brother

W1000Xd1 Series

COMPACT MACHINING CENTER SPEEDIO

CNC-D00
PLC SYSTEM MANUAL
(OPTION)
(For Installers & Machine Setup Specialists)

Please read this manual carefully before operating the machine.
Be careful not to lose the manuals, and keep them handy at all times.

69A809001(3)

PLC SYSTEM MANUAL (OPTION)

Read this manual before performing work.

brother

(This page was intentionally left blank.)

CHAPTER 1 INSTALLATION ---------- 1-1
1.1 Before Installation ---------- 1-2
1.2 PLC Programming Software D00 EDITOR Installation ---------- 1-2
1.3 Uninstall PLC Programming Software D00---------- 1-4

CHAPTER 2 SYSTEM CONFIGURATION---------- 2-1
2.1 Overall System ---------- 2-2
2.2 File Structure ---------- 2-3

CHAPTER 3 PROCEDURE FOR BASIC OPERATIONS ---------- 3-1
3.1 Export Ladder from NC Unit---------- 3-2
3.2 Launch Application & Import Project ---------- 3-2
3.3 Edit POUs ---------- 3-3
3.4 Convert POUs ---------- 3-3
3.5 Print POUs ---------- 3-3
3.6 Save Project---------- 3-3
3.7 Export Project ---------- 3-4
3.8 Import to NC Unit---------- 3-4

CHAPTER 4 LAUNCH & EXIT EDITOR ---------- 4-1
4.1 Launch PLC Programming Software D00 EDITOR ---------- 4-2
4.2 Close Edit Screen ---------- 4-3
4.3 Exit PLC Programming Software D00 EDITOR---------- 4-6

CHAPTER 5 PROCEDURE TO CREATE PROGRAMS ---------- 5-1
5.1 Create Projects---------- 5-2
5.1.1 Create New Projects ----------5-3
5.1.2 Open Projects----------5-5
5.1.3 Open Recently Used Projects----------5-7
5.1.4 Save As----------5-8
5.1.5 Save All ----------5-8
5.2 Create POUs (LD/ST/FBD Programs)---------- 5-9
5.2.1 Procedure to Create New POU ----------5-9
5.2.2 Open POUs ---------- 5-11
5.2.3 Save POUs ---------- 5-12
5.3 Register Global Variables ---------- 5-13
5.3.1 Create Global Variables---------- 5-13
5.3.2 Global Variable Settings ---------- 5-15
5.3.3 Delete Global Variables---------- 5-18
5.4 Parameter Settings ---------- 5-19
5.5 Register DFC (Dynamic Function Call) ---------- 5-23
5.6 Program Edit---------- 5-23
5.6.1 Create LD Programs ---------- 5-23
5.6.1.1 Edit screen adjustment (option settings)---------- 5-23
5.6.1.2 Edit LD programs ---------- 5-25
5.6.1.3 Create OM comments, inline comments and coil remarks (overview) ---------- 5-27
5.6.2 Create ST Programs ---------- 5-28
5.6.2.1 Edit screen adjustment (option settings)---------- 5-28
5.6.2.2 Edit ST programs---------- 5-29
5.6.3 Create FBD Programs ---------- 5-31
5.6.3.1 Edit screen adjustment (option settings)---------- 5-31
5.6.3.2 Edit FBD programs ---------- 5-32
5.7 Program Check ---------- 5-35
5.7.1 Double Coil Check---------- 5-35
5.7.1.1 Check items ---------- 5-35
5.7.1.2 Check Timing / Check Results---------- 5-35

5.7.2 Design Check ---------- 5-38
5.7.2.1 Check items ---------- 5-38
5.7.2.2 Check timing / check results ---------- 5-38
5.8 Program Execution Settings ---------- 5-39
5.9 Conversion ---------- 5-41
5.9.1 Conversion ---------- 5-41
5.9.2 Conversion Mode Settings ---------- 5-43
5.10 Write/Read Programs ---------- 5-44
5.10.1 Communication Settings with NC ---------- 5-44
5.10.2 Write Programs ---------- 5-45
5.10.3 Read Programs ---------- 5-48
5.10.4 Write During Run ---------- 5-50
5.11 Launch Simulation ---------- 5-54
5.12 NC(BUILT IN PLC) Monitoring ---------- 5-55
5.13 Run & Stop in NC(BUILT IN PLC) ---------- 5-59
5.14 POU Debug Operation ---------- 5-61
5.15 Close Project ---------- 5-61

CHAPTER 6 BASIC OPERATION PROCEDURE FOR PROJECTS ---------- 6-1
6.1 Create, Open & Save Projects ---------- 6-2
6.2 Create & Edit POUs ---------- 6-2
6.2.1 Create New POU ---------- 6-2
6.2.2 Edit POUs ---------- 6-2
6.2.3 Change POU Properties ---------- 6-3
6.2.4 Delete POUs ---------- 6-4
6.2.5 Use POUs (Copy, Cut & Paste) ---------- 6-5
6.2.6 Create Folders ---------- 6-6
6.2.7 Rename Folders ---------- 6-7
6.2.8 Delete Folders ---------- 6-8
6.3 Program Execution Settings ---------- 6-9
6.4 Register Global Variables ---------- 6-9
6.5 Register DFC ---------- 6-9
6.6 Parameter Settings ---------- 6-9
6.7 Double Coil Check ---------- 6-9
6.8 Conversion ---------- 6-9
6.9 Communication Settings ---------- 6-9
6.10 Write Programs ---------- 6-9
6.11 Read Programs ---------- 6-9
6.12 Verify Projects ---------- 6-10
6.13 Launch Simulation ---------- 6-13
6.14 Option Settings ---------- 6-13
6.15 Display Manual ---------- 6-13
6.16 Version Information ---------- 6-14
6.17 B00/C00 Project Conversion ---------- 6-14
6.18 Import & Export ---------- 6-15
6.18.1 Import Function ---------- 6-15
6.18.2 Export Function ---------- 6-16
6.19 Power Failure Hold Area / Writing Range Check Screen ---------- 6-18
6.20 Check for Writing to OMs Outside Writing Range ---------- 6-20

CHAPTER 7 LD EDIT OPERATION PROCEDURE ---------- 7-1
7.1 LD Program Edit Screen Layout ---------- 7-2
7.1.1 Layout of PLC Programming Software D00 LD/ST Programming Editor/ LD Programming Edit Window ---------- 7-2

7.1.2 LD Program Components ---------- 7-3
7.1.3 Scroll Through Edit Screen ---------- 7-3
7.1.4 Change Display Order in LD Program ---------- 7-4
7.1.5 Change Comment Display ---------- 7-5
7.1.5.1 Change OM Comment Display ---------- 7-5
7.1.5.2 Change inline comment / label comment display ---------- 7-6
7.1.5.3 Change coil remark display ---------- 7-7
7.2 Edit LD Programs ---------- 7-8
7.2.1 Procedure to Create LD Programs ---------- 7-8
7.2.1.1 Contact command input method ---------- 7-9
7.2.1.2 Output command input method ---------- 7-10
7.2.1.3 Application command input method ---------- 7-12
7.2.1.4 CALC command input method ---------- 7-13
7.2.1.5 Label (L) input method ---------- 7-15
7.2.1.6 Revise existing commands ---------- 7-16
7.2.1.7 Insert or delete commands ---------- 7-17
7.2.1.8 Wrapping ---------- 7-19
7.2.1.9 Input and delete vertical or horizontal line ---------- 7-21
7.2.1.10 Insert and delete rung ---------- 7-23
7.2.1.11 Design check ---------- 7-24
7.2.1.12 Conversion ---------- 7-24
7.2.2 Undo ---------- 7-24
7.2.3 Copy, Cut & Paste ---------- 7-25
7.2.3.1 Copy, cut and paste commands ---------- 7-25
7.2.3.2 Copy, cut and paste rungs ---------- 7-27
7.2.3.3 Copy, cut and paste circuit blocks ---------- 7-28
7.2.3.4 Specify range to copy, cut and paste ---------- 7-30
7.2.3.5 Select all ---------- 7-32
7.2.4 Reuse Circuit ---------- 7-33
7.3 Replace All Contacts, OMs & IXs ---------- 7-34
7.3.1 Replace All Contacts a/b ---------- 7-34
7.3.2 Replace All OMs ---------- 7-36
7.3.3 Replace All IXs ---------- 7-38
7.4 Create Comments ---------- 7-42
7.4.1 Create OM Comment ---------- 7-42
7.4.2 Edit OM Comment List ---------- 7-44
7.4.3 Create Inline Comment ---------- 7-47
7.4.4 Create Label Comment ---------- 7-48
7.4.5 Create Coil Remark ---------- 7-49
7.5 Edit Lock ---------- 7-50
7.6 Search ---------- 7-51
7.6.1 Contact/Coil List ---------- 7-51
7.6.2 OM List ---------- 7-52
7.6.3 Timer & Counter List ---------- 7-55
7.6.4 OM Search ---------- 7-56
7.6.4.1 OM search in OM search dialog ---------- 7-56
7.6.4.2 Search circuit blocks for OM specified by cursor position ---------- 7-58
7.6.5 OM Search (Contact) ---------- 7-58
7.6.6 OM Search (Coil) ---------- 7-60
7.6.7 OM Search (All) ---------- 7-61
7.6.8 Command Search ---------- 7-63
7.6.9 Comment Search ---------- 7-64
7.7 Search & Jump ---------- 7-66
7.7.1 Step Number Search ---------- 7-66
7.7.2 OM Jump ---------- 7-67
7.7.3 OM Jump (Contact) ---------- 7-68
7.7.4 OM Jump (Coil) ---------- 7-70
7.7.5 Command Jump ---------- 7-71
7.8 Jump to Beginning, End & Specified Label in LD Program ---------- 7-73
7.9 Launch Simulation ---------- 7-74
7.10 NC(BUILT IN PLC) Monitoring ---------- 7-74

7.11 Run & Stop in NC(BUILT IN PLC) ---------- 7-74
7.12 Monitoring Operations ---------- 7-75
7.12.1 Change Base Number for Monitoring Values ---------- 7-76
7.12.2 Monitoring Registration ---------- 7-76
7.12.3 SET/RST OM ---------- 7-78
7.12.4 Change Current Value ---------- 7-79
7.12.5 Batch Monitoring ---------- 7-81
7.12.5.1 Start batch monitoring ---------- 7-81
7.12.5.2 Change display ---------- 7-84
7.12.5.3 Change OM value ---------- 7-85
7.12.6 Clear all OMs & Variables ---------- 7-86
7.12.7 Monitor CALC Commands ---------- 7-87
7.13 DEBUG Functions ---------- 7-88
7.13.1 Step Execution ---------- 7-88
7.13.2 Stop Condition Settings and Execution ---------- 7-89
7.13.2.1 Stop condition settings ---------- 7-89
7.13.2.2 LD program debug execution ---------- 7-91
7.13.3 Stop Monitoring ---------- 7-92
7.13.3.1 Pause monitoring ---------- 7-92
7.13.3.2 Stop monitoring using bit OM change setting ---------- 7-92
7.14 Print ---------- 7-94
7.15 Display Manual ---------- 7-98

CHAPTER 8 ST EDIT OPERATION PROCEDURE ---------- 8-1
8.1 ST Program & Edit Screen Layout ---------- 8-2
8.1.1 Scroll Through Edit Screen ---------- 8-2
8.1.2 Layout of PLC Programming Software D00 LD/ST Programming Editor/ ST Programming Edit Window ---------- 8-3
8.2 Variable Settings ---------- 8-4
8.2.1 Create & Edit Parameter Variables ---------- 8-4
8.2.1.1 Create new parameter variable ---------- 8-4
8.2.1.2 Parameter variable settings ---------- 8-5
8.2.1.3 Variable edit operations ---------- 8-6
8.2.1.4 Create using copy ---------- 8-7
8.2.1.5 Cut and delete variables ---------- 8-8
8.2.2 Create & Edit External Variables ---------- 8-8
8.2.2.1 Create new external variable ---------- 8-8
8.2.2.2 Change external variable ---------- 8-9
8.2.2.3 Cut and delete external variables ---------- 8-10
8.3 Edit ST Programs ---------- 8-11
8.3.1 Procedure to Create ST Programs ---------- 8-11
8.3.2 Undo ---------- 8-13
8.3.3 Design Check ---------- 8-13
8.3.4 Conversion ---------- 8-13
8.4 Copy, Cut & Paste Character Strings ---------- 8-14
8.4.1 Delete 1 Character ---------- 8-14
8.4.2 Specify Range ---------- 8-14
8.4.3 Copy, Cut, Paste & Delete Operations ---------- 8-15
8.5 Replace ---------- 8-16
8.5.1 Replace Character Strings ---------- 8-16
8.5.2 Replace All OMs ---------- 8-17
8.6 Edit Lock ---------- 8-20
8.7 Search ---------- 8-21
8.7.1 Character String Search ---------- 8-21
8.7.2 OM List ---------- 8-22
8.7.3 OM Search ---------- 8-23
8.7.4 Jump ---------- 8-24
8.8 Launch Simulation ---------- 8-25
8.9 NC(BUILT IN PLC) Monitoring ---------- 8-25

8.10 Run & Stop in NC(BUILT IN PLC) ---------- 8-25
8.11 Monitoring Operations ---------- 8-26
8.11.1 Change Base Number for Monitoring Values ---------- 8-27
8.11.2 Monitoring & Changing Current Variable Values ---------- 8-27
8.11.2.1 Monitoring parameter variables and external variables ---------- 8-27
8.11.2.2 Array variable monitoring ---------- 8-28
8.11.3 Monitoring Registration ---------- 8-29
8.11.4 SET/RST OM ---------- 8-30
8.11.5 Change Current Values ---------- 8-31
8.11.6 Batch Monitoring ---------- 8-32
8.11.7 Clear All OMs & Variables ---------- 8-32
8.12 DEBUG Function ---------- 8-33
8.12.1 Step Execution ---------- 8-33
8.12.2 Setting and Execution of Stop Position Based on Breakpoint Setting ---------- 8-34
8.12.2.1 Breakpoint settings ---------- 8-34
8.12.2.2 ST Program debug execution ---------- 8-35
8.12.3 Stop Monitoring Using OM Change Setting ---------- 8-36
8.13 Print ---------- 8-37
8.14 Display Manuals ---------- 8-39

CHAPTER 9 FBD EDIT OPERATION PROCEDURE ---------- 9-1
9.1 FBD Program & Edit Screen Layout ---------- 9-2
9.1.1 Layout of PLC Programming software D00 FBD Programming Editor ---------- 9-3
9.2 Variable Settings ---------- 9-4
9.2.1 Create & Edit Parameter Variables ---------- 9-4
9.2.1.1 Create new parameter variable ---------- 9-4
9.2.1.2 Parameter variable settings ---------- 9-5
9.2.1.3 Variable edit operations ---------- 9-7
9.2.1.4 Re-use (copy) ---------- 9-8
9.2.1.5 Cut and delete variables ---------- 9-8
9.2.2 Create & Edit External Variables ---------- 9-9
9.2.2.1 Create new external variable ---------- 9-9
9.2.2.2 Change external variable ---------- 9-11
9.2.2.3 Cut and delete external variables ---------- 9-12
9.3 Edit FBD Programs ---------- 9-13
9.3.1 Operations for Creating FBD Programs ---------- 9-13
9.3.1.1 Enable Input/Output for Function Box (EN/ENO) ---------- 9-15
9.3.1.2 OM Settings in OM Box ---------- 9-16
9.3.1.3 Value Setting in Literal (Constant) Box ---------- 9-17
9.3.1.4 Box size adjustment ---------- 9-18
9.3.1.5 Drag & Move Box ---------- 9-18
9.3.1.6 Connector ---------- 9-19
9.3.1.7 Delete boxes and connectors ---------- 9-20
9.3.1.8 Insert and delete networks ---------- 9-20
9.3.1.9 Change network size ---------- 9-22
9.3.1.10 Change setting for variable, user defined function or system defined function ---------- 9-22
9.3.1.11 Re-assign box ---------- 9-24
9.3.2 Undo ---------- 9-24
9.3.3 Conversion ---------- 9-25
9.4 Copy, Cut, Paste & Delete ---------- 9-26
9.4.1 Specify Range ---------- 9-26
9.4.1.1 When specifying only a function box ---------- 9-26
9.4.1.2 When specifying multiple function boxes or connectors ---------- 9-26
9.4.1.3 When selecting all items in a network ---------- 9-27
9.4.1.4 When selecting networks ---------- 9-27
9.4.2 Copy, Cut, Paste & Delete Operations ---------- 9-28
9.5 Replace ---------- 9-29
9.5.1 Replace All OMs ---------- 9-29
9.6 Search ---------- 9-31
9.6.1 Character String Search ---------- 9-31

9.6.2 Jump ---------- 9-32
9.6.3 OM List ---------- 9-33
9.6.4 OM Search ---------- 9-34
9.7 Edit Lock ---------- 9-36
9.8 Launch Simulation ---------- 9-36
9.9 Monitoring FBD Program in NC(BUILT IN PLC) ---------- 9-37
9.10 Run & Stop in NC(BUILT IN PLC) ---------- 9-37
9.11 Monitoring Operations ---------- 9-38
9.11.1 Change Base Number for Monitoring Values ---------- 9-39
9.11.2 Monitoring & Changing Current Variable Values ---------- 9-40
9.11.2.1 Monitoring parameter variables and external variables ---------- 9-40
9.11.2.2 Array variable monitoring ---------- 9-42
9.11.3 SET/RST OM & Change Current OM Value ---------- 9-43
9.11.4 Clear All OMs & Variables ---------- 9-44
9.12 DEBUG Function ---------- 9-45
9.12.1 Step Execution ---------- 9-45
9.12.2 Setting and Execution of Stop Position Based on Breakpoint Setting ---------- 9-46
9.12.2.1 Breakpoint settings ---------- 9-46
9.12.2.2 FBD program debug execution ---------- 9-47
9.12.3 Stop Monitoring Using OM Change Setting ---------- 9-48
9.13 Print ---------- 9-49
9.14 Display Manuals ---------- 9-51

CHAPTER 10 FUNCTIONS FOR CREATING PROGRAM ---------- 10-1
10.1 Undo ---------- 10-2
10.2 Copy, Cut & Paste ---------- 10-3
10.2.1 Copy, Cut & Paste Variables (LD, ST & FBD Languages) ---------- 10-3
10.2.2 Copy, Cut & Paste POUs ---------- 10-5
10.2.3 Copy, Cut & Paste Circuit Symbols (LD Language) ---------- 10-5
10.2.4 Copy, Cut & Paste One Line (LD Language) ---------- 10-7
10.2.5 Copy, Cut & Paste Circuit Blocks (LD Language) ---------- 10-8
10.2.6 Reuse Circuit (LD Language) ---------- 10-9
10.2.7 Copy, Cut & Paste (Character String) in ST Program (ST Language) ---------- 10-10
10.2.8 Copy, Cut & Paste Boxes (FBD Language) ---------- 10-10
10.2.9 Copy, Cut & Paste Networks (FBD Language) ---------- 10-12
10.3 Delete ---------- 10-15
10.4 Search & Jump ---------- 10-16
10.4.1 Contact/Coil List (LD Language) ---------- 10-17
10.4.2 OM List (LD Language) ---------- 10-18
10.4.3 Timer/Counter List (LD Language) ---------- 10-19
10.4.4 OM Search (LD Language) ---------- 10-20
10.4.5 OM Search (Contact) (LD Language) ---------- 10-21
10.4.6 OM Search (Coil) (LD Language) ---------- 10-22
10.4.7 OM Search (All) (LD Language) ---------- 10-23
10.4.8 Command Search (LD Language) ---------- 10-24
10.4.9 Search Up & Search Down (LD Language) ---------- 10-25
10.4.10 Step Number Search (LD Language) ---------- 10-26
10.4.11 OM Jump (LD Language) ---------- 10-27
10.4.12 OM Jump (Contact) (LD Language) ---------- 10-28
10.4.13 OM Jump (Coil) (LD Language) ---------- 10-29
10.4.14 Command Jump (LD Language) ---------- 10-30
10.4.15 Character String Search (ST Language) ---------- 10-31
10.4.16 OM List (ST Language) ---------- 10-32
10.4.17 OM Search (ST Language) ---------- 10-33
10.4.18 Comment Search (LD Language) ---------- 10-34
10.4.19 Character String Search (FBD Language) ---------- 10-35
10.4.20 OM List (FBD Language) ---------- 10-36
10.4.21 OM Search (FBD Language) ---------- 10-37
10.5 Replace ---------- 10-39

10.5.1 Replace All Contacts a/b (LD Language) ----------10-39
10.5.2 Replace All OMs (LD, ST and FBD Languages) ----------10-40
10.5.3 Replace All IXs (LD Language) ----------10-42
10.5.4 Replace Character String (ST Language) ----------10-43
10.6 Jump ----------10-45
10.7 Create Comments & Drawings ----------10-47
10.8 Edit Lock ----------10-49
10.9 Program Conversion Function ----------10-50
10.9.1 Conversion ----------10-50
10.9.2 Conversion Mode Settings ----------10-53
10.10 Program Check Functions ----------10-54
10.10.1 Double Coil Check ----------10-54
10.10.2 Design Check ----------10-56
10.10.3 File Checksum Function ----------10-58
10.11 Display Function ----------10-59
10.11.1 Display Comments (LD Language) ----------10-59
10.11.2 Step Number Display (LD Language) ----------10-60
10.11.3 Search Results Display (LD Language) ----------10-61
10.12 Print Function ----------10-62
10.12.1 Print Content ----------10-62
10.12.2 Print Range ----------10-62
10.12.3 Print Preview Screen ----------10-62
10.13 Verify Projects ----------10-63
10.14 B00/C00 Project Conversion ----------10-66
10.15 Re-assign Boxes (FBD Language) ----------10-67
10.16 Import & Export ----------10-68

CHAPTER 11 PROGRAM EXECUTION FUNCTIONS ----------11-1
11.1 Write & Read Programs ---------- 11-2
11.1.1 Write programs ---------- 11-2
11.1.2 Read programs ---------- 11-3
11.1.3 Write During Run ---------- 11-3
11.1.3.1 What is write during run? ---------- 11-3
11.1.3.2 Conditions for write during run ---------- 11-4
11.1.3.3 Execute write during run operation ---------- 11-4
11.1.3.4 Special notes and restrictions ---------- 11-5
11.2 Program Execution ---------- 11-6
11.3 RUN & STOP Mode ---------- 11-6
11.4 Monitoring Function of Program Operation ---------- 11-7
11.4.1 Start, Stop & End Monitoring ---------- 11-8
11.4.1.1 Monitoring location ---------- 11-8
11.4.1.2 Start, stop & end monitoring ---------- 11-8
11.4.2 Display in Monitor Mode ---------- 11-9
11.4.2.1 Status bar ----------11-10
11.4.2.2 Monitoring status bar ----------11-10
11.4.2.3 Program editing area (LD language) ----------11-11
11.4.2.4 Program editing area (ST language) ----------11-12
11.4.2.5 Program editing area (FBD language) ----------11-13
11.4.2.6 Monitoring registration (LD and ST languages) ----------11-14
11.4.2.7 Variable editing area (ST and FBD languages) ----------11-14
11.4.2.8 Array variable monitoring ----------11-14
11.4.2.9 Message display area (ST and FBD languages) ----------11-15
11.4.3 Batch monitoring (LD and ST languages) ----------11-15
11.4.4 Monitoring registration (LD and ST Languages) ----------11-17
11.4.5 SET/RESET OM (LD and ST Languages) ----------11-18
11.4.6 Change Current OM Value ----------11-19
11.4.7 Change Current Variable Value ----------11-20
11.4.8 Clear all OMs & Variables ----------11-21
11.4.9 Rising Edge Trigger & Falling Edge Trigger ----------11-22

11.4.10 Change Monitoring Base Number ---------- 11-23
11.4.11 Restrictions ---------- 11-24
11.5 Debug Function ---------- 11-25
11.5.1 Start & End Debug Function ---------- 11-25
11.5.2 Step Execution ---------- 11-26
11.5.3 Program Execution Stop Using Breakpoint/Stop Step ---------- 11-29
11.5.4 Stop Using Specified Condition (LD Language) ---------- 11-31
11.6 Simulation Function ---------- 11-32

CHAPTER 12 COMMANDS (FBD) ---------- 12-1
12.1 FBD Language Specifications ---------- 12-2
12.1.1 Create FBD Programs ---------- 12-2
12.1.2 Variables & OMs ---------- 12-4
12.1.2.1 List of data types ---------- 12-4
12.1.2.2 Usable OMs ---------- 12-5
12.1.2.3 Specifying indexing and digit-specified access for OMs ---------- 12-5
12.1.2.4 External variable ---------- 12-6
12.1.2.5 Array variables ---------- 12-6
12.1.3 Constants ---------- 12-8
12.1.4 Input & Output Connections ---------- 12-9
12.1.5 EN/ENO ---------- 12-10
12.1.6 FBD Notation in Networks ---------- 12-11
12.1.7 Feedback Paths ---------- 12-11
12.1.8 Reserved Words and Reserved Characters ---------- 12-12
12.2 FBD Language Standard Functions ---------- 12-13
12.2.1 List of FBD Language Standard Functions ---------- 12-13
12.2.2 Detailed Descriptions of FBD Language Standard Functions ---------- 12-15
12.2.2.1 Data type conversion function ---------- 12-16
12.2.2.2 Binary copy function ---------- 12-20
12.2.2.3 Arithmetic operation function ---------- 12-22
12.2.2.4 Numerical value operation function ---------- 12-34
12.2.2.5 Bit shift function ---------- 12-45
12.2.2.6 Bit operation function ---------- 12-53
12.2.2.7 Selection function ---------- 12-57
12.2.2.8 Comparison function ---------- 12-67
12.2.2.9 Character string function ---------- 12-79
12.2.2.10 Time data type function ---------- 12-96
12.2.2.11 Expanded function ---------- 12-110
12.2.3 Create & Use User Defined Function / System Defined Function ---------- 12-114
12.2.3.1 Return value box ---------- 12-114
12.2.3.2 User defined function and system defined function ---------- 12-114
12.3 Error Message List During FBD Program Conversion ---------- 12-116

CHAPTER 1	INSTALLATION	1
CHAPTER 2	SYSTEM CONFIGURATION	2
CHAPTER 3	PROCEDURE FOR BASIC OPERATIONS	3
CHAPTER 4	LAUNCH & EXIT EDITOR	4
CHAPTER 5	PROCEDURE TO CREATE PROGRAMS	5
CHAPTER 6	BASIC OPERATION PROCEDURE FOR PROJECTS	6
CHAPTER 7	LD EDIT OPERATION PROCEDURE	7
CHAPTER 8	ST EDIT OPERATION PROCEDURE	8
CHAPTER 9	FBD EDIT OPERATION PROCEDURE	9
CHAPTER 10	FUNCTIONS FOR CREATING PROGRAM	10
CHAPTER 11	PROGRAM EXECUTION FUNCTIONS	11
CHAPTER 12	COMMANDS (FBD)	12

(This page was intentionally left blank.)

CHAPTER 1

INSTALLATION

1.1 Before Installation
1.2 PLC Programming Software D00 EDITOR Installation
1.3 Uninstall PLC Programming Software D00

1.1 Before Installation

Operation requirements for PLC Programming software D00 EDITOR

Make sure that the computer meets the following requirements before installing the PLC Programming software D00 EDITOR.

Required system

Item	System requirement
Compatible OS	Windows 8.1 32bit/64bit Windows 10 32bit/64bit
CPU *1	1 GHz processor or higher is recommended
Memory *1	Windows 8.1/10 32-bit, 2 GB or higher Windows 8.1/10 64-bit, 4 GB or higher
Computer	Computer with the aforementioned OS installed. CD-ROM drive is required to install the PLC Programming software D00.

*1 Only use a computer with a specification that meets the OS system requirements.

(NOTE) Log in as an administrator for Windows 8.1 or 10.

1.2 PLC Programming Software D00 EDITOR Installation

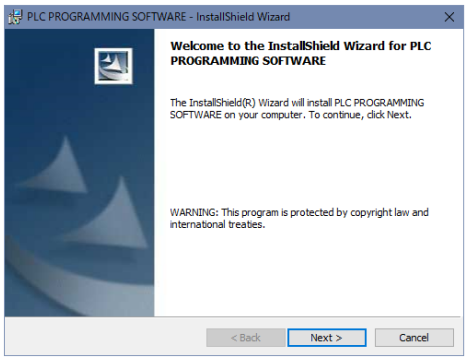
The PLC Programming software D00 EDITOR is installed onto the hard disk.
Run the installer program that was provided to install the PLC Programming software D00 EDITOR.

Operation procedure

1. Start up Windows 8.1/10. (This procedure uses Windows 10 as an example.)

2. Insert the CD-ROM from PLC Programming software D00 starter kit into the CD-ROM drive. Launch “PLC Programming software D00”-“setup.exe” that is found in the PLC Programming software D00 CD-ROM, and then run the setup program. If a confirmation dialog appears for the user account control, click on <Yes>.

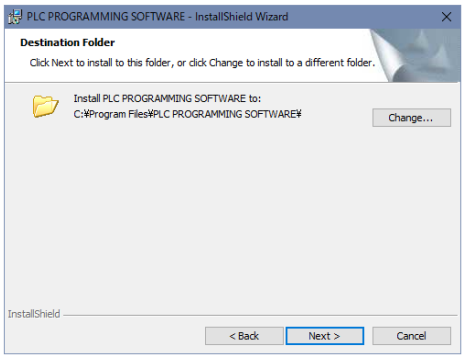
3. The setup screen for the PLC Programming software D00 is displayed.
   Click on the <Next> button. Or, click on the <Cancel> button to cancel the PLC Programming software D00 setup.



4. A screen is displayed for specifying the location to install the software. Select a directory to install the software. The following are the default locations to install.

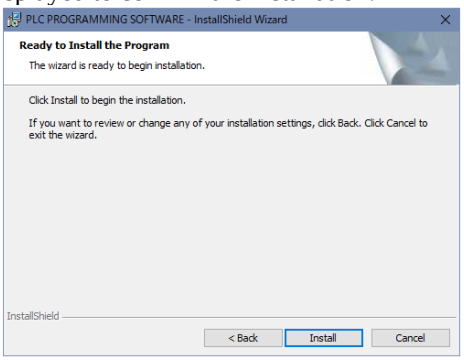
OS	Default install location
Windows 8.1 32bit Windows10 32bit	C:¥Program Files¥PLC PROGRAMMING SOFTWARE
Windows 8.1 64bit Windows10 64bit	C:¥Program Files (x86)¥PLC PROGRAMMING SOFTWARE

To change the installation directory, click on the <Browse> button and specify the desired drive and directory (if a directory is specified that does not exist, one is automatically created).



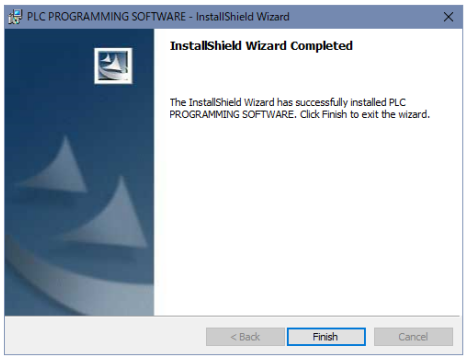
After specifying the directory, click on the <Next> button.

5. A screen is displayed to confirm the installation.



After clicking on the <Install> button, installation begins and the files are copied.
A status bar is displayed to show the progress during installation.

6. When the installation is complete, a finish screen is displayed. Click on the <Finish> button to finish and exit the installation.



(NOTE) When a message is displayed prompting the user to restart the computer, always restart the computer before using the PLC Programming software D00.

Installation of Microsoft Visual C++ Runtime Components

Next, the runtime components for Microsoft Visual C++ will install.

1. In the installation CD-ROM, “vcredist_x86.exe” (for a 32-bit OS) or “vcredist_x64.exe” (for a 64-bit OS) in the “Microsoft Visual C++ Redistributable Package” for the PLC Programming software D00 will launch.

2. “Microsoft Visual C++ 2012 Redistributable(x86)” or “Microsoft Visual C++ 2012 Redistributable(x64)” is installed.

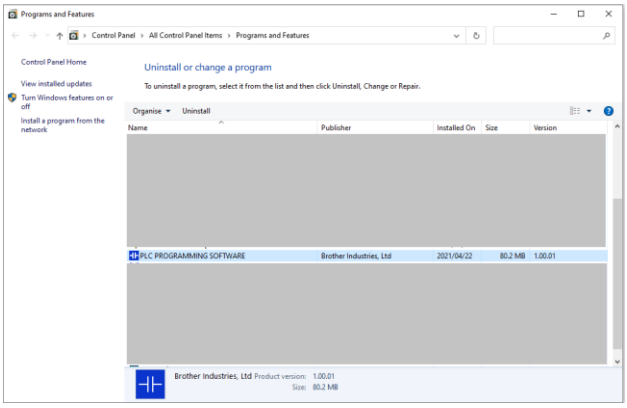
1.3 Uninstall PLC Programming Software D00

Operation procedure

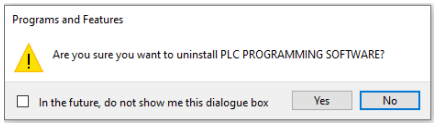
1. Start up Windows 8.1/10. (This procedure uses Windows 10 as an example.)

2. Click on the <Start> button, and then select <Control Panel> under the <Windows System Tools>.
   From the <Control Panel>, select <Uninstall a program>.

3. When a list of currently installed programs is displayed, select <PLC Programming Software D00> and then click on the <Uninstall> button located above.



4. When the following dialog is displayed, select <Yes>.



If a confirmation dialog appears for the user account control, click on <Yes>.

CHAPTER 2

SYSTEM CONFIGURATION

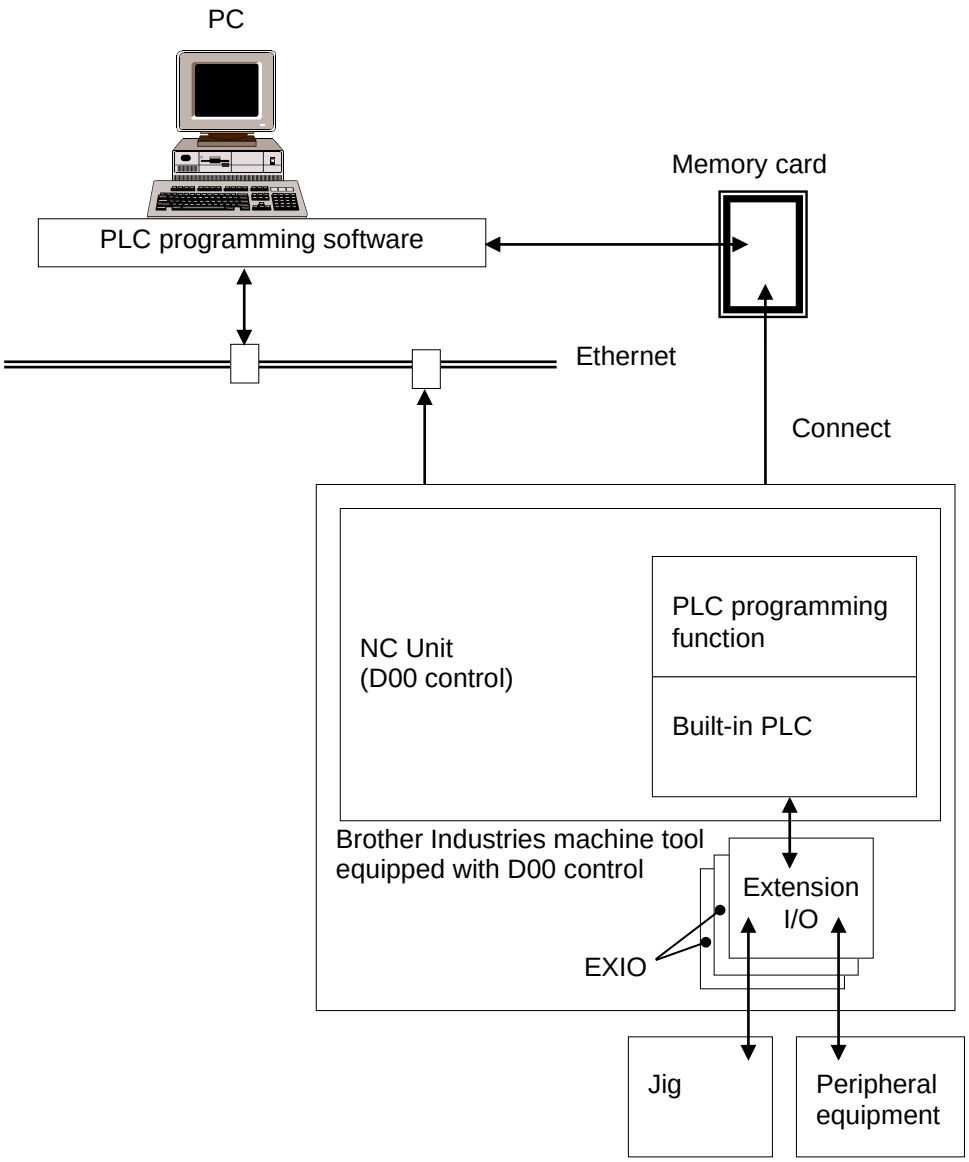
2.1 Overall System
2.2 File Structure

2.1 Overall System

*　Read this document together with the PLC System Manual.

This product is the PLC programming software that is designed for editing PLC programs used with Brother Industries machine tools that have a built-in PLC function.

The PLC programming software can be used to change PLC programs within the built-in PLC function by accessing them either through the Ethernet or a memory card.



2.2 File Structure

* Read this document together with the PLC System Manual.

In the PLC programming software, ladder projects can be created and edited. In the ladder project that is created, there are folders and files that cannot be viewed in the Windows default settings. The customer is able to use the PLC programming software to access these files. The user can export files from the ladder project to make them portable and then output the files in a format (PLCPRJ.BPP) so that they can be imported into the NC. Use this file format as well for backing up data, etc.

(NOTE 1) Do not copy, move, delete, rename, change the attributes or perform any other direct operations on the folders or files created in the PLC programming software. Otherwise, it will affect the structure that links each individual file and operation will not run or execute properly.

(NOTE 2) Do not intermix and use the files in a given project with other project files.

Functions used for data transfer between equipment and devices

Export: Exports files (PLCPRJ.BPP) for a project (created in the PLC programming software) in a file format so that the files can be input into the NC.
Or, it exports files (PLCPRJ.BPP) for a project in the built-in PLC system (in the NC unit) using a file format so that the PLC programming software can import the files.

Import: Inputs files (PLCPRJ.BPP) from the PLC programming software into the NC unit built-in PLC system.
Or, it imports files (PLCPRJ.BPP) into the PLC programming software from the NC unit built-in PLC system.

(This page was intentionally left blank.)

CHAPTER 3

PROCEDURE FOR BASIC OPERATIONS

3.1 Export Ladder from NC Unit
3.2 Launch Application & Import Project
3.3 Edit POUs
3.4 Convert POUs
3.5 Print POUs
3.6 Save Project
3.7 Export Project
3.8 Import to NC Unit

3.1 Export Ladder from NC Unit

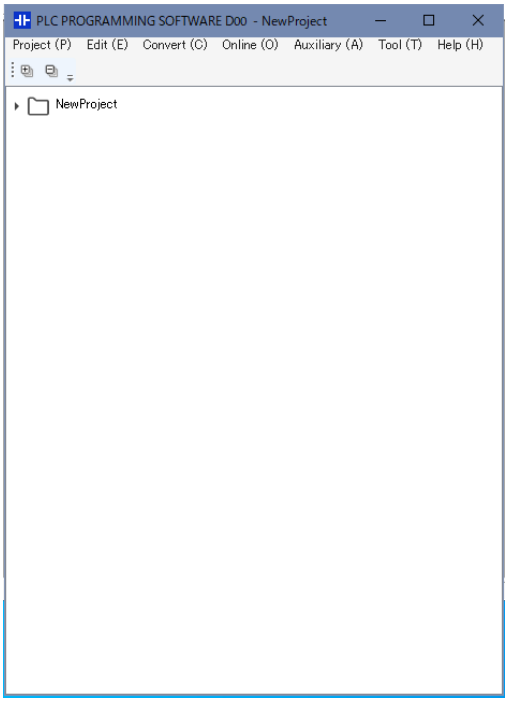
This section describes how to get a ladder from the NC using a memory card.

Items needed: Memory card (USB memory)

1. Connect a memory card to the memory card interface on the operation panel.
2. Display the external I/O (data bank) screen on the operation panel for the machine.
   (1) Display the <Program edit menu> screen.
   Press the **[Edit]** key.
   Press the **[PRGRM]** key.
   (2) Display the <External I/O menu> screen.
   Select <External I/O> from the menu.
   (3) Display the <Data bank (external program I/O)> screen.
   Select <Data bank> from the menu.
   Make sure that the connection reads <Memory card>. If the connecting destination or device is different, change the connecting destination so it matches.
3. In the listed items, move the cursor to <PLC program>
4. Enter “1” into the data input field, and press the **[Output]** function.
   The export files in the PLC project are output to the memory card.

3.2 Launch Application & Import Project

1. Save the file (PLCPRJ.BPP), which was output to the memory card in the previous section, to the computer.
2. To launch the application, select <PLC PROGRAMING SOFTWARE> located in the <START MENU> under <ALL PROGRAMS>.
   The Project Explorer window is displayed as shown below.



3. Create a new project.
   (1) Select <Create new project> from the project menu.
   (2) Enter the project name and the folder to save the project, and then click on the <OK> button.
   The edit screen is launched.
   (NOTE) Use the default folder for saving the project.
4. Import the file (PLCPRJ.BPP) saved in the computer to the new project (that was created).
   Refer to “6.18 Import & Export”

3.3 Edit POUs

Change the project that was saved in the previous section.
Select a POU to edit from the Project Explorer window, or create a new one. Editing is carried out on the edit screen. Refer to “5.6 Program Edit” for further details on each editing function.
The following POUs are saved in the project per the factory-default settings.

- High speed control program

Program10 (H_SPEEDIO_SYSTEM)	Do not change this POU. This POU (for manufacturer use) is overwritten when the firmware version is updated.

- Standard control program

Program 1 (S1LADDER)	This POU is used by the customer to change the indicator, etc.
Program10 (S_SPEEDIO_SYSTEM)	Do not change this POU. This POU (for manufacturer use) is overwritten when the firmware version is updated.

(NOTE 1) Use high speed control programs (1 to 9) and standard control programs (1 to 9) for jig control.

(NOTE 2) “Standard control program 10 (S_SPEEDIO_SYSTEM)” is already loaded and included in the factory-set ladder programs for the following Brother Industries issued parts and items: coolant unit (including chip shower, tool wash, jig shower and coolant valve), automatic door, CTS device, spindle air blow, start/pallet start switch, chip conveyor, machine lights, cyclone filter, chip auger, LB internal start input and LB internal reset input. Machine operation is based on this control program setup. Do not make any changes as described below for the Brother issued devices and parts. If the user changes a program, the product may no longer be covered under warranty.

- Do not change the “Standard control program 10 (S_SPEEDIO_SYSTEM)”. Do not change related parameters.
- Do not change any contact used in the “Standard control program 10 (S_SPEEDIO_SYSTEM)”.
- Do not perform the debug function (stop, ON/OFF for related contacts, etc.) for “Standard control program 10 (S_SPEEDIO_SYSTEM)” while this function is operating.

3.4 Convert POUs

The conversion operation is required when creating or modifying a POU. The user can use <Convert (C) F4> on the edit screen to convert a POU while in edit mode.
Refer to “5.9 Conversion” for further details.

* The POU edit and conversion operations must be carried out as a set until the editing is finished.

3.5 Print POUs

When adding a new POU program or when making a change to one, be sure to print out a copy of that ladder circuit and post it near the machine. The user can print out a ladder circuit by selecting <Project> from the edit screen menu and then selecting <Print>.
Refer to “7.14 Print”, “8.13 Print” or “9.13 Print”.

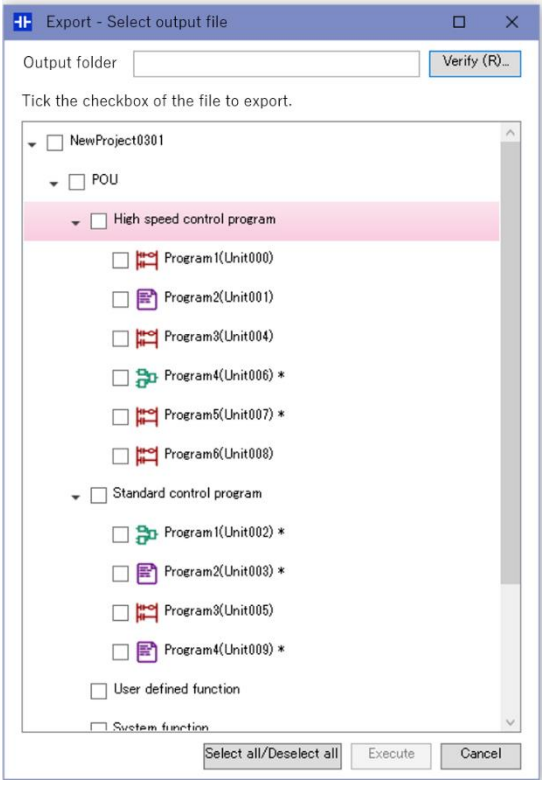
* The POU edit and conversion operations must be carried out as a set until the editing is finished.

3.6 Save Project

After the POU editing and conversion is complete, save the project.
Select <Project> from the menu in the Project Explorer window, and then select <Save all> to save all the POUs.

3.7 Export Project

This function exports a project.
Select <Project> from the menu in the Project Explorer window, and then select <Export> to create a file (PLCPRJ.BPP) for that data in a given folder.



Tick all the checkboxes to select all the files.

3.8 Import to NC Unit

1. Put the file (PLCPRJ.BPP) created in the previous section into the memory card, and then connect the memory card to the memory card interface on the operation panel.
   (NOTE) The data protection needs to be set to an operation level that allows the ladder program to be input. Set the user parameter <PLC ladder program protection> to <0: Invalid>.
2. Display the external I/O (data bank) screen on the operation panel for the machine.
   (1) Display the <Program edit menu> screen.
   Press the **[Edit]** key.
   Press the **[PRGRM]** key.
   (2) Display the <External I/O menu> screen.
   Select <External I/O> from the menu.
   (3) Display the <Data bank (external program I/O)> screen.
   Select <Data bank> from the menu.
   Make sure that the connection reads <Memory card>. If the connecting destination or device is different, change the connecting destination so it matches.
3. In the listed items, move the cursor to <PLC program>
4. Enter “1” into the data input field, and press the **[Input]** function. When the overwrite confirmation message appears, select <Yes>.
5. Turn OFF the power for this machine, or press the **[RST]** key. This will start operation for the PLC project that was changed.
   (NOTE 1) After importing files, the power ON time (after turning ON the power again) will take longer than normal, or the light on the reset LED (after pressing the **[RST]** key) will light up longer than normal.
   (NOTE 2) The customer must take the proper safety measures (such as press the **[EMERGENCY]** switch to ensure operation does not occur) after the import operation.

CHAPTER 4

LAUNCH & EXIT EDITOR

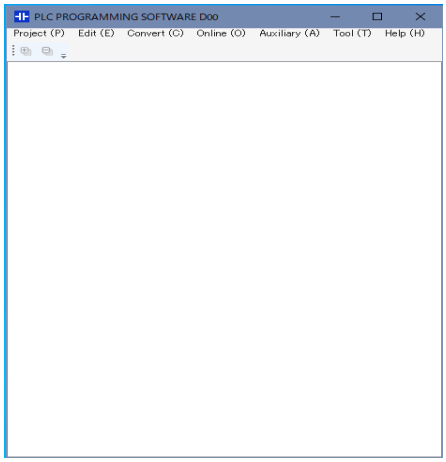
4.1 Launch PLC Programming Software D00 EDITOR
4.2 Close Edit Screen
4.3 Exit PLC Programming Software D00 EDITOR

4.1 Launch PLC Programming Software D00 EDITOR

This section describes the startup procedure for the PLC Programming software D00 EDITOR.

Select <PLC PROGRAMMING SOFTWARE> → <PLC Programming software D00> from the Windows start menu in order to launch the PLC Programming software D00 EDITOR. (When using Windows 10)

When PLC Programming software D00 EDITOR is launched, only the PLC Programming software D00 is displayed.



Use the PLC Programming software D00 to create a new project and read it.
For further details, refer to “Chapter 5 Procedure to create programs”.

4.2 Close Edit Screen

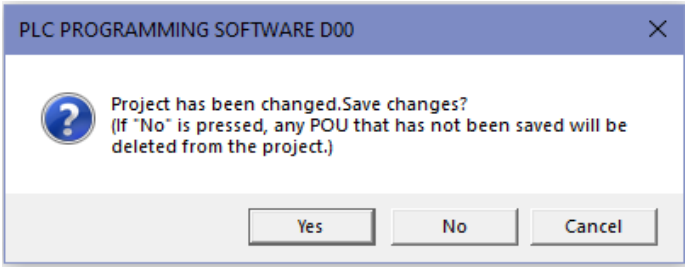
This section describes the operations for closing the PLC Programming software D00, the LD program edit window, the ST program edit window and the FBD Programming Editor.

- Close operations using the PLC Programming software D00

  Operation procedure

  Perform one of the following operations.
  - Select <End> under the <Project> menu.
  - Click on on the top-right corner in the PLC Programming software D00.
  - Select <Close> by right-clicking on the title bar.
  - Press the **[Alt]** + **[F4]** keys when PLC Programming software D00 is the active window.

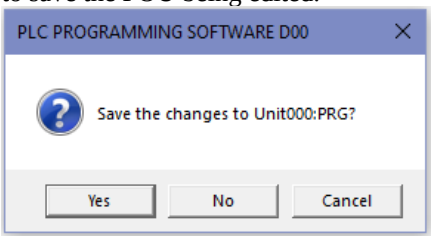
  When PLC Programming software D00 is closed, if a project still being edited is not saved, a confirmation message appears asking whether or not to save the project.



  When <Yes> is selected, the project and POU being edited are saved and then the project is closed. At this time, if the POU has not been converted, it will be saved as is.
  When <No> is selected, the project being edited is not saved and the project is closed. At this time, if there is a POU that has never been saved, that POU is deleted.
  When <Cancel> is selected, saving the project and exiting PLC Programming software D00 are both cancelled.

- Close operations using PLC Programming software D00 LD/ST Programming Editor
  Operation procedure
  Perform one of the following operations.
  - Select <End> under the <Project> menu.
  - Click on on the top-right corner in the PLC Programming software D00 LD/ST Programming Editor.
  - Select <Close> by right-clicking on the title bar.
  - Press the **[Alt]** + **[F4]** keys when PLC Programming software D00 LD/ST Programming Editor is the active window.

  If the POU being edited is not saved, a confirmation message appears asking whether or not to to save the POU being edited.



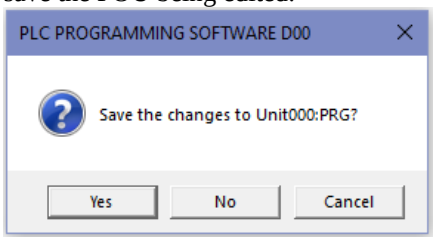
  When <Yes> is selected, the POU being edited is saved and then it exits the PLC Programming software D00 LD/ST Programming Editor. At this time, if the POU has not been converted, it will be saved as is.
  When <No> is selected, the POU is closed without saving it, and then it exits the PLC Programming software D00 LD/ST Programming Editor.
  When <Cancel> is selected, saving the POU and exiting PLC Programming software D00 LD/ST Programming Editor are both cancelled.

  When there are multiple POUs being edited, once the user confirms whether or not to save all of the POUs, it exits PLC Programming software D00 LD/ST Programming Editor. However, if <Cancel> is selected just one time, then the exit processing is cancelled at that point.

(NOTE) <Save all> can be used in the PLC Programming software D00 to save all of the POUs that are still being created.

- Close operations in the LD program edit window and ST program edit window
  Operation procedure
  Perform one of the following operations.。
  - When the LD program edit window or the ST program edit window is active, select <Close> from the <Project> menu in the PLC Programming software D00 LD/ST Programming Editor.
  - Click on on the top-right corner in the LD program edit window or in the ST program edit window.

  If the POU being edited is not saved, a confirmation message appears asking whether or not to save the POU being edited.



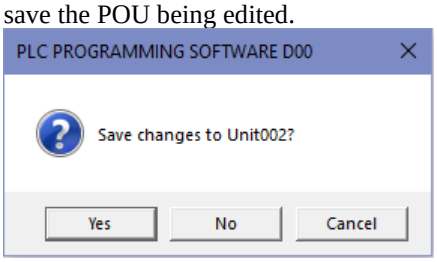
  When <Yes> is selected, the POU being edited is saved and then the edit window is closed. At this time, if the POU has not been converted, it will be saved as is.
  When <No> is selected, the edit window is closed for the POU without saving it.
  When <Cancel> is selected, the POU is not saved and the edit window does not close.

(NOTE) <Save all> can be used in the PLC Programming software D00 to save all of the POUs that are still being created.

- Close operation using FBD Programming Editor
  If a program is not saved after performing a close operation, the user is prompted with a message to save the program, and then it exits the FBD Programming Editor.
  If a program is not converted, it will be saved in an unconverted state.

  Operation procedure
  Perform one of the following operations.
  - Select <End> under the <Project> menu.
  - Click on on the top-right corner in the FBD Programming Editor.
  - Select <Close> by right-clicking on the title bar.
  - Press the **[Alt]** + **[F4]** keys when FBD Programming Editor is the active window.

  If the POU being edited is not saved, a confirmation message appears asking whether or not to save the POU being edited.



  When <Yes> is selected, the POU being edited is saved and then the FBD Programming Editor is closed. At this time, if the FBD program has not been converted, it will be saved as is.
  When <No> is selected, the POU is not saved and the FBD Programming Editor is closed.
  When <Cancel> is selected, saving the POU and exiting FBD Programming Editor are both cancelled.

4.3 Exit PLC Programming Software D00 EDITOR

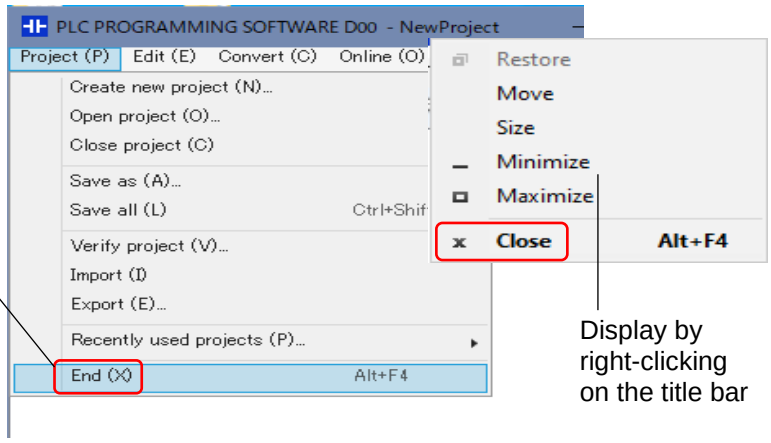
This section describes how to exit the PLC Programming software D00 EDITOR.

To exit the PLC Programming software D00 EDITOR, the user must exit the PLC Programming software D00 and then all the windows will close down.
When performing the exit operation, if the program is not saved, then a confirmation message appears asking whether or not to save the program. Thereafter, it exits the PLC Programming software D00, the LD/ST Programming Editor and the FBD Programming Editor.
If a POU is not converted when saving, it will be saved in an unconverted state.

Operation procedure

Perform one of the following operations.

- Select <End> under the <Project> menu.
- Click on on the top-right corner in the PLC Programming software D00.
- Select <Close> by right-clicking on the title bar.
- Press the **[Alt]** + **[F4]** keys when PLC Programming software D00 is the active window.



CHAPTER 5

PROCEDURE TO CREATE PROGRAMS

5

5.1 Create Projects
5.2 Create POUs (LD/ST/FBD Programs)
5.3 Register Global Variables
5.4 Parameter Settings
5.5 Register DFC (Dynamic Function Call)
5.6 Program Edit
5.7 Program Check
5.8 Program Execution Settings
5.9 Conversion
5.10 Write/Read Programs
5.11 Launch Simulation
5.12 NC(BUILT IN PLC) Monitoring
5.13 Run & Stop in NC(BUILT IN PLC)
5.14 POU Debug Operation
5.15 Close Project

5.1 Create Projects

This section describes the operations for creating a project in the PLC Programming software D00 EDITOR.
The following shows the basic flow of operations.

