch2” outputs[c] can be turned on. ・”Remote Start” output[c] can be turned on.

NOTE
a. This is Peripheral equipment I/O. For details, refer to the “Peripheral Equipment I/O” section.
b. This is YLS laser oscillator I/O. See the section “YLS Laser Oscillator I/O”.
c. This is YLR laser oscillator I/O. See the section “YLR Laser Oscillator I/O”.

Procedure 70-8 Display Laser Enalbe/Disable Screen

Step
1 Press the [MENU] key then screen menu will be displayed.
2 Select [2 TEST CYCLE].
3 Show Menu by pressed F1 [TYPE].
4 Select “Laser”.



70.5 LASER INSTRUCTION

Laser instructions are for instructing the robot when and how to perform laser processing.
There are four laser instructions:

- PRE_LS instruction: Preparation instruction for LS instruction
- LS instruction: Laser start instruction of the type that specifies processing schedules (S, L, P, Q, V)
- LS_SCH instruction: Laser start instruction of the type that specifies processing schedule number
- LE instruction: Laser end instruction

70.5.1 PRE_LS Instruction

- This is a preparation instruction for executing the LS instruction. Teach the PRE_LS instruction before the LS instruction.
- When "Beam switch CH switching" is ENABLE, the argument CH is added to PRE_LS instruction. ENABLE/DISABLE of " Beam switch CH switching" can be changed on the laser initial setup screen.

PRE_LS or PRE_LS[CH=*]
– CH : Beam switch channel number

Fig. 70.5.1(a) PRE_LS instruction

YLS

- When PRE_LS[CH=*] instruction is executed, the value of the argument CH is output from “Beam Switch CH” output[b]. In addition, the "Beam switch channel number" on YLS laser setup screen is overwritten to the value of the argument CH.

When Laser Enable/Disable state[a] is Enable

YLS

- When “Time-sharing” is ENABLE, “Laser Request” output[b] is turned on. When “Time-sharing” is DESABLE, “Laser Request” output is on from the beginning. ENABLE/DISABLE of “Time-sharing” can be changed on the laser initial setup screen.
- Wait for “Laser is Assigned” input[b] to turn on. An alarm will occur if “Laser is Assigned” input does not turn on after “Laser assigned DI latency” has passed. “Laser assigned DI latency” can be changed on YLS laser setup screen.
- “Laser ON” output[b] is turned on.
- Wait for “Laser is ON” input[b] to turn on. An alarm will occur if “Laser is ON” input does not turn on after “Laser ON DI latency” has passed. “Laser ON DI latency” can be changed on YLS laser setup screen.
- “Analog Control ON” output[b] is turned on.
- Wait for “Analog Ctrl is ON” input[b] to turn on. If “Analog Ctrl is ON” input does not turn on after a certain period of time, an alarm will occur.

- Wait for “Beam Switch CH” input[b] and “Beam Switch CH” output[b] to match. If “Beam Switch CH” input and “Beam Switch CH” output does not match after a certain period of time, an alarm will occur.
- Wait for “Laser Ready” input[b] to turn on. An alarm will occur if “Laser Ready” input does not turn on after “Laser ON DI latency” has passed. “Laser ON DI latency” can be changed on YLS laser setup screen.
- “Program Start” output[b] is turned on.

YLR

- Wait for “Power On” input[c] to turn on. An alarm will occur if “Power On” input does not turn on after “Power On DI latency” has passed. “Power On DI latency” can be changed on YLR laser setup screen.
- ”Interlock Ch1” and “Interlock Ch2” outputs[c] are turned on.
- “Remote Start” output[c] is turned on.
- Wait for “P_S_Active” input[c] to turn on. An alarm will occur if “P_S_Active” input does not turn on after “P_S_Active DI latency” has passed. “P_S_Active DI latency” can be changed on YLR laser setup screen.
- When the “P_S_Active” input[c] turns on, “Remote Start” output[c] is turned off.
- “Emission Enable” output[c] is turned on.
- “Modulation” output[c] is turned on.

COMMON

- “Air/Gas” output[d] is turned on. By turning on “Air/Gas” output signal before executing the LS instruction, Air/Gas is surely being output when the LS instruction is executed.

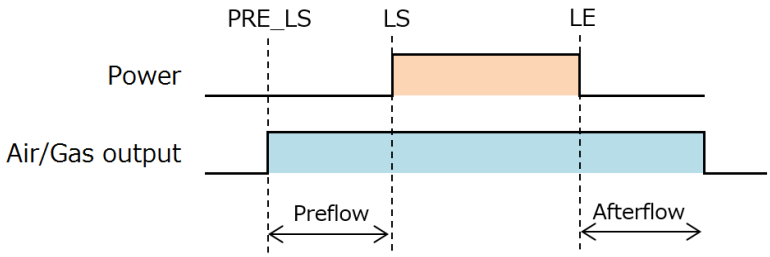


Fig. 70.5.1(b) Preflow by PRE_LS instruction

When Laser Enable/Disable state[a] is Disable

YLS

- Does not check whether “Laser is Assigned” input[b] is on.
- Does not turn on “Laser ON” output[b].
- Does not check whether “Laser is ON” input[b] is on.
- Does not turn on “Analog Control ON” output[b].
- Does not check whether “Analog Ctrl is ON” input[b] is on.
- Does not check the status of “Beam Switch CH”input[b].
- Does not check whether “Laser Ready” input[b] is on.
- Does not turn on “Program Start” output[b].

YLR

- Does not check whether “Power On” input[c] is on.
- Does not turn on ”Interlock Ch1” and “Interlock Ch2” outputs[c].
- Does not turn on “Remote Start” output[c].
- Does not check whether “P_S_Active” input[c] is on.
- Does not turn on “Emission Enable” output[c].
- Does not turn on “Modulation” output[c].

COMMON

- Does not turn on “Air/Gas” output[d].

When Process disable for restart is Enable

YLS

- Does not turn on “Laser ON” output[b].
- Does not check whether “Laser is ON” input[b] is on.
- Does not turn on “Analog Control ON” output[b].
- Does not check whether “Analog Ctrl is ON” input[b] is on.
- Does not check whether “Laser Ready” input[b] is on.
- Does not turn on “Program Start” output[b].

YLR

- Does not check whether “Power On” input[c] is on.
- Does not turn on ”Interlock Ch1” and “Interlock Ch2” outputs[c].
- Does not turn on “Remote Start” output[c].
- Does not check whether “P_S_Active” input[c] is on.
- Does not turn on “Emission Enable” output[c].
- Does not turn on “Modulation” output[c].

COMMON

- Does not turn on “Air/Gas” output[d].

NOTE

a. For details, refer to the “LASER ENABLE/DISABLE” section.
b. This is YLS laser oscillator I/O. See the section “YLS Laser Oscillator I/O”.
c. This is YLR laser oscillator I/O. See the section “YLR Laser Oscillator I/O”.
d. This is Peripheral equipment I/O. For details, refer to the “Peripheral Equipment I/O” section.

70.5.2 LS Instruction

- LS instruction is an instruction to start laser processing.

LS[S=*, L=*, P=*, Q=*, V=*]

- S : Power (W)
- L : Bottom power (W)
- P : Frequency (Hz)
- Q : Duty (%)
- V : Processing speed (mm/sec、cm/min、inch/min)

Fig. 70.5.2(a) LS instruction

- Meanings of processing schedules (S, L, P, Q) are shown below.

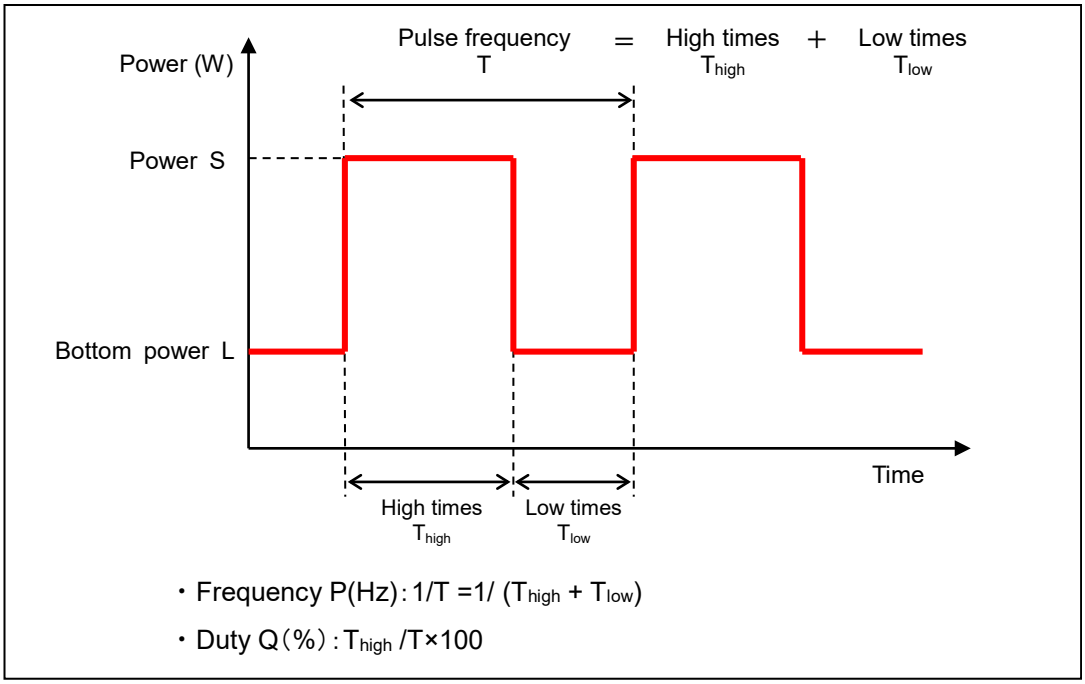


Fig. 70.5.2(b) Processing schedule of LS instruction

- Processing speed V is used for speed tracking functions such as power control function and wire control function. Based on the processing speed specified here, the power or wire feed speed is calculated according to the speed change.

NOTE
If you do not use the speed tracking function, you do not need to specify the processing speed V.

- The maximum value of processing scheduldes (S, L, P) and the unit of processing speed V can be changed on the laser initial setup screen.
- It is possible to switch the processing scheduldes(S, L, P, Q, V) during processing.

70.5.2.1 When there is LS before PRE_LS (recommended)

- If PRE_LS instruction precedes LS instruction and LS instruction is taught with the CNT motion, the robot will not stop.

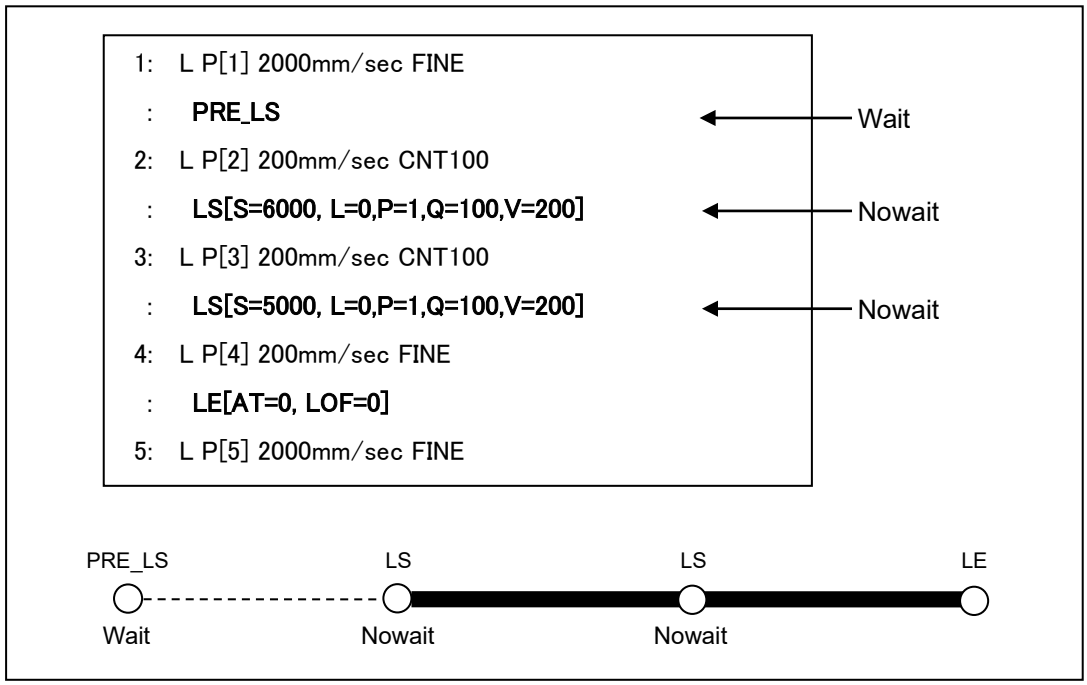


Fig. 70.5.1.2 When there is LS before PRE_LS

When Laser Enable/Disable state[a] is Enable

YLS

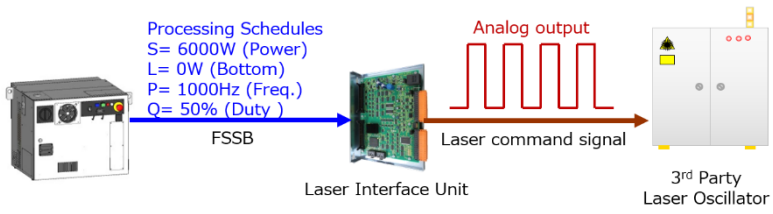
- It is confirmed whether "Emission is ON" input[b] is on. This confirmation is done periodically during laser processing.

YLR

- It is confirmed whether "Emission ON" input[c] is on. This confirmation is done periodically during laser processing.

COMMON

- Laser command according to the processing schedules (S, L, P, Q) is output.



- It is confirmed whether "Air/Gas" input[d] is on. This confirmation is done periodically during laser processing.
- "Wire Feed" output[d] is turned on.
- "Wire Feed Speed" output[d] is turned on.
- It is confirmed whether "Wire Feed" input[d] is on. An alarm will occur if "Wire Feed" input does not turn on after "Wire feed OK latency" has passed. "Wire feed OK latency" can be changed on the wire control setup screen. This confirmation is done periodically during laser processing.

When Laser Enable/Disable state[a] is Disable
When Process disable for restart is Enable

YLS

- Does not check whethe "Emission is ON" input[b] is on.

YLR

- Does not check whethe “Emission ON” input[c] is on.

COMMON

- Laser command is not output.
- Does not check whether “Air/Gas” input[d] is on.
- Does not turn on “Wire Feed” output[d].
- Does not turn on “Wire Feed Speed” output[d].
- Does not check whethe “Wire Feed” input[d] is on.

NOTE

a. For details, refer to the “LASER ENABLE/DISABLE” section.
b. This is YLS laser oscillator I/O. See the section “YLS Laser Oscillator I/O”.
c. This is YLR laser oscillator I/O. See the section “YLR Laser Oscillator I/O”.
d. This is Peripheral equipment I/O. For details, refer to the “Peripheral Equipment I/O” section.

70.5.2.2 When there is not LS before PRE_LS

- Laser processing can be performed without PRE_LS instruction, but some features cannot be used without PRE_LS instruction. It is recommended to add PRE_LS instruction.
- Robot performs FINE motion even if LS instruction is taught together with CNT motion.

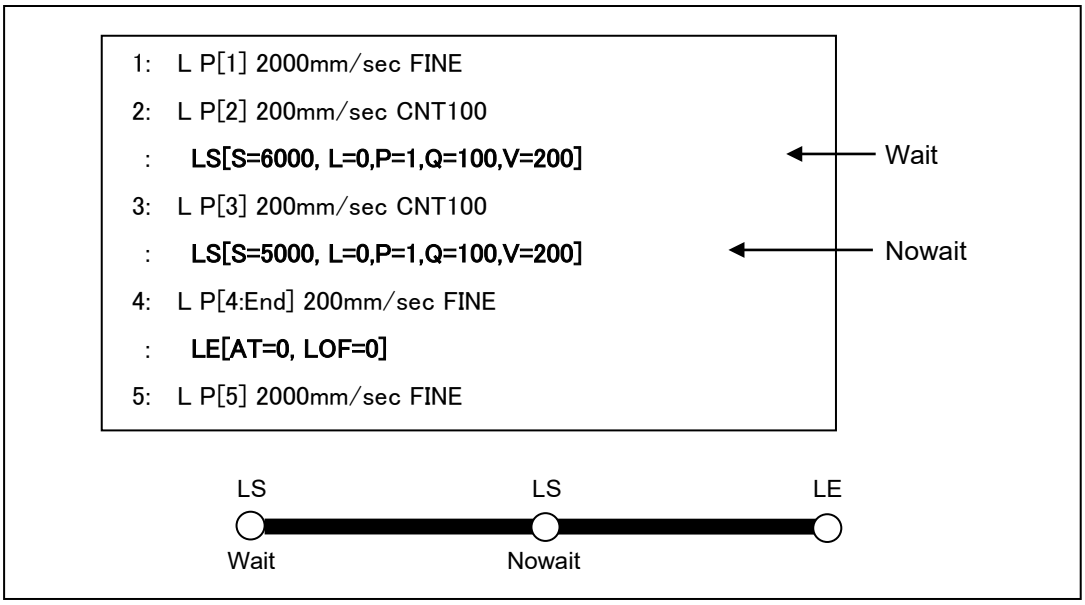


Fig. 70.5.2.2 When there is not LS before PRE_LS

When Laser Enable/Disable state[a] is Enable

YLS

- When “Time-sharing” is ENABLE, “Laser Request” output[b] is turned on. When “Time-sharing” is DESABLE, “Laser Request” output is on from the beginning. ENABLE/DISABLE of “Time-sharing” can be changed on the laser initial setup screen.
- Wait for “Laser is Assigned” input[b] to turn on. An alarm will occur if “Laser is Assigned” input does not turn on after “Laser assigned DI latency” has passed. “Laser assigned DI latency” can be changed on YLS laser setup screen.
- “Laser ON” output[b] is turned on.

- Wait for “Laser is ON” input[b] to turn on. An alarm will occur if “Laser is ON” input does not turn on after “Laser ON DI latency” has passed. “Laser ON DI latency” can be changed on YLS laser setup screen.
- “Analog Control ON” output[b] is turned on.
- Wait for “Analog Ctrl is ON” input[b] to turn on. If “Analog Ctrl is ON” input does not turn on after a certain period of time, an alarm will occur.
- Wait for “Beam Switch CH” input[b] and “Beam Switch CH” output[b] to match. If “Beam Switch CH” input and “Beam Switch CH” output does not match after a certain period of time, an alarm will occur.
- Wait for “Laser Ready” input[b] to turn on. An alarm will occur if “Laser Ready” input does not turn on after “Laser ON DI latency” has passed. “Laser ON DI latency” can be changed on YLS laser setup screen.
- “Program Start” output[b] is turned on.
- It is confirmed whether “Emission is ON” input[b] is on. This confirmation is done periodically during laser processing.

YLR

- Wait for “Power On” input[c] to turn on. An alarm will occur if “Power On” input does not turn on after “Power On DI latency” has passed. “Power On DI latency” can be changed on YLR laser setup screen.
- ”Interlock Ch1” and “Interlock Ch2” outputs[c] are turned on.
- “Remote Start” output[c] is turned on.
- Wait for “P_S_Active” input[c] to turn on. An alarm will occur if “P_S_Active” input does not turn on after “P_S_Active DI latency” has passed. “P_S_Active DI latency” can be changed on YLR laser setup screen.
- When the “P_S_Active” input[c] turns on, “Remote Start” output[c] is turned off.
- “Emission Enable” output[c] is turned on.
- “Modulation” output[c] is turned on.
- It is confirmed whether “Emission ON” input[c] is on. This confirmation is done periodically during laser processing.

COMMON

- “Air/Gas” output[d] is turned on.
- It is confirmed whether “Air/Gas” input[d] is on. An alarm will occur if “Air/Gas” input does not turn on after “Air/Gas OK latency” has passed. “Air/Gas OK latency” can be changed on the laser setup screen. This confirmation is done periodically during laser processing.
- “Wire Feed” output[d] is turned on.
- “Wire Feed Speed” output[d] is turned on.
- It is confirmed whether “Wire Feed” input[d] is on. An alarm will occur if “Wire Feed” input does not turn on after “Wire feed OK latency” has passed. “Wire feed OK latency” can be changed on the wire control setup screen. This confirmation is done periodically during laser processing.
- Laser command according to the processing schedules (S, L, P, Q) is output.

When Laser Enable/Disable state[a] is Disable

YLS

- Does not check whether “Laser is Assigned” input[b] is on.
- Does not turn on “Laser ON” output[b].
- Does not check whether “Laser is ON” input[b] is on.
- Does not turn on “Analog Control ON” output[b].
- Does not check whether “Analog Ctrl is ON” input[b] is on.
- Does not check the status of “Beam Switch CH”input[b].
- Does not check whether “Laser Ready” input[b] is on.
- Does not turn on “Program Start” output[b].

- Does not check whethe “Emission is ON” input[b] is on.

YLR

- Does not check whether “Power On” input[c] is on.
- Does not turn on ”Interlock Ch1” and “Interlock Ch2” outputs[c].
- Does not turn on “Remote Start” output[c].
- Does not check whether “P_S_Active” input[c] is on.
- Does not turn on “Emission Enable” output[c].
- Does not turn on “Modulation” output[c].
- Does not check whether “Emission ON” input[c] is on.

COMMON

- Does not turn on “Air/Gas” output[d].
- Does not check whether “Air/Gas” input[d] is on.
- Does not turn on “Wire Feed” output[d].
- Does not turn on “Wire Feed Speed” output[d].
- Does not check whethe “Wire Feed” input[d] is on.
- Laser command is not output[d].

When Process disable for restart is Enable

YLS

- Does not turn on “Laser ON” output[b].
- Does not check whether “Laser is ON” input[b] is on.
- Does not turn on “Analog Control ON” output[b].
- Does not check whether “Analog Ctrl is ON” input[b] is on.
- Does not check whether “Laser Ready” input[b] is on.
- Does not turn on “Program Start” output[b].
- Does not check whethe “Emission is ON” input[b] is on.

YLR

- Does not check whether “Power On” input[c] is on.
- Does not turn on ”Interlock Ch1” and “Interlock Ch2” outputs[c].
- Does not turn on “Remote Start” output[c].
- Does not check whether “P_S_Active” input[c] is on.
- Does not turn on “Emission Enable” output[c].
- Does not turn on “Modulation” output[c].
- Does not check whether “Emission ON” input[c] is on.

COMMON

- Does not turn on “Air/Gas” output[d].
- Does not check whether “Air/Gas” input[d] is on.
- Does not turn on “Wire Feed” output[d].
- Does not turn on “Wire Feed Speed” output[d].
- Does not check whethe “Wire Feed” input[d] is on.
- Laser command is not output[d].

NOTE
a. For details, refer to the “LASER ENABLE/DISABLE” section.
b. This is YLS laser oscillator I/O. See the section “YLS Laser Oscillator I/O”.
c. This is YLR laser oscillator I/O. See the section “YLR Laser Oscillator I/O”.
d. This is Peripheral equipment I/O. For details, refer to the “Peripheral Equipment I/O” section.

70.5.3 LS_SCH Instruction

- LS_SCH instruction is an instruction to start laser processing like the LS instruction.
- Specify the processing scheudle number in the argument SN of the LS_SCH instruction. Based on the specified processing schedule number, laser processing will start according to the processing schedules (S, L, P, Q) on the processing schedule list screen.

LS_SCH[SN=*]
– SN: Processing schedule number (0-30)

Fig. 70.5.3(a) LS_SCH instruction

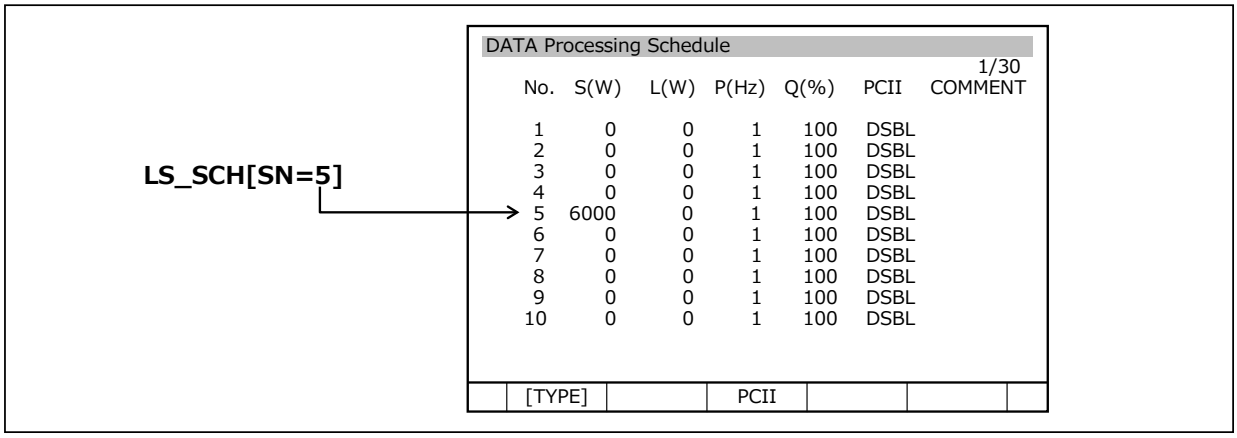
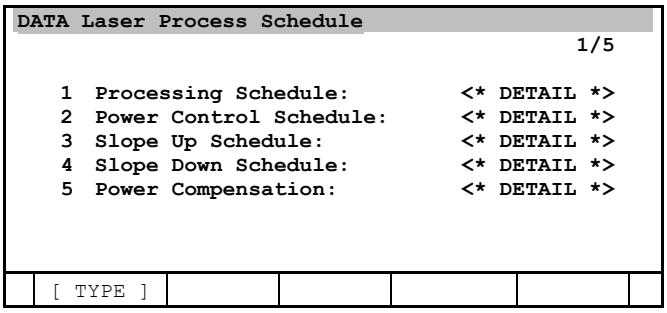


Fig. 70.5.3(b) LS_SCH instruction

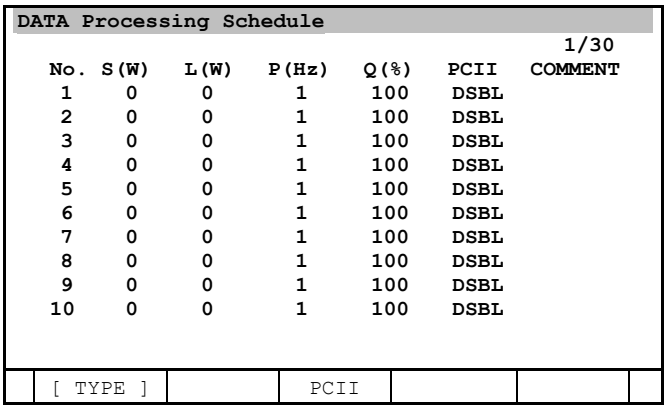
- 30 processing schedules can be prepared.
- If 0 is specified for the processing schedule number SN, no laser command is output.
- Usage of LS_SCH instruction is the same as that of LS instruction. See section on LS instruction for how to use the LS_SCH instruction.
- Processing schedule list screen can be displayed by the following procedure.

Procedure 70-9 Display processing schedule list screen

Step
1 Press the [MENU] key and display the screen menu.
2 Press [0. -- NEXT --] and select [3. DATA].
3 Press F1[Type] and display the screen menu.
4 Select “Laser Schedule”.



5 Move the cursor to <*DETAIL*> of Processing Schedule, and press [ENTER] key.



70.5.4 LE Instruction

- LE instruction is an instruction to end laser processing.

LE[AT=*, LOF=*]

- AT: Afterflow time (0 – 9999msec)
- LOF: Laser ON output is turned OFF (0 or 1)

Fig. 70.5.4(a) LE instruction

YLS

- "Program Start" output[a] is turned off.
- "Analog Control ON" output[a] is turned off.
- When LOF is 0, "Laser ON" output[a] is not turned off. When LOF is 1, "Laser ON" output is turned off.
- If "Time-sharing"is ENABLE and Teach Pendant is disabled, "Laser Request" output[a] is turned off. Otherwise, "Laser Request" output will not be turned off. ENABLE/DISABLE of "Time-sharing" can be changed on the laser initial setup screen.

YLR

- "Emission Enable" output[b] is turned off.
- "Modulation" output[b] is turned off.
- When LOF is 0, "Interlock Ch1" and "Interlock Ch2" outputs[b] are not turned off. When LOF is 1, "Interlock Ch1" and "Interlock Ch2" outputs are turned off.

COMMON

- Stops the output of laser command.

- When the afterflow time (AT =) is 0, “Air/Gas” output[c] is turned off. If the afterflow time (AT =) is set, “Air/Gas” output is turned off after the afterflow time has elapsed. If the next PRE_LS or LS instruction is executed during the afterflow, the afterflow will be cancelled.

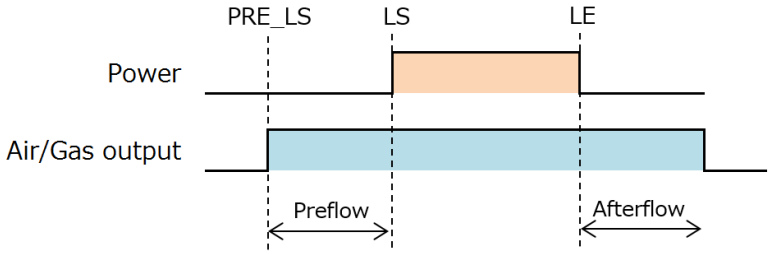


Fig. 70.5.4(b) Afterflow by LE instruction

- “Wire Feed” output[c] and “Wire Feed Speed” output[c] are turned off.
- If LE instruction is taught with the CNT motion, robot will not stop.

NOTE
a. This is YLS laser oscillator I/O. See the section “YLS Laser Oscillator I/O”.
b. This is YLR laser oscillator I/O. See the section “YLR Laser Oscillator I/O”.
c. This is Peripheral equipment I/O. For details, refer to the “Peripheral Equipment I/O” section.

70.5.5 Teaching laser instruction

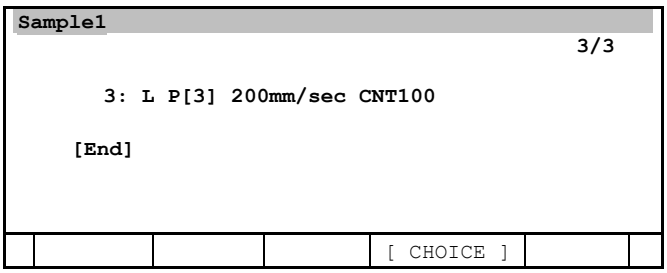
Laser instructions have PRE_LS, LS, LS_SCH and LE instruction. The teaching procedure is the same. Therefore, only the LS instruction is explained.

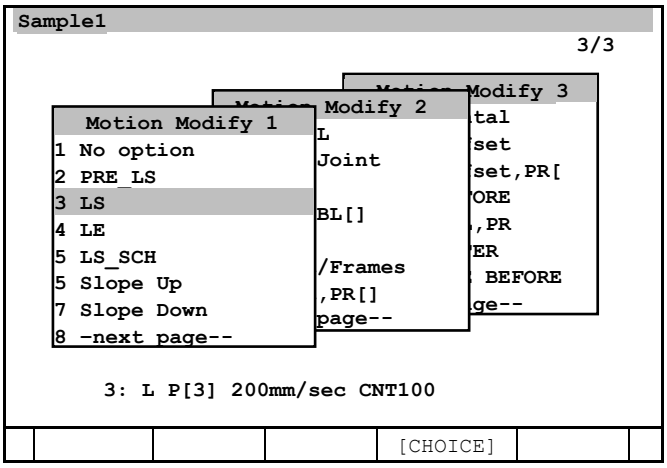
NOTE
Additional motion instruction and single instruction are provided for laser instruction. Normally, additional motion instruction is used.

Procedure 70-10 Teaching laser instruction

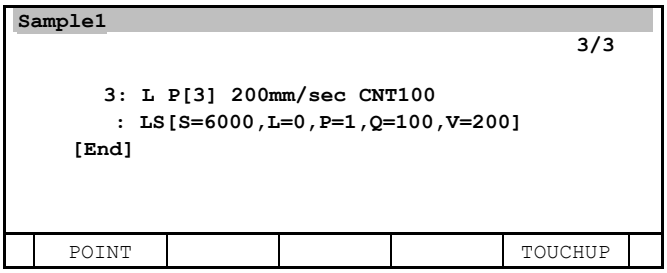
Step

1 Place the cursor behind the motion instruction and press F4 [CHOICE] to display the additional motion instruction menu.





2 Select “LS” from the menu.



70.6 TIMING CHART

Timing chart from laser processing to the end is shown below.

NOTE
The contents of timing chart differ depending on the model settings on the laser initial setup screen.

NOTE
- It is not necessary to assign all signals. Assign only the signals you need.
- Even if all signals are not assigned, if the Laser Enable/Disable[a] is Enable, the robot can output the laser command (processing schedules).

70.6.1 YLS Timing Chart

70.6.1.1 From laser start to end when is Time-sharing is DISABLE

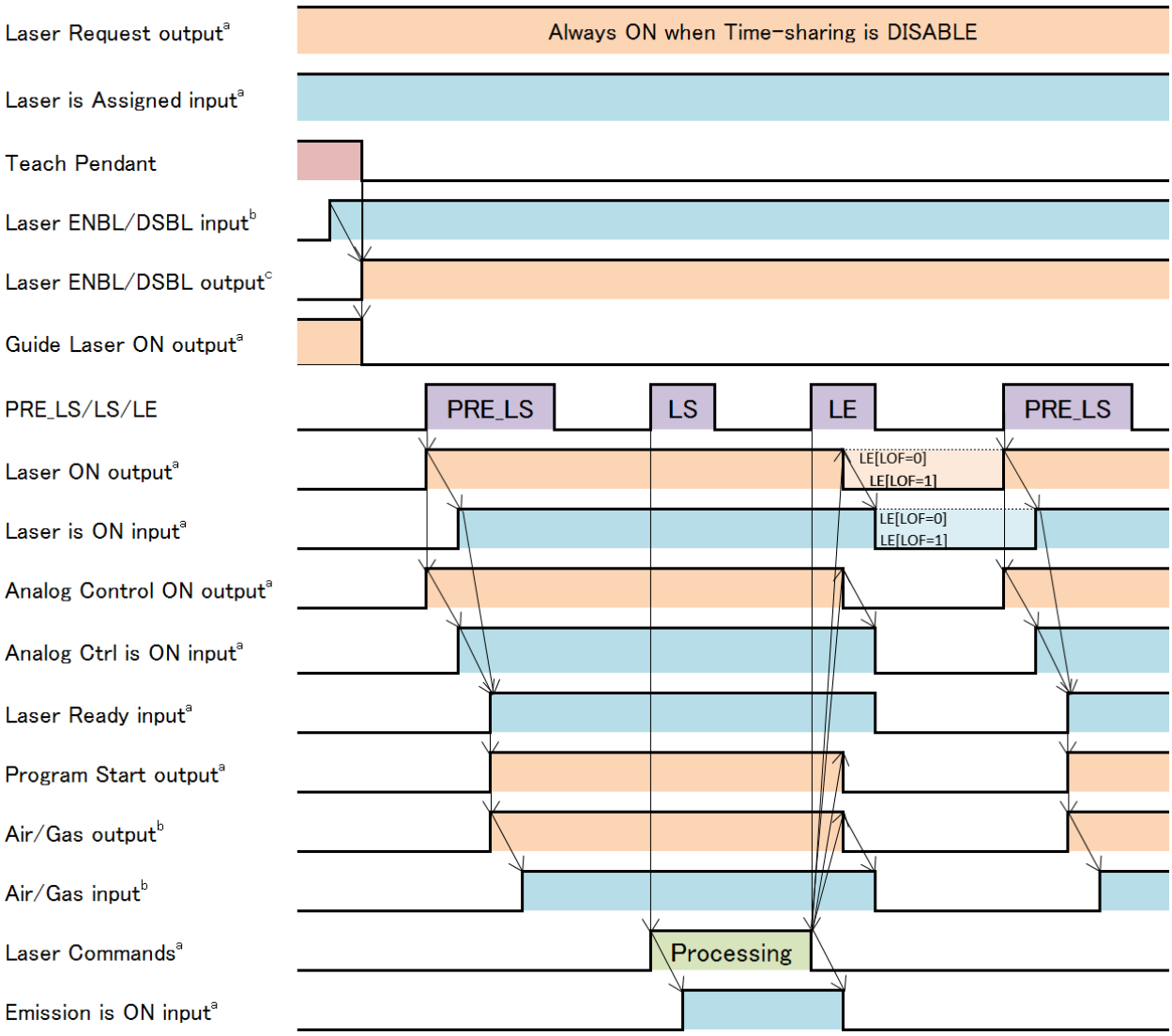


Fig. 70.6.1.1 From laser start to end when is Time-sharing is DISABLE

NOTE
a. This is YLS laser oscillator I/O. See the section “YLS Laser Oscillator I/O”.
b. This is Peripheral equipment I/O. For details, refer to the “Peripheral Equipment I/O” section.
c. This signal is on both Laser oscillator I/O and Peripheral equipment I/O.