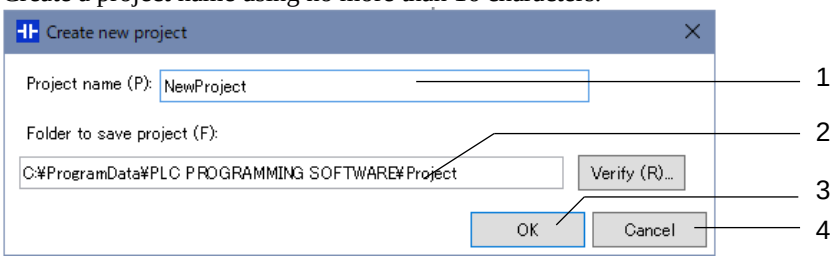
Create new project Launch PLC Programming software D00 EDITOR (PLC Programming software D00) Create new project	Refer to “5.1.1 Create new projects” for further details.
⇩	
Create control program Create new POU Register variable POU programming Program execution setting (POU execution order setting) Parameter setting Double coil check	Refer to “5.2 Create POUs (LD/ST/FBD programs)”, “5.3 Register global variables”, “5.4 Parameter settings”, “5.6 Program edit”, “5.7 Program check” and “5.8 Program execution settings”
⇩	
Convert	Refer to “5.9 Conversion” for further details.
⇩	
Operation check using PC simulation function Execute simulation function Check status change in OM/variable Execute debug function	Refer to “5.10 Write/Read programs”, “5.11 Launch simulation”, “5.12 NC(BUILT IN PLC) monitoring”, “5.13 Run & Stop in NC(BUILT IN PLC)” and “5.14 POU debug operation”
⇩	
Operation check on target board Communication settings Write program to NC(BUILT IN PLC) Execute debug function	Refer to “5.10 Write/Read programs”, “5.12 NC(BUILT IN PLC) monitoring”, “5.13 Run & Stop in NC(BUILT IN PLC)” and “5.14 POU debug operation”

5.1.1 Create New Projects

The PLC Programming software D00 projects are created in the PLC Programming software D00.

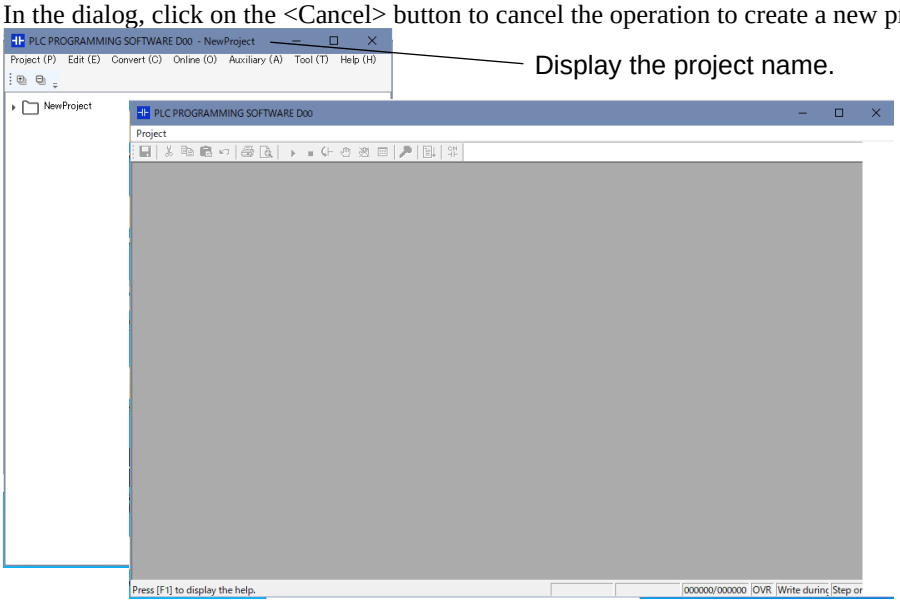
Operation procedure

- When the B00 and C00 project function is disabled
  This occurs when the checkbox is not ticked for the <Enable function> - <B00 and C00 project> (under the <Auxiliary> menu, go to <Option settings> and then the <Customize settings> tab to access).
  1. Select <Create new project> from the project menu in the PLC Programming software D00.
  2. Set the project name.
     Create a project name using no more than 16 characters.

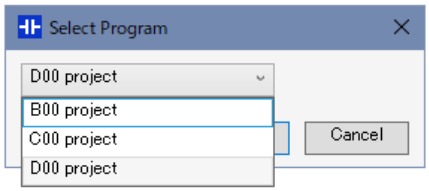


No.	Item	Description
1	Project name	Set a project name using no more than 16 characters. The following symbols cannot be used: ¥, /, :, *, ?, ", <, > and |
2	Folder to save project	Specifies the folder for saving the project file. After pressing the <Browse> button to display the <Folder selection> dialog, the user can select a folder.
3	<OK> button	Creates a new project and closes the dialog.
4	<Cancel> button	Cancels the operation to create a new project and closes the dialog.

  3. When the dialog to create a new project appears, click on the <OK> button.
     On the title bar of the PLC Programming software D00, the new project name is displayed. The LD/ST Programming Editor is also displayed, allowing the user to edit the program.
     In the dialog, click on the <Cancel> button to cancel the operation to create a new project.



- When the B00 and C00 project function is enabled
  This occurs when the checkbox is ticked for the <Enable function> - <B00 and C00 project> (under the <Auxiliary> menu, go to <Option settings> and then the <Customize settings> tab to access).
  1. Select <Create new project> from the project menu in the PLC Programming software D00.
  2. In the <Select program> dialog, select <D00 project> for the project type and then click on the <OK> button.

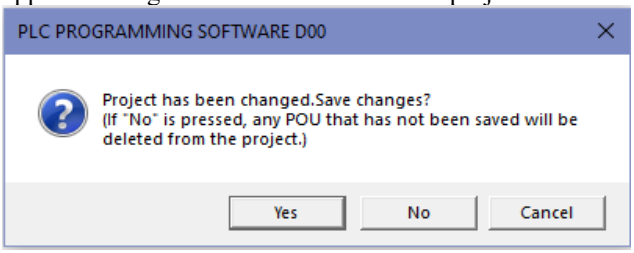


  3. Execute steps 2 and 3 noted above when the aforementioned is disabled.

In step 2, after selecting a <B00 project> or <C00 project> and clicking on the <OK> button, the PLC Programming software D00 EDITOR automatically closes and the B00/C00 EDITOR launches.

5

(NOTE 1) If a new project is created while editing an unsaved project, when the user clicks on the <OK> button in the <Create new project> dialog, a confirmation message appears asking whether or not to save the project.



When <Yes> is selected, the project being edited is saved and closed, and then a new project is created.
When <No> is selected, the project being edited is closed without it being saved, and then a new project is created.
When <Cancel> is selected, the new project operation is cancelled. The project being edited is also not saved.

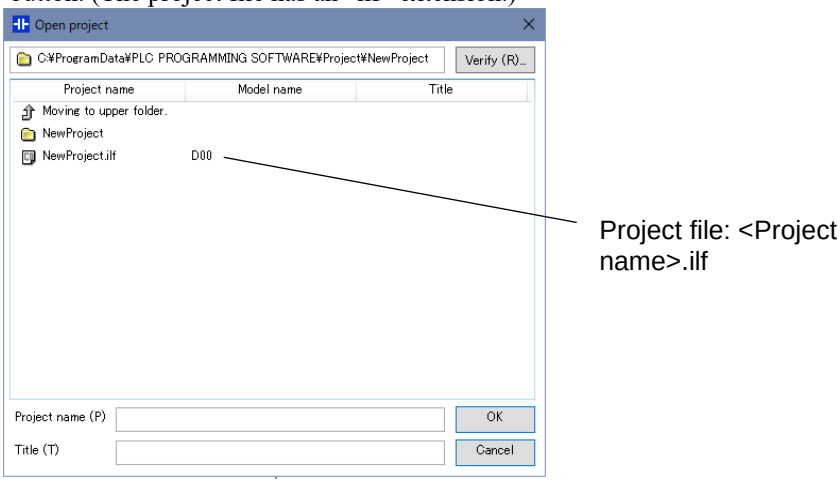
(NOTE 2) In the <Create new project> dialog, if a folder is specified that does not exist at the save location, then that folder is automatically created.

5.1.2 Open Projects

This section describes the operations for reading a POU that is saved to a recording media device.

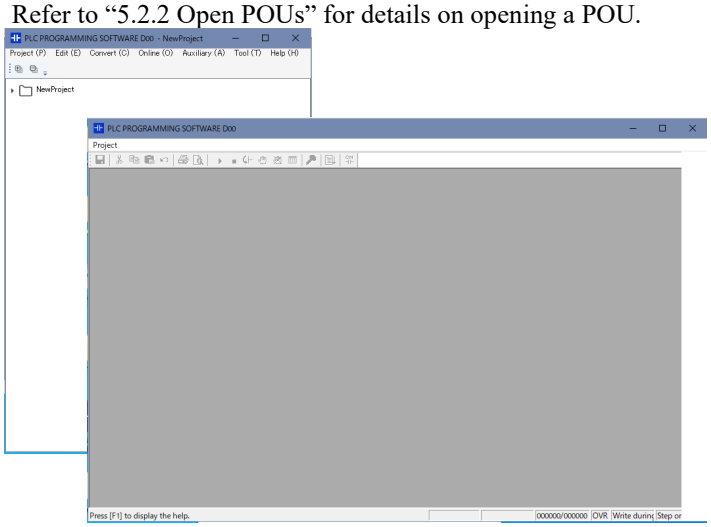
Operation procedure

1. Select <Open project> from the <Project> menu in the PLC Programming software D00.
2. In the dialog for opening files, double-click on the folder of the project, or select the folder and click on the <Open> button.
   B00, C00 and D00 projects are displayed in the project list.
3. Double-click on the D00 project file, or select the D00 project file and click on the <Open> button. (The project file has an “ilf” extension.)

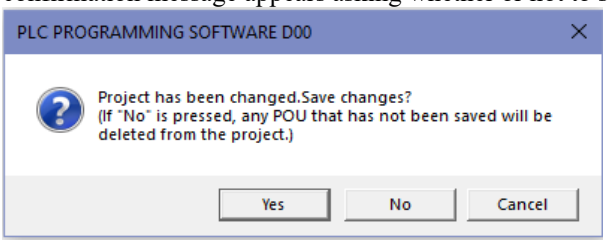


When a B00 or C00 project is selected, the PLC Programming software D00 EDITOR automatically closes, the B00/C00 EDITOR launches and the project opens up.

4. The specified project opens up.
   Refer to “5.2.2 Open POUs” for details on opening a POU.

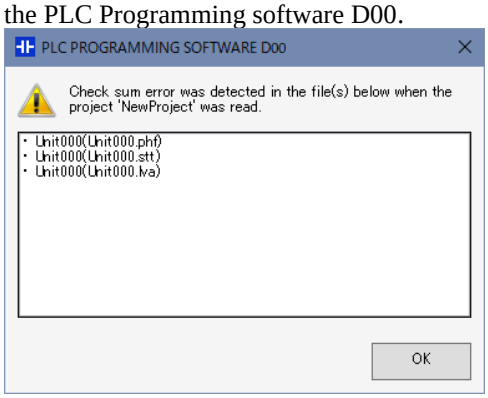


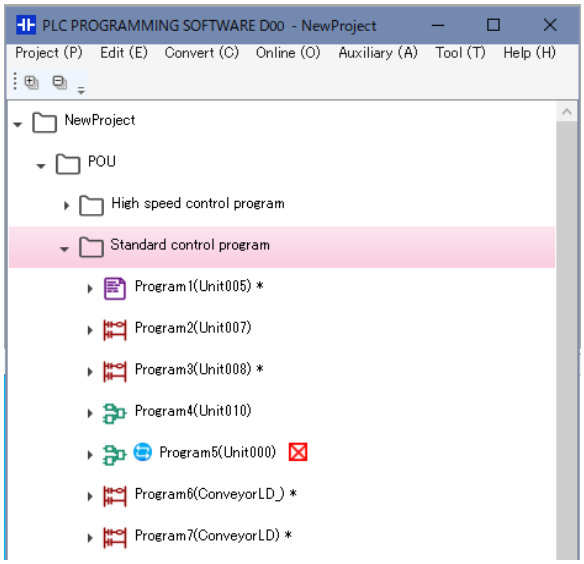
(NOTE) When an attempt is made to open another project while editing an unsaved project, a confirmation message appears asking whether or not to save the project.



When <Yes> is selected, the project being edited is saved and closed, and then the other project is opened.
When <No> is selected, the project being edited is closed without it being saved, and then the other project is opened.
When <Cancel> is selected, the project being edited is not saved, and the other project is not opened.

In order to confirm whether or not project data has been corrupted or altered while using the PLC Programming software D00 EDITOR, a checksum is used when the project is opened to verify the integrity of the files that contain project data.
When a checksum error has been detected, a data list is displayed and an "×" symbol appears in the PLC Programming software D00.





When there is project data that contains a checksum error, the program cannot be written to the NC(BUILT IN PLC)
Check and correct the program and settings where the checksum error was detected, and then save the project and POU.

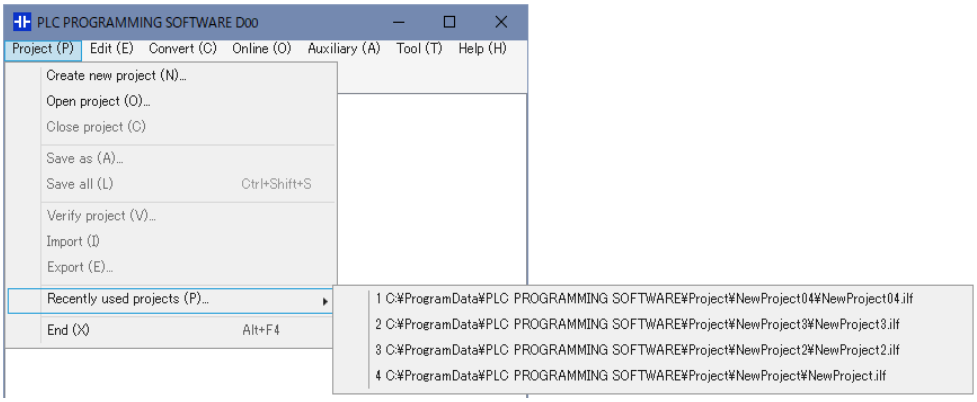
(NOTE 1) Even when there are differences in the program content and/or settings, the checksum values may still match. When writing the program, be sure to check the data before executing the writing operation.

(NOTE 2) Only use the PLC Programming software D00 EDITOR (and not any other tool) to open and edit the files with the PLC Programming software D00 project data. If a project is edited using another tool besides the PLC Programming software D00 EDITOR, then the edited project operation cannot be guaranteed.

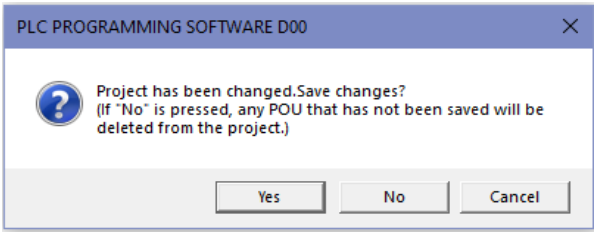
(NOTE 3) A project where a checksum error has been detected may not open properly in the PLC Programming software D00 EDITOR. We recommend backing up project data frequently.

5.1.3 Open Recently Used Projects

The user can open a project via <Recently used projects> under the <Project> menu in the PLC Programming software D00. Specify a particular project file to open a project.
The <Recently used projects> menu lists the 4 most recently opened projects.



(NOTE) When an attempt is made to open another project while editing an unsaved project, a confirmation message appears asking whether or not to save the project.



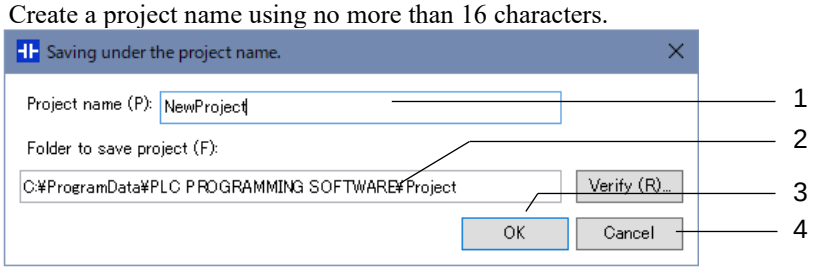
When <Yes> is selected, the project being edited is saved and closed, and then the other project is opened.
When <No> is selected, the project being edited is closed without it being saved, and then the other project is opened.
When <Cancel> is selected, the project being edited is not saved, and the other project is not opened.

5.1.4 Save As

This section describes the operation procedure to save an open project under a different name.

Operation procedure

1. Select <Save as> from the <Project> menu in the PLC Programming software D00.
2. Set a new project name.
   Create a project name using no more than 16 characters.



No.	Item	Description
1	Project name	Set a project name using no more than 16 characters. The following symbols cannot be used: ¥, /, :, *, ?, ", <, > and |
2	Folder to save project	Specifies the folder for saving the project file. After pressing the <Browse> button to display the <Folder selection> dialog, the user can select a folder.
3	<OK> button	Saves the project and closes the dialog.
4	<Cancel> button	Closes the dialog without saving the project.

3. When the <Save under the project name> dialog appears, click on the <OK> button.
   After the project is saved, the project name on the title bar of the PLC Programming software D00 changes to the new name. The editing status of the POU continues as is.
   Even if there is an unconverted POU, the save operation is executed.

5.1.5 Save All

All data for the open project is overwritten and saved in its current state.

Operation procedure

1. Select <Save all> from the <Project> menu in the PLC Programming software D00.
2. The project is saved.
   The editing status of the POU continues as is.
   Even if there is an unconverted POU, the save operation is executed.

(NOTE) Refer to “5.2.3 Save POUs” for details on how to individually save POUs being edited.

5.2 Create POUs (LD/ST/FBD Programs)

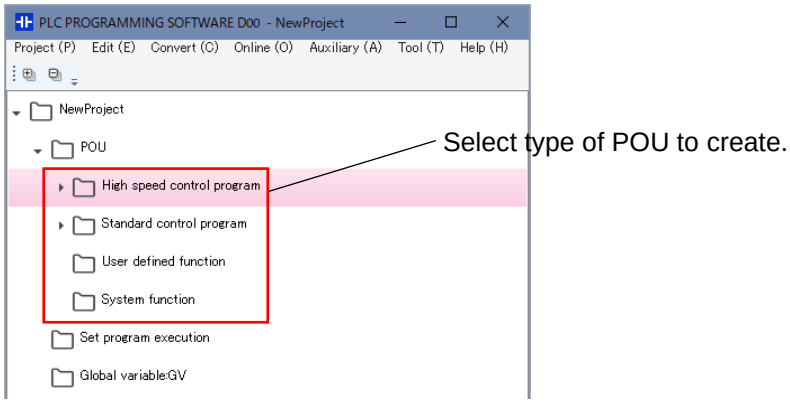
This section describes the operations for creating and saving POUs.
Refer to the following chapters for details on how to edit LD/ST/FBD programs.
LD program edit procedure :"Chapter 7 LD Edit operation procedure"
ST program edit procedure :"Chapter 8 ST Edit operation procedure"
FBD program edit procedure :"Chapter 9 FBD Edit operation procedure"

5.2.1 Procedure to Create New POU

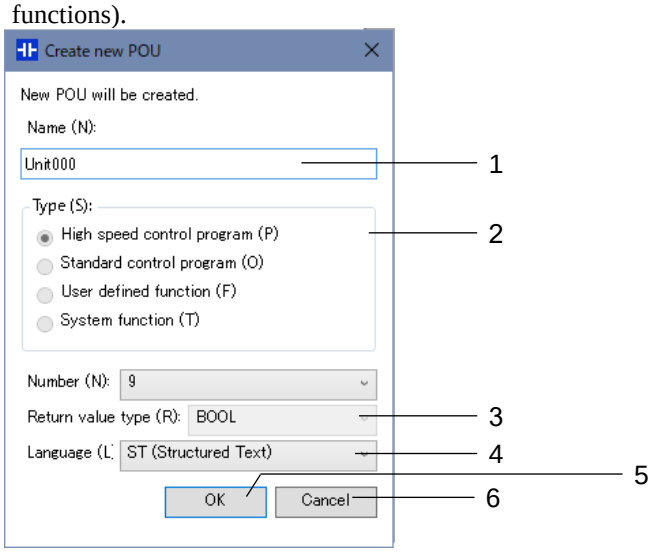
This section describes the procedure to display the LD/ST/FBD program edit screen.

Operation procedure

1. Select the type of the POU to create from: <High speed control program>, <Standard control program>, <User defined function> or <System defined function>.



2. Perform one of the following operations.
   - Select <Create POU> from the <Edit> menu
   - Click on the <High speed control program>, <Standard control program>, <User defined function> or <System defined function>, or right-click on the folder created by the user and then select <New> and <POU>
3. In the <Create new POU> dialog, set the POU name, language and return value type (only for functions).



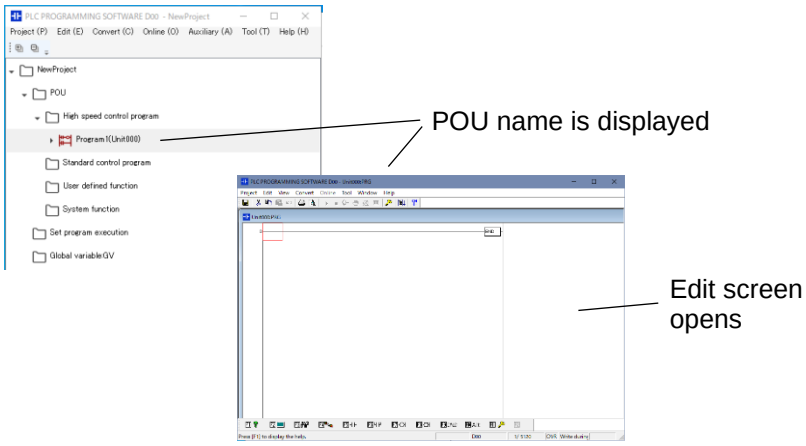
No.	Item	Description
1	Name	Sets the POU name. Set the name using no more than 16 characters. Single-byte alphanumeric characters and a half-width underscore (_) can be used for the name. However, numbers cannot be used as the first character.
2	Type	Displays the type of POU to create.
3	Return value type	Sets the return value data type when creating a user defined or system defined function. The following types can be set. BOOL/BYTE/WORD/DWORD/INT/DINT/LINT/UINT/UDINT/ULINT/REAL/LREAL/DATE/TIME/TOD/DT/STRING
4	Language	Selects one of the following languages to create the POU. ・ST ・LD ・FBD When creating a user defined or system defined function, the "ST" and "FBD" languages can be selected.
5	<OK> button	Creates a POU and closes the dialog.
6	<Cancel> button	Cancels the POU operation and closes the dialog.

4. When the <Create new POU> dialog appears, click on the <OK> button.
   The POU specified in the <Create new POU> dialog is added in the PLC Programming software D00, and the edit screen is displayed in the specified language.

When creating a POU, the LD program and ST program display the edit screen in the LD/ST Programming Editor.
The FBD program is launched in the FBD Programming Editor.

In the dialog to create a new POU, click on the <Cancel> button to cancel creating a POU.

Ex: When an LD program is created



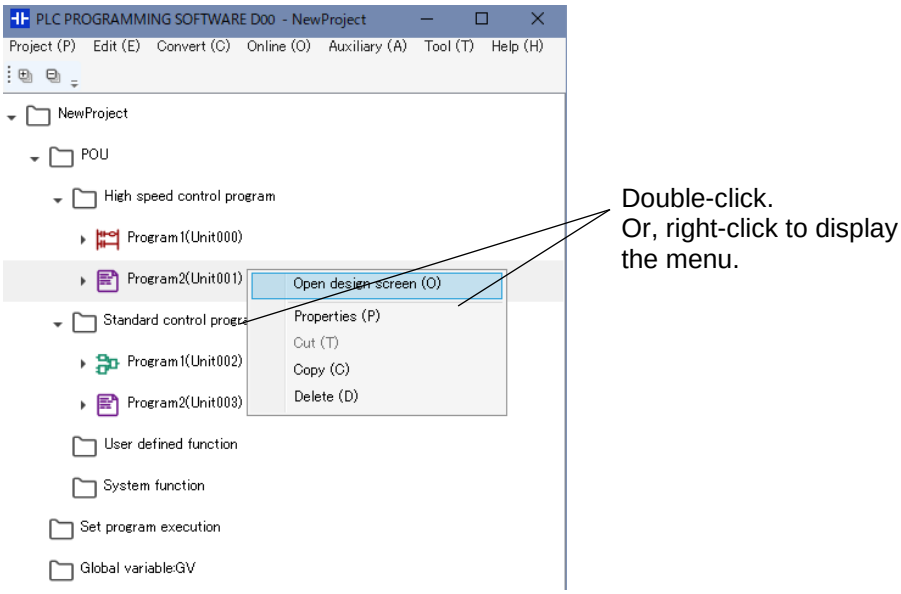
(NOTE 1) Refer to "12.1.6.1 List of data types" in the PLC System Manual or "12.1.2.1 List of data types" for details on the data types.
(NOTE 2) When creating a high speed or standard control program, the "Return value type" is not set.
(NOTE 3) The POU "Name" and "Return value type" can be changed after creating the POU. Refer to "6.2.3 Change POU properties" for details on how to change the properties.
(NOTE 4) The user can use (copy/paste) an existing POU. Refer to "6.2.5 Use POUs (Copy, Cut & Paste)" for details on operations with existing POUs.

5.2.2 Open POUs

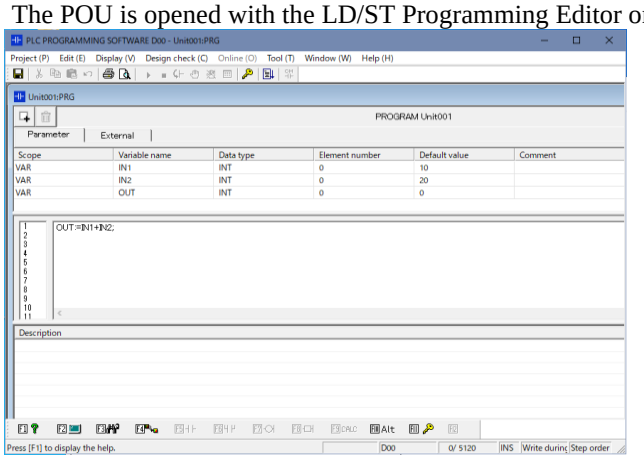
When opening a POU (LD program, ST program or FBD program) that is saved, the user can display the POUs from the tree in the PLC Programming software D00 and specify a POU to open.

Operation procedure

1. Perform one of the following operations in the PLC Programming software D00.
   - Double-click on a POU to open.
   - Right-click on a POU to open, and then select <Open design screen> from the displayed menu.



2. The POU is opened with the LD/ST Programming Editor or the FBD Programming Editor.



(NOTE) When opening a POU, if the LD/ST Programming Editor or the FBD Programming Editor is not launched, then the LD/ST Programming Editor/FBD Programming Editor will automatically launch in the language of the POU.

5.2.3 Save POUs

The user can save each POU (LD program, ST program or FBD program) individually in the LD/ST Programming Editor or FBD Programming Editor.
In the LD/ST Programming Editor, it saves the POU that is in the active window.

Operation procedure

1. Select <Save POU> from the <Project> menu in the LD/ST Programming Editor or FBD Programming Editor.
2. The POU is saved.

(NOTE 1) To save all the POUs, select <Save all> from the <Project> menu in the PLC Programming software D00. However, the <Save all> operation saves all the project data and not just the POUs.
(NOTE 2) Even if the POU is unconverted, it is still saved.
(NOTE 3) When a POU is converted, that POU is automatically saved.

5.3 Register Global Variables

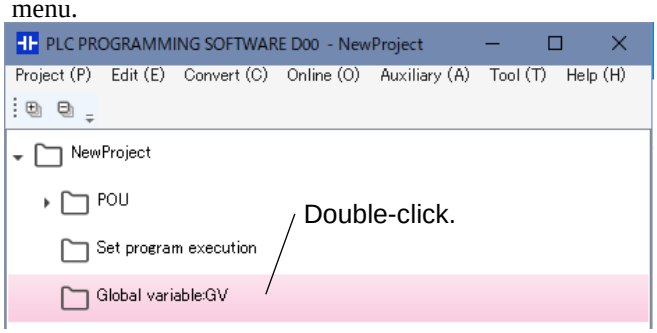
5.3.1 Create Global Variables

The global variables can be used throughout the entire project regardless of the POUs.
This section describes the procedure to register the global variables.

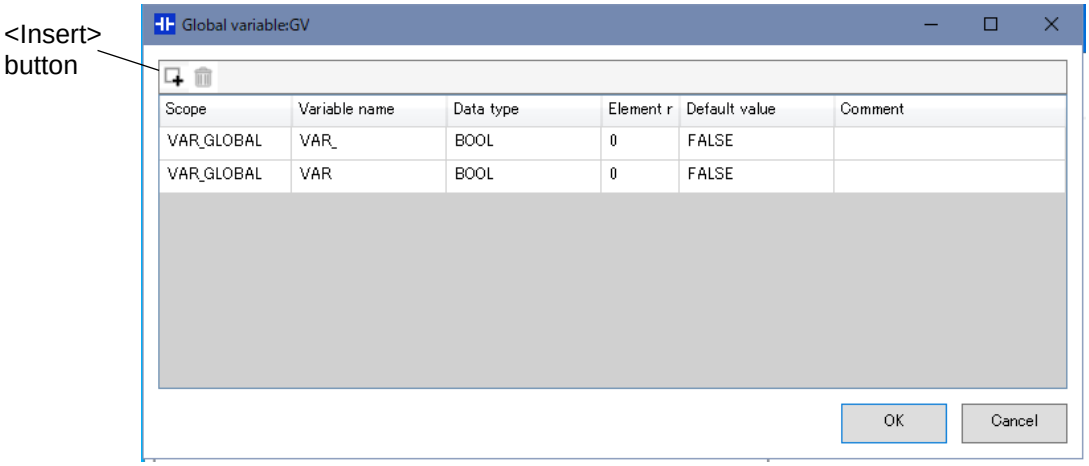
(NOTE) Refer to the "6.3 Global variable settings" in the PLC System Manual for details on the global variables.

Operation procedure

1. Double-click on <Global variable> in the PLC Programming software D00.
   Or, right-click on <Global variable>, and then select <Open design screen> from the displayed menu.

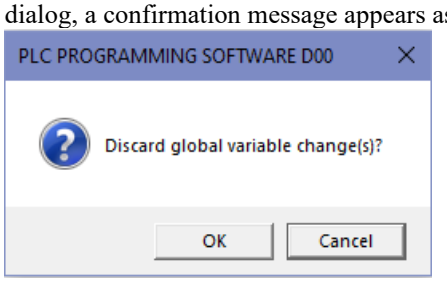


2. In the <Global variable> dialog, perform one of the following operations.
   - Press the <Insert> button.
   - Right-click with the mouse, and select <Insert> from the context menu.
   - Press the **[Ctrl]** + **[I]** keys.



3. Click on the <OK> button.

(NOTE 1) When the user clicks on the <Cancel> button after editing in the <Global variable> dialog, a confirmation message appears asking whether or not to discard the settings.



When <Yes> is selected, the settings are discarded and the <Global variable> dialog closes.
When <No> is selected, it goes back to the <Global variable> dialog.

(NOTE 2) The global variables can be copied and pasted. The user can copy and paste global variables to other projects.
To copy a global variable, select a variable in the <Global variable> dialog, right-click on the mouse and then select <Copy> from the context menu. Or, press the **[Ctrl]** + **[C]** keys to perform the copy operation.
To paste a global variable, select a variable, right-click on the mouse in the <Global variable> dialog and then select <Paste> from the context menu. Or, press the **[Ctrl]** + **[V]** keys to perform the paste operation.

5

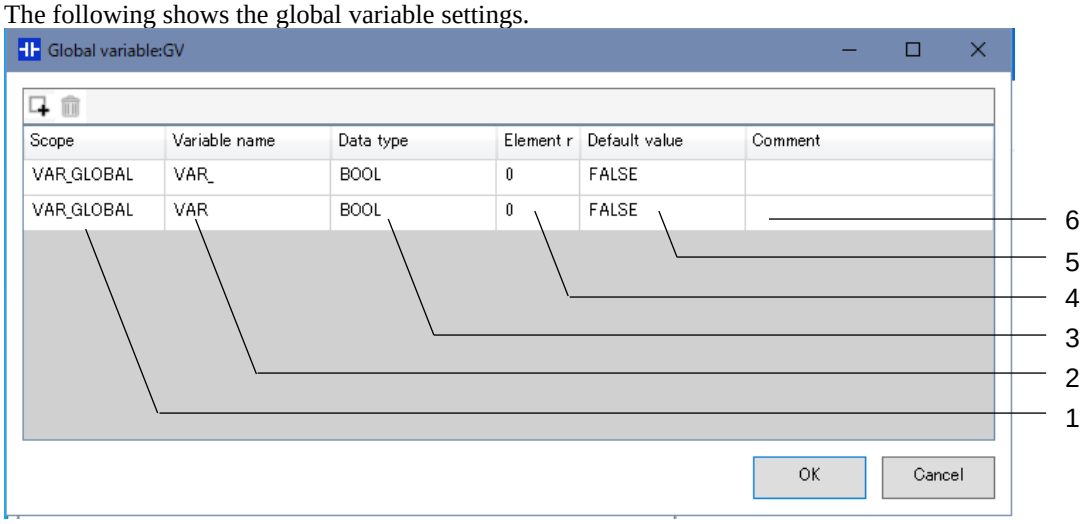
The following shows the maximum number of global variables that can be set.
BOOL maximum: 1024
Non-BOOL maximum: 1024

After registering 1024 BOOL type variables, the user must perform the following procedure to register non-BOOL type variables.

Operation procedure
1. Delete a BOOL type variable. Or, change it to another data type.
   (Write down the variable settings before deleting or changing them.)
2. Register a non-BOOL type variable.
3. Re-register the variable that was deleted in step 1. Or, return the data type (that was changed in step 1) back to the BOOL type.

5.3.2 Global Variable Settings

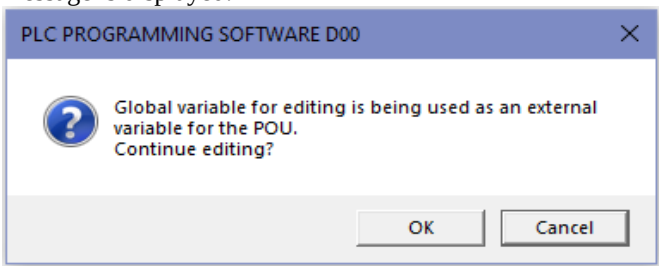
The following shows the global variable settings.



No.	Item	Description
1	Scope	"VAR_GLOBAL" is a fixed setting.
2	Variable name	Sets the variable name. • The user can set a maximum of 32 characters. • Single-byte alphanumeric characters and a half-width underscore (_) can be used. • Numbers cannot be used for the first character. • The name of another parameter variable cannot be duplicated in the same POU. • The variable name cannot be the same as any reserved words.
3	Data type	Selects the variable data type from the following. BOOL/BYTE/WORD/DWORD/INT/DINT/LINT/UINT/UDINT/ULINT/REAL/LREAL/DATE/TIME/TOD/DT/STRING
4	Element number	Sets the element number of the array between 1 and 1024 when being used as an array variable. The number is set to 0 when it is not being used as an array variable.
5	Default value	Sets the default value of the variable when the NC(BUILT IN PLC) is launched. The values that can be set vary depending on the data type. Refer to "12.1.6.1 List of data types" in the PLC System Manual or "12.1.2.1 List of data types" for further details. This is the default value for all elements when using an array.
6	Comment	No more than 32 characters can be used for a comment.

Note that hexadecimal numbers, unsigned decimal numbers, binary numbers and octal numbers can be used to set the default value for BYTE, WORD and DWORD variables.
Signed decimal numbers and hexadecimal numbers can be used to set the default value for INT, DINT and LINT variables.
Unsigned decimal numbers and hexadecimal numbers can be used to set the default value for UINT, UDINT and ULINT variables.

In the <Global variable> dialog, when an attempt is made to change the settings (variable name, data type or element number) of a variable being used in a POU, the following confirmation message is displayed.

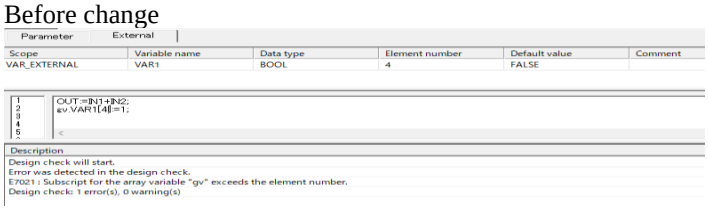


Select <OK> to change the settings. Select <Cancel> to cancel the setting change.

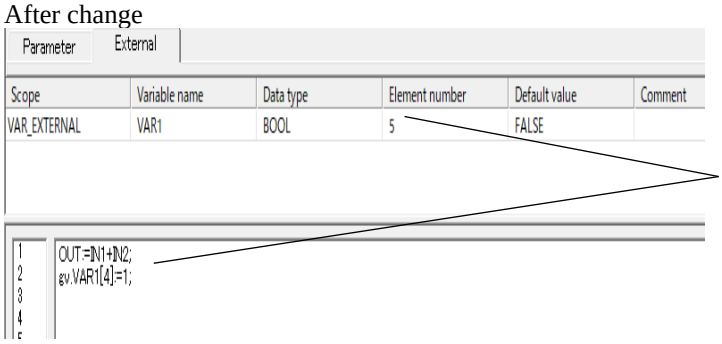
If the “Data type”, “Element number”, “Default value” and/or “Comment” is changed for a global variable, then that change is automatically reflected in the settings of the external variable registered in the variable area of each POU. However, that change is not reflected in the locations where it is used within the program. Therefore, the user must correct the program according to setting change(s) of the global variable.
If the “Variable name” of the global variable is changed, the setting in the variable area of each POU will become blank. In addition, the change is not automatically reflected in the locations where it is used within the program. Therefore, the user must re-register the external variable and also correct the program.

5

Example 1: When the element number of the global variable VAR1 is changed from 5 to 4.
Before change

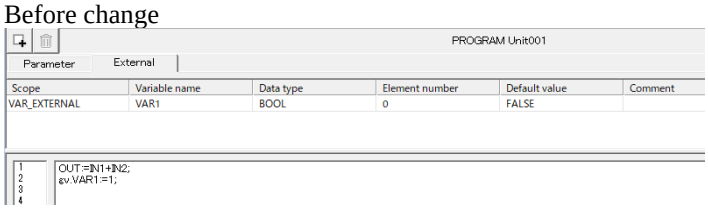


After change

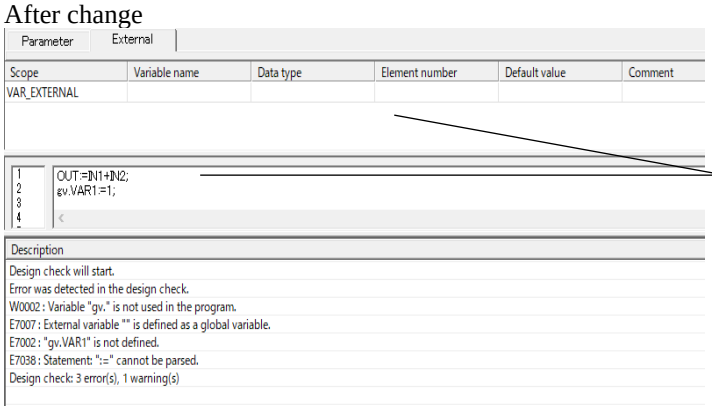


The global variable setting changes are reflected automatically in the external variable settings. However, the program itself does not change. Therefore, a conversion error (element number exceeded) will trigger if the program is not corrected.

Example 2: When the name of the global variable VAR1 is changed to VAR2.
Before change



After change



The external variable setting becomes blank, but the program does not automatically change. Therefore, the user must change the program.
If the program is not corrected, a conversion error (such as an undefined variable) will trigger.

(NOTE 1) The setting in the <Global variable> dialog can be changed for a variable being used in a POU, but the changes noted below will not be automatically reflected in each POU.

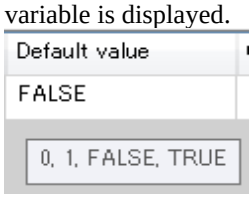
- When the variable name is changed
- Locations where the variable is used in the program
  For the above changes, correct the external variables of each POU and the program manually.

(NOTE 2) If a global variable is added or changed in the <Global variable> dialog, then all the POUs in the project become unconverted.

(NOTE 3) When a default value is set for a REAL or LREAL type variable, then the set value is rounded off to the valid digits of that type (7 digits and 15 digits, respectively). Examples are shown below.

Data type	Input value	Set value
REAL type	1234567	1234567
	12345678	1.234568E+07
	0.0001234567	0.0001234567
	0.00012345678	0.0001234568
LREAL type	123456789012345	123456789012345
	1234567890123456	1.23456789012346E+15
	0.000123456789012345	0.000123456789012345
	0.0001234567890123456	0.000123456789012346

(NOTE 4) When the mouse cursor is moved to the <Default value> field in the <Global variable> dialog, then the valid input range according to the data type of that variable is displayed.



(NOTE 5) In the <Global variable> dialog, when an attempt is made to change the settings (variable name, data type or element number) of a variable being used in a POU, the following confirmation message is displayed.

(NOTE 6) The default value of the array variable is the default value for all the elements of that array variable. To set individual values for each element, the user must set them in the program.

(NOTE 7) Refer to “12.1.6.1 List of data types ” in the PLC System Manual or “12.1.2.1 List of data types” for details on the data types.
Refer to “Chapter 12 Commands (ST)” in the PLC System Manual or “Chapter 12 Commands (FBD)” for details on reserved words. To set individual values for each element, the user must set them in the program.

(NOTE 8) The user can enter a user defined or system defined function name for the variable name, but an error will trigger at conversion.

5.3.3 Delete Global Variables

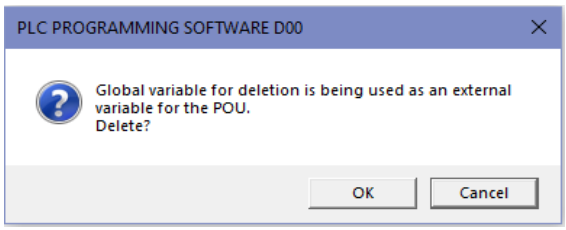
The user can delete registered global variables in the <Global variable> dialog.
After selecting the variable, the user can perform one of the following operations to delete it.

Operation procedure

Select the variable in the <Global variable> dialog, and perform one of the following operations.

- Click on the button.
- Press the **[Delete]** key.
- Right-click the mouse on the line where the selection was made, and then select <Delete> from the context menu.

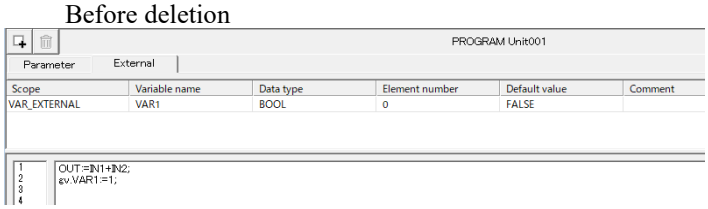
In the <Global variable> dialog, when an attempt is made to delete a variable being used in a POU, the following confirmation message is displayed.



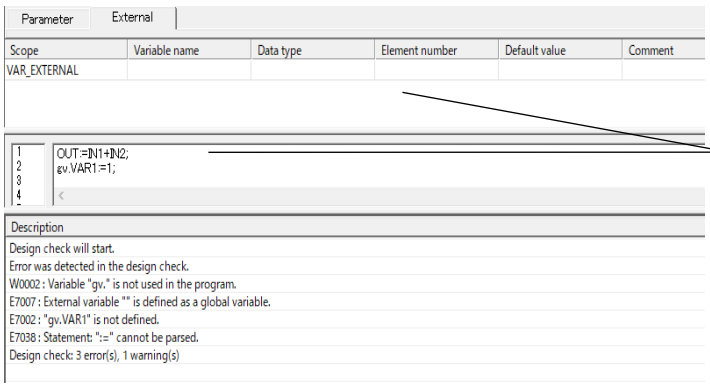
Select <OK> to delete the variable. Select <Cancel> to cancel the deletion.
If a global variable is deleted, the setting in the variable area of each POU will become blank. In addition, the change is not automatically reflected in the locations where it is used within the program. Therefore, the user must re-register the external variable and also correct the program.

Ex: When the global variable VAR1 is deleted.

Before deletion



After deletion



The external variable setting becomes blank, but the program does not automatically change. Therefore, the user must change the program.
If the program is not corrected, a conversion error (such as an undefined variable) will trigger.

(NOTE 1) The setting in the <Global variable> dialog can be deleted for a variable being used in a POU, but the deletion will not be automatically reflected in each POU. Correct the external variables of each POU and the program manually.

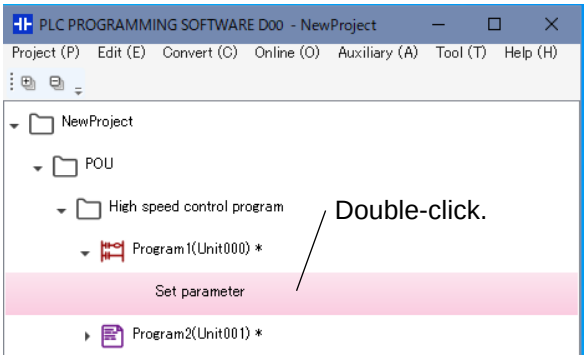
(NOTE 2) If a global variable is deleted in the <Global variable> dialog, then all the POUs in the project become unconverted.

5.4 Parameter Settings

Every time a program is executed in the NC(BUILT IN PLC), the user can set the <Scan time>, <Monitor WDT>, <Operation method when an error occurs>, <Output method when mode is changed from STOP to RUN> and <Power failure hold area setting>.
The parameters can be set in the PLC Programming software D00.

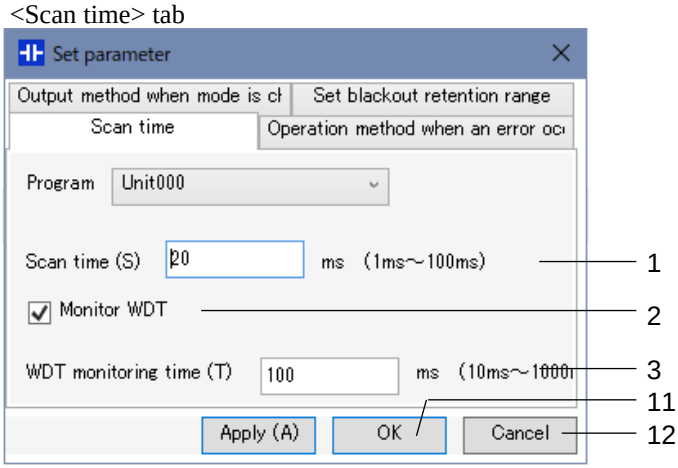
Operation procedure

1. Perform one of the following operations in the PLC Programming software D00.
   - Double-click on <Parameter setting> in the tree.

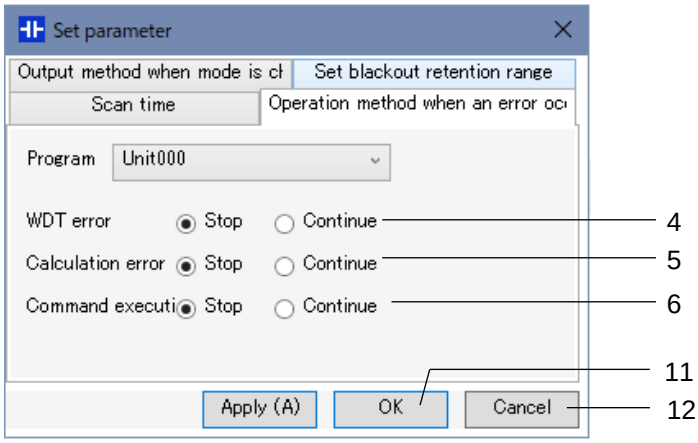


   - Right-click on <Parameter setting> in the tree, and then select <Open design screen> from the displayed menu.

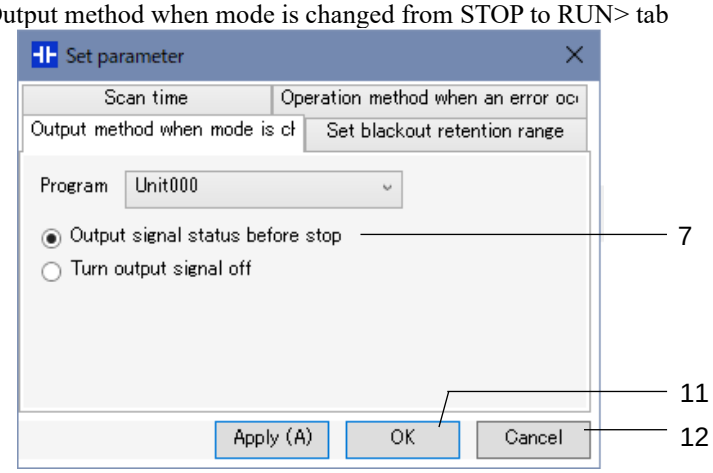
2. The <Parameter setting> dialog is displayed.
   <Scan time> tab



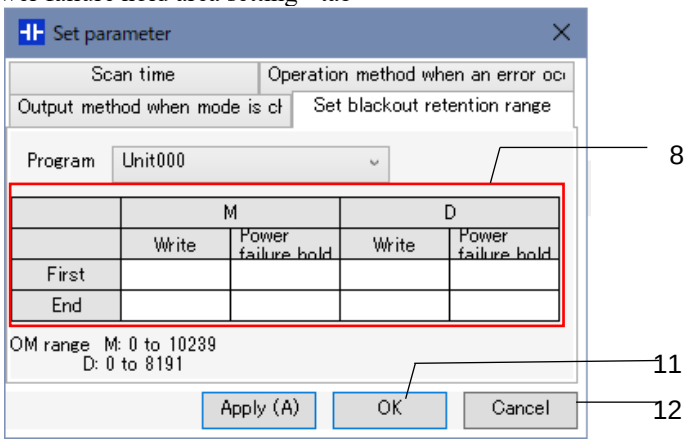
<Operation method when an error occurs> tab



<Output method when mode is changed from STOP to RUN> tab



<Power failure hold area setting> tab



No.	Tab	Item	Description
1	Scan time	Scan time	Sets the scan time between 1 and 100 ms when using a high speed control program. (Unit: 1 ms) Sets the scan time between 10 and 1000 ms when using a standard control program. (Unit: 10 ms)
2		Monitor WDT	Selects whether or not to monitor the WDT. • Checkbox ticked: Monitors. • Checkbox not ticked: No monitoring.
3		WDT monitor time	Sets the WDT monitor time between 10 and 1000 ms when using a high speed control program. (Unit: 1 ms) Sets the WDT monitor time between 100 and 10000 ms when using a standard control program. (Unit: 10 ms)
4	Operation method when an error occurs	WDT error	Selects <Stop> or <Operate> when there is a WDT error.
5		Calculation error	Selects <Stop> or <Operate> when there is a calculation error.
6		Command execution error	Selects <Stop> or <Operate> when there is a command execution error.
7	Output method when mode is changed from STOP to RUN	Output same status before STOP / Turn OFF output	Selects <Output signal status before stop> or <Turn output signal off>.
8	Power failure hold area setting	Power failure hold/write range setting field	Sets the first and last number of the power failure hold area and the writing range for the selected program.

No.	Tab	Item	Description
9	Common tab	Program	The user can select a program that is set in the program execution settings.
10		<Apply> button	Applies the setting changes, but the dialog does not close.
11		<OK> button	Applies the setting changes, and the dialog closes.
12		<Cancel> button	Discards the setting changes, and the dialog closes.

3. Click on the <OK> button.

(NOTE 1) If the settings are configured so that the operation continues in the NC(BUILT IN PLC) when a WDT error is triggered, be careful not to write a program where an infinite loop occurs into the NC(BUILT IN PLC).
If an infinite loop occurs, the END processing is not carried out in the NC(BUILT IN PLC), and communication is no longer possible between the PLC Programming software D00 EDITOR and NC(BUILT IN PLC).

(NOTE 2) In the simulation function, operation runs with fixed values regardless of the parameter settings.
The scan time is 100 ms.
WDT monitoring is also carried out, and it uses a 5000 ms monitor time.
When an error is triggered, the simulation stops regardless of the error type.

(NOTE 3) The user can change the parameter default values.
The default values can be set with “custom.ini” in the “Setting” folder under the install folder for the PLC Programming software D00.
However, when changing the default values using “custom.ini”, always set values that fall inside the valid setting range.
The following shows the “custom.ini” settings.

———————————————————————————————————————

[ParameterSettings]
;ScanTime default:20(ms) range:1-2000ms
ScanTime = 20← Set default value for scan time

;WdtTime default:100(ms) range:100-100,000ms(The last digit must be 0)
WdtTime = 100 ← Set default value for WDT monitor time to multiples of 10

;WdtSurveillance default:true range:true or false
WdtSurveillance = true ←Set default value for “Monitor WDT” (True: Yes, False: No)

;WdtError default:false range:true or false
WdtError = false ←Set default value for “WDT error” (True: Operate, False: Stop)

;CalculationError default:false range:true or false
CalculationError = false ←Set default value for “Calculation error” (True: Operate, False: Stop)

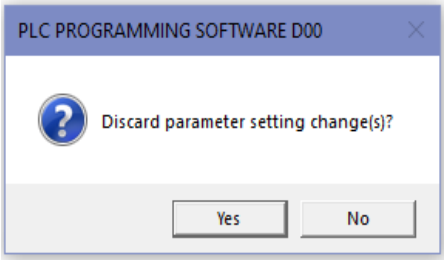
;OrderExeError default:false range:true or false
OrderExeError = false ←Set default value for “Command execution error” (True: Operate, False: Stop)

———————————————————————————————————————

The PLC Programming software D00 EDITOR must be re-launched to apply the setting changes in “custom.ini”.

The following is the default install folder for the PLC Programming software D00.
C:¥Program Files¥PLC PROGRAMMING SOFTWARE¥D00¥PLC (When using a 32-bit OS)
C:¥Program Files (x86)¥PLC PROGRAMMING SOFTWARE¥D00¥PLC (When using a 64-bit OS)

(NOTE 4) When the user clicks on the <Cancel> button after editing in the <Parameter setting> dialog, a confirmation message appears asking whether or not to discard the settings.



When <Yes> is selected, the settings are discarded and the <Parameter setting> dialog closes.
When <No> is selected, it goes back to the <Parameter setting> dialog.

5.5 Register DFC (Dynamic Function Call)

The description in this chapter is omitted because this product is not equipped with this function.

5.6 Program Edit

This section describes the procedure to create LD programs, ST programs and FBD programs.

5.6.1 Create LD Programs

The following operations describe the settings related to creating LD programs and the editing of LD programs.

- Edit LD programs (Refer to “5.6.1.2 Edit LD programs”)
- Convert (Refer to “5.9 Conversion”)
- Create comments (Refer to “5.6.1.3 Create OM comments, inline comments and coil remarks (overview)”)

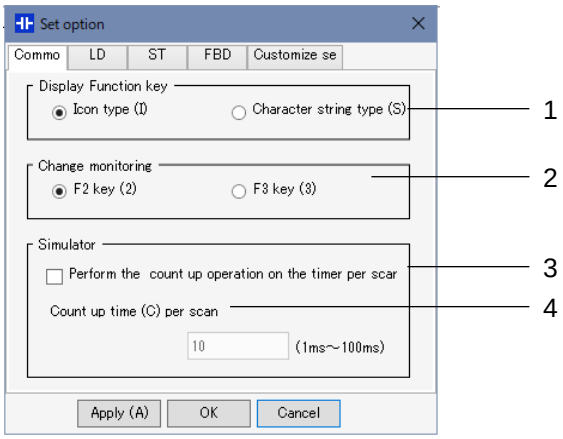
5.6.1.1 Edit screen adjustment (option settings)

The user can adjust the display and some operations on the edit screen in order to make the LD program editing more user friendly.
These settings are applied to all the LD program edit windows.

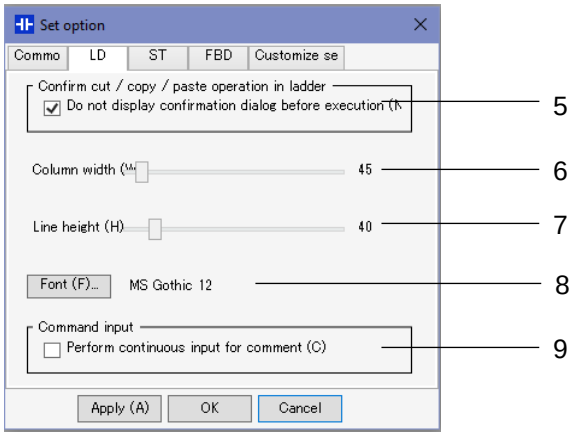
Operation procedure

Select <Option settings> from the <Auxiliary> menu in the PLC Programming software D00.