70.6.1.2 When laser error or robot E-Stop occurs

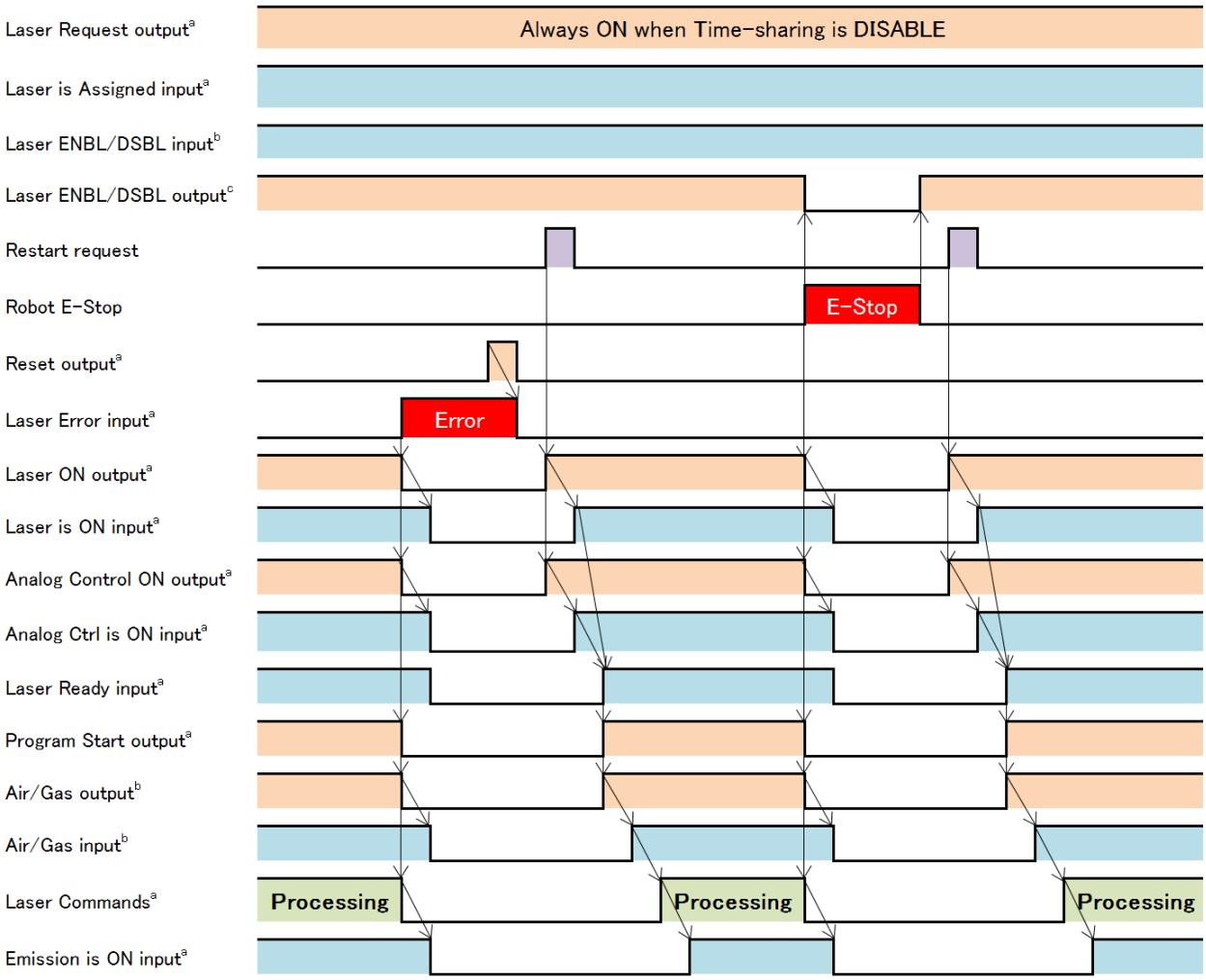


Fig. 70.6.1.2 When laser error or robot E-Stop occurs

NOTE
a. This is YLS laser oscillator I/O. See the section “YLS Laser Oscillator I/O”.
b. This is Peripheral equipment I/O. For details, refer to the “Peripheral Equipment I/O” section.
c. This signal is on both Laser oscillator I/O and Peripheral equipment I/O.

70.6.1.3 From laser start to end when Time-sharing is ENABLE

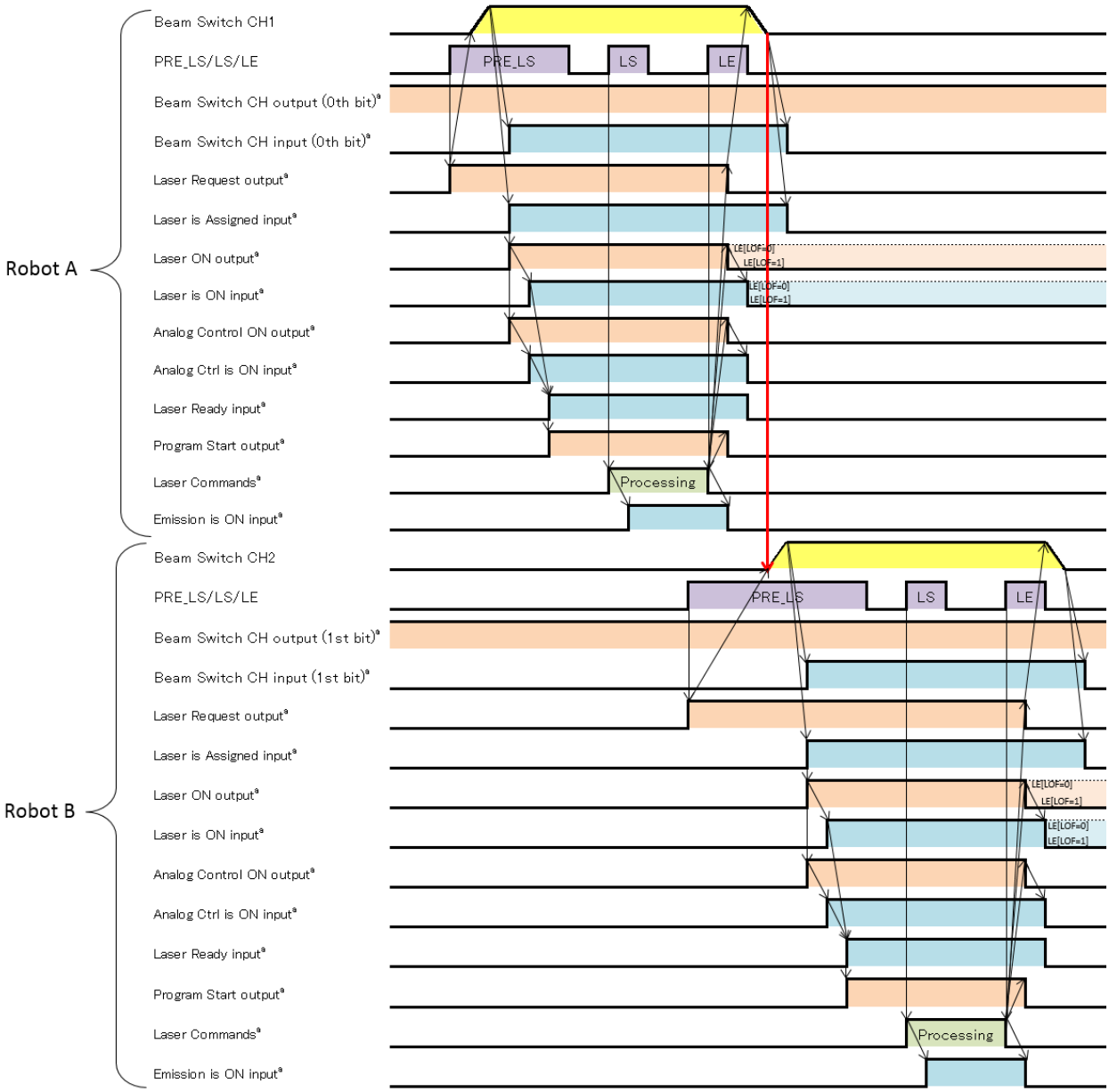


Fig. 70.6.1.3 From laser start to end when Time-sharing is ENABLE

NOTE
a. This is YLS laser oscillator I/O. See the section “YLS Laser Oscillator I/O”.

70.6.2 YLR Timing Chart

70.6.2.1 From laser start to end

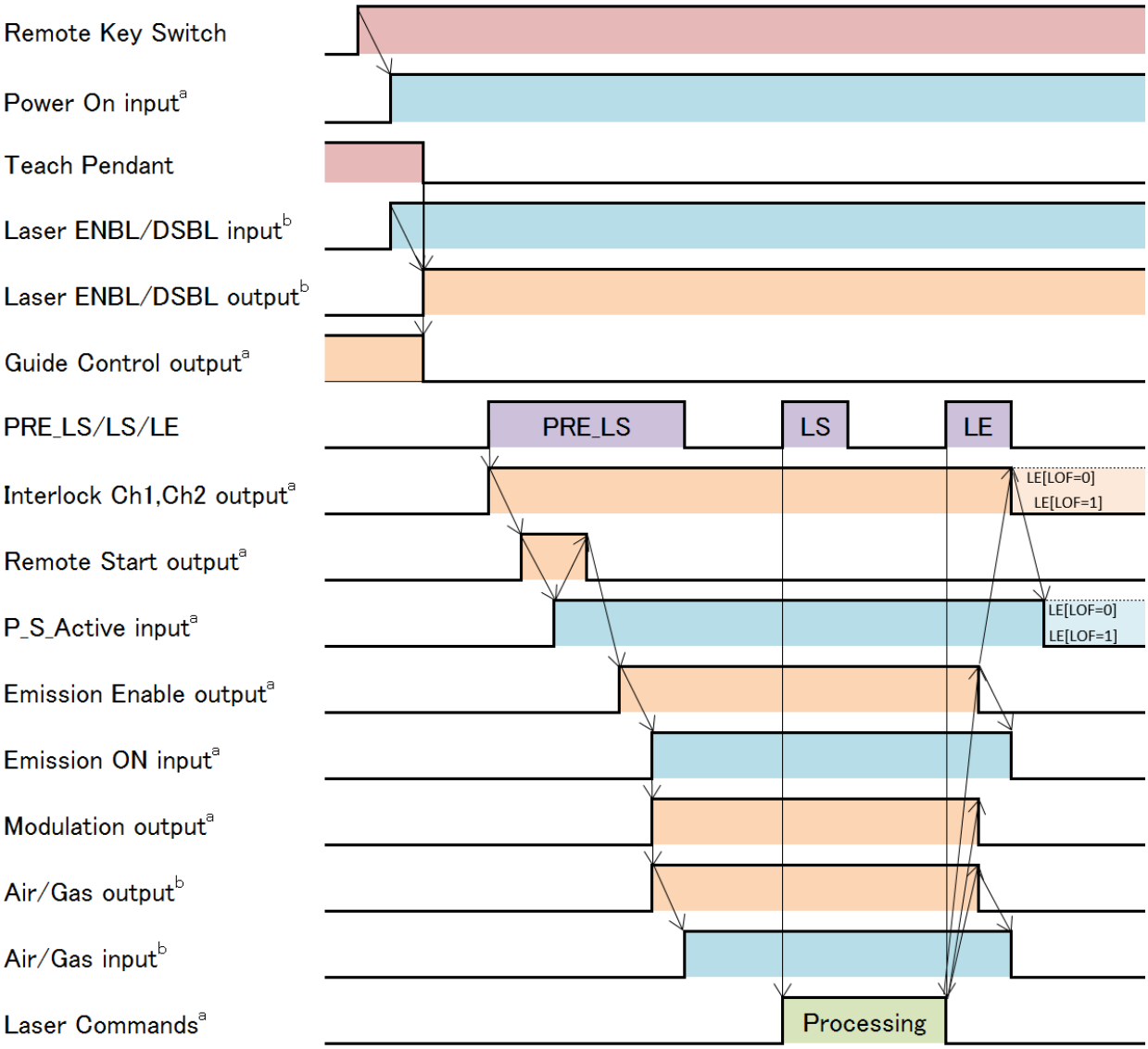


Fig. 70.6.2.1 From laser start to end

NOTE
a. This is YLR laser oscillator I/O. See the section “YLR Laser Oscillator I/O”.
b. This is Peripheral equipment I/O. For details, refer to the “Peripheral Equipment I/O” section.

70.6.2.2 When laser error or robot E-Stop occurs

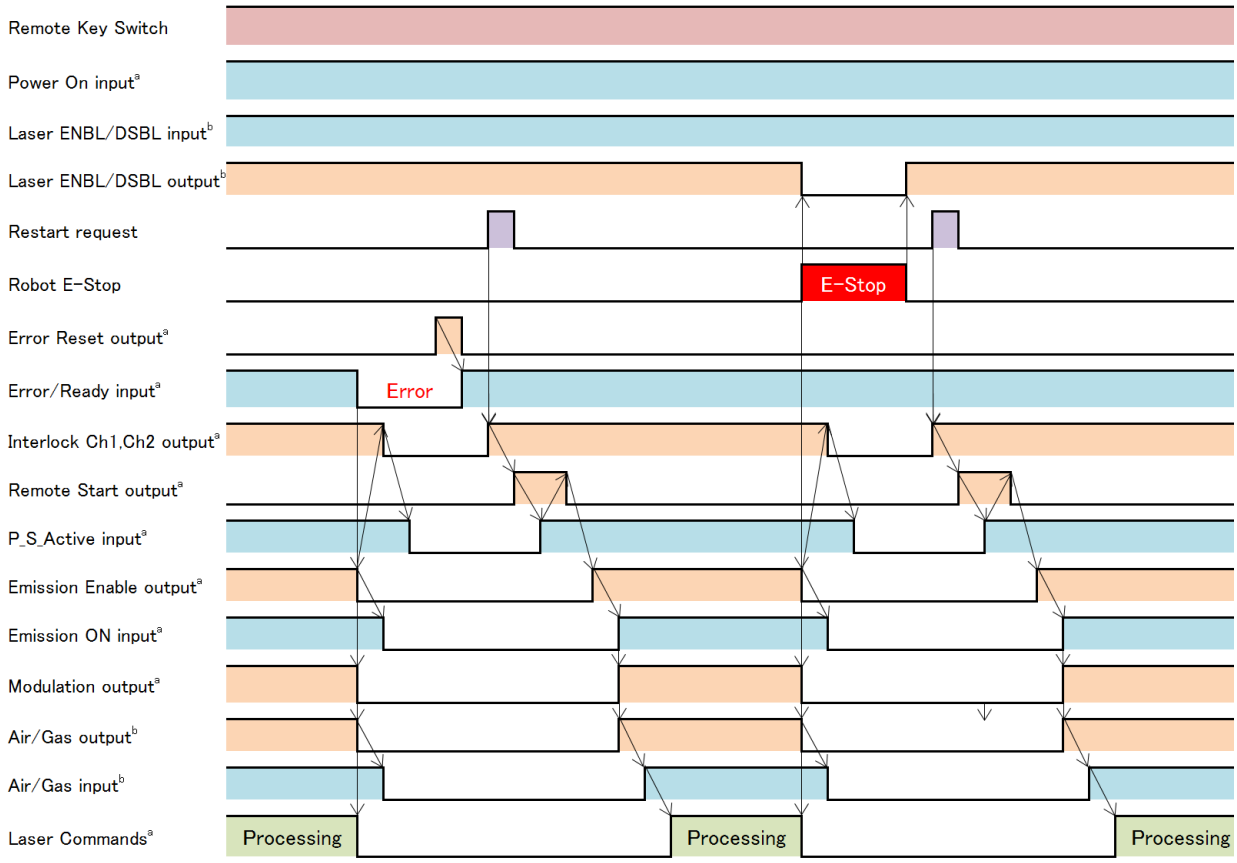


Fig. 70.6.2.2 When laser error or robot E-Stop occurs

NOTE
a. This is YLR laser oscillator I/O. See the section “YLR Laser Oscillator I/O”.
b. This is Peripheral equipment I/O. For details, refer to the “Peripheral Equipment I/O” section.

70.7 POWER CONTROL

Power control is a function that performs uniform processing by controlling the power, bottom power, frequency, and duty when the actual speed changes at the corners, etc. with respect to the processing V speed of the LS instruction.

70.7.1 Details

When the power control instruction is executed, the power control is activated according to the power control schedules defined in the power control instruction.
During power control, the output of each processing schedules (S, L, P, Q) is changed according to the ratio of the LS instruction processing speed V and the actual processing speed within the range of each processing schedules and power control schedules.

```
DATA Power Control Schedule
                                                    1/6

     Power control schedule number:                   1
  1  Comment:                  [                     ]
  2  Peak power(W):                        ENABLE/ 0
  3  Bottom power(W):                      ENABLE/ 0
  4  Frequency(Hz):                        ENABLE/ 5
  5  Duty(%):                              ENABLE/ 0
  6  Speed tolerance(%):                             0


[ TYPE ]                          END
```

Table 70.7.1 Power control schedule

	Power control schedules	
	Power control schedules ≦ Processing scheduls	Power control schedules ＞ Processing scheduls
Minimum power	Peak power (Power control scheule)	0 (Clamp value)
Minimum bottom power	Bottom power (Power control scheule)	0 (Clamp value)
Minimum frequency	Frequency (Power control scheule)	5 (Clamp value)
Minimum duty	Duty (Power control scheule)	0 (Clamp value)
Processing speed 0 power	0 (Clamp value)	Peak power (Power control scheule)
Processing speed 0 bottom power	0 (Clamp value)	Bottom power (Power control scheule)
Processing speed 0 frequency	5 (Clamp value)	Frequency (Power control scheule)
Processing speed 0 duty	0 (Clamp value)	Duty (Power control scheule)

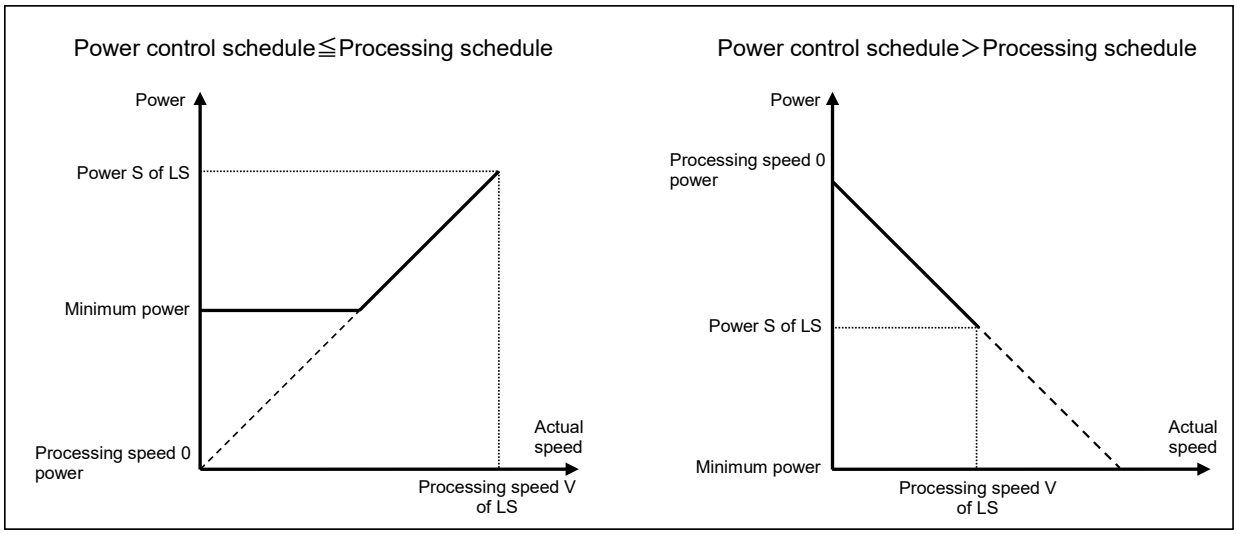


Fig. 70.7.1 Details of power control

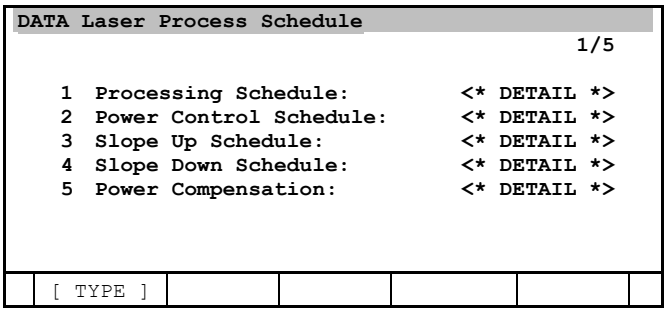
70.7.2 Power Control Schedules

Ten power control schedules are prepared.

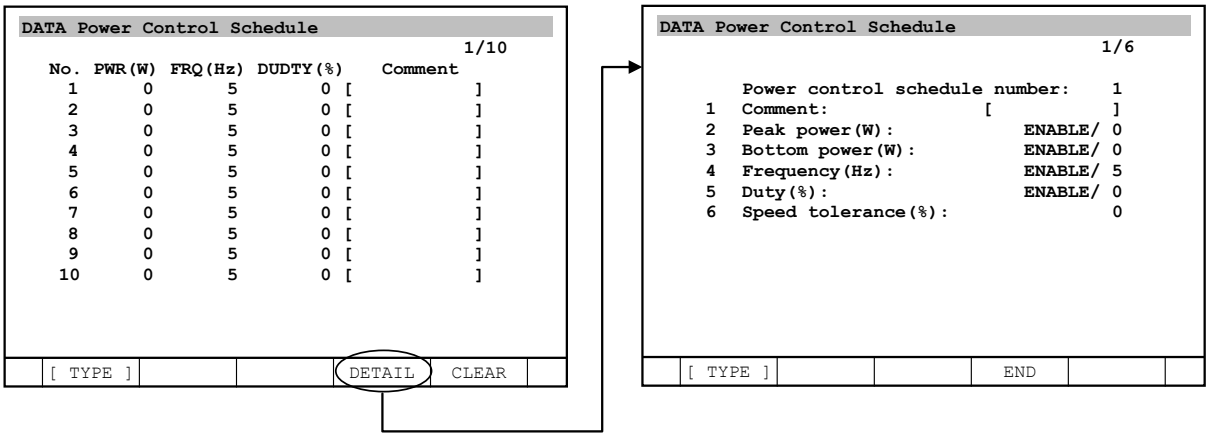
Procedure 70-11 Display power control schedule screen

Step

1 Press the [MENU] key and display the screen menu.
2 Press [0. -- NEXT --] and select [3. DATA].
3 Press F1[Type] and display the screen menu.
4 Select “Laser Schedule”.



5 Move the cursor to <*DETAIL*> of Power Control Schedule, and press [ENTER] key.



70.7.3 Power Control Instruction

- The format of power control instruction is shown below.

POWER_CONTROL[*]

– * : Power control schedule number (0 - 10)

Fig. 70.7.3(a) Power control instruction

- Teach the power control instruction before the LS instruction.
- If the power control schedule has been specified in range of 1 to 10 by the power control instruction before the LS instruction is executed, the power control becomes active. And, when 0 is specified to it, the power control is canceled.
- In the example below, the power control becomes active by POWER CONTROL [1] of the second line, and the power control is canceled by POWER CONTROL [0] of the 24th line.

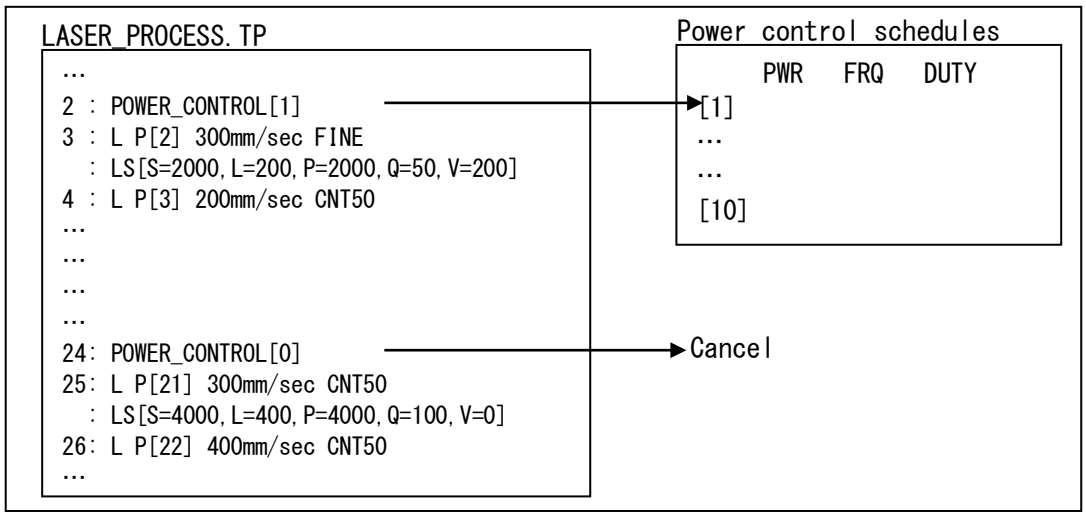


Fig. 70.7.3(b) Example of power control instruction execution

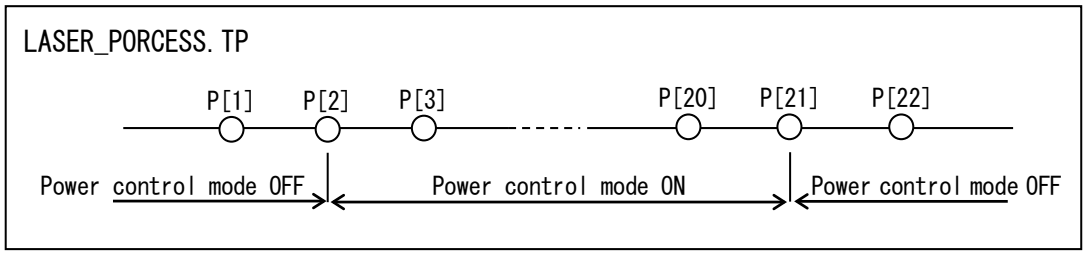
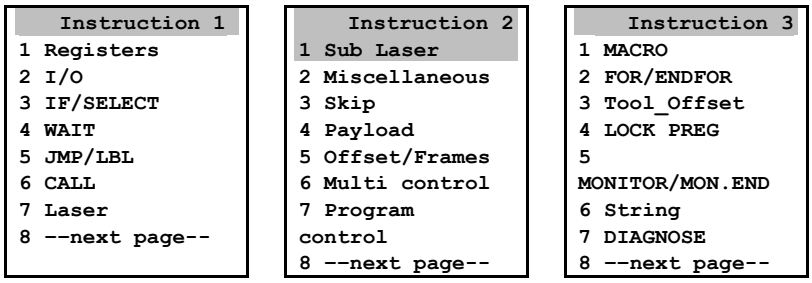


Fig. 70.7.3(c) Example of power control instruction execution

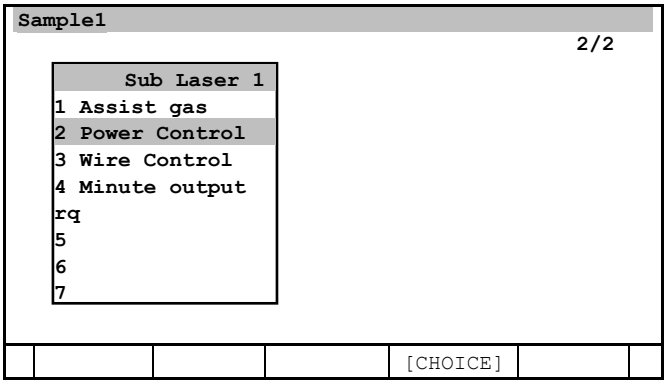
Procedure 70-12 Teaching power control instruction

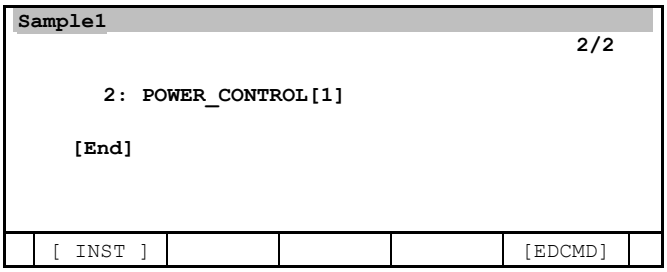
Step

1 To teach this instruction, press F1[INST], display sub menu, and select “Sub Laser”.



2 Select “Power Control” from Sub Laser menu.





70.7.4 Speed Tolerance

When speed tolerance is set, if the actual motion speed of the robot is less than or equal to the set speed tolerance, it is considered that the robot is moving at the command processing speed. Therefore, the power control is canceled only during that time.

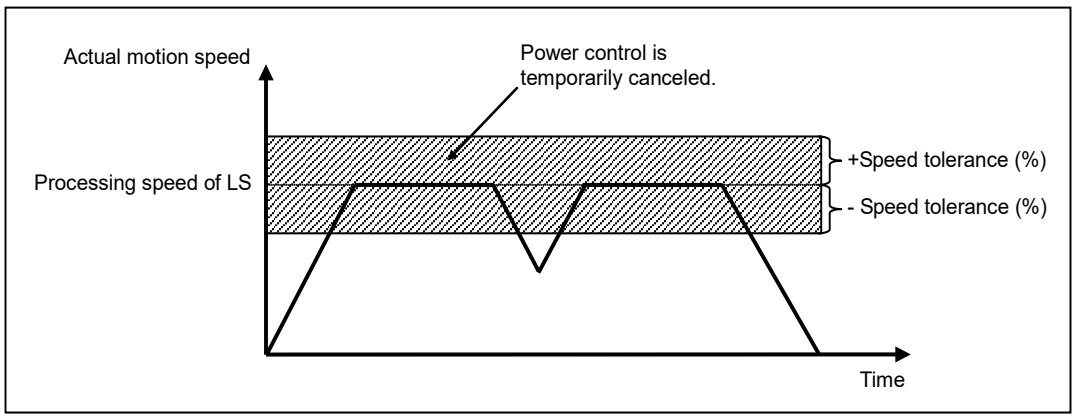
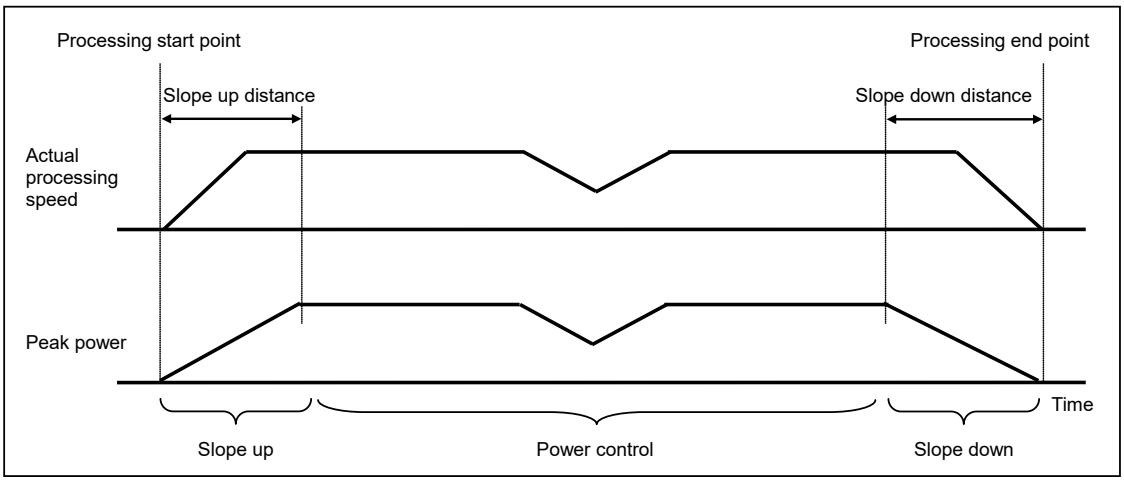


Fig. 70.7.4 Speed tolerance

70.7.5 Notes

- When both the power control schedules and the slope schedules are specified in one TP program, the slope control is given to priority in the vicinity of the processing start point or the processing end point, and the power control becomes active in other parts.



- Specify either of the linear motion or the circular motion to the motion format of the processing path if you want to use power control. You can't use power control when the joint motion is specified to the motion format.

- When the program is terminated or another line is selected, power control schedule registered by power control instruction is cleared.
- Power control can also be used with the LS_SCH instruction.

70.8 POWER CONTROL II

Power Control II allows you to still set up to 5 combinations of processing speed V and processing schedules (S, L, P, Q), so you can make finer adjustments than Power Control.

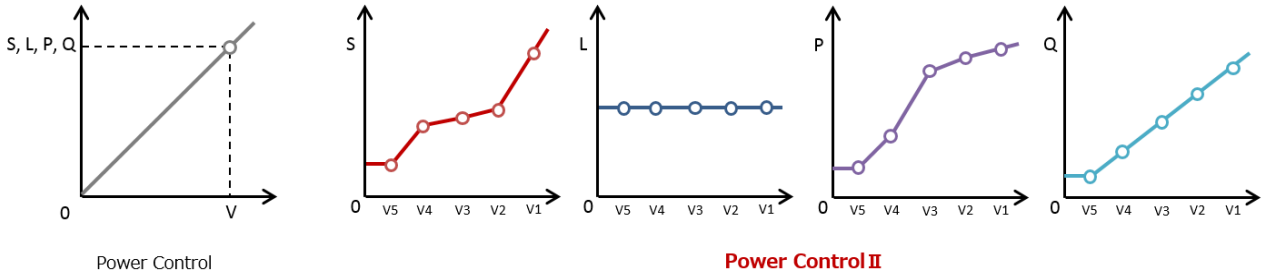


Fig. 70.8 Difference between Power Control and Power Control II

70.8.1 Details

- Power Control II settings are prepared for each processing schedule on the processing schedule list screen.
- Power Control II works when the processing schedule for which Power Control II is enabled are executed with the LS_SCH instruction.

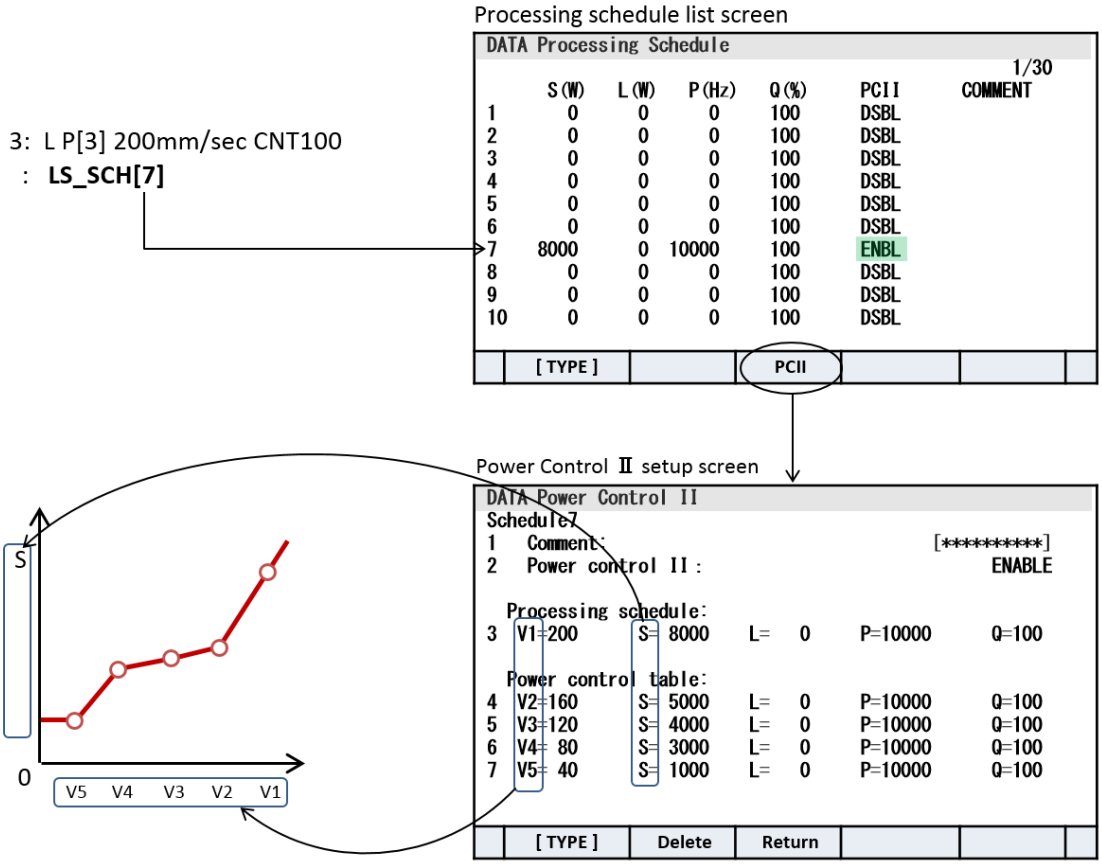


Fig. 70.8.1(a) Power Control II configuration

- While Power Control II is working, the laser command is obtained from the linear interpolation set in Power Control II and the actual robot speed (Act V).

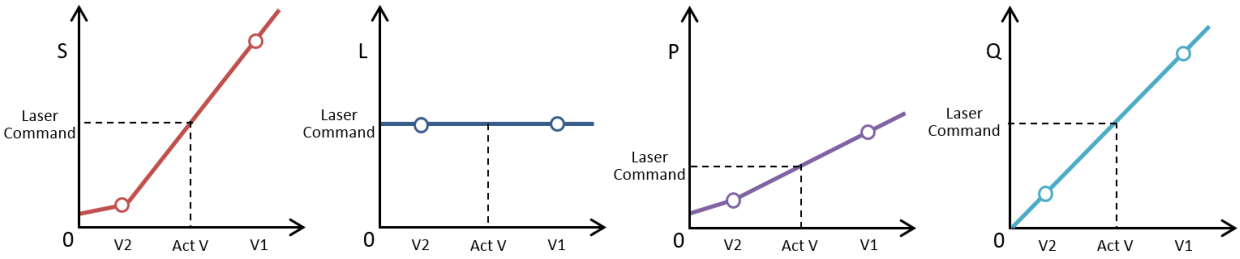


Fig. 70.8.1(b) Power Control II

Procedure 70-13 Power Control II settings

Step

1 Press the [MENU] key and display the screen menu.
2 Press [0. -- NEXT --] and select [3. DATA].
3 Press F1[Type] and display the screen menu.
4 Select “Laser Schedule”.