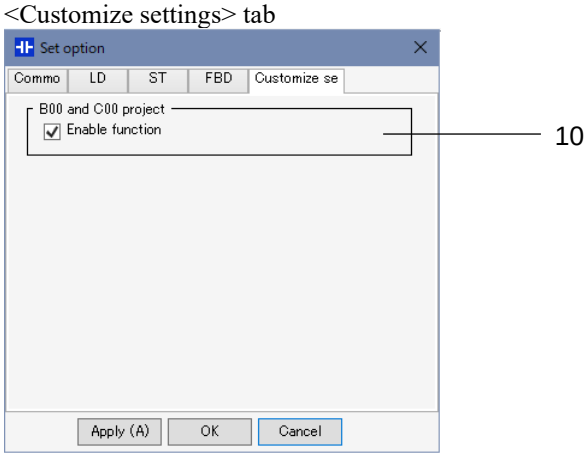
<Common> tab



<LD> tab



<Customize settings> tab



No.	Tab	Item	Description
1	Common	Function key display	Selects whether to use icons or text on the tool bar for the function keys in the LD/ST Programming Editor.
2		Change monitoring	Selects whether to use **[F2]** or **[F3]** as the function key to monitor the program.
3		Perform count up operation on timer per scan	Selects one of the following count up methods on the timer when the simulation is executed. Checkbox not ticked: Counts up based on the actual measured scan time after the input time is carried out. Checkbox ticked: User sets the time count per scan.
4		Count up time per scan	Sets the time count per scan between 1 and 100 ms. The unit is in ms.
5	LD	Confirm cut / copy / paste operation in ladder	Selects whether or not to display a confirmation message when cutting, copying or pasting a circuit while editing an LD program. Checkbox not ticked: Show Checkbox ticked: Hide
6		Column width	Sets the width of a column (40 to 100 pixels) on a ladder circuit.
7		Line height	Sets the line height (30 to 100 pixels) on a ladder circuit.
8		Font	Sets the font and font size used in the ladder circuit display.
9		Perform continuous input for comment	Selects whether or not to input a comment after a command input in the LD program. Checkbox not ticked: No comment input Checkbox ticked: Comment input Refer to “7.4.1 Create OM comment” for details on comment input.
10	Customize settings	B00 and C00 project	Selects whether or not to enable the <B00/C00 project conversion> and B00/C00 project editing. Checkbox not ticked: Disable Checkbox ticked: Enable

2. Click on the <OK> button or <Apply> button.
   After clicking on the <Apply> button, the <Option settings> dialog does not close and the LD/ST Programming Editor display is changed.
   After clicking on the <Cancel> button, the setting changes in the <Option settings> dialog are discarded.

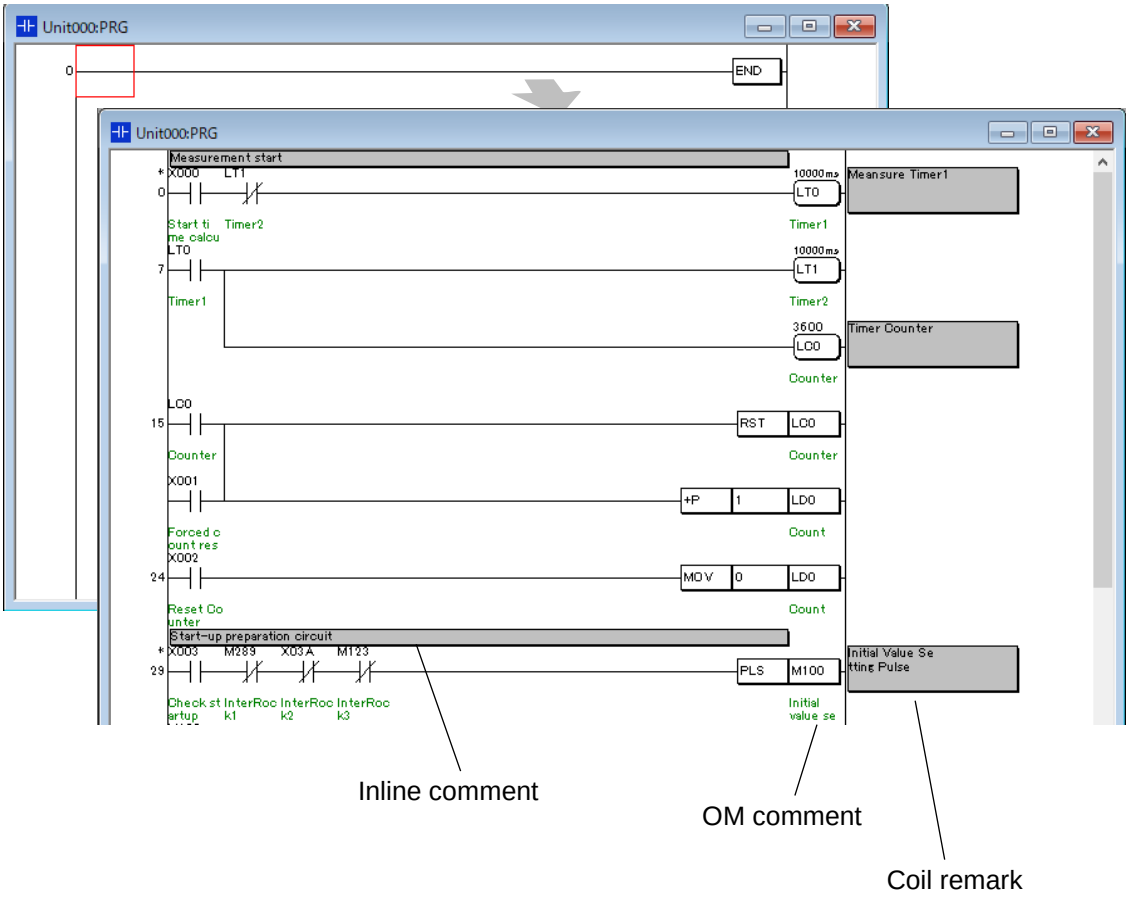
(NOTE 1) When changing the display font in the LD program, the characters may not fit inside the frame depending on the font and size that are set.

(NOTE 2) The <Option settings> is a common setting in the PLC Programming software D00 regardless of the project.

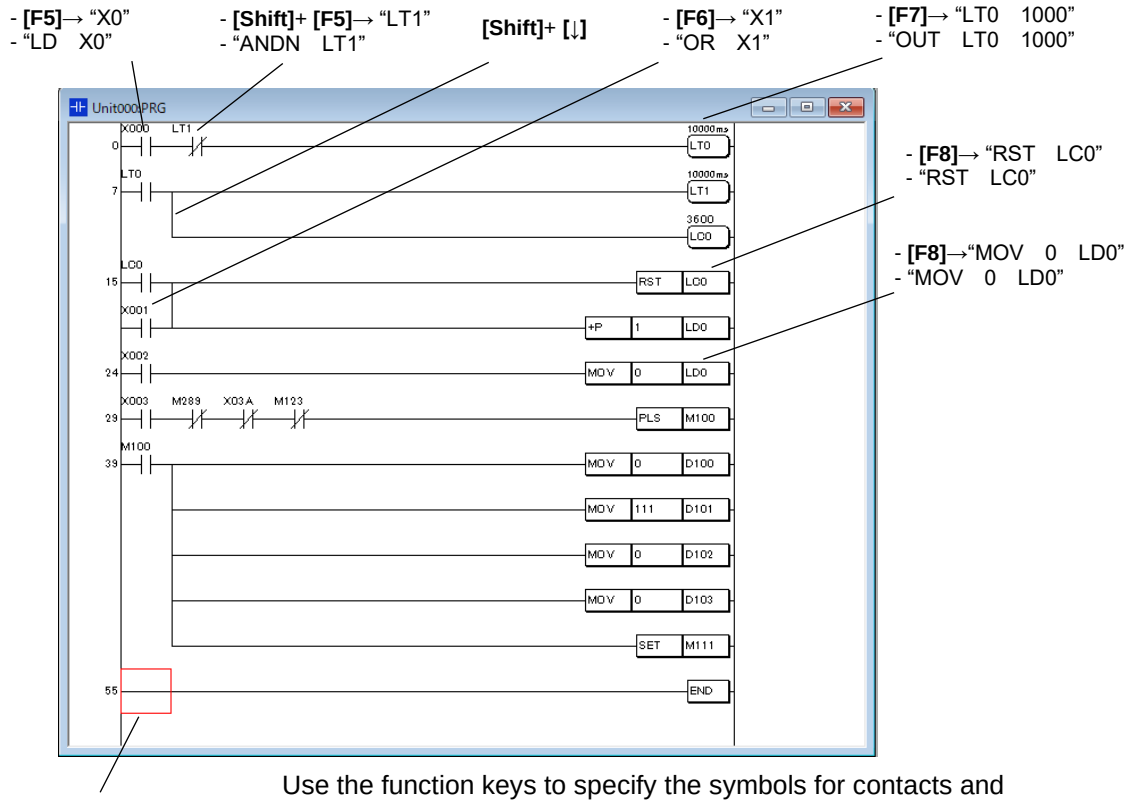
5.6.1.2 Edit LD programs

The LD program is edited using the LD/ST Programming Editor.
This section provides an overview on how to edit an LD program.
Refer to the “Chapter 7 LD edit operation procedure” for details on editing an LD program.

Initial screen



- Command input method (overview)



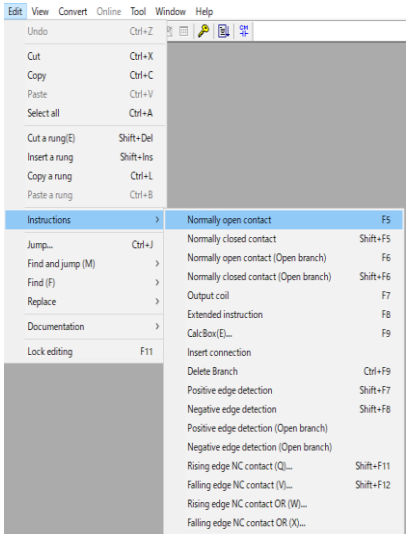
Cursor
Commands are inserted at the cursor position

Use the function keys to specify the symbols for contacts and coils to input on the upper part

Specify commands to directly input on the lower part

Instead of pressing the function keys, command symbols can also be specified using the following method.

Select each command from the <Instructions> under the <Edit> menu.



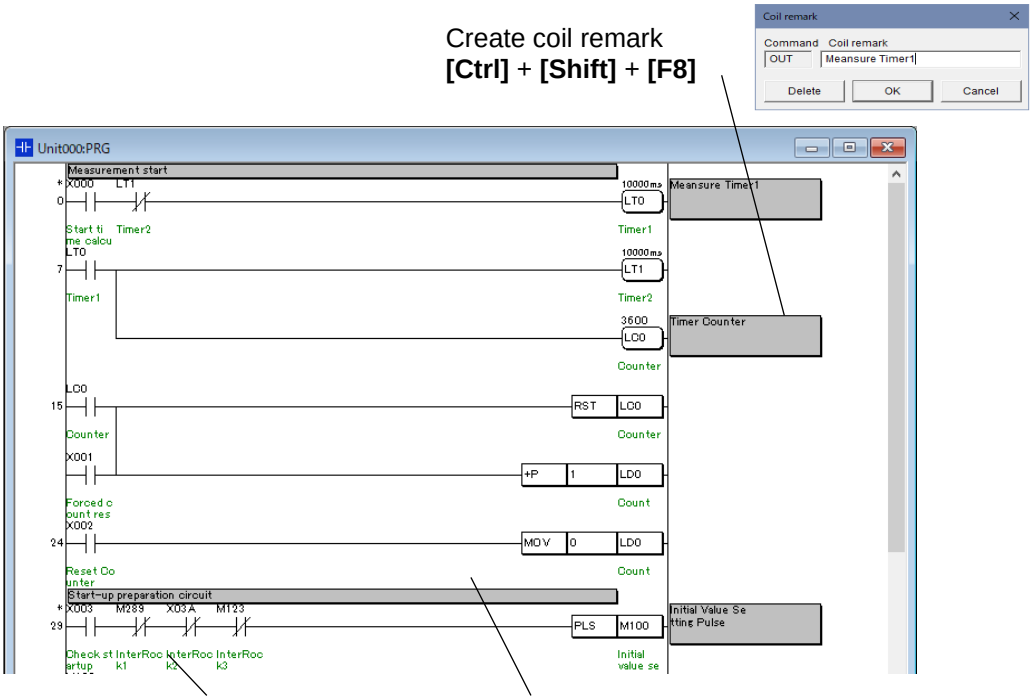
Click directly on the function key tool bar.

Refer to "Chapter 11 Commands (LD)" in the PLC System Manual for details about each command in the LD program.

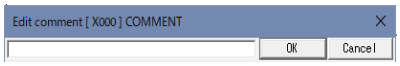
5.6.1.3 Create OM comments, inline comments and coil remarks (overview)

This section describes how to input OM comments, inline comments and coil remarks.

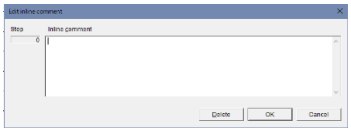
Create coil remark
[Ctrl] + **[Shift]** + **[F8]**



Create OM comment
[Shift] + **[F9]** or **[Ctrl]** + **[Shift]** + **[F5]**



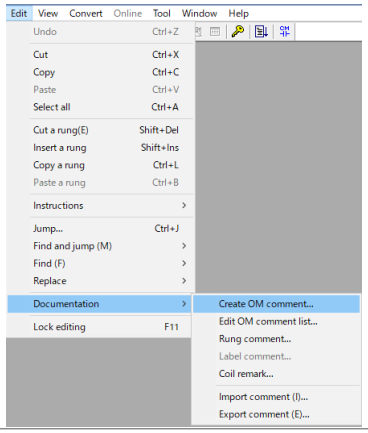
Create inline comment
[Ctrl] + **[Shift]** + **[F7]**



The display can be changed for the OM comment, inline comment or coil remark that is set.

Show/Hide OM comment:	**[Ctrl]** + **[F5]**
Show/Hide inline comment:	**[Ctrl]** + **[F7]**
Show/Hide coil remark:	**[Ctrl]** + **[F8]**

OM comments, inline comments and coil remarks can also be created with the <Create drawing> under the <Edit> menu.



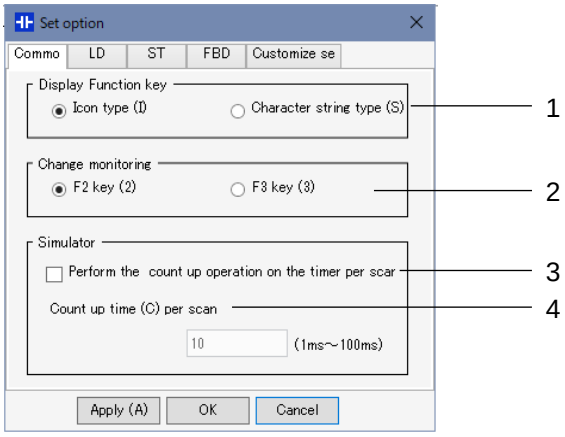
5.6.2 Create ST Programs

This section describes the settings related to creating ST programs and the editing of ST programs.

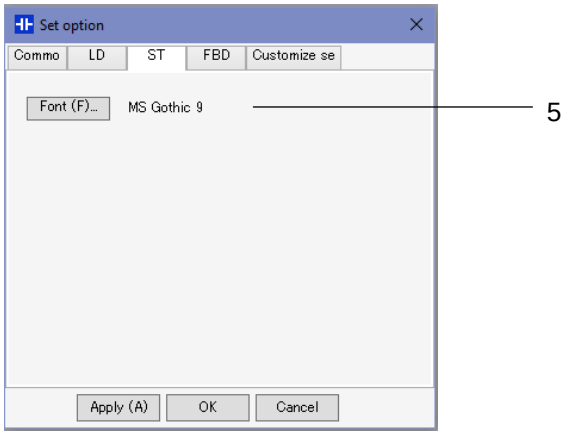
5.6.2.1 Edit screen adjustment (option settings)

The user can adjust the display and some operations on the edit screen in order to make the ST program editing more user friendly.
These settings are applied to all the ST program edit windows.

Operation procedure
Select <Option settings> from the <Auxiliary> menu in the PLC Programming software D00.
<Common> tab



<ST> tab



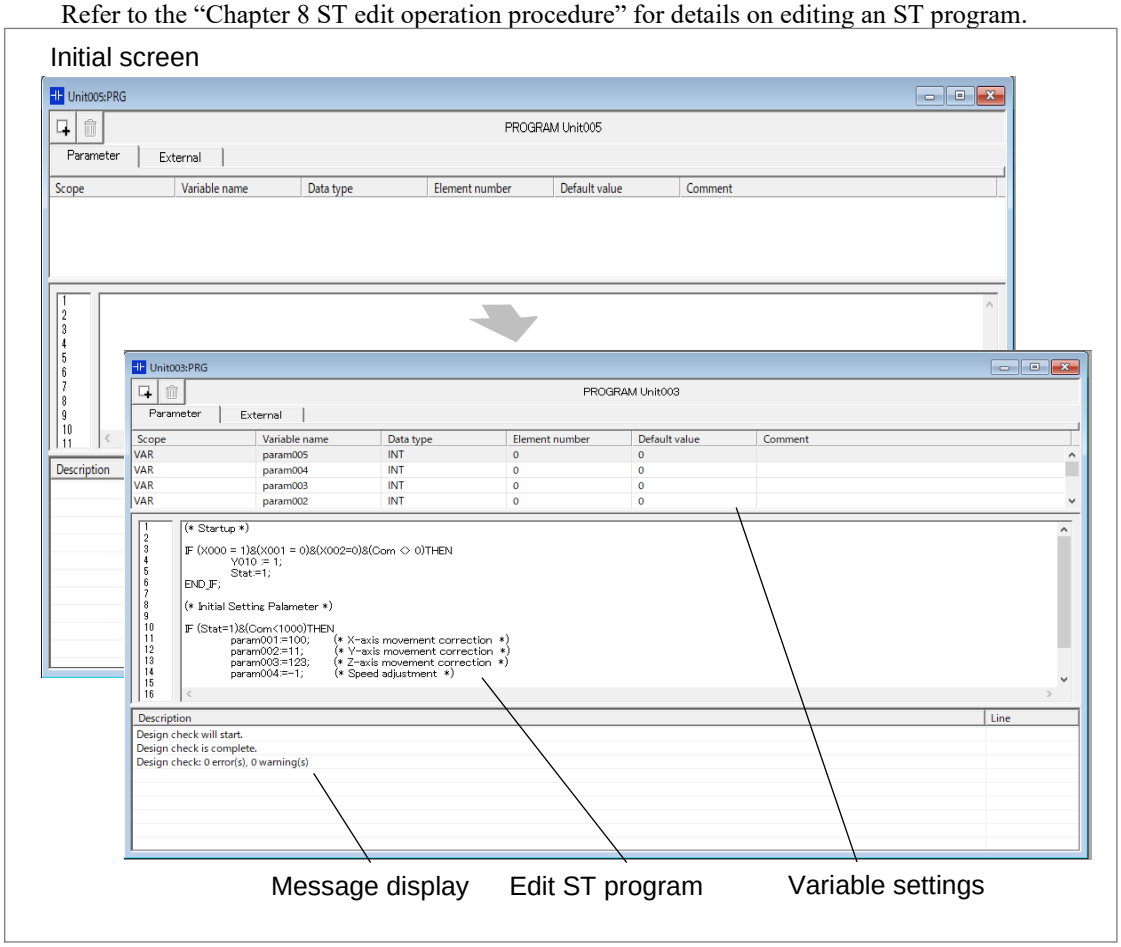
No.	Tab	Item	Description
1	Common	Function key display	Selects whether to use icons or text on the tool bar for the function keys in the LD/ST Programming Editor.
2		Change monitoring	Selects whether to use **[F2]** or **[F3]** as the function key to monitor the program.
3		Perform count up operation on timer per scan	Selects one of the following count up methods on the timer when the simulation is executed. Checkbox not ticked: Counts up based on the actual measured scan time after the input time is carried out. Checkbox ticked: User sets the time count per scan.
4		Count up time per scan	Sets the time count per scan between 1 and 100 ms. The unit is in ms.
5	ST	Font	Sets the font and font size used in the ST program display.

2. Click on the <OK> button or <Apply> button.
   After clicking on the <Apply> button, the <Option settings> dialog does not close and the LD/ST Programming Editor display is changed.
   After clicking on the <Cancel> button, the setting changes in the <Option settings> dialog are discarded.

(NOTE 1) The <Function key display> setting under the <Common> tab is only valid when editing an LD program.
(NOTE 2) When changing the display font in the ST program, the characters may not fit inside the frame depending on the font and size that are set.
(NOTE 3) The <Option settings> is a common setting in the PLC Programming software D00 regardless of the project.

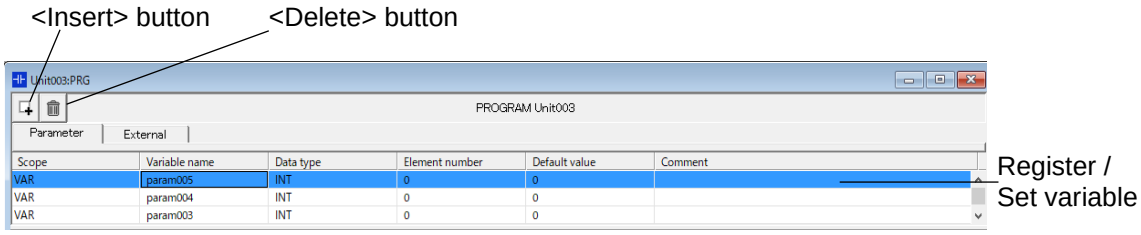
5.6.2.2 Edit ST programs

The ST programs can be edited in the LD/ST Programming Editor.
This section provides an overview on how to edit an ST program.
Refer to the “Chapter 8 ST edit operation procedure” for details on editing an ST program.



- Variable settings
  This section describes how to edit variables.
  Perform one of the following operations to create or add a new variable.

  Operation procedure
  Right-click the mouse on the variable setting field, and then select <Insert> from the context menu.
  Or, press the <Insert> button at the top of the variable edit area.



  Refer to “8.2 Variable settings” for details on the variable settings.

- ST program input method (overview)
  The user can create an ST program using the same input method as in a commercially available text editor.
  Refer to the following input example when setting a variable or OM in an ST program.

```
Ex:     IF (X000=1) THEN
            Y010:=1 ;
            DATA:=10 ;
            END_IF;
```

  Refer to “8.3.1 Procedure to create ST programs” for details on the ST program input method.
  Refer to “Chapter 12 Commands (ST)” in the PLC System Manual for details about the ST program syntax and each command.

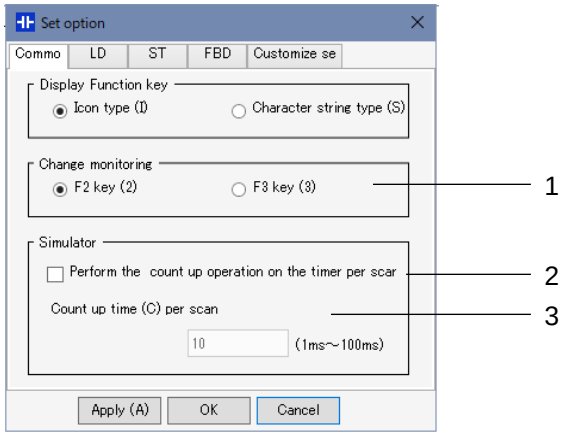
5.6.3 Create FBD Programs

5.6.3.1 Edit screen adjustment (option settings)

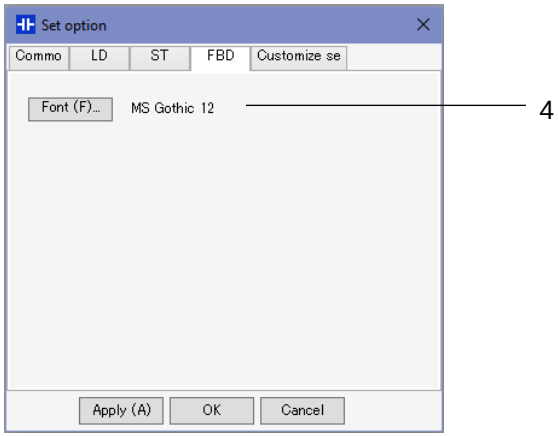
The user can adjust the display and some operations on the edit screen in order to make the FBD program editing more user friendly.
These settings are applied to all the FBD program edit windows.

Operation procedure

1. Select <Option settings> from the <Auxiliary> menu in the PLC Programming software D00.
   <Common> tab



<FBD> tab



No.	Tab	Item	Description
1	Common	Change monitoring	Selects whether to use **[F2]** or **[F3]** as the function key to monitor the program.
2		Perform count up operation on timer per scan	Selects one of the following count up methods on the timer when the simulation is executed. Checkbox not ticked: Counts up based on the actual measured scan time after the input time is carried out. Checkbox ticked: User sets the time count per scan.
3		Count up time per scan	Sets the time count per scan between 1 and 100 ms. The unit is in ms.
4	FBD	Font	Sets the font and font size used in the FBD program display.

2. Click on the <OK> button or <Apply> button
   After clicking on the <Apply> button, the <Option settings> dialog does not close and the FBD Programming Editor display is changed.
   After clicking on the <Cancel> button, the setting changes in the <Option settings> dialog are discarded.

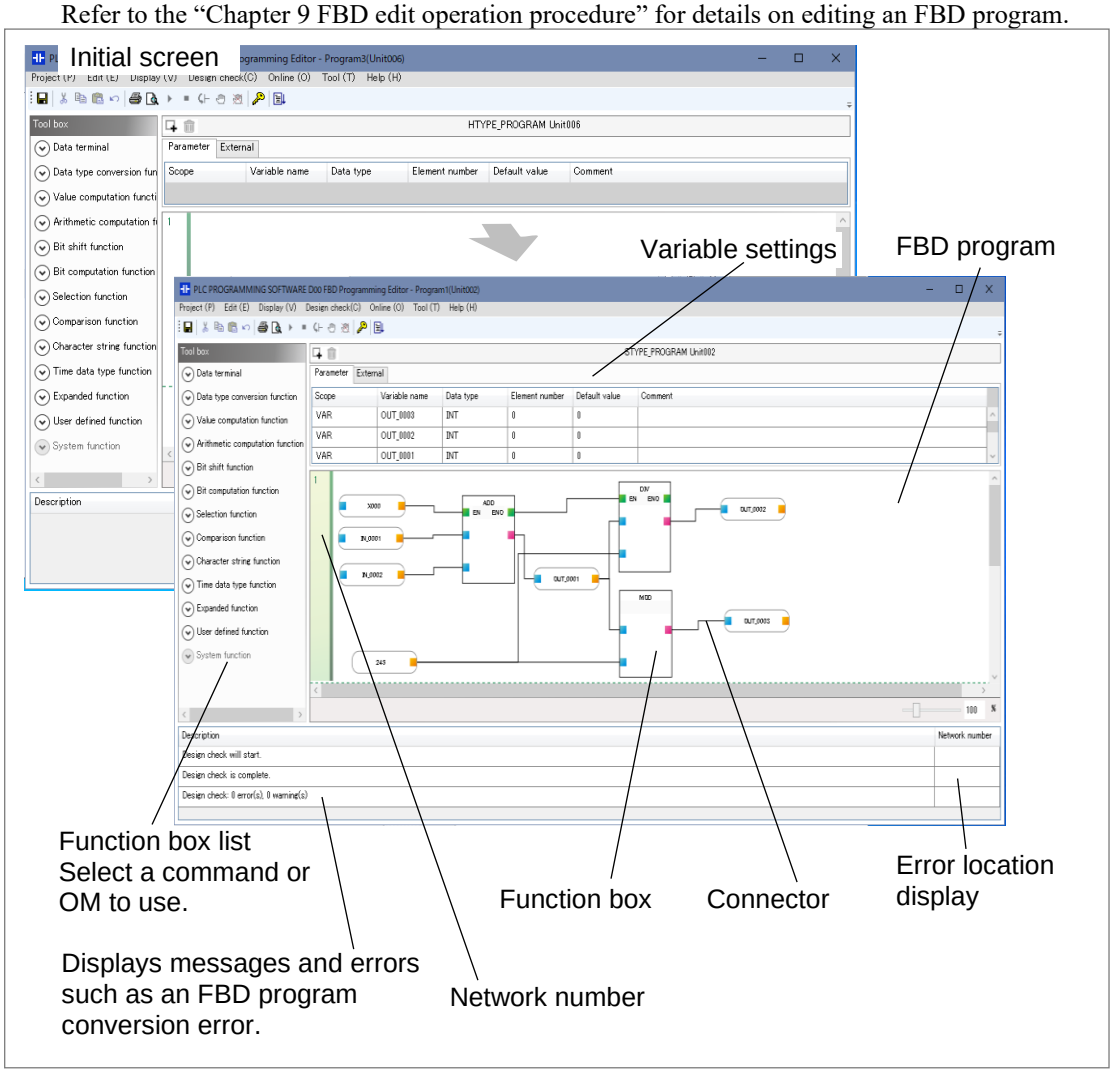
(NOTE 1) The <Function key display> setting under the <Common> tab is not reflected in the FBD Programming Editor because the setting is related to the LD/ST Programming Editor.

(NOTE 2) When changing the display font in the FBD program, the characters may not fit inside the frame depending on the font and size that are set.

(NOTE 3) The <Option settings> is a common setting in the PLC Programming software D00 regardless of the project.

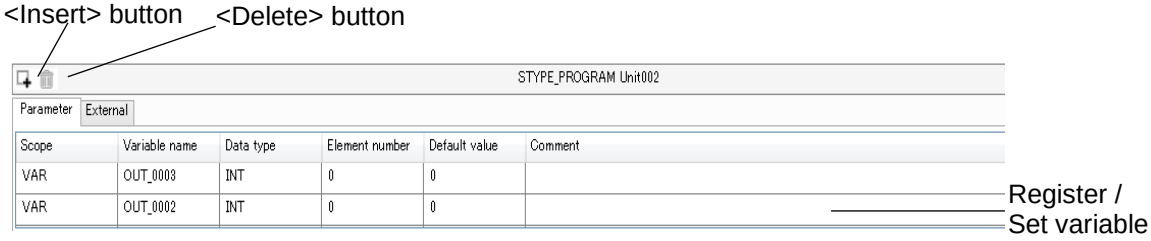
5.6.3.2 Edit FBD programs

The FBD programs can be edited in the FBD Programming Editor.
This section provides an overview on how to edit an FBD program.
Refer to the "Chapter 9 FBD edit operation procedure" for details on editing an FBD program.



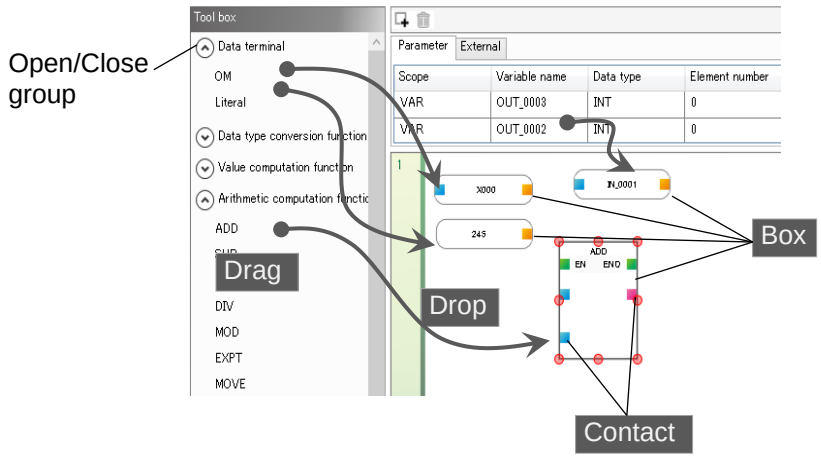
- Variable settings
  This section describes how to edit variables.
  Perform the following operations to create or add a new variable.

  Operation procedure
  Right-click the mouse on the variable setting field, and then select <Insert> from the context menu.
  Or, click on the <Insert> button at the top of the variable edit area.

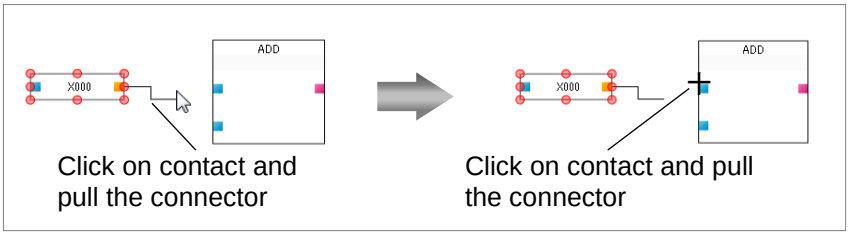


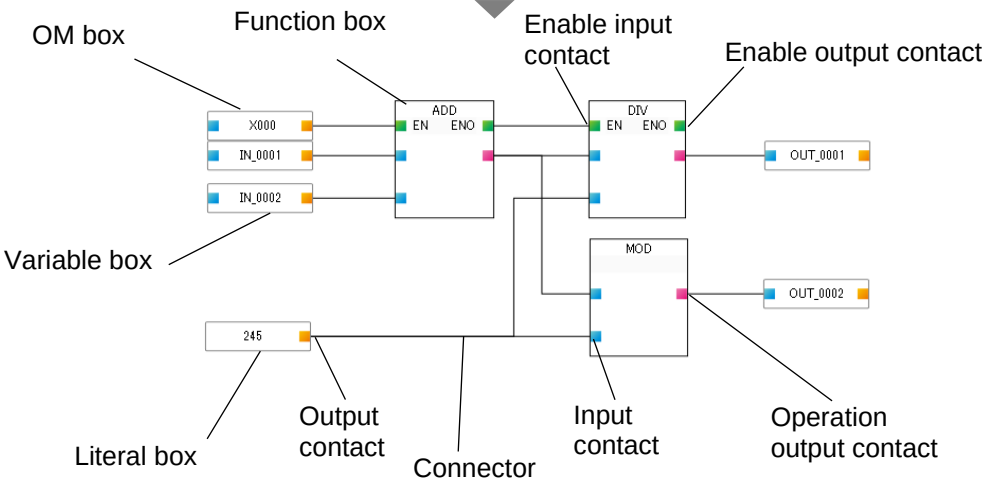
Refer to “8.2 Variable settings” for details on the variable settings.

- Command input method (overview)



Use drag & drop to create a box in the edit area, and set the box position and connect it with a “Connector”.
When connecting with the “Connector”, click the mouse at each contact position, and then re-click the mouse at the contact position for the connecting box. (Valid when mouse cursor is “+”.)





Refer to “Chapter 12 Commands (FBD)” for details about the FBD program syntax and each command.

5.7 Program Check

The program check function in the PLC Programming software D00 is equipped with a “Double coil check”.
The double coil check is a function that checks if multiple places have been written for the same OM.
However, it checks at conversion if there is an invalid statement in the program for the POU that was created.
Refer to “5.9 Conversion” for details about conversion.
Refer to “10.10 Program Check Functions” for details on the program check function.

5.7.1 Double Coil Check

5.7.1.1 Check items

The double coil check is executed for the following items.

Check item	Description	
Double coil check for LD program	Checks if there is a duplication for the OM that is set to an OUT command (including local timer/local counter) for the POU being edited.	
Double coil check for all POUs	Checks POUs registered to the <Program execution settings> and the user defined and system defined functions that are used in those POUs.	
	LD program	Corresponding OMs: Y、BY(*1) Corresponding commands: OUT, SET, RST, PLS, PLSN and SFT(P) commands, application command destination and ST language processing in CALC commands
	ST program FBD program	Corresponding OMs: Y、BY(*1) Corresponding commands: Assignment, function output argument and function input/output argument

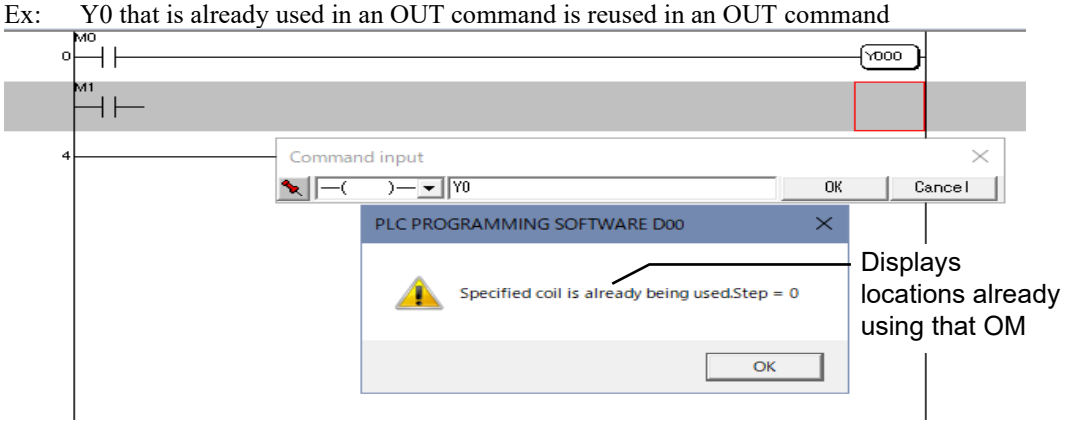
*1 OMs with an index are not differentiated. For example, a double coil is detected for “OUT Y0IX0” and “OUT Y0IX1”.

5.7.1.2 Check Timing / Check Results

The double coil check is executed for the following items.

- Double coil check for LD program
  The double coil check is carried out when the the user clicks on the <OK> button in the <Command input> dialog. When an LD program OUT command (including local timers and local counters) is input, if an OM is set that is already used in the output, then a warning is displayed.
  Correct the OM when necessary.

Ex: Y0 that is already used in an OUT command is reused in an OUT command



When the same OM is used in multiple OUT commands, the lowest step number where it is used is displayed. If the OM is being used on a circuit block that is unconverted or with a design that is not checked, then “?” is displayed at the locations where it is used.

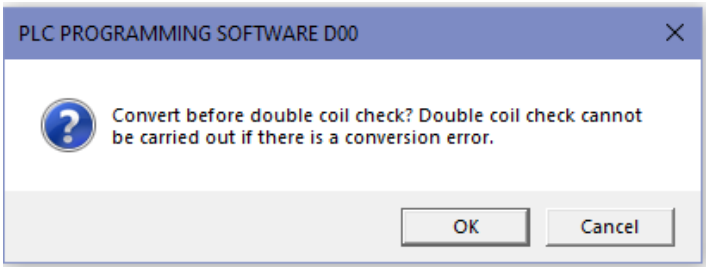
The double coil check shown in this section is not carried out at other times, such as for a circuit paste operation or for a conversion operation. In addition, the double coil check does not check if there is a double coil between this POU and another.
For these scenarios, perform the check using the “Double coil check for all POUs” (described later).

(NOTE 1) OMs being used in other commands (besides OUT commands) are not checked.
(NOTE 2) When an OM with an index is used in an OUT command, then that OM (with the same content) is also targeted in the check.
For example, when “OUT Y0IX0” is used at multiple locations, that is targeted in the check.
“OUT Y0IX0” and “OUT Y0IX1” are not checked if another OM is being used.

- Double coil check for all POUs
  This double coil check is executed at the following times.
- Executes from the <Double coil check> menu under <Edit> in the PLC Programming software D00
- Executes automatically when the program write operation is executed or when the write during run operation is executed

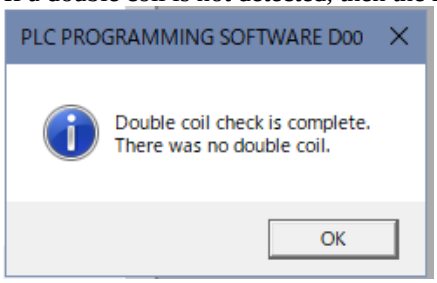
Refer to “5.10.2 Write programs” and “5.10.4 Write during run” for details about the program write execution and the write during run operation.

Conversion is required to perform a double coil check. When a double coil check is carried out for an unconverted POU, a confirmation message appears asking whether to perform a conversion.

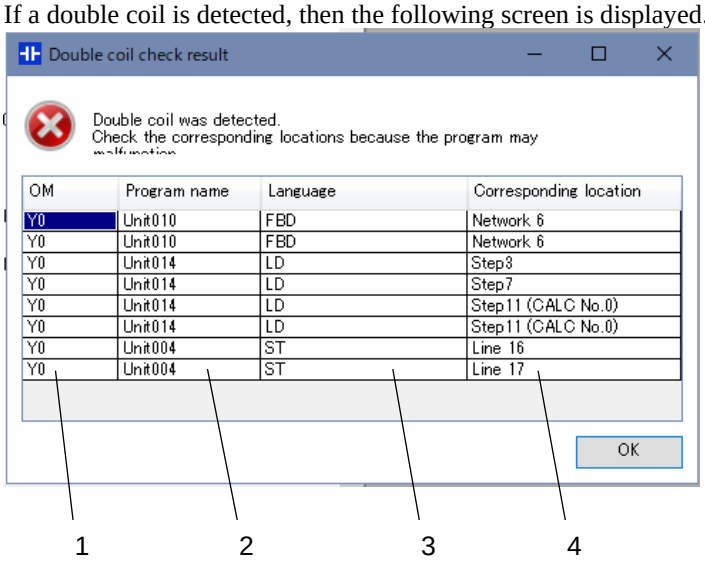


Execute the conversion to perform the double coil check.
However, if there is a conversion error, the double coil check will not be carried out. Correct the error locations and perform the conversion and the double coil check again.

If a double coil is not detected, then the following message is displayed.



If a double coil is detected, then the following screen is displayed.



No.	Item	Description
1	OM	Displays an OM name that is written from multiple locations.
2	Program name	Displays a POU name where a double coil is detected.
3	Language	Displays the POU program language.
4	Double coil location	Displays the location where a double coil is detected. • Step number is displayed when using an LD program. The CALC command number is also displayed when a double coil is detected in a CALC command. • Line number is displayed when using an ST program. • Network number is displayed when using an FBD program. After double-clicking on a line, the POU where a double coil is detected is opened with the LD/ST Programming Editor or with the FBD Programming Editor. A maximum of 1023 double coil locations can be displayed. If 1024 double coils or more are detected, then any data on those coils (that exceed the maximum) is not displayed.

After clicking on the <OK> button, the <Double coil check result> dialog closes.

5.7.2 Design Check

5.7.2.1 Check items

The design check function is for making sure there are no invalid notations in the POU that was created.
The design check is carried out for the POU that is displayed at the front in the LD/ST Programming Editor/FBD Programming Editor and for the user defined function / system defined function that is called from that POU.
However, if the design check is carried out for a POU that is set in the <Program execution settings> dialog, then the check is carried out for all the POUs that are set in the <Program execution settings> dialog and for the user defined functions / system defined functions that are called from those POUs.

5.7.2.2 Check timing / check results

The design check is carried out when one of the following operations is performed in the LD/ST Programming Editor or FBD Programming Editor.

- Select <Design check> from the <Design check> menu.
- Press the **[F4]** key.

Note that the design check is also carried out if one of the following operations is performed when editing an LD program in the LD/ST Programming Editor.

- Press **[End]** key
- Press **[Shift]** + **[End]** keys
- Execute replacement
- Change all set values using the **[Timer/Counter list]**

When the design check is carried out and an error or a warning is detected, an error message is displayed, or an error or warning description is displayed in the message display area for the LD/ST Programming Editor or FBD Programming Editor.
The error or warning description varies depending on the program language.
Refer to “5.9.1 Conversion” for further details.

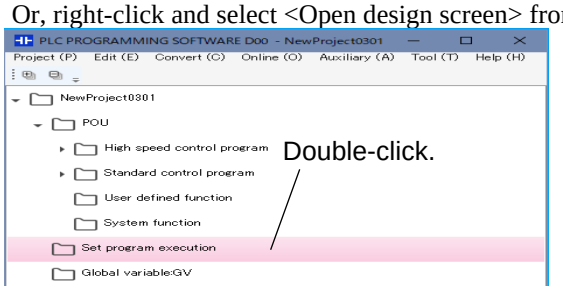
5.8 Program Execution Settings

Up to two high speed control programs and up to three standard control programs created in the PLC Programming software D00 EDITOR can be executed in the NC(BUILT IN PLC). The execution order of the high speed and standard control programs is set in the PLC Programming software D00.

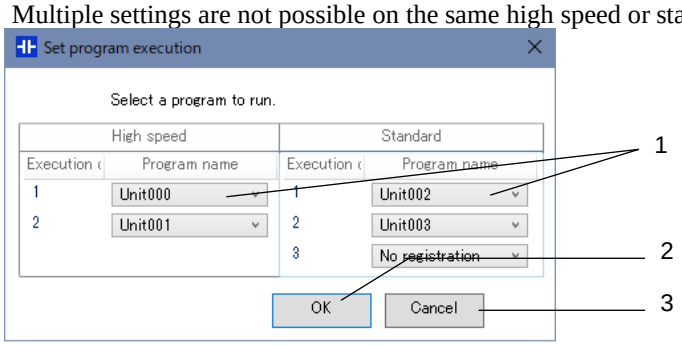
(NOTE) Refer to “Chapter 3 Built-in PLC” in the PLC System Manual for details about the NC (BUILT IN PLC) operation and processing.

Operation procedure

1. Double-click on <Program execution settings> in the PLC Programming software D00.
   Or, right-click and select <Open design screen> from the displayed menu.



2. Set the execution order in the <Program execution settings> dialog.
   Multiple settings are not possible on the same high speed or standard control program.



No.	Item	Description
1	Program name	Click on <▼> to display a list of high speed/standard control programs that were created. Select a high speed or standard control program to execute. Select <No registration> to cancel the execution of the high speed or standard control program that was set.
2	<OK> button	Applies the setting changes, and the dialog closes.
3	<Cancel> button	Discards the setting changes, and the dialog closes.

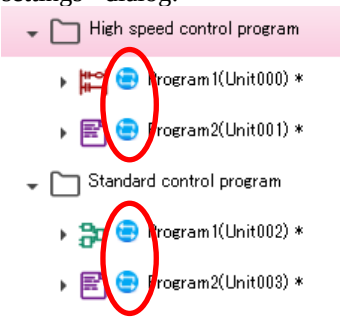
3. When the dialog to name and save a project appears, click on the <OK> button.

(NOTE 1) In the <Program execution settings> dialog, even if the high speed or standard control programs are not set sequentially (for example, standard control program is set to positions 1 and 3 in the execution sequence but <No registration> is set for position 2), the specified order is still executed in the NC(BUILT IN PLC).

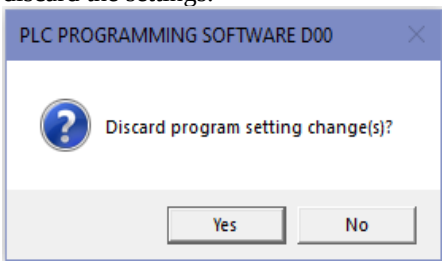
(NOTE 2) When a high speed or standard control program (set in the <Program execution settings> dialog) is deleted from the PLC Programming software D00, then it is automatically deleted from the <Program execution settings> dialog.

(NOTE 3) When a program name is changed in the PLC Programming software D00 for a high speed or standard control program (set in the <Program execution settings> dialog), then the name change is automatically reflected in the <Program execution settings> dialog.

(NOTE 4) The symbol is displayed in the tree in the PLC Programming software D00 for high speed or standard control programs that are set in the <Program execution settings> dialog.



(NOTE 5) When the user clicks on the <Cancel> button after editing in the <Program execution settings> dialog, a confirmation message appears asking whether or not to discard the settings.



When <Yes> is selected, the settings are discarded and the <Program execution settings> dialog closes.
When <No> is selected, it goes back to the <Program execution settings> dialog.

5.9 Conversion

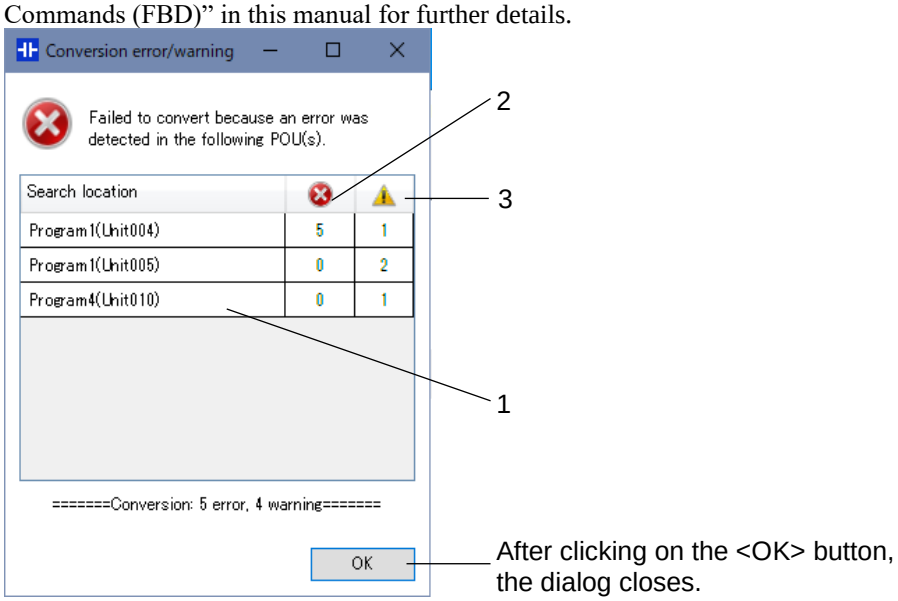
5.9.1 Conversion

Conversion is required to use the NC(BUILT IN PLC) to run POUs created in the PLC Programming software D00 EDITOR. Conversion refers to converting the POUs (created in a language in the PLC Programming software D00 EDITOR) to code (intermediate code) that allows execution in the NC(BUILT IN PLC).
This conversion is for high speed or standard control programs set in the <Program execution settings> dialog and the user defined and system defined functions that are called by those high speed or standard control programs.
In addition, when conversion is executed, a check is carried out to look for invalid statements in the POU (for conversion). If there is an error, the intermediate code will not be created.
The POU is automatically saved when converted.
After conversion, save the project.
Conversion is carried out in the PLC Programming software D00.

Operation procedure

1. Select <Convert> from the <Convert> menu in the PLC Programming software D00.
   Or, press the **[F4]** key.
2. The design check is carried out for the POU to be converted.
3. If an error is not detected during the design check, conversion is executed.

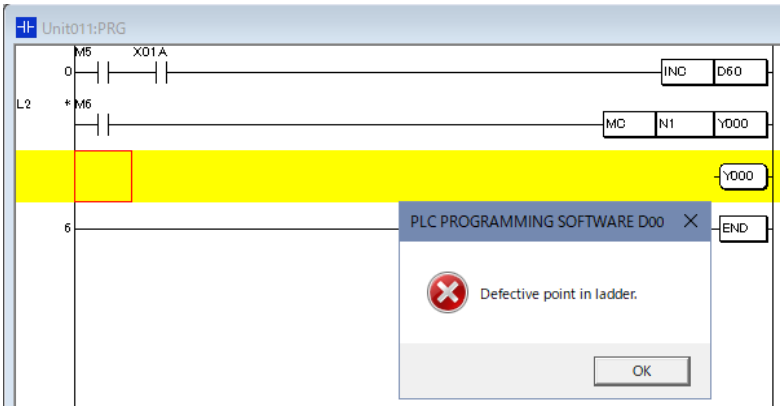
If a conversion error is detected, the <Conversion error/warning> dialog is displayed, showing the number of errors per POU.
If only a warning is detected, the <Conversion error/warning> dialog is displayed, showing the number of warnings per POU, but the conversion processing is carried out.
The error or warning description varies depending on the program language. Refer to “Chapter 11 Commands (LD)” or “Chapter 12 Commands (ST)” in the PLC System Manual, or “Chapter 12 Commands (FBD)” in this manual for further details.



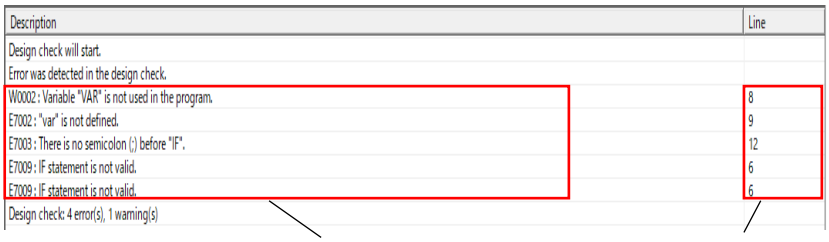
No.	Item	Description
1	Error or warning location	Displays a POU name where an error or warning is detected. After double-clicking on a line, the edit window for the POU is opened in the LD/ST Programming Editor or the FBD Programming Editor.
2	Number of errors	Displays the number of errors detected in the POU.
3	Number of warnings	Displays the number of warnings detected in the POU.

When the POU edit window is opened from the <Conversion error/warning> dialog, an error message is displayed, or the design check results are displayed in the message display area.
If there is an error, the project data that was created cannot be executed in the NC(BUILT IN PLC). Therefore, check the error description and correct the error.
Correct the warning when necessary.
After correcting it, re-execute the conversion.

- When using the LD language
  The error message is displayed, and the circuit block with the error is highlighted in yellow.
  In addition, the cursor is positioned at the error location.



- When using the ST/FBD language
  The check results are displayed in the message display area for the LD/ST Programming Editor and FBD Programming Editor.
  After double-clicking on the line with the error or warning, it jumps to the error or warning location.



Description	Line
Design check will start.	
Error was detected in the design check.	
W0002 : Variable "VAR" is not used in the program.	8
E7002 : "var" is not defined.	9
E7003 : There is no semicolon (;) before "IF".	12
E7009 : IF statement is not valid.	6
E7009 : IF statement is not valid.	6
Design check: 4 error(s), 1 warning(s)	

"Exxxx:" indicates the errors and "Wxxxx:" indicates the warnings.

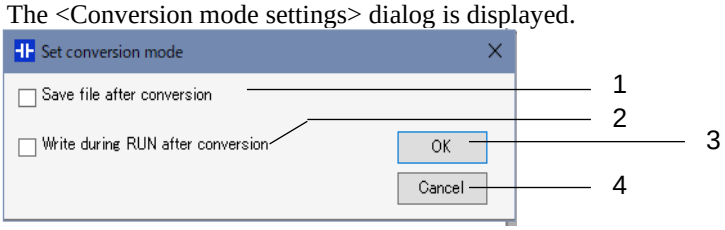
Shows the corresponding locations.
ST language: Line number
FBD language: Network number

5.9.2 Conversion Mode Settings

In the PLC Programming software D00 EDITOR, the user can set the processing to automatically carry out when conversion is executed. This is called the conversion mode settings.
The conversion mode settings are common settings in the PLC Programming software D00 EDITOR regardless of the project.

Operation procedure

1. Select <Set conversion mode> from the <Auxiliary> menu in the PLC Programming software D00.
2. The <Conversion mode settings> dialog is displayed.



No.	Item	Description
1	Save file after conversion	Automatically saves the whole project after conversion when the checkbox is ticked. However, if there is a conversion error, it does not save the data.
2	Write during RUN after conversion	Write during run operation is carried out to the NC(BUILT IN PLC) after conversion when the checkbox is ticked. Refer to “5.10.4 Write during run” for details on the write during run operation.
3	<OK> button	Applies the setting changes, and the dialog closes.
4	<Cancel> button	Discards the setting changes, and the dialog closes.

3. Click on the <OK> button.

5.10 Write/Read Programs

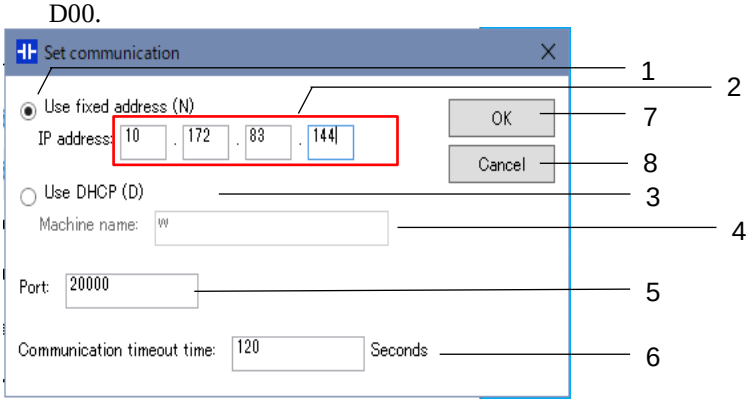
This section describes how a program created in the PLC Programming software D00 EDITOR can be written to the NC(BUILT IN PLC) or read in the PLC Programming software D00 Linux EDITOR.

5.10.1 Communication Settings with NC

This section describes the communication port settings on the PLC Programming software D00 EDITOR side when connecting between a system with the PLC Programming software D00 Linux EDITOR and a device that has the PLC Programming software D00 EDITOR installed.
The communication port settings are configured in the PLC Programming software D00.

Operation procedure

1. Select the <Set communication> from the <Online> menu in the PLC Programming software D00.



No.	Item	Description
1	Use fixed address	Displays “Enable” in the IP address input field and “Disable” in the machine name input field when this is selected.
2	IP address	Sets the IP address.
3	Use DHCP	Displays “Enable” in the machine name input field and “Disable” in the IP address input field when this is selected.
4	Machine name	Enter the machine name when using the DHCP.
5	Port	Enter the port number. <Use fixed address> or <Use DHCP> can be used as the data input field. Only integers can be entered, and the data input range is between 20000 and 29999.
6	Communication timeout time	Sets the time for the communication timeout. <Use fixed address> or <Use DHCP> can be used as the data input field. Only integers can be entered, and the data input range is between 15 and 180.
7	<OK> button	Applies the setting changes, and the dialog closes.
8	<Cancel> button	Discards the setting changes, and the dialog closes.

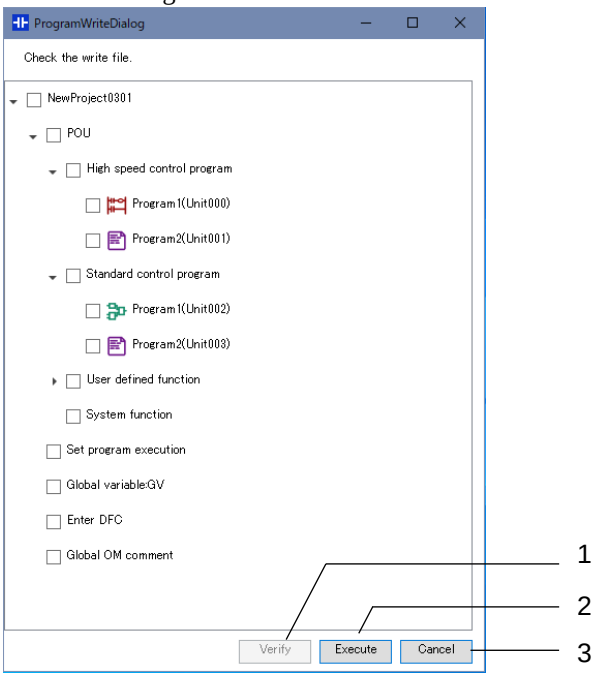
2. Click on the <OK> button.

5.10.2 Write Programs

This section describes how to operate and write programs to the NC(BUILT IN PLC).
POUs (LD program, ST program or FBD program) that are created can be written from the PLC Programming software D00.
All POUs set in the <Program execution settings> dialog are written.
The NC(BUILT IN PLC) must be in a stopped state in order to write a program.

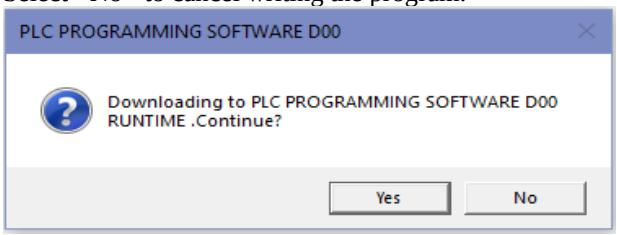
- Write to simulation
  When there is a checkmark "✓" next to the <Simulation> in the <Tool> menu (for each tool), then the program is written to the simulation.

  Operation procedure
  1. Select <Write program> from the <Online> menu in the PLC Programming software D00.
  2. Selection dialog

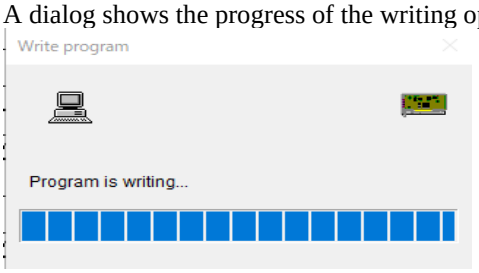


No.	Item	Description
1	<Verify> button	
2	<Execute> button	Writes the files with a ticked checkbox ☑.
3	<Cancel> button	Closes the dialog and cancels the program writing operation.

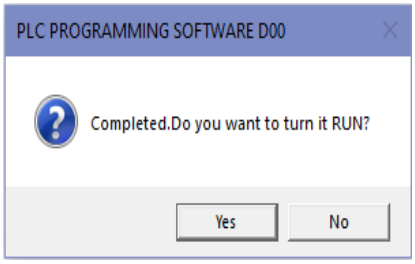
  3. A confirmation message appears asking whether or not to write to the NC(BUILT IN PLC).
     Select <Yes> to write the program.
     Select <No> to cancel writing the program.



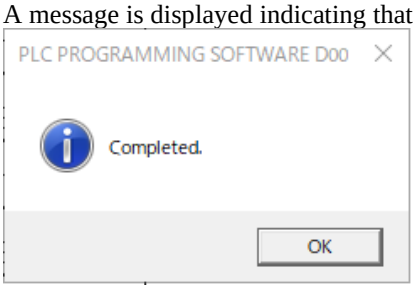
4. A dialog shows the progress of the writing operation.



5. When the program writing is completed, a confirmation appears asking whether or not to change the NC(BUILT IN PLC) to a RUN state.
Select <Yes> to change it to a RUN state.
Select <No> to keep the NC(BUILT IN PLC) in a STOP state.



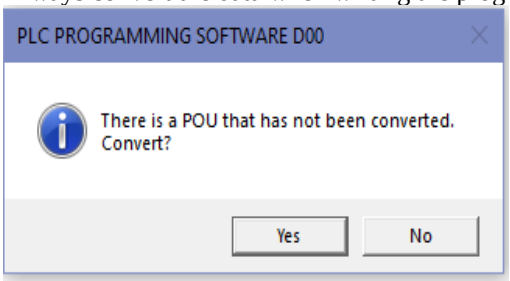
6. A message is displayed indicating that the program writing is complete.



(NOTE 1) After the program is written, take the necessary safety precautions when changing the NC(BUILT IN PLC) to RUN state.
(NOTE 2) Before writing a program, be sure to save the whole project.
(NOTE 3) All POUs set in the <Program execution settings> dialog are written. Even when only some POUs are changed, there is no function to write those POUs individually.

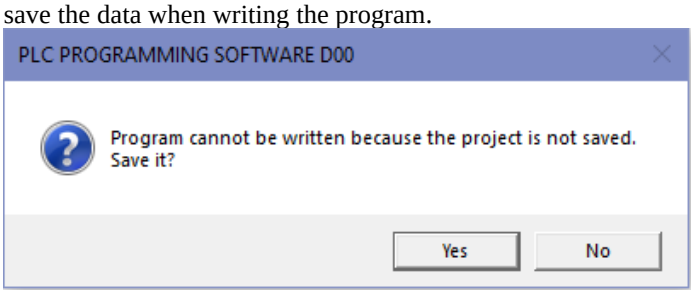
In addition to the procedure shown above, the following confirmation message is displayed depending on the program writing execution, the project status and the NC(BUILT IN PLC) state.

- When a project is not converted
  When the project data is not converted, a message prompts the user about converting the data. Always convert the data when writing the program.



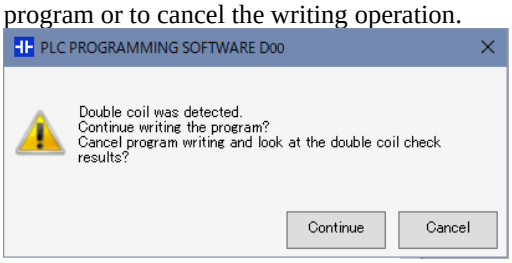
Select <Yes> to convert the data and proceed with the processing to write the program.
Select <No> to cancel the conversion and cancel the processing to write the program.

- When a project is not saved
  When the project data is not saved, a message prompts the user about saving the data. Always save the data when writing the program.

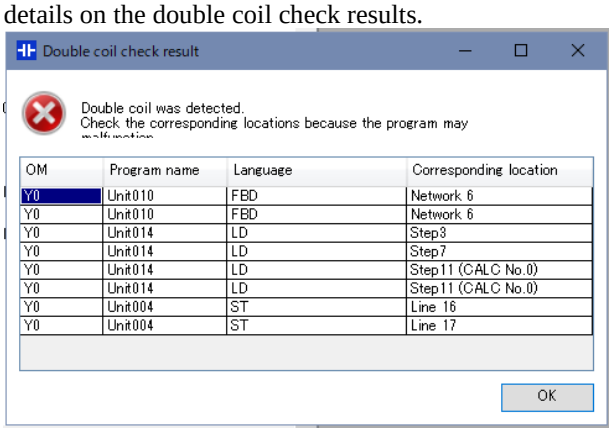


  Select <Yes> to save the data and proceed with the processing to write the program.
  Select <No> to cancel saving the data and cancel the processing to write the program.

- When a double coil exists
  When a double coil exists, a confirmation message appears asking whether to write the program or to cancel the writing operation.



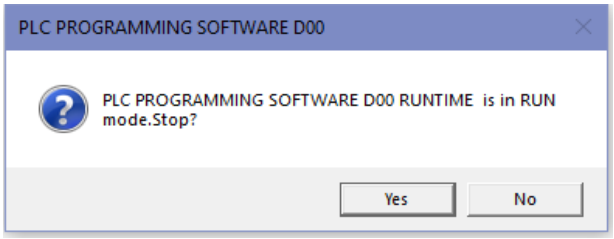
  Select <Continue> to proceed with the processing to write the program.
  Select <Cancel> to cancel the processing to write the program. A dialog is displayed indicating the location(s) where there is a double coil. Refer to “5.7.1 Double coil check” for details on the double coil check results.



OM	Program name	Language	Corresponding location
Y0	Unit010	FBD	Network 6
Y0	Unit010	FBD	Network 6
Y0	Unit014	LD	Step3
Y0	Unit014	LD	Step7
Y0	Unit014	LD	Step11 (CALC No.0)
Y0	Unit014	LD	Step11 (CALC No.0)
Y0	Unit004	ST	Line 16
Y0	Unit004	ST	Line 17

(NOTE) In the PLC Programming software D00, executing a program with a double coil is prohibited.
Note, a program could malfunction depending on the program content. Therefore, the user should check the program content carefully in the event that a program with a double coil is executed in the NC(BUILT IN PLC).

- When NC(BUILT IN PLC) is in a RUN/DEBUG-RUN state
  When the NC(BUILT IN PLC) is in a RUN/DEBUG-RUN state, a confirmation message appears asking whether or not to change the NC(BUILT IN PLC) to a STOP state. Always change it to a STOP state when writing the program.



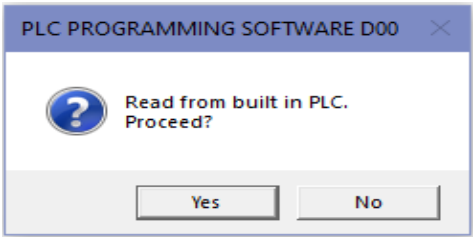
Select <Yes> to change the NC(BUILT IN PLC) to a STOP state in order to proceed with the processing to write a program.
Select <No> to keep the NC(BUILT IN PLC) in an operating state and to cancel the processing to write a program.

5.10.3 Read Programs

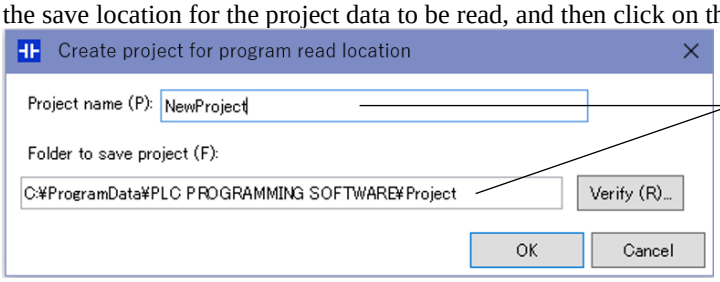
The user can read all the files in the PLC Programming software D00 Linux EDITOR.
The PLC Programming software D00 EDITOR must also be in edit mode (mode to edit a program) to read a program.

Operation procedure

1. Select <Read program> from the <Online> menu in the PLC Programming software D00.
2. Click on the <Yes> button in the confirmation dialog.

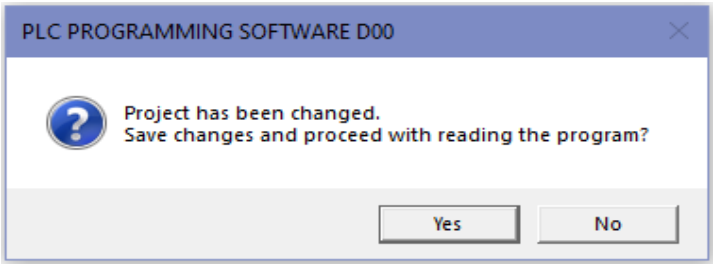


3. In the <Create location to read program> dialog that is displayed, enter the project name and the save location for the project data to be read, and then click on the <OK> button.



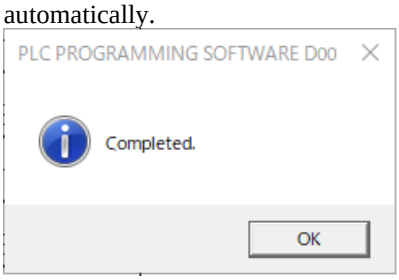
Specify the project name and the folder for saving the project data to be read.

4. If a project was already opened in the PLC Programming software D00 EDITOR, that project closes automatically.
   When the project data is not saved, a confirmation message appears about saving the data.
   Select <Yes> to execute the program read operation.
   Select <No> to cancel reading the program.

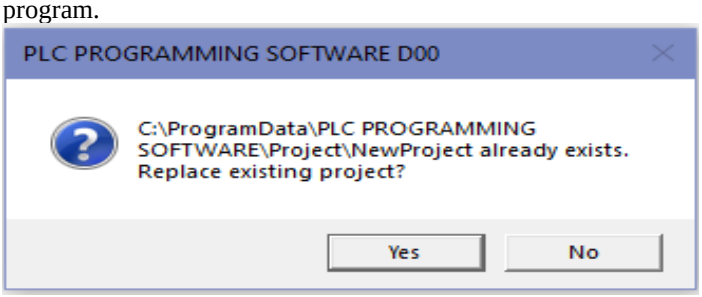


5. The <Read program> dialog is displayed.
   Select <Execute> to execute the program read operation.
   Select <Cancel> to cancel reading the program.

6. When the save operation is complete, a message appears indicating that it is complete.
   When the message is closed, the PLC Programming software D00 reads that project data automatically.



(NOTE) If an existing project name is specified in the <Create location to read program> dialog, a message appears asking whether to overwrite the data. Select <Yes> to overwrite the project data and read the program. Select <No> to cancel reading the program.



5.10.4 Write During Run

The write during run operation is a function that allows a program to be written while the NC(BUILT IN PLC) is in a RUN state or DEBUG-RUN state.
The write during run operation is only possible for changes to the logic of a high speed or standard control program.
Refer to “11.1.3 Write During Run” for further details.

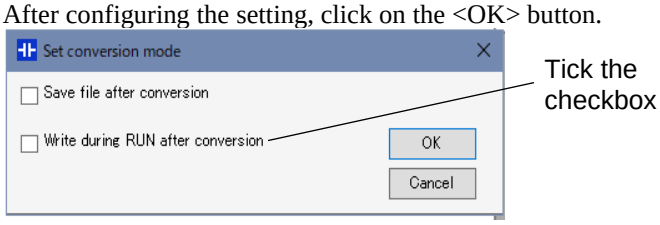
(NOTE 1) When using the write during run operation, take the necessary safety precautions with the device that has the NC(BUILT IN PLC) and also with the operating environment.

(NOTE 2) The write during run operation is processed during the END processing in the NC(BUILT IN PLC). As a result, the size of the program that is written affects the scan time.

(NOTE 3) When the LD program is changed by the write during run operation, then the rising and falling edge command (included in the edited circuit block) status is cleared. Take into consideration that these rising and falling edge commands are cleared when performing the write during run operation.

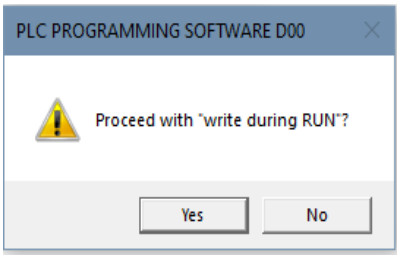
Operation procedure

1. Read the project in the NC(BUILT IN PLC) by accessing the <Read program> menu under the <Online> menu in the PLC Programming software D00 (recommendation).
   Or, open the project that is running in the NC(BUILT IN PLC) by accessing <Open project> under the <Project> menu.
2. Select <Set conversion mode> from the <Auxiliary> menu in the PLC Programming software D00.
   When the <Conversion mode settings> dialog is displayed, set to <Write during RUN after conversion>.
   After configuring the setting, click on the <OK> button.

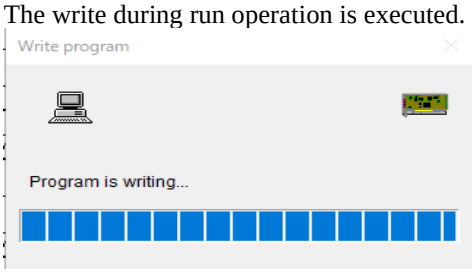


(NOTE) The conversion mode settings are common settings in the PLC Programming software D00 EDITOR regardless of the project.
Check the conversion mode settings and if the write during run operation is not needed, untick the checkbox (no checkmark) for <Write during RUN after conversion> to make sure the write during run operation is not carried out unintentionally.

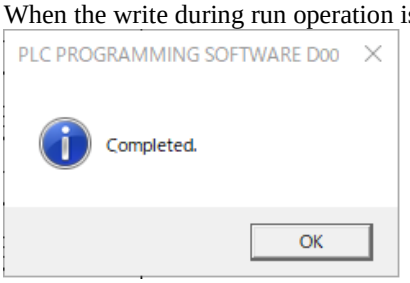
3. Modify or edit a high speed or standard control program in the LD/ST Programming Editor or in the FBD Programming Editor.
   When changing the POU in the LD language, modify or edit the program for one circuit block.
4. Select <Convert> from the <Convert> menu in the PLC Programming software D00 to execute the conversion.
5. When the data is compared with the project stored in the NC(BUILT IN PLC) and that data matches, then a confirmation message appears asking whether or not to perform the write during run operation.
   Select <Yes> to perform the write during run operation.
   Select <No> to cancel the write during run operation.



6. The write during run operation is executed.



7. When the write during run operation is complete, the following message is displayed.



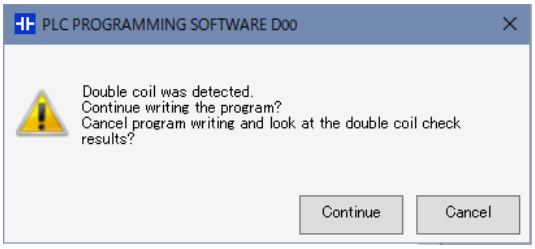
Be aware of the following points when performing the write during run operation.

- Read program
  A program written with the write during run operation cannot be read using the <Read program> function. The project data, before the write during run operation, is read. Therefore, we recommend ultimately using <Write program> to write the program.

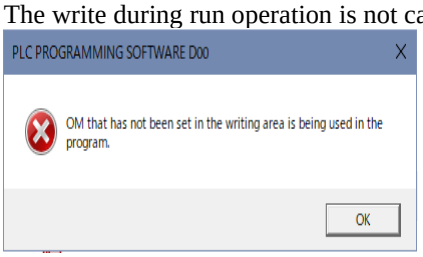
- Exiting NC(BUILT IN PLC)
  When a program is written using the write during run operation, after exiting and then re-launching the NC(BUILT IN PLC), the project data from before the write during run operation is used for operation.
  Therefore, we recommend ultimately using <Write program> to write the program.

- Rising edge/Falling edge command
  When the LD program is changed by the write during run operation, then the rising and falling edge command (in the edited circuit block) status is cleared.

- When a double coil is detected at conversion
  When a double coil is detected at conversion, the PLC Programming software D00 EDITOR displays a confirmation message asking whether or not to proceed with the write during run operation.
  Select whether to proceed with the write during run operation, or to cancel that operation and check the double coil location.

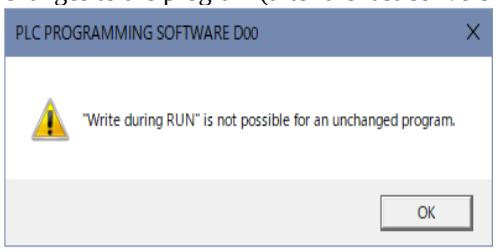


- When an OM that has not been set in the writing area is being used in the program
  The write during run operation is not carried out. Only conversion is carried out.

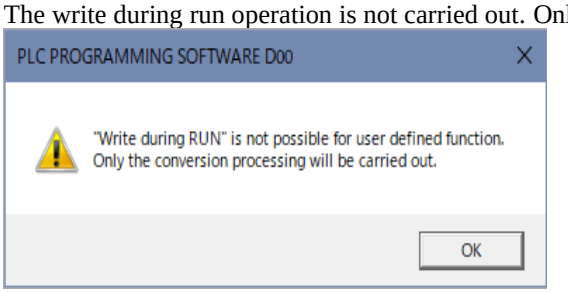


- When an error or warning is detected at conversion
  The write during run operation is not possible when an error is detected at conversion.
  The write during run operation is carried out when only a warning is detected at conversion.

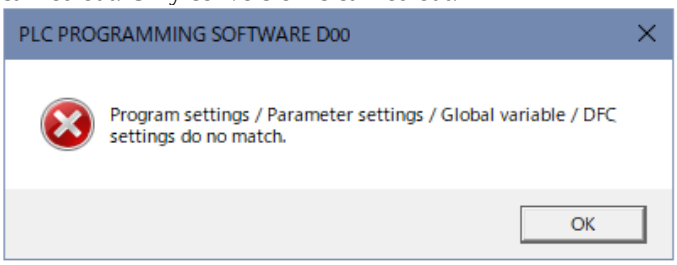
- When there are no changes to the program
  The write during run operation is not carried out, even after conversion, if there are no changes to the program (after the last conversion).



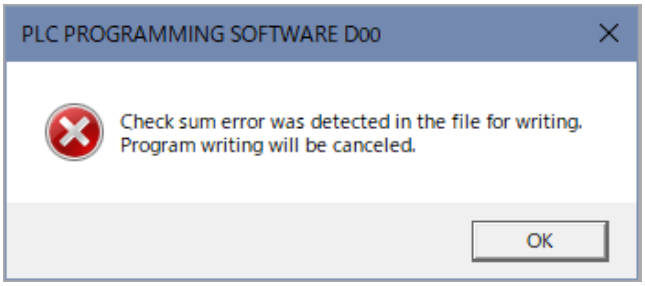
- When making changes to a user defined function or system defined function
  The write during run operation is not carried out. Only conversion is carried out.



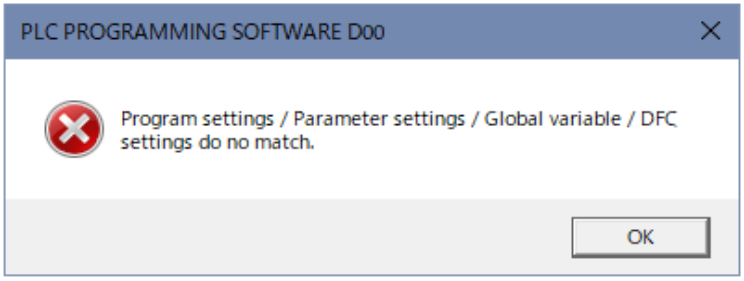
- When making changes to other items besides the POU
  If changes are made to the POU program and to the program execution settings, parameter settings, global variables and/or DFC registration, then the write during run operation is not carried out. Only conversion is carried out.



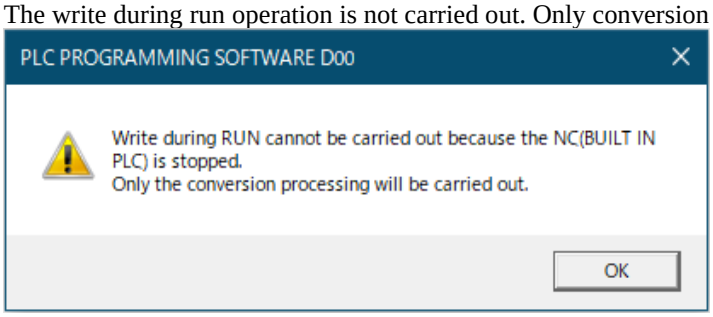
- When a file checksum error is detected for writing
  The write during run operation is not carried out. Correct and save the program, and then re-execute it.



- When a project is converted that is different from the NC(BUILT IN PLC)
  The write during run operation is not carried out. Only conversion is carried out.



- Converting while NC(BUILT IN PLC) is in a STOP state
  The write during run operation is not carried out. Only conversion is carried out.



- Converting while simulation is executing
  The write during run operation is not carried out for the simulation. Only conversion is carried out.

5.11 Launch Simulation

The user can launch a simulation to check the program operation.
Refer to “11.6 Simulation Function” for details on the simulation function.

(NOTE) The simulation function can be used to confirm whether the logic for the program is correct or not. However, there is a difference between the function and actual operation in the NC(BUILT IN PLC), and therefore the timing and/or operation do not necessarily match. Before starting the operation of a device with the NC(BUILT IN PLC), always be sure to thoroughly check the operation on the actual machine.

Simulation can be launched from the PLC Programming software D00, the LD/ST Programming Editor or the FBD Programming Editor.

Operation procedure
Select <Simulation> from the <Tool> menu for each tool.

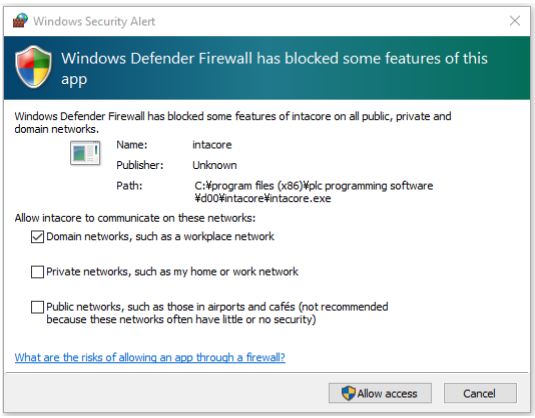
To exit simulation, select <Simulation> again while there is a checkmark “✓” next to <Simulation> in the <Tool> menu (for each tool) in order to deselect it.
After exiting simulation, the checkmark “✓” next to the <Simulation> disappears.

5

(NOTE 1) When the simulation is launched, the icon is displayed on the task bar.

(NOTE 2) After clicking on the icon, the window is displayed. However, do not close the window at this time.
To exit the simulation function, select the <Simulation> menu under the <Tool> (for each tool) in order to deselect it.

(NOTE 3) If the following message is displayed when simulation is executed, select <Allow access>.



5.12 NC(BUILT IN PLC) Monitoring

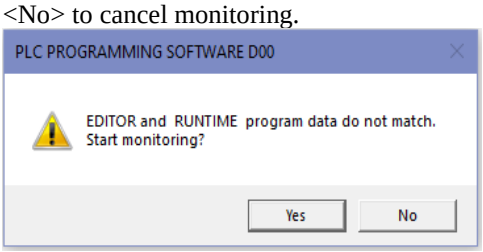
This section describes the user operations for monitoring the operating state of the NC(BUILT IN PLC).
Monitoring is a function that allows the user to confirm the current status of the OMs and variables and check the conditions that the POU is operating under.

When there are multiple programs displayed in the LD/ST Programming Editor, the selected LD/ST program is monitored.

- Simulation monitoring
  When there is a checkmark "✓" next to the <Simulation> in the <Tool> menu (for each tool), then the monitoring is carried out for the simulation.

(NOTE 1) Program editing, variable editing and program writing are not possible during monitoring.

(NOTE 2) Monitor the program that is written to the NC(BUILT IN PLC). The user can use the LD/ST Programming Editor or FBD Programming Editor to open a program that is not written to the NC(BUILT IN PLC) and switch to monitor mode. However, the displayed content and the actual operation will not match.
When an attempt is made to monitor POUs that are not written into the NC(BUILT IN PLC), the following confirmation message is displayed. Select <Yes> to start monitoring POUs that are different from the data in the NC(BUILT IN PLC). Select <No> to cancel monitoring.



(NOTE 3) The NC(BUILT IN PLC) carries out monitoring in 200 ms cycles.
If one scan time of a program run simultaneously does not exceed 200 ms, then the NC(BUILT IN PLC) monitors each program and waits for the 200 ms to elapse and then proceeds to the next monitor process. If one scan time of a program run simultaneously does exceed 200 ms, then the NC(BUILT IN PLC) monitors each program and proceeds to the next monitor process without waiting for a time cycle to elapse. In this situation, the monitoring cycle in the NC(BUILT IN PLC) exceeds 200 ms.

(NOTE 4) Refer to "11.4 Monitoring Function of Program Operation" for details on monitoring functions and restrictions.

Monitoring can be executed using the following operations.

Operation procedure

Perform one of the operations below in the LD/ST Programming Editor or the FBD Programming Editor.

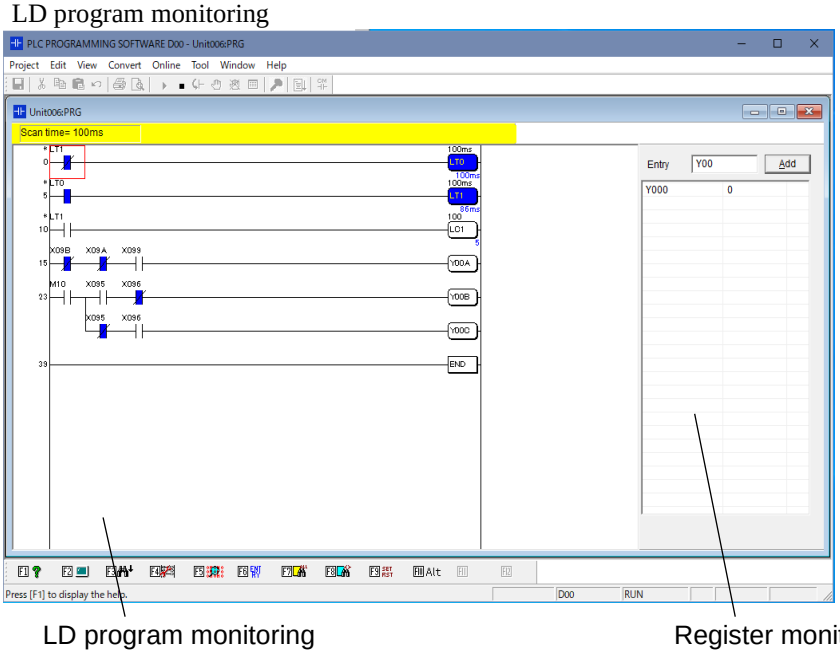
- Press **[F2]** key
- Select <Start monitoring> under the <Online> menu
- Click on from the function key tool bar

In an LD program or ST program, the user can change the monitoring operation from using the **[F2]** key to using the **[F3]** key in the option settings. Refer to "5.6.1.1 Edit screen adjustment (option settings)" for further details.

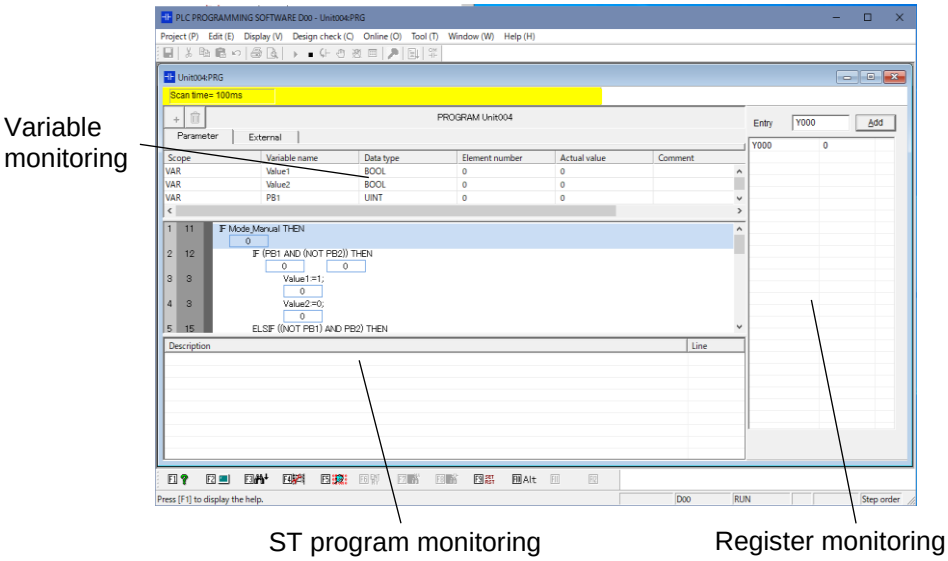
- Exit monitoring
  Press the **[F4]** key to exit monitoring.

- Stop / Re-start monitoring
  During monitoring, the user can press the **[Esc]** key in the LD/ST Programming Editor or the FBD Programming Editor to stop the monitoring (in the LD/ST Programming Editor or the FBD Programming Editor) without changing the operation state of the NC(BUILT IN PLC). In addition, the user can check the OM/variable values when the monitoring is stopped. To re-start the monitoring, press the **[F2]** key.

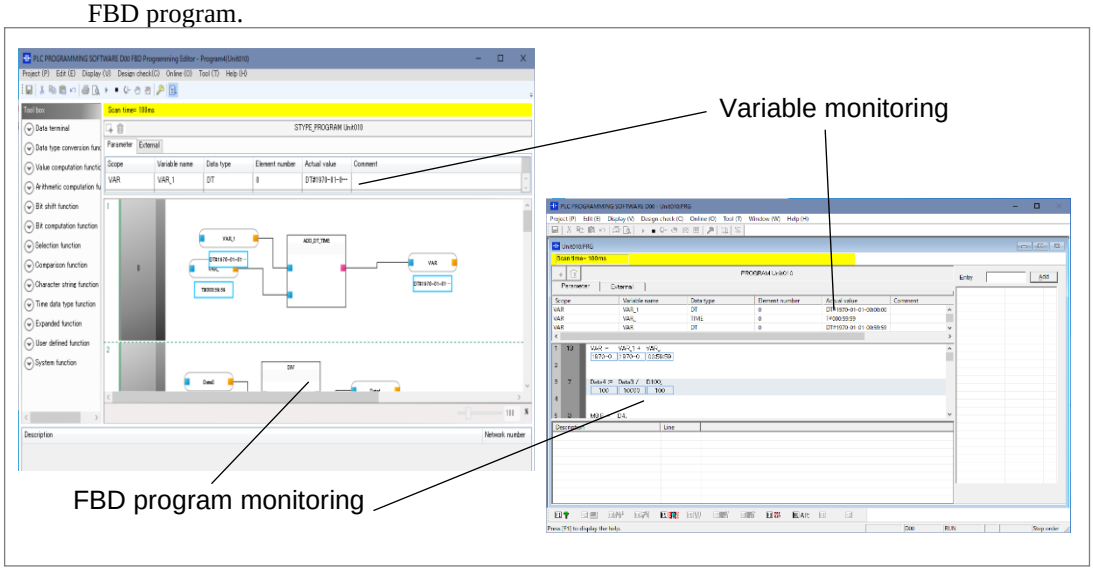
- LD program monitoring



- ST program monitoring



- FBD program monitoring
  The ST program that was created after conversion is also monitored at the same time as the FBD program.



Refer to the following for operation details about monitoring in each program.

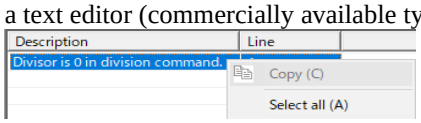
When using an LD program :"7.12 Monitoring operations"
When using an ST program :"8.11 Monitoring operations"
When using an FBD program :"9.11 Monitoring operations"

When monitoring in an ST/FBD program, if an error is triggered in the NC(BUILT IN PLC), then the error description is displayed in the message display area. (The figure below is taken when monitoring an ST program)

Description	Line
Divisor is 0 in division command.	1

Refer to "11.4 Monitoring Function of Program Operation" for further details on the display.

(NOTE) The user can copy the text data displayed in the message display area and paste it to a text editor (commercially available type).



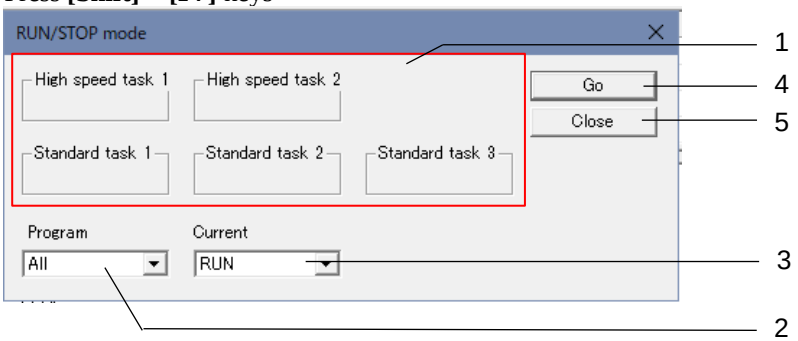
To copy data, select the desired line, and then right-click the mouse and select <Copy> from the menu, or press the **[Ctrl]** + **[C]** keys.
In addition, the user can also select all the lines in the message display area by choosing <Select all> from the menu or by pressing the **[Ctrl]** + **[A]** keys.

5.13 Run & Stop in NC(BUILT IN PLC)

This section describes how to change to RUN mode and to STOP mode in the NC(BUILT IN PLC).

Operation procedure

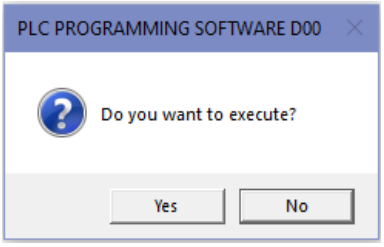
1. Connect the PLC Programming software D00 EDITOR to the NC(BUILT IN PLC).
2. Perform one of the operations below in the LD/ST Programming Editor or the FBD Programming Editor to display the <RUN/STOP mode> dialog.
   - Select <RUN/STOP mode> under the <Online> menu
   - Press **[Shift]** + **[F7]** keys



No.	Item	Description
1	Current status	Displays the current state of high speed task 1, high speed task 2, standard task 1, standard task 2 and standard task 3.
2	Program selection	Selects the program for changing the mode. The following selections are available: <All>, <High speed task 1>, <High speed task 2>, <Standard task 1>, <Standard task 2> and <Standard task 3>. A program that is not set in the program execution settings does not appear in the list.
3	Operation selection	Selects the desired mode.
4	<Execute> button	Changes to the selected mode.
5	<Close> button	Closes the dialog.

3. Select the desired item and operation state, and then click on the **[Execute]** button.

4. A confirmation appears asking whether to execute the RUN/STOP mode change.
   Select <Yes> to execute the RUN/STOP mode change.

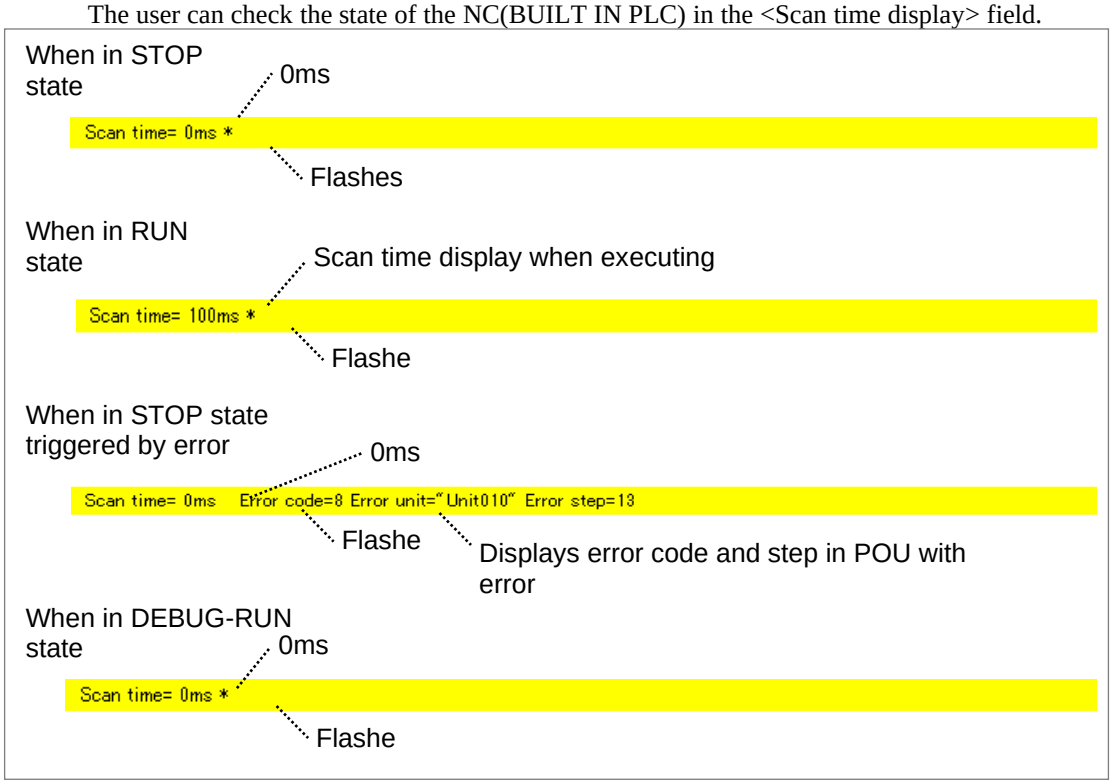


5. After the mode change is complete, the new state is reflected in the <Current status> in the <RUN/STOP mode> dialog.

6. Click on the <Close> button to close the <RUN/STOP mode> dialog.

- RUN/STOP mode change in simulation
  When there is a checkmark "✓" next to the <Simulation> in the <Tool> menu (for each tool), then the program is written to the simulation.

- Mode display for NC(BUILT IN PLC)
  The user can check the state of the NC(BUILT IN PLC) in the <Scan time display> field.



The RUN/STOP mode change is performed in the LD/ST Programming Editor or the FBD Programming Editor.

(NOTE) In the LD/ST Programming Editor, the state of the NC(BUILT IN PLC) is displayed on the status bar. (Each state is displayed as follows: RUN, STOP and DEBUG-RUN.)

RUN	STOP	DEBUG-RUN

5.14 POU Debug Operation

This section describes the user operations for debugging a program to run in the NC(BUILT IN PLC).
Refer to “11.5 Debug Function” for details on the debug function.

In each program, this function allows the user to check the operation of a program.

- When using an LD program
  Step execution
  Executes each command in order.
  Stop condition settings and execution
  The user can stop the LD program at a step that is set or when an OM changes to a specified status. The user can also execute a step from that position.

- When using an ST program
  Step execution
  Executes each command in order.
  Setting and execution of stop position based on breakpoint setting
  The user can stop an ST program at the line that is set. The user can also execute a step from that position.

- When using an FBD program
  Step execution
  Executes each command in order.
  Setting and execution of stop position based on breakpoint setting
  The user can stop an FBD program at the network that is set. The user can also execute a step from that position.

In addition, during monitoring, the user can press the **[Esc]** key in the LD/ST Programming Editor or the FBD Programming Editor to stop the monitoring (in the LD/ST Programming Editor or the FBD Programming Editor) without changing the operation state of the NC(BUILT IN PLC). The user can also check the OM/variable values when the monitoring is stopped.
To re-start the monitoring, press the **[F2]** key.

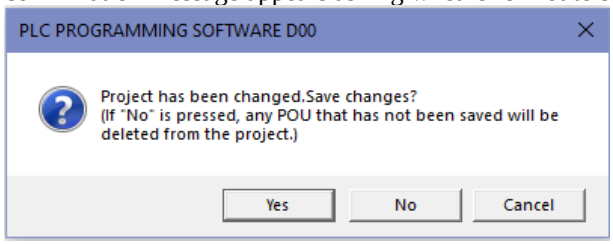
Refer to the following for operation details about debugging an LD/ST/FBD program.
When using an LD program: “7.13 DEBUG function”
When using an ST program: “8.12 DEBUG function”
When using an FBD program: “9.12 DEBUG function”

5.15 Close Project

This section describes the operations for closing a project that is being edited.

Operation procedure
Select <Close project> from the <Project> menu in the PLC Programming software D00.

(NOTE) When a project being edited is not saved and the <Close project> menu is selected, a confirmation message appears asking whether or not to save the project.



When <Yes> is selected, the project being edited is saved and the project closes.
When <No> is selected, the project being edited is not saved and the project closes.
When <Cancel> is selected, the project is not saved, and the project will also not close.

(This page was intentionally left blank.)

CHAPTER 6

BASIC OPERATION PROCEDURE FOR PROJECTS

6.1 Create, Open & Save Projects
6.2 Create & Edit POUs
6.3 Program Execution Settings
6.4 Register Global Variables
6.5 Register DFC
6.6 Parameter Settings
6.7 Double Coil Check
6.8 Conversion
6.9 Communication Settings
6.10 Write Programs
6.11 Read Programs
6.12 Verify Projects
6.13 Launch Simulation
6.14 Option Settings
6.15 Display Manual
6.16 Version Information
6.17 B00/C00 Project Conversion
6.18 Import & Export
6.19 Power Failure Hold Area / Writing Range Check Screen
6.20 Check for Writing to OMs Outside Writing Range

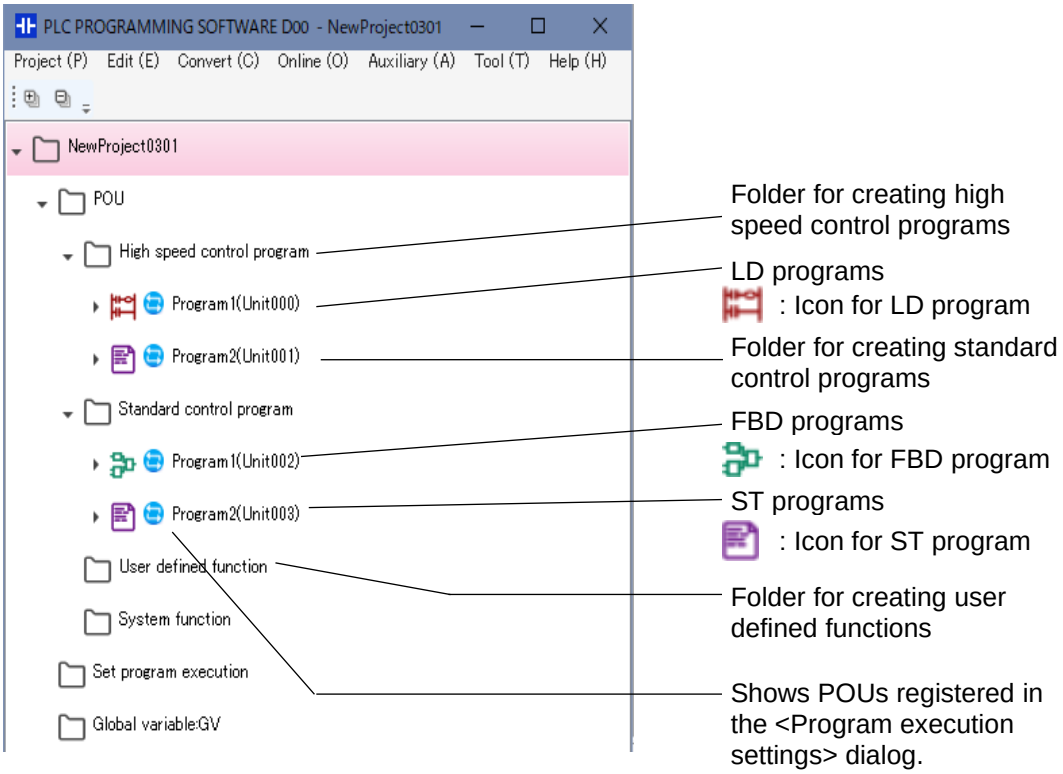
6.1 Create, Open & Save Projects

Refer to the following for details on how to create new projects, save them to media devices and read the saved projects from media devices.

- Create projects "5.1.1 Create new projects"
- Save projects "5.1.4 Save as" and "5.1.5 Save all"
- Open projects "5.1.2 Open projects"
- Close projects "5.15 Close projects"

6.2 Create & Edit POUs

This section describes operations for creating POUs and editing (delete, copy and paste) them.
The user can create a maximum of 10 high speed and 10 standard control programs in any of the program languages (LD/ST/FBD). A maximum of 924 user defined functions can be created.
POUs are created in the POU folder for the PLC Programming software D00.
POUs can be deleted, copied and pasted.



6.2.1 Create New POU

Refer to "5.2 Create POUs (LD/ST/FBD programs)" for details on creating new POUs.

6.2.2 Edit POUs

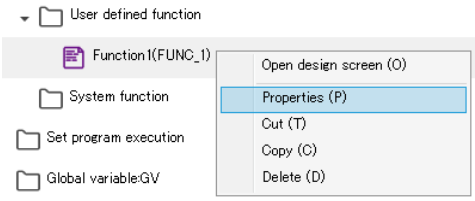
Refer to the following chapters for details about POU edit operations.
"Chapter 7 LD Edit operation procedure"
"Chapter 8 ST Edit operation procedure"
"Chapter 9 FBD Edit operation procedure"

6.2.3 Change POU Properties

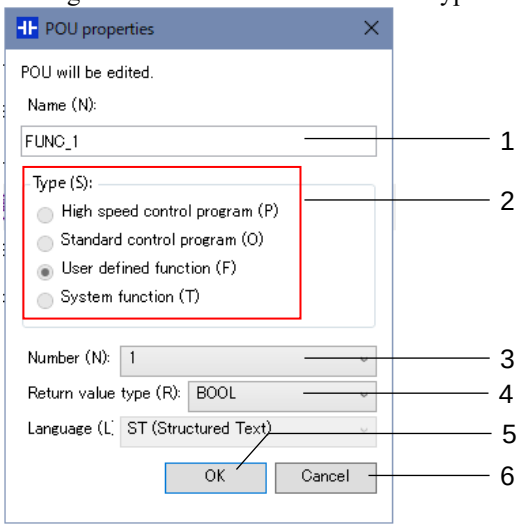
The user can change the name of the POU that is created and the return value.
The following shows the change procedure.

Operation procedure

1. Right-click the mouse on the POU name that was created, and then select <Properties> from the menu.



2. Change the "Name" and "Return value type" in the <POU properties> dialog that is displayed.



No.	Item	Description
1	Name	Sets the POU name. Set the name using no more than 16 characters. Single-byte alphanumeric characters and a half-width underscore (_) can be used for the name. However, numbers cannot be used as the first character.
2	Type	"Disable" is normally displayed. Displays the type that is set when it is registered.
3	Number	Sets the numbers. The numbers that are currently being used and those that are available are displayed as selections in the drop-down list. 1 through 10 can be selected for high speed and standard control programs. 1 through 924 can be selected for user defined functions.
4	Return value type	Sets the return value data type when changing the properties of a user defined or system defined function. The following types can be set. BOOL, BYTE, WORD, DWORD, INT, DINT, LINT, UINT, UDINT, ULINT, REAL, LREAL, DATE, TIME, TOD, DT and STRING
5	<OK> button	Applies the setting changes, and the dialog closes.
6	<Cancel> button	Discards the setting changes, and the dialog closes.

3. Click on the <OK> button.

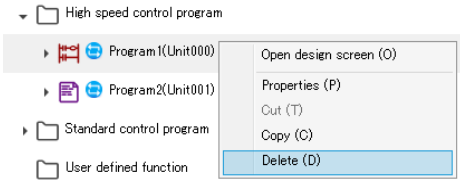
(NOTE 1) If a user or system defined function name or a return value is already being used by another POU and then it is changed, that change is not automatically reflected in the (call) source program. In this situation, correct the program manually.
(NOTE 2) When a POU is opened in a software editor or when the tools are in monitoring mode, the <Properties> menu cannot be selected.
(NOTE 3) Refer to “12.1.6.1 List of data types” in the PLC System Manual or “12.1.2.1 List of data types” for details on the data types.
(NOTE 4) The POU “type” and “language” cannot be changed.

6.2.4 Delete POUs

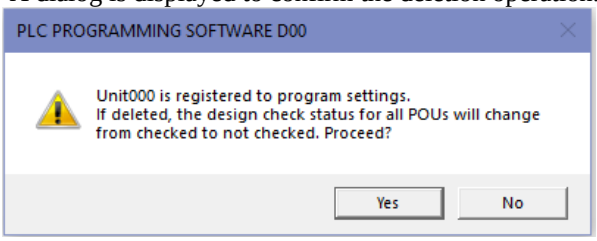
The POUs that are created can be deleted.
The following shows the POU deletion procedure.

Operation procedure

1. One of the following operations can be carried out:
   - Right-click the mouse on the POU name that was created, and then select <Delete> from the menu.
   - Select the desired POU, and then press the **[Delete]** key.
   - Select the desired POU, and then select <Delete> from the <Edit> menu.



2. A dialog is displayed to confirm the deletion operation.



3. Click on the <Yes> button to execute the deletion.

(NOTE 1) The deleted POU cannot be restored. Therefore, be careful when performing a deletion.
(NOTE 2) When a user defined or system defined function used by another POU is deleted, all of the POUs reset to an unconverted state.
(NOTE 3) When a POU is opened in a software editor or when the tools are in monitoring mode, deletion is not possible.
First, close the POU to be deleted, or exit monitoring mode, and then execute the deletion.

6.2.5 Use POUs (Copy, Cut & Paste)

The user can copy or cut POUs that have been created to re-use them.
However, a POU that has not been saved cannot be copied or cut. In addition, user and system defined function POUs can only be cut.
POUs that are copied or cut can be pasted to a folder that is the same type.
POUs that are pasted are set with the same name as the copy or cut source. Change the POU name after pasting it.
However, if the same name already exists, an underscore (_) is added to the end of the POU name after it is pasted.

Ex: Copy source :Program0_Name
When pasted :Program0_Name_ (An underscore (_) is added at the end.)

If the same name already exists, then a number is added after the underscore (_). The next available number starting from 1 is assigned and added.

Ex: If “Program0_Name_” already exists
Copy source :Program0_Name
When pasted :Program0_Name_1

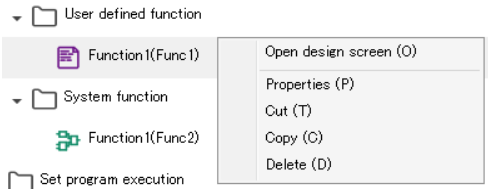
If the name is equal to or greater than 17 characters after being pasted, then the last letter is deleted and an underscore (_) is added.

Ex: Copy source :Program0000_Name
When pasted :Program0000_Nam_

The following shows the copy, cut and paste procedure.

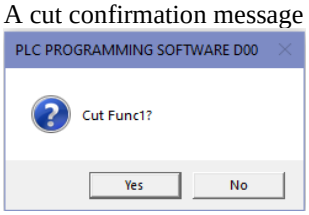
Operation procedure

1. Perform one of the following operations to copy a POU.
   - Right-click the mouse on the POU name that was created, and then select <Copy> from the menu.
   - Select the desired POU, and then press the **[Ctrl]** + **[C]** keys.
   - Select the desired POU, and then select <Copy> from the <Edit> menu.



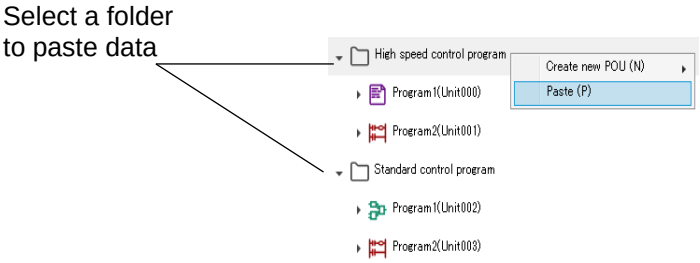
Perform one of the following operations to cut a POU (user/system defined function).
- Right-click the mouse on the POU name that was created, and then select <Cut> from the menu.
- Select the desired POU, and then press the **[Ctrl]** + **[X]** keys.
- Select the desired POU, and then select <Cut> from the <Edit> menu.

A cut confirmation message is displayed before it is carried out.



Click on the <Yes> button to perform the cut operation.

2. Select one of the following folders: <High speed control program>, <Standard control program>, <User defined function> or <System defined function>. Then, perform one of the operations below.
   - Select <Paste> from the <Edit> menu.
   - Press the **[Ctrl]** + **[V]** keys.
   - Right-click on the <POU> folder, and then select <Paste> from the displayed menu.