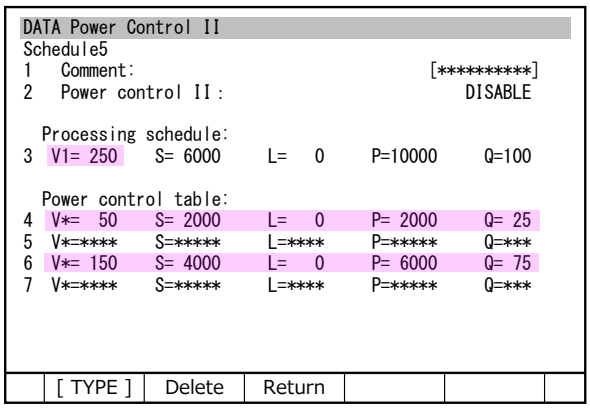
```
DATA Laser Process Schedule
                                                    1/5

   1  Processing Schedule:          <* DETAIL *>
   2  Power Control Schedule:       <* DETAIL *>
   3  Slope Up Schedule:            <* DETAIL *>
   4  Slope Down Schedule:          <* DETAIL *>
   5  Power Compensation:           <* DETAIL *>

 [ TYPE ]
```

5 Move the cursor to <*DETAIL*> of Processing Schedule, and press [ENTER] key.
6 Move the cursor to the processing schedule for which you want to set the power control II, and press F3 “PCII”.

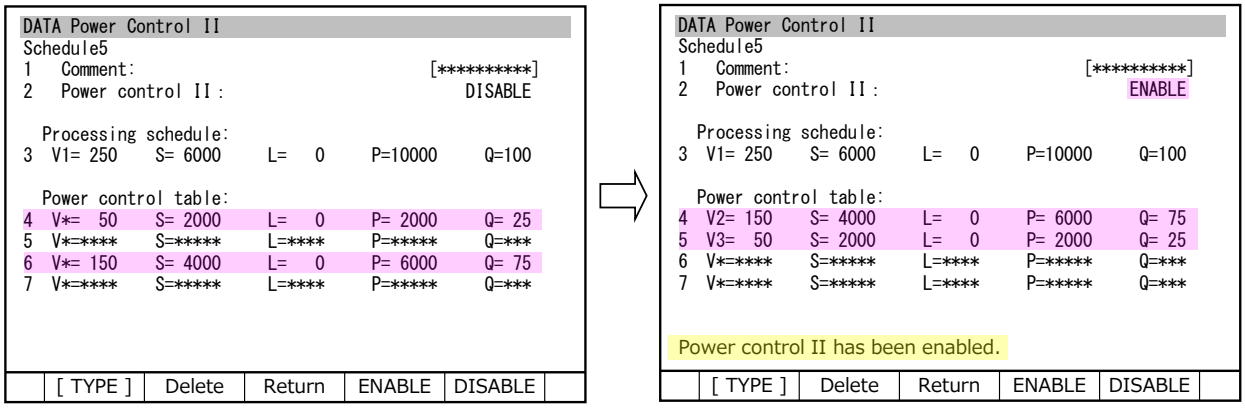
```
DATA Processing Schedule
                                                    1/30
 No.  S(W)  L(W)  P(Hz)  Q(%)  PCII   Comment
  1     0     0     1    100   DSBL
  2     0     0     1    100   DSBL
  3     0     0     1    100   DSBL
  4     0     0     1    100   DSBL
  5  6000     0     1    100   DSBL
  6     0     0     1    100   DSBL
  7     0     0     1    100   DSBL
  8     0     0     1    100   DSBL
  9     0     0     1    100   DSBL
 10     0     0     1    100   DSBL

 [ TYPE ]  |  PCII  |
```

```
DATA Power Control II
Schedule5
1   Comment:                                  [**********]
2   Power control II :                            DISABLE

  Processing schedule:
3  V1=   0    S= 6000     L=   0     P=    1     Q=100

  Power control table:
4  V*=****    S=*****     L=****     P=*****     Q=***
5  V*=****    S=*****     L=****     P=*****     Q=***
6  V*=****    S=*****     L=****     P=*****     Q=***
7  V*=****    S=*****     L=****     P=*****     Q=***

 [ TYPE ]  Delete  Return
```

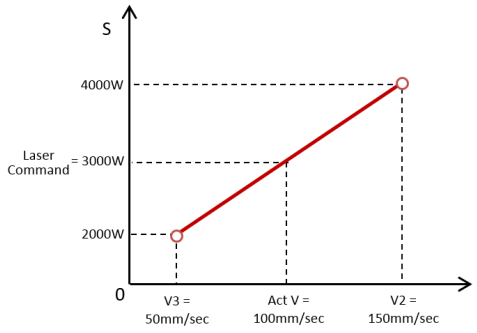
7 Set V1 of the processing schedule.
8 Add as many combinations of processing speed V and processing schedules (S, L, P, Q) as necessary in the power control table.



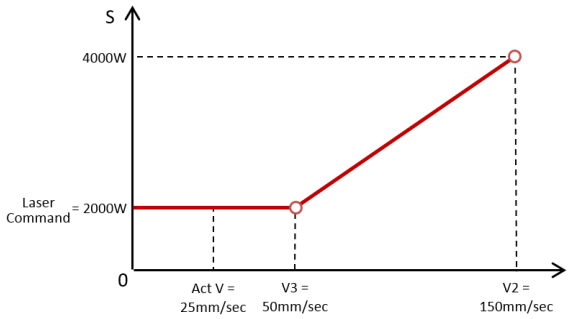
9 If you move the cursor to the item of Power Control II and press F4 “ENABLE”, after sorting processing speed in descending order, linear interpolation is created internally and Power Control II is enabled.



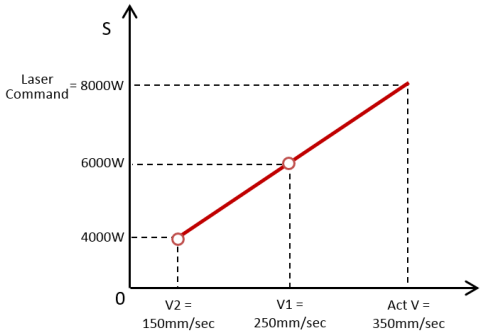
For example, if the robot actual speed (Act V) is 100mm/sec, 3000W will be output as a laser command using linear interpolation of (150mm/sec, 4000W) and (50mm/sec, 2000W).



If the robot actual speed is less than the set minimum of the power table, the laser command will be clamped to the minimum. For example, if the robot actual speed (Act V) is 25mm/sec, 2000W will be output as a laser command.



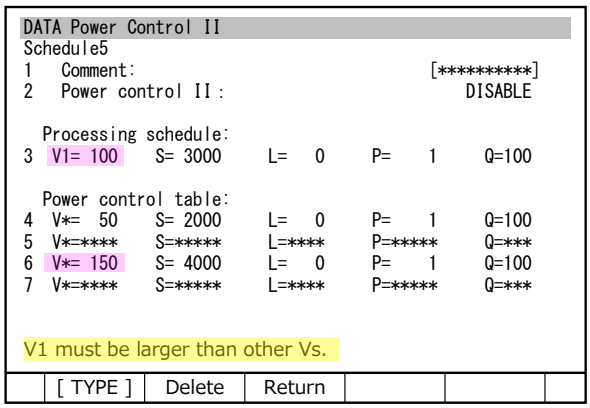
If the robot actual speed is greater than the processing schedule V1, the linear interpolation between the two largest points will be extended and compensated. For example, if the robot actual speed (Act V) is 350mm/sec, 8000W will be output as a laser command using the extension of the linear interpolation of (250mm/sec, 6000W) and (150mm/sec, 4000W).



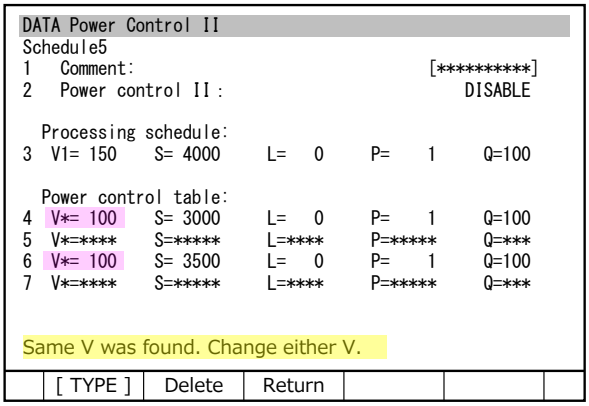
10 If you move the cursor to the item of Power Control II and press F5 “DISABLE”, Power Control II will be disabled and you can re-edit the power control table.

70.8.2 Notes

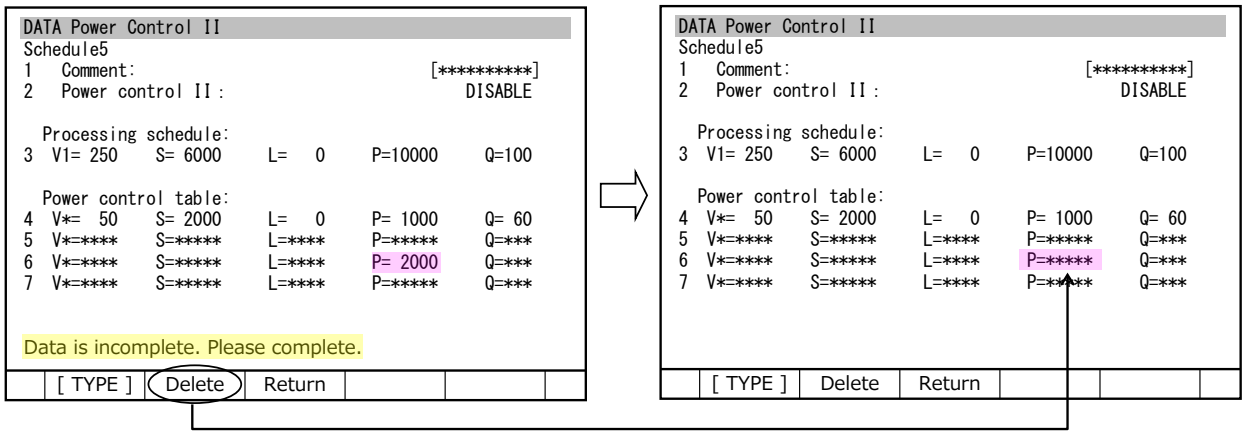
- If V in power control table is larger than V1 of processing schedule, “V1 must be larger than other Vs.” is displayed at prompt at bottom of the screen, and power control II forcibly returns to DISABLE.



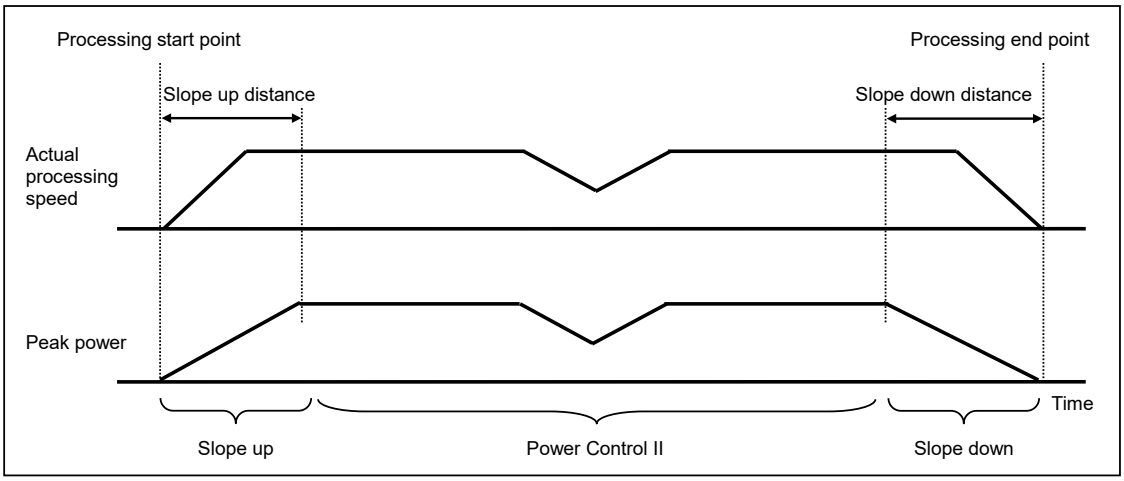
- If there are multiple same Vs in the power control table, prompt at bottom of the screen will display “Same V was found. Change either V.“ and Power Control II will be forcibly returned to DISABLE. Make sure that the Vs in the power control table do not overlap.



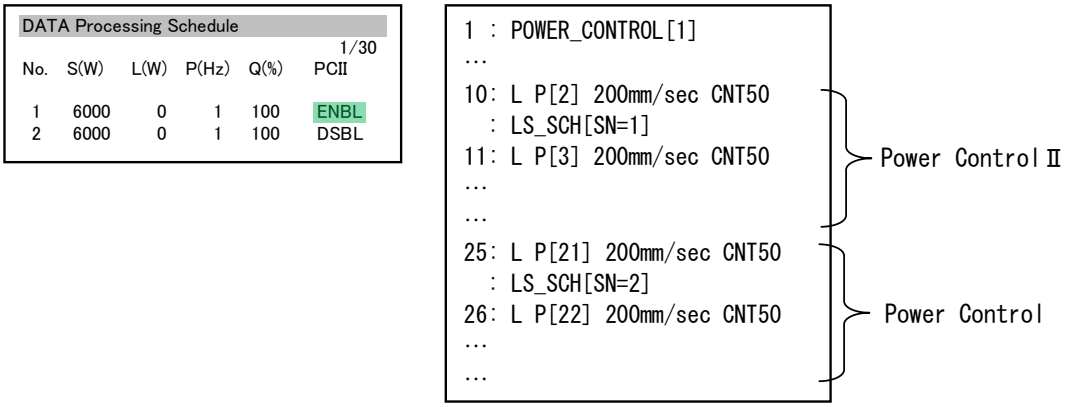
- If there is incomplete data in the power control table, prompt at bottom of the screen will display “Data is incomplete. Please complete.“ and Power Control II will be forcibly returned to DISABLE. If there is unnecessary data, move the cursor to the unnecessary data and press F2 “Delete” to delete it.



- When both Power Control II and slope schedules are specified in one TP program, slope control is given to priority in the vicinity of the processing start point or the processing end point, and Power Control II becomes active in other parts.



- If both Power Control II and Power Control are enabled, Power Control II takes precedence.



- When LS instruction is executed, Power Control II ends.
- Specify either of the linear motion or the circular motion to the motion format of the processing path if you want to use Power Control II. You can't use Power Control II when the joint motion is specified to the motion format.

70.9 SLOPE CONTROL

Slope control is a function that gradually increases or decreases the power, bottom power, frequency, and duty of the LS instruction during laser processing the specified section, and realizes the smooth rise and fall of the laser command.
Slope control has the following two functions.

- Slope up control: This can be used with LS instruction.
- Slope down control : This can be used with LS, LE instruction.

70.9.1 Slope Up Control

When the slope up instruction is executed, the slope up control is activated according to the slope up schedules defined in the slope up instruction.
When the slope up control works, the power, bottom power, frequency, and duty of the LS instruction gradually increase from the position where the slope up instruction is attached to the distance specified in the slope up schedules.

70.9.1.1 Details

The robot outputs the laser by using "Start peak power" until processing time passes "Stop time" while pausing at P[2], and changes the output of the laser toward power of LS after the processing time passes "Stop time". An actual laser output is changed in proportion to moved distance so that "Start peak power" may reach power of LS. An actual laser output is decreased in proportion to moved distance if power of LS is lower than "Start peak power".
"Start peak power", "Stop time" and "Slope up distance" can be specified by slope up schedule.

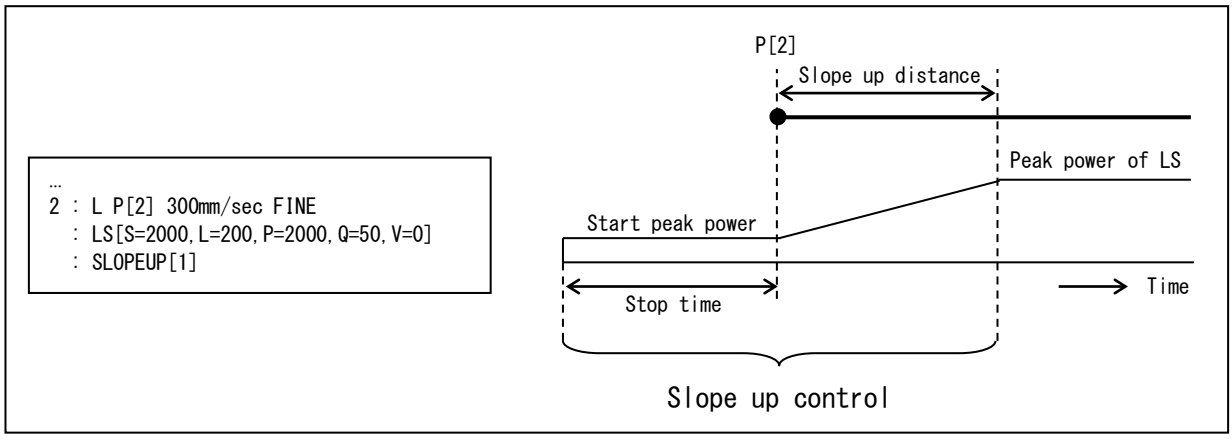


Fig. 70.9.1.1 Details of slope up control

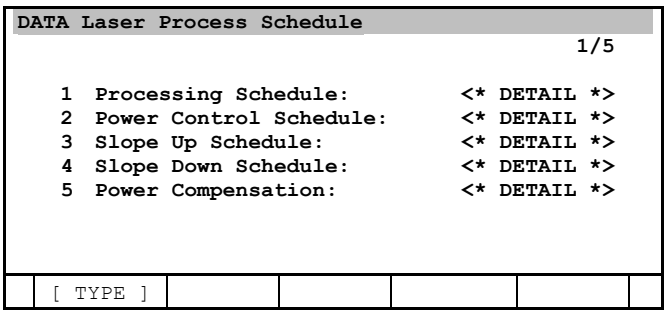
70.9.1.2 Slope up schedule

Ten slope up schedules are prepared.

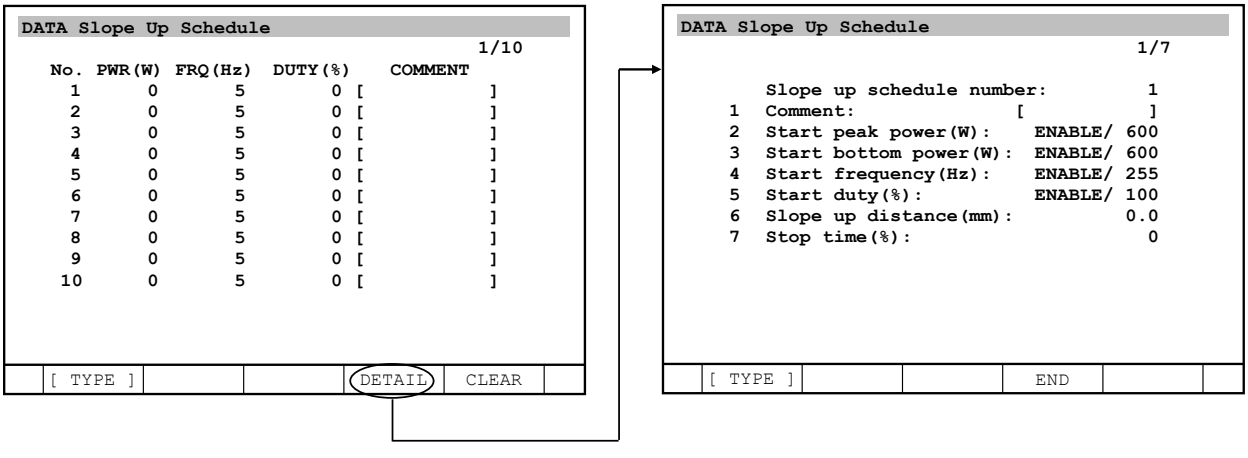
Procedure 70-14 Display slope up schedule screen

Step

1 Press the [MENU] key and display the screen menu.
2 Press [0. -- NEXT --] and select [3. DATA].
3 Press F1[Type] and display the screen menu.
4 Select “Laser Schedule”.



5 Move the cursor to <*DETAIL*> of Slope Up Schedule, and press [ENTER] key.



70.9.1.3 Slope up instruction

- The format of slope up instruction is shown below.

SLOPEUP[*]

– * : slope up schedule number (1 - 10)

Fig. 70.9.1.3(a) Slope up instruction

- Slope up instruction is provided as an additional motion instruction.
- Slope up instruction can only be added to the same line as the LS instruction.
- When the slope up instruction is executed, the slope-up control is activated according to the slope up schedules defined in the slope up instruction.
- In the example below, the slope up control becomes active by SLOPEUP [1] of the second line, and the slope up control continues until the robot reaches the distance in the slope up schedule. There is no slope up control in the 25th line because the 25th line LS instruction doesn't have the slope up instruction.

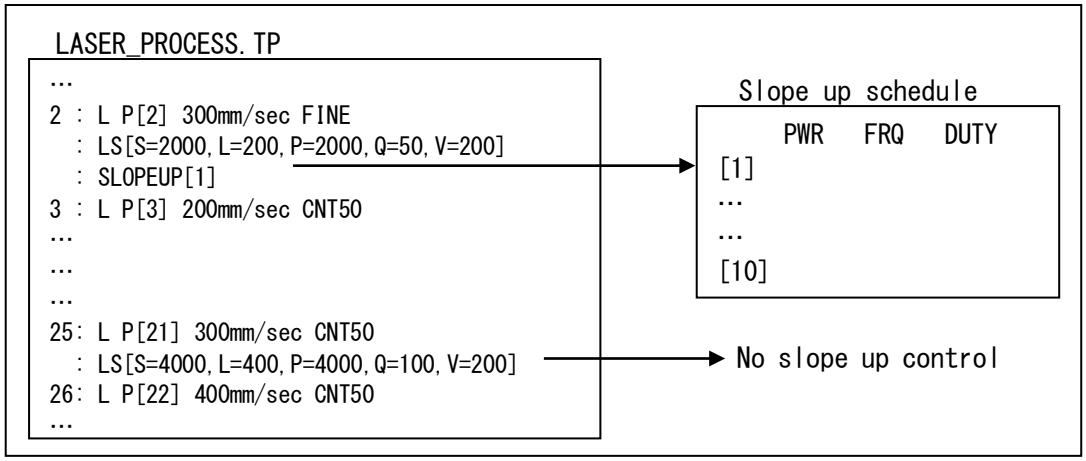


Fig. 70.9.1.3(b) Example of slope up instruction execution

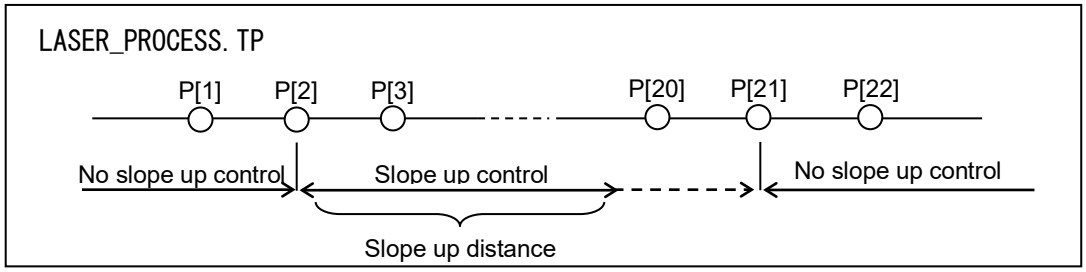


Fig. 70.9.1.3(c) Example of slope up instruction execution

NOTE

If there is a PRE_LS instruction, the robot will not stop even if “Stop time” of the slope up schedule is specified. Do not use the PRE_LS instruction if you want to stop by slope up.

Without stop

```
1 : L P[1] 200mm/sec FINE
  : PRE_LS
2 : L P[2] 200mm/sec FINE
  : LS[S=6000,L=0,P=1,Q=100,V=0]
  : SLOPEUP[1]
```

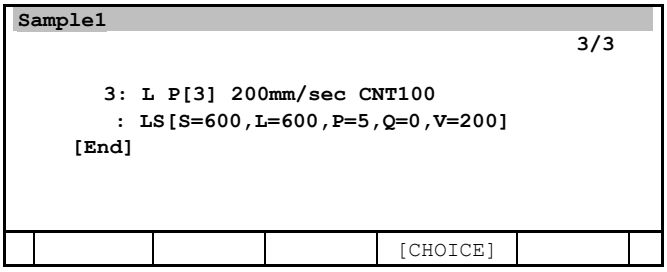
With stop

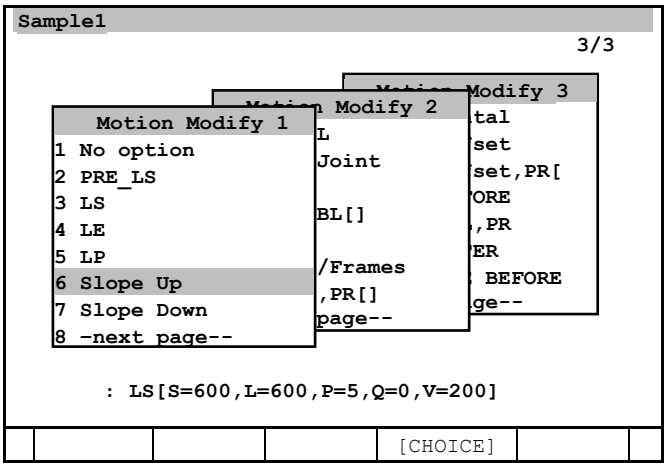
```
1 : L P[1] 200mm/sec FINE
2 : L P[2] 200mm/sec FINE
  : LS[S=6000,L=0,P=1,Q=100,V=0]
  : SLOPEUP[1]
```

Procedure 70-15 Teaching slope up instruction

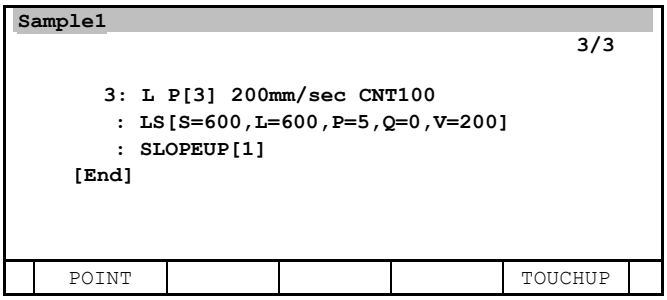
Step

1 To teach this instruction, place the cursor behind the LS instruction and press F4[CHOICE] to display the supplementary motion instruction menu.





2 Select “Slope Up” from the menu.



70.9.2 Slope Down Control

When the slope down instruction is executed, the slope down control is activated according to the slope down schedules defined in the slope down instruction.
When the slope down control works, the power, bottom power, frequency, and duty of the LS instruction gradually decrease from the position specified in the slope down schedule to the position where the slope down instruction is added.

70.9.2.1 Details

When robot is at the “Slope down distance” front from P[4], the robot changes the power of LS toward “End peak power” and an actual laser output is changed in proportion to moved distance so that power of LS may reach “End peak power” at P[4]. An actual laser output is increased in proportion to moved distance if “End peak power” is higher than power of LS.
“End peak power” and “Slope down distance” can be specified by slope down schedule.

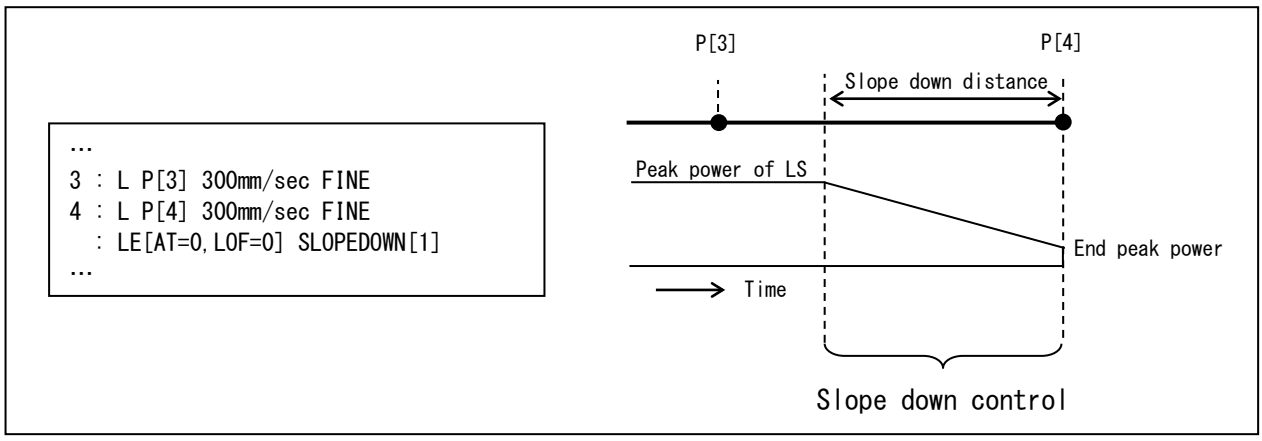


Fig. 70.9.2.1 Details of slope down control

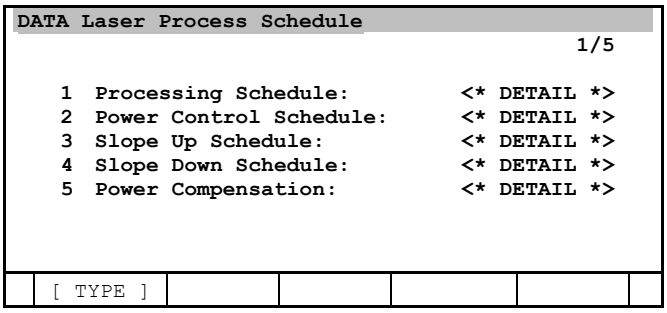
NOTE
In the case of above figure, slope down distance should be smaller than the distance between P[3] and P[4].

70.9.2.2 Slope down schedules

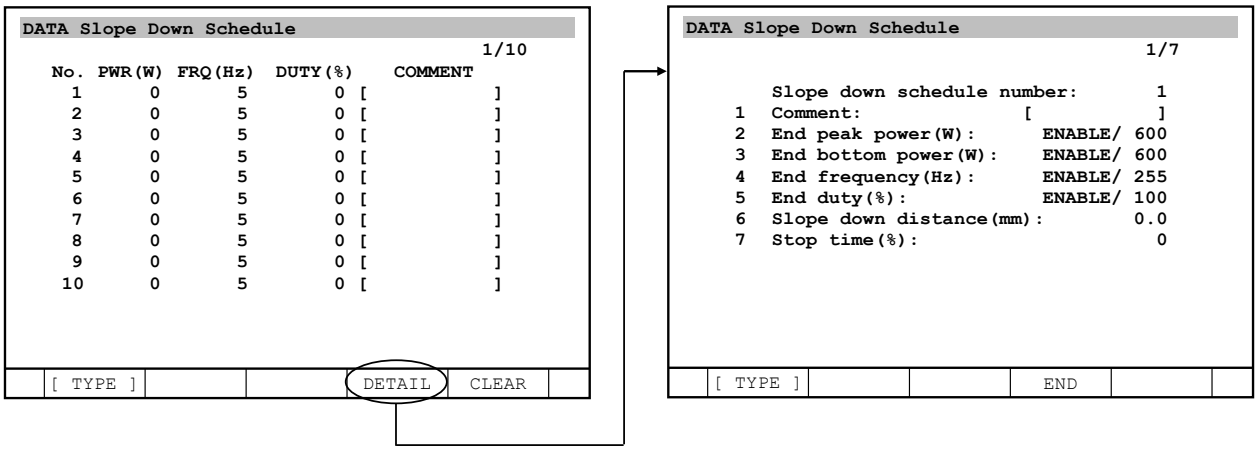
Ten slope down schedules are prepared.

Procedure 70-16 Display slope down schedule screen

Step
1 Press the [MENU] key and display the screen menu.
2 Press [0. -- NEXT --] and select [3. DATA].
3 Press F1[Type] and display the screen menu.
4 Select “Laser Schedule”.



5 Move the cursor to <*DETAIL*> of Slope Down Schedule, and press [ENTER] key.



70.9.2.3 Slope down instruction

- The format of slope up instruction is shown below.

SLOPEDOWN[*]

– * : Slope down schedule number (1 - 10)

Fig. 70.9.2.3(a) Slope down instruction

- Slope down instruction is provided as an additional motion instruction.
- Slope down instruction can only be added to the same line as the LS, LE instruction.
- When the slope down instruction is executed, the slope down control is activated according to the slope down schedules defined in the slope down instruction.
- In the example below, the slope down control is activated from the position where the robot enters the slope down distance to P[4].

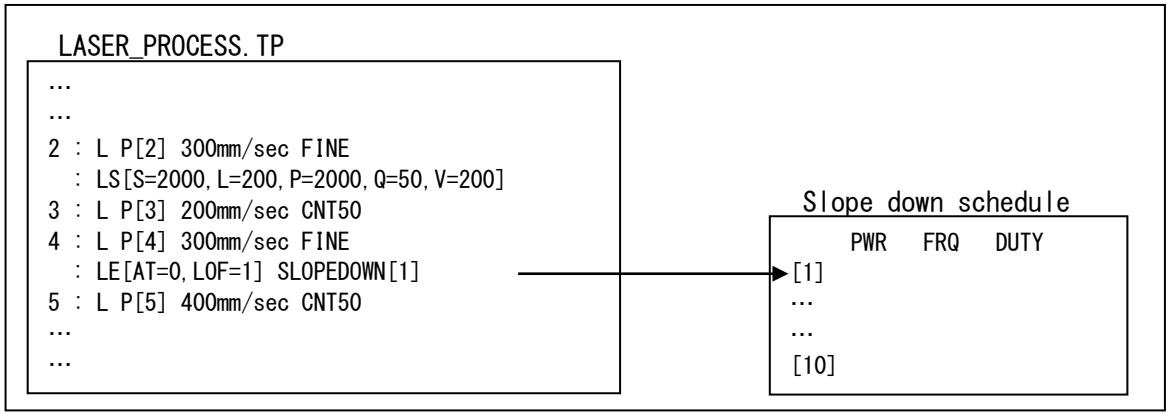


Fig. 70.9.2.3(b) Example of slope down instruction execution

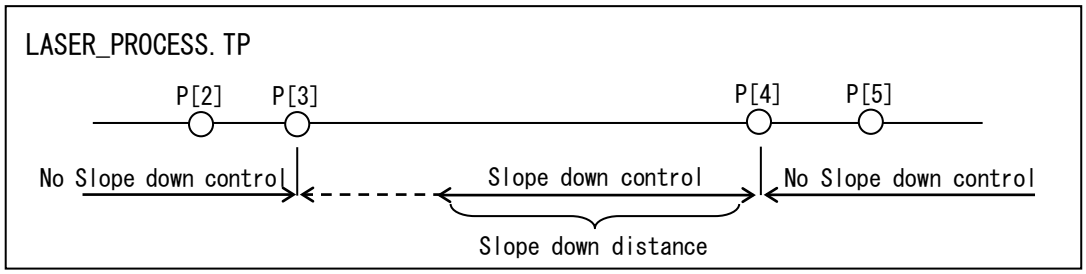
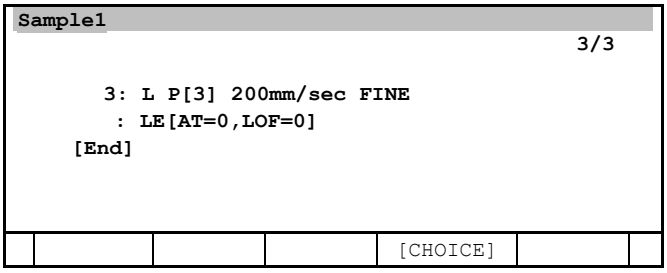


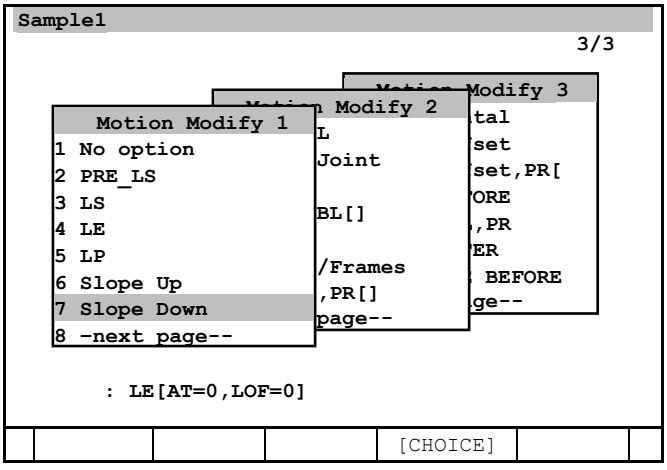
Fig. 70.9.2.3(c) Example of slope down instruction execution

Procedure 70-17 Teaching slope down instruction

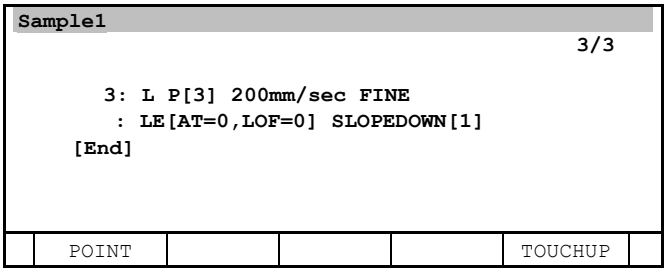
Step

1 To teach this instruction, place the cursor behind the LS, LE instruction and press F4[CHOICE] to display the supplementary motion instruction menu.



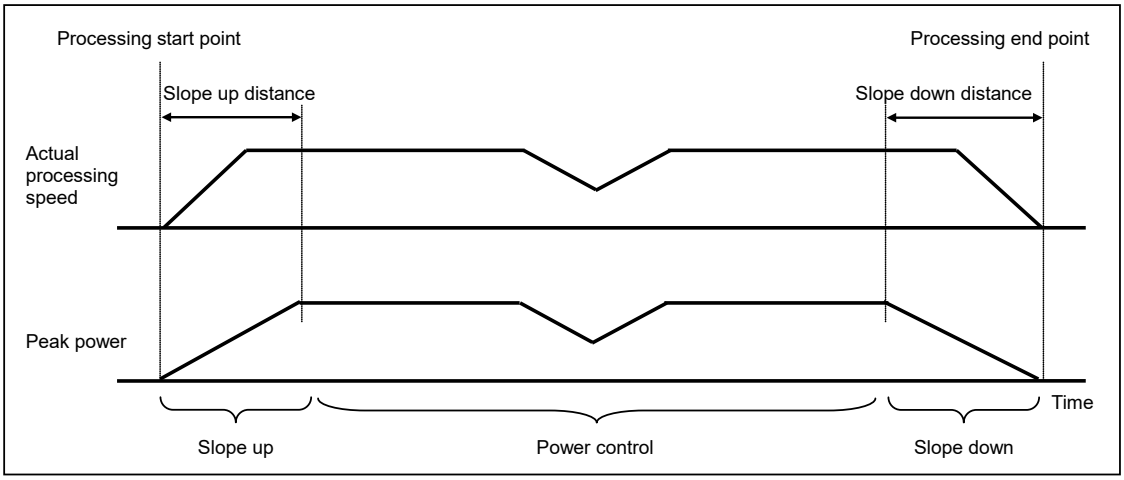


2 Select “Slope Down” from the menu.



70.9.3 Notes

- When both the power control schedules and the slope schedules are specified in one TP program, the slope control is given to priority in the vicinity of the processing start point or the processing end point, and the power control becomes active in other parts.



- Specify either of the linear motion or the circular motion to the motion format of the processing path if you want to use slope control. You can’t use slope control when the joint motion is specified to the motion format.
- When the program is terminated or another line is selected, slope up/down schedule registered by slope up/down instruction is cleared.
- Slope up/down instruction can also be used with the LS_SCH instruction.

70.10 POWER COMPENSATION

Power compensation function is a function that internally compensates the laser command so that "Power in processing schedules " and "Power under the processing head" are the same.

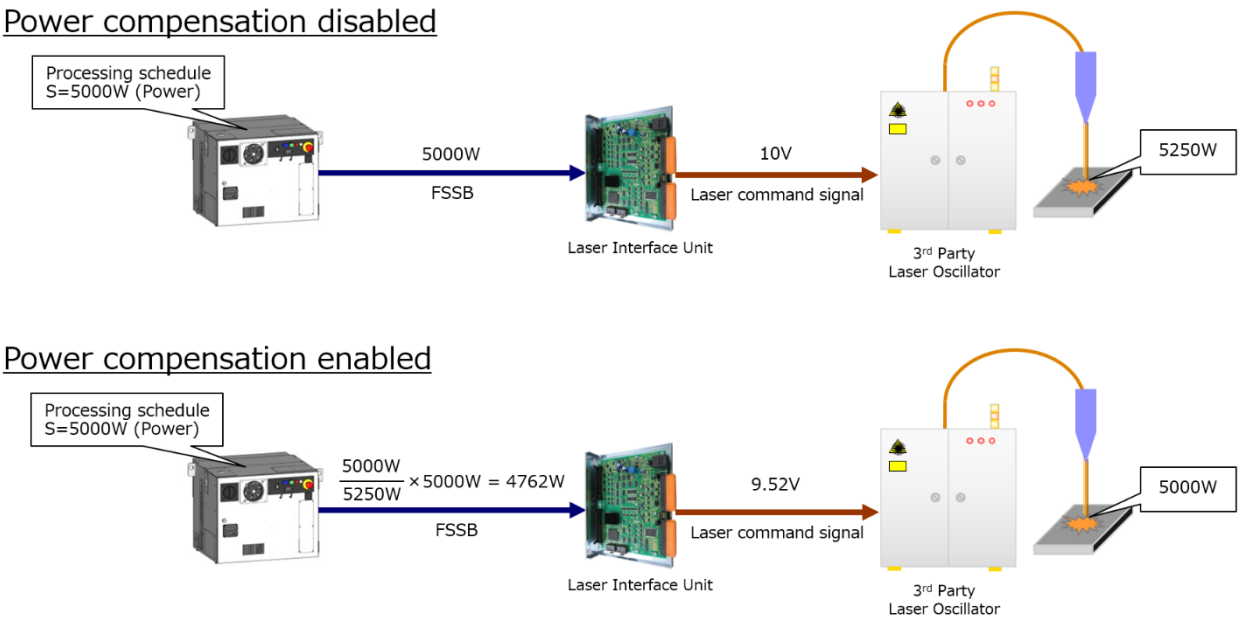


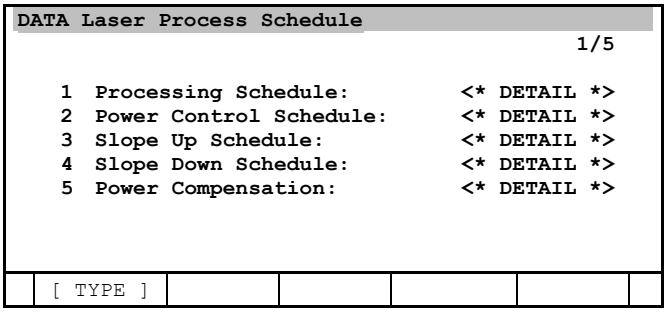
Fig. 70.10 (a) Power Compensation Function

Before using the power compensation function, it is necessary to set CmndPwr and ActPwr in the power compensation table.
CmndPw is the power commanded from the robot. ActPwr is the power actually measured under the procesing head.

Procedure 70-18 Display power compensation table

Step

1 Press the [MENU] key and display the screen menu.
2 Press [0. -- NEXT --] and select [3. DATA].
3 Press F1[Type] and display the screen menu.
4 Select “Laser Schedule”.



5 Move the cursor to <*DETAIL*> of “Power Compensation”, and press [ENTER] key.

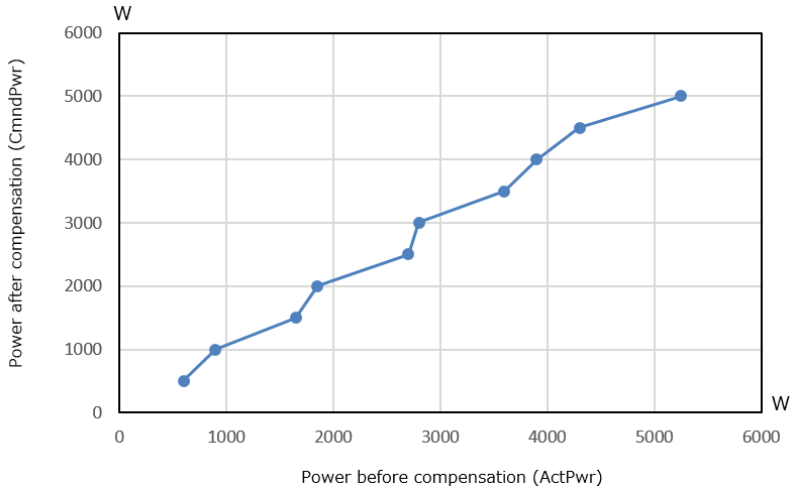
```
DATA Power Compensation
                                                    1/11

  1  Power Compensation:                    DISABLE

                CmndPwr                 ActPwr
  2               0  W                    0  W
  3               0  W                    0  W
  4               0  W                    0  W
  5               0  W                    0  W
  6               0  W                    0  W
  7               0  W                    0  W
  8               0  W                    0  W
  9               0  W                    0  W
 10               0  W                    0  W
 11               0  W                    0  W


[ TYPE ]
```

6 Set as many CmndPwr and ActPwr as necessary in the power compensation table. Up to 10 combinations of CmndPwr and ActPwr can be set to here.

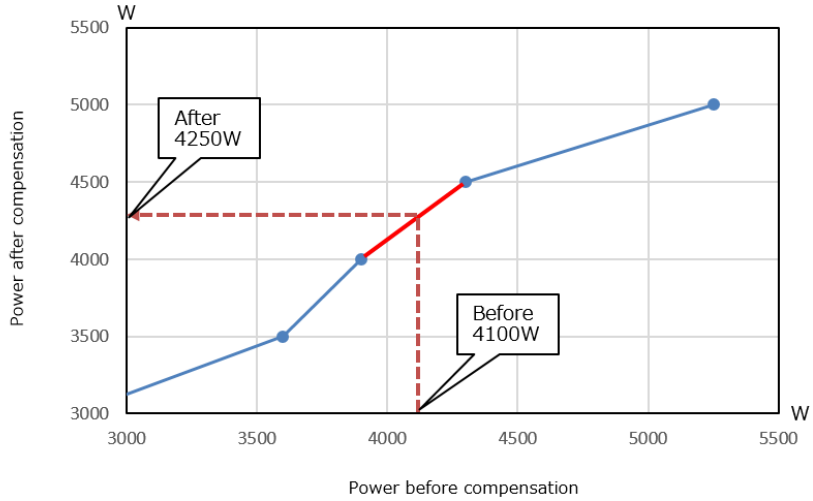
NOTE
Power compensation can be enabled if there is at least one combination of CmndPwr and ActPwr.

```
DATA Power Compensation
                                                    1/11

  1  Power Compensation:                    DISABLE

                CmndPwr                 ActPwr
  2             500  W                  600  W
  3            1000  W                  900  W
  4            1500  W                 1650  W
  5            2000  W                 1850  W
  6            2500  W                 2700  W
  7            3000  W                 2800  W
  8            3500  W                 3600  W
  9            4000  W                 3900  W
 10            4500  W                 4300  W
 11            5000  W                 5250  W


[ TYPE ]
```

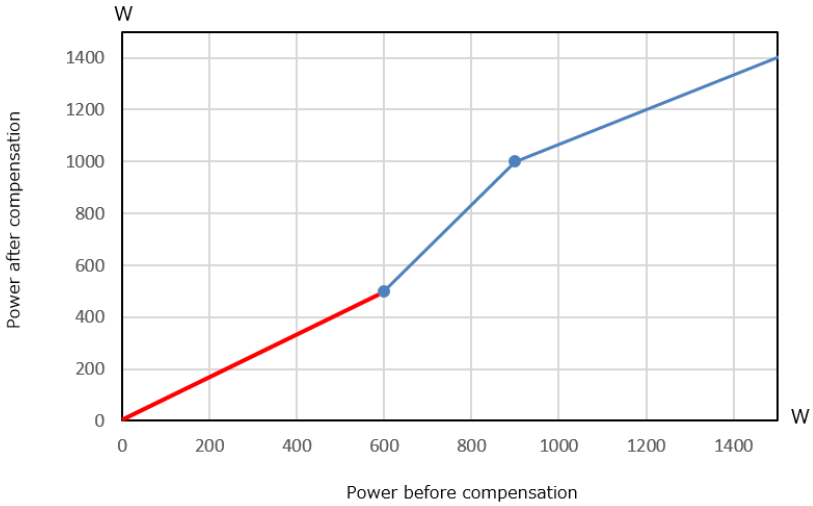
7 If you move the cursor to “Power Compensation” item and press F4 “ENABLE”, linear interpolation will be performed internally and power compensation will be enabled.



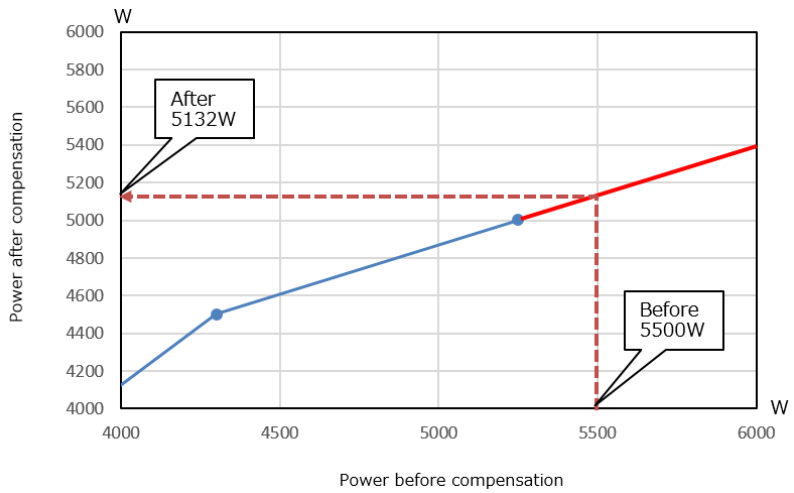
For example, when commanding 4100W, the compensation is performed using the linear interpolation part of the red line part below, and the power command after compensation is 4250W.



If power command is smaller than the minimum value in the set compensation table, it will be compensated by linear interpolation of 0W and the minimum value.



If power command is larger than the maximum value in the set compensation table, the linear interpolation between the two largest points will be extended and compensated.



8 If you press F4 “DISABLE”, power compensation will be disabled and you can re-edit the power compensation table.

70.10.1 Notes

NOTE
Even if power command after compensation becomes too large, the laser command signal of the laser interface unit will not output more than 10V.

- Power compensation can be enabled if there is at least one combination of CmndPwr and ActPwr.
- If F4 “ENABLE” is pressed without any CmndPwr, “CmndPwr data not found” will be displayed at the prompt at the bottom of the screen, and “Power Compensation” item will be forcibly returned to “DISABLE”.
- If F4 “ENABLE” is pressed while there are multiple same CmndPwr, “Same CmndPwr data was found” will be displayed at the prompt at the bottom of the screen, and “Power Compensation” item will be forcibly returned to “DISABLE”.
- If F4 “ENABLE” is pressed without any ActPwr, “ActPwr data not found” will be displayed at the prompt at the bottom of the screen, and “Power Compensation” item will be forcibly returned to “DISABLE”.
- If F4 “ENABLE” is pressed while there are multiple same ActPwr, “Same ActPwr data was found” will be displayed at the prompt at the bottom of the screen, and “Power Compensation” item will be forcibly returned to “DISABLE”.
- If F4 “ENABLE” is pressed on the data when ActPwr is decreasing, “ActPwr data is decreasing” is displayed at the prompt at the bottom of the screen, and “Power Compensation” item will be forcibly returned to “DISABLE”.
- If F4 “ENABLE” is pressed when there is no valid data, “Useful data not found” will be displayed at the prompt at the bottom of the screen, and “Power Compensation” item will be forcibly returned to “DISABLE”.

70.11 WIRE CONTROL

This function controls ON/OFF of wire feeder from robot. According to wire feed schedule defined beforehand by wire control instruction, wire control is instructed to turn ON by LS instruction and turn OFF by LE instruction or PAUSE/END of the program.
When “Wire Feed” input is assigned, “Wire Feed” input is start to be checked with LS instruction. The check continues during laser processing.
In case that Laser Enable/Disable state is Disable or that Process disable for restart is disabled, wire control doesn’t function.

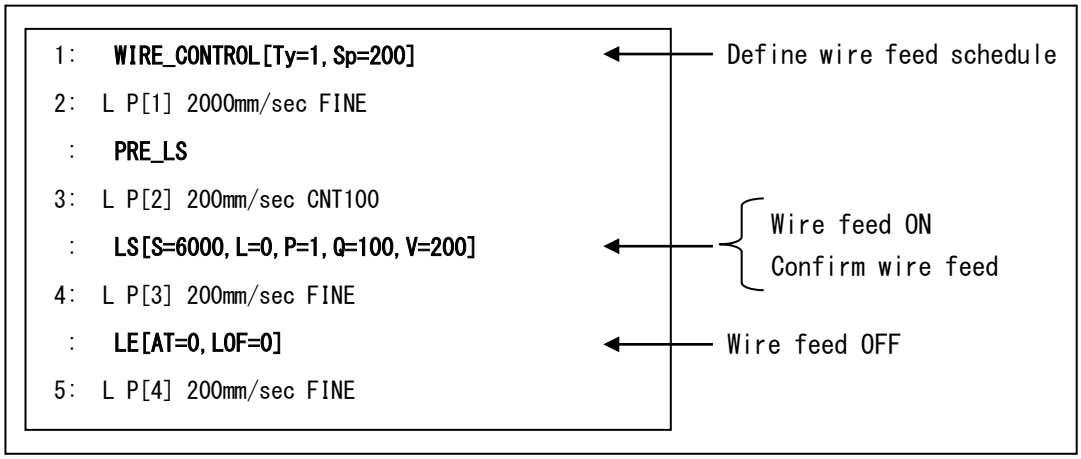
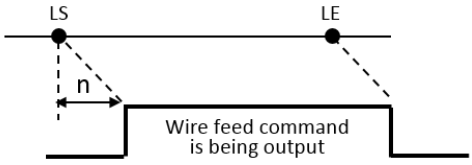


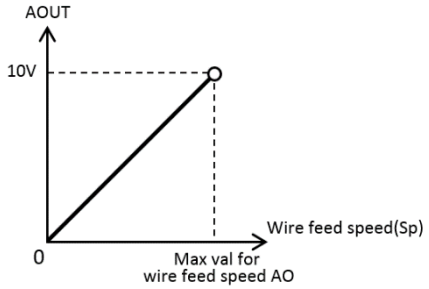
Fig. 70.11 Sample program of wire control instruction

70.11.1 Wire Control Setup

Wire control setup are settings related to wire control. Make the settings as necessary.

Table 70.11.1 Wire control setup screen

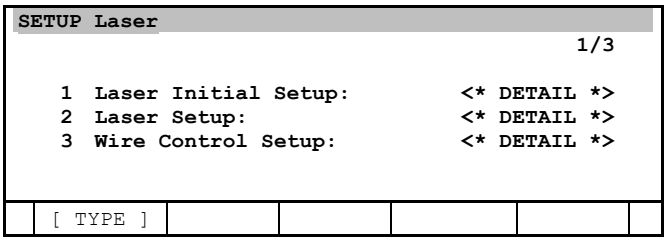
Items	Descriptions
Wire feed delay time(ms)	This item can delay the output of the wire feed command (“Wire Feed” output and “Wire Feed Speed” output) for a certain period of time.  The minimum unit for this item is ITP (it cannot be fine-tuned than ITP).
Wire feed OK latency(s)	If “Wire Feed” input is assigned, check whether “Wire Feed” input is on after executing the LS instruction. If “Wire Feed” input does not turn on after the time set here, an alarm is occurred.
Slope for wire feed speed GO	This item is used to convert the wire feed speed specified in wire control instruction to the desired unit. Group output of wire feed speed output ＝ Wire feed speed in wire control instruction × Slope for wire feed speed GO ＋ Offset for wire feed speed GO
Offset for wire feed speed GO	This item is used to convert the wire feed speed specified in wire control instruction to the desired unit. Group output of wire feed speed output ＝ Wire feed speed in wire control instruction × Slope for wire feed speed GO ＋ Offset for wire feed speed GO

Items	Descriptions
Max val for wire feed speed AO	The analog output signal of “Wire Feed Speed” output outputs the value in the range of 0 to “Max val for wire speed AOUT” at 0 to 10-15V. For example, if you want to output a value in the range of 0 to 200 cm/min of wire feed speed at 0 to 10-15V, set 200 to this item. 

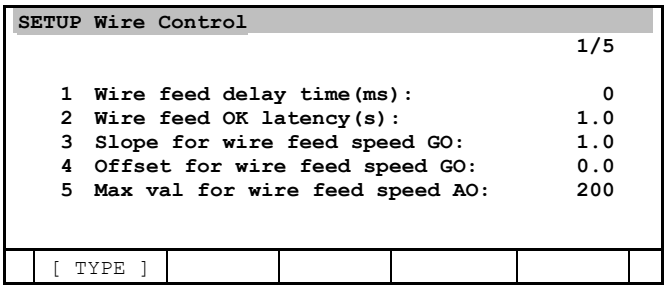
Procedure 70-19 Display laser setup screen

Step

1 Press the [MENU] key then screen menu will be displayed.
2 Select [6 SETUP].
3 Show Menu by pressed F1 [TYPE].
4 Select “Laser Setup”.



5 Move cursor on <* DETAIL *> of “Wire Control Setup” and press [ENTER] key.



70.11.2 Wire Control Instruction

- The format of wire control instruction is shown below.

WIRE_CONTROL[Ty=*, Sp=*]

– Ty : Wire feed type (0 - 2)
– Sp : Wire feed speed (1 - 32767)

Fig. 70.11.2(a) Wire control instruction

- Teach wire control instruction before LS instruction.
- For the wire feed type (Ty =), specify the type of wire feed.
  Ty=0: Off is output to “Wire Feed” output, and 0 is output to “Wire Feed Speed” output.
  Ty=1: It is a constant speed feeding type. “Wire Feed” output is turned on, and the value specified by the wire feed speed (Sp =) is output as “Wire Feed Speed” output. “Wire Feed Speed” output does not change even if the actual prcessing speed changes.
  Ty=2: It is a speed tracking type. “Wire Feed” output is turned on, and the value assigned to wire feed speed (Sp=) changes with proportion to the ratio of processing speed V of LS instruction and actual processing speed, and it is output as “Wire Feed Speed” output.

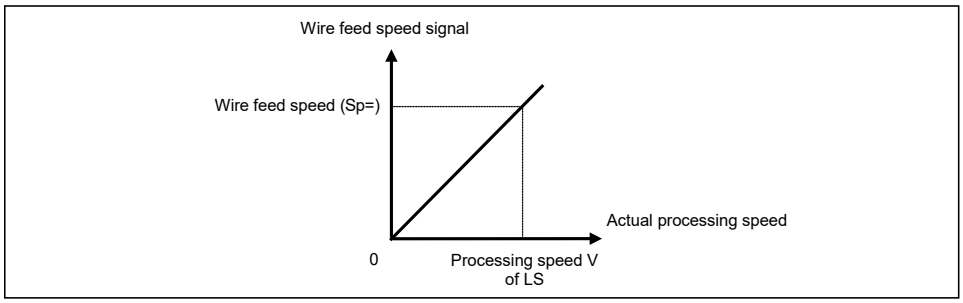
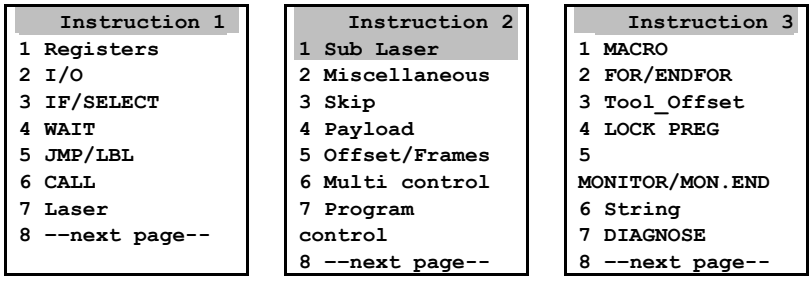


Fig. 70.11.2(b) Speed tracking type

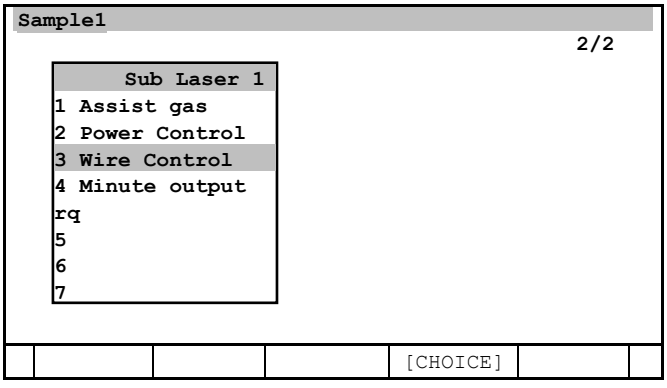
Procedure 70-20 Teaching wire control instruction

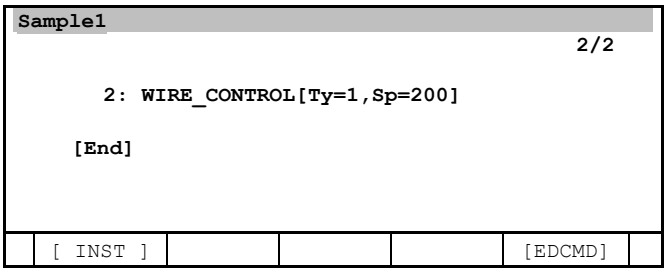
Step

1 To teach this instruction, press F1[INST], display sub menu, and select “Sub Laser”.



2 Select “Wire Control” from Sub Laser menu.





70.11.3 Wire Control Signal

70.11.3.1 Wire Feed output

- If you specify a wire feed type (Ty =) other than 0 in the wire control instruction, “Wire Feed” output is turned on when the LS instruction is executed.
- “Wire Feed” output is assigned on the peripheral equipment output screen.

70.11.3.2 Wire Feed Speed output

- When wire feed speed (Sp=) is assigned by wire control instruction, wire feed speed is output by “Wire Feed Speed” output as Group Output with the execution of LS instruction. By receiving this signal outside, wire feeder can adjust wire feed speed automatically.
- By changing the "Slope for wire feed speed GO" and "Offset for wire feed speed GO" on the wire control setup screen, wire feed speed (Sp=) in wire control instruction can be converted to the desired unit and output.

  Group output of wire feed speed output =
  Wire feed speed in wire control instruction × Slope for wire feed speed GO
  + Offset for wire feed speed GO

- When wire feed type (Ty =) is 2, the value assigned to wire feed speed (Sp=) changes with proportion to the ratio of processing speed V of LS instruction and actual processing speed, and it is output as “Wire Feed Speed” output.
- “Wire Feed Speed” output is assigned on the peripheral equipment output screen.
- “Wire Feed Speed” output signal can also be output in Analog Output.
- Output range of “Wire Feed Speed” analog output is 0 to “Max val for wire feed speed AO” as output from 0 to 10-15V. For example, if you want to output wire feed speed the range of which is 0 to 200 inch/min as 0 to 10-15V, set 200 to “Max val for wire feed speed AO” and set the value of 0~200 to Sp. “Max val for wire feed speed AO” can be changed on wire control setup screen.

70.11.3.3 Wire Feed input

- The wire feeder that feeds the wire is equipped with a wire break detector, which allows you to check the wire break during feeding.
- Make sure that on is input to this signal when the wire is being fed correctly, and off is input when the wire is broken. If this signal is on during laser processing, laser processing will continue.
- During laser processing, if this signal does not turn on even after the "Wire feed OK latency" on the wire control setup screen has elapsed, an alarm is occurred.
- “Wire Feed” input is assigned on the peripheral equipment input screen.

70.11.4 Notes

- Wire control can also be used with the LS_SCH instruction.

70.12 LS_TA INSTRUCTION WITH TIME ADJUSTMENT

70.12.1 LS_TA Instruction

- LS_TA instruction is an instruction to start laser processing like the LS instruction.
- The format of the LS_TA instruction is shown below.

LS TA[S=*, L=*, P=*, Q=*, V=*, TA=*]

- S : Power(W)
- L : Bottom power(W)
- P : Frequency(Hz)
- Q : Duty(%)
- V : Processing speed(mm/sec, cm/min, inch/min)
- TA : Time adjustment(-99999msec – 99999msec)

Fig. 70.12.1(a) LS_TA instruction

- Execute the PRE_LS instruction before executing the LS_TA instruction.
- The LS_TA instruction is executed using a PR_CALL instruction.

NOTE
- If there is no PRE_LS instruction before the LS_TA instruction, the LS_TA instruction will not be executed.
- The LS_TA instruction cannot be executed when the direct power override mode is enabled.
- You cannot use the CALL instruction to execute the LS_TA instruction.

- When the PR_CALL LS_TA instruction is executed, the output timing of the processing schedules (S, L, P, Q and V) can be adjusted by the argument TA.
  If the argument TA is a negative value, the processing schedules (S, L, P, Q and V) will be output in advance for the specified time based on the teaching position or the closest point on the motion path. If the argument TA is a positive value, the processing schedules (S, L, P, Q and V) will be output delayed by the specified time, based on the teaching position or the closest point on the motion path.

```
2:  L P[2] 200mm/sec FINE PRE_LS
3:  L P[3] 200mm/sec CNT100
 :    PR_CALL LS_TA[S=5000,L=0,P=5,Q=100,V=200,TA=-20]
4:  L P[4] 200mm/sec CNT100
 :    PR_CALL LS_TA[S=6000,L=0,P=5,Q=100, V=200,TA=15]
5:  L P[5] 200mm/sec CNT100
```

Fig. 70.12.1(b) Sample program for PR_CALL LS_TA instruction

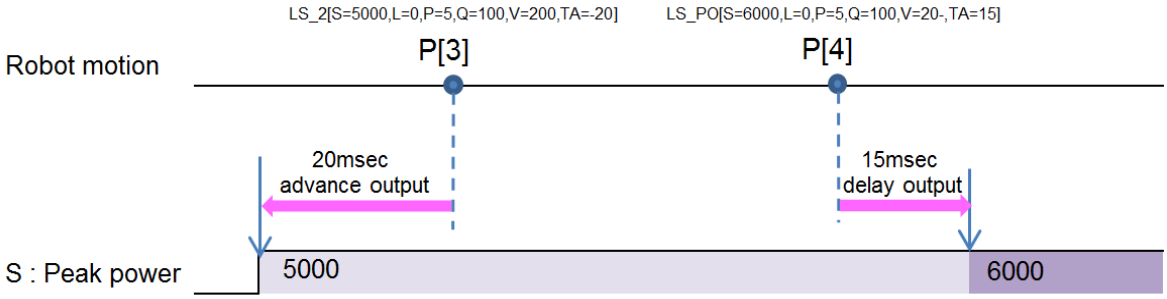


Fig. 70.12.1(c) Time chart when using PR_CALL LS_TA instruction

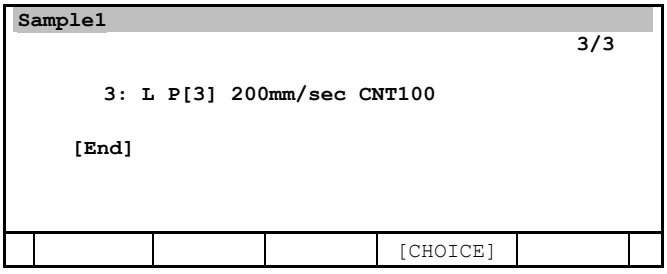
- When the "LSTA start time adjustment" on the laser setup screen is set, the output timing of the processing schedules (S, L, P, Q and V) is the sum of the "LSTA start time adjustment" and the argument TA. For example, if the "LSTA start time adjustment" is -24 msec and the argument TA is -6 msec, the processing schedules (S, L, P, Q and V) will be output -30 msec ahead.

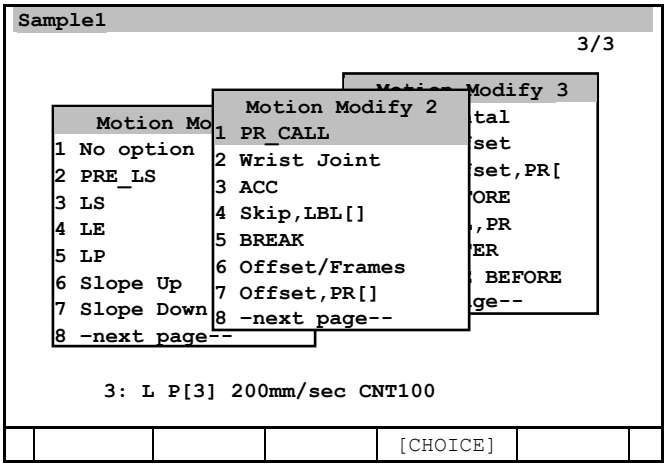
NOTE
- Slope-up and slope-down instructions cannot be used with the PR_CALL LS_TA instruction.
- PR_CALL LS_TA instruction supports only G1. If there are multiple groups in the TP program, PR_CALL LS_TA instruction works only for G1. If G1 does not exist in the TP program, PR_CALL LS_TA instruction will have no work.

Procedure 70-21 Teaching PR_CALL LS_TA instruction

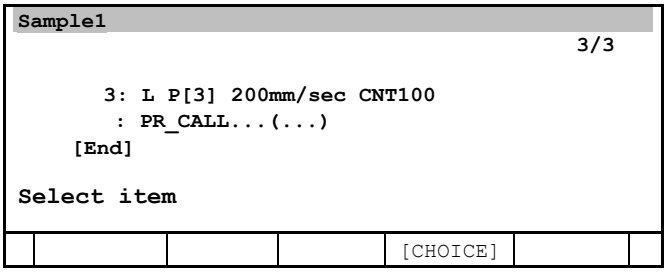
Step

1 Place the cursor behind the motion instruction and press F4 [CHOICE] to display the additional motion instruction menu.

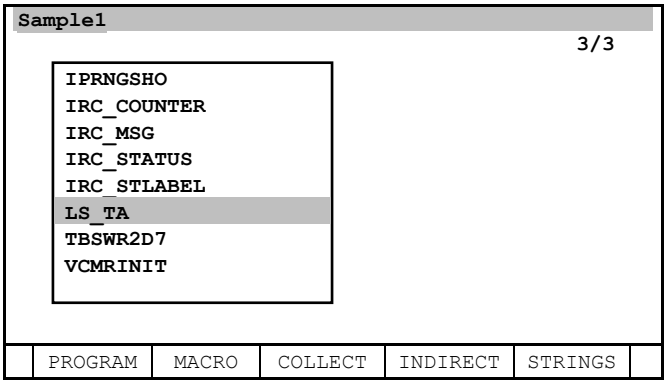


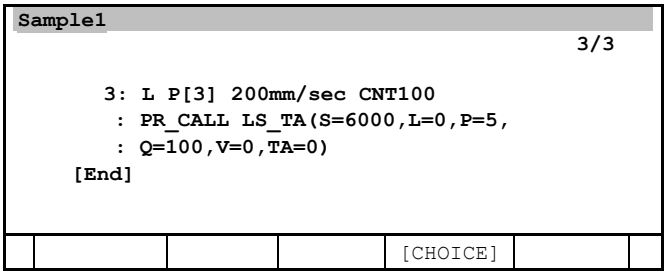


2 Select "PR_CALL" from the motion instruction menu.



3 Press F4 [CHOICE] then F3 “COLLECT” and select “KAREL Progs”. A list will be displayed. Select LS_TA.





70.12.2 LS_SCH_TA Instruction

- Like the LS_SCH instruction, the LS_SCH_TA instruction is an instruction to start laser processing using the processing schedule number.
- The format of the LS_SCH_TA instruction is shown below.

LS_SCH_TA[SN=*, TA=*]

– SN : Processing schedule number (0 – 30)

– TA : Time adjustment (-99999msec – 99999msec)

Fig. 70.12.2 LS_SCH_TA instruction

- Usage of LS_SCH_TA instruction is the same as that of LS_TA instruction. See section on LS_TA instruction for how to use the LS_SCH_TA instruction.

70.13 PROCESSING STATE RECORD FUNCTION

Processing status record function is a function that records the following data in chronological order during laser processing.

Table 70.13 Recorded data

Items	Descriptions
TIME(ms)	Time
POWER(W)	Power sent to laser oscillator
BOTTOM_POWER(W)	Pulse bottom power sent to laser oscillator
FREQUENCY(Hz)	Frequency sent to laser oscillator
DUTY(%)	Duty sent to laser oscillator
ACTUAL_POWER(W)	Power Monitor on laser oscillator input screen
PROCESS_SPEED	Actual processing speed
FEED_SPEED	Actual wire feed speed

- These data are recorded in the ITP cycle, and when the LE instruction is executed, the file in which the data is recorded is output to an external device.
- The output file is made with the format below.

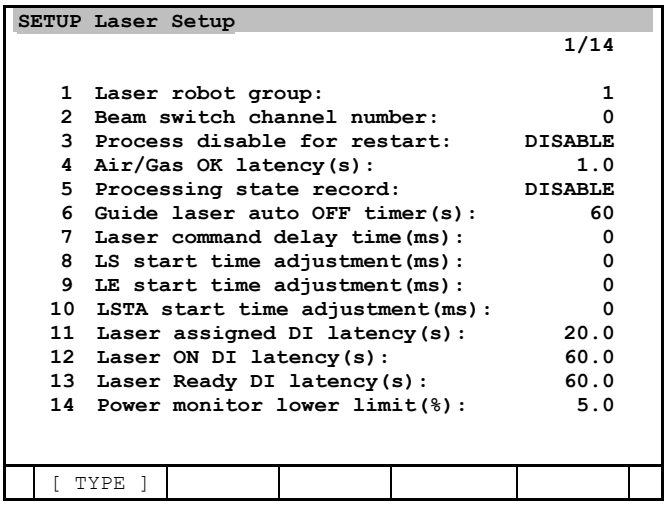
```
TIME(ms), POWER(W), BOTTOM_POWER(W), FREQUENCY(Hz), DUTY(%), ACTUAL_POWER(W), PROCESS_SPEED, FEED_SPEED
8, 2000, 1000, 500, 100, 1999,    200.00, 0
16, 2000, 1000, 500, 100, 1998,    200.00, 0
24, 2000, 1000, 500, 100, 2000,    200.00, 0
```

Fig. 70.13 Format

Procedure 70-22 Processing state record function

Step

1 Press the [MENU] key then screen menu will be displayed.
2 Select [6 SETUP].
3 Show Menu by pressed F1 [TYPE].
4 Select “Laser Setup”.
5 Move cursor on <* DETAIL *> of “Laser Setup” and press [ENTER] key.



6 Move the cursor to the item of the processing state record, and change it to ENABLE.
7 Execute the laser processing. When the laser processing is finished, the file is output to an external device, and the item of the processing state record become DISABLE automatically.
- The file is output to the external device directory that is selected on file screen.
- The file name that is restored to the external device is shown below.

FLPRCLOGaabbcc.DT (aa: hour, bb: minute, cc: second)

70.14 ALARM

70.14.1 Trouble Shooting About This Function's Alarm

The cause and remedy of alarm/message displayed related to this function are described below.

NOTE
In the alarm code, following signs are used.
%d : decimal number
%f : floating-point number
%x : hex number
%s : character string
When the number is added after %, the number means a display width.