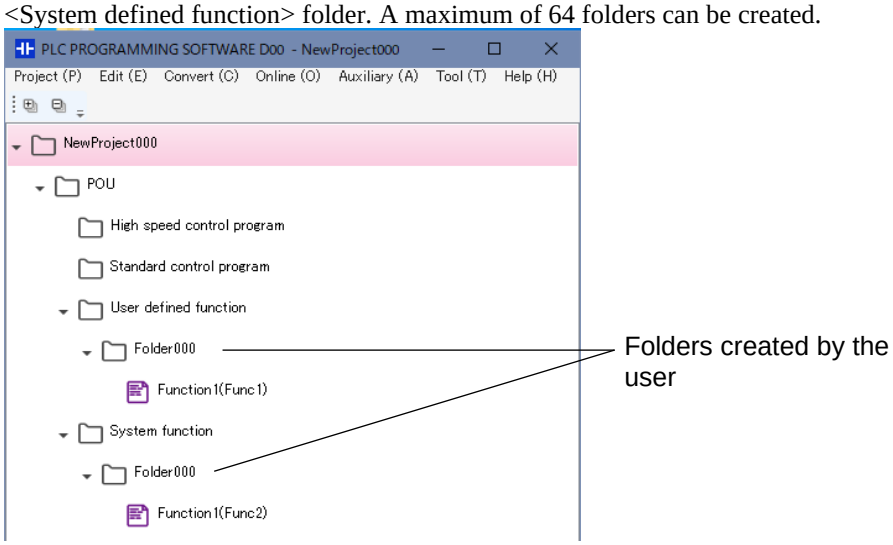


3. When necessary, change the name of the POU after pasting it.
   Refer to "6.2.3 Change POU properties" for details on how to rename it.

(NOTE 1) POUs that are copied cannot be pasted to a folder for a different type of POU.
(NOTE 2) After pasting POUs, save the project.
(NOTE 3) When a user or system defined function used by another POU is cut, all of the POUs reset to an unconverted state.
(NOTE 4) When a POU is opened in a software editor or when the tools are in monitoring mode, copy, cut and paste operations are not possible.
First, close the POU to be copied/pasted, or exit monitoring mode, and then execute the operation.

6.2.6 Create Folders

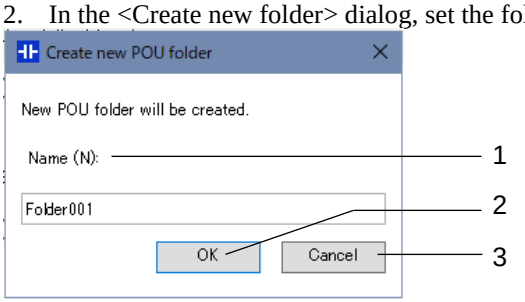
In the PLC Programming software D00, the user can create folders under the <User defined function> folder and <System defined function> folder in order to make it easier to manage the user and system defined function POUs.
These subfolders can be created only one level under the <User defined function> folder or <System defined function> folder. A maximum of 64 folders can be created.



Perform one of the following operations to create a folder.

Operation procedure

1. Select the <User defined function> folder or the <System defined function> folder, and then perform one of the operations below.
   - Right-click with the mouse, and select <New> and <Folder> from the menu.
   - Select <Create folder> from the <Edit> menu.
2. In the <Create new folder> dialog, set the folder name.



No.	Item	Description
1	Name	Sets the folder name. Set the name using no more than 16 characters. The following symbols cannot be used: ¥, /, :, *, ?, ", <, > and | The same name of another folder cannot be specified. At this time, the uppercase and lowercase letters are not differentiated.
2	<OK> button	Applies the setting, and the dialog closes.
3	<Cancel> button	Discards the settings, and the dialog closes.

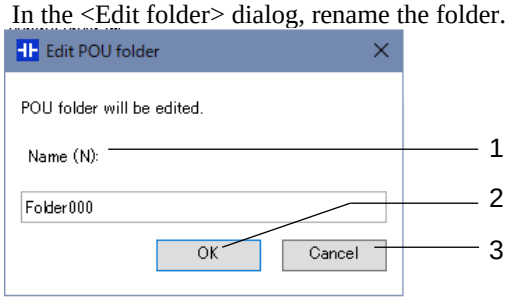
3. Click on the <OK> button.

6.2.7 Rename Folders

The names of folders created by the user can be changed.
Perform one of the following operations to name a folder.

Operation procedure

1. Select the desired folder, right-click the mouse, and then select <Edit> from the menu.
2. In the <Edit folder> dialog, rename the folder.



No.	Item	Description
1	Name	Sets the folder name. Set the name using no more than 16 characters. The following symbols cannot be used: ¥, /, :, *, ?, ", <, > and | The same name of another folder cannot be specified. At this time, the uppercase and lowercase letters are not differentiated.
2	<OK> button	Applies the setting, and the dialog closes.
3	<Cancel> button	Discards the settings, and the dialog closes.

3. Click on the <OK> button.

6.2.8 Delete Folders

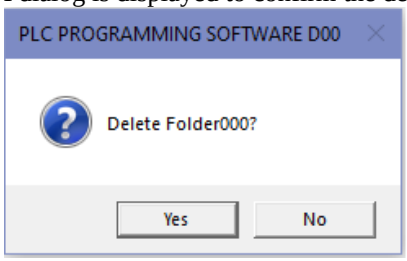
The folders created by the user can be deleted.
When a folder is deleted, all of the POUs included in that folder are deleted.

(NOTE) The deleted folder and POU(s) cannot be restored. Therefore, be careful when performing a deletion.

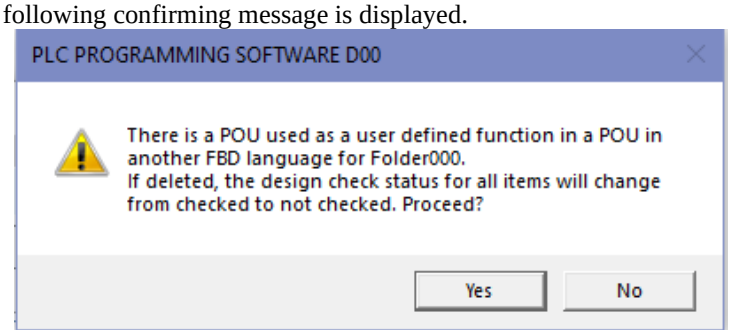
Perform one of the following operations to delete a folder.

Operation procedure

1. Select the desired folder and perform one of the following operations.
   - Right-click with the mouse, and select <Delete> from the menu.
   - Select <Delete folder> from the <Edit> menu.
   - Press the **[Del]** key.
2. A dialog is displayed to confirm the deletion operation.



In addition, when a folder is deleted and it contains POUs being used in a FBD program, the following confirming message is displayed.



3. Click on the <Yes> button to execute the deletion.
4. The folder is deleted.

(NOTE) When the POU saved in that folder is opened in a software editor or when the tools are in monitoring mode, deletion is not possible.
First, close the POU to be deleted, or exit monitoring mode, and then execute the deletion.

6.3 Program Execution Settings

The program execution settings are a function for setting POUs and their execution order in the NC(BUILT IN PLC).
Refer to “5.8 Program execution settings” for details on the settings.

6.4 Register Global Variables

The global variables can be used throughout the entire project regardless of the POUs.
Refer to “5.3 Register global variables” for details on the registration settings.

6.5 Register DFC

The description in this chapter is omitted because this product is not equipped with this function.

6.6 Parameter Settings

When executing a program in the NC(BUILT IN PLC), the parameter settings are a function for setting the scan time, the WDT monitoring time, the operation method when an error occurs, the output method when the mode is changed from STOP to RUN and the power failure hold area setting.
Refer to “5.4 Parameter settings” for details on the settings.

6.7 Double Coil Check

Double coil check is a function that checks programs that run in the NC(BUILT IN PLC) in order to detect if the same OM has been written from multiple places.
Refer to “5.7.1 Double coil check” for details on the settings.

6.8 Conversion

Conversion refers to converting the POUs (created in a language in the PLC Programming software D00 EDITOR) to code (intermediate code) that allows execution in the NC(BUILT IN PLC).
Refer to “5.9.1 Conversion” for details on the settings.

In addition, after conversion is executed in the PLC Programming software D00 EDITOR, the file can be automatically saved and the write during run operation can be carried out. Refer to “5.9.2 Conversion mode settings” for details on the settings. Refer to “5.10.4 Write during run” for details on the write during run operation.

6.9 Communication Settings

The communication settings are a function that set the communication ports for communicating between the PLC Programming software D00 EDITOR and the NC(BUILT IN PLC).
Refer to the “5.10.1 Communication settings with PLC programming software D00 Linux EDITOR” for details on the settings.

6.10 Write Programs

The program write operation refers to writing programs that can run in the NC(BUILT IN PLC) and in the simulation.
Refer to “5.10.2 Write programs” and “5.10.4 Write during run” for details about the write settings.

6.11 Read Programs

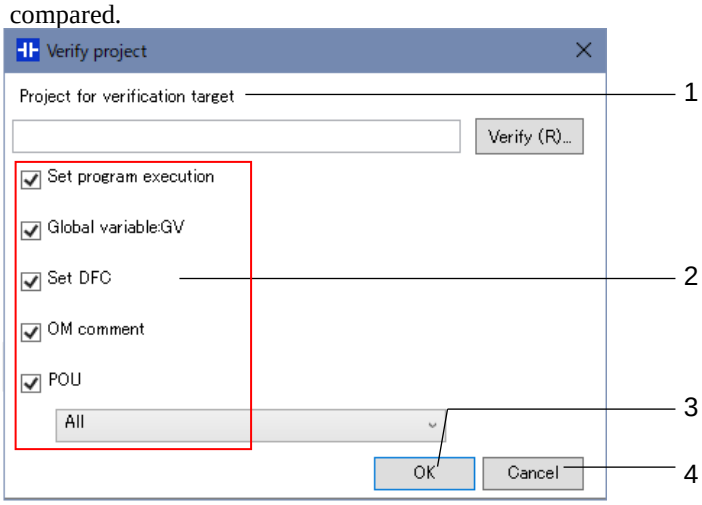
The program read operation is a function that allows the PLC Programming software D00 EDITOR to read all the files saved to PLC Programming software D00 Linux EDITOR.
Refer to “5.10.3 Read programs” for details on the user operations.

6.12 Verify Projects

The user can compare two projects and check whether the content matches or not.
The project that is currently opened in the PLC Programming software D00 EDITOR can be compared with another project that is saved.
Before verifying projects, both projects (verification source and target) need to be converted and saved.

Operation procedure

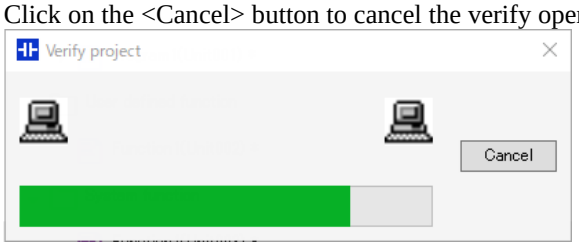
1. Open the project (verification) source in the PLC Programming software D00 EDITOR.
2. Select <Verify project> from the <Project> menu in the PLC Programming software D00.
3. In the <Verify project> dialog, select the other project for the comparison and the items to be compared.



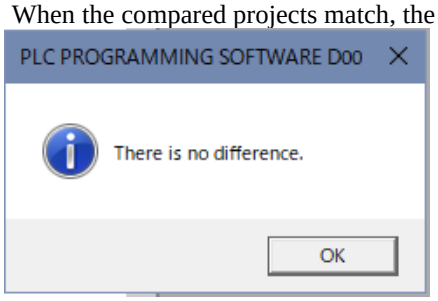
No.	Item	Description
1	Project for verification target	Specifies the project to be verified. The user can select a project by clicking on the <Browse> button and displaying the <Open file> dialog.
2	Compare items	The user can compare the following data between the two projects (source and target). Tick the checkbox for the items to compare. • Program execution settings • Global variables • OM comments • POU When comparing all of the POUs, select <All> from the drop-down list. When comparing the POUs individually, select the POU name from the drop-down list. If the “POU” checkbox is not ticked, the program selection combo box is disabled.
3	<OK> button	Verifies the project.
4	<Cancel> button	Cancels the verify operation.

4. Click on the <OK> button.

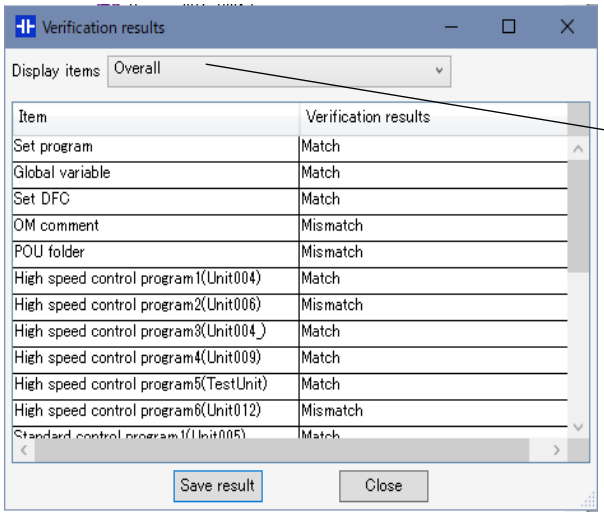
5. The <Verify project> dialog is displayed to indicate the progress.
Click on the <Cancel> button to cancel the verify operation.



6. When the project verification is complete, the results are displayed.
When the compared projects match, the following message is displayed.

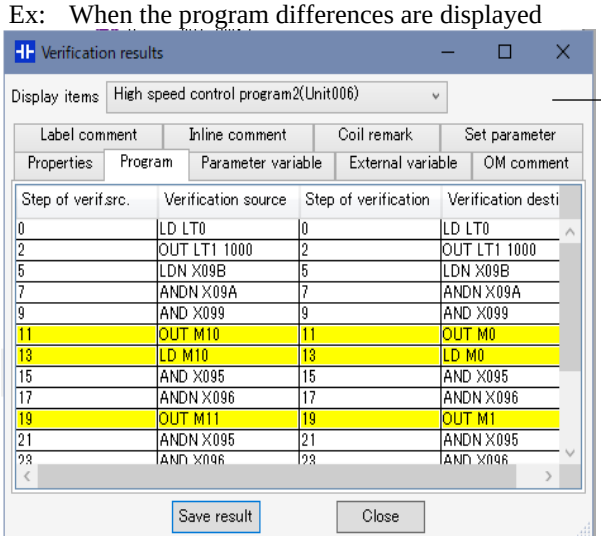


When the compared projects do not match, the results (match/mismatch) are shown for each item that is compared.
The user can confirm the details of each item selected in the <Display items>.



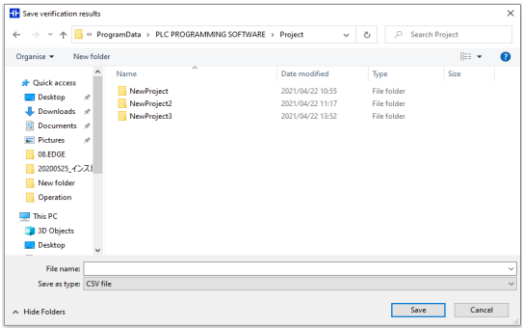
Displays the details of the differences for the selected items.

Ex: When the program differences are displayed



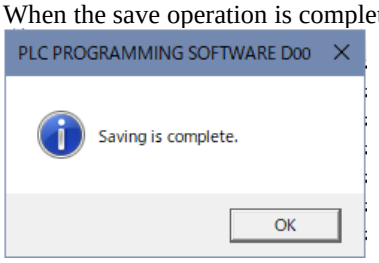
The displayed content varies depending on the selected items.

Click on the <Save result> button to save the verification results as a CSV file (tab delimited).



Specify no more than 16 characters for the name when saving as a CSV file.

When the save operation is complete, the following message is displayed.



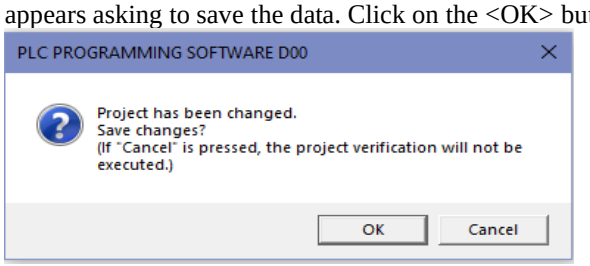
When comparing with a project that runs in the NC(BUILT IN PLC), the project cannot be read directly from the NC(BUILT IN PLC). Make sure that the project saved in the PLC Programming software D00 Linux EDITOR is the same as the project that runs in the NC(BUILT IN PLC). Then, access <Read program> to read the project that is saved in the PLC Programming software D00 Linux EDITOR and then verify the project. Refer to “5.10.3 Read programs” for details on the program read operation.

(NOTE 1) This function is not for comparing projects that run in the NC(BUILT IN PLC).

(NOTE 2) When comparing POUs, the projects may not be compared properly if a conversion error is included in a POU.

(NOTE 3) Even if a checksum error is detected in a project, the project can be verified but it may not be compared properly.

(NOTE 4) If a project that is currently opened in the PLC Programming software D00 EDITOR is not saved and an attempt is made to verify that project, a confirmation message appears asking to save the data. Click on the <OK> button to verify the project.



(NOTE 5) The details about the differences cannot be checked when a POU that is missing in the verification source or target.

6.13 Launch Simulation

The simulation is a function for checking the operation in the PLC Programming software D00 EDITOR before running the program on the actual machine.
The following is the procedure for launching the simulation and running project data in the simulation.

Operation procedure

1. When there is no checkmark "✓" next to <Simulation> in the <Tool> menu of the PLC Programming software D00, select <Simulation>.
   After the simulation is launched, a checkmark "✓" appears before the item name <Simulation>.
2. Select <Write program> from the <Online> menu.
   Refer to the "5.10.2 Write programs" for details about writing programs.

- Exit simulation
  When there is a checkmark "✓" next to <Simulation> in the <Tool> menu of the PLC Programming software D00, select <Simulation> again to deselect it.
  After exiting the simulation, the checkmark "✓" next to the <Simulation> disappears.

(NOTE 1) The LD/ST Programming Editor and the FBD Programming Editor can both be used to launch and exit simulation.
(NOTE 2) Projects that execute in the simulation function are projects that are currently opened in the PLC Programming software D00 EDITOR.
(NOTE 3) In the option settings, the user can change the count up method on the timer when the simulation executes.
Refer to "5.6.1.1 Edit screen adjustment (option settings)", "5.6.2.1 Edit screen adjustment (option settings)" or "5.6.3.1 Edit screen adjustment (option settings)" for details on configuring the option settings.

6.14 Option Settings

The option settings allow the user to configure the display and operation settings in the LD/ST Programming Editor and FBD Programming Editor, such as changing the font used in the program display or changing the functions assigned to the function keys.
Refer to "5.6.1.1 Edit screen adjustment (option settings)", "5.6.2.1 Edit screen adjustment (option settings)" or "5.6.3.1 Edit screen adjustment (option settings)" for details on the settings and operations.

6.15 Display Manual

This section describes how to display a manual.
The user can display the different manuals in the PLC Programming software D00.

To display the manuals, the user must install an application software beforehand that can display PDF files, such as Acrobat Reader provided by Adobe.

- User's manual
  Operation procedure
  Select the <User's manual> under <Manual> in the <Help> menu.

- Operation manual
  Operation procedure
  Select the <Operation manual> under <Manual> in the <Help> menu.

- Programming manual
  Operation procedure
  Select the <Programming manual> under <Manual> in the <Help> menu.

6.16 Version Information

This section describes the version information display in the PLC Programming software D00 EDITOR.

Operation procedure

Select <Version display> from the <Help> menu.

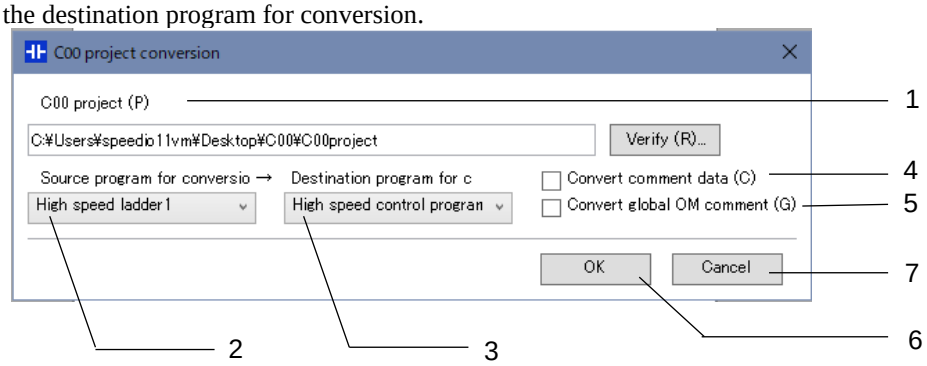
6.17 B00/C00 Project Conversion

The format of a B00 or C00 project can be converted to D00 and added to an open D00 project.

(NOTE) Refer to "5.6.1.1 Edit screen adjustment (option settings)" for details on how to disable the B00/C00 project conversion.

Operation procedure

1. Select <B00/C00 project conversion> under the <Tool> menu.
2. In the <B00/C00 project conversion> dialog, set the B00/C00 project, the source program and the destination program for conversion.



No.	Item	Description
1	B00/C00 project	Specifies the B00 or C00 project. After pressing the <Browse> button to display the <Open project> dialog, the user can select a project.
2	Source program for conversion	Displays the B00 project name when a B00 project is specified in (1). When a C00 project is specified in (1), the user can select a program to convert from the drop-down list. When there is no project, nothing is displayed in the drop-down list.
3	Destination program for conversion	Selects an available program number (from the drop-down list) that is not used in the project. When there is no available number, nothing is displayed in the drop-down list.
4	Convert comment data	Selects the comment data to import.
5	Convert global OM comment	Selects the global OM comment data to import.
6	<OK> button	Applies the setting, and the dialog closes.
7	<Cancel> button	Discards the settings, and the dialog closes.

3. After clicking on the <OK> button, the project data is converted.
4. After the project data is converted, the programs (selected as the destination programs for conversion) are added as unconverted programs. Therefore, convert the program data.

(NOTE) In the PLC Programming software D00, the maximum number of steps in one POU varies.
- High speed : 5K steps
- Standard : 40K steps

When registering POUs, be sure that the maximum number of steps is not exceeded in the PLC Programming software D00.

6.18 Import & Export

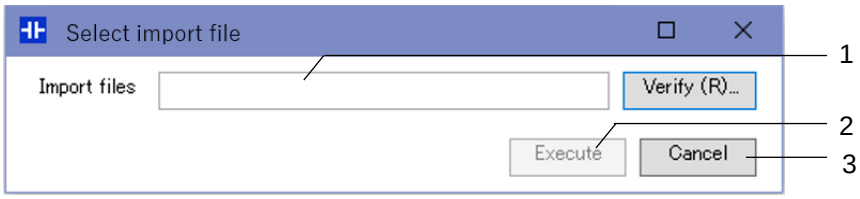
This section describes the import and export function for projects in the PLC Programming software D00 EDITOR.

6.18.1 Import Function

Perform the following procedure to import a project.

Operation procedure

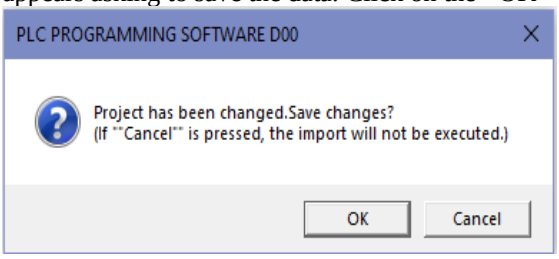
1. Select <Import> from the <Project> menu.
   When a project is not open, <Import> cannot be selected.
2. In the <Select import file> dialog, select an import file (bpp file).



No.	Item	Description
1	Import file	Sets the import file. The user can select an import file by clicking on the <Browse> button and displaying the <Open file> dialog.
2	<Execute> button	Applies the setting, and the dialog closes.
3	<Cancel> button	Discards the settings, and the dialog closes.

3. In the <Select import file> dialog, click on the <Execute> button.
4. When the import is complete, a message appears indicating that it is complete.

(NOTE) If a project that is currently opened in the PLC Programming software D00 EDITOR is not saved and an attempt is made to import that project, a confirmation message appears asking to save the data. Click on the <OK> button to import.

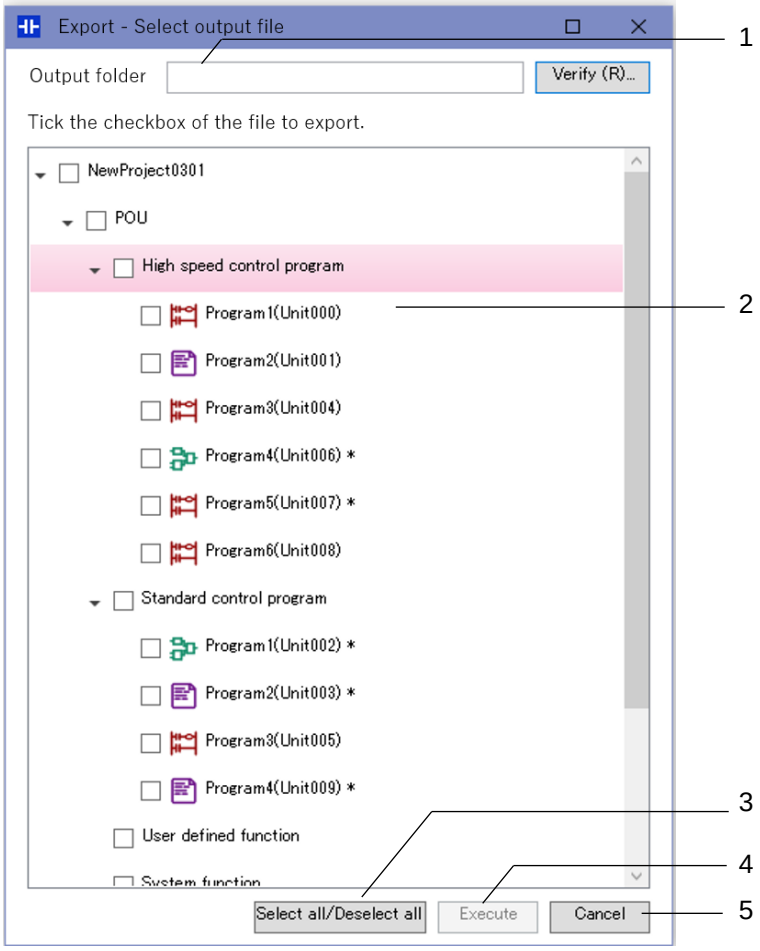


6.18.2 Export Function

Perform the following procedure to export a project.

Operation procedure

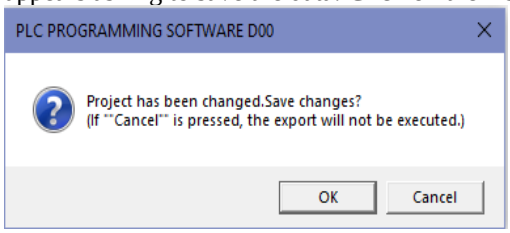
1. Select <Export> from the <Project> menu.
   When a project is not open, <Export> cannot be selected.
2. In the <Export output file selection> dialog, select the output folder and the export file.



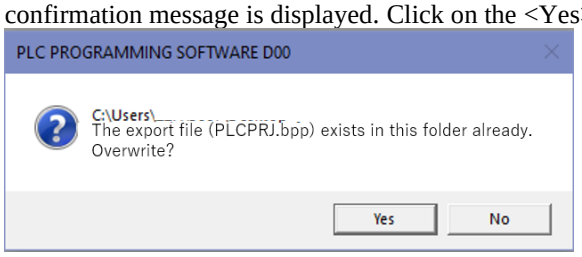
No.	Item	Description
1	Output folder	Sets the output folder. The user can select the output folder by clicking on the <Browse> button and displaying the <Folder selection> dialog.
2	Export file	Tick the checkbox for the file(s) to export. The following data is displayed only for files where a project is opened. • High speed control programs • Standard control programs • User defined functions • Program execution settings • Global variables • Global OM comments
3	Select all / Deselect all	Selects or deselects all the files.
4	<Execute> button	Applies the setting, and the dialog closes.
5	<Cancel> button	Discards the settings, and the dialog closes.

3. In the <Export output file selection> dialog, click on the <Execute> button.
4. When the <Export output file selection> dialog is closes, the bpp formatted file is exported.
   The export file name is “PLCPRJ.bpp”.

(NOTE 1) If a project that is currently opened in the PLC Programming software D00 EDITOR is not saved and an attempt is made to export that project, a confirmation message appears asking to save the data. Click on the <OK> button to export.



(NOTE 2) When there is already a file with the same name in the output folder, an overwrite confirmation message is displayed. Click on the <Yes> button to overwrite the data.



6

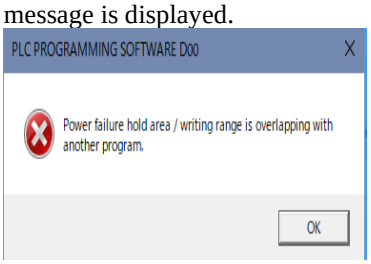
6.19 Power Failure Hold Area / Writing Range Check Screen

This section describes the function for checking the power failure hold area / writing range when writing a control program. Refer to “5.4 Parameter settings” for details on the power failure hold area / writing range settings.

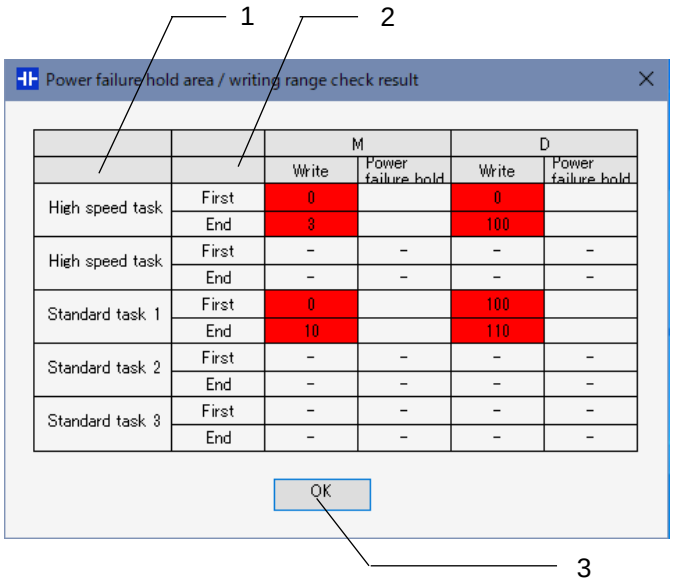
Perform the following procedure if there are check errors that are detected when writing a program.

Operation procedure

1. If there are check errors that are detected when writing a program, an error confirmation message is displayed.

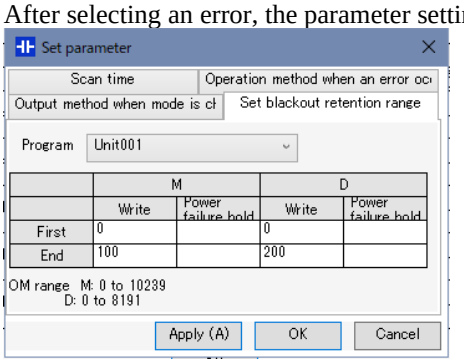


2. After clicking on the <OK> button, the <Power failure hold area / writing range check result> dialog is displayed.



No.	Item	Description
1	Control program	Displays the control programs.
2	Check range and error location	Displays the range that is set in the parameter settings. The duplicate error locations appear in red.
3	<OK> button	Closes the dialog.

3. After selecting an error, the parameter setting screen is displayed for that program.



4. Correct the setting so there is no duplicate setting, and click on the <OK> button.
5. In the <Power failure hold area / writing range check result> dialog, click on the <OK> button.
6. Write the program again.

(NOTE) While the <Power failure hold area / writing range check result> dialog is displayed, if a parameter setting is corrected, it does not update the dialog display.

6

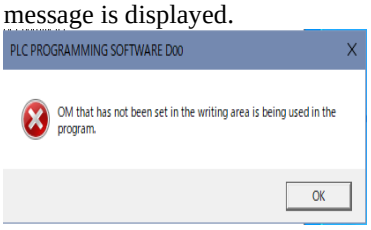
6.20 Check for Writing to OMs Outside Writing Range

This section describes the function to check for writing to OMs outside the writing range when writing a control program. Refer to “5.4 Parameter settings” for details on the writing range settings.

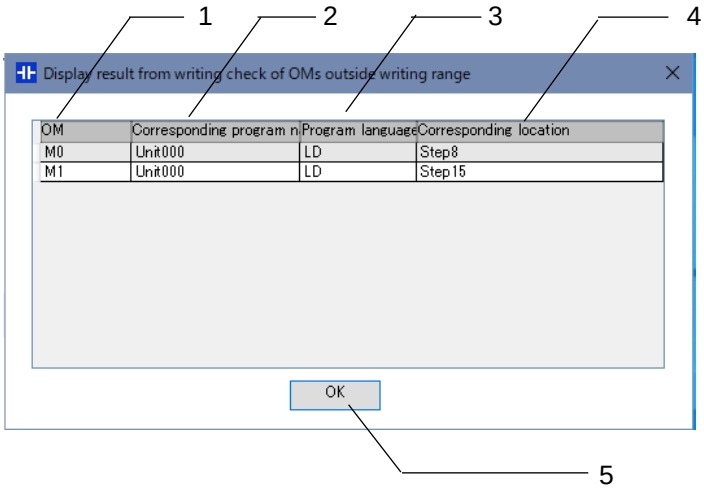
Perform the following procedure if there are check errors that are detected when writing a program.

Operation procedure

1. If there are check errors that are detected when writing a program, an error confirmation message is displayed.



2. After clicking on the <OK> button, the <Display result from writing check of OMs outside writing range> dialog is displayed.
   In the <Display result from writing check of OMs outside writing range> dialog, the maximum number of results that can be displayed is 1023. If there are 1024 results or more, then any data on those results (that exceed the maximum) is not displayed.



No.	Item	Description
1	OM name	Displays OMs outside the writing range that are written.
2	Program name	Displays the name of the program where the OMs in 1 are used.
3	Program language	Displays the language of the program in 2.
4	Check range and error location	Displays the locations where the OMs in 1 are being used. The location format is as follows. LD program: Corresponding step number When detected inside a CALC BOX in an LD program: Corresponding step number (“CALC BOX number” - “ Line number”) ST program: Corresponding line number FBD program: Corresponding network number
5	<OK> button	Closes the dialog.

3. After clicking on the <OK> button, the dialog closes.
4. Correct the program and then write the program again.

CHAPTER 7

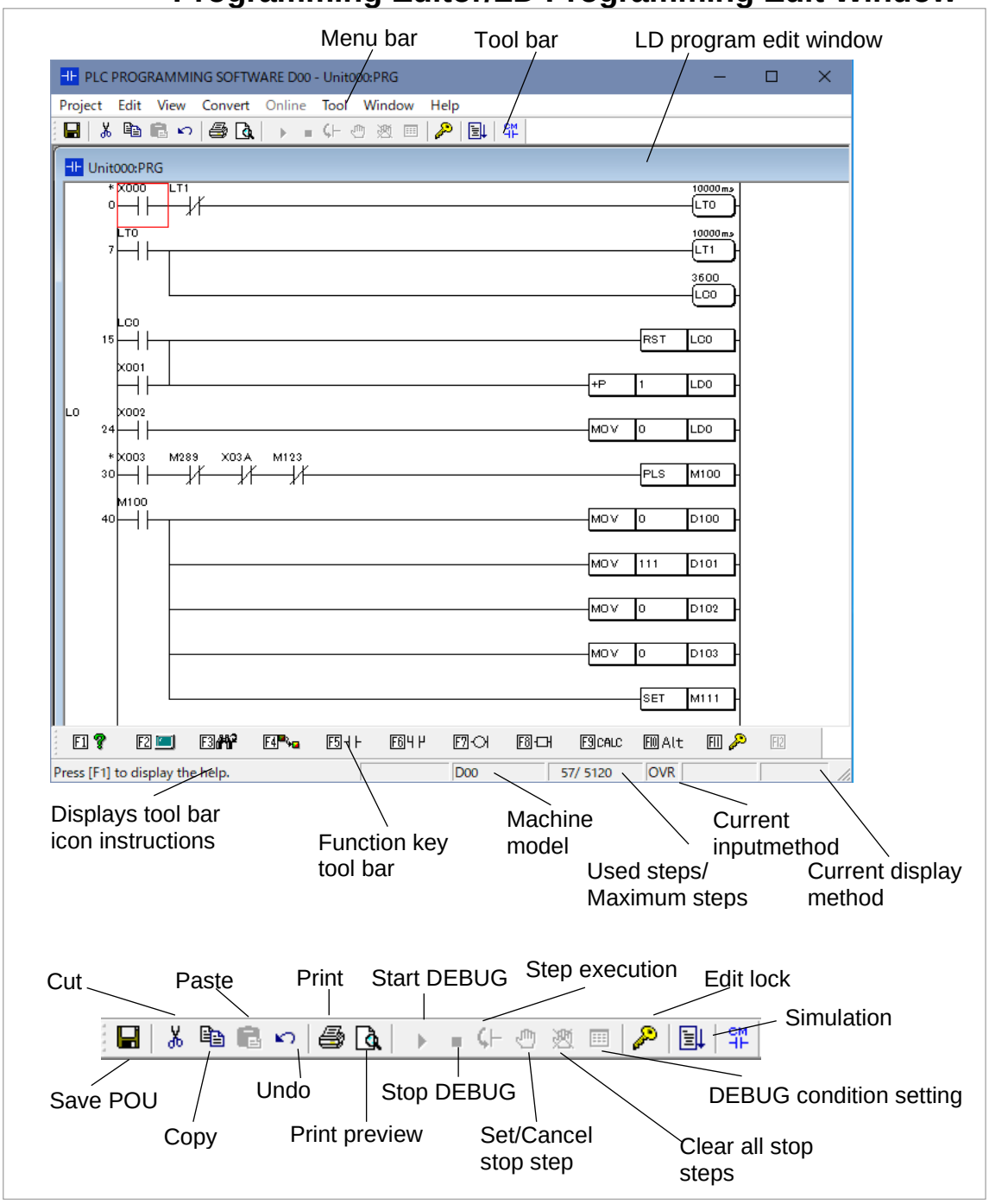
LD EDIT OPERATION PROCEDURE

7.1 LD Program Edit Screen Layout
7.2 Edit LD Programs
7.3 Replace All Contacts, OMs & IXs
7.4 Create Comments
7.5 Edit Lock
7.6 Search
7.7 Search & Jump
7.8 Jump to Beginning, End & Specified Label in LD Program
7.9 Launch Simulation
7.10 NC(BUILT IN PLC) Monitoring
7.11 Run & Stop in NC(BUILT IN PLC)
7.12 Monitoring Operations
7.13 DEBUG Functions
7.14 Print
7.15 Display Manual

7.1 LD Program Edit Screen Layout

This section describes the LD program, its display and the layout of the edit screen. In addition, it also describes how to change the display.

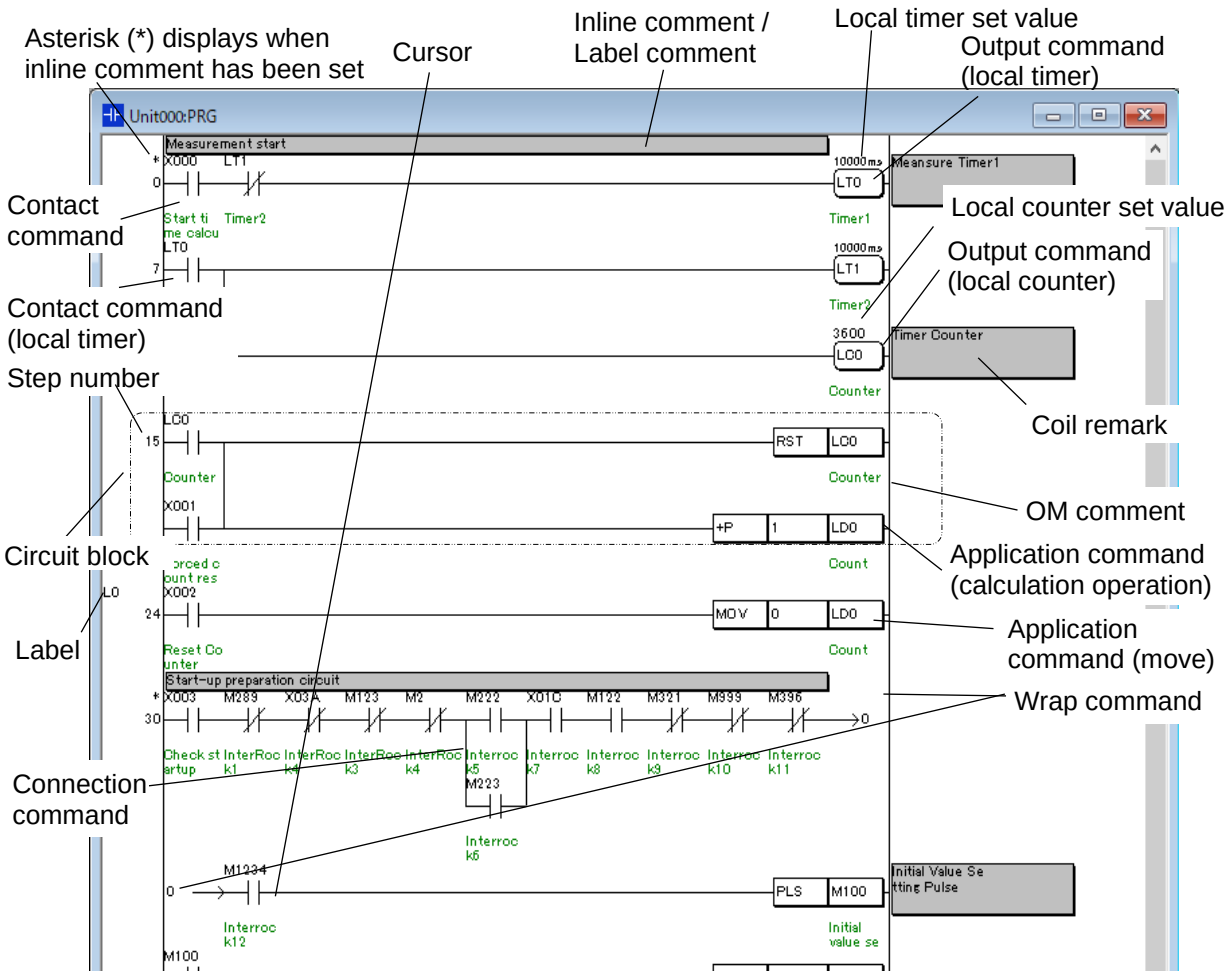
7.1.1 Layout of PLC Programming Software D00 LD/ST Programming Editor/LD Programming Edit Window



When the LD program edit window is shown in the full screen display, the window merges with the PLC Programming software D00 LD/ST Programming Editor into one screen display.

7.1.2 LD Program Components

The screenshot below shows the layout of the LD program display.



The user can change the column number, row number and display font in the LD program display.
Refer to “5.6.1.1 Edit screen adjustment (option settings)” for details on the operations.

7.1.3 Scroll Through Edit Screen

When displaying the LD program, the user can scroll, search through and jump to places in the LD program that are not displayed in the LD program edit window.
Refer to “7.6 Search”, “7.7 Search & Jump” and “7.8 Jump to beginning, end & specified label in LD program” for details on search and jump operations.

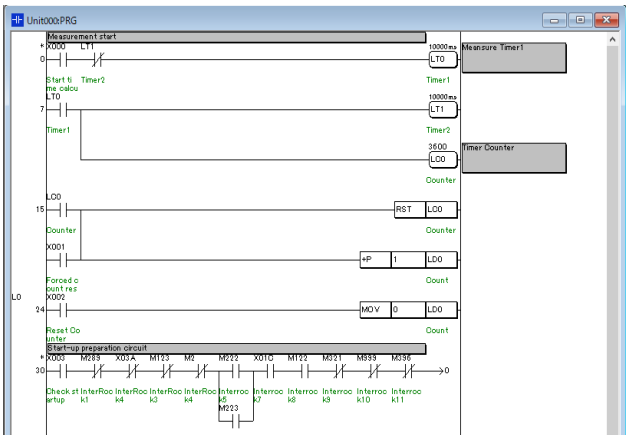
Scrolling through the screen can be executed using the following operations.
- Use mouse to conduct scroll bar operations on the right side of the screen
- Conduct operations with scroll wheel on mouse

The arrow keys and other keys can also be used to execute the following operations.
- Show circuits below current screen display : Moves cursor to bottom of screen **[↓]**
- Show circuits above current screen display : Moves cursor to top of screen **[↑]**
- Scroll up, jumping to next full screen : **[Page Up]**
- Scroll down, jumping to next full screen : **[Page Down]**
- Show screen from first circuit block : **[Home]**, **[Shift]**+**[Page Up]**, **[Ctrl]**+**[Page Up]**
- Move to last circuit (END command) : **[Shift]**+**[Page Down]** , **[Ctrl]**+**[Page Down]**

7.1.4 Change Display Order in LD Program

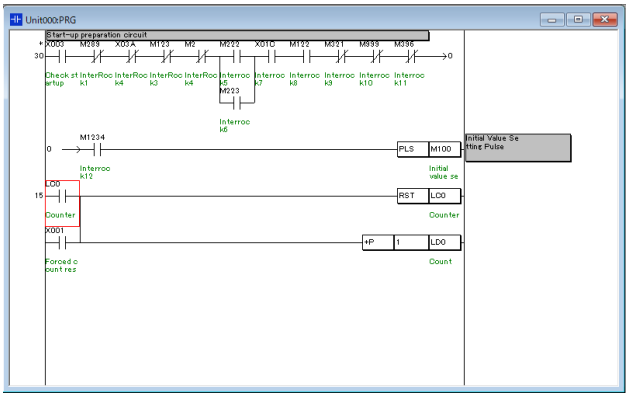
The LD program can be displayed or sorted based on the following two orders.
Step number display: Displays the LD program based on the step number order.
Search results display: Displays the circuit blocks in the order of the search results.

Step number display



Circuit blocks are displayed in order based on the step numbers.

Search results display



After conducting a search, the circuit blocks are displayed in the order of the search results.

(NOTE 1) If the user has not conducted a search or if the search information has been cleared, then the search results will not display. Even if the user performs the display operation, the operation is invalid.
(NOTE 2) Refer to “7.6 Search” for details on the search operations.

The user can perform one of the following operations to switch between the step number display and the search results display.
Operation procedure
Perform one of the following operations.
- Select <Find order> under <Change display> in the <Display> menu
- Press the **[Shift]** + **[Home]** keys

The user can perform one of the following operations to switch from the the search results display to the step number display.
Operation procedure
Perform one of the following operations.
- Select <Sort by step number> under <Change display> in the <Display> menu.
- Press the **[ESC]** keys

When changing from the search results display to the step number display, the user must perform an operation to change the display. When the display is changed, then the circuit block with the cursor becomes the top of the display.
Note that the search information is deleted when the user starts editing the LD program.

If the user performs an OM search or command search while the step number display is shown, it automatically changes to the search results display.
If the user starts editing the LD program, performs a step number search or performs a jump while the search results display is shown, then the display will change automatically to the step number display.

7.1.5 Change Comment Display

Each type of comment that the user sets is displayed in the LD program.
The following comments can be set in the LD program.

- OM comment
- Inline comment
- Label comment
- Coil remark

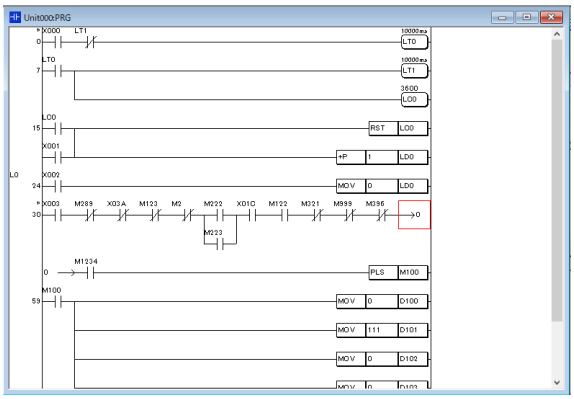
Refer to “7.4.1 Create OM comment” and “7.4.2 Edit OM comment list” for details on editing OM comments.
Refer to “7.4.3 Create inline comment” for details on editing inline comments.
Refer to “7.4.4 Create label comment” for details on editing label comments.
Refer to “7.4.5 Create coil remark” for details on editing coil remarks.

Refer to “10.11.1 Display Comments (LD Language)” for further details on comments.

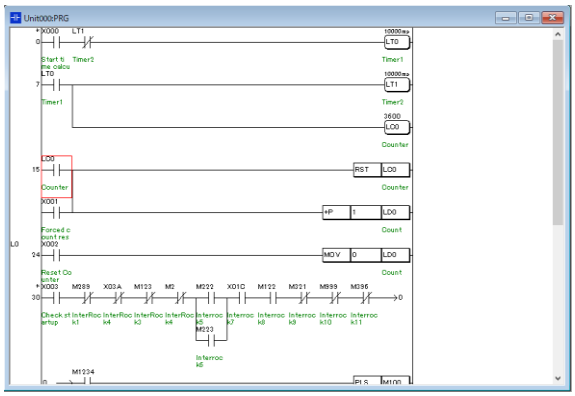
7.1.5.1 Change OM Comment Display

An OM comment that the user sets is displayed in the LD program.

Hide OM comments



Show OM comments



OM comments are displayed in green.

The user can switch between showing and hiding the OM comments by performing one of the following operations.

<u>Operation procedure</u>

Perform one of the following operations.

- Press the **[Ctrl]** + **[F5]** keys
- Select <Display with OM comment> under <Change display> in the <Display> menu.

In addition, when the display shows the comments, the user can change the number of characters in the comment display to 16 (half-width) characters or 32 (half-width) characters.

<u>Operation procedure</u>

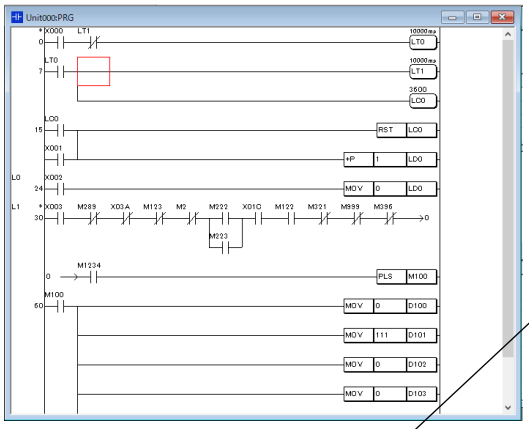
Select “16→32” to change the number of characters in the OM comment display under <Change display> in the <Display> menu.

(NOTE) If the OM comment is set to a 32 character display, the aforementioned menu item will then change to “32→16” (number of characters in the OM comment display). When this menu item is selected, then the OM comment changes to a 16 character display.

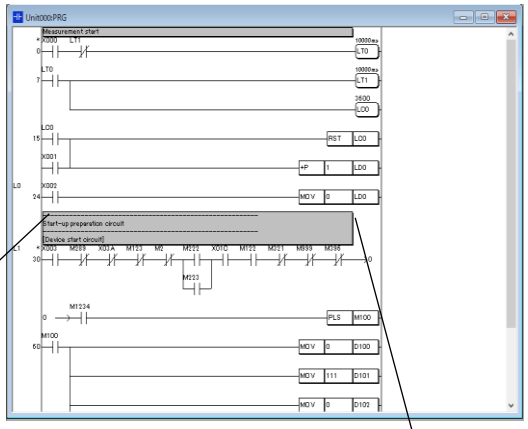
7.1.5.2 Change inline comment / label comment display

In an LD program, the user can display comments (inline comments) at the beginning of the circuit block.
In addition, when a label comment is set, it is displayed in the same area as the inline comment.

Hide inline comments



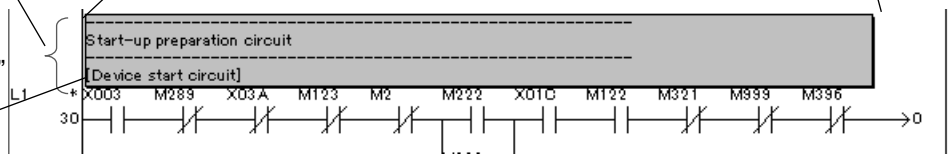
Show inline comments



Inline comment / label comment set by user is displayed above the circuit block.

Inline comment

Quotation marks " " are automatically added to the label comment "L1"



The user can switch between showing and hiding the inline comments by performing one of the following operations.

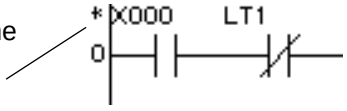
<u>Operation procedure</u>

Perform one of the following operations.

- Press the **[Ctrl]** + **[F7]** keys
- Select <Display with inline comment> under <Change display> in the <Display> menu.

When an inline comment or label comment is set, an asterisk (*) is displayed above the step number.

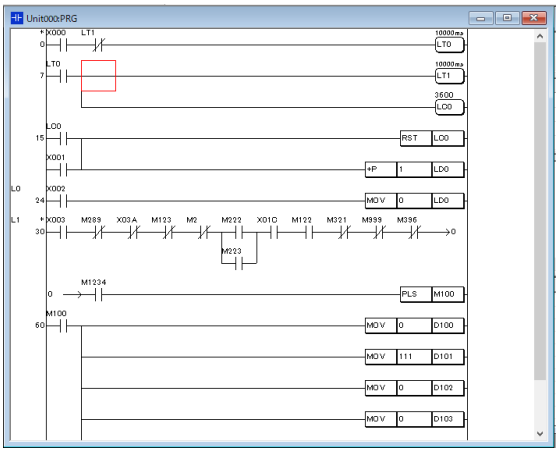
Displays when an inline comment or label comment is set.



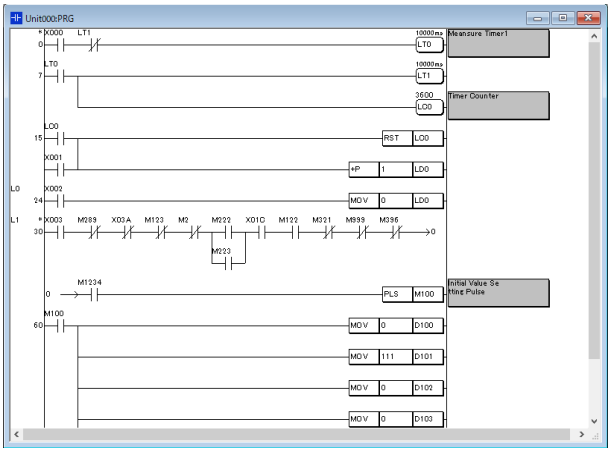
7.1.5.3 Change coil remark display

In an LD program, the user can display comments (coil remarks) for output commands and application commands.

Hide coil remarks



Show coil remarks



Coil remarks set by the user are displayed to the right of the specified output commands.

The user can switch between showing and hiding the coil remarks by performing one of the following operations.

<u>Operation procedure</u>

Perform one of the following operations.

- Press the **[Ctrl]** + **[F8]** keys
- Select <Display with coil remark> under <Change display> in the <Display> menu.

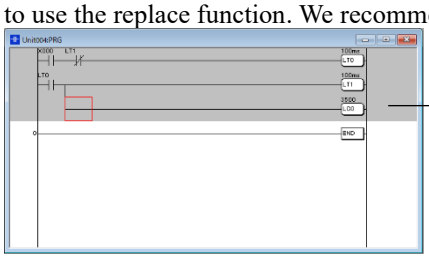
7.2 Edit LD Programs

This section describes the operations to edit an LD program.

7.2.1 Procedure to Create LD Programs

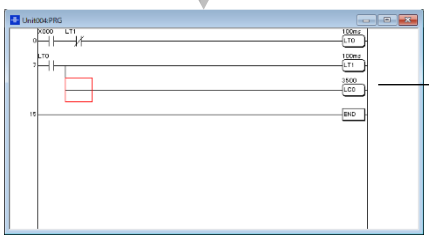
When the LD program edit window is launched, the LD program is displayed and can be edited. The user can enter contact commands, output commands, application commands and other commands to edit the LD program.
A circuit block being edited is highlighted in gray, and after it is converted, the color turns white.
Refer to “Chapter 11 Commands (LD)” in the PLC System Manual for details on how to create an LD program.

When the user performs a “Conversion” operation while an LD program is being created or after it is created, that operation not only checks the LD program for invalid statements, but converts the program to a format that can be executed in the NC(BUILT IN PLC).
When the program data is not converted, there are restrictions on operations, such as being unable to use the replace function. We recommend converting and saving the LD programs frequently.



LD program being editing turns gray.

Conversion operation



LD program after conversion

Refer to “5.9 Conversion” for details about conversion.
Refer to “5.2.3 Save POUs” for details about saving an LD program.

- “Overwrite mode” / “Insert mode”
  There are two modes for editing an LD program: “Overwrite mode” and “Insert mode”.
  In “Overwrite mode”, the specified command at the cursor position is overwritten. This mode is used when overwriting a command in the LD program.
  In “Insert mode”, the specified command at the cursor position is inserted. This mode is used when adding a command to the LD program.
  Each time the **[Insert]** key is pressed, the mode changes back and forth between “Overwrite mode” and “Insert mode”.