LSR-001 Internal error (%s)
Cause: Internal error is occurred.
Remedy: Execute Diagnostic log function [FNCT menu / Diagnostic log] before you do other operation such as power off or jogging, and get the image backup. And contact your FANUC technical representative. If you cannot execute Diagnostic log function, document the events that led to the error and get the image backup. And contact your FANUC representative.

LSR-002 Laser ENBL/DSBL state is DSBL
Cause: Laser Enable/Disable state is Disable.
Remedy: No laser command is output.
This is not an alarm, it's just a message.

LSR-003 Resume processing disabled
Cause: The laser was not output when restarting during laser processing because "Process disable for restart" on Laser Setup screen is enabled.
Remedy: This is not an alarm, it's just a message.

LSR-004 %s wait timeout
Cause: The input signal displayed in %s did not turn on.
Remedy: Check the status of the output source device.
If you can change the wait time, check if the wait time is appropriate.

LSR-005 Processing speed is too large
Cause: Processing speed of the LS instruction is too high.
Remedy: Change the processing speed to an appropriate value.

LSR-006 Received Laser Error
Cause: "Laser Error" input has turned on.
Remedy: Take measures on the laser oscillator side so that "Laser Error" input is turned off. After taking measures, reset the error.

LSR-010 Guide laser OFF automatically
Cause: The guide laser turned off automatically to save the life of guide laser.
Remedy: Adjust the "Guide laser auto OFF timer" on laser setup screen.

LSR-011 %s mismatch
Cause: The status of the input signal and output signal displayed on %s do not match.
Remedy: Check the status of the equipment that is outputting the signal.

LSR-012 %s is ON
Cause: The input signal displayed in %s did not turn off.
Remedy: Check the status of the equipment that is outputting the signal.

LSR-013 %s is OFF
Cause: The input signal displayed in %s did not turn on.
Remedy: Check the status of the equipment that is outputting the signal.

LSR-014 %s
Cause: Refer to message in string.
Remedy: Refer to remedy for message in string.

LSR-015 Power On DI is OFF
Cause: "Power On" input is off.
Remedy: Turn on "Power On" input.

LSR-016 Laser ENBL/DSBL state is ENBL
Cause: Laser Enable/Disable state is Enable.
Remedy: This is not an alarm, it's just a message.

LSR-017 Out of range (%s)
Cause: The value specified for the argument of the LS instruction is incorrect.
Remedy: Refer to the character string displayed in %s and specify a value suitable for the argument of the LS instruction.

LSR-018 Laser instruction error
Cause: Processing schedules of the laser instruction are inappropriate.
Remedy: Specify a value suitable for the argument of the laser instruction.

LSR-019 This head number is invalid
Cause: Specified laser prcocessing head number is invalid.
Remedy: Specify the correct laser prcocessing head number.

LSR-020 %s cannot ON
Cause: The output signal displayed in %s could not be turned on.
Remedy: Refer to cause code.

LSR-021 %s cannot OFF
Cause: The output signal displayed in %s could not be turned off.
Remedy: Refer to cause code.

70.15 LASER INTERFACE UNIT

70.15.1 Outline

The processing schedules (S, L, P, Q) commanded by the robot control device are transmitted to the laser interface unit via the FSSB cable, and the laser command is output from the laser interface unit to the laser oscillator as an analog output.
Laser interface unit could be mounted in Top Box of R-30iB A-cabinet or in the customer's cabinet. The cables showed in thicker line have to be connected by the customer.

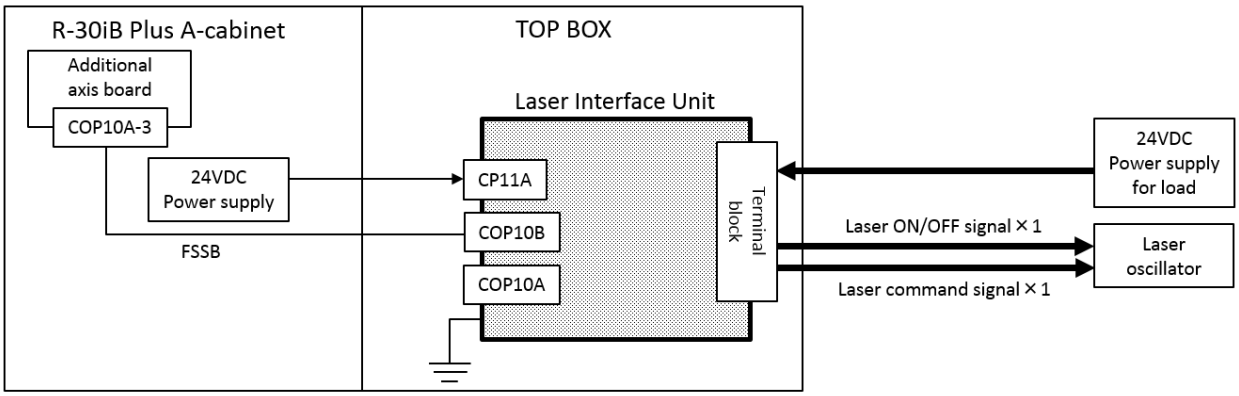


Fig. 70.15.1(a) In case Laser Interface Unit is mounted in Top Box

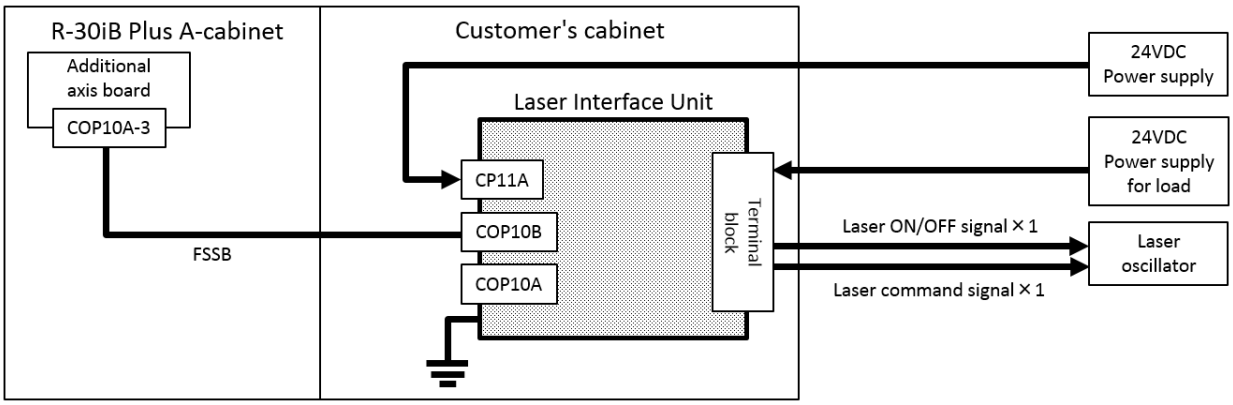


Fig. 70.15.1(b) In case Laser Interface Unit is mounted in customer's cabinet

70.15.2 System

Prepare the following hardwares.

- Hardware

Name	Specifications	Remarks
Laser Interface Unit	A05B-2626-J201	Laser Interface Unit is mounted in Top Box
Laser Interface Unit	A05B-2626-K152	Laser Interface Unit is mounted in customer's cabinet.
Additional axis board	A05B-2670-J031	
FSSB Cable 10m	A66L-6001-0026#L10R03	From Additional Axis card to Laser Interface Unit (In case Laser Interface Unit is mounted in customer's cabinet)
FSSB Cable 20m	A66L-6001-0026#L20R03	
FSSB Cable 30m	A66L-6001-0026#L30R03	

70.15.3 Hardware Specifications

70.15.3.1 Overview

- I/O signal specifications

Name	Number of points	Specification
Laser ON/OFF signal	1	Digital output - Polarity : Source type - Rated output voltage : 24VDC ± 10% - Rated output current : 0.1A

Name	Number of points	Specification
Laser command signal	1	Analog output - Voltage output : 0VDC ~ 10VDC - Current output : 0mADC ~ 25mADC - Maximum load : 10kΩ or more (voltage output) 400Ω or less (current output)

- Installation conditions

It must be satisfied with the following installation conditions, when installing the laser interface unit separately.

1 Use the units in a completely sealed cabinet.
2 Use the units under the following ambient temperature conditions:
Operation: 0℃ to 55℃
Storage and transportation: -20℃ to 80℃
3 Be sure to install the units on a vertical surface and provide a clearance of 100 mm or more both above and below the units. Do not place equipment that generates a large amount of beat below the units.

- Power supply rating

Unit	Supply voltage	Current rating	Remarks
Laser Interface Unit	24 VDC±10%	0.44A	The power consumption of the laser ON/OFF signal load current is not included.

- Heat dissipation

Unit	Heat dissipation (W)
Laser Interface Unit	11

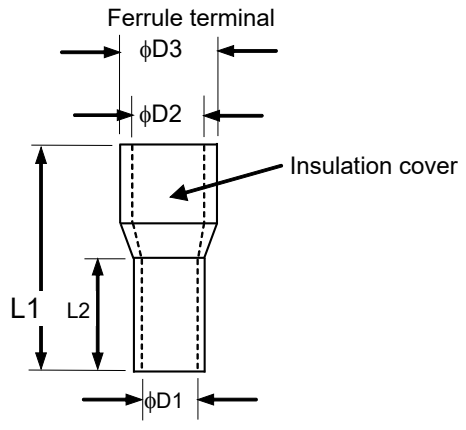
- Weight

Unit	Weight (g)
Laser Interface Unit	1000

- Applicable wire

Electric wires and ferrule terminals used for connection with a terminal block of this module should have the following dimensions:

Table 70.15.3.1 Applicable wire

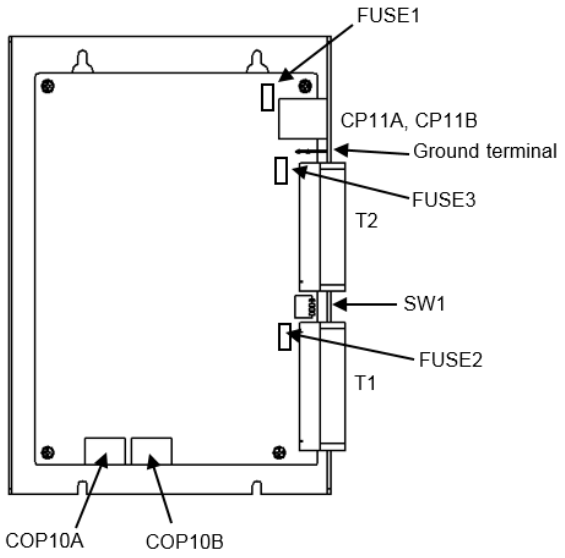
Electric wire size	 The dimensions indicated below are based on the products of Weidmuller Color., Ltd.

Cross sectional area of electric wire (mm^2)	JIS VSF KIV (mm^2)	JIS IV (mm^2)	UL1007 (AWG)	UL1015 (AWG)	Electric wire sheath stripped length (mm)	Overall length L1 (mm)	Length of metallic part L2 (mm)	Inner diameter of conductor D1 (mm)	Inner diameter of sheath D2 (mm)	Outer diameter of sheath D3 (mm)
0.5	-	-	20	-	10	14	8	1	2.6	3.1
					12	16	10			
0.75	0.5	-	18	20	10	14	8	1.2	2.8	3.3
					12	16	10			
1.0	0.75	0.9	-	18	10	14	8	1.4	3	3.5
					12	16	10			

NOTE
Use a ferrule terminal from the viewpoint of long-term reliability.

70.15.3.2 External view and dimensions

- Component names



Name	Use
CP11A	24VDC power supply input
CP11B	24VDC output
T1	Terminal block 1
T2	Terminal block 2
COP10A	Second FSSB stage
COP10B	First FSSB stage
SW1	DIP switch
FUSE1	Fuse 2A(For 24VDC Power supply)
FUSE2	Fuse 2A(For DO output)
FUSE3	Fuse 03A(For HDO0 output)

- Dimensions and Installation

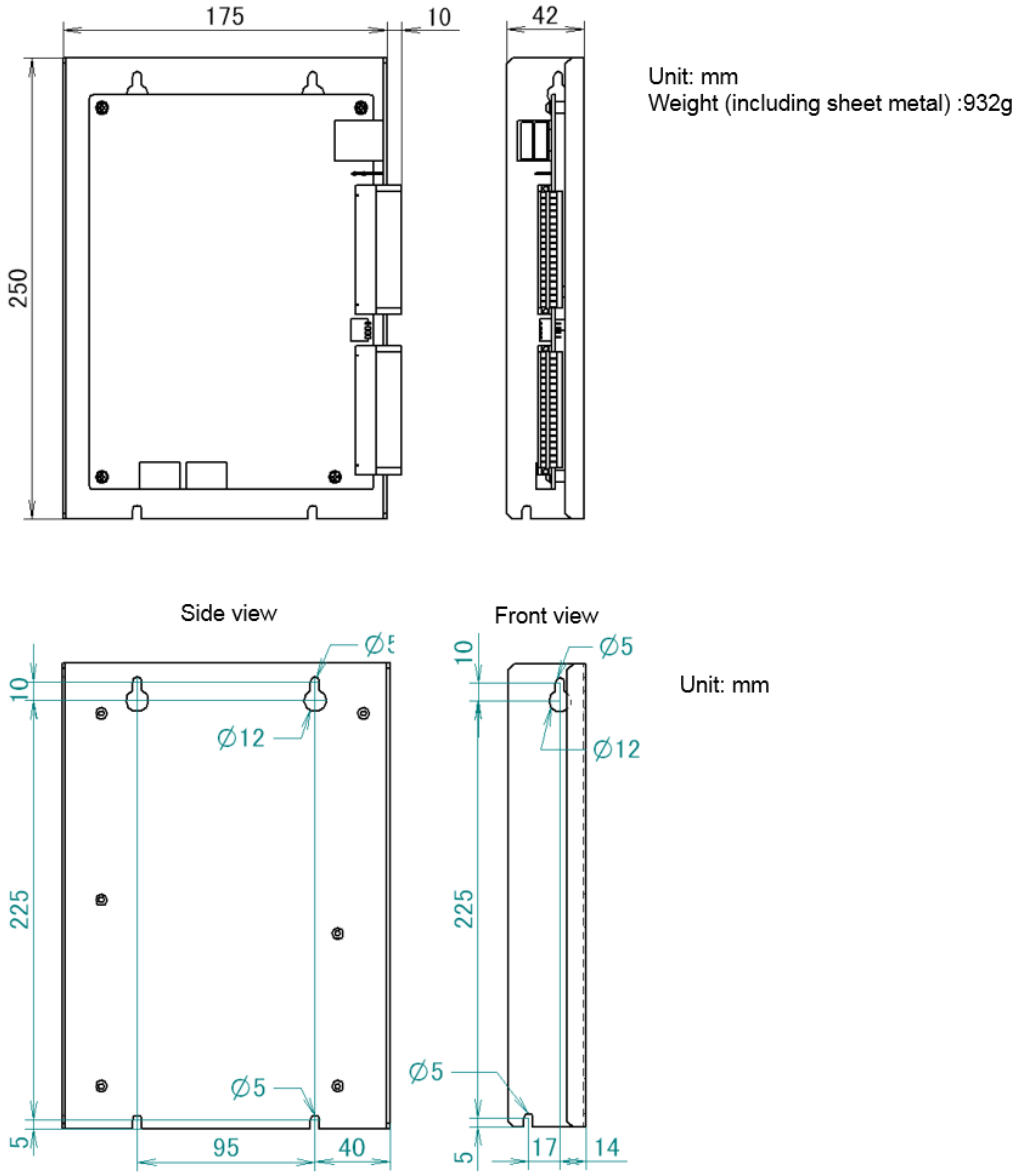


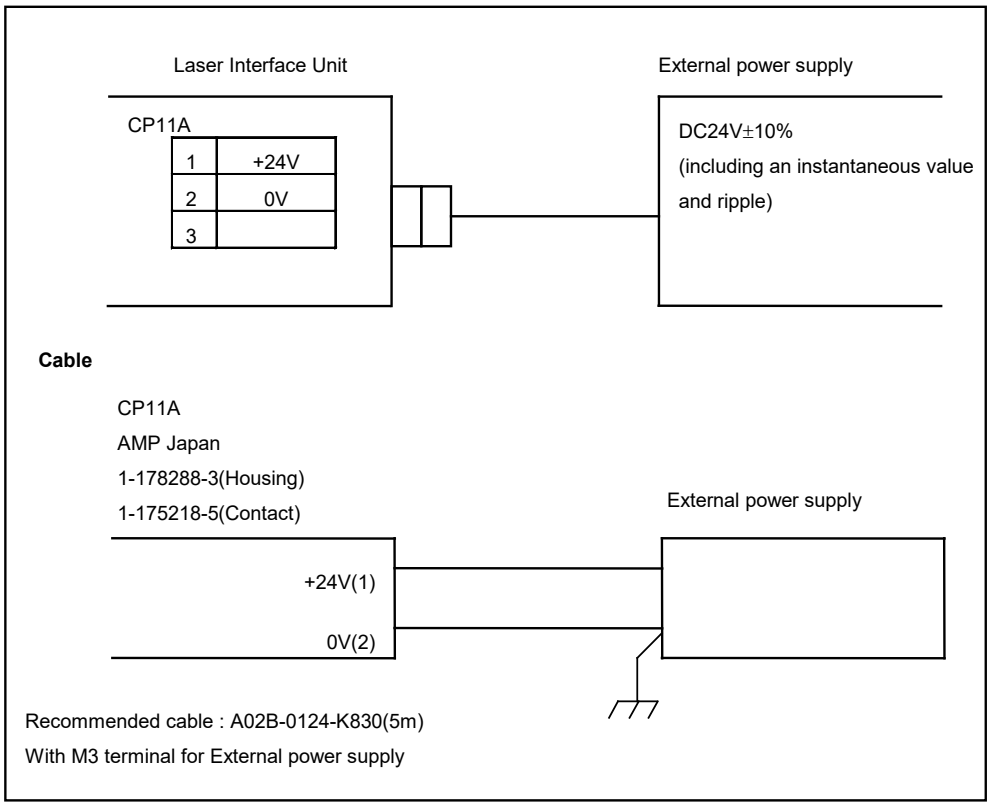
Fig. 70.15.3.2 Dimensions and Mounting holes for screwing

70.15.3.3 Connection

- Connection of power supply

When the laser interface unit is mounted in customer's cabinet, power to the laser interface unit should be supplied from an external 24V DC power supply.

It is possible to use the robot controller power supply, but please contact us as it depends on the system configuration.



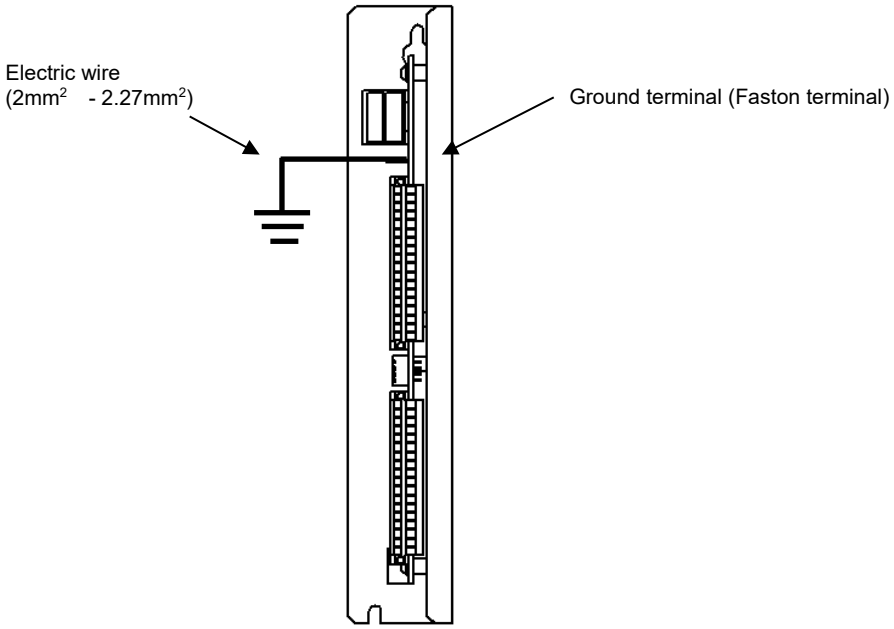
The 24VDC input to CP11A can be output at CP11B for use in branching. A cable with the same specifications as for CP11A can be connected to connector CP11B. In this case, the power supplied to CP11A should be equal to the sum of the rating of the laser interface unit and that of the units after CP11B. A current of up to 1.0 A can be supplied via CP11B.

Be sure to ground the 0V line of the power supply to the laser interface unit. In addition, keep any noise source (such as an AC power cable and contactor) away from the power line of the laser interface unit as far as possible to prevent noise from being picked up through the power line.

NOTE
The order of power on/off is as follows.
- Power-on order : Turning on the power at the same time(From 200ms before turning on robot controller to 500ms after turning on) as robot controller.
- Power-off order : Turning off the power at the same time(After 500ms before turning off robot controller) as robot controller.

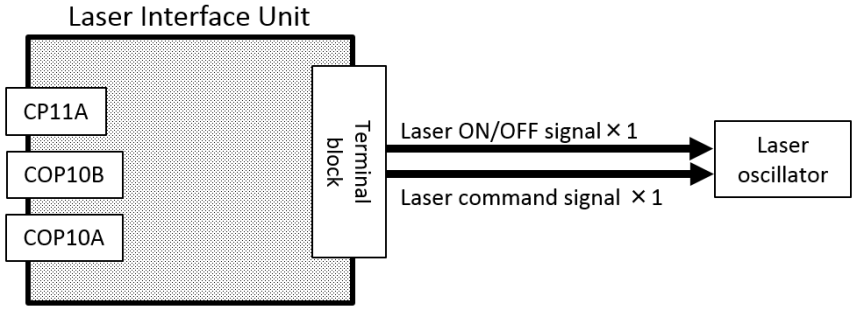
- Connection of the ground terminal

When the laser interface unit is mounted in customer's cabinet, please ground the ground terminal of the laser interface unit.
Use the Faston terminal (FANUC specification: A02B-0166-K330)



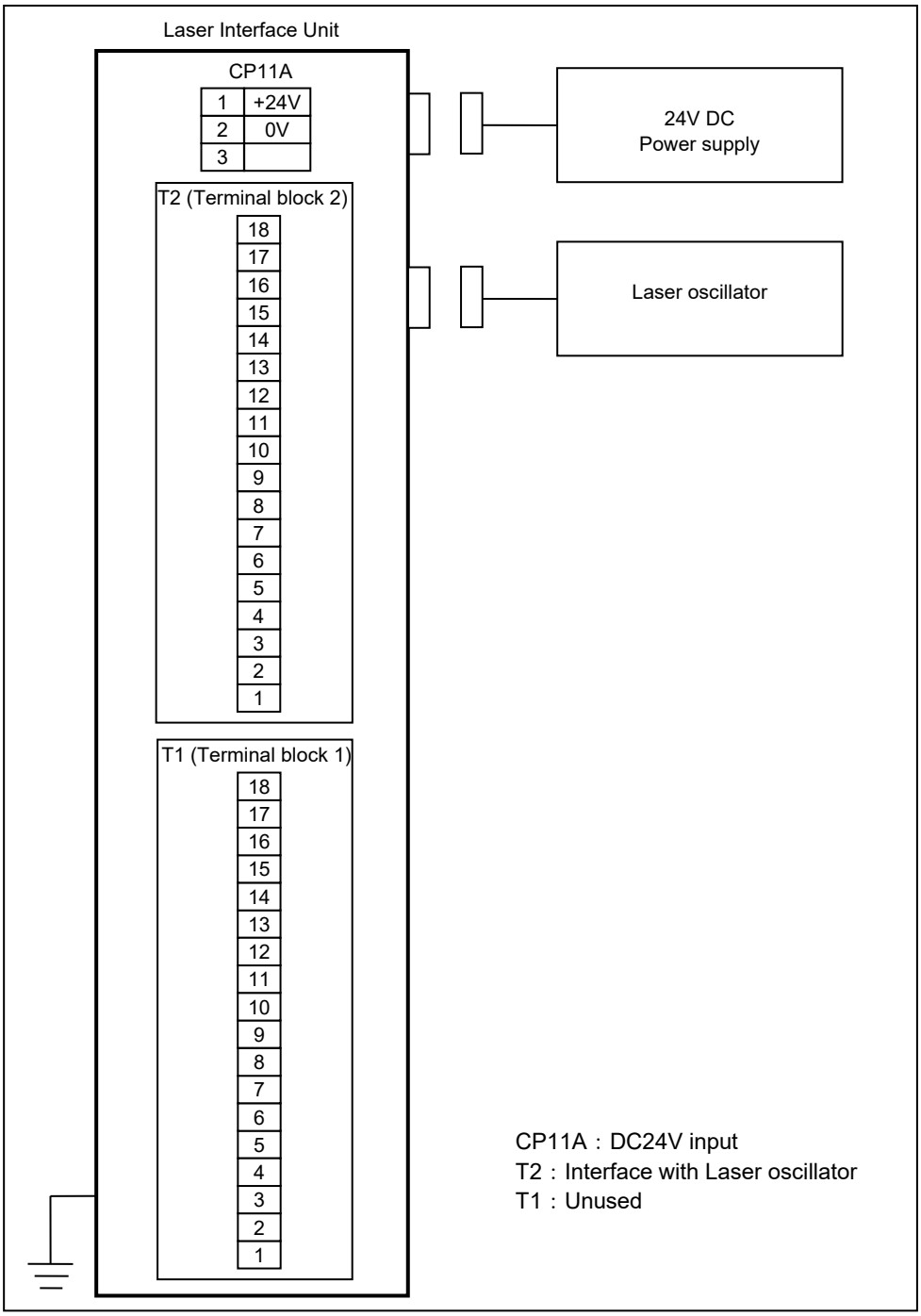
-Connection diagram

The laser interface unit and the laser oscillator are connected by two types of signals as shown in the connection diagram below.



NOTE
- Connect the required signal according to the specifications of the laser oscillator.
- Maximum allowable cable length between the terminal of the laser interface unit and Laser oscillator is 20m.

The terminal block position and pin layout of the laser interface unit are as shown in the figure below.



NOTE
Do not connect T1 (Terminal block 1), because it is unused.

-Terminal block specification

The adjacent T1 (terminal block 1) is unused, so be careful not to connect it.

Terminal block name	Terminal block specification on cable side
T2	Weidmuller BLZF3.5/18/180F

-Terminal block and Number of points

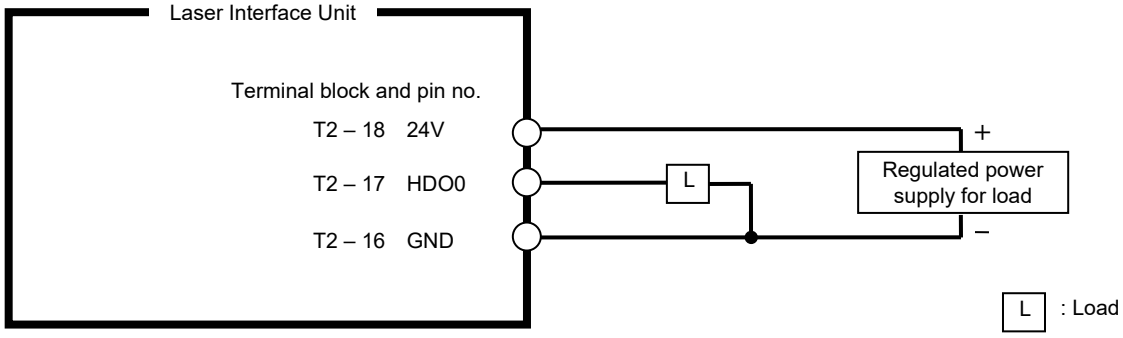
The specifications of the pin numbers of T2 (terminal block 2) are as shown in the table below.

Table 70.15.3.3 T2 (Terminal block 2)

Name	Number of points	Use	Specifications
	1		
	2		
	3		
	4		
	5		
	6		
	7		
	8		
	9		
VOUT0	10	Laser command signal	Voltage output : 0VDC to +10VDC
0V	11	GND	
IOUT0	12	Laser command signal	Current output : 0mADC to +25mADC
	13		
	14		
	15		
0V	16	GND	
HDO0	17	Laser ON/OFF signal	24V Digital output
24V	18	24VDC Power supply for HDO0	

NOTE
Please do not connect anything to the unused points.

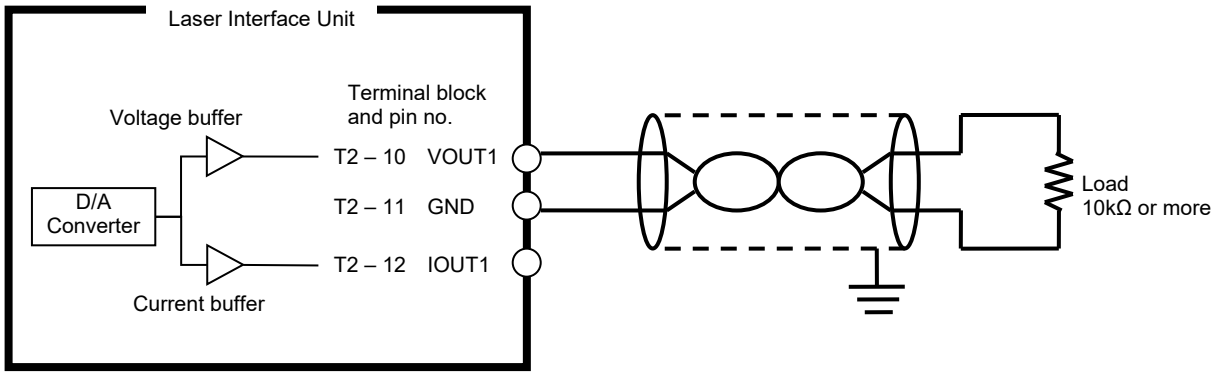
- Connection of the laser ON/OFF signal



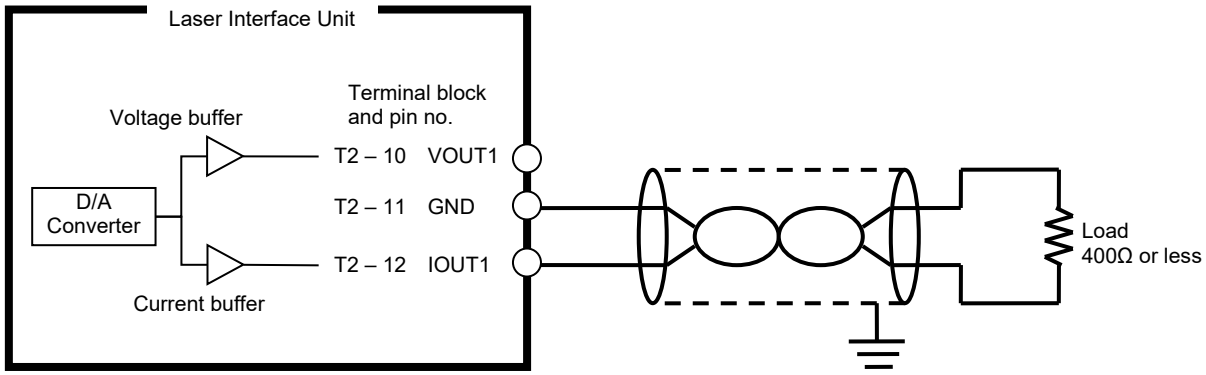
NOTE
While communication between mainboard and laser interface unit is unestablished or the moment laser interface unit is turned on and off, and also, the moment the load power is turned on and off, HDO0 is designed to turn off. However, it is not guaranteed that the signal is surely turned off. So it is required that an external safety circuit must be configured on the laser oscillator side to ensure the safety, regardless of the state of HDO0 signal.

- Connection of the laser command signal

For voltage output



For current output



NOTE

Laser command signal is designed in such a way that, At the moment when the power of laser interface unit is turned on/off, the signals (VOUT1, IOUT1) are designed to turned off (output 0V or 0mA). However, it is not guaranteed that the signals are surely turned off. So it is required that an external safety circuit must be configured on the laser oscillator side to ensure the safety, regardless of the state of the signal (VOUT1, IOUT1).

NOTE

< Laser command signal cable >

Laser command signal is an analog signal. Waveform distortion due to deterioration or noise may affect processing, so please consider the following points.

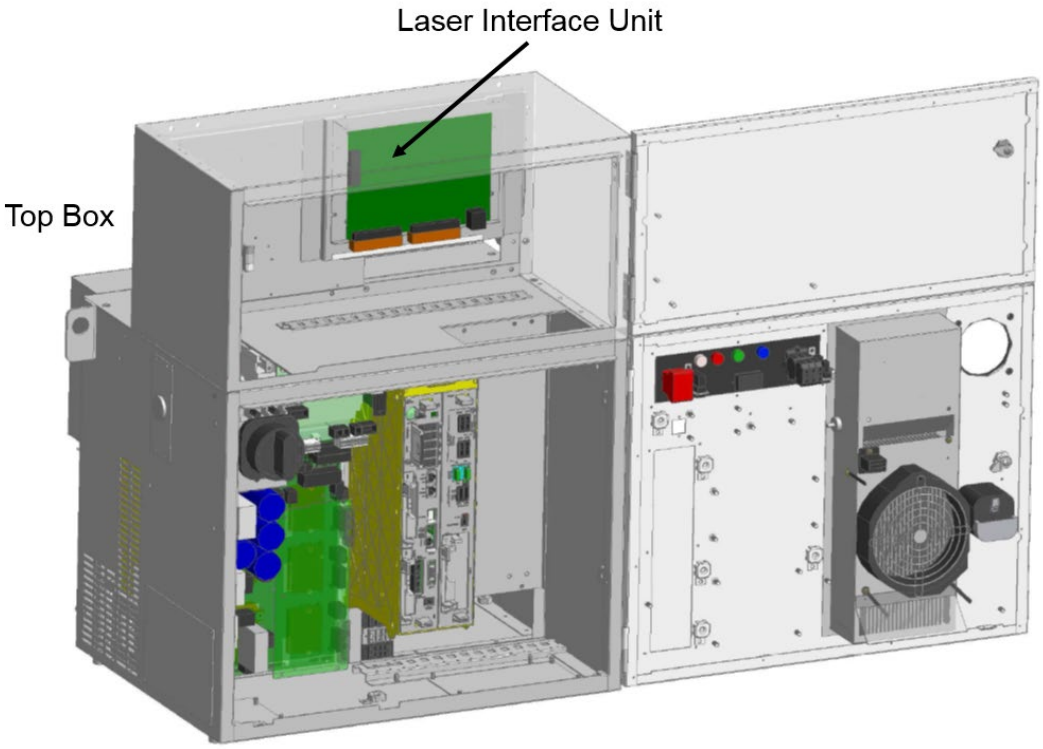
- To prevent signal quality degradation, please use a shielded twisted pair cable or coaxial cable and keep the wire length as short as possible.
- Because it may be affected by noise, please keep away from noise sources such as motor power lines. And ground the cable shield on the laser oscillator side.
- If you are concerned about signal quality deterioration or noise, please measure the waveform on the oscillator side and confirm that there are no problems.

-Setting of SW1

Setting is unnecessary, because it is unused.

70.15.3.4 Mounting position in the Robot controller

Laser interface unit is mounted inside the top box.



70.15.4 Laser Command Signal Analog Output

Set “Maximum power” and “Dead zone area” on the laser initial setup screen, and define the relationship between power S and laser command signal analog output (VOUT0, IOUT).

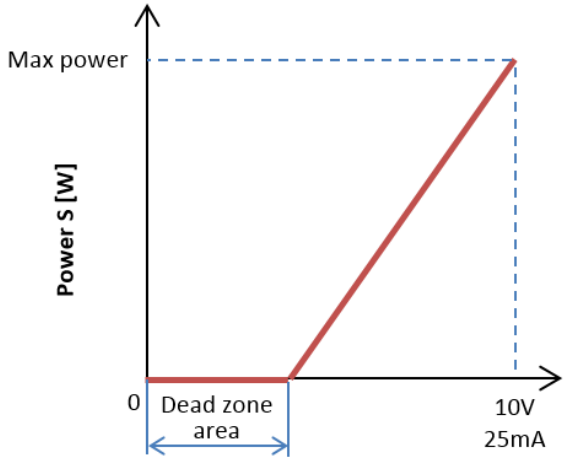


Fig. 70.15.4(a) Power S vs Laser command signal analog output

When PULSE is specified for “Analog output type” on the laser initial setup screen, the analog output (VOUT0, IOUT) of the laser command signal becomes a pulse.

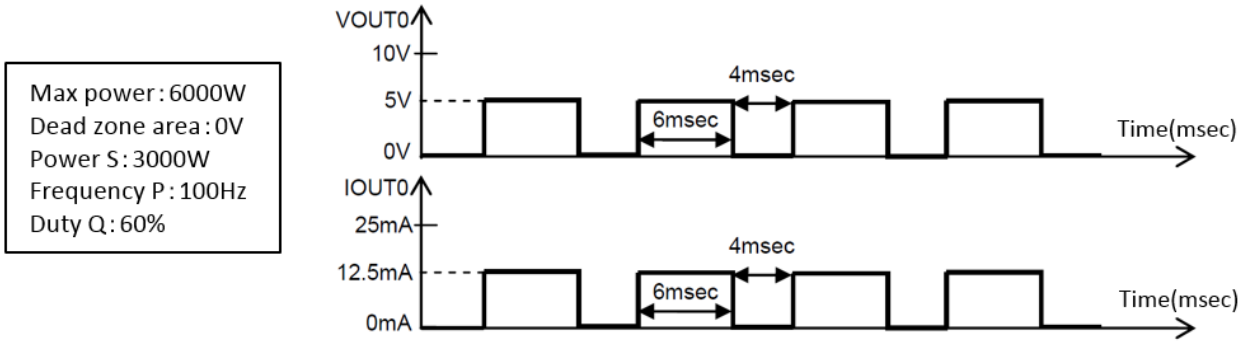


Fig. 70.15.4(b) Analog output type is PULSE

If CONT. is specified for "Analog output type" on the laser initial setup screen, the analog output (VOUT0, IOUT) of the laser command signal will be continuous. If you want the laser to be a pulse wave, combine the analog output of the laser command signal with the digital output of the laser ON/OFF signal.

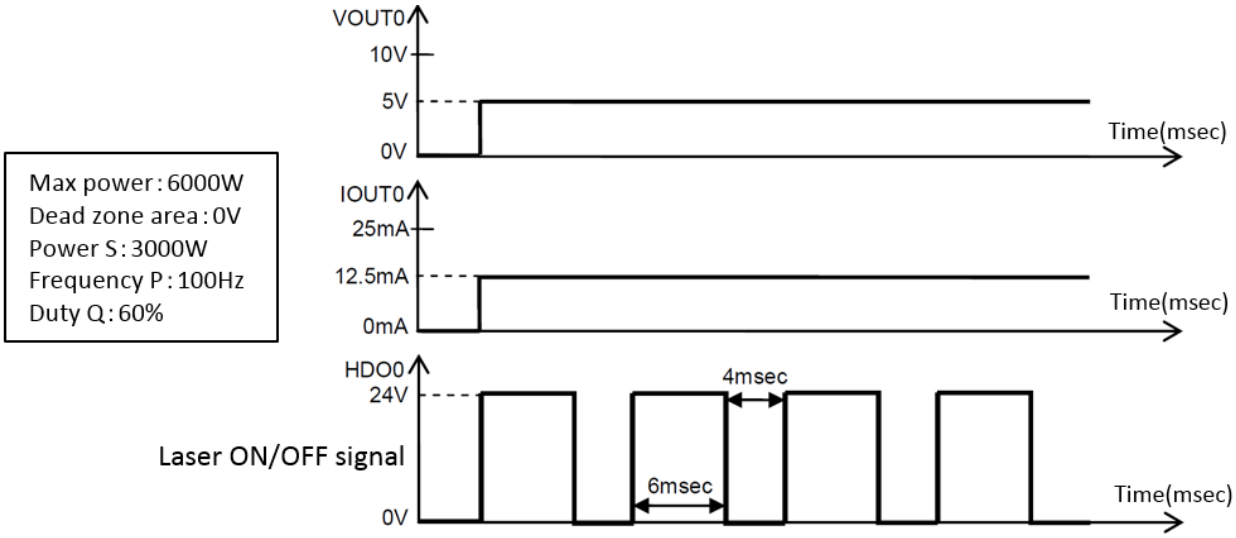
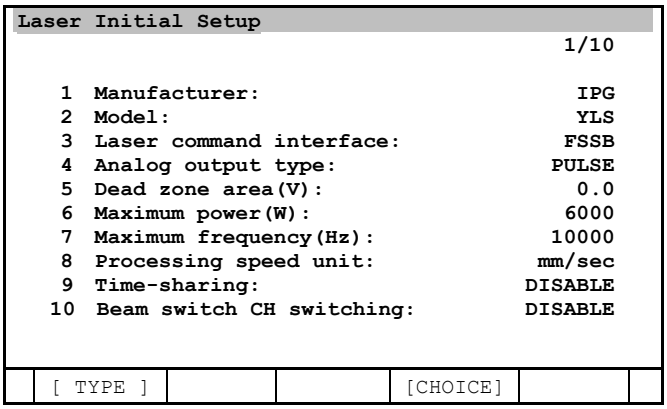


Fig. 70.15.4(c) Analog output type is CONT.

Procedure 70-23 Display laser initial setup screen

Step

1 Perform controlled start.
2 After controlled start, press the [MENU] key. Controlled start menu will be displayed.
3 Select [0 – NEXT --]. Select [4 Laser Initial Setup].



71 SERVO TOOL FUNCTION

Similar to the TORQUE LIMIT FUNCTION (Chapter 27), Servo Tool function can limit the maximum motor torque as desired. Servo Tool function can perform various settings compared to the TORQUE LIMIT FUNCTION, and in addition, some of the necessary settings are automated, simplifies and improves programming for servo grippers or other servo-driven axes.
For example, as shown in Fig.71(a), conventionally, you need to teach the hand position according to the size of the workpiece. By using Servo Tool function, the torque applied to the hand can be limited, and it is possible to handle multiple workpieces by using the closed hand position as the teaching position. In addition, the floating mechanism of the tool can be omitted or simplified because it can absorb variations in the size of the workpiece. Servo Tool function allows fine adjustment of the gripping position and torque limit value after gripping the workpiece. Therefore, it is possible to grip workpieces of different size at the same teaching position without crushing them.

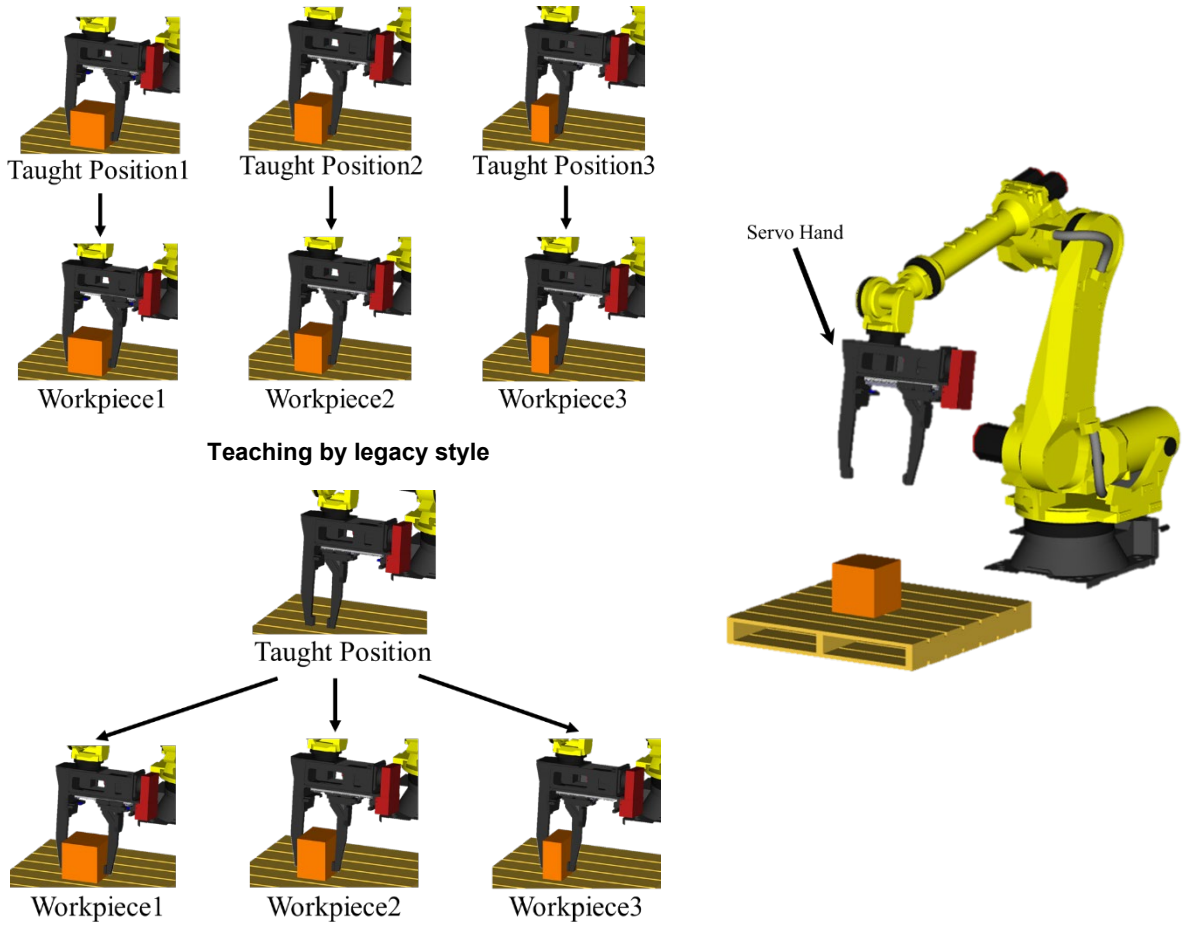


Fig. 71 (a) Example of using the Servo Tool Function

This option provides a schedule-based user interface that allows the user to specify torque limits for specified auxiliary or robot axes. The Servo Tool TP instruction allows the user to call particular servo tool schedule during TP program execution.

71.1 SERVO TOOL FUNCTION

Servo Tool function works as follows:

- Provides a schedule-based user interface to allow users to assign torque limits for a specified auxiliary or robot axis.
- Although the deviation (error) between the commanded position and the actual position becomes large for the axis which the torque limit is specified, the error check level suitable for the system is automatically adjusted. You do not need to change and set them yourself.
- The controller remembers the position where the specified axis stops when it contacts with the workpiece (contact position). The destination position can be automatically updated to contact position (Auto Position Update).
- In addition to the torque limit before contact, the torque limit after contact can be set. By combining with the above Auto Position Update, the torque after contact can be increased without crushing the part.

⚠ WARNING

1 Enabling torque limit on robot axes could cause damage to equipment or personnel due to fall down by the gravity or deviation from its original trajectory.
2 Regular error checking will be reduced or disabled when Servo Tool is enabled. Verify the path with extra caution.

71.2 SETUP THE SERVO TOOL FUNCTION

The servo tool schedules are specified on the [SETUP Servo Tool] menu, which consists of the following two menus.

- List menu
- Detail menu

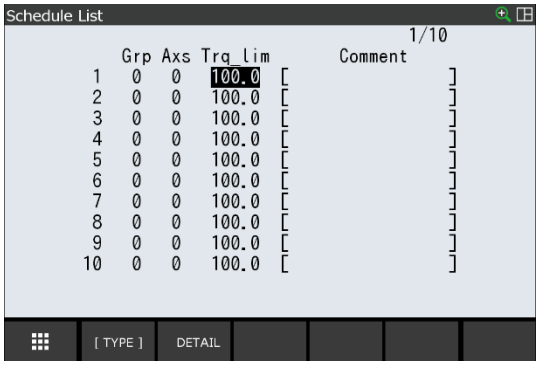
Up to 10 schedules can usually be specified for the Servo Tool function.

Procedure 71-1 Setting up Servo Tool

Step

1. Press the [MENU] key. The screen menu will be displayed.
2. Select SETUP.
3. Press F1, [TYPE]. Select Servo Tool from the screen switching menu.
4. Move the cursor to the schedule you want to set and press F2, DETAIL. The detail menu will be displayed.
5. Set the item as desired. To return to the list of schedule, press F2, LIST.

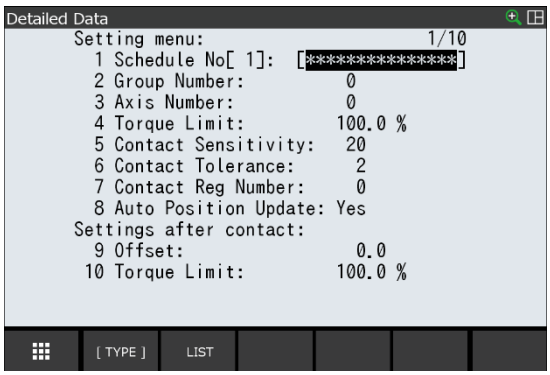
List menu



F2, DETAIL

F2, LIST

Detail menu



Setting items

The following data can be specified on the detail menu.

Table 71.2 (a) Setting items of Servo Tool function

Item		Default	Description
Setting menu:	Schedule No[ ]		Servo Tool schedule number. Pressing the [ENTER] key with the cursor on line 1 enables entering a comment.
	Group Number	0	This item identifies the group number of the axes to controlled.
	Axis Number	0	This item identifies the axis number whose torque you want to specify. With some exceptions, you cannot select the robot axis (See section 1.4 for details).
	Torque Limit	100%	This item identifies the percentage torque limit for the specified axis. This value is the ratio to the torque corresponding to the maximum current in the specification of the amplifier. Range : 0% to 100%
	Contact Sensitivity	20	This item identifies the duration that the servo axis remains idle in system ITP time. One ITP is usually 4msec or 8msec(You can confirm the temporary ITP by $SCR.$ITP_TIME). When the servo tool axis remains idle over this time period, the contact between the servo axis and the part will be determined, and the next action will be executed.
	Contact Tolerance	2	This item identifies the idle condition of the servo tool axis in encoder counts. For example, if this item is 2, then this servo axis is idle as long as encoder reading for this servo axis maintains within 2 counts each time.
	Contact Reg Number	0	This item identifies the numeric register number for contact conditions after SERVOTOOL[ ] TP instruction is used. The meaning of each Contact Register value is as follows: 0: Servo Tool has not started 1: axis starts moving after SERVOTOOL TP instruction 2: axis is in contact with the part (box) and next motion line will start -1: axis as not contacted the part (box) but destination position reached.
	Auto Position Update	Yes	If set to Yes, position data of the specified axis between the SERVOTOOL[ ] and SERVOTOOL END instructions is updated automatically to the contact position. You do not need to use an offset command or PR[ ] value when teaching the Servo Tool program. If set to No, the destination position is not updated even when contact is made, and the original taught point 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 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 end effector and part. If Auto Position Update is set to NO, this item will be disabled and the original taught point will be used. UNIT : Rotary axis : [rad] Linear axis : [mm]
	Torque Limit	100%	This item specifies the new torque limit to be applied the workpiece after contact is made. This value can be used to control the tightness between end effector and part. Range : 0% to 100%

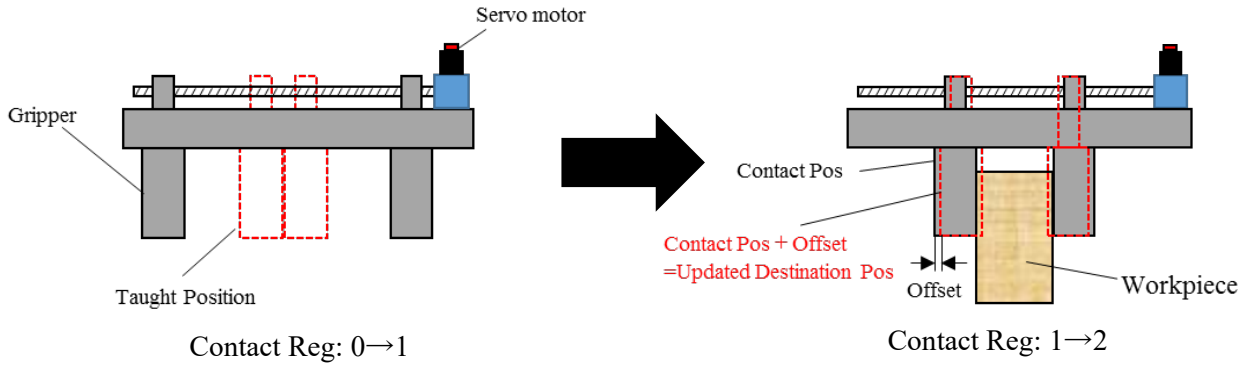


Fig. 71.2(a) Example of contact / non-contact between servo hand and workpiece (Auto Position Update is set to Yes): With workpiece

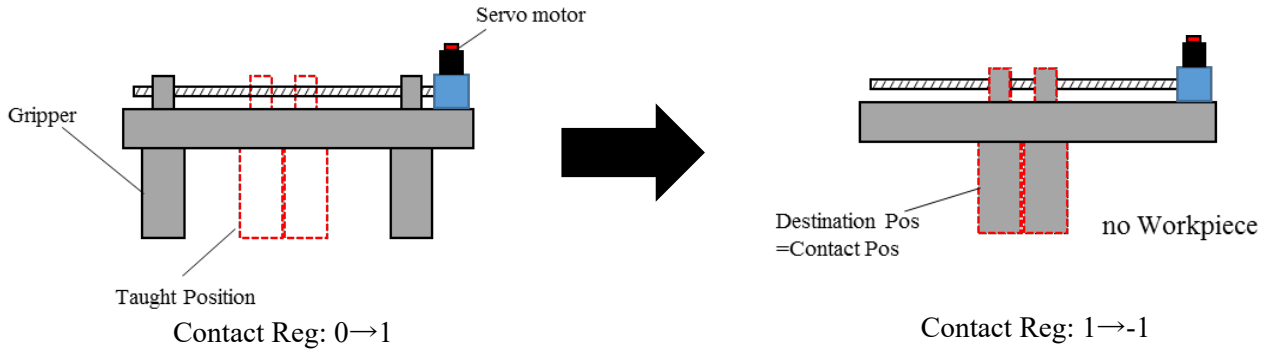


Fig. 71.2(b) Example of contact / non-contact between servo hand and workpiece (Auto Position Update is set to Yes): No workpiece

71.3 PROGRAM INSTRUCTION

There are the following two TP instructions for the Servo Tool function.
- SERVOTOOL[schedule number]
- SERVOTOOL END / SERVOTOOL END[schedule number]

71.3.1 SERVOTOOL[schedule number]

The Servo Tool function of the specified schedule is enabled by the SERVOTOOL[schedule number] instruction. The following occurs:

- The torque limit of the specified axis becomes effective.
- The error check for the Servo Tool function is enabled.
- When contact is made (for example, at gripper contact with the part), quick stop is used at the moment, and the deceleration stop process of normal operation is omitted. This prevents hesitation that could affect cycle time.
- If Auto Position Update is enabled, contact position of the specific axis is updated to destination position. If the taught point is reached without contact, the taught point is updated as the destination position. This contact position (destination position) is applied until SERVOTOOL END is executed.

Torque limits and position offsets can be kept in the subsequent motion lines until SERVOTOOL END is reached.

71.3.2 SERVOTOOL END / SERVOTOOL END[schedule number]

Servo Tool function is disabled by SERVOTOOL END instruction. The following occurs:

- 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.
- SERVOTOOL END[schedule number] can be used to disable Servo Tool per the schedule.

NOTE
If multiple Servo Tool enabled axes are configured in the same group, even though SERVOTOOL END[schedule number] may be disabled independently, all motion in the same group needs to be stopped at the same time; therefore all axes in the same group may still be affected.

71.3.3 TP Program Example

Figure 71.3.3 (a) shows an example of gripping a workpiece with the servo hand using the Servo Tool function. The details of Figure 71.3.3 (a) are described below.
- The robot and servo hand are operated at the same time.
- Use the SERVOTOOL[1] on the 4th line to enable the torque limit of the servo hand. Also, enable the automatic update of contact position.
- The taught position (P[3]) of the servo hand on the 5 the line is in the state where the servo hand is completely closed, but the gripping position (P’[3]) of the work is updated to the destination position of the servo hand by the Servo Tool function.
- The grip position of the servo hand is kept until the servo tool is executed, and the robot moves while gripping the work.

```
:
:
4 : SERVOTOOL[1]
5 : L P[3] 1500mm/s    FINE
6 : J P[4] 100%    CNT100
7 :SERVOTOOL END
```

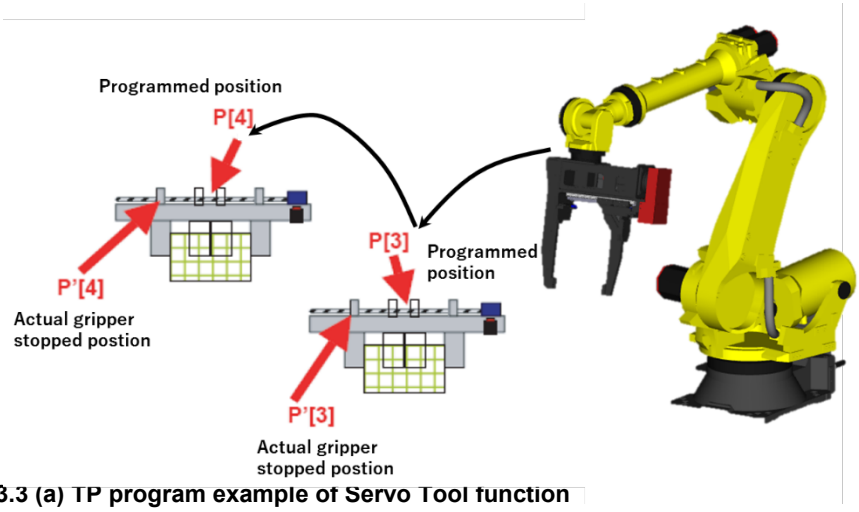


Fig. 71.3.3 (a) TP program example of Servo Tool function

71.4 LIMITATIONS

Servo tool has the following limitations:

- When used in Single Step mode or when the program is paused, the ServoTool[ ] TP instruction will only activate the torque limit servo setting. No position update or offset will 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 ServoTool[] and ServoTool End, no ServoTool schedule will be used.
- Multi-arm multi-tasking application is not supported.
- If DCS includes Servo Tool enabled axes, a DCS FB_CMP alarm may occur when or after Servo Tool is used. If needed, this can be resolved by setting $DCS_CFG.$EXCLUDE[motion group] to -1. However, please refer to DCS manual for details.

NOTE
If $DCS_CFG.$EXCLUDE[motion group] is set to -1, Position/Speed Check function cannot be used in this motion group.

⚠ WARNING
Motor torque is low and regular error checking is disabled when a program is paused while Servo Tool is enabled.

In the following software versions, Servo Tool function cannot be enabled for robot axes, with exceptions. When SERVOTOOL instruction is executed, the alarm “INTP-663 **(%s^4, %d^5)** Torque Limit cannot be used” will be posted.

Software versions:

- 7DC1/33 (V8.10P/33) or later
- 7DC2/36 (V8.20P/36) or later
- 7DC3/63 (V8.30P/63) or later
- 7DF1/37 (V9.10P/37) or later
- 7DF3/21 (V9.30P/21) or later
- 7DF5/21 (V9.40P/21) or later

Exceptions:

- J1 axis of a robot whose J1 axis is a rail.
- J3 axis of SCARA robot with 6kg or less payload capacity.
- Final axis of an opener robot for painting application.

71.5 CAUTIONS

- It is possible to use ServoTool for robot axes, but please do NOT enable this function for any robot axes which may be fallen down by the gravity.
- If auto position update is disabled, contact position will not be destination position even if it touches a work piece. When an axis is stopped by Torque Limit, the external force is equal to the limited motor torque, and the distance between current(contact) position and destination position is considered as position error. If the torque limit after contact is increased the axis moves to the destination with larger torque than the torque before contact, therefore you may break Servo Hand or a work piece.

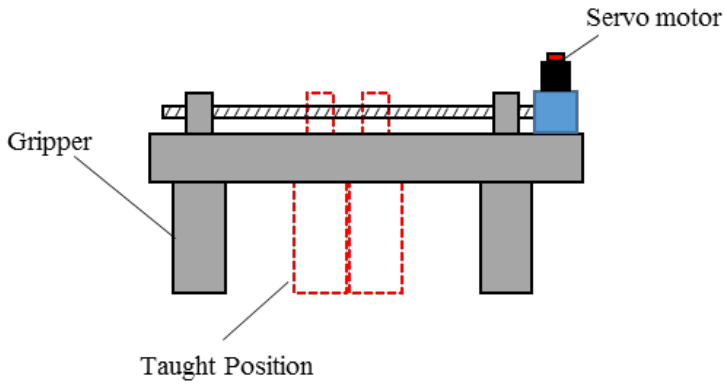


Fig. 71.5(a) Taught Position of the servo hand

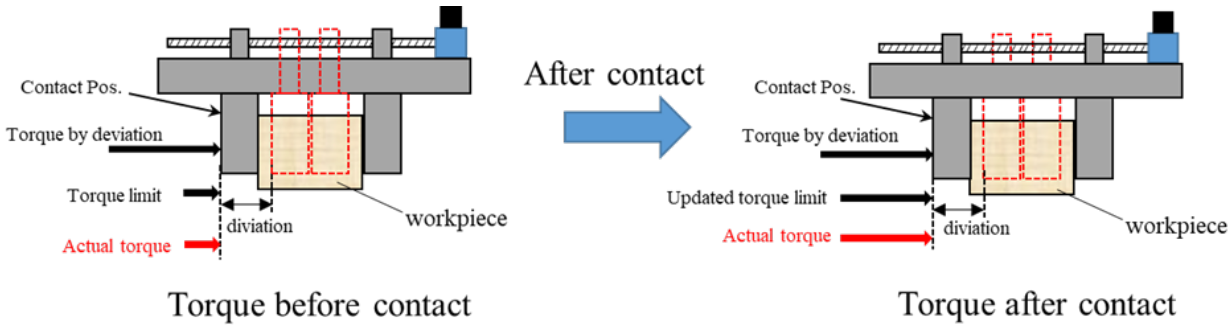


Fig. 71.5(b) Auto position update disabled
If the torque limit after contact is large, the torque for the error amount is applied to the workpiece

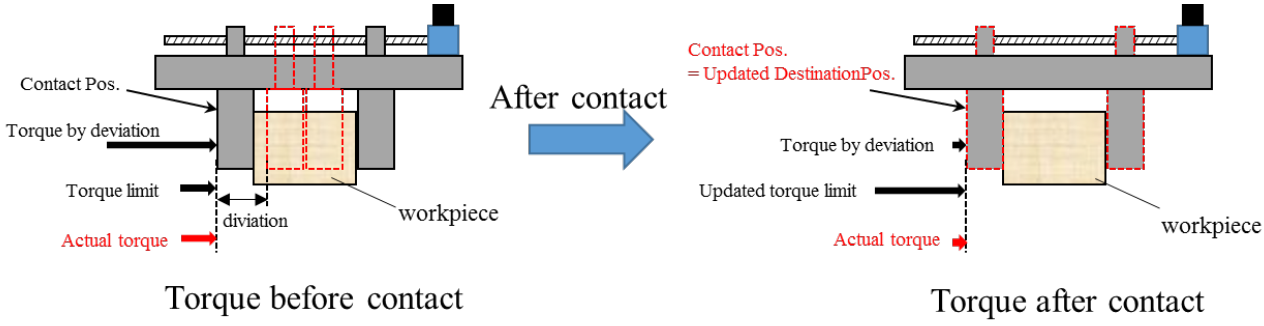


Fig. 71.5(c) Auto position update enabled.
Even if the torque limit after contact is large, the actual torque is small.

72 AUTO BACKWARD EXIT

72.1 OVERVIEW

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.

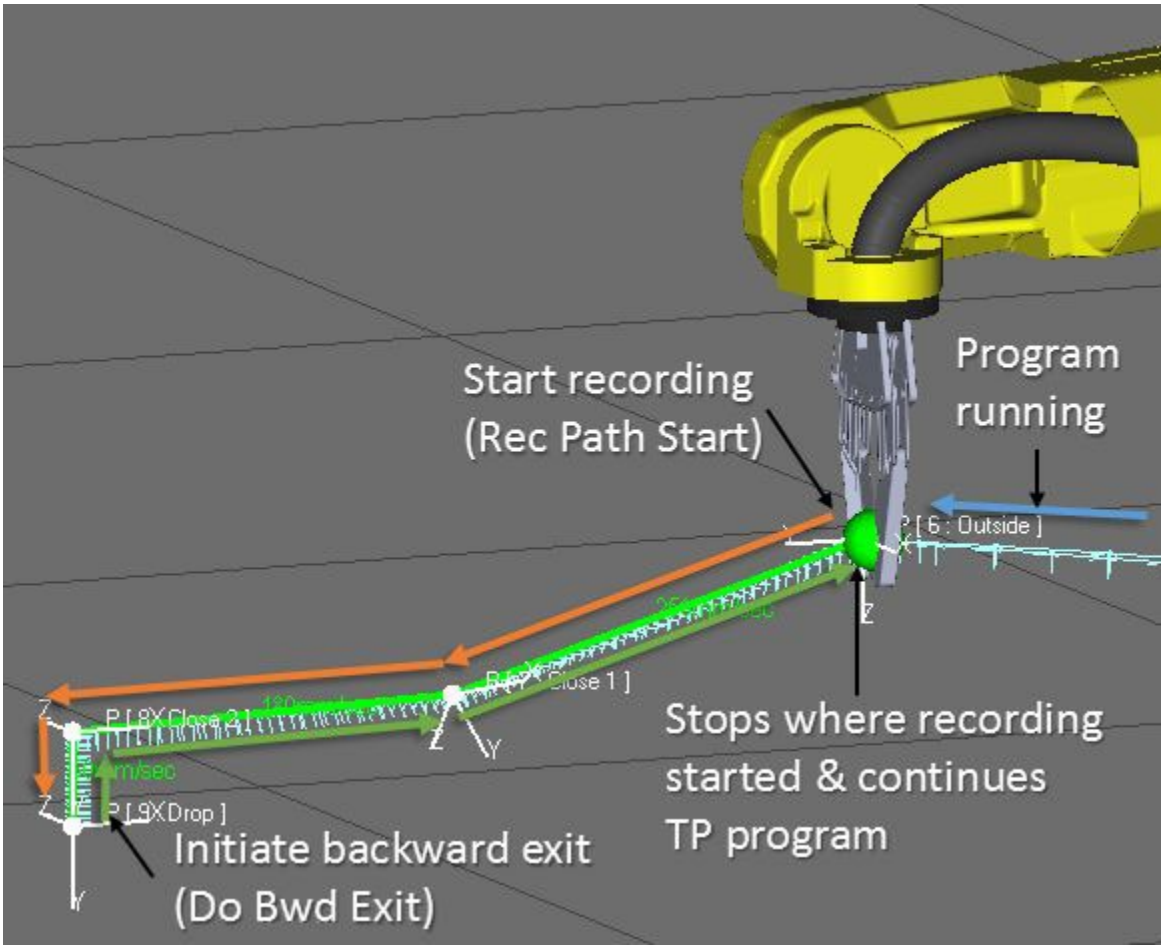


Fig. 72.1 (a) Auto Backward Exit Application

72.2 SETUP AND OPERATIONS

72.2.1 Specification Overview

Auto Backward Exit can be enabled at Controlled start. Setup items can be set at both Cold Start and Controlled start. System Variables are also used to configure Auto Backward Exit.

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 1-1 .

Table 72.2.1 (a) Auto Backward Exit Macros

MACRO	DESCRIPTION
Rec Path Start	Starts recording of robot path.
Rec Path End	Stops recording if currently recording. Recorded path is no longer available.
Rec Path Pause	Pauses recording of the robot path until resumed or ended.
Rec Path Resume	Resumes paused path recording.
Do Bwd Exit	Starts exit motion backward along the recorded path.

72.2.2 Setup

Auto Backward Exit can be set up and configured via the Auto Bwd Exit menu entry at Controlled start. Refer to Table 72.2.2 (a) 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 72.2.2 (a) : 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: (1-12000mm)	Defines the maximum path recording length. Longer distances use more memory.
Recording Interval: (1-100mm)	Defines the point recording interval during robot motion recording. Shorter distances use more memory.
Cold Start or Controlled Start Setup Items	
Auto Exit Speed: (1-250mm/sec)	Defines the robot speed for backward exit motion up to 250 millimeters per second.
Jog Away Warning: (1-1000mm)	A prompt asks to continue if the robot has been jogged or moved away from the last recorded point more than this distance. The current position is compared to the last recorded point when 'Do Bwd Exit' is run.
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 = not enabled)	A digital output can be turned on when 'do bwd exit' is executing for this group.

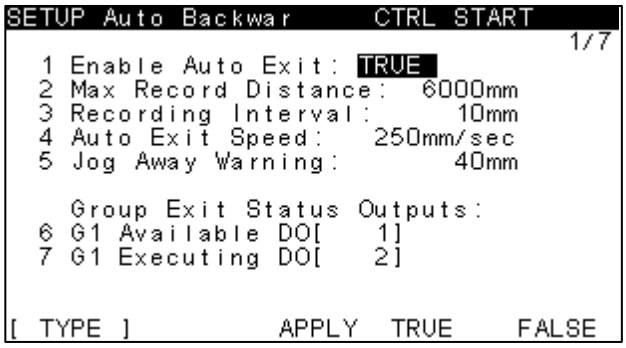


Fig. 72.2.2 (a) Auto Backward Exit Setup Screen

Operation Setting Up Auto Backward Exit

Procedure

1 Perform a Controlled start. (Turn the power ON with “Prev” and “Next” keys pressed, then select “3 Controlled start”)
2 Press the [MENU] key, select 0 until you can select Auto Bwd Exit, and press the [ENTER] key.
3 Press F4, TRUE to set Auto Backward Exit to TRUE. Set all other items as necessary. Refer to Table 1–2 .
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.

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 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.

72.2.3 System Variables

Alternatively, you can directly configure Auto Bwd Exit using the system variables shown in Table 72.2.3 (a). Two advanced settings— recording options and exit options can only be changed via system variables.

Table 72.2.3 (a) Auto Backward Exit System Variables

SYSTEM VARIABLE	DESCRIPTION
$AUTOEX_CONF.$ENABLE	Setting to 1 enables the feature and allocates recording memory. Setting to 0 disables feature and deallocates memory. (Changes require a Controlled Start.)
$AUTOEX_CONF.$MAX_DIST	Defines the maximum path recording length up to 12,000mm. Longer distances use more memory. (Changes require control start.)
$AUTOEX_CONF.$REC_INTERVL	Defines the point recording interval during robot motion recording. Can be set from 1 to 100mm. (Changes require control start.)

SYSTEM VARIABLE	DESCRIPTION
$AUTOEX_CONF.$REC_OPTION	Recording Options (This is a mask, so you must add values together for multiple options. Changes require control start.) 1 — Use interpolated position. 2 — Records jog moves while recording a path forward. (During backward exit jog moves are always recorded.) 4 — Allows Do Bwd Exit to be used after Rec Path End if the robot has been jogged, but not after program execution. 8 — Allows Do Bwd Exit to be used after Rec Path End or after executing another program.
$AUTOEX_CONF.$EXIT_SPEED	Defines the robot speed for backward exit motion up to 250mm/sec.
$AUTOEX_CONF.$MAXTOL	Exit path resume tolerance (same as jog away warning). A prompt asks to continue if the robot has been jogged or moved away from the last recorded point more than this number in millimeters. The current position is compared to the last recorded point when 'Do Bwd Exit' is run.
$AUTOEX_CONF.$POS_REG_IDX	Specifies the position register to use if AUTOEX_CONF.$EXIT_OPTION is set to use a position register instead of a point.
$AUTOEX_CONF.$EXIT_OPTION	Exit Options (This is a mask, so you must add values together for multiple options. Changes require control start.) 1 — Do not use Wjnt (wrist joint) motion option during backward exit motion. 4 — Use a position register instead of a point during backward motion
$AUTOEX_CONF.$MACRO_START	Allows the 5 auto exit macro's table numbers to be re-allocated
$AUTOEX_PORT.$AVAIL_DO_T[Grp_n]	Defines the type of output to use for the Exit Available status output for each motion group.
$AUTOEX_PORT.$AVAIL_DO_I[Grp_n]	Defines the index to use for the Exit Available status output for each motion group. (0 = not enabled)
$AUTOEX_PORT.$EXEC_DO_T[Grp_n]	Defines the type of output to use for the Exit Executing status output for each motion group.
$AUTOEX_PORT.$EXEC_DO_I[Grp_n]	Defines the index to use for the Exit Executing status output for each motion group. (0 = not enabled)

72.2.4 Programming

The following is a simple example of how the feature can be used. In this example, the robot starts recording at P[6] before moving to P[7]. At the end of the program, 'Do Bwd Exit' macro will move the robot all the way back to P[6] where recording started. The program would then continue to P[1].

```
  1:L  P[6:Outside] 500mm/sec CNT50
  2:
  3:  !Start recording
  4:  Rec Path Start
  5:L  P[7:Close 1] 250mm/sec CNT25
  6:L  P[8:Close 2] 180mm/sec CNT10
  7:L  P[9:Drop] 50mm/sec FINE
  8:  CALL DROP_PART
  9:  !Initiate backward exit
 10:  Do Bwd Exit
 11:J  P[1:Home] 100% FINE
```

Fig. 72.2.4 (a) Example Program

Auto Backward Exit can be implemented into a wide variety of error recovery solutions due to the capabilities of macro programs.

73 CNC CUSTOM MACRO VARIABLE ACCESS

73.1 OVERVIEW

CNC Custom Macro Variable Access is a function to Read/Write Custom Macro Variable of CNC. CNCs and a Robot controller is connected with Ethernet cable. A Robot can be connected several CNCs.

If this software option is added, you can use Custom Macro Variable Access instruction. You can operate following operation, if use this instruction.

- It can be accessed by specifying machine tools and its path number.
- It can access local variable, common variable and system variable.
- There are cases that part of system variables are restricted by CNC. If restricted system variable is accessed, an alarm occurs.
- Instruction of "Read Custom Macro Variable" can read Custom macro variable and write it to Numeric register.
- Instruction of "Write Custom Macro Variable" can write arbitrary value to Custom macro variable.

This function support following CNC models that can use embedded Ethernet.

Series 30*i*/31*i*/32*i*-MODEL B Plus
Series 30*i*/31*i*/32*i*/35*i*-MODEL B
Series 30*i*/31*i*/32*i*-MODEL A
Series 16*i*/18*i*/20*i*/21*i*-MODEL B (7df5(V9.40P)/17 and later)
Series 0*i*-MODEL F Plus
Series 0*i*-MODEL F
Series 0*i*-MODEL D/0*i* Mate- MODEL D
Power Motion *i*-MODEL A

This function has the following limitations.

- This function cannot execute multiple instructions at the same time.

NOTE

1 When interlocking a machine tool and a robot using this function, check the specifications of the machine tool and examine the safety and functional aspects.
2 Some custom macro variables may already be used by the machine tool. Check which custom macro variable can be used for this function.
3 Some machine tools may not be equipped with the chuck open/close commands required for automation. Some machine tool builders may provide optional functions necessary for robotization.
4 Normally automatic operation of machine tool stops when the manual door is opened. Some machine tool builders may offer optional functions necessary for robotization.
5 If you have any questions about machine tools, please contact the machine tool builders.

73.2 REQUIRED OPTION

This function required following robot software option.