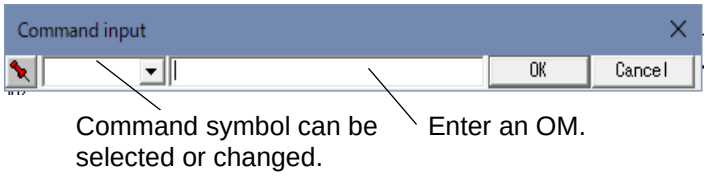
- Edit lock
  There is an “Edit lock” function that prohibits edit operations in order to avoid inadvertent operations and changes to an LD program that was created.
  When using the “Edit lock” function, changes to the LD program are prohibited, and comments (such as OM comments, etc.) as well as coil remarks cannot be edited.
  Refer to “7.5 Edit lock” for details on the “Edit lock” function.

7.2.1.1 Contact command input method

The following can be used to select a circuit symbol.

- Function key
- Function key tool bar
- Menu for the <Instructions> under the <Edit> menu

When a circuit symbol is selected, the <Command input> dialog is displayed. Enter an OM into the data input field, and then click on the <OK> button.

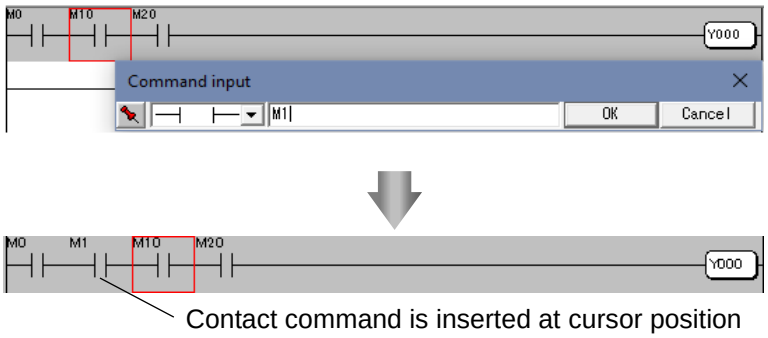


Function key	Circuit symbol	Command
[F5]	─┤ ├─	LD, AND
[F6]	└┤ ├┘	OR
[Shift]+[F5]	─┤/├─	LDN, ANDN
[Shift]+[F6]	└┤/├┘	ORN
[Shift]+[F7]	─┤↑├─	LDP, ANDP
[Shift]+[F8]	─┤↓├─	LDNP, ANDNP
[Shift]+[F11]	─┤↑/├─	LDPI, ANDPI
[Shift]+[F12]	─┤↓/├─	LDNPI, ANDNPI

When the user moves the cursor to the desired position (to input a command) and after a letter or symbol key is input, a <Command input> dialog is displayed and the user can enter a command and OM.

In this situation, the input is carried out from a drop-down command list. A space is input between the command and OM.

- When in insert mode
  When in insert mode, if the user enters a contact command, that contact command is inserted at the cursor position.
  Example



(NOTE 1) When correcting a command that has already been input, use “Overwrite mode” to make the correction.
In “Insert mode”, a command is added or inserted at the cursor position.

(NOTE 2) Connection commands such as AND-BLK and OR-BLK cannot be entered in the command input dialog. When vertical and horizontal lines are input to make connections, the connection commands are automatically applied.

(NOTE 3) After clicking on the <OK> button while the button is turned ON, the <Command input> dialog does not close and the user can continue entering another command.

(NOTE 4) When the user also double-clicks on the cursor position, the <Command input> dialog is displayed.

(NOTE 5) Move the cursor to the position where there is already a circuit symbol, and double-click on it or press the **[Enter]** key. The circuit symbol and the OM name / application command are displayed in the circuit input dialog.

(NOTE 6) When creating an OR circuit and the next circuit is not an END command, use "Insert mode" to input. Or, insert a blank line before the circuit on the next line and then input an OR command. To insert a blank line before the circuit on the next line, move the cursor to the circuit on the next line and then press the **[Shift]** + **[Insert]** keys.

(NOTE 7) If the number of contacts does not fit on 1 line, the "Wrap" function can be used to increase the contact commands that connect in a series. Refer to "7.2.1.8 Wrapping" for details on the wrapping function.

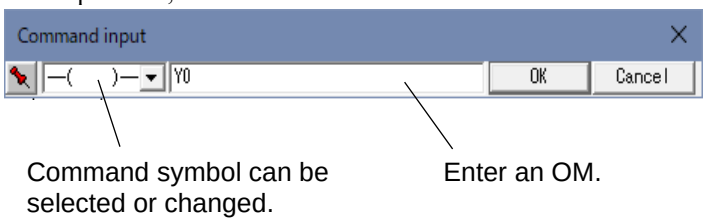
(NOTE 8) Refer to "Chapter 11 Commands (LD)" in the PLC System Manual for details on the commands.

7.2.1.2 Output command input method

The following can be used to select a circuit symbol.

- Function key
- Function key tool bar
- Menu for the <Instructions> under the <Edit> menu

When a circuit symbol is selected, the <Command input> dialog is displayed. Enter an OM into the data input field, and then click on the <OK> button.

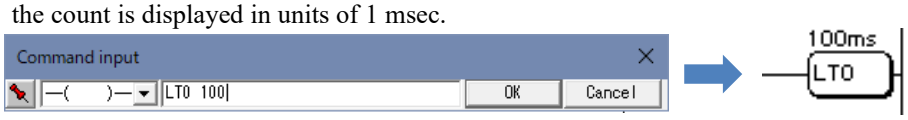


Function key	Circuit symbol	Command
[F7]	—< >—	OUT
[F8]	—[ ]—	Output besides OUT

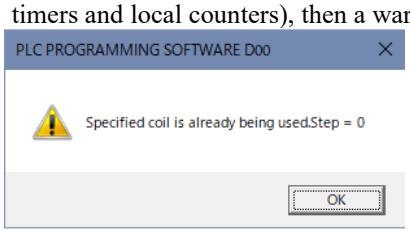
When the user moves the cursor to the desired position (to input a command) and after a letter or symbol key is input, a <Command input> dialog is displayed and the user can enter a command and OM.

In this situation, the input is carried out from a drop-down command list. A space is input between the command and OM.

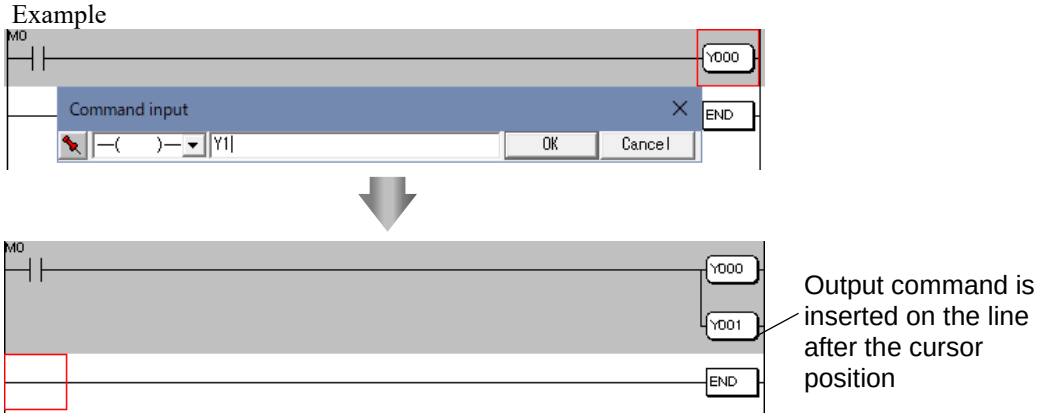
- Local timer and local counter input
  When an OM is a local timer (LT) and local counter (LC), the set value is input after the OM. The value for counting up can be set in units of 10 msec. However, when actually counting up, the count is displayed in units of 1 msec.



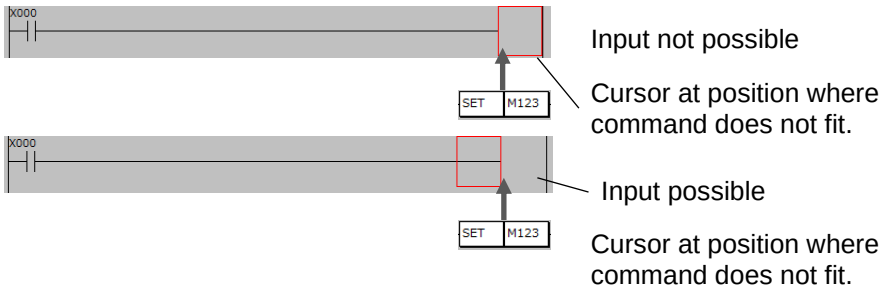
- Double coil check
  When an OM is set that is already used in the output for an OUT command (including local timers and local counters), then a warning is displayed. Correct the OM when necessary.



- When in insert mode
  When in insert mode, if the user enters an output command, that output command is inserted on the line after the cursor position.
  Example



(NOTE 1) When entering an output command, insert it when the cursor is to the left of the position where the command goes.
If the output command is not an OUT command, then it cannot be entered when the cursor is at the rightmost column. Otherwise, the message "Invalid execution point." is displayed.



(NOTE 2) When correcting a command that has already been input, use "Overwrite mode" to make the correction. In "Insert mode", a command is added or inserted at the line following the cursor position.

(NOTE 3) When entering an output command, a horizontal line is automatically drawn from the cursor position to the output command and the output command is connected to the right rail (vertical line).

(NOTE 4) After clicking on the <OK> button while the button is turned ON, the <Command input> dialog does not close and the user can continue entering another command.

(NOTE 5) When the user also double-clicks on the cursor position, the <Command input> dialog is displayed.

(NOTE 6) Move the cursor to the position where there is already a circuit symbol, and double-click on it or press the **[Enter]** key. The circuit symbol and the OM name / application command are displayed in the circuit input dialog.

(NOTE 7) When entering multiple output commands in parallel under the same conditions and the next circuit is not an END command, use "Insert mode" to input. Or, insert a blank line before the next line's circuit, enter a vertical line and then enter the output commands. To insert a blank line before the circuit on the next line, move the cursor to the circuit on the next line and then press the **[Shift]** + **[Insert]** keys. To enter a vertical line, press **[Shift]** + **[↑]**, or press **[Shift]** + **[↓]**.

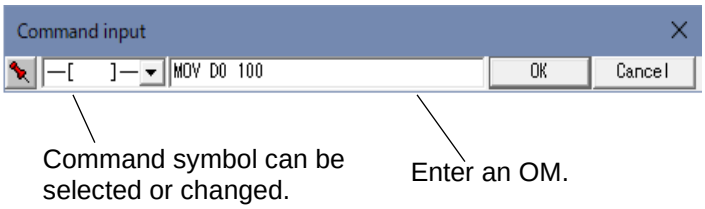
(NOTE 8) Refer to "Chapter 11 Commands (LD)" in the PLC System Manual for details on the commands.

7.2.1.3 Application command input method

The following can be used to select a circuit symbol.

- Function key
- Function key tool bar
- Menu for the <Instructions> under the <Edit> menu

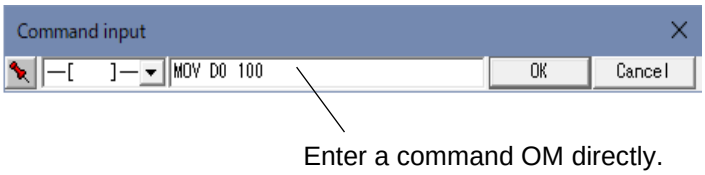
When a circuit symbol is selected, the <Command input> dialog is displayed. Enter an OM into the data input field, and then click on the <OK> button.



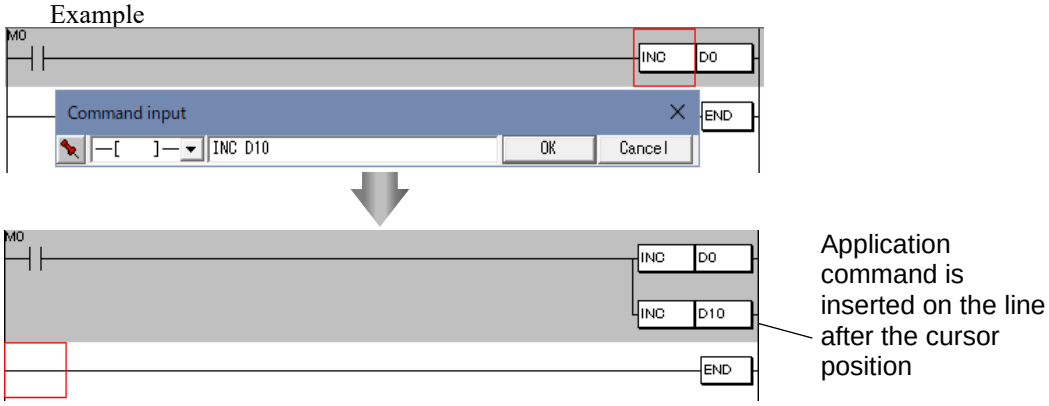
Function key	Circuit symbol	Command
[F8]	⊣[ ]⊢	Application command

When the user moves the cursor to the desired position (to input a command) and after a letter or symbol key is input, a <Command input> dialog is displayed and the user can enter a command and OM.

In this situation, the input is carried out from a drop-down command list. A space is input between the command and OM.

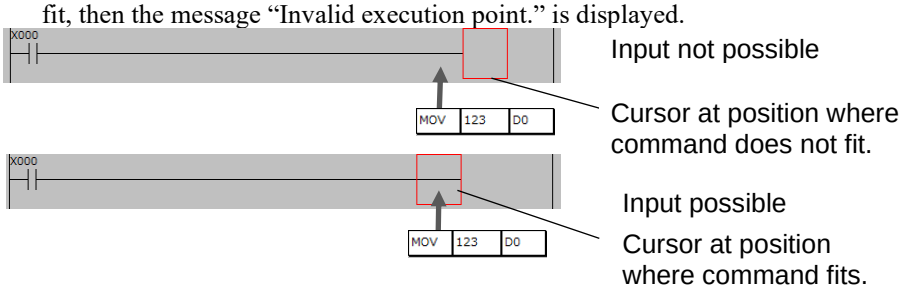


- When in insert mode
  When in insert mode, if the user enters an application command, that application command is inserted on the line after the cursor position.
  Example



However, when a data comparison command is entered in input mode, the data comparison command is inserted at the cursor position.

(NOTE 1) When entering an application command, insert it when the cursor is to the left of the position where the command goes.
If a command is entered and the cursor is at a position where the command does not fit, then the message "Invalid execution point." is displayed.



(NOTE 2) When correcting a command that has already been input, use "Overwrite mode" to make the correction.
In "Insert mode", a command is added or inserted at the cursor position.

(NOTE 3) When entering an application command (besides "data comparison"), a horizontal line is automatically drawn from the cursor position to the application command and the application command is connected to the right rail (vertical line).

(NOTE 4) After clicking on the <OK> button while the button is turned ON, the <Command input> dialog does not close and the user can continue entering another command.

(NOTE 5) When the user also double-clicks on the cursor position, the <Command input> dialog is displayed.

(NOTE 6) Move the cursor to the position where there is already a circuit symbol, and double-click on it or press the **[Enter]** key. The circuit symbol and the OM name / application command are displayed in the circuit input dialog.

(NOTE 7) When entering multiple output commands in parallel under the same conditions and the next circuit is not an END command, use "Insert mode" to input. Or, insert a blank line before the next line's circuit, enter a vertical line and then enter the output commands. To insert a blank line before the circuit on the next line, move the cursor to the circuit on the next line and then press the **[Shift]** + **[Insert]** keys. To enter a vertical line, press **[Shift]** + **[↓]**, or press **[Shift]** + **[↑]**.

(NOTE 8) Refer to "Chapter 11 Commands (LD)" in the PLC System Manual for details on the commands.

7.2.1.4 CALC command input method

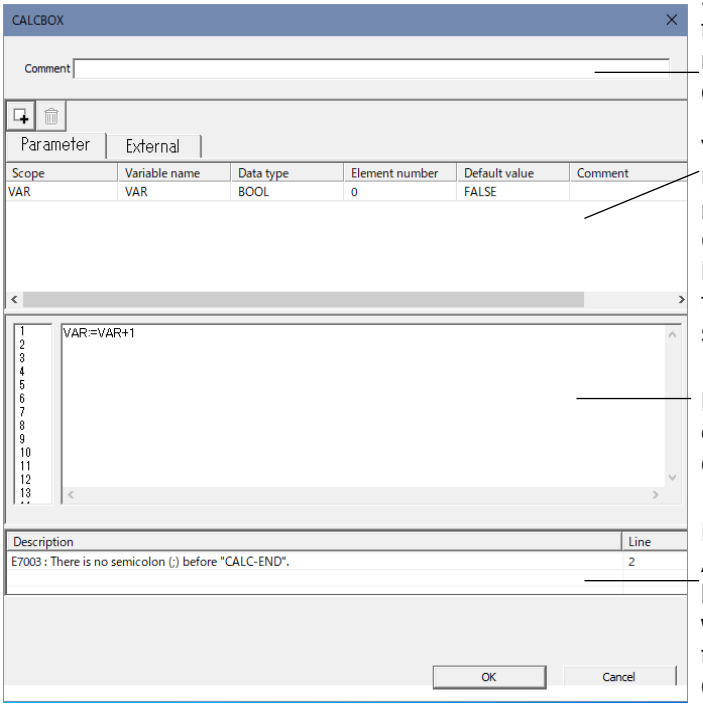
This section describes how to enter a CALC command.
The CALC command is a function that adds processing via an ST program within an LD program. The ST language and those languages that are used in an ST program can be used to write in a CALC command. Refer to "Chapter 8 ST edit operation procedure" and the "Chapter 12 Commands (ST)" in the PLC System Manual for further details.

The user can use a maximum of 10 CALC commands in one LD program. The user can enter a maximum of 1024 characters in an ST program for one CALC command.
In a CALC command, the user can register variables that can only be used in a CALC command.

<u>Operation procedure</u>

1. Move the cursor to the desired location to create a CALC command.
2. Use one of the following operations to display the <CALCBOX> dialog.
   - Press the **[F9]** keys
   - Select <CalcBox> under <Instructions> in the <Edit> menu.
   - Select <CALCBOX> from the circuit symbols in the <Command input> dialog.
3. Use the ST language to enter the process into the <CALCBOX>, and then click on the <OK> button.
   The dialog closes if there is no error or warning in the ST program.

The layout of the <CALCBOX> dialog screen is shown below.



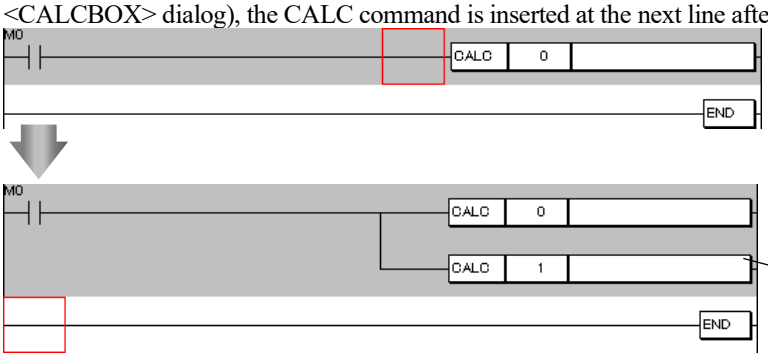
Set the comment to display in the CALC command. Use no more than 20 characters for the comment.

Variable setting area
Under the <Parameter> tab, register variables to use in a CALC command.
Refer to “8.2 Variable settings” for details on the registration settings.

Refer to “8.3.1 Procedure to create ST programs” for details on ST program notation.

Message display area
After pressing the <OK> button, if there is an error or warning found when checking the program, this message is displayed.

- When in insert mode
  After entering a CALC command in insert mode (after clicking on the <OK> button in the <CALCBOX> dialog), the CALC command is inserted at the next line after the cursor position.



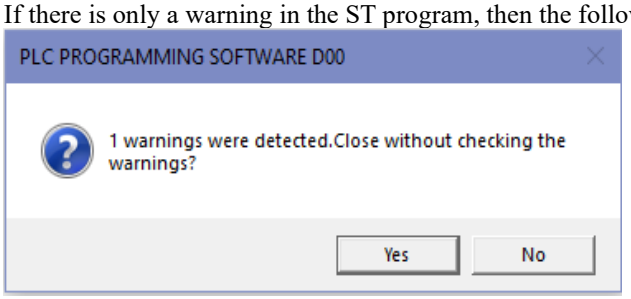
CALC command is inserted on the next line after the cursor position

(NOTE 1) When there is an error message in the message display area, the edit dialog cannot be closed using the <OK> button. Revise the ST program.

(NOTE 2) When correcting a command that has already been input, use “Overwrite mode” to make the correction. In “Insert mode”, a command is added or inserted at the cursor position.

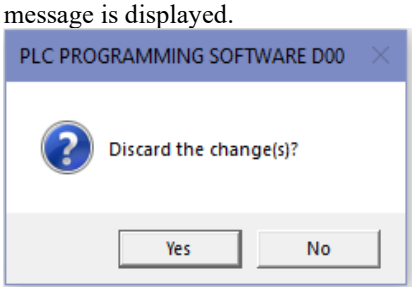
When there is an error or warning in the ST program for the CALC command, after clicking on the <OK> button for the <CALCBOX>, an error description appears in the message display area.
Correct the error, and then click on the <OK> button again.

If there is only a warning in the ST program, then the following message is displayed.



Select <No> to check and correct the warning location(s). Select <Yes> to close the dialog without checking or correcting the warning location(s).

If the user clicks on the <Cancel> button after changing the ST program, then the following message is displayed.

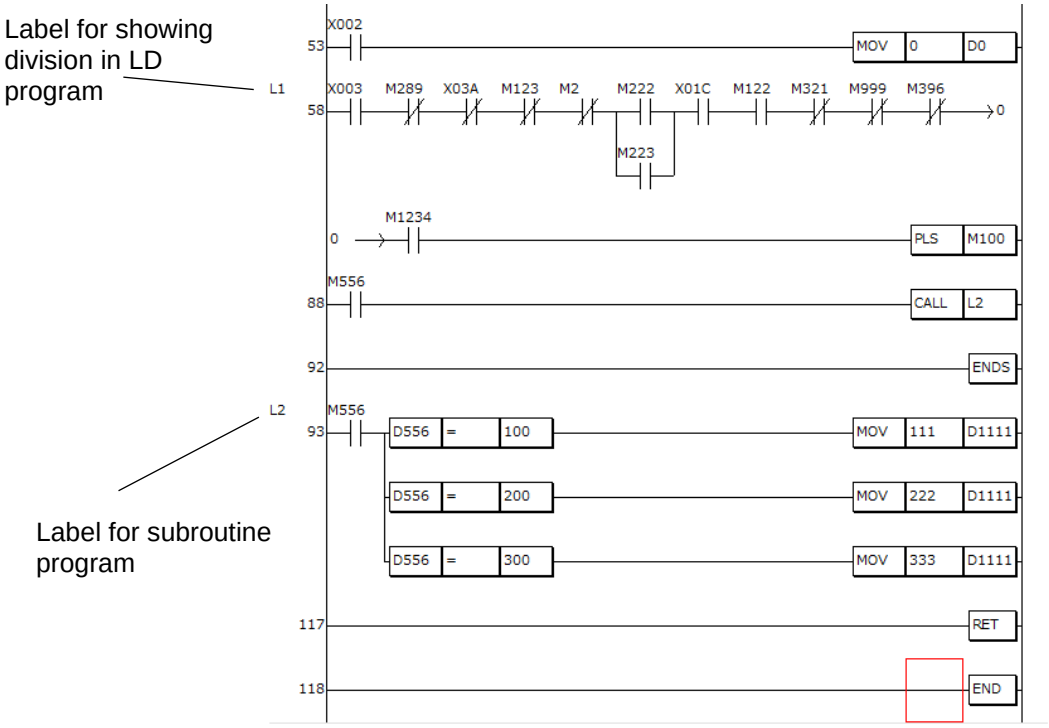


Select <Yes> to discard the changes. Select <No> to go back to the dialog.

Refer to “Chapter 12 Commands (ST)” in the PLC System Manual for details about errors and warnings in the ST program.

7.2.1.5 Label (L) input method

This section describes how to insert a label (L) for identifying subroutine programs or making divisions in the program to make it easier to understand.



A label can be inserted only at the beginning of the circuit block by using the <Command input> dialog.
However, L1023 refers to the end of the program and cannot be input by the user.
Use one of the following operations for label input.

Operation procedure

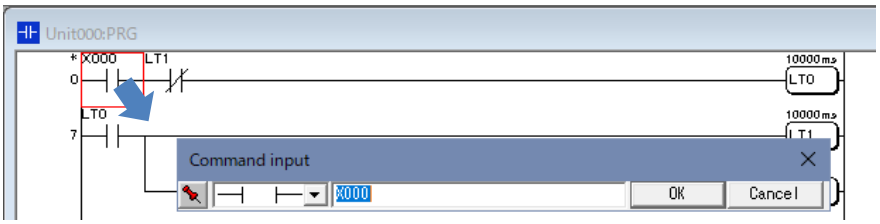
Perform one of the following operations.

- Move the cursor to the desired position for label input, and enter a label (Lx, X: 0 to 254).
- Double-click on the mouse at the desired position for label input.
- Move the cursor to the desired position for label input, and press the **[Enter]** key.

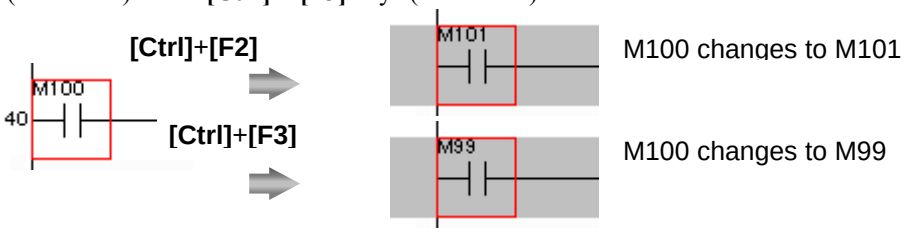
- Delete label
  Move the cursor to the desired position to delete the label, and press the **[Delete]** key.

7.2.1.6 Revise existing commands

- Revise commands
  When revising an existing command, change the mode to "Overwrite mode". Then, move the cursor to the desired command and double-click on it, or press the **[Enter]** key.
  The command input dialog is displayed with the current command settings.
  In the command input dialog, revise the command symbol or OM and then click on the <OK> button.



- OM number increase (increment) or decrease (decrement)
  When editing an LD program, the user can increase or decrease an OM number for an existing command in the PLC Programming software D00 LD/ST Programming Editor.
  Move the cursor to the desired OM in the LD program, and then press the **[Ctrl]** + **[F2]** keys (increment) or the **[Ctrl]** + **[F3]** keys (decrement).



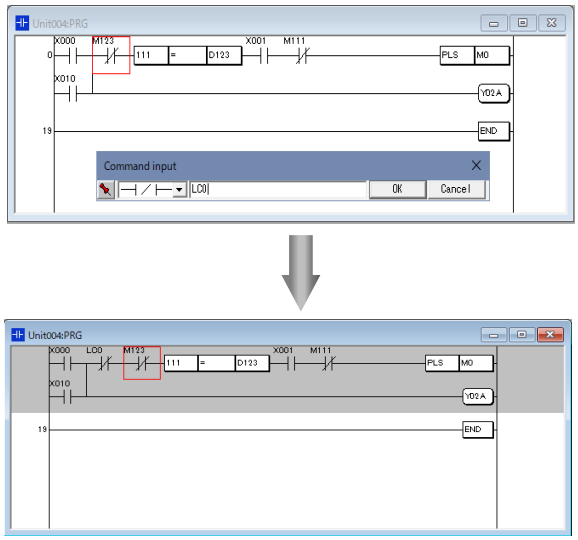
The user can increase (increment) or decrease (decrement) the OM number for a contact command, output command, application command or label.

7.2.1.7 Insert or delete commands

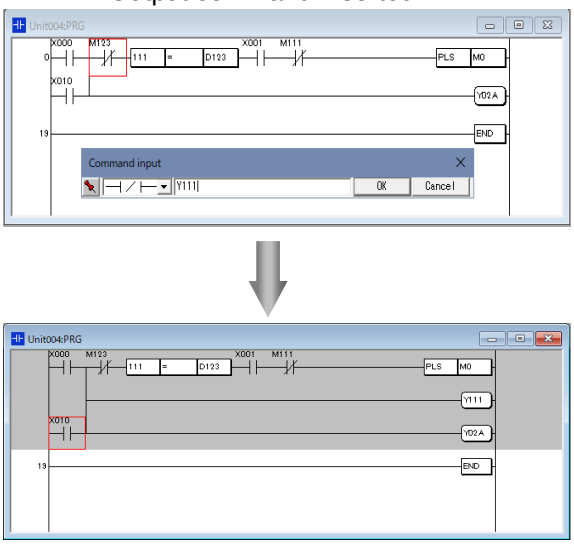
This section describes how to insert or delete a command.
The command input and deletion operations vary depending on the whether the input format is in "Overwrite mode" or "Insert mode".

- Insert commands
  When inserting a command, change the input format to "Insert mode" and then enter a command.
  The command is inserted at the cursor position.

Contact command inserted

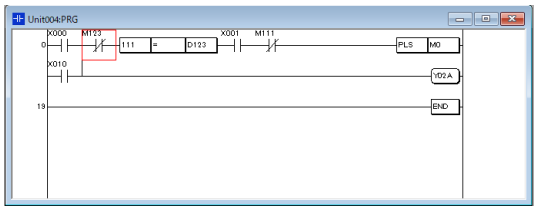


Output command inserted



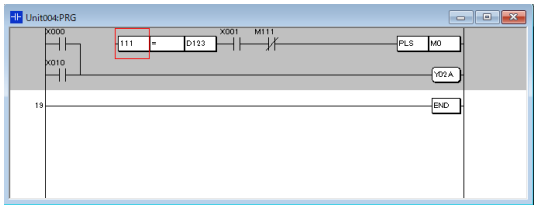
However, when using insert mode and when a circuit is created that extends beyond the right rail after inserting a command, a "Wrap" is automatically applied. Refer to "7.2.1.8 Wrapping" for details on the wrapping function.

- Delete command and "horizontal line"
  Press the **[Delete]** key delete a command (where the cursor is) and the corresponding horizontal line.



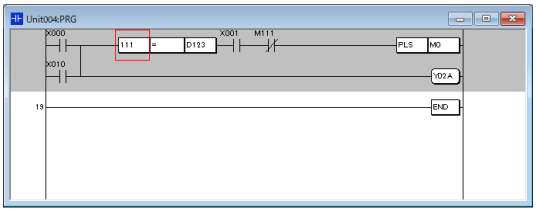
When deleting a "Command" where the cursor is.

When in overwrite mode



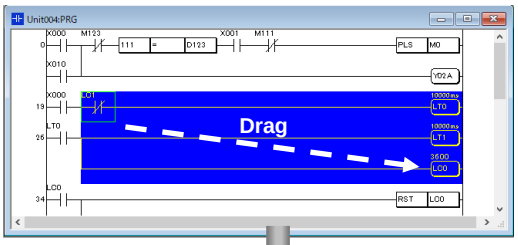
Command is deleted and the line becomes blank.

When in insert mode



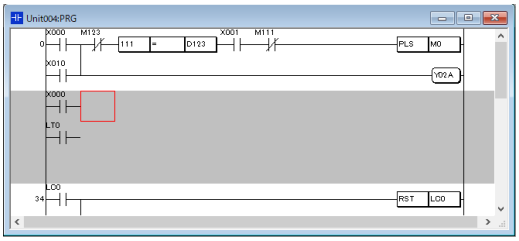
Command is deleted and becomes a "Horizontal line".

- Specify range to delete circuit
  The user can use the mouse to drag and specify the range to delete.
  The specified range is highlighted in blue. (If the circuit is already highlighted in yellow due to a conversion error, then the selected range is displayed in black.)
  Press the **[Delete]** key to delete the specified range.



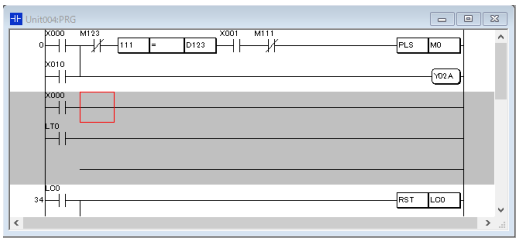
Use mouse to specify the range.

When in overwrite mode



Command is deleted and the lines becomes blank.

When in insert mode



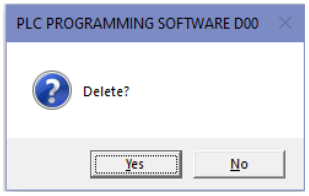
Command is deleted and becomes a “Horizontal line”.

- Delete circuit block
  A circuit is deleted in circuit block units.
  When deleting a circuit block, the circuit block at the cursor position is deleted regardless of the input format (mode).

  Operation procedure1

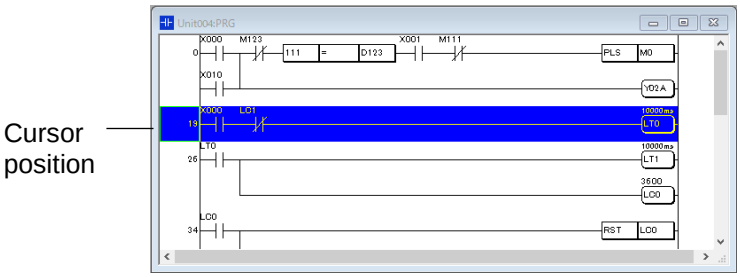
  1. Move the cursor to the desired circuit block, and press the **[Ctrl]** + **[F1]** keys.

  2. Confirmation message is displayed.
     Select <Yes> to delete the circuit block.

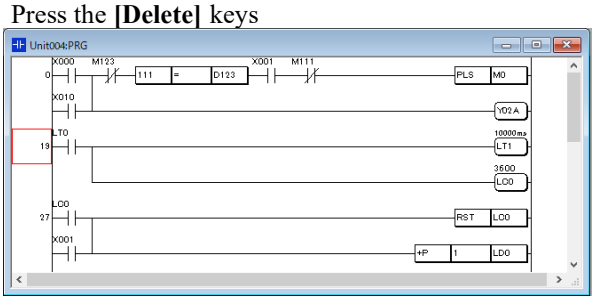


Operation procedure2

1. Move the cursor to the left side of the left rail (where step number is displayed) on the desired circuit block and then press the **[Shift]** + **[↓]** keys. The circuit block becomes selected.

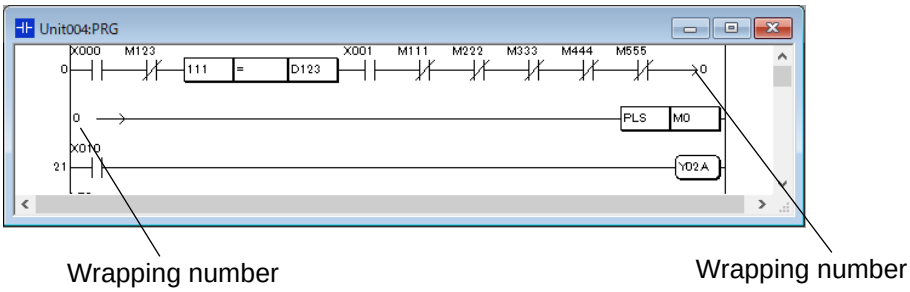


2. Press the **[Delete]** keys



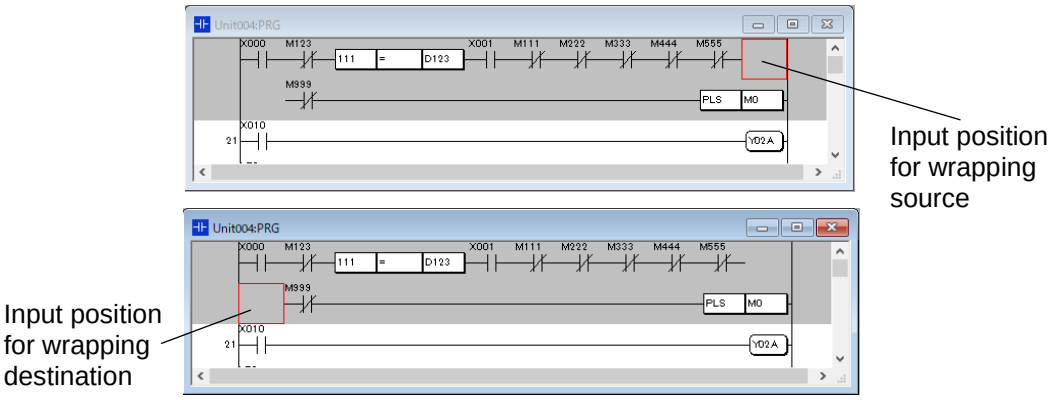
7.2.1.8 Wrapping

If a condition does not fit on one line, this function is used to wrap the condition notation to the next line.



"Wrapping" can only be inserted under the following conditions at the following locations.

- Wrapping source: Input possible only when cursor is positioned to the left of the right rail.
- Wrapping destination: Input possible only when cursor is positioned to the right of the left rail.



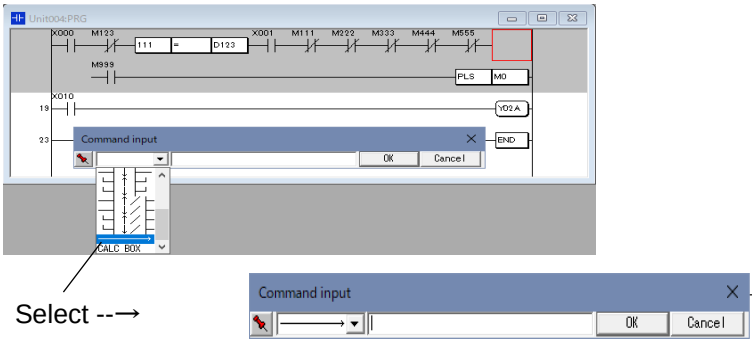
Always use the same number for the wrapping number in the wrapping source and destination. If a different wrapping number is used in the wrapping source and destination, then an invalid circuit will be created.

In addition, if another circuit block is input between the wrapping source and destination, an invalid circuit will be created.
Any number from “0 to 32767” can be used for the wrapping number.
After conversion or a design check, the wrapping numbers are automatically renumbered from “0”.

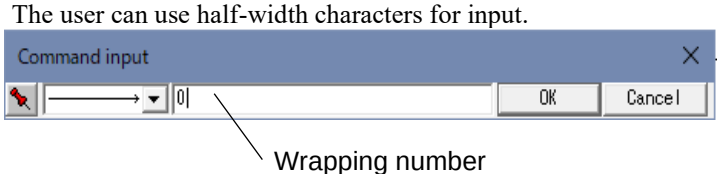
(NOTE) When the program is edited with another circuit block input between the wrapping source and destination, then an invalid program may be created. Therefore, do not insert another circuit block between the wrapping source and destination.

Operation procedure1

1. Perform one of the following procedures to select a wrapping symbol.
   - Press the **[F5]** or **[F6]** key to display the command input dialog (show below) and select a <Wrapping symbol> from the list box
   - Select <Insert connection> under <Instructions> in the <Edit> menu



2. Enter a “Wrapping number”.
   Enter a “Wrapping number” into the data input field next to the command symbol.
   The user can use half-width characters for input.

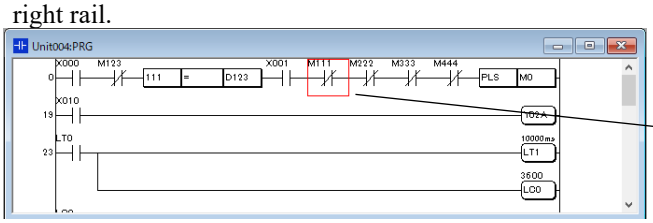


Wrapping input uses overwrite regardless of whether the mode is in “Overwrite mode” or “Insert mode”.

Operation procedure2

Set the input format to <Insert mode> and insert a horizontal line.

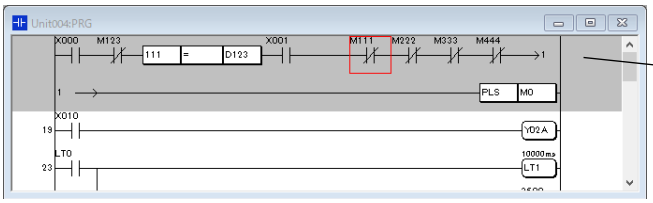
A wrap is automatically created when the horizontal line (or a created command) exceeds the right rail.



When cursor is positioned at a command on the line or on the horizontal line

<Insert mode>

[Shift] + **[→]** or **[Shift]** + **[←]**



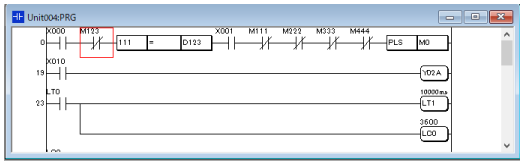
Wrapping number is automatically assigned and a wrap is created.

7.2.1.9 Input and delete vertical or horizontal line

This section describes how to input horizontal and vertical lines (alternatives to connection commands) as well as how to delete a line.

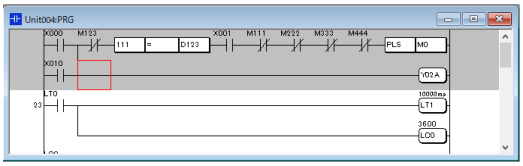
Horizontal and vertical line input and deletion operations vary depending on the whether the input format is in “Overwrite mode” or “Insert mode”.

- Processing when inserting a vertical line



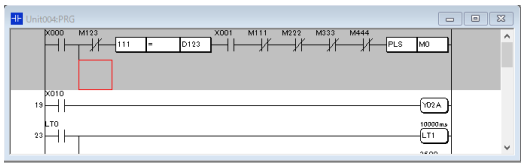
Vertical line is entered at the cursor position.

When in overwrite mode



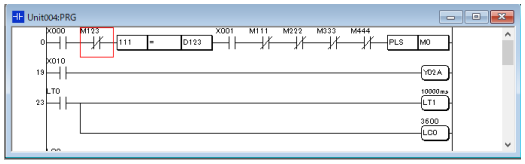
Connects to the next circuit with the vertical line.

When in insert mode



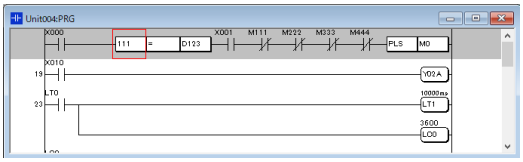
Vertical line is inserted.

- Processing when inserting a horizontal line



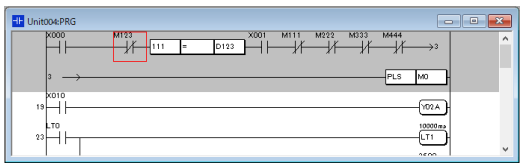
Horizontal line is entered at the cursor position.

When in overwrite mode



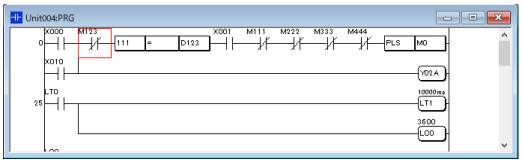
Overwrites command at the cursor position with a horizontal line.

When in insert mode



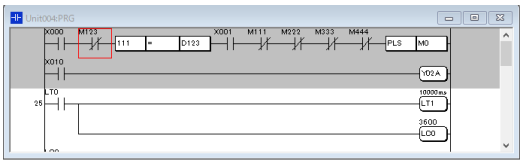
Horizontal line is inserted at the cursor position.

- Processing when deleting a vertical line



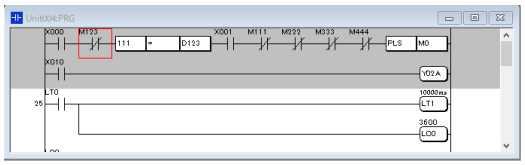
Vertical line at the cursor position is deleted.

When in overwrite mode



Vertical line is deleted.

When in insert mode



Vertical line is deleted.

<u>Operation procedure</u>

Use the following operations to enter and delete a vertical or horizontal line, or to delete a command.

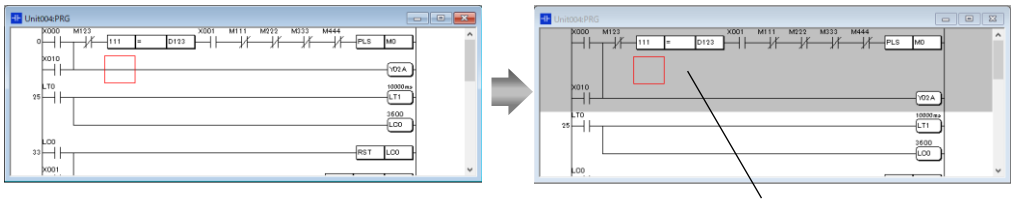
- Enter vertical line: **[Shift]** + **[↓]** keys, or **[Shift]** + **[↑]** keys
- Enter horizontal line: **[Shift]**+ **[→]**keys, or**[Shift]**+ **[←]** keys
- Delete vertical line: **[Ctrl]** + **[F9]** keys
  (deletes vertical line to bottom-left of cursor)
- Delete horizontal line: **[Delete]** key
- Delete command: **[Delete]** key

7.2.1.10 Insert and delete rung

The user can insert a blank rung or delete a rung.
When inserting and deleting a rung, the corresponding operation is the same regardless of the input format.
When the input format is set to “Insert mode” and a command is entered, a rung and command are inserted below the cursor position.

- Insert rung
  A blank rung is inserted at the cursor position.
  The user can perform one of the following operations to insert a rung.

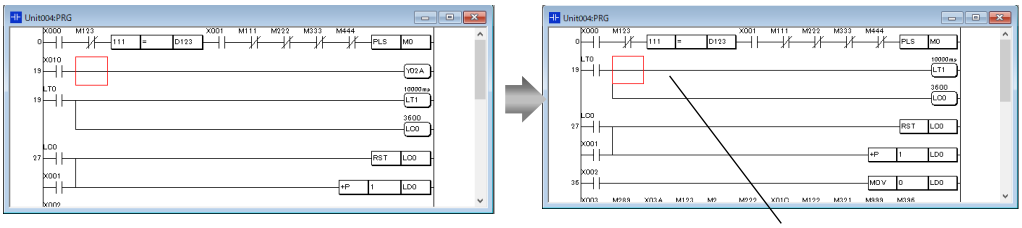
  Operation procedure
  Perform one of the following operations.
  - Press the **[Shift]** + **[Insert]** keys
  - Select <Insert rung> from the <Edit> menu



Blank rung is inserted.

- Delete rung
  Rung at the cursor position is deleted.
  The user can perform one of the following operations to delete a rung.

  Operation procedure
  Perform one of the following operations.
  - Press the **[Shift]** + **[Delete]** keys
  - Select <Delete rung> from the <Edit> menu.
  - Move the cursor to the left side of the left rail (where step number is displayed) on the desired rung and then press the **[Shift]** + **[↓]** keys to select the rung. Then, while selected, press the **[Delete]** key.



Rung above the cursor position is deleted

7.2.1.11 Design check

The design check function is for making sure there are no invalid notations in the POU that was created.
Refer to “5.7.2 Design check” for further details.

7.2.1.12 Conversion

Conversion refers to checking for invalid statements in POUs (created in a language in the PLC Programming software D00 EDITOR) and converting the data to code (intermediate code) that allows execution in the NC(BUILT IN PLC).
Refer to “5.9 Conversion” for further details.

- Write during run
  If the “Write during RUN after conversion” setting is used in the “Conversion mode settings” in the PLC Programming software D00, then write during run operation is executed after a conversion operation is carried out.
  Refer to “5.10.4 Write during run” for details on the write during run operation.

(NOTE) When using the write during run operation, take the necessary safety precautions with the device that has the NC(BUILT IN PLC) and also with the operating environment.

7.2.2 Undo

This section describes the undo operation in the LD program edit window.

The user can perform the undo operation a maximum of 20 times and it applies to the following edit operations.
- Cut and paste
- Write, insert or delete circuits
- Replace (all *a*/*b* contacts, all OMs or all IXs)
- Enter, delete or edit inline comments
- Enter, delete or edit OM comments
- Enter, delete or edit label comments
- Enter, delete or edit coil remarks
- Increase (increment) or decrease (decrement)
- Timer/Counter list
- Convert circuit

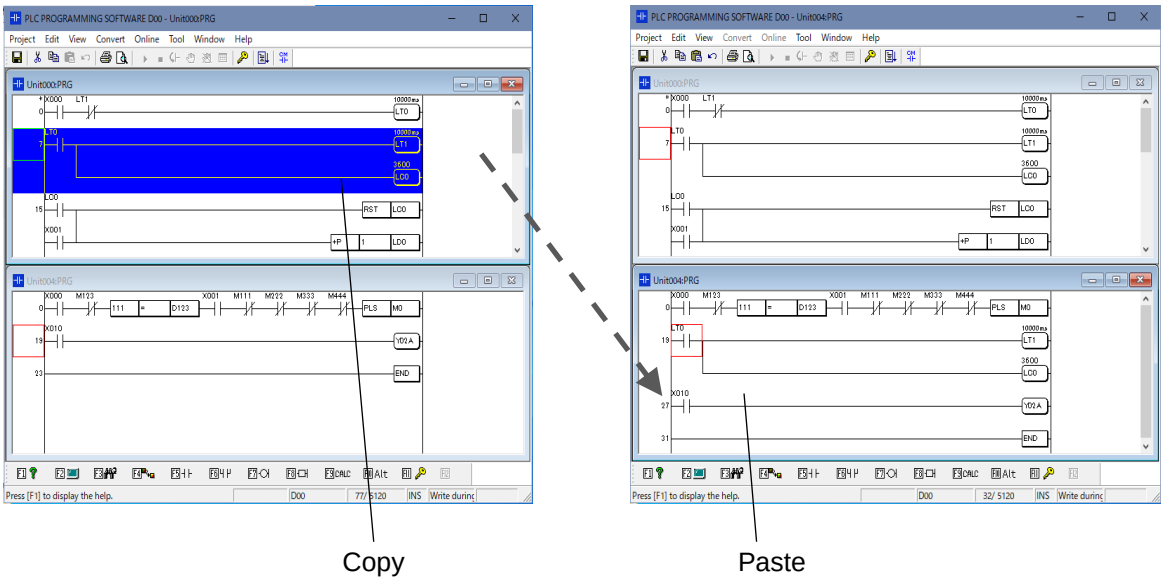
When the user changes the display in an LD program or when a screen is displayed after a search operation, the screen will not go back to the original or previous display screen.
The user can perform one of the following operations to undo.

<u>Operation procedure</u>
Perform one of the following operations.
- Select <Undo> from the <Edit> menu.
- Press the **[Ctrl]** + **[Z]** keys
- Click on the button in the tool bar.

7.2.3 Copy, Cut & Paste

This section describes the copy, cut and paste operations for a circuit that was created.
The paste operation varies depending on whether the input format is in “Overwrite mode” or “Insert mode”.

The LD program “Copy and paste” as well as the “Cut and paste” operations can also be carried out between LD program edit windows.
The user can perform a copy or cut operation in the LD program edit window that is active, and then the user can switch to another LD program edit window (make that window active) and paste that data into it.



7.2.3.1 Copy, cut and paste commands

The user can copy or cut a command at the cursor position and can paste a command to the cursor position.

The user can perform one of the following operations to copy.

Operation procedure
Perform one of the following operations.
- Select <Copy> from the <Edit> menu
- Press the **[Ctrl]** + **[C]** keys
- Click on the button in the tool bar.

The user can perform one of the following operations to cut.
After a cut operation, that location becomes blank.

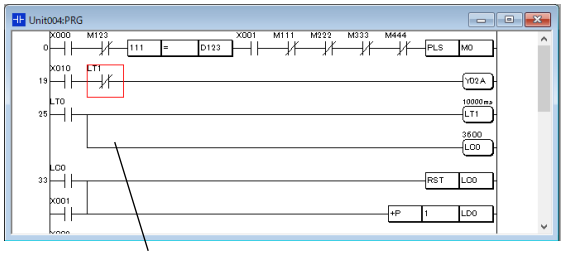
Operation procedure
Perform one of the following operations.
- Select <Cut> from the <Edit> menu
- Press the **[Ctrl]** + **[X]** keys
- Click on the button in the tool bar.

The user can perform one of the following operations to paste.
The paste operation varies depending on the whether the input format is in “Overwrite mode” or “Insert mode”.

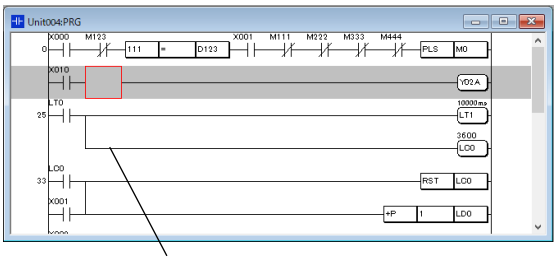
Operation procedure

Perform one of the following operations.

- Select <Paste> from the <Edit> menu
- Press the **[Ctrl]** + **[V]** keys
- Click on the button in the tool bar.

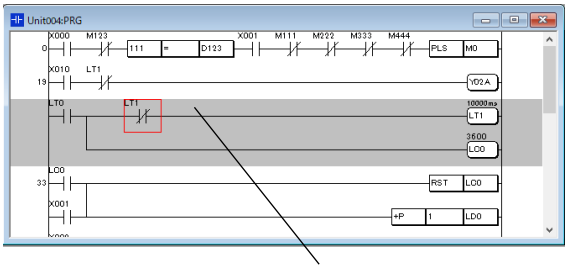


Copy



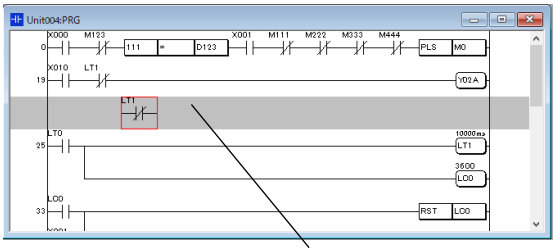
Cut

When in overwrite mode



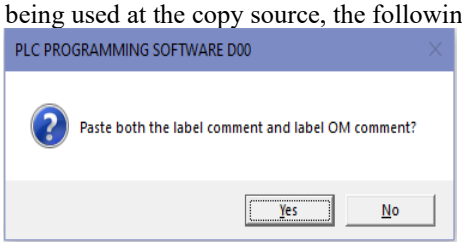
When performing a paste operation, the copied or cut command is pasted to the cursor position.

When in insert mode



When performing a paste operation, a rung is inserted at the cursor position and the copied or cut command is pasted.

When pasting a copied command to another POU, if there is a label (L) with an OM comment being used at the copy source, the following message is displayed.



Select <Yes> to paste the label's OM comment to the copy destination.
Select <No> to not paste the label's OM comment.

Even if <Yes> is selected, the OM comment for a label that is not included in the command (at the copy source) is not pasted.

(NOTE) If a coil remark is set at the copy source, then that coil remark is also pasted to the copy destination.

7.2.3.2 Copy, cut and paste rungs

The user can copy or cut a rung at the cursor position and can paste it to the rung at the cursor position.

The user can perform one of the following operations to copy a rung.

Operation procedure

Perform one of the following operations.

- Select <Copy rung> from the <Edit> menu.
- Press the **[Ctrl]** + **[L]** keys

The user can perform one of the following operations to cut or delete a rung.

Operation procedure

Perform one of the following operations.

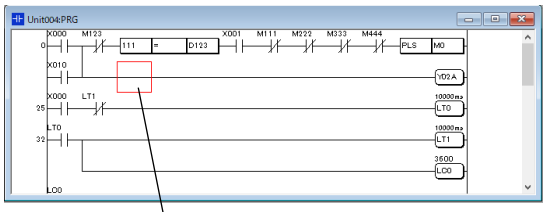
- Select <Delete rung> from the <Edit> menu.
- Press the **[Shift]** + **[Delete]** keys

The user can perform one of the following operations to paste a rung.
The paste operation varies depending on the whether the input format is in “Overwrite mode” or “Insert mode”.

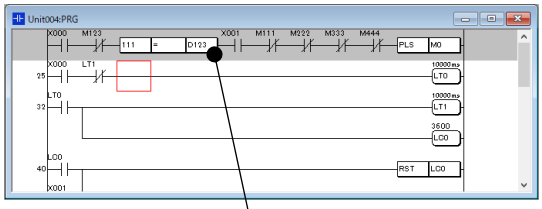
Operation procedure

Perform one of the following operations.