- CNC Custom Macro Variable Access (A05B-2600-S533)

CNC Custom Macro Variable Access is supported from 7df5 (V9.40P)/15.

This function required the following option of CNC.

Series 30*i*/31*i*/32*i*/35*i* - MODEL B
Series 30*i*/31*i*/32*i* - MODEL A
- Custom macro (A02B-03xx-J873)

Series 16*i*/18*i*/20*i*/21*i* -MODEL B
- Custom macro B (A02B-02xx-J873)

Power Motion *i* - MODEL A
- Included in BASIC Option 1 (A02B-0334-R702)

Series 30*i*/31*i*/32*i* - MODEL B Plus
Series 0*i* - MODEL F Plus
Series 0*i* - MODEL F
Series 0*i* - MODEL D/0*i* Mate- MODEL D
- Option is not required.

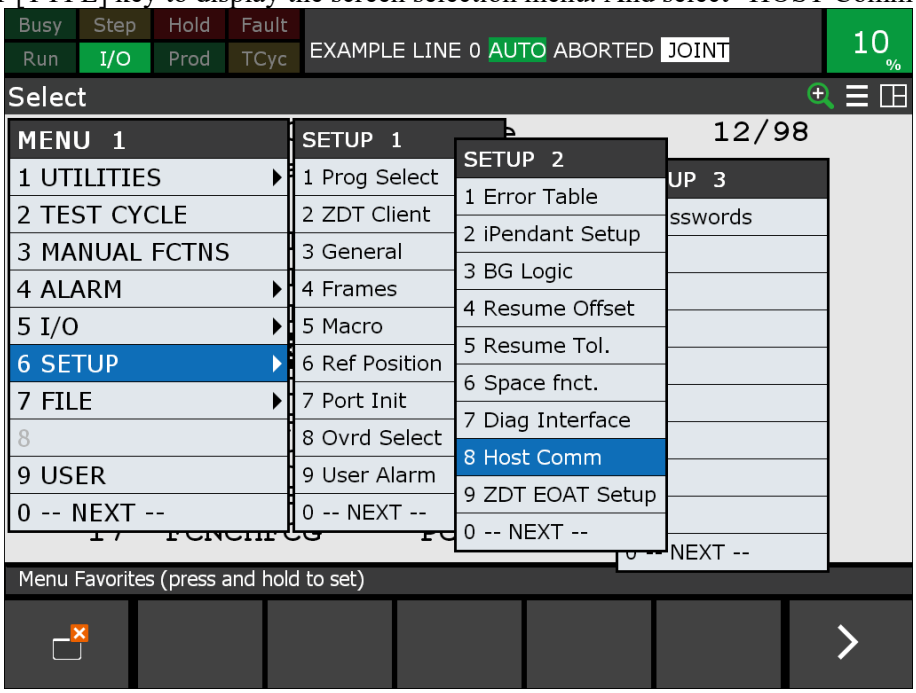
73.3 ROBOT SETUP

This section explains the settings required to use CNC Custom Macro Variable Access.
Procedure of setup is the following.

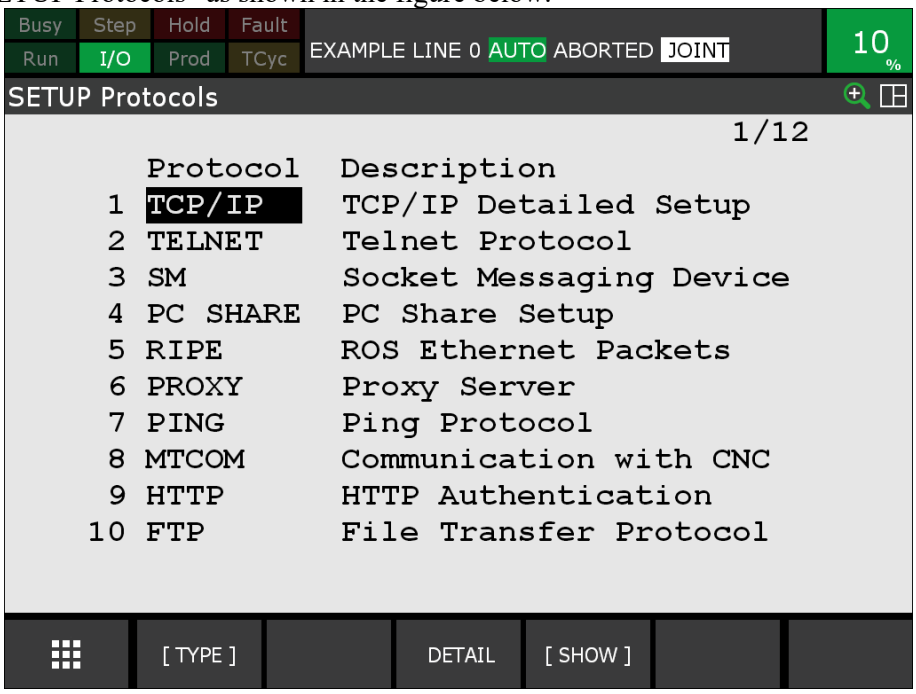
Procedure 73-1 Setup

Step

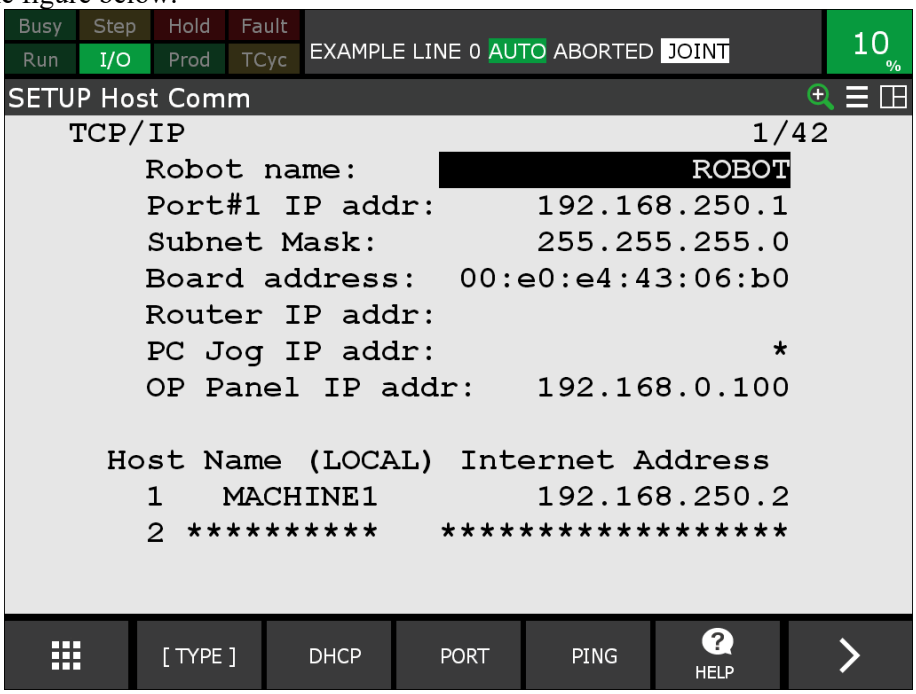
1 Insert Ethernet cable into port that you want to use.
2 Press the [MENU] key to display the screen menu.
3 Select “6 SETUP”.
4 Press the F1 [TYPE] key to display the screen selection menu. And select “HOST Comm”.



5 Display “SETUP Protocols” as shown in the figure below.



6 Move the cursor to “TCP/IP” and press the F3 [DETAIL] key. Display “SETUP Host Comm” as shown in the figure below.

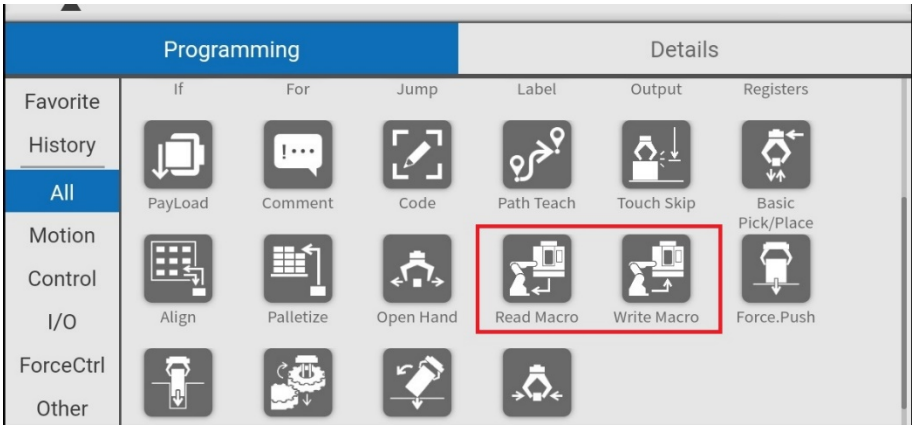


7 Press the F3 [PORT] key to switch port number that you insert the Ethernet cable.
8 Setup IP address, Subnet Mask, Router IP address.
9 Setup Host name and IP address of CNC.

Finish robot settings.

73.3.1 IF USE TABLET TP

Add icon of Read Macro and icon of Write Macro on Programming tab of Editor in Tablet UI.



Refer to “FANUC Robot series Tablet UI (B-84274EN)” for the operation of Editor in Tablet TP.

Read Custom Macro Variable

This instruction can get value of Custom macro variable that be held specified CNC and Path. And it can write the value to Numeric Register specified.

Setup items of Details tab are as follows.

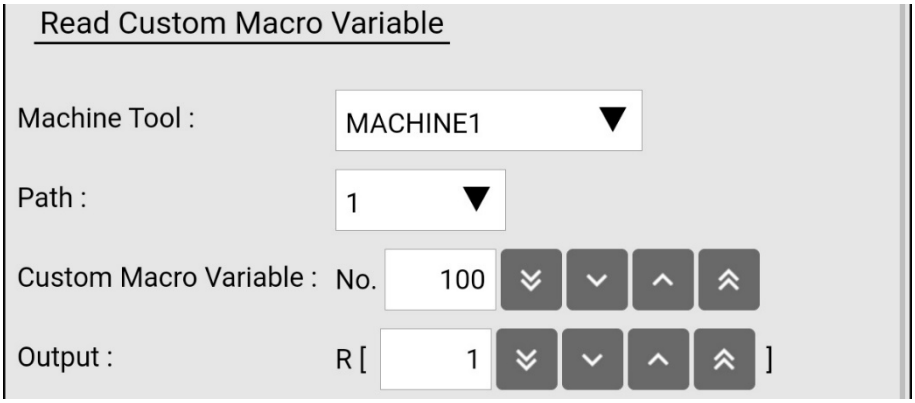
- Machine Tool
  - ➢ Set host name of machine tool to access. You can select host name set in “Host Comm” screen.

- Path
  - ➢ Set number of path to access. You can select 1 to 15 of path number.

- Custom Macro Variable
  - ➢ Set number of custom macro variable to read.

- Output
  - ➢ Set number of numeric register to write the value that get from custom macro variable.
  - ➢ If value of custom macro variable is <null>, write “0” to the numeric register.

NOTE
If value of custom macro variable cannot represented by single-precision floating point numbers is read, Rounding will occur.



Write Custom Macro Variable

This instruction can write the value to custom macro variable that be held specified CNC and Path.

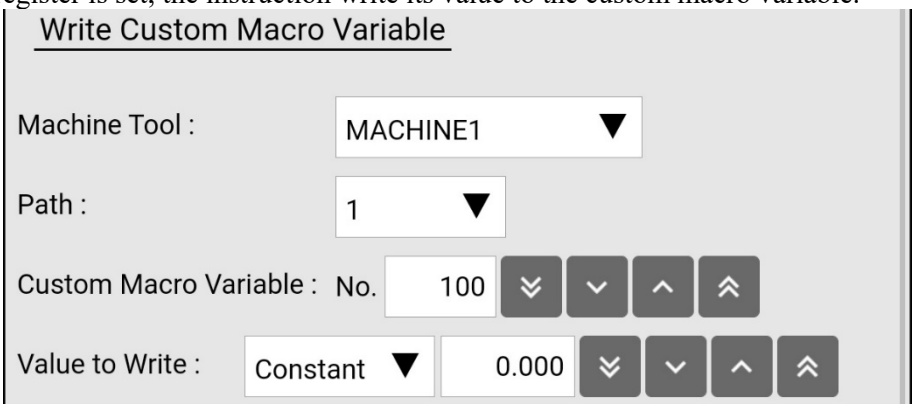
Setup items of Details tab are as follows.

- Machine Tool
  - Set host name of machine tool to access. You can select host name set in “Host Comm” screen.

- Path
  - Set number of path to access. You can select 1 to 15 of path number.

- Custom Macro Variable
  - Set number of the custom macro variable to write.

- Value to Write
  - Set the value to write to the custom macro variable.
  - Constant and register can be set.
  - If a constant is set, the instruction write the set constant to the custom macro variable.
  - If a register is set, the instruction write its value to the custom macro variable.



73.3.2 IF USE iPendant

Add karel program to read custom macro variable and write custom macro variable. If you use editor of *i*Pendant, can be used by calling out the karel programs with call instruction.

Read Custom Macro Variable

This instruction can get value of Custom macro variable that be held specified CNC and Path. And it can write the value to Numeric Register specified.
It can be used by calling out the “RD_CNC_MACRO” of karel program with call instruction.

The explanation of the arguments are as follows.

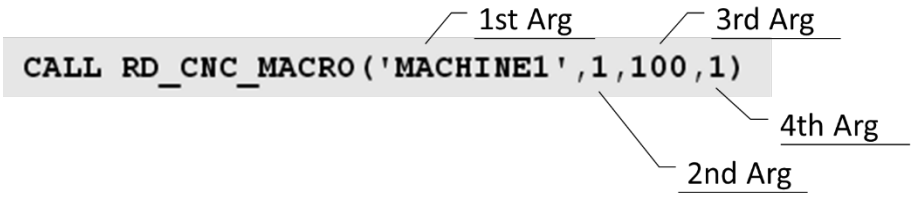
- First argument
  - Set host name of machine tool to access. You can select host name set in “Host Comm” screen.

- Second argument
  - Set number of path to access. You can select 1 to 15 of path number.

- Third argument
  - Set number of custom macro variable to read.

- Fourth argument
  - Set number of numeric register to write the value that get from custom macro variable.
  - If value of custom macro variable is <null>, write “0” to the numeric register.

NOTE
If value of custom macro variable cannot represented by single-precision floating point numbers is read, Rounding will occur.



Write Custom Macro Variable

This instruction can write the value to custom macro variable that be held specified CNC and Path.

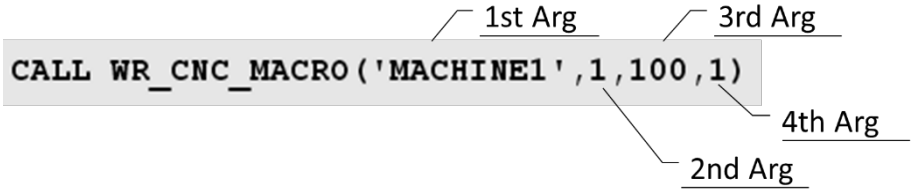
The explanation of the arguments are as follows.

- First argument
  - Set host name of machine tool to access. You can select host name set in “Host Comm” screen.

- Second argument
  - Set number of path to access. You can select 1 to 15 of path number.

- Third argument
  - Set number of the custom macro variable to write.

- Fourth argument
  - Set the value to write to the custom macro variable.
  - Constant and register can be set.
  - If a constant is set, the instruction write the set constant to the custom macro variable.
  - If a register is set, the instruction write its value to the custom macro variable.



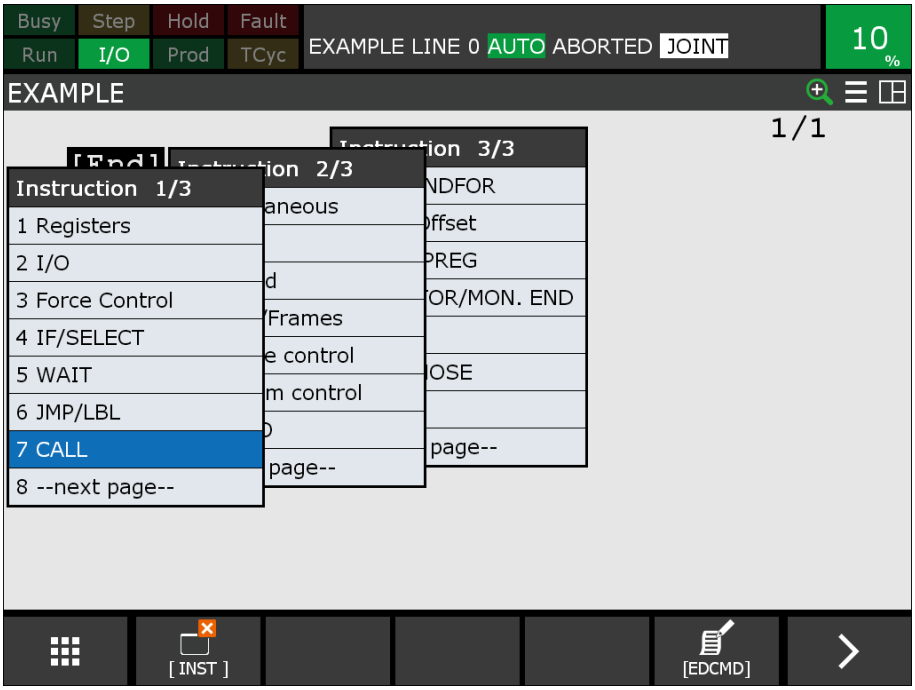
Procedure to use instruction of Custom Macro Variable Access

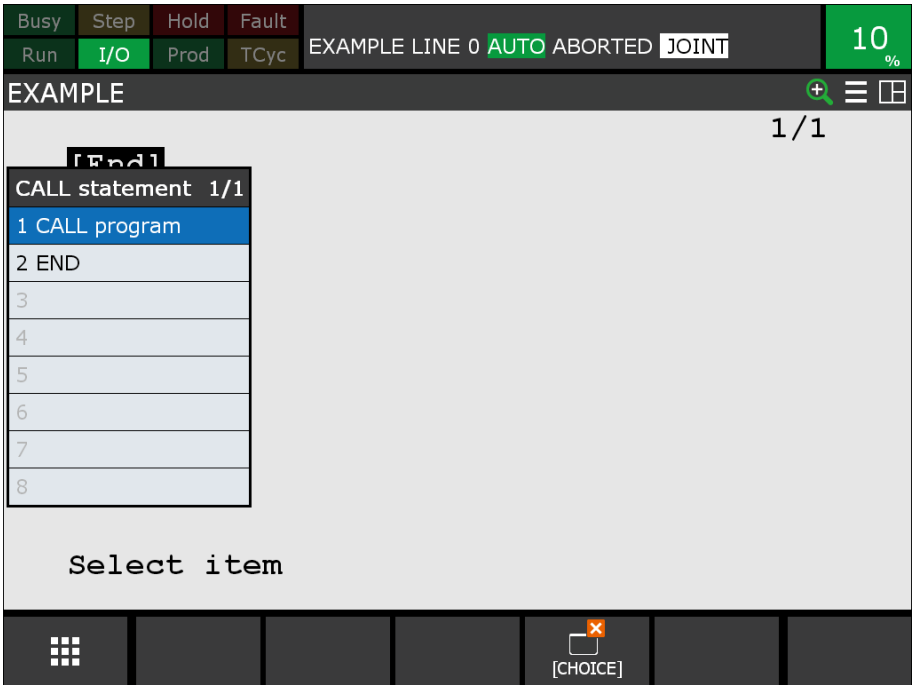
Explain procedure to use instruction of Custom Macro Variable Access

Procedure 73-2 use instruction of Custom Macro Variable Access

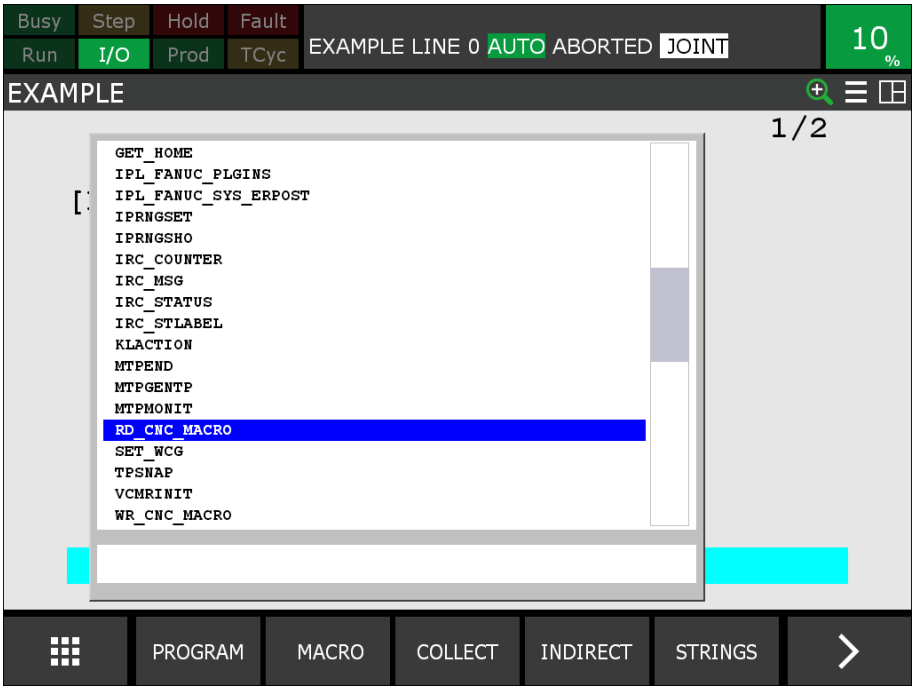
Step

1 Press the F1 [INST] key. And then select the “CALL” and the “CALL program”.

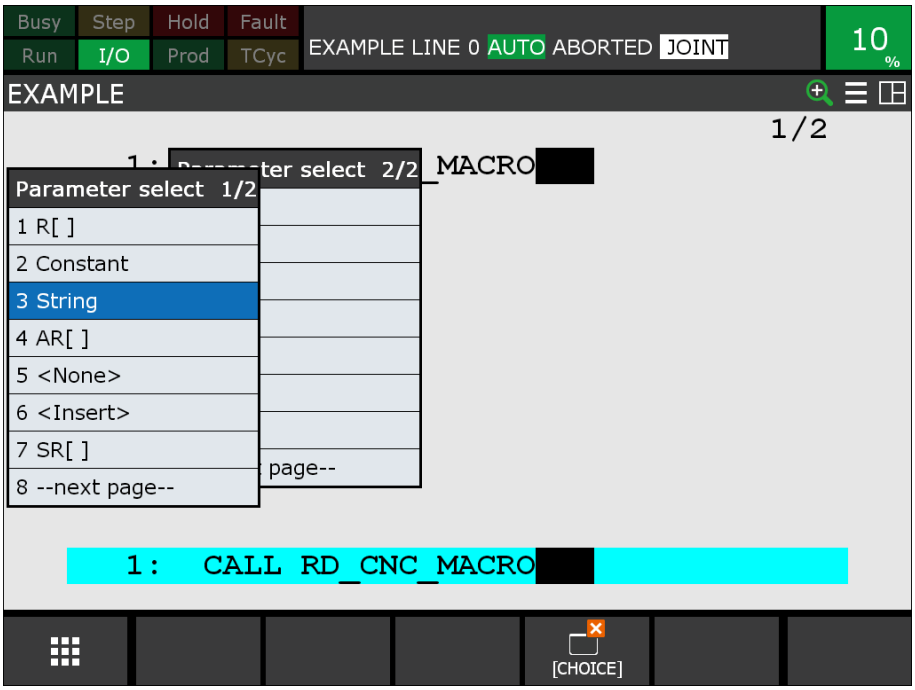




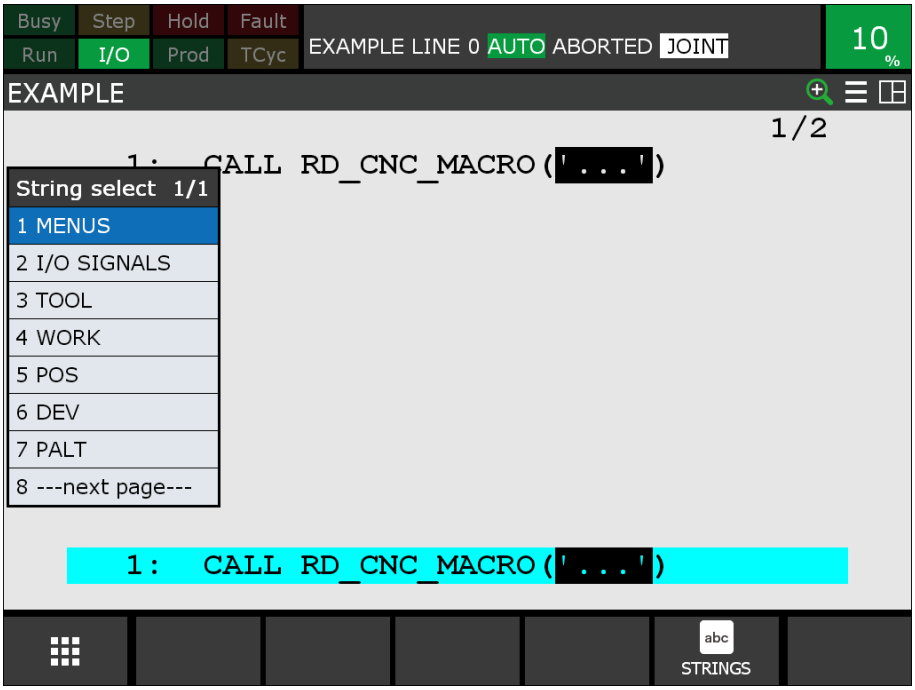
2 Press F3 [COLLECT] key and select the “RD_CNC_MACRO” from List of karel programs.



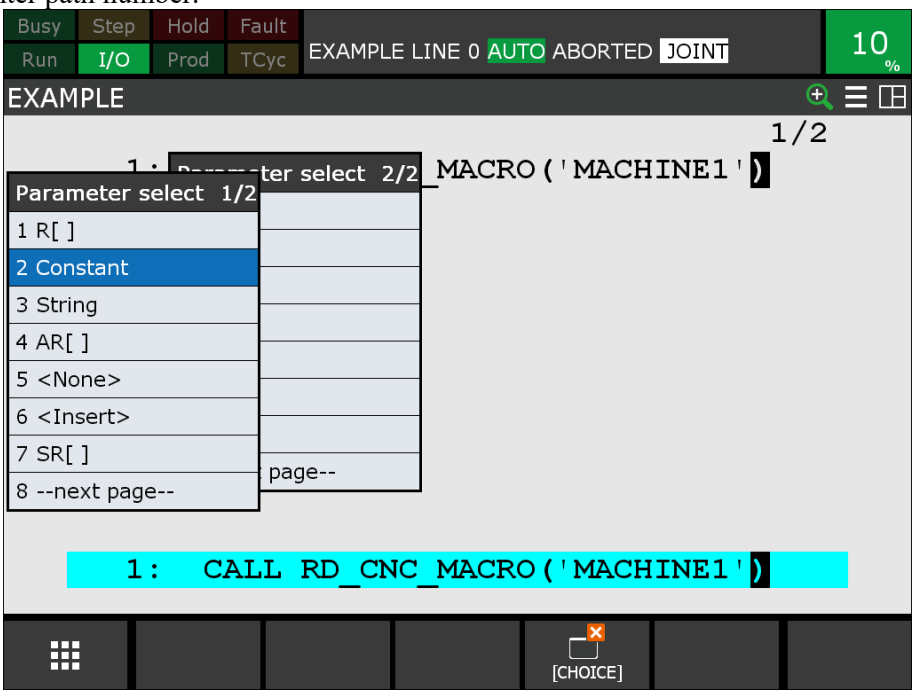
3 Move cursor to right side of the “RD_CNC_MACRO” and press the F4 [SELECT] key and select the “String”.



4 Press the F5 [STRINGS] key and enter host name.



5 Move cursor to right side of host name and press the F4 [SELECT] key and select the “Constant”. And then enter path number.



6 Move cursor to right side of the path number and press the F4 [SELECT] key and select the “Constant”. And then enter number of custom macro variable.

7 Move cursor to right side of the custom macro variable number and press the F4 [SELECT] key and select the “Constant”. And then enter number of numeric register.

73.4 CNC SETUP

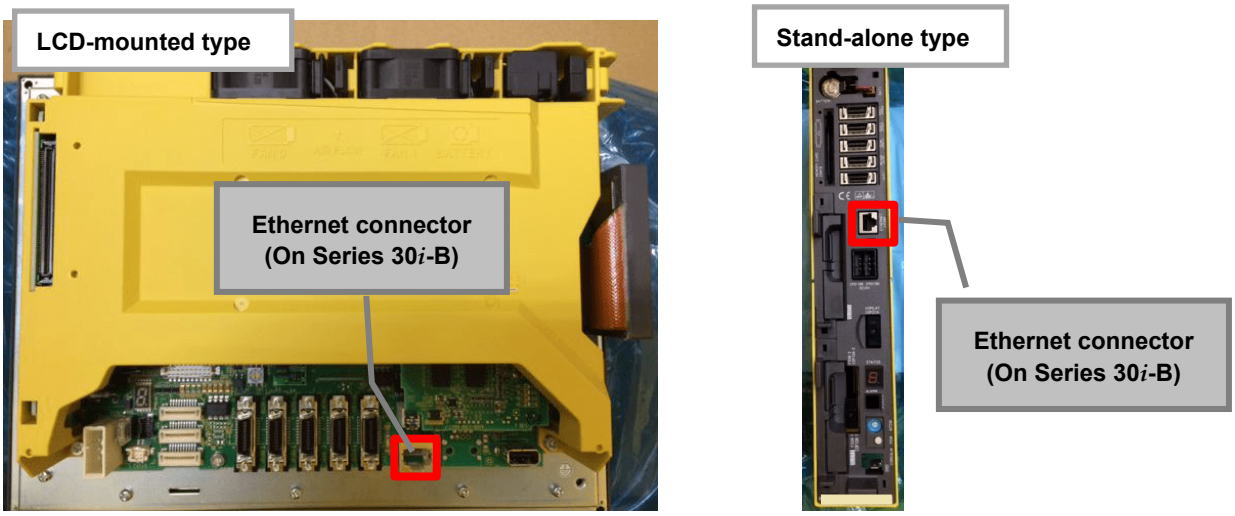
This chapter describes the CNC settings required to use custom macro variable access.
The configuration procedure is as follows.

Procedure 73-3 Setup

1 Identify the Ethernet connector on the CNC for embedded Ethernet.

CNCs are categorized in the following two types:
- The LCD-mounted type CNC, which has its control unit on the back side of its display unit.
- The stand-alone type CNC, which has its control unit separated from its display unit.

Following are examples of those two types.



Depending on its CNC model and display unit type, a stand-alone type CNC may require either that connector CD38A be connected to its control unit or that connector CD38S be connected to its display unit.
To specify which Ethernet connector to use, set either of the following parameters.

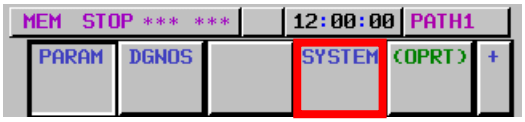
Parameter	Description	Value to Set	Built-in Ethernet Port
14896	This parameter is available for stand-alone type Series 30*i* /31*i* /32*i* with display unit supporting Windows CE to use embedded Ethernet.	0 (or 1)	To use the port on the control unit. Connector CD38A
		2 (or 3)	To use the port on the back side of the display unit. Connector CD38S
14897	This parameter is available for stand-alone type Series 30*i* /31*i* /32*i* with 15 inch color LCD / 10.4 inch color LCD B display unit to use embedded Ethernet.	0 (or 1)	To use the port on the control unit. Connector CD38A
		2 (or 3)	To use the port on the back side of the display unit. Connector CD38S

NOTE
Those parameters listed above require turning off the power to the CNC after they are set. For the details on how to operate a CNC, refer to the "CONNECTION MANUAL (FUNCTION)" for its model.

2 Verify the settings on the [SYSTEM CONFIG(SOFTWARE)] screen whether embedded Ethernet software has been configured in the CNC.
* The following steps take Series 31*i*-B (with 10.4 inch display unit) as example for explanation purpose.

1) Press the function key [SYSTEM].

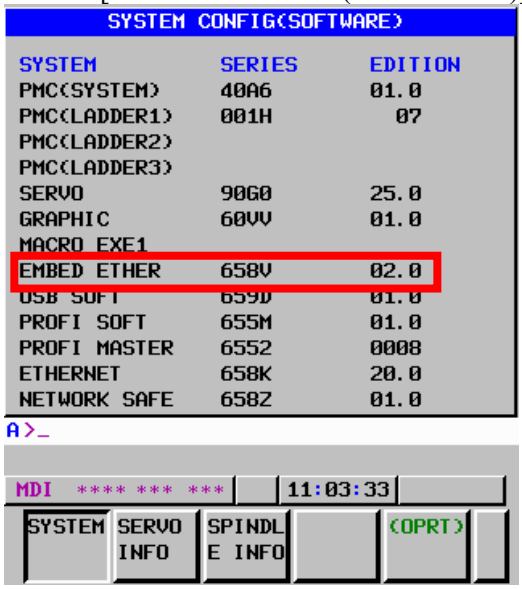
2) When the soft key [SYSTEM] appears, press it.



3) Press the change page key [PAGE] until the [SYSTEM CONFIG(SOFTWARE)] screen appears.

4) Check the [EMBED ETHER] row whether its [SERIES] and [EDITION] columns have values, which means that the embedded Ethernet function is available on the CNC.

Example of the [SYSTEM CONFIG(SOFTWARE)] screen

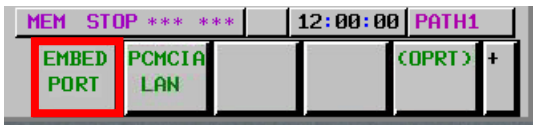


3 Set up Ethernet function on the CNC.

* The following steps take Series 31i-B (with 10.4 inch display unit) as example for explanation purpose.

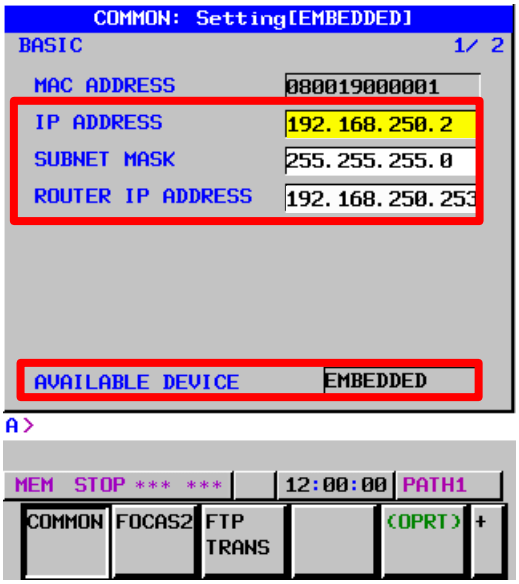
1) Press the function key [SYSTEM].

2) Press the soft key [+] (or the continue key) a few times to display the soft key [EMBED PORT].



3) Press the soft key [EMBED PORT] to open the [COMMON: Setting [EMBEDDED]] screen.

4) Specify values for the [IP ADDRESS] and the [SUBNET MASK] fields. If needed, specify a router IP address for the [ROUTER IP ADDRESS] field. Make sure that "EMBEDDED" is populated in the [AVAILABLE DEVICE] field (See NOTE below).



The following table lists the fields to set and the display-only fields.

Field	Example Value	Description
IP ADDRESS	192.168.250.2	Specify the IP address of the embedded Ethernet board.
SUBNET MASK	255.255.255.0	Specify the mask address for the IP address of the network.
ROUTER IP ADDRESS	192.168.250.253	Specify the IP address of a router deployed in the network.
MAC ADDRESS	* display-only	This field displays the MAC address of the embedded Ethernet board.
AVAILABLE DEVICE	* display-only	This field displays the device that is currently selected to use embedded Ethernet function. Either "EMBEDDED" or "PCMCIA" will be displayed in this field.

NOTE

When the [AVAILABLE DEVICE] field displays "PCMCIA" instead of "EMBEDDED," communication via FOCAS2/Ethernet functions is disabled. To address this issue, perform the following steps to set the [AVAILABLE DEVICE] field to "EMBEDDED" and restart the embedded Ethernet process:

1) Press function key [SYSTEM].

2) Soft keys [EMBED PORT] and [PCMCIA LAN] appear.
If they do not appear, press soft key [+] (or the continue key) a few times until they appear.

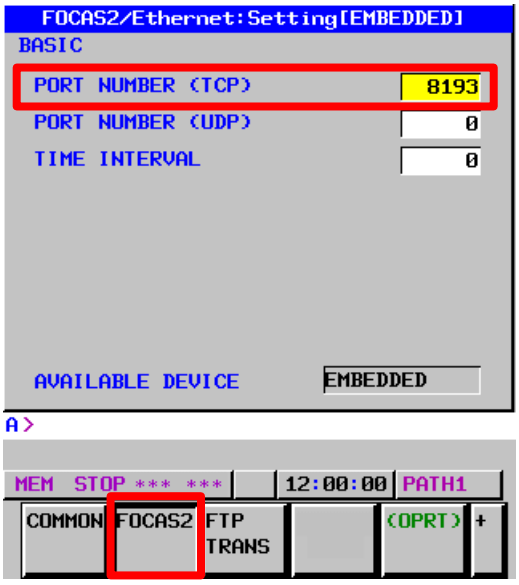
3) Press either soft key [EMBED PORT] or [PCMCIA LAN], and then press soft keys [COMMON] and [<OPRT>] to display soft key [EMB / PCMCIA].

4) Pressing soft key [EMB / PCMCIA] toggles the [AVAILABLE DEVICE] field between "EMBEDDED" and "PCMCIA." Make sure that the [AVAILABLE DEVICE] field is displaying "EMBEDDED."

5) Press soft key [RSTART] to reset and restart the embedded Ethernet communication process.

5) Press the soft key [FOCAS2] to open the [FOCAS2/Ethernet: Setting [EMBEDDED]] screen.

6) Specify a port number for the [PORT NUMBER (TCP)] field.



The following table lists fields to set.

Field	Example Value	Description
PORT NUMBER (TCP)	8193	Specify a port number of the port for FOCAS2/Ethernet functions. This port number must be in the range of 5001 to 65535.
PORT NUMBER (UDP)	0	Specify 0.
TIME INTERVAL	0	Specify 0.

73.5 ABOUT ALARM

This section explains the alarms related to this function.

If fail to load the custom macro variable, the following alarm occurs.
HOST-335 "Failed to read CUSTOM MACRO Variable"

If fail to write the custom macro variable, the following alarm occurs.
HOST-336 "Failed to write CUSTOM MACRO Variable"

If Selected CNC cannot be access, the following alarm occurs.
HOST-300 "Can't connect to CNC (%s)"
HOST-301 "Not Connected (Tag=%s)"

If select illegal host name and execute the instruction, the following alarm occurs.
HOST-245 "Illegal Hostname"
HOST-331 "IP Address mis-configuration"

If select the path that don’t exist in CNC and execute the instruction, the following alarm occurs.
HOST-330 "CNC Path is invalid"

If select illegal number of custom macro variable and execute the instruction, the following alarm occurs.
HOST-332 "CUSTOM MACRO ID is invalid"

If select illegal numeric register number as the output destination and execute the instruction, the following alarm occurs.
HOST-334 "Output Register is invalid"

If the loaded value from the custom macro variable is greater than max value of single precision floating point number or less than min value, the following alarm occurs.
HOST-337 "Read CUSTOM MACRO Value has overflowed"

If the input the value that cannot be used in custom macro variable and execute the instruction, the following alarm occurs.
HOST-333 "Input CUSTOM MACRO value is invalid"

If select the host name of the CNC that don’t be supported and execute the instruction, the following alarm occurs.
HOST-303 "Not supported CNC"

If accessing the CNC from two or more programs at the same time, the following alarm occurs.
HOST-339 "Accessing CUSTOM MACRO VAR of same CNC"

INDEX

<Number>

2 CONTACT POINT SWITCH 456
4D EDITOR FUNCTION 361
4D GRAPHICS DCS 360
4D GRAPHICS Display.....352
4D GRAPHICS edit node map.....356
4D GRAPHICS Frame Display.....358
4D GRAPHICS FUNCTION 348
4D GRAPHICS Node Map 355
4D GRAPHICS Position Register 359
4D GRAPHICS SCENE.....352
4D GRAPHICS select node map.....357
4D GRAPHICS TCP Trace.....358

<$>

$NC_DEFMODAL 633
$NC_DEST_DRV.....634
$NC_GCOD_TBL : To define G-code numbers with their meaning.....632
$NC_MCOD_TBL : To define M-code numbers with their meaning.....632
$NC_OFST_CMD 634
$NC_SYTX_TBL 631
$NC_TOFS_CMD 634
$NC_VOFS_CMD : Command string to be used for Vision offset command.....634

<A>

A Different Tool from that Specified by the Attach Instruction is Attached.....311
Abort KAREL Program 220
Abort Program by UOP.....148
ABOUT ALARM.....867
ABOUT MANUAL.....1
Absolute（ABS） 281
Accessing the Full Screen 4D Display 362
Accuracy Around Singularity.....622
Addition of Button Change Control 185
Addition of Command Button Control.....188
Addition of Edit Box Control.....189
Addition of Label Control 190
Addition of Toggle Button Control 190
Addition of Toggle Lamp Control.....192
ADDITIONAL OPTION 512
ADJUSTMENT.....489
ADVANCED CONSTANT PATH 99
Advanced Function 614
After Replacement 323
ALARM 710,830
ALARM CAUSE/REMEDY DISPLAY FUNCTION .507
Alarm ID 154
Alarm Number 155
ALARMS REPORTING SCREEN.....566
Ambient Temperature Setting 517
ANTI-DEFLECTION FOR EXTERNAL FORCE 433
APPENDIX.....244,491,719
APPLICATION 466
“APPLY” FUNCTION KEY 747
ASCII File Example.....245
ASCII File General Event Information 239
ASCII File Specific Event Information.....240
Assign Alarm History 542
Assign comment.....549
Assign Current Position 540
Assign I/O data and simulation status 551
Assign Integrated PMC address data.....552
Assign Position Registers.....536
Assign Program Execution Status 544
Assign Robot Registers 534
Assign String Registers 540
Assign Symbol and Comment of Integrated PMC address.....553
Assign System Variables.....545
Assigning Tool Numbers to Servo Tool Axes.....303
ASSIGNMENT OF HOLDING REGISTERS.....532
ASSIGNMENT OF TOUCH SENSING I/O 396
ATPERCH 144
ATTENTION.....124
“Auto Add TP Points” ITEM.....746
AUTO BACKWARD EXIT 850
AUTOMATIC ERROR RECOVERY FUNCTION 27
AUXILIARY AXIS SERVO OFF (LOCAL STOP) FUNCTION.....118
Available *i*Pendant Controls 180
AVAILABLE MOTION OPTIONS 738

<B>

BACKGROUND OPERATION OF MATH FUNCTION.....282
BACKUP AND RESTORE 464
Backup/Restore 202
BACKWARD EXECUTION.....732
Backward Execution (BWD) 300
BASIC FUNCTION.....604
Basic Setup.....581
Basic Specification.....286
Battery-less Type Tools 297
Battery-Mounted Type Tools.....298
BENEFITS OF SIGNATURE CALIBRATED ROBOT596
Binary 756
BRAKE CHECK FUNCTION.....174
BUTTON DETAIL SETUP SCREEN.....446
BUTTON TYPE SETUP SCREEN 441
Button Types Supported.....451

<C>

CABLE CONNECTION.....568
Calculate Parameters Screen.....611
Calibration Motion Failed. 311

CALIBRATION PROCEDURE (for 6-Axis Robots) 70
Calibration Recovery 614
Calibration System 597
Calibration Type 598,605
Case the Path Has High Curvature 734
CASES SPLINE CANNOT BE PLANNED 733
Cause and Remedy for Alarm Occurrence 266
CAUTION 176,251,391,638,763
Caution for Creation of KAREL Program 200
CAUTIONS 61,110,427,622,849
CAUTIONS / RESTRICTIONS 14
Ch (Channels) Switching by Pre_Ls Instruction 678
“Change Blend-In Speed” ITEM 745
Change Display Format of Cycle Time Data 270
CHANGE POSITION DATA DURING S MOTION EXECUTION 739
CHANGING CONDITIONS FOR EXECUTING THE RESUME PROGRAM 48
Check Maintenance Time 518
CHECK MAINTENANCE TIME AND COMPLETE MAINTENANCE 518
CMDENBL#1 to #N 142
CMOS Programs 257
CNC CUSTOM MACRO VARIABLE ACCESS 854
CNC SETUP 862
CNT 722
COL DETECT ON / COL DETECT OFF 58
COL GUARD ADJUST 59
COLLISION GUARD SETUP SCREEN 57
COMMON PROPERTIES 449
Common Setting 513
Common Setting of All Shapes 584
Common Settings 725
Communication Method 758
Communication setting 643
Communication setting on the laser oscillator side 647
Condition to use 22
Configuration of the Data Connection 575
Configuring Microsoft® Internet Explorer 362
Confirm communication setting 645
Confirm Communication Setting for Laser Oscillator 719
CONFLICT BETWEEN DATA WRITES 391
Connection 835
CONNECTION OF HMI DEVICE 526
CONNECTION OF LASER AND ROBOT 639
CONSTRAINTS 120
CONSTRUCTION OF SERVO TOOL CHANGE SCREEN 302
CONTINUOUS ROTATION FUNCTION 16
Controllers 562
Conversion Operation 624
Coordinate Systems 602
COORDINATED TOUCH SENSING 423
Copy and Paste 444
COPY NC program to MD drive 625
Copy Programs 264
CORETEC BEAD INSPECTION INTERFACE FUNCTION 568
CORNER REGION 102
Correspondence of Modbus Address to Robot Data 531
Create and Edit Shape program 592
Create Calibration Program Screen 607
Create Signature Calibration Parameters Setup File 613
CREATING AND EDITING VISUAL DIAGNOSTIC SCREENS 493
Creation of Run Button 200
Creation Of Welding Shape Program 725
CRW SHAPE GENERATION FUNCTION 724
CRW Shape Generation Screen 725
CSTOPI#1 to #N 141
Custom 590
CUSTOM MENU 224
CUSTOM SIGNATURE CALIBRATION 597
Custom Signature Calibration Preparation 599
CUSTOMIZE SUPPORT FUNCTION 218
Cut/Copy Paste of Control 197
Cycle Mode 268
Cycle Power (R-30*i*B Controller) 224
CYCLE TIME LOGGING 267

<D>

DATA FORMATS 756
DATA MONITOR CHART 171
DATA MONITOR FUNCTION 156
DATA MONITOR OPERATION 173
DATA MONITOR SCHEDULE 167
DATA MONITOR SETUP 158
DATA OFFSET FUNCTION 470
Data supported by node map 357
DATA TRANSFER BETWEEN ROBOTS FUNCTION 367
Data type of Holding Registers 534
DCS Safety Signal: RPI 430
DEFAULT 722
Delete of Control 196
Delete Set 227
Destination Information Settings 758
Detail 328
Detail Screen of KAREL Config 222
DETAILS 578,800,805,810,814
Details Comparison 331
Details of Power Control 680
Details of slope down control 687
Details of slope up control 684
DIGITAL DISPLAY 460
DIGITAL I/O JOG FUNCTION 761,868
DIGITAL SWITCH 459
DIRECT POWER OVERRIDE MODE 696
Direct Power Override Mode in Laser Oscillator 696
Direct Power Override Mode in Robot 697
Display in laser status screen 711
Display in status window 710
Display in the active alarm screen 711
Display Maintenance Record 522
DISPLAY MODES OF CYCLE TIME 268
Display of Panel 201

Display Reference Cycle Time Line 272
Display Target Cycle Time Line.....271
DUAL MOTOR DRIVE 129
Dump Selections Screen.....229
DYNAMIC ASSIGNMENT TO HOLDING REGISTERS.....555

<E>

Enable or Disable All Event Logging 237
Enable/Disable Ch# Laser Ctrl Rq 677
Enable/Disable Ch# Laser Ctrl Rq Switching 676
Enable/Disable Direct Power Override Mode 697
Enable/Disable iRCalibration Signature Compensation618
“End Line” ITEM.....744
ENHANCED MIRROR IMAGE 203
ERROR CODE OUTPUT FUNCTION 151
Error Display of Button.....448
Error message.....23
Estimation Execution and Applying the Result.....69
Estimation Motion Setup.....65
Estimation Result 70
Ethernet Cable Connection to Laser Oscillator 641
Ethernet cable connection to robot.....640
Event Class Selection Screen 235
Event Detail Selection Screen 237
Example of a Setting 481
Example of use.....20
EXCEPTIONS AND RESTRICTION 285
Execute Program 323
EXECUTION EXAMPLE OF SPOT PROGRAM 273
Execution Of A Path Welding Shape Program.....727
Execution Of A Point Welding Shape Program 727
EXECUTION OF THE RESUME PROGRAM FROM THE TEACH PENDANT AND TEST MODE.....48
EXECUTION OF TOUCH SENSING PROGRAM 416
EXECUTION OF WELDING SHAPE PROGRAM ...727
EXPANDED REGISTERS FUNCTION 482
Expansion of Peripheral Equipment I/O.....677
Exponent 280
EXTENDED ALARM LOG 93
EXTERNAL I/F PANEL SELECTION SETUP SCREEN.....464
External view and dimensions.....834

<F>

F code – Feed Rate / Speed 627
FANUC FIBER LASER INTERFACE.....638
FAST EXIT/ENTRY FEATURE.....28
FAULT & INCIDENT REPORT 566
FAULT#1 to #N.....143
Feature of Function 286
Feed Rate Conversion 470
FILE Programs 258
FINE.....722
FINISHING FUNCTION PACKAGE 479
Flow of Stitch Process.....487
For multi-task 639,763
For the safety.....638,763
FORMAT OF THE TELEGRAMS SENT BY THE ROBOT 573
Forward Execution (FWD).....299
From laser start to end.....799
From laser start to end when is Time-sharing is DISABLE.....796
From laser start to end when Time-sharing is ENABLE798
FULL SCREEN 4D DISPLAY 362
FUNCTION SPECIFICATION OF MATH FUNCTIONS 277
Functional Specifications 623
Functions that cannot be used together 20

<G>

G code – Motion modifiers 626
“Geometric Data” ITEM 745
“Geometric Type” ITEM 744
Graphic Models.....349
GRAVITY COMPENSATION.....79
Guidance of GC Mastering 82

<H>

Hardware.....228,742
HARDWARE AND SOFTWARE.....228,742
Hardware and Software Requirements.....228
Hardware Specifications 832
HELD#1 to #N.....143
HELP AND DIAGNOSTICS DISPLAY 506
HELP/DIAGNOSTICS MENU 510
HELP/DIAGNOSTICS SCREEN.....510
HIGH SENSITIVITY COLLISION DETECTION 56
HIGH SPEED POSITION OUTPUT.....756
HIGH SPEED SHIFT JOG OPERATION.....317
HIGH SPEED SHIFT KEY FUNCTION 317
HIGH SPEED SHIFT TEST EXECUTION OPERATION 318
Hints.....554
HMI DEVICE COMMUNICATION.....525
HOLD#1 to #N.....140
Hourly Mode.....269
How to Choose Mastering Method 81
HOW TO COMPENSATE 434
How to Display Alarm Log.....94
HOW TO DISPLAY PANEL 437
How to Use 100,480,722
HOW TO USE MROT.....428
HOW TO USE SINGULARITY AVOIDANCE 109
How to Use Singularity Avoidance in Jogging 109
How to Use Singularity Avoidance in TPE Program ...109

<I>

I/O and Data Communication Setting 642
i-cable Change 524
IF USE iPendant.....858
IF USE TABLET TP.....856
i-grease Change.....524
INITIAL SETTING BEFORE USE.....174
INITIAL SETUP.....289

Input and Output Signals....151
Instruction....485
Instruction Format of Assignment Statements....276
INSTRUCTION FORMAT OF MATH FUNCTION..275
Instruction Format of Relational Statements....276
Instruction Format of Wait Command Statements....277
Intelligent Cable Change Reminder (S511)....512
Intelligent Grease Change Reminder(J766)....512
INTERFACE PANEL....438
INTERFACE PANEL FUNCTION....436
INTERFACE PANEL SETUP SCREEN....440
Inverse Trigonometric Function（ACOS）....279
Inverse Trigonometric Function（ASIN）....278
Inverse Trigonometric Function（ATAN）....280
Inverse Trigonometric Function（ATAN2）....279
*i*RCalibration SIGNATURE....595
*i*RCalibration Signature Screen....604
iRCalibration Signature Status Screen....619
*i*RDIAGNOSTICS....320

<J>

JOG OPERATION WITHOUT SHIFT....314
Jog preview....354
JOINT / CARTESIAN SOFTFLOAT....5
Joint Angles with Binary....757
Joint Angles with XML....758
JOINT POSITION OUTPUT FUNCTION....480

<K>

K code - Config string....628
KAREL BUILT-IN....375
KAREL BUILT-IN ROUTINES....750
KAREL CONFIG....218
KAREL Config Screen....218
KAREL Program Example....244
KAREL PROGRAM EXECUTION HISTORY RECORD....228
KAREL PROGRAM SAMPLE....753
KAREL PROGRAMS AND KAREL BUILT-IN ROUTINES....575
KNOWHOW OF CARTESIAN SOFTFLOAT....13
KNOWHOW OF SOFTFLOAT....12

<L>

LAMP....458
Laser Alarm....710
Laser Command Signal Analog Output....841
LASER ENABLE/DISABLE....782
Laser Enalbe/Disable Screen....782
LASER I/O....654,773
Laser Initial Setup....648,763
Laser Instruction....665,784
LASER INTERFACE FUNCTION....763
LASER INTERFACE UNIT....831
Laser Oscillator I/O....654
Laser oscillator input screen....660
Laser oscillator output screen....654
Laser Setup....649,768
LASER STATUS SCREEN....706
LE Instruction....793
LE instruction (laser processing end instruction)....670
“LEARN” FUNCTION KEY....746
Learning....748
LEARNING PATH CONTROL....742
Limitations....762
Limitation and Caution of KAREL Config....223
Limitation of the Specifiable CR Value....103
LIMITATIONS76,101,104,107,108,110,176,251,252,391,428,465,490,723,749,848
LINEAR DISTANCE....99
Line-by-Line Mode....269
List Menu....339
List Selected Tasks Screen....234
LOAD ESTIMATION....62
LOAD ESTIMATION PROCEDURE (for 6-Axis Robots)....62
Load NC program in FILE menu....624
LOADING EXPANDED xxxREG.VR....483
LOADING NOT EXPANDED xxxREG.VR....483
LOADING PROCESS IN PROGRAM EXECUTION 264
LOGGING CYCLE TIME....270
LOGGING EVENTS....238
Logging Events to an ASCII File....239
LOGIC INSTRUCTIONS BETWEEN SPLINE MOTION....736
LP instruction (piercing instruction)....672
LPC BACKUP....749
LPC LEARNING AND PLAYBACK....748
LPC SETUP MENU....743
LS Instruction....786
LS instruction (laser processing start instruction)....666
LS_PO Instruction....699
LS_SCH Instruction....792
LS_SCH_TA Instruction....828
LS_TA Instruction....703,826
LS_TA INSTRUCTION WITH TIME ADJUSTMENT703,826

<M>

M code – Miscellaneous codes....628
MAIN MENU....513
“Main Program” ITEM....744
MAINT_PROG INSTRUCTION....32
Maintenance Item Setting....515
MAINTENANCE RECORD....522
Maintenance Record File....523
Maintenance Remind....518
MAINTENANCE REMINDER....512
Make Backup of Programs....264
MANAGING SCREENS....500
MANUAL OPERATION SCREEN OF THE RESUME PROGRAM FUNCTION....46
MASTERING....80,620
Mastering for Signature Calibrated Robot....620
Mastering Procedure....81
MATH FUNCTION INSTRUCTION....275
MAX SPEED....107

"Max. Iterations" ITEM 746
MEANING OF ALARM CODE 152
MENU UTILITY FUNCTION 333
METHOD OF OPERATING 430
MINUTE OUTPUT MODE 694
Minute output request instruction 675,695
MIRROR IMAGE OF EXTENDED AXES 208
MIRROR IMAGE USING EXISTING FRAMES AND MIRROR PLANES, WITH CONTROLLED ORIENTATION 207
MISCELLANEOUS SETTING SCREEN 461
MODBUS DATA MODEL 530
Modbus Function Code 532
Modbus RS-232-C Connection 526
Modbus/TCP Slave Ethernet Connection 529
Modification and Addition of Signal 139
Modification of Control 195
Modification of Page 198
Modification of Panel 194
Monitoring the Change of Mastering Data 621
MOTION INSTRUCTION ENHANCED EDITING ... 468
MOTION INSTRUCTION INSERT AND INSTRUCTION DELETE FUNCTION 468
MOTION PROFILER 325
MOTION SCREEN 79
MotionDevices 559
MotionDeviceSystem 558
Mounting position in the Robot controller 841
MQTT 750
MQTT_CONNECT 750
MQTT_DISCON 752
MQTT_PUBLISH 751
MROT INSTRUCTION 428
MULTI UOP INTERFACE FUNCTION 136
MULTIPLE SEARCHES 422

<N>

N code – Line numbers 626
Natural Logarithm 280
NAVIGATING WITH THE PDF VIEWER 505
NC PROGRAM CONVERSION 623
NC PROGRAM CONVERSION EXAMPLE 635
NC program syntax 623,625
Necessary Software Option 638,724,763
"No. CIR Lines" ITEM 746
"Normal Mastering" and "GC Mastering" 80
NORMAL MOTION 729
Note 23,480
Notes 19,804,808,817,821,825
NOTES / LIMITATIONS 20
Notice for Teaching 301
Notifications For Power Control 683
Notifications for Slope Control 691

<O>

O code – Program Name/Number 625
ONLINE HELP FUNCTION 506
OPC UA DATA MODEL 556
OPC UA Organization of Modbus data model 530
OPC UA Server Ethernet Connection 529
OPENING A PDF DOCUMENT 504
OPERATING PROCEDURE 62
Operation 362,624
OPERATION CONDITION 451
Operation Condition Setup Screen 452
OPERATION GROUP DO OUTPUT FUNCTION 25
OPERATION LOG BOOK 83
Operation Panel 430
Operation Procedure 349
OPERATION WITHOUT SHIFT FUNCTION 314
OPERATIONS 89
Operator Entry Menu 344
"Orientation Control" ITEM 746
Other 148
Other (user defined) M-codes 631
Other notes 21
OTHER RELATED MATTERS 73
Other Specifications 488
OTHER SPECIFICATIONS AND RESTRICTIONS ... 49
OUTLINE 286,831
Outline of Installation 289
Output Data Setup 759
OVERVIEW 3,16,177,224,244,349,361,362,579,595,623,638,724,742,750,761,763,832,850,854

<P>

PANEL WIZARD 177
PARALLEL MIRROR IMAGE 204
PATH SWITCHING FUNCTION 112
"Path Tolerance" ITEM 746
Pause and Resume from Another Spline Motion Instruction 735
Pause Program by UOP 148
PAYLOAD OVER AND PAYLOAD SETTING CONFIRM FUNCTION 75
PDF VIEWER FUNCTION 504
Performance 229
Peripheral Equipment I/O 663,780
Peripheral equipment input screen 664,781
Peripheral equipment output screen 663,780
PERIPHERAL I/O 137
PNSTROBE#1 to #N,PROD_START#1 to #N 142
Position data 627
Position Data Conversion (2 point teach method) 476
Position Data Conversion (direct method) 473
Position Program 600
POSITIONING PATH OF THE SPLINE MOTION ... 732
POWER COMPENSATION 818
POWER CONTROL 680,800
Power control instruction 675,681,802
Power Control Schedule 681
Power Control Schedules 801
POWER CONTROLII 805
PRE_LS Instruction 784
PRE_LS instruction (preparation instruction for ls instruction) 665

PRECAUTION 266
Precautions 469,478
Pre-defined M-codes 629
Predefined M-codes 629
PREFACE 1
PRELIMINARY TOOL ATTACH OPERATION....290
Preparation 62
PREPARATION TO CREATE COMPENSATION PROGRAM 433
Preview of Button 446
PROCEDURE 76,742
Procedure to Show Payload Over and Payload Setting Confirm Screen....76
Procedure to Use 102
Procedure to Use Payload Over Confirm (When The Program or Payload is Changed) 77
Procedure to Use Payload Setting Confirm (When Payload is Changed) 76
“Process Disabled” ITEM 746
PROCESS SPEED 105
“Process Speed” ITEM....745
Processing Speed Allowable Variation 683
PROCESSING STATE RECORD FUNCTION ...709,828
Production with LPC Playback 749
Program Edit Screen Display When Operating Motion Group is Changed 146
PROGRAM EXAMPLE 570
PROGRAM EXCHANGE FUNCTION WITHOUT ENOUGH MEMORY SPACE 265
PROGRAM INSTRUCTION....847
PROGRAM INSTRUCTIONS 58
Program Number Select (PNS) 148
Program to Get Numeric Register 370
Program to Get Position Register 371
Program to Set Numeric Register....371
Program to Set Position Register....372
PROGRAM TOOLBOX 96
Program used for Shape Generation Function....580
PROGRAMMING 125,171,853
PROGRUN#1 to #N,PAUSED#1 to #N....143
Prompt Box Msg 334
Prompt Box Yes/No Menu 336
PUSH BUTTON 454
PUSH BUTTON LAMP 455
PUSHOUT SOFTFLOAT 9

<Q>

Quick Mastering....620
Quick Mastering Reference Position Setup....309

<R>

RECORDED EVENTS 85
RECOVERY FROM ERROR....374
Re-creation of Panel....199
Reference Frame Function 617
Reference Position Setup for Calibration Types 3 and 4305
Reference Position Setup for Calibration Types 5 and 6307
RELATED VIEW 747
Release and recover of laser alarm....711
REMOTE TCP FUNCTION....51
REQUIRED OPTION....854
Requirements 362
RESET ROTATION OF POSITIONER AND INDEPENDENT AXIS BY KAREL PROGRAM....22
RESTART AFTER CHANGING PROGRAM....740
RESTRICTION OF TEACHING MATH FUNCTION284
Restriction of the number of group 638
Restrictions 286,435,622,741
“RESULT” FUNCTION KEY....747
Results....327
RESULTS OF BRAKE CHECK 176
Resume After a Pause During Backward Execution 736
Resume After JOG 735
RESUME AFTER PAUSE 735
RESUME PROGRAM FUNCTION....27
RESUME_PROG INSTRUCTION....28
RETURN_PATH_DSBL INSTRUCTION....31
RGET_PORTCMT Built-in ROUTINE....375
RGET_PORTSIM Built-in ROUTINE 376
RGET_PORTVAL Built-in ROUTINE 377
RGET_PREGCMT Built-in ROUTINE....378
RGET_REG Built-in ROUTINE 378
RGET_REG_CMT Built-in ROUTINE 379
RGET_SREGCMT Built-in ROUTINE....380
RGET_STR_REG Built-in ROUTINE 380
RNUMREG_RECV Built-in ROUTINE 381
RNUMREG_SEND Built-in ROUTINE....382
Robot Condition Detection Status....323
ROBOT CONDTION DETECTION 320
Robot ID setting (in case of connecting several robots)642
ROBOT ISOLATION FUNCTION....430
Robot Service Request (RSR)....147
ROBOT SETTING 524
ROBOT SETUP....855
RobotInformation....557
ROTATIONAL MIRROR IMAGE 206
Round Off（ROUND） 281
RPOSREG_RECV Built-in ROUTINE 383
RPOSREG_SEND Built-in ROUTINE....384
RSET_INT_REG Built-in ROUTINE....385
RSET_PORTCMT Built-in ROUTINE 385
RSET_PORTSIM Built-in ROUTINE....386
RSET_PORTVAL Built-in ROUTINE....387
RSET_PREGCMT Built-in ROUTINE 387
RSET_REALREG Built-in ROUTINE....388
RSET_REG_CMT Built-in ROUTINE....389
RSET_SREGCMT Built-in ROUTINE 389
RSET_STR_REG Built-in ROUTINE....390
RSI INTERFACE 572
RSI_CREATE_I( )....576
RSI_DELETE_I( ) 576
RSI_OFF_I ( ) 577
RSI_ON_I ( )....577
RSI_RESET_I( ) 576
RSI_WRXML_I ( ) 578
RSR....142

Run Calibration Program Screen....608
Run KAREL Program....219
RUN KAREL PROGRAM BY PANEL....200
RUN PROGRAM....147

<S>

SAFETY PRECAUTIONS....s-1
SafetyStates....565
Sample Program....299,302
Save / Load LS Files....264
SAVE / LOAD PROGRAMS....263
Save / Load TP Files....263
Save Cycle Time Data....271
SAVEING AND LOADING FILES....483
Scene visibility....365
SEARCH PATTERN....401
SELECT PROGRAM....144
Select Program at Running or Paused....146
SERVO DIAGNOSIS....324
SERVO TOOL CHANGE FUNCTION....286
SERVO TOOL FUNCTION....843,844
SERVOTOOL END / SERVOTOOL END[schedule number]....847
SERVOTOOL[schedule number]....847
Set Custom Menu....225
Set visibility: 4D GRAPHICS display....354
Set Visibility: node map position number....357
Set Visibility: position register number....360
Setting....480,574
SETTING (SYSTEM VARIABLES)....569
Setting at UTILTIES screen....592
Setting Motion Parameters for Servo Tool Axes....303
SETTING OF LASER....647
Setting of Motion Groups Intended by Each UOP Set....145
SETTING OF THE AUTOMATIC ERROR RECOVERY FUNCTION....35
Setting of the Number of UOP Set....138
SETTING OF TYPE OF BUTTON....444
Setting System Variables....305
SETTING THE NUMBER OF REGISTERS....482
SETTING THE REFERENCE POSITION....297
SETTING UP....177
SETTING UP BOOK....89
Setting up Events....238
Setting up Fast Label....181
Setting up Fast Lamp....182
Setting up Fast Switch....184
SETTING UP TCP SPEED OUTPUT....253
Setting Up the KAREL Program Execution History Record....229
SETTINGS....56,121
SETUP17,53,93,130,320,326,362,368,466,513,758,761,763,851
SETUP ABOUT MENU UTILITY....333
SETUP AND OPERATIONS....229,850
SETUP CUSTOMIZATION....435
SETUP FOR HIGH SPEED SHIFT KEY FUNCTION317
Setup for Maintenance Reminder....521
SETUP OF DUAL DRIVE PAIR FOR 1-AXIS POSITIONER....132
SETUP OF TOUCH FRAME....398
Setup Screen....606
SETUP THE SERVO TOOL FUNCTION....844
Severity of Alarm....153
SHADOW ONDEMAND Programs....258
SHADOW Programs....257
SHAPE GENERATION FUNCTION....579
SHAPE GENERATION SETUP....580
SHAPE GENERATION UTILITY....592
Shortage of the Spline Motion Instruction....733
SINGLE STEP EXECUTION....732
SINGULARITY AVOIDANCE FUNCTION....109
SLOPE CONTROL....684,810
Slope Down Control....687,814
Slope down instruction....675,689,815
Slope down schedule....688
Slope down schedules....814
Slope up Control....684,810
Slope up instruction....675,685,812
Slope up schedule....685,811
SOFT LIMIT SETTING....96
SOFTFLOAT FUNCTION....5
Software....228,742
SPECIAL JOG SEQUENCE....466
Special M-codes....629
Specific Setup of Circular Shape....585
Specific Setup of Hexagon Shape....587
Specific Setup of Keyhole Shape....588
Specific Setup of Pommel Shape....589
Specific Setup of Rectangle Shape....588
Specific Setup of Slot Shape....586
SPECIFICATION....56,118,151,430,485,761
Specification Overview....850
Speed Tolerance....804
SPLINE MOTION INSTRUCTION....729
Square Root（SQRT）....277
STANDARD DATA TRANSFER PROGRAM....370
START BRAKE CHECK....175
“Start Line” ITEM....744
Start Mode Config of KAREL program....221
Start Output....760
Start Program from Spline Motion Instruction....736
START#1 to #N....141
Starting continuous rotation....19
Starting Custom Menu....224
STARTING OF LASER OSCILLATOR....653
Status Menu....341
Stitch Condition....485
STITCH FUNCTION....485
Stop Logging Tasks Screen....232
STOP PROGRAM....148
Stopping continuous rotation....19
STORAGE CONFIGURATION....258
STORAGES....257
Sub Laser Instruction....675
Supported Shapes....724

SWITCHING CH# FOR LASER CTRL RQ 676
Switching to Direct Power Override Mode by PRE_LS698
Switching to FANUC Fiber Laser Interface Function..767
SYSRDY#1 to #N 142
System.....832
System Configuration.....288
SYSTEM VARIABLE: $ROBOT_ISOLC 431
SYSTEM VARIABLES.....79,631,852
SYSYTEM VARIABLES.....148

<T>

Task Selection Screen 231
TCP Cartesian Position with Binary 757
TCP Cartesian Position with XML.....758
TCP SPEED OUTPUT.....252
TCP SPEED OUTPUT INSTRUCTION 255
TCP/IP SETUP FOR ROBOGUIDE.....369
TCP/Uframe Calibration 615
TEACH MATH FUNCTION INSTRUCTION 282
Teach Pendant Program Example 245
TEACHING 301
Teaching laser instruction 674,794
Teaching of Same Position.....734
TEACHING TO MOTION INSTRUCTION.....432
TERMINOLOGY.....368
TEST EXECUTION OPERATION WITHOUT SHIFT315
The Attach Instruction is Executed when the Tool is not Attached. 310
The Attached Tool has been Detached by Mistake (without Using the Detach Instruction). 312
The Battery Ran Low while the Tool was Detached....313
The Battery Voltage has Fallen.....312
THE EFFECT OF ONE TEACHING POINT ON THE PATH 733
The Robot Stopped during Calibration.....310
The Tool Axis of a Detached Tool has Moved. 312
TIME OUT AND RETRY 390
Time Sharing Control Setup.....652
TIMING CHART.....795
TOOL ATTACH Instruction.....299
TOOL CHANGE INITIAL SETUP.....303
TOOL CHANGE INSTRUCTION 298
TOOL CHANGE REFERENCE POSITION SETUP METHOD (BATTERY-LESS TYPE).....305
TOOL CHANGE SEQUENCE.....300
TOOL CHANGE SETUP 292
TOOL CHANGE STATUS.....301
TOOL DETACH Instruction.....298
TORQUE LIMIT FUNCTION.....247
TORQUE LIMIT FUNCTION FEATURE.....248
TORQUE LIMIT MULTI-AXIS SETUP FUNCTION250
TOUCH SCHEDULE 405
TOUCH SENSING 394
TOUCH SENSING PROGRAMMING 410
TOUCH SKIP FUNCTION 426
TOUCH SKIP PROGRAM.....426
TOUCH SKIP SCREEN 426
TOUCHING UP OF TOUCH SENSING PROGRAM 419
TP DRAM/FILE STORAGE FUNCTION 257
TP Program Example 847
Trigonometric Function（COS） 278
Trigonometric Function（SIN） 277
Trigonometric Function（TAN） 278
TROUBLE SHOOTING.....392
Trouble Shooting About This Function'S Alarm..712,830
TROUBLESHOOTING.....310
Truncate（TRUNC） 281
TYPE OF BUTTON 449
TYPE OF MATH FUNCTIONS.....275
Types of Alarms.....151

<U>

Update Cycle Time Data.....271
Upon Completion of Maintenance 519
Usage 22,201,466,468
Use KAREL Config Screen 218
Use of Continuous Rotation function with Constant Path function 21
USING THE FUNCTION.....19

<V>

Variables with Binary 757
Variables with XML 758
Version.....761
VIA INSTRUCTION 722
View adjustment mode.....364
View/Apply Calibration Result.....612
VIEWING VISUAL DIAGNOSTIC SCREENS 501
VISUAL DIAGNOSTICS.....492
Visual jog.....353

<W>

WARNINGS (Be sure to read this section for safety.)...50
WELDING SHAPE PROGRAM CREATION BY iHMI.....725
When laser error or robot E-Stop occurs.....797,800
When there is LS before PRE_LS (recommended).....787
When there is not LS before PRE_LS.....789
Wire Contol Signal.....693
WIRE CONTROL.....691,821
Wire control instruction 675,692,823
Wire Control Setup 822
Wire Control Signal 825
Wire Feed input.....825
Wire feed OK signal 694
Wire feed on signal 693
Wire Feed output.....825
Wire Feed Speed output 825
Wire feed speed signal 693

<X>

XML.....758

<Y>

YLR Laser Oscillator I/O.....777
YLR laser oscillator input screen.....779

YLR laser oscillator output screen 777
YLR Laser setup 771
YLR Timing Chart 799
YLS Laser Oscillator I/O 773
YLS laser oscillator input screen.....775
YLS laser oscillator output screen.....773
YLS Laser setup.....768
YLS Timing Chart.....796

<Z>

ZDT / *i*RConnect.....521

REVISION RECORD

Edition	Date	Contents
10	Jan, 2022	• Add R-30*i*B Mini Plus. • Following new functions have been added. MQTT, HIGH SPEED POSITION OUTPUT, DIGITAL I/O JOG FUNCTION, LASER INTERFACE FUNCTION, SERVO TOOL FUNCTION, AUTO BACKWARD EXIT, CNC CUSTOM MACRO VARIABLE ACCESS • New features are added in *i*RCalibration Signature and LOAD ESTIMATION. • Explanation changes (singularity avoidance function, data monitor, CRW shape generation function) • Correction of errors
09	Jul.,2020	• Following new functions have been added. Add data for OPC UA Data Model. • Other minor changes.
08	Sep., 2019	• Following new functions have been added. Add data for HMI DEVICE COMMUNICATION, OPC UA, NC Program Conversion, FANUC FIBER LASER INTERFACE FUNCTION,VIA Instruction • Other minor changes
07	May.,2018	• Addition of R-30*i*B Compact Plus. • Following new functions have been added. *i*RCalibration Signature
06	Nov.,2017	• Addition of R-30*i*B Plus and R-30*i*B Mate Plus. • Following new functions have been added. CORETEC BEAD INSPECTION INTERFACE FUNCTION, RSI INTERFACE, SHAPE GENERATION • Correction of errors
05	Sep.,2015	• Following new functions have been added. VISUAL DIAGNOSTICS, PDF VIEWER FUNCTION, HELP AND DIAGNOSTICS DISPLAY, MAINTENANCE REMINDER, HMI DEVICE COMMUNICATION • Correction of errors
04	Aug.,2013	Addition of new software series 7DC2 (V8.20).
03	Mar.,2013	Addition of R-30*i*B Mate.
02	Nov.,2012	• Following new functions have been added. PAYLOAD OVER AND PAYLOAD SETTING CONFIRM FUNCTION, DATA TRANSFER BETWEEN ROBOTS FUNCTION, TOUCH SENSING, TOUCH SKIP FUNCTION, MROT INSTRUCTION, ROBOT ISOLATION FUNCTION • The function names have been modified in the following functions. SINGULARITY AVOIDANCE FUNCTION, CYCLE TIME LOGGING • The chapter of COLLISION DETECTION FOR AUXILIARY AXIS has been moved to "FANUC Robot series R-30*i*B CONTROLLER OPERATOR'S MANUAL (Basic Operation)" (B-83284EN). • Specifications have been modified in some functions. • Some errors in writing have been corrected.
01	Sep., 2012	

B-83284EN-2/10