- Select <Paste line> from the <Edit> menu.
- Press the **[Ctrl]** + **[B]** keys

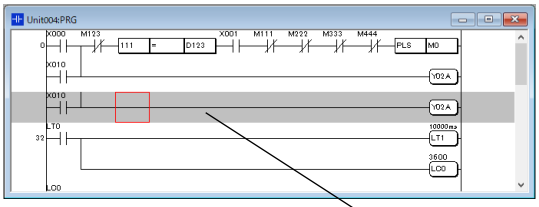


Copy rung



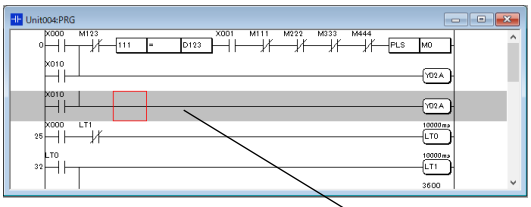
Cut rung (when cut)

When in overwrite mode



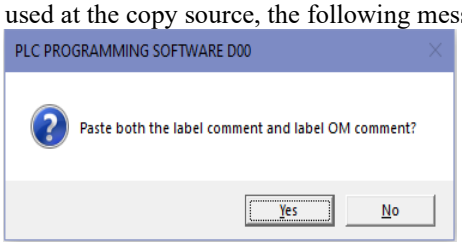
When pasting a rung, the copied or cut rung is pasted (overwriting) at the cursor position.

When in insert mode



When pasting a rung, a rung is inserted at the cursor position and the copied or cut commands are pasted.

When pasting a copied rung to another POU, if there is a label (L) with an OM comment being used at the copy source, the following message is displayed.



Select <Yes> to paste the label's OM comment to the copy destination.
Select <No> to not paste the label's OM comment.

Even if <Yes> is selected, the OM comment for a label that is not included in the rung (at the copy source) is not pasted. In addition, if the copy source rung has both labels with and without OM comments, when <Yes> is selected, only the OM comments that are set are pasted to the copy destination.

(NOTE) If an inline comment or coil remark is set at the copy source, then that inline comment or coil remark is also pasted to the copy destination.

7.2.3.3 Copy, cut and paste circuit blocks

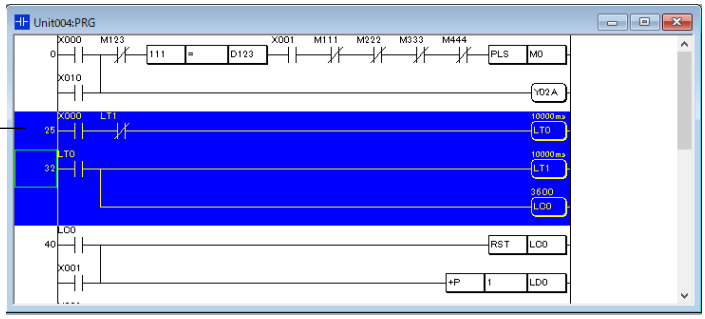
The user first selects the desired circuit block to perform a copy or cut operation, and then can paste it to the circuit block at the cursor position.

The user can use perform the following procedure to specify a range of circuit blocks to copy or cut. The specified range is highlighted in blue.

Operation procedure

1. Move the cursor to the step number position (left side of left rail) for the desired circuit block.
2. Press the **[Shift]** + **[↓]** keys to specify the circuit block(s) to copy or cut.
   Multiple and consecutive circuit blocks can be specified by pressing the **[Shift]** + **[↓]** keys repeatedly.

Initial cursor position



The user can perform one of the following operations to copy.

Operation procedure

Perform one of the following operations.

- Select <Copy> from the <Edit> menu.
- Press the **[Ctrl]** + **[C]** keys
- Click on the button in the tool bar.

The user can perform one of the following operations to cut.
After a cut operation, that location becomes blank.

Operation procedure

Perform one of the following operations.

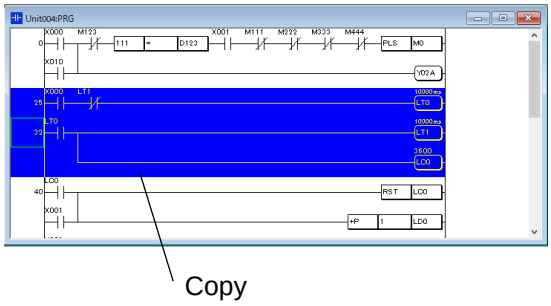
- Select <Cut> from the <Edit> menu.
- Press the **[Ctrl]** + **[X]** keys
- Click on the button in the tool bar.

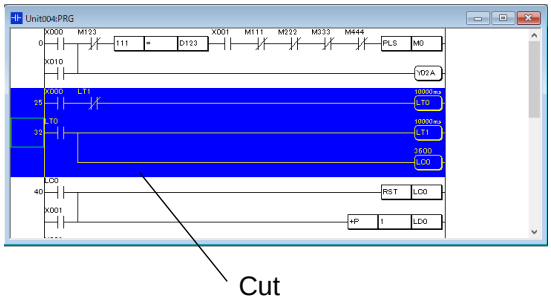
The user can perform one of the following operations to paste.
The paste operation is inserted at the cursor position regardless of whether the input format is in "Overwrite mode" or "Insert mode".

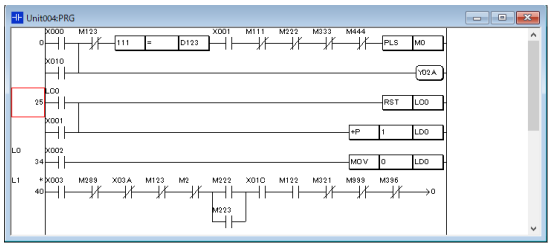
Operation procedure

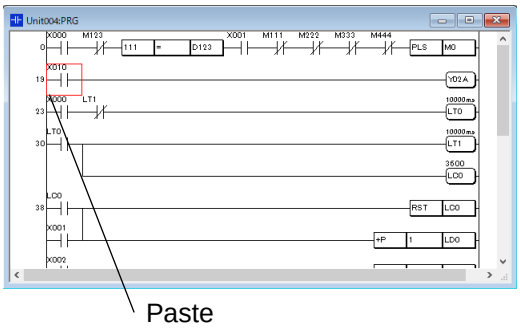
Perform one of the following operations.

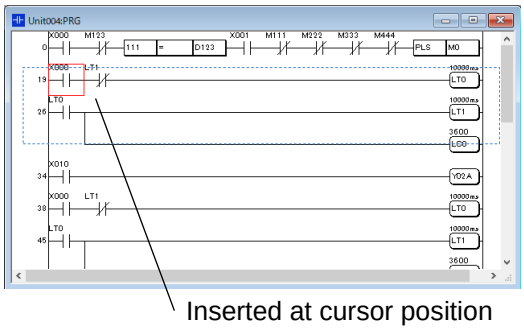
- Select <Paste> from the <Edit> menu
- Press the **[Ctrl]** + **[V]** keys
- Click on the button in the tool bar.



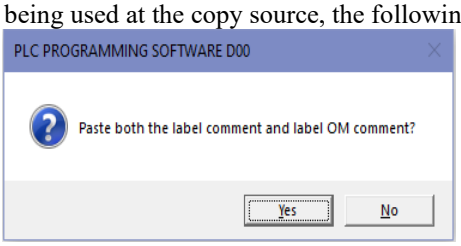








When pasting a copied circuit block to another POU, if there is a label (L) with an OM comment being used at the copy source, the following message is displayed.



Select <Yes> to paste the label's OM comment to the copy destination.
Select <No> to not paste the label's OM comment.

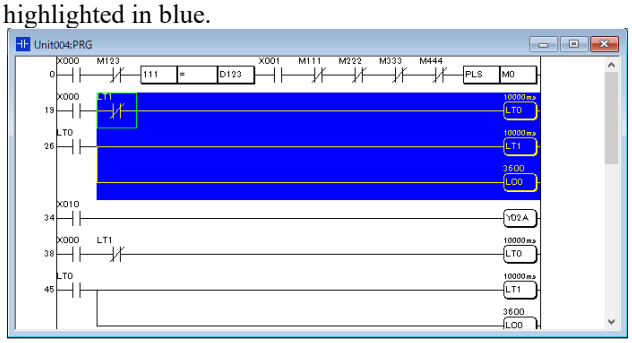
Even if <Yes> is selected, the OM comment for a label that is not included in the circuit block (at the copy source) is not pasted. In addition, if the copy source circuit block has both labels with and without OM comments, when <Yes> is selected, only the OM comments that are set are pasted to the copy destination.

(NOTE) If an inline comment or coil remark is set at the copy source, then that inline comment or coil remark is also pasted to the copy destination.

7.2.3.4 Specify range to copy, cut and paste

The user first selects the desired range of circuit blocks to perform a copy or cut operation, and then can paste that range to the cursor position.

The user can drag the mouse to specify the range of circuits to copy or cut. The specified range is highlighted in blue.



Operation procedure

The user can perform one of the following operations to copy.

- Select <Copy> from the <Edit> menu.
- Press the **[Ctrl]** + **[C]** keys
- Click on the button in the tool bar.

The user can perform one of the following operations to cut.

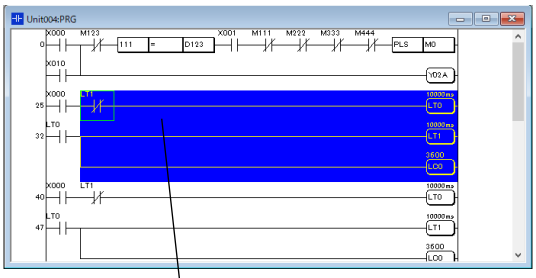
- Select <Cut> from the <Edit> menu.
- Press the **[Ctrl]** + **[X]** keys
- Click on the button in the tool bar.

After a cut operation, that location becomes blank.

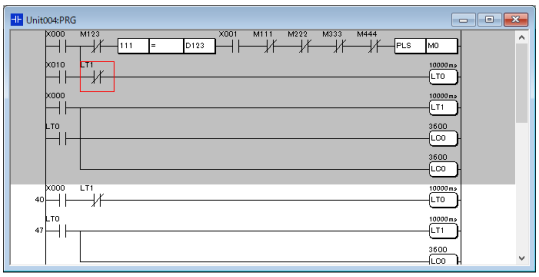
The user can perform one of the following operations to paste.

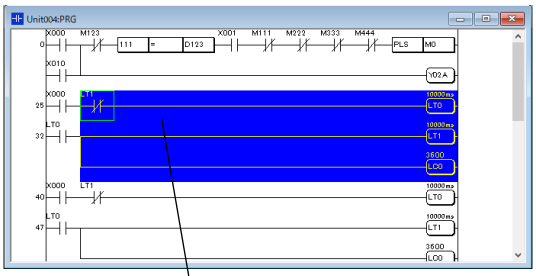
- Select <Paste> from the <Edit> menu.
- Press the **[Ctrl]** + **[V]** keys
- Click on the button in the tool bar.

The paste operation varies depending on the whether the input format is in "Overwrite mode" or "Insert mode".

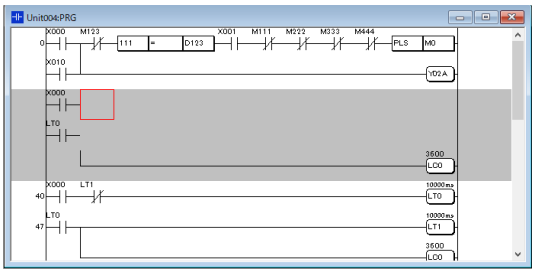


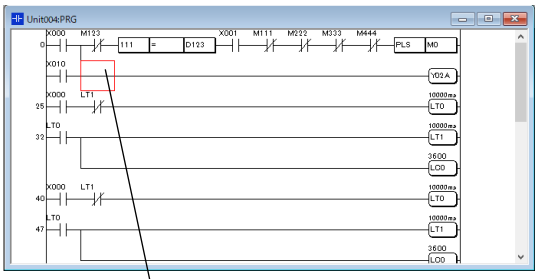
Copy





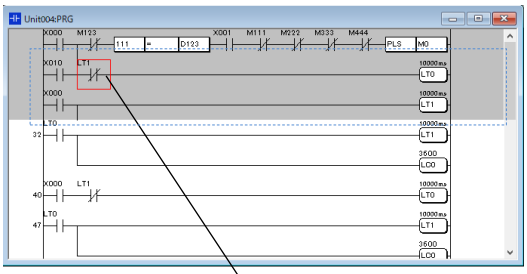
Cut





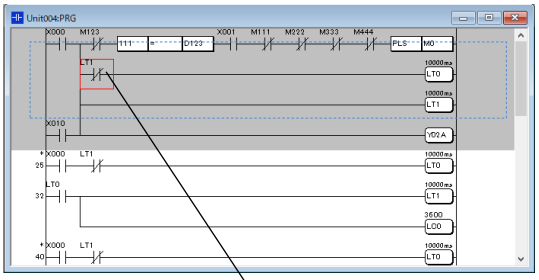
Paste

When in overwrite mode



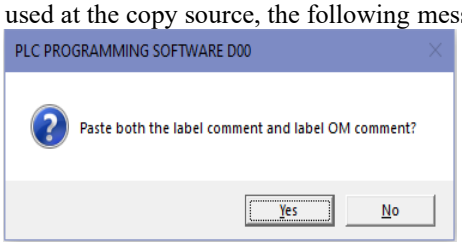
Overwrites circuit at cursor position

When in insert mode



Inserts circuit at cursor position

When pasting a copied section to another POU, if there is a label (L) with an OM comment being used at the copy source, the following message is displayed.



Select <Yes> to paste the label's OM comment to the copy destination.
Select <No> to not paste the label's OM comment.

Even if <Yes> is selected, the OM comment for a label that is not included in the command (at the copy source) is not pasted.

(NOTE) If a coil remark is set at the copy source, then that coil remark is also pasted to the copy destination.

7.2.3.5 Select all

When the select all operation is used, all commands in the LD program edit window are selected. The user can reuse the LD program in the LD program edit window to copy, cut or paste and create an LD program.

Operation procedure

Perform one of the following operations.

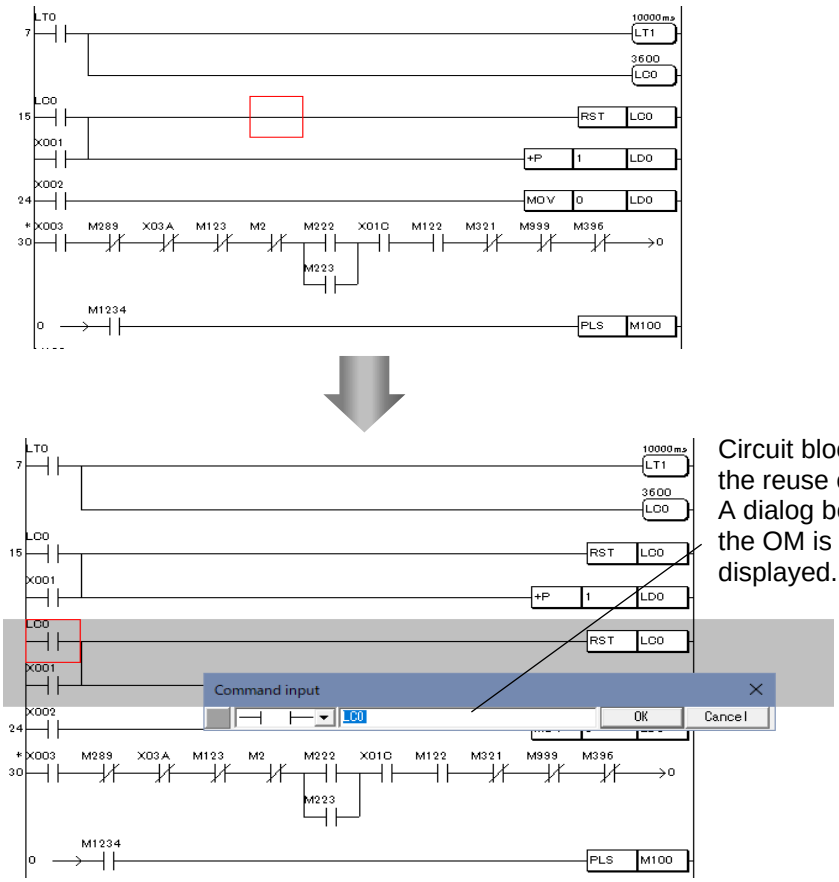
- Select <Select all> from the <Edit> menu.
- Press the **[Ctrl]** + **[A]** keys

The copy, cut and paste operations are the same as "7.2.3.3 Copy, cut and paste circuit blocks".

7.2.4 Reuse Circuit

The same circuit block as the one at the cursor position can be inserted below the circuit block at the cursor position.
This function is used when creating multiple circuit blocks with the same command.

(NOTE) The OM number attached to the command is also reused, therefore always edit the OM number for the circuit block that was reused.
If the OM number is not edited, then the circuit may perform an unintentional operation.



Circuit block inserted with the reuse operation.
A dialog box for changing the OM is automatically displayed.

Operation procedure

Move the cursor to the desired circuit block, and press the **[Ctrl]** + **[F10]** keys.

7.3 Replace All Contacts, OMs & IXs

This section describes how to replace contacts, OMs and indexes (IXs) being used in an LD program.

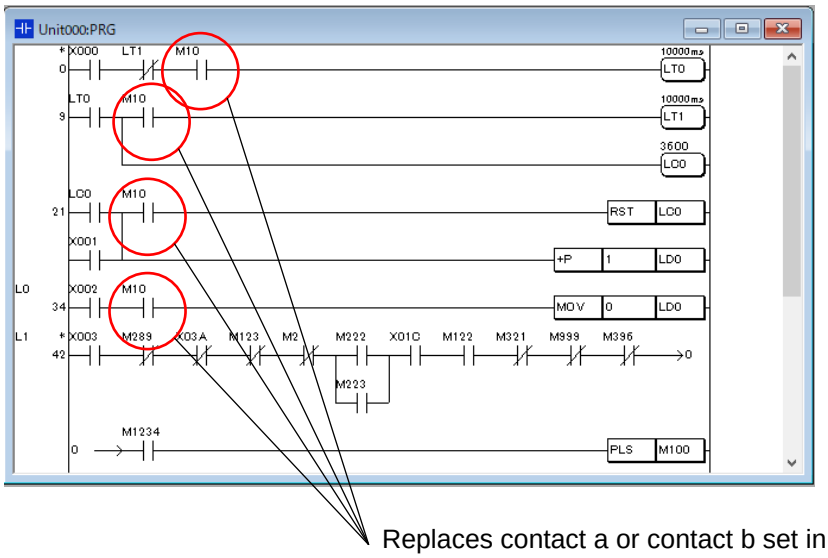
(NOTE 1) The replacement function can be used to edit an LD program after it has been converted.

(NOTE 2) After the replacement is completed, the design check is automatically executed. Refer to “5.7.2 Design check” for details on the design check.

7.3.1 Replace All Contacts a/b

Contact commands used in a specified OM are logically inverted.
A contact command used as an NO contact (contact *a*) is replaced with an NC contact (contact *b*). Or, a contact command used as an NC contact (contact *b*) is replaced with an NO contact (contact *a*).
The user can either replace the contact commands being used in an LD program all at once, or check and replace each contact command one at a time.

When “M10” is specified

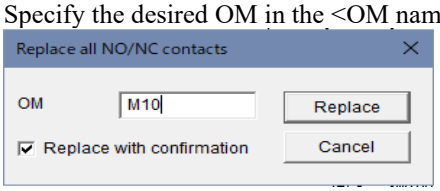


(NOTE) We recommend checking and replacing each one at a time to avoid replacing a contact accidentally or unintentionally.

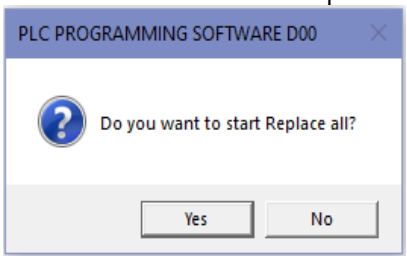
Operation procedure

1. Select <Replace all contacts a/b> under <Replace> in the <Edit> menu.

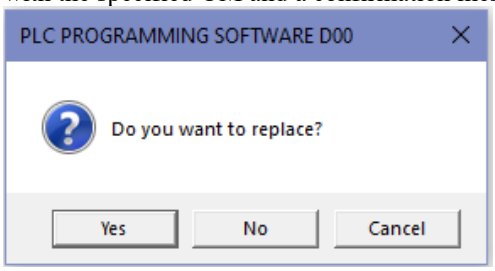
2. The <Replace all contacts a/b> dialog is displayed.
   Specify the desired OM in the <OM name> field.



3. After clicking on the <Replace> button, a confirmation message appears asking whether to start the replacement.
   Select <Yes> to execute the replacement operation.



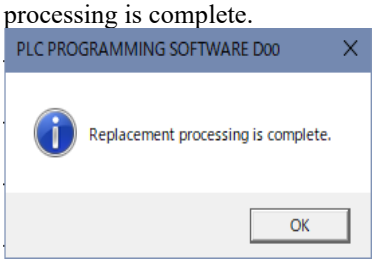
4. When the checkbox <Check and replace> is ticked, the cursor moves to contact command with the specified OM and a confirmation message appears asking whether or not to replace it.



   Select <Yes> to replace the contact command (at the cursor) and then search for the next one.
   Select <No> to skip replacing the contact command (at the cursor) and search for the next one.
   Select <Cancel> to cancel the replacement operation.

   When the checkbox <Check and replace> is not ticked, the confirmation message does not appear and it replaces all the contact commands used in the specified OM all at once.

5. When all the contact commands have been replaced, a message appears indicating the processing is complete.



6. After closing that message, the design check is automatically executed.
   The <Replace all contacts a/b> dialog automatically closes.

(NOTE 1) If <Cancel> is selected when the confirmation message appears (step 4 above), the contacts that had already been replaced will not go back to their previous state. In addition, the design for the ladder program will not be checked.
To change back the contacts that were replaced, use <Undo> in the <Edit> menu.

(NOTE 2) After clicking on the <Cancel> button in the <Replace all contacts a/b> dialog, the dialog closes without performing any replacements.

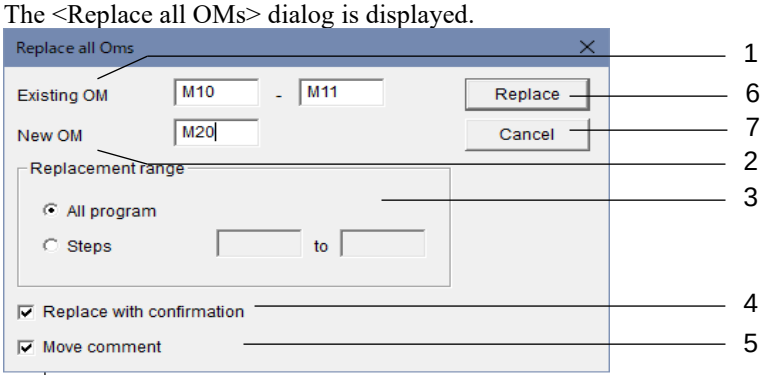
7.3.2 Replace All OMs

This function replaces an OM being used in an LD program with another OM.
The user can either replace the OMs being used in an LD program all at once, or check and replace each command one at a time where the OM is used.
In addition, the user can specify a range in the LD program and can also transfer the original comment data (that is set to the source OM) when replacing OMs.

(NOTE 1) When replacing OMs, the user can only replace bit OMs with bit OMs and only replace word OMs with word OMs.
(NOTE 2) When "X" is specified for the source OM, "X" can only be set for the target OM.
(NOTE 3) When the user clicks on the <Replace> button and the OM number setting exceeds the upper limit after the replacement, then the error message <<Not possible to replace OM number because new OM number will be out of range.>> is displayed.
(NOTE 4) When a comment is moved or transferred, the OM comment (source) overwrites the target OM comment and then the source OM comment is deleted.
If the same OM is used at multiple locations and one of those OMs is replaced, then that OM comment is also transferred with the data.
(NOTE 5) After the replacement is completed, the design check is automatically executed.
(NOTE 6) If there is a circuit block that has not been converted, replacement cannot be executed.

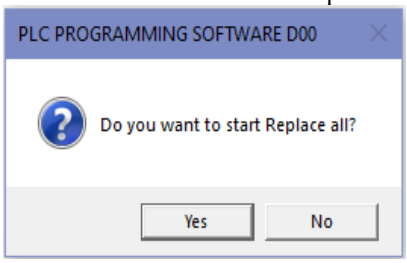
Operation procedure

1. Select <Replace all OMs> under <Replace> in the <Edit> menu.
2. The <Replace all OMs> dialog is displayed.

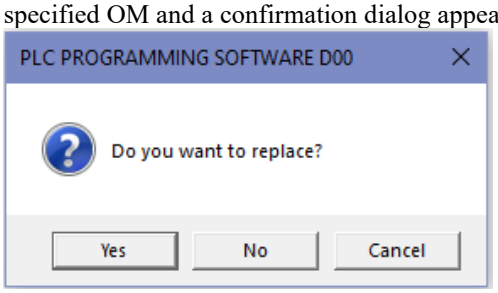


No.	Item	Description
1	OM name for change source	Specifies the first and the last OMs in the source.
2	OM name for change destination	Specifies the first OM in the destination. The range for replacing OMs is specified in the "OM name for change source".
3	Replacement range	Selects and specifies the replacement range. "POU at forefront": Specifies the replacement range for the LD program at the forefront. "Specify step range": Specifies the first and last step numbers in the range. This setting is valid for POUs that have been converted.
4	Check and replace	Selects whether to replace all OMs at once or to check and replace each one at a time. When the checkbox is ticked, it will check and replace each OM one at a time.
5	Move comment	Selects whether or not to move the comment from the source OM to the destination when replacing the OM. Tick the checkbox to move the OM comment.
6	<Replace> button	Starts the replacement.
7	<Cancel> button	Closes the dialog.

3. After clicking on the <Replace> button, a confirmation dialog appears asking whether to start the replacement.
   Select <Yes> to execute the replacement operation.



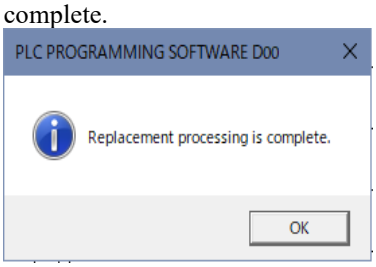
4. When the checkbox <Check and replace> is ticked, the cursor moves to the command with the specified OM and a confirmation dialog appears asking whether or not to replace the OM.



   Select <Yes> to replace the OM (at the cursor) and then search for the next OM.
   Select <No> to skip replacing the OM (at the cursor) and search for the next OM.
   Select <Cancel> to cancel the OM replacement operation.

   When the checkbox <Check and replace> is not ticked, the confirmation message does not appear and it replaces the specified OMs all at once.

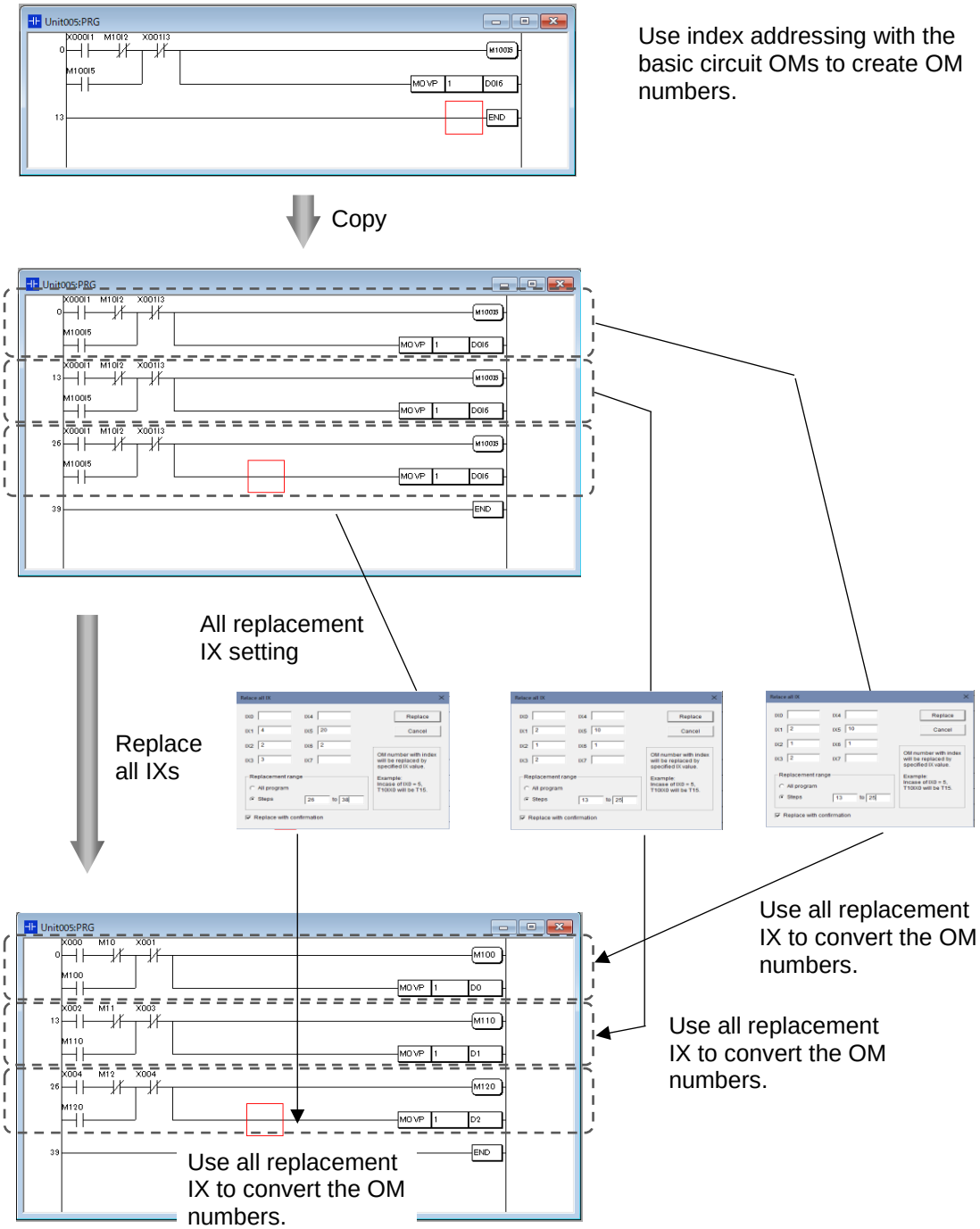
5. When all the specified OMs have been replaced, a dialog appears indicating the processing is complete.



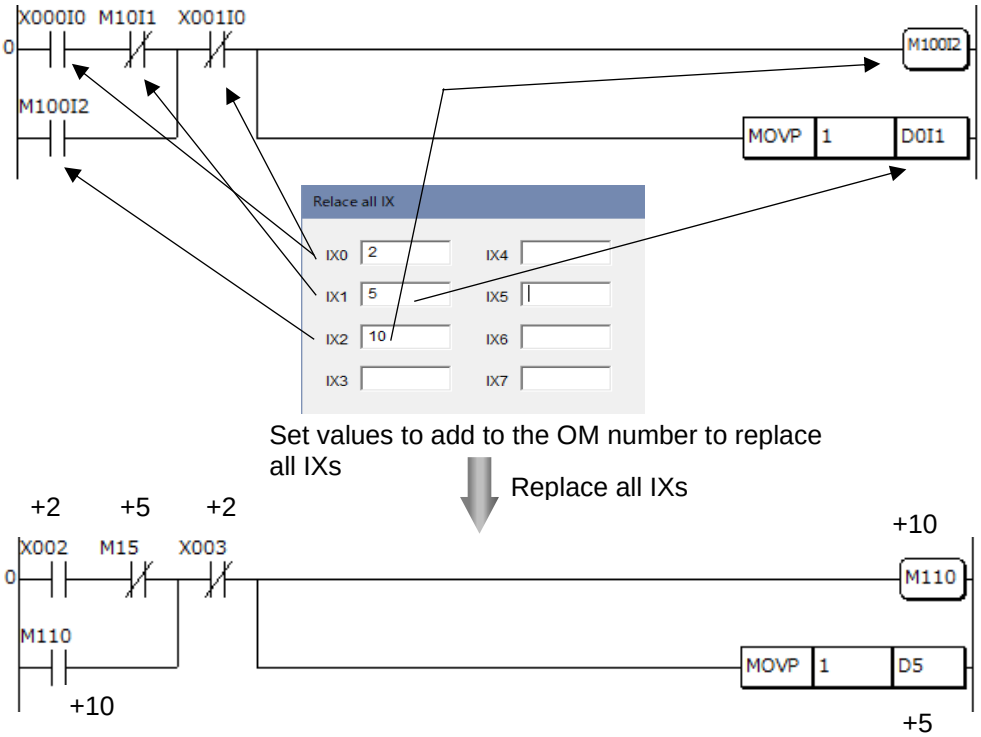
6. Click on the <Cancel> button to close the <Replace all OMs> dialog.
   After the <Replace all OMs> dialog closes, the design check is automatically executed.

7.3.3 Replace All IXs

The replace all IX function is used to create multiple LD programs that have the same structure but different OM numbers.
The indexed OM number is changed by adding a specified value to it.



If the added value is the same, the same index can be used. A maximum of 8 different settings are available to add a value.



The user can either replace the indexes being used in an LD program all at once, or check and replace each OM one at a time where the index is used.
In addition, the user can specify a range in the LD program for replacement.

(NOTE 1) Pay particular attention when performing a replacement operation to ensure the wrong OM number is not set.
We recommend checking and replacing each one at a time to avoid replacing a contact accidentally or unintentionally.

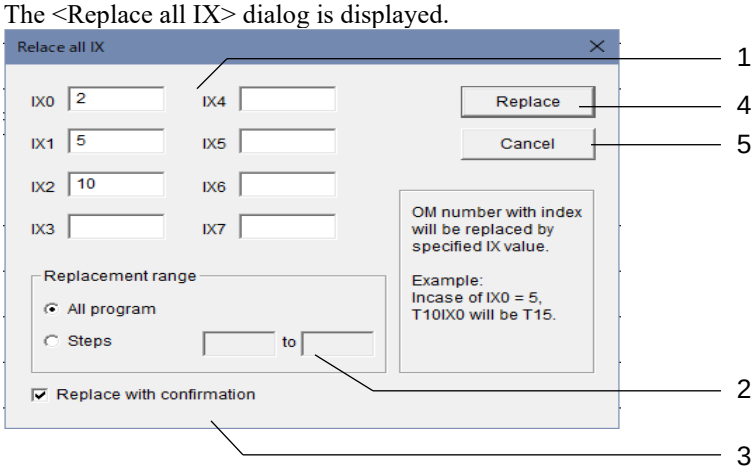
(NOTE 2) When the user clicks on the <Replace> button and the OM number setting exceeds the upper limit after the replacement, then the error message <<Not possible to replace OM number because new OM number will be out of range.>> is displayed.

(NOTE 3) After the replacement is completed, the design check is automatically executed.

(NOTE 4) If there is a circuit block that has not been converted, replacement cannot be executed.

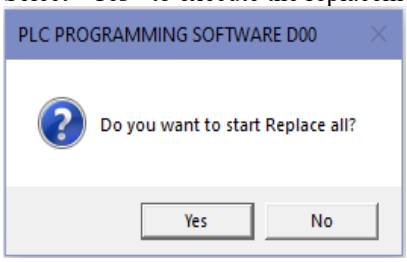
Operation procedure

1. Select <Replace all IX> under <Replace> in the <Edit> menu.
2. The <Replace all IX> dialog is displayed.

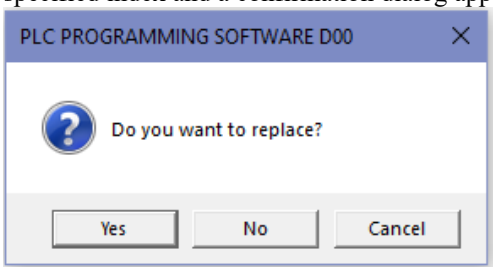


No.	Item	Description
1	IX0~IX7	Sets the added value for the index addressing.
2	Replacement range	Selects and specifies the replacement range. "POU at forefront": Specifies the replacement range for the LD program at the forefront. "Specify step range": Specifies the first and last step numbers in the range. This setting is valid for POUs that have been converted.
3	Check and replace	Selects whether to replace all OMs at once or to check and replace each one at a time. When the checkbox is ticked, it will check and replace each index one at a time.
4	<Replace> button	Starts the replacement.
5	<Cancel> button	Closes the dialog.

3. After clicking on the <Replace> button, a confirmation dialog appears asking whether to start the replacement.
   Select <Yes> to execute the replacement operation.



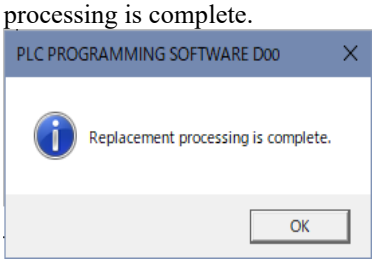
4. When the checkbox <Check and replace> is ticked, the cursor moves to the command with the specified index and a confirmation dialog appears asking whether or not to replace the index.



Select <Yes> to replace the index (at the cursor) and then search for the next index.
Select <No> to skip replacing the index (at the cursor) and search for the next index.
Select <Cancel> to cancel the IX replacement operation.

When the checkbox <Check and replace> is not ticked, the confirmation message does not appear and it replaces the specified OMs all at once.

5. When all the contact commands have been replaced, a dialog appears indicating the processing is complete.



6. After closing that message, the design check is automatically executed.
The <Replace all IX> dialog automatically closes.

(NOTE) If <Cancel> is selected when the confirmation message appears (step 4 above), the OMs that had already been replaced will not go back to their previous state. In addition, the design for the ladder program will not be checked.
To change back the OMs that were replaced, use <Undo> in the <Edit> menu.

7.4 Create Comments

This section describes how to create comments for an OM and how to create comments for an LD program.

7.4.1 Create OM Comment

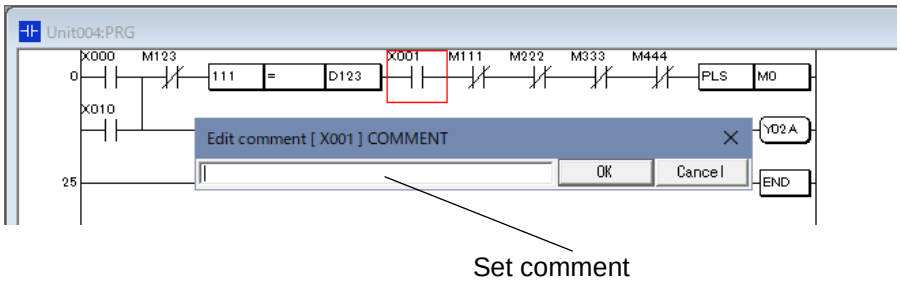
A comment for a specific use can be created for each OM.
This section describes how to create a comment for an OM (at the cursor position) in the LD program.
The user can create comments in an OM list. Refer to “7.4.2 Edit OM comment list” to create a comment in the OM list.
In addition, refer to “7.1.5.1 Change OM comment display” for details on changing the OM comment display.

OM comments have the following character limits: Maximum of 32 characters when using half-width character and 16 character when using full-width characters.
The user can switch between two different comment displays in the LD program: 16-character and 32-character display for half-width characters (8-character and 16-character display for full-width characters).

The user can create OM comments using the following procedure.
<u>Operation procedure</u>
Move the cursor to the desired OM to create a comment.

Perform one of the following operations to display the comment setting dialog.
- Press the **[Shift]** + **[F9]** keys
- Press the **[Ctrl]** + **[Shift]** + **[F5]** keys.
- Select <Create OM comment> under <Create drawing> in the <Edit> menu.

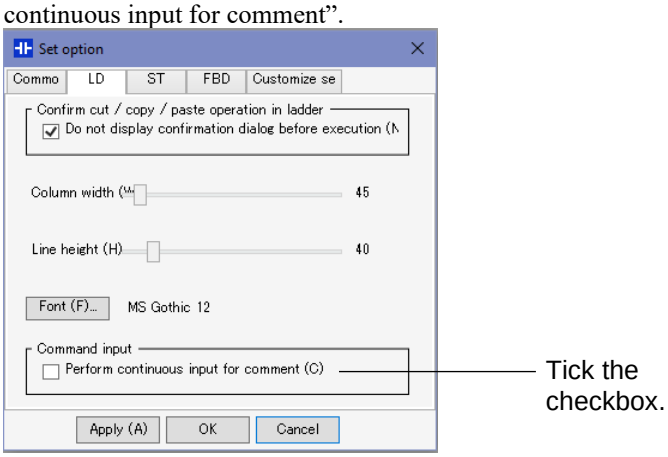
Enter a comment, and then click on the <OK> button.
Click on the <Cancel> button to cancel input.



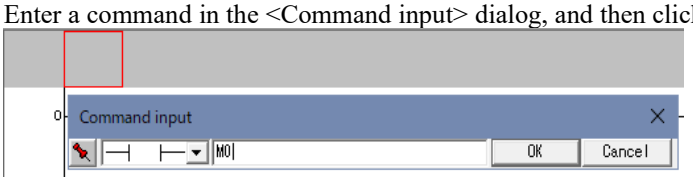
While editing the LD program, the user can enter a command and then enter a comment for the OM being used in that command. (excluding CALC commands)

Operation procedure

1. Select <Option settings> from the <Auxiliary> menu in the PLC Programming software D00.
2. Open the <LD> tab in the <Option settings> dialog, and tick the checkbox for "Perform continuous input for comment".



3. After clicking on the <OK> button, the <Option settings> dialog closes.
4. Open the LD program in the PLC Programming software D00 LD/ST Programming Editor.
5. Enter a command in the <Command input> dialog, and then click on the <OK> button.

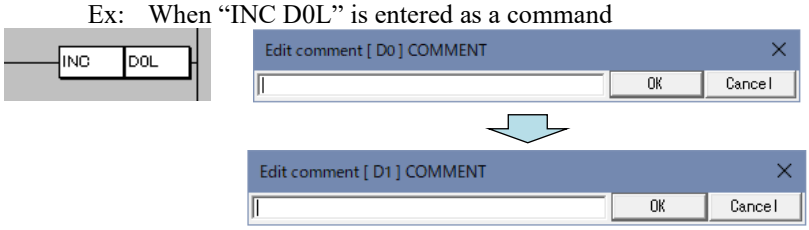


6. When the <Command input> dialog automatically opens, enter a comment for the OM being used in the command (entered in step 5) and click on the <OK> button.

When there are multiple commands being used in one command, then another <Command input> dialog opens up.

When a digit-specified OM has been entered in the <Command input> dialog, then the user is prompted to enter an OM comment for all the OMs that are included.

Ex: When "INC D0L" is entered as a command



After closing the <Command input> dialog for D0, then a <Command input> dialog for D1 is displayed.

In the <Command input> dialog, if the user enters an OM (IX or N) where an OM comment cannot be set, then the user is not prompted to enter a comment for that OM.

Ex: When "INC D0I0" is entered as a command

After closing the <Command input> dialog for D0, there is no <Command input> dialog that is displayed for IX0.

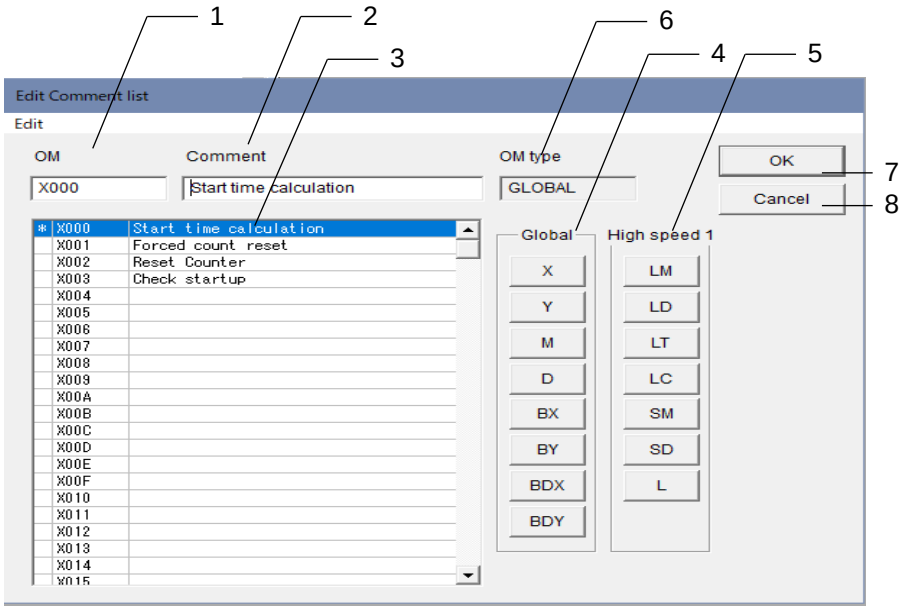
(NOTE) Even when the button is ON in the <Command input> dialog, the continuous input function (for a comment) can be used.

7.4.2 Edit OM Comment List

A comment for a specific use can be created for each OM.
This section describes how to create a comment using the OM list.
The comments can be created even in the LD program. Refer to “7.4.1 Create OM comment” to create a comment in the LD program.
In addition, refer to “7.1.5.1 Change OM comment display” for details on changing the OM comment display.

OM comments have the following character limits: Maximum of 32 characters when using half-width character and 16 character when using full-width characters.
The user can switch between two different comment displays in the LD program: 16-character and 32-character display for half-width characters (8-character and 16-character display for full-width characters).
When creating comments in the OM list, the user can copy, cut and paste multiple comments.

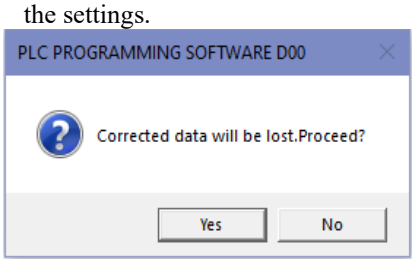
The user can create OM comments using the following procedure.
Operation procedure
1. Select <Edit OM comment> under <Create drawing> in the <Edit> menu.



No.	Item	Description
1	OM	Enter an OM. After clicking on an OM in the OM list, that OM name (clicked position) is displayed.
2	Comment	Sets a comment. After clicking on an OM in the OM list, that comment (clicked position) is displayed.
3	OM list	Displays a list of OMs.
4	Global OM selection	Selects a global OM. After pressing an OM button, a list of OMs is displayed in the OM list display area.
5	Local OM selection	Selects a local OM for the program in the active window. After pressing an OM button, a list of OMs is displayed in the OM list display area.
6	OM type	Displays the type of OM that is selected. One of the following is displayed. “Global”, “High speed 1” to “High speed 10”, or “Standard 1” to “Standard 10”.
7	<OK> button	Applies the setting, and the dialog closes.
8	<Cancel> button	Discards the settings, and the dialog closes.

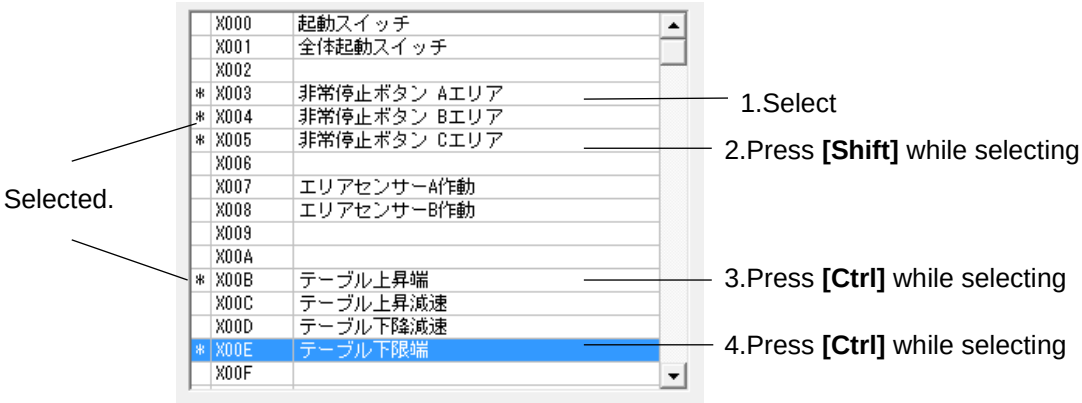
2. Select an OM to create a comment in the “OM name” field or “Comment display” field.
3. Enter a comment in the “Comment” field. The comment entered in the “Comment display” field is displayed.
4. Click on the <OK> button.

(NOTE) When the user clicks on the <Cancel> button after editing in the <Edit OM comment> dialog, a confirmation message appears asking whether or not to discard the settings.

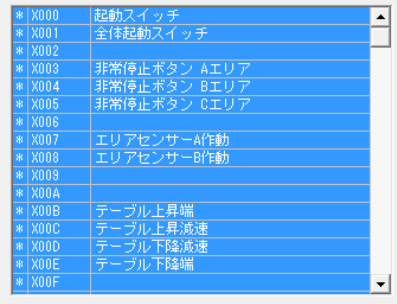


When <Yes> is selected, the settings are discarded and the <Edit OM comment> dialog closes.
When <No> is selected, it goes back to the <Edit OM comment> dialog.

- Specify comment to copy or cut
  Select a comment in the “Comment display” field to copy or cut.
  An asterisk (*) is displayed at the beginning of the comment line.
  To select multiple comments, hold down the **[Shift]** key or the **[Ctrl]** key while selecting.

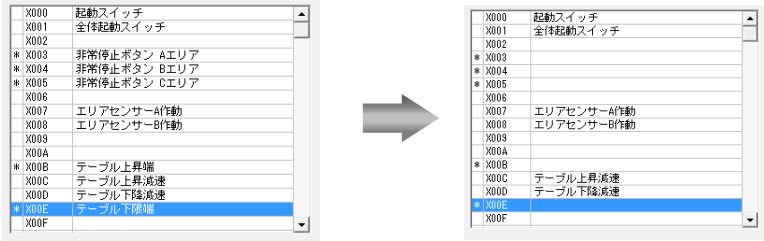


In addition, the user can use one of the following operations to specify a comment for a range of OMs by type.
- Select <Select all> from the <Edit> menu in the <Edit OM comment> dialog.
- Press the **[Ctrl]** + **[A]** keys

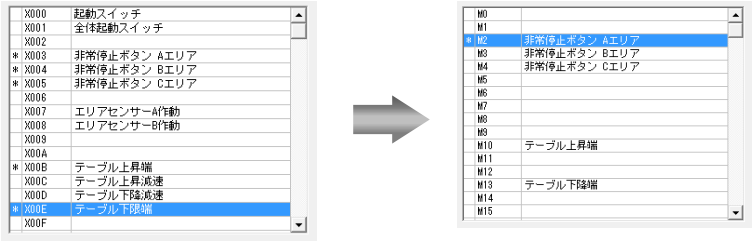


- Copy comment
  <u>Operation procedure</u>
  Select a comment in the “Comment display” field to copy.
  Select <Copy> from the <Edit> menu in the <Edit OM comment> dialog.
  Or, press the **[Ctrl]** + **[C]** keys to perform the copy operation.
  A message appears indicating that the copy operation is complete.

- Cut comment
  Operation procedure
  Select a comment in the “Comment display” field to cut.
  Select <Cut> from the <Edit> menu in the <Edit OM comment> dialog.
  Or, press the **[Ctrl]** + **[X]** keys to perform the copy operation.
  A message appears indicating that the cut operation is complete.

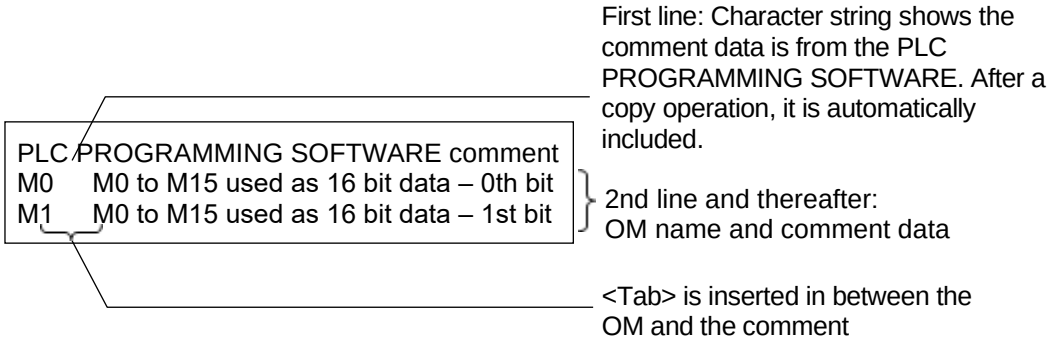


- Paste comment
  Operation procedure
  1. Select a comment in the “Comment display” field to paste.
     When multiple comments are selected, the first comment to paste is selected.
  2. Select <Paste> from the <Edit> menu in the <Edit OM comment> dialog.
     Or, press the **[Ctrl]** + **[V]** keys to perform the copy operation.
  3. A message appears indicating that the paste operation is complete.



The user can take comment data copied in the edit OM comment dialog and paste it to another OM. In addition, the user can edit that data in a text editor (commercially available) or similar tool and then re-paste the data into the edit OM comment dialog.

The following shows the format of the comment data that is copied from the edit OM comment dialog.



When editing comments in a text editor (commercially available) or similar tool and then pasting that data into the edit OM comment dialog, be sure to use the same format as above.

When pasting the comment data, be sure to copy and include the first line data as well. If it is not copied and included, a format error will trigger when pasting it.

The comment data can also be pasted to another OM apart from the copy source.
Ex: After copying comments from X10 to X12 and selecting Y22 to paste there, the comments are pasted as follows.
X10 comment is pasted to the Y22 comment.
X11 comment is pasted to the Y23 comment.
X12 comment is pasted to the Y24 comment.

If the OM numbers for the comments are random and not sequential, then those intervals are maintained and pasted accordingly.

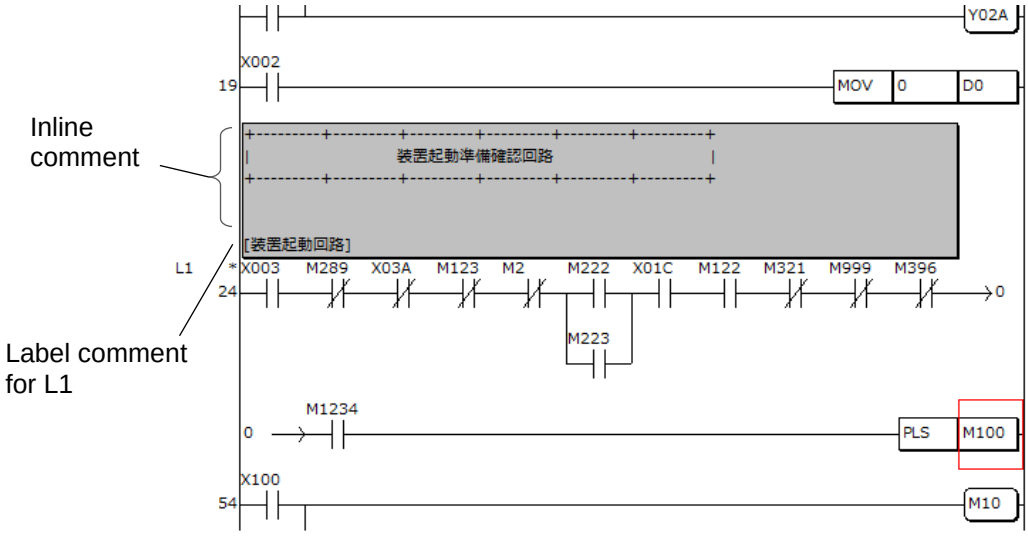
Ex: After copying comments X10 and X12 and selecting Y22 to paste there, the comments are pasted as follows.
X10 comment is pasted to the Y22 comment.
There is no change to the Y23 comment.
X12 comment is pasted to the Y24 comment.

If the pasted comment data has a problem with the format somewhere in the data, the valid comment data is pasted, but any data after the problem point is not pasted.

7.4.3 Create Inline Comment

Inline comments are used to help improve the readability of the LD program by enabling the user to write titles and explanations per circuit block or at divisions in the program.

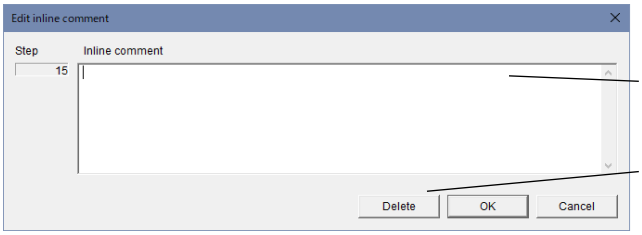
A maximum of 64 half-width characters (32 full-width characters) per line with a total of 5 lines can be used for an inline comment.
Refer to “7.1.5.2 Change inline comment / label comment display” for details on changing the inline comment display. In addition, when there is a label (L) set to a circuit block with an inline comment and a label comment is also created, the inline comment and the label comment are displayed in the same field.
Refer to “7.4.4 Create label comment” for further details on label comments.



When an inline comment is created, an asterisk (*) is displayed at the beginning of the circuit block.

Operation procedure

1. Move the cursor to the beginning of the desired circuit block to create an inline comment.
2. Perform one of the following operations.
   - Select <Create inline comment> under <Create drawing> in the <Edit> menu.
   - Press the **[Ctrl]** + **[Shift]** + **[F7]** keys.
   - Double-click on the inline comment display area while the inline comment is displayed.
3. Create an inline comment.

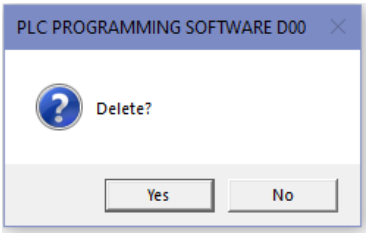


4. Click on the <OK> button.

(NOTE 1) If a circuit block with an inline comment is copied or cut and then the circuit block is pasted, then the inline comment is also pasted together with it.

(NOTE 2) After clicking on the <Cancel> button in the <Create inline comment> dialog, any setting changes are discarded and the dialog closes.

(NOTE 3) After clicking on the <Delete> button in the <Create inline comment> dialog, a confirmation message appears asking whether or not to delete.



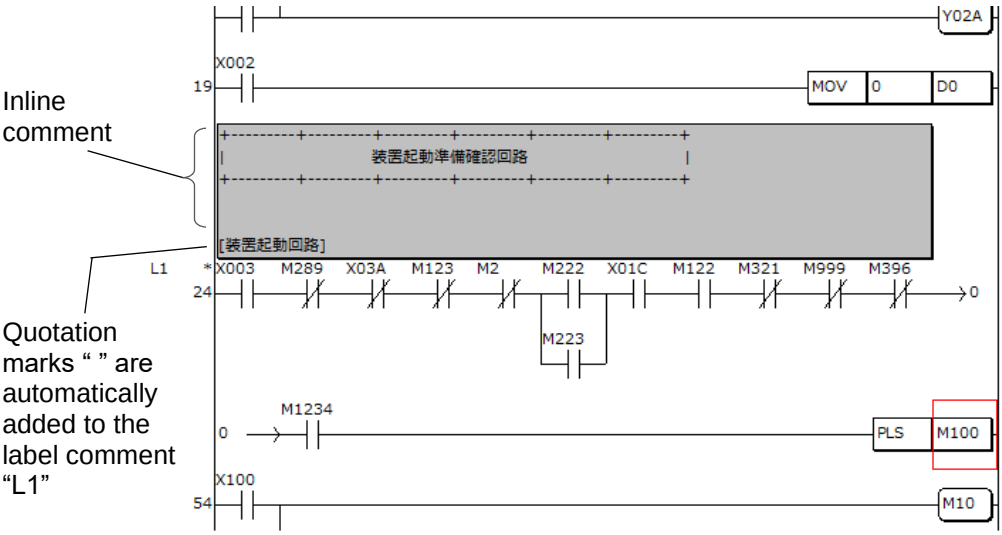
Select <Yes> to delete the inline comment.
Select <No> to go back to the <Create inline comment> dialog.

(NOTE 4) Even when only a label comment is displayed in the inline comment display area, if the user double-clicks on that area, the <Create inline comment> dialog is displayed.

7.4.4 Create Label Comment

Label (L) comments are not only created for OMs but they also play a role in the circuit indexing. They are used to help improve the readability of the LD program as well as make it easier to search circuits.

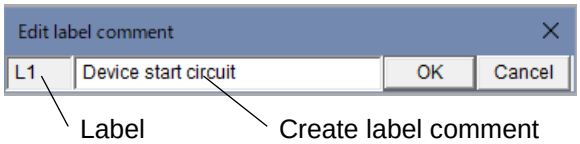
The user can only write one line for the label comment with a maximum of 64 half-width characters (32 full-width characters).
A label comment is displayed in the same field as an inline comment.
The user can change the inline comment display to either show or hide the label comments. Refer to “7.1.5.2 Change inline comment / label comment display” for details on changing the inline comment display.
Refer to “7.4.3 Create inline comment” for further details on inline comments.



When a label comment is created, an asterisk (*) is displayed at the beginning of the circuit block.

Operation procedure

1. Move the cursor to the desired label to create a label comment.
2. Select <Create label comment> under <Create drawing> in the <Edit> menu.
3. Enter a label comment, and then click on the <OK> button.
   Click on the <Cancel> button to cancel input.



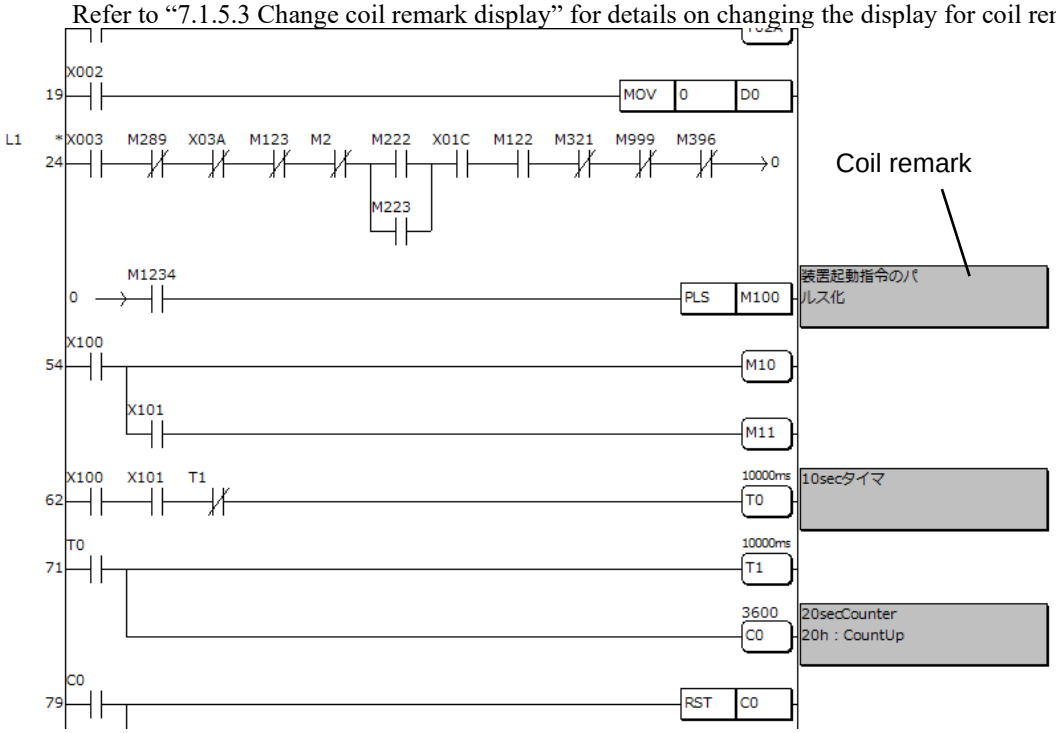
To delete a label comment, display the <Edit label comment> dialog and then delete all the characters for the label comment.

7.4.5 Create Coil Remark

The user can create comments with a specific purpose such as ON/OFF conditions for output commands and application commands (besides comparison operations).
Coil remarks can be created in circuit blocks that have been converted.

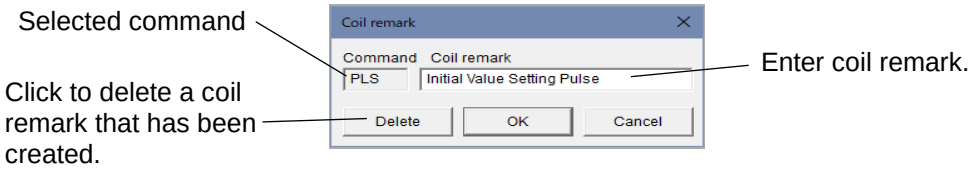
Coil remarks have a maximum of 32 half-width characters or 16 full-width characters.

Refer to “7.1.5.3 Change coil remark display” for details on changing the display for coil remarks.



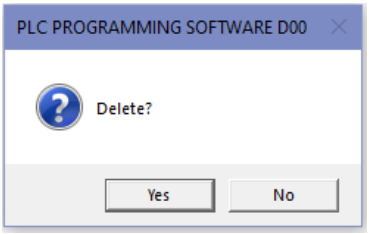
Operation procedure

1. Move the cursor to the desired command to create a coil remark.
2. Perform one of the following operations.
   - Select <Create coil remark> under <Create drawing> in the <Edit> menu.
   - Press the **[Ctrl]** + **[Shift]** + **[F8]** keys.
3. Enter a coil remark, and then click on the <OK> button.



(NOTE 1) If a command with a coil remark is copied or cut and then the command is pasted, that coil remark is also pasted together with it.

(NOTE 2) After clicking on the <Delete> button in the <Edit coil remark> dialog, a confirmation message appears asking whether or not to delete it.



Select <Yes> to delete the coil remark.
Select <No> to go back to the <Edit coil remark> dialog.

7.5 Edit Lock

This section describes the operations to prohibit or lock editing in the LD program in order to protect the program from an accidental or inadvertent input operation.

The edit lock is a function to prevent an inadvertent operation from writing data to the LD program and specifically prohibits the following edit operations.

- Edit LD program
- Change timer/counter list
- Edit OM comment
- Edit inline comment
- Edit label comment
- Edit coil remark
- Conversion

The user can unlock the editing by performing the edit lock operation (while the edit lock is on), allowing the user to edit the data.

When a key is pressed while the edit lock is on, the search dialog is displayed and the user can perform a “Search”.

- Letter key: <Find by OM> dialog
  [+], **[-]**, **[*]** and **[/]** keys
- **[0]** to **[9]** keys: <Find by step number> dialog

Refer to “7.6 Search” for details on the search function.

When opening a new window and displaying multiple windows of the same LD program, if edit lock is enabled on one of the windows, then the edit lock applies to all of the opened windows for the same LD program.
When the program edit window is closed while the edit lock is on, the edit lock will be released (unlocked) the next time the same POU edit window is opened. However, if the edit window is opened using the function “Open new window”, then the edit lock will be in the same status as the edit window that is already opened for the same LD program.

The “Edit lock/edit unlock” can be performed using one of the following operations.

- Select <Edit lock> from the <Edit> menu
- Press the **[F11]** keys
- Click on the button in the tool bar.

7.6 Search

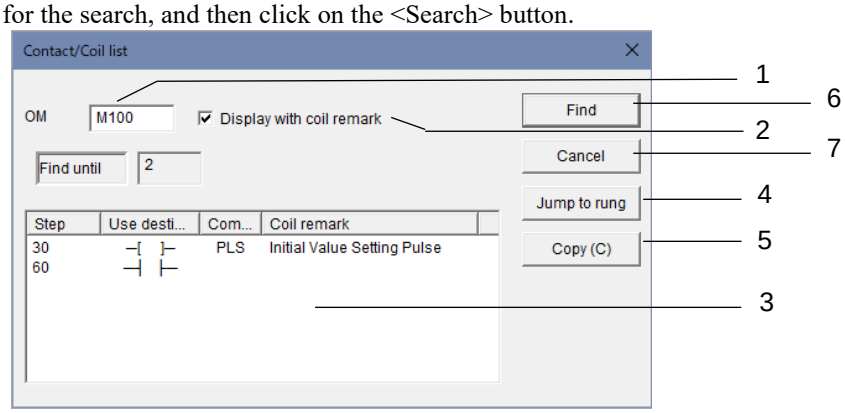
This section describes the operations on how to check the OM status and how to search for step numbers, OMs and commands.

7.6.1 Contact/Coil List

The contact/coil list can be used to conduct a search to check where and how an OM is being used in the LD program.

Operation procedure

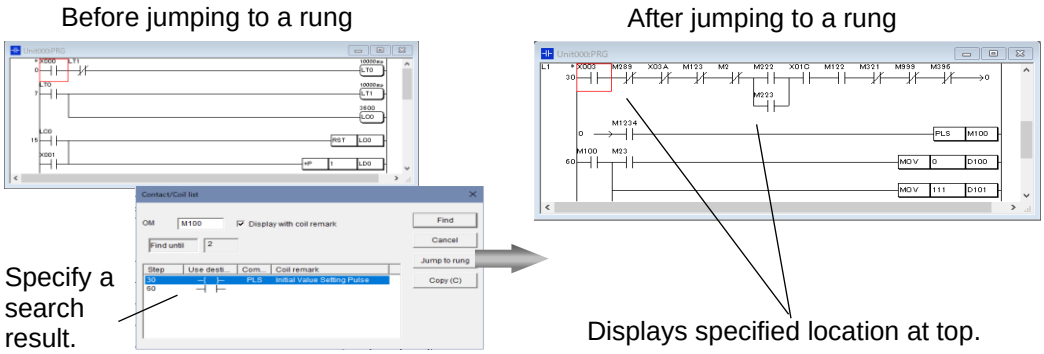
1. Perform one of the following operations.
   - Select <Contact/Coil list> under <Find> in the <Edit> menu.
   - Press the **[Shift]** + **[F2]** keys
2. Set the OM for the search, and then click on the <Search> button.



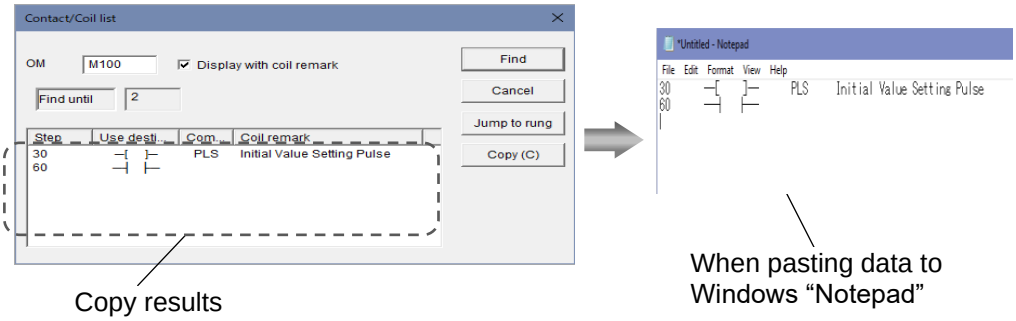
No.	Item	Description
1	OM name	Specifies the OM in the search.
2	Show coil remarks	Tick the checkbox to show the coil remarks in the search results.
3	List of search results	Displays the search results in a list. <Step> : Displays the first step in the circuit block where the specified OM is used. <Use destination> : Displays the command symbol where the specified OM is used. <Command> : Displays the command where the specified OM is used. <Coil remark> : Displays the coil remark that is set to the command when the checkbox "Display with coil remark" is ticked.
4	<Jump to rung> button	After selecting a line (rung) from the list of search results and clicking on <Jump to rung>, the cursor moves or jumps to the circuit block for the displayed step number.
5	<Copy> button	Copies the search results to the clipboard. This function can be used to paste the search data into a word processing program or a spreadsheet program.
6	<Search> button	Starts the search.
7	<Cancel> button	Closes the dialog.

3. Click on the <Cancel> button to close and exit the search.

- Jump to rung
  "Jump to rung" refers to when the user specifies a specific search result and then clicks on the <Jump to rung> button. Then, this function displays the specified circuit block at the top of the LD program edit window.



- Copy
  The search results can be copied as text to the Windows clipbboard. The user can paste the copied search results to another application like a commercially available text editor software.



Refer to "10.4.1 Contact/Coil List (LD Language)" for details on the "Contact/Coil list".

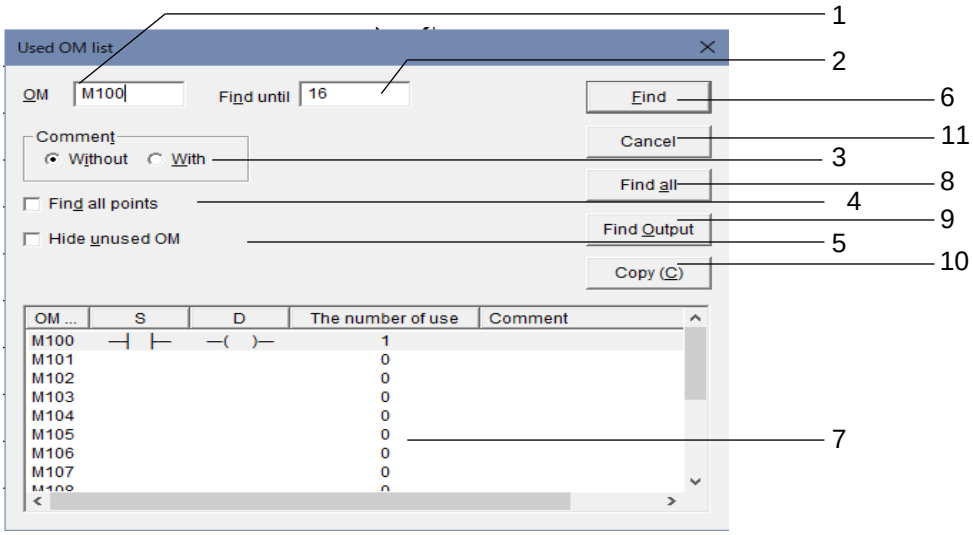
7.6.2 OM List

The OM list can be used to conduct a search to check if an OM is being used or not in the LD program by specifying the search range from a certain OM.

<u>Operation procedure</u>

1. Perform one of the following operations.
   - Select <Used OM list> under <Find> in the <Edit> menu.
   - Press the **[Shift]** + **[F4]** keys

2. Set the OM and search range in the <Used OM list> dialog.



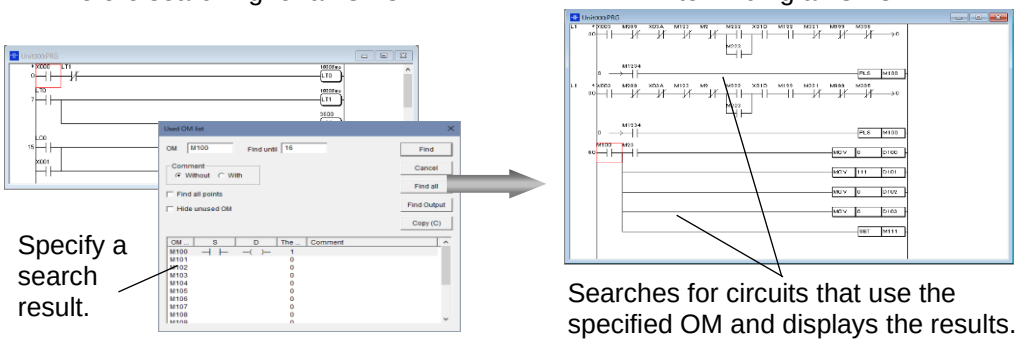
No.	Item	Description
1	OM name	Specifies the OM in the search.
2	Search number	Specifies the search range from the set OM.
3	Comment	Select “With” to display the OM comments in the search results.
4	Search all points	Tick the checkbox to search all locations from the specified OM.
5	Hide unused OMs	Tick the checkbox to display the search results excluding the unused OMs.
6	<Search> button	Starts the search.
7	List of search results	Displays the search results in a list. <OM name> : Displays the OMs that are found. <S> : Displays the “Contact” symbol when it is used as the source for the specified contact, data comparison command or application command. <D> : Displays the “Coil” symbol when the specified OM is being used as the “Destination” for an output command or an application command. <Qty. used> : Displays how many times the specified OM is used in the output of the LD program and in the CALC commands. When the OM is only being used as either a source for a contact/application command or a destination for an output/application command, then an “E” is displayed after the quantity number. <Comment> : Displays the comment that is set to the OM when “Comment” is set to “With”.
8	<Find all> button	Click on the button to close the dialog and display all the circuits where the OM is used.
9	<Find Output> button	Click on the button to close the dialog and display all the circuits where the selected OM is used as an output.
10	<Copy> button	Copies the character string that is displayed in the “Search results list”.
11	<Cancel> button	Ends and exits the search, and the dialog closes.

3. Set the range of the search in the “Search number” field, or tick the checkbox “Search all points” to search all the points after the specified OM.
4. Select whether to show or hide the OM comments.
5. Tick the checkbox “Hide unused OMs” to display only the OMs that are used.
6. Click on the <Search> button.
7. Click on the <Cancel> button to close and exit the search.

- Find all
  Specify an OM in the search results and click on the <Find all> button, and this function shows all the circuits where the specified OM is used in the search results display.

Before searching for all OMs

After finding all OMs

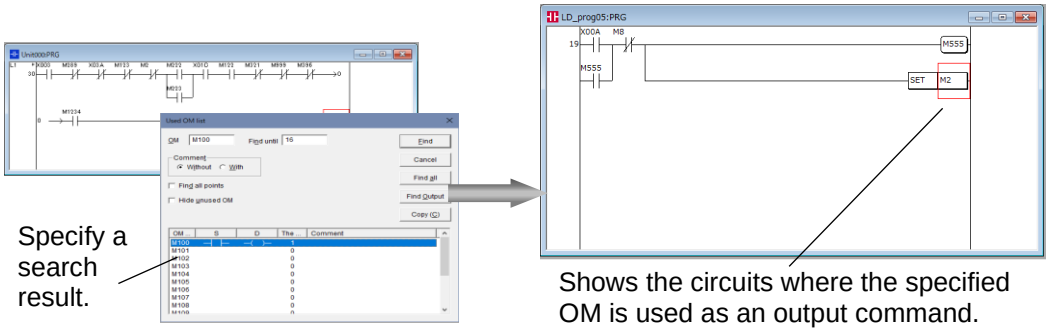


Refer to "7.1.4 Change display order in LD Program" for details about the search results display.

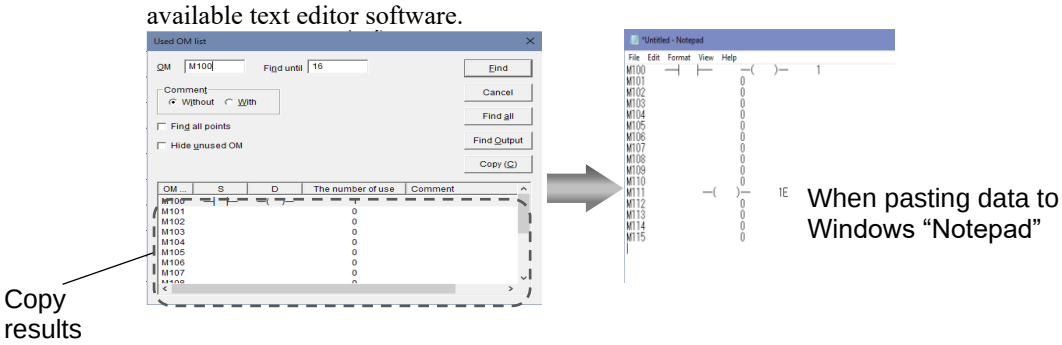
- Find output
  Specify an OM in the search results and click on the <Find Output> button, and this function shows all the circuits where the specified OM is used as an output command in the search results display.

Before searching for outputs

After finding outputs



- Copy
  The search results can be copied as text to the Windows clipboard.
  The user can paste the copied search results to another application like a commercially available text editor software.



Refer to "10.4.2 OM List (LD Language)" for details on the "Used OM list".

7.6.3 Timer & Counter List

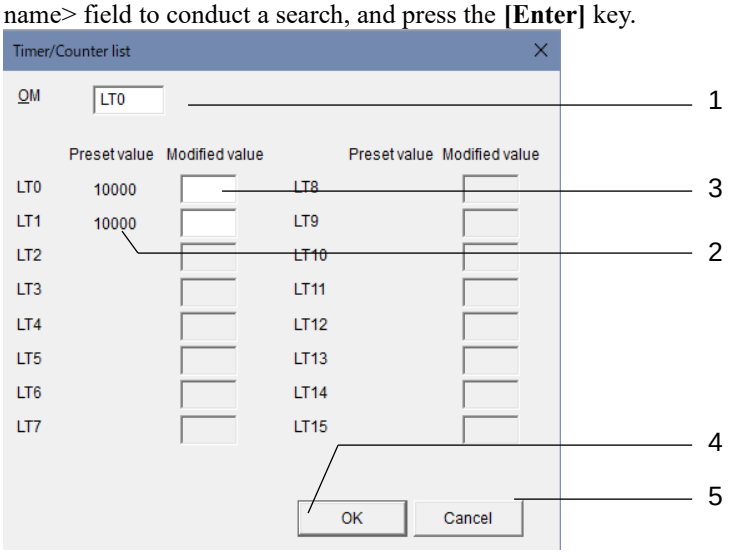
This list shows the set values for the local timer (LT) and the local counter (LC) that are used in the LD program.
The user can also change the set values.

(NOTE) The "Timer/Counter list" cannot be displayed while the edit lock is on or when the data is unconverted.

Operation procedure

1. Perform one of the following operations.
   - Select <Timer/Counter list> under <Find> in the <Edit> menu.
   - Press the **[Shift]** + **[F1]** keys

2. In the <Timer/Counter list> dialog, set a local timer (LT) or local counter (LC) in the <OM name> field to conduct a search, and press the **[Enter]** key.

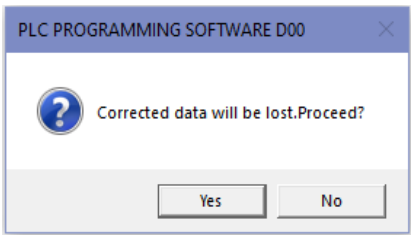


No.	Item	Description
1	OM name	Specifies the OM (LT or LC) in the search.
2	Preset value	Displays the current set values. The set values for an unused LT or LC are blank.
3	Modified value	Enter the new modified value. The user can specify a word OM (only D). If the modified value is left blank, the preset value for that local timer or local counter does not change. A value cannot be entered for an LT or LC that is not used.
4	<OK> button	Applies the setting changes, and the dialog closes.
5	<Cancel> button	Discards the setting changes, and the dialog closes.

3. Press the **[Page Down]** or **[Page Up]** key to change the display to next 16 or previous 16 values.
4. When changing a set value, set the "Change value" field to a value between 0 and 32767 or to D (OM) for the local timer (LT) or local counter (LC). Then, click on the <OK> button.

Refer to "10.4.3 Timer/Counter List (LD Language)" for details on the "Timer/Counter list".

(NOTE) When the user clicks on the <Cancel> button after changing values in the <Timer/Counter list>, a confirmation message appears asking whether or not to discard the setting changes.

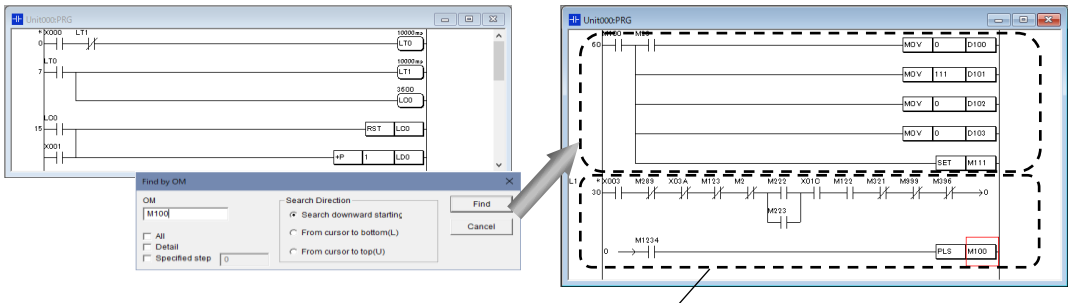


When <Yes> is selected, the setting changes are discarded and the dialog closes.
When <No> is selected, it goes back to the <Timer/Counter list> dialog.

7.6.4 OM Search

The user can specify an OM and search for a circuit where that OM is used.
In the OM search, it searches for circuit blocks with the specified OM and displays the search results.
Refer to "7.1.4 Change display order in LD Program" for details about the step number display and search results display.

(NOTE) The user cannot specify an OM to conduct a search when the data is not converted.



Each time a search is conducted, it searches for an OM following the step number order and displays a circuit block where the specified OM is used.

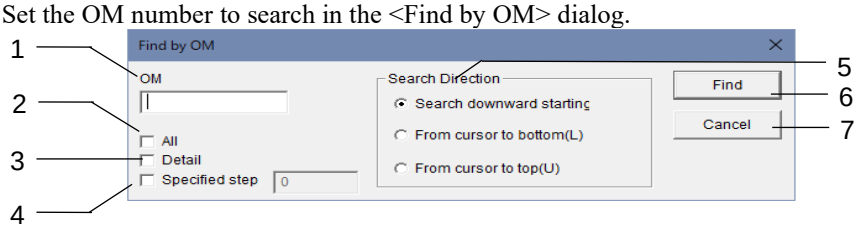
Refer to "10.4.4 OM Search (LD Language)" for details on the OM search.

7.6.4.1 OM search in OM search dialog

In the <Find by OM> dialog, the user can search for a given OM.

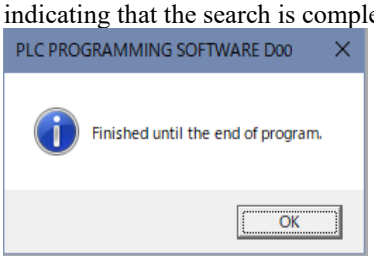
Operation procedure

1. Perform one of the following operations.
   - Select <Find by OM> under <Find> in the <Edit> menu
   - Press **[Shift]** + **[D]** keys or **[Ctrl]** + **[Shift]** + **[D]** keys
   - Enter OM while the <Edit lock> is on
2. Set the OM number to search in the <Find by OM> dialog.

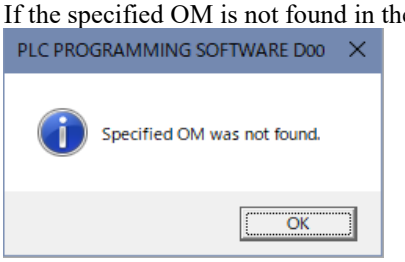


No.	Item	Description
1	OM name	Specifies the OM in the search. When the cursor is at an OM and the dialog is displayed, then the OM (at the cursor position) is automatically inserted.
2	All	Tick the checkbox to search all circuit blocks that use the specified OM.
3	Details	When a bit OM word is specified, tick the checkbox to search for OMs that are not displayed in the LD program.
4	Specified step	Tick the checkbox to search for OMs starting from a specified step number.
5	Search direction	Specifies the search direction. The user can select the following search methods. • Start from top and search down • Start from cursor position and search down • Start from cursor position and search up
6	<Start search> button	Starts the search.
7	<Cancel> button	Ends and exits the search, and the dialog closes.

3. Select the conditions (<All>, <Details>, <Specified step> or <Search Direction>) as necessary and click on the <Start search> button. If <All> is not selected, when the user clicks on the <Start search> button again, it searches for and displays the next circuit block.
4. When the search has finished checking all circuit blocks, the following message appears indicating that the search is complete.



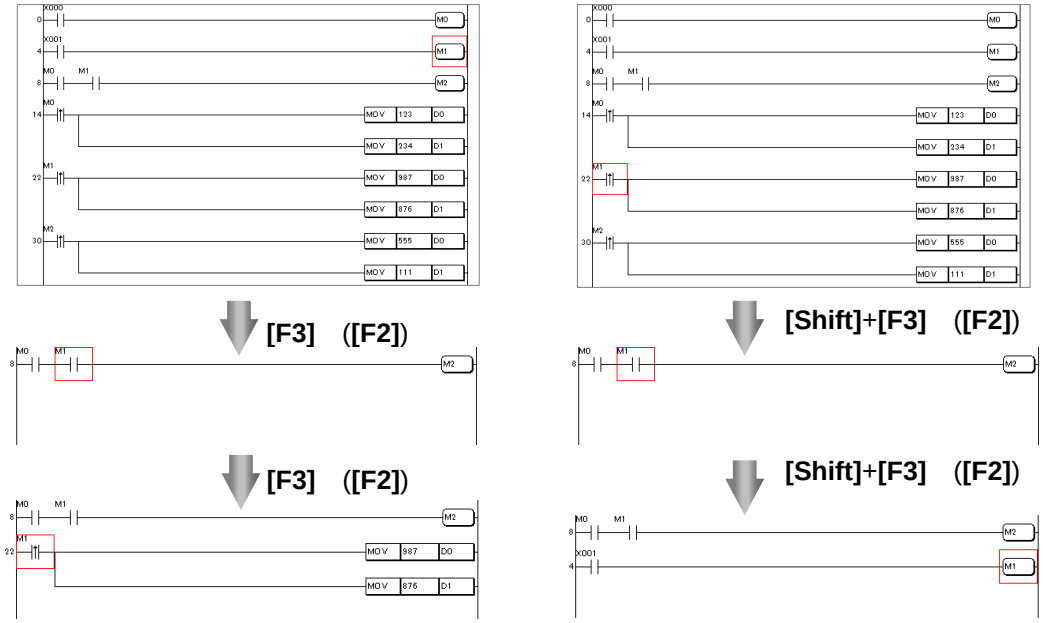
If the specified OM is not found in the search, then the following message is displayed.



When these messages are closed, then the <Find by OM> dialog also automatically closes.

7.6.4.2 Search circuit blocks for OM specified by cursor position

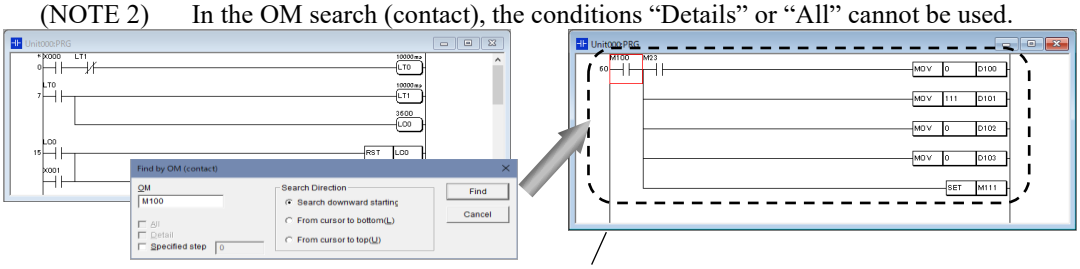
Press the **[F3]** key to display the next circuit block where the OM at the cursor position is used. In addition, the user can press the **[Shift]** + **[F3]** keys to display the next circuit block.
If the user accesses the option settings and assigns a monitoring operation to the **[F3]** key, then the **[F2]** key can be used for search operations.
Refer to "5.6.1.1 Edit screen adjustment (option settings)" for further details on the option settings.



7.6.5 OM Search (Contact)

The user can specify an OM and search for a circuit where that OM is used as a contact command.
In the OM search (contact), it searches for circuit blocks where the specified OM is used as a contact command and displays the search results.
Refer to "7.1.4 Change display order in LD Program" for details about the step number display and search results display.

(NOTE 1) The user cannot specify an OM (contact) to conduct a search when the data is not converted.
(NOTE 2) In the OM search (contact), the conditions "Details" or "All" cannot be used.

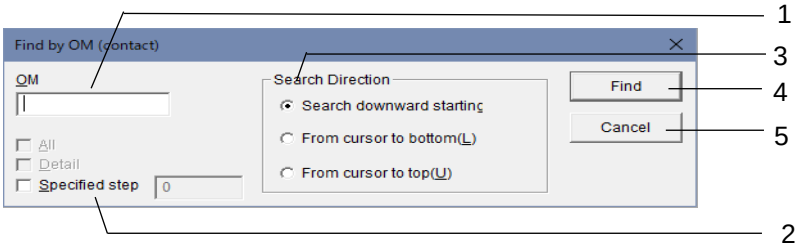


Each time a search is conducted, it searches following the step number order and displays a circuit block where the specified OM is used as a contact command.

Operation procedure

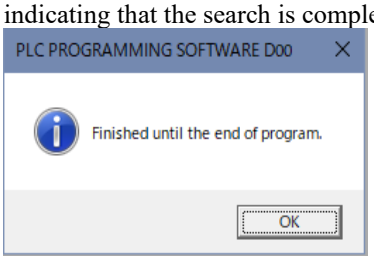
1. Perform one of the following operations.
   - Select <Find by OM (contact)> under <Find> in the <Edit> menu
   - Press the **[Shift]** + **[A]** keys

2. Set the OM number to search in the <Find by OM (contact)> dialog.

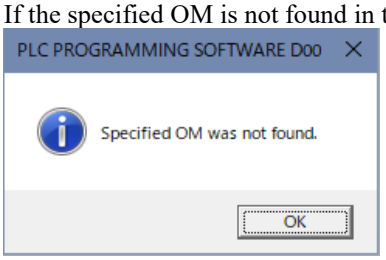


No.	Item	Description
1	OM name	Specifies the OM in the search. When the cursor is at an OM and the dialog is displayed, then the OM (at the cursor position) is automatically inserted.
2	Specified step	Tick the checkbox to search for OMs starting from a specified step number.
3	Search direction	Specifies the search direction. The user can select the following search methods. • Start from top and search down • Start from cursor position and search down • Start from cursor position and search up
4	<Start search> button	Starts the search.
5	<Cancel> button	Ends and exits the search, and the dialog closes.

3. Select the conditions (<Specified step> or <Search Direction>) as necessary and click on the <Start search> button. When the user clicks on <Start search> button again, it searches for the next circuit block.
4. When the search has finished checking all circuit blocks, the following message appears indicating that the search is complete.



If the specified OM is not found in the search, then the following message is displayed.



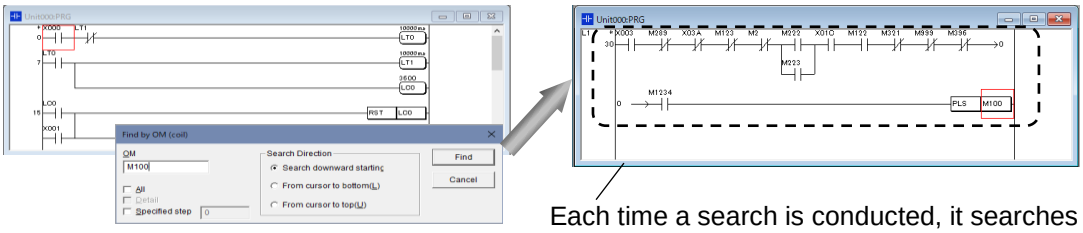
When these messages are closed, then the <Find by OM (contact)> dialog also automatically closes.

Refer to “10.4.5 OM Search (Contact) (LD Language)” for details on the OM search (contact).

7.6.6 OM Search (Coil)

The user can specify an OM and search for a circuit where that OM is used as an output command. In the OM search (coil), it searches for circuit blocks where the specified OM is used as an output command and displays the search results.
Refer to "7.1.4 Change display order in LD Program" for details about the step number display and search results display.

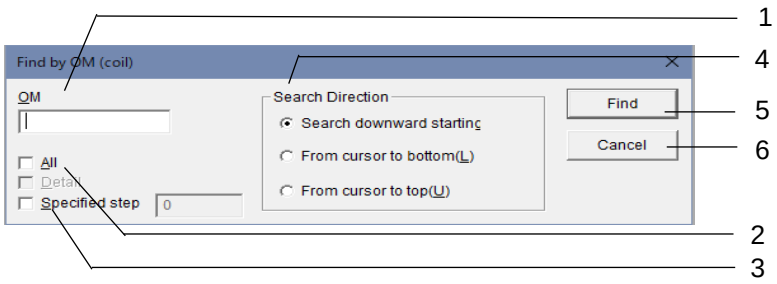
(NOTE 1) The user cannot specify an OM (coil) to conduct a search when the data is not converted.
(NOTE 2) In the OM search (coil), the search conditions "Details" cannot be used.



Each time a search is conducted, it searches following the step number order and displays a circuit block where the specified OM is used as an output command.

Operation procedure

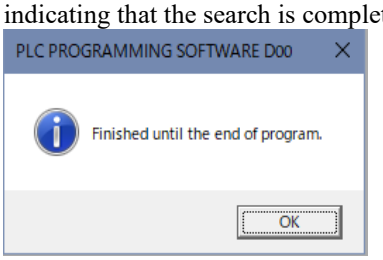
1. Perform one of the following operations.
   - Select <Find by OM (coil)> under <Find> in the <Edit> menu
   - Press the **[Shift]** + **[C]** keys
2. Set the OM number to search in the <Find by OM (coil)> dialog.



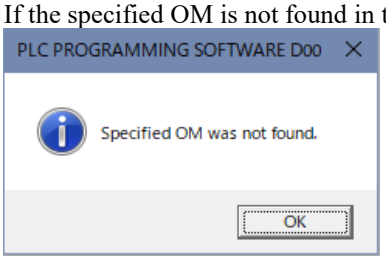
No.	Item	Description
1	OM name	Specifies the OM in the search. When the cursor is at an OM and the dialog is displayed, then the OM (at the cursor position) is automatically inserted.
2	All	Tick the checkbox to search all circuit blocks that use the specified OM as an output command.
3	Specified step	Tick the checkbox to search for OMs starting from a specified step number.
4	Search direction	Specifies the search direction. The user can select the following search methods. • Start from top and search down • Start from cursor position and search down • Start from cursor position and search up
5	<Start search> button	Starts the search.
6	<Cancel> button	Ends and exits the search, and the dialog closes.

3. Select the conditions (<All>, <Specified step> or <Search Direction>) as necessary and click on the <Start search> button. If <All> is not selected, when the user clicks on the <Start search> button again, it searches for and displays the next circuit block.

4. When the search has finished checking all circuit blocks, the following message appears indicating that the search is complete.



If the specified OM is not found in the search, then the following message is displayed.



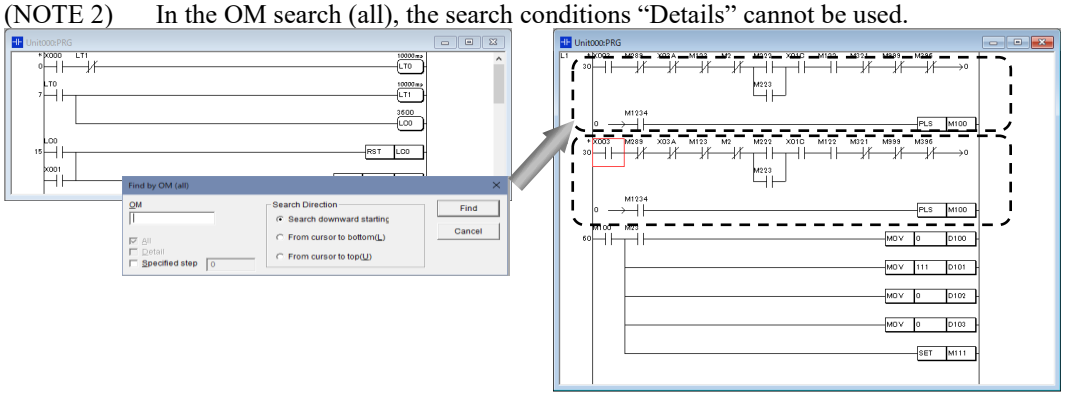
When these messages are closed, then the <Find by OM (coil)> dialog also automatically closes.

Refer to “10.4.6 OM Search (Coil) (LD Language)” for details on the OM search (coil).

7.6.7 OM Search (All)

The user can specify an OM and search all circuits to check where that OM is used.
In the OM search (all), it searches for circuit blocks with the specified OM and displays the search results.
Refer to “7.1.4 Change display order in LD Program” for details about the step number display and search results display.

(NOTE 1) The user cannot specify an OM (all) to conduct a search when the data is not converted.
(NOTE 2) In the OM search (all), the search conditions “Details” cannot be used.

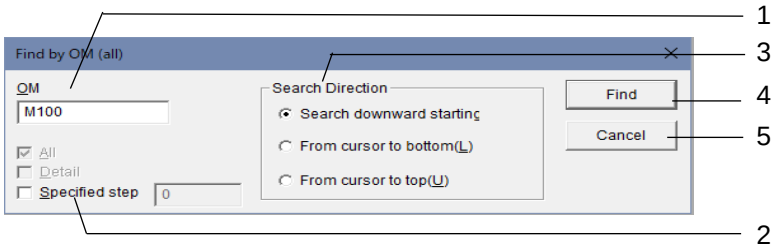


Each time a search is conducted, it searches all OMs following the step number order and displays a circuit block where the specified OM is used.

Operation procedure

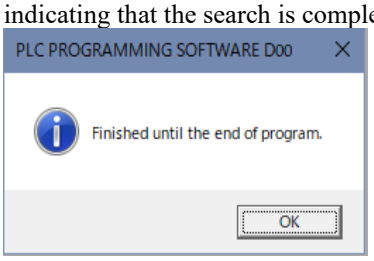
1. Perform one of the following operations.
   - Select <Find by OM (All)> under <Find> in the <Edit> menu.
   - Press the **[Shift]** + **[B]** keys

2. Set the OM number to search in the <Find by OM (All)> dialog.

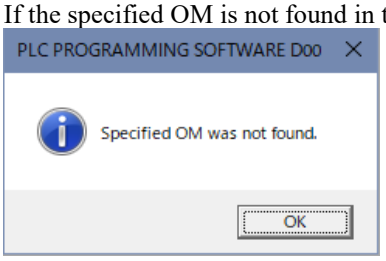


No.	Item	Description
1	OM name	Specifies the OM in the search. When the cursor is at an OM and the dialog is displayed, then the OM (at the cursor position) is automatically inserted.
2	Specified step	Tick the checkbox to search for OMs starting from a specified step number.
3	Search direction	Specifies the search direction. The user can select the following search methods. • Start from top and search down • Start from cursor position and search down • Start from cursor position and search up
4	<Start search> button	Starts the search.
5	<Cancel> button	Ends and exits the search, and the dialog closes.

3. Select the condition <Specified step> as necessary and click on the <Start search> button.
4. When the search has finished checking all circuit blocks, the following message appears indicating that the search is complete.



If the specified OM is not found in the search, then the following message is displayed.



When these messages are closed, then the <Find by OM (All)> dialog also automatically closes.

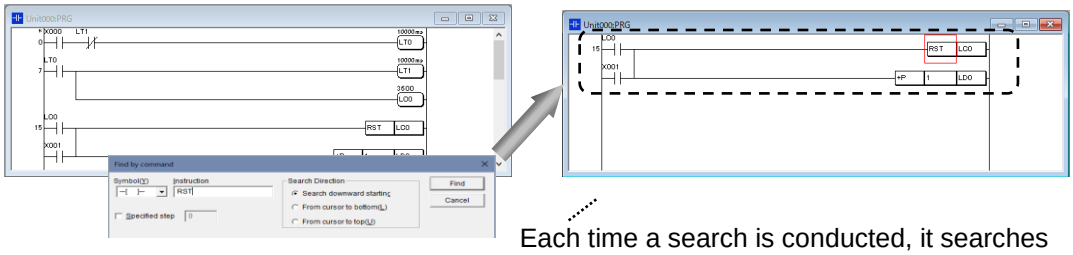
Refer to “10.4.7 OM Search (All) (LD Language)” for details on the OM search (all).

7.6.8 Command Search

The user can specify a command and search the circuits to check where the command is used.
In the command search, it searches for circuit blocks where the specified command is used and displays the search results.
Refer to "7.1.4 Change display order in LD Program" for details about the step number display and search results display.

(NOTE) The user cannot specify a command to conduct a search when the data is not converted.

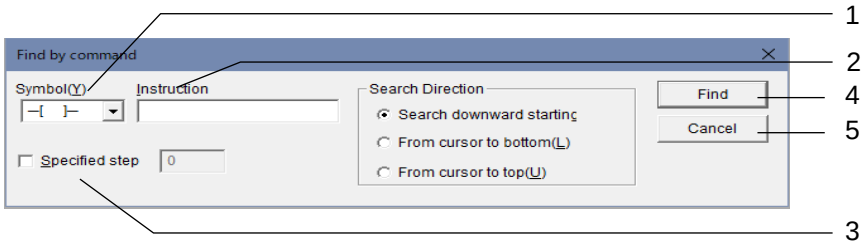
In the command search, the user can specify an output command and application command.



Each time a search is conducted, it searches following the step number order and displays a circuit block where the specified command is used.

Operation procedure

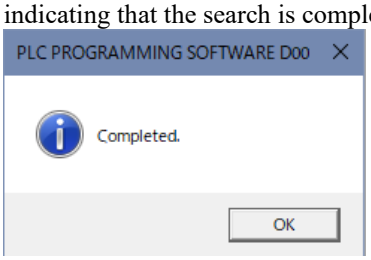
1. Perform one of the following operations.
   - Select <Find by command> under <Find> in the <Edit> menu.
   - Press the **[Shift]** + **[I]** keys
2. Set the command to search in the <Find by command> dialog.



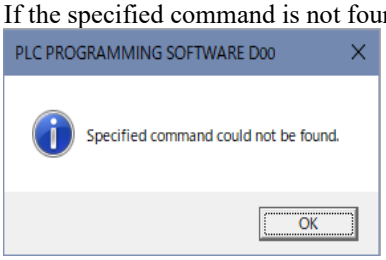
No.	Item	Description
1	Symbol	Sets the command symbol in the search. When the cursor is at a command and the dialog is displayed, then the command symbol (at the cursor position) is automatically inserted.
2	Command	Sets the command in the search. When the cursor is at a command and the dialog is displayed, then the command (at the cursor position) is automatically inserted.
3	Specified step	Tick the checkbox to search for OMs starting from a specified step number.
4	<Start search> button	Starts the search.
5	<Cancel> button	Ends and exits the search, and the dialog closes.

3. Select the conditions (<Specified step> or <Search Direction>) as necessary and click on the <Start search> button. When the user clicks on <Start search> button again, it searches for the next circuit block.

4. When the search has finished checking all circuit blocks, the following message appears indicating that the search is complete.



If the specified command is not found in the search, then the following message is displayed.



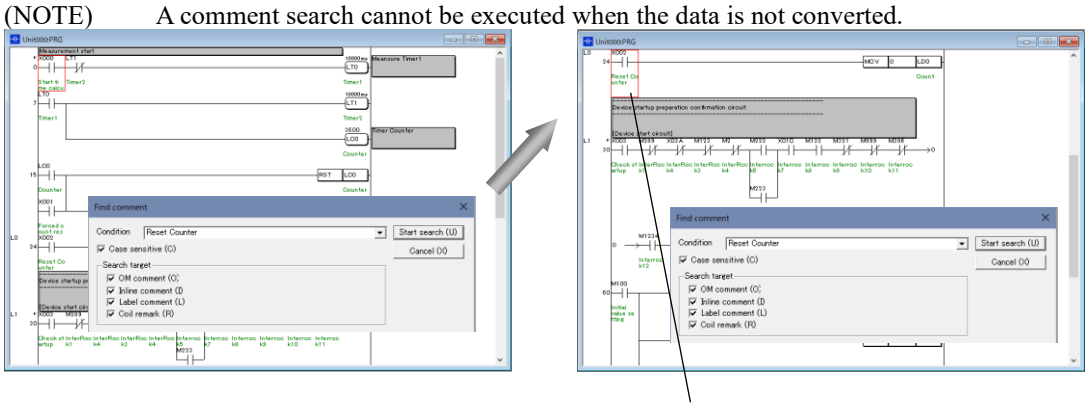
When these messages are closed, then the <Find by command> dialog also automatically closes.

Refer to “10.4.8 Command Search (LD Language)” for details on the command search.

7.6.9 Comment Search

The comment search conducts a search to check if the specified character string is being used or not in an OM comment, inline comment, label comment or coil remark.
The comment search is carried out in the step number display. If the comment search is started and the search results display is shown, the display will change automatically to the step number display. In addition, if the comment targeted in the search is not shown in the editing window, the display changes automatically to show it. When performing an OM comment search, a 32 character display is used for the OM comment.
The comment search starts from the beginning of the program, regardless of the current cursor position.
Refer to “7.1.4 Change display order in LD Program” for details about the step number display and search results display.

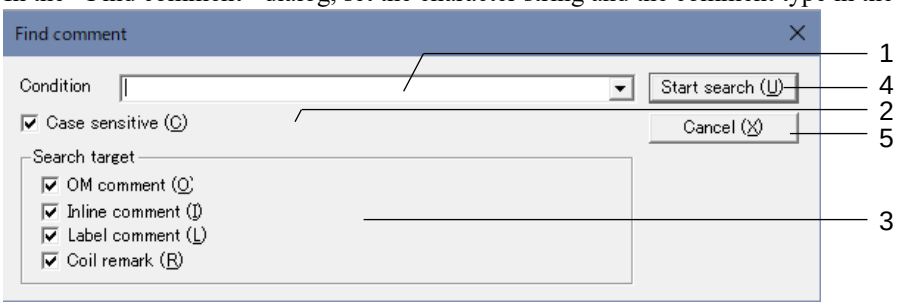
(NOTE) A comment search cannot be executed when the data is not converted.



Cursor moves to the location where the comment has the specified character string.

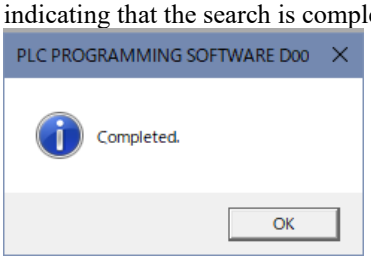
Operation procedure

1. Perform one of the following operations.
   - Select <Find comment> under <Find> in the <Edit> menu.
   - Press the **[Shift]** + **[F]** keys
2. In the <Find comment> dialog, set the character string and the comment type in the search.

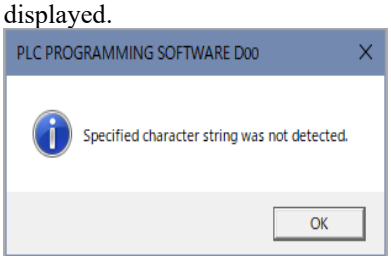


No.	Item	Description
1	Conditions	Sets a character string in the search using no more than 64 characters. When the user clicks on the ▼ button, a maximum of 32 hits is displayed in the search history in the PLC Programming software D00 LD/ST Programming Editor that is launched.
2	Case sensitive	Selects whether or not to differentiate between uppercase and lowercase letters in the character string search. Tick the checkbox to differentiate between uppercase and lowercase letters.
3	Search target	Selects the following types of the comments in the search. • OM comment • Inline comment • Label comment • Coil remark Tick a checkbox to include the corresponding type in the search.
4	<Start search> button	Starts the search.
5	<Cancel> button	Ends and exits the search, and the dialog closes.

3. Click on the <Start search> button. When the user clicks on <Start search> button again, it searches for the next comment.
4. When the search has finished checking all comments, the following message appears indicating that the search is complete.



If the specified character string is not found in the search, then the following message is displayed.



5. Even if the search is complete, it does not close and exit the <Find comment> dialog. The user can continue doing other searches.
   Click on the <Cancel> button to close and exit the search.

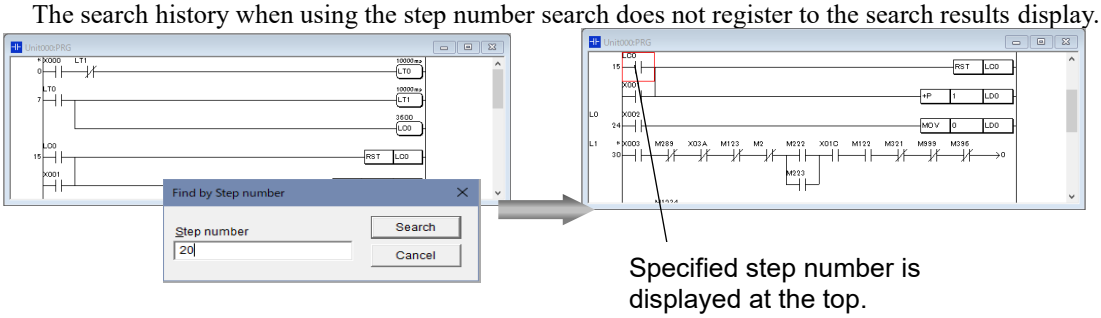
Refer to “10.4.18 Comment Search (LD Language)” for further details on the comment search.

7.7 Search & Jump

This section describes the operations on how to check the OM status and how to search for step numbers, OMs and commands.

7.7.1 Step Number Search

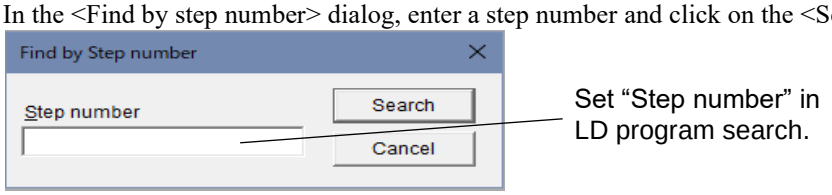
The user can specify a step number in the LD program and display it.
In the step number search, the circuit block with the specified step is displayed at the top of the LD program.
The search history when using the step number search does not register to the search results display.



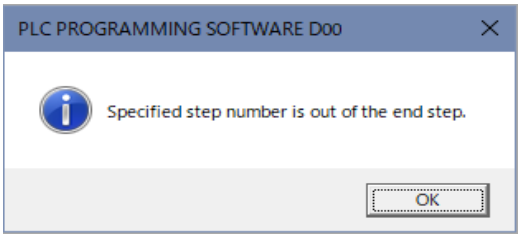
The operation procedure for the step number search is as follows.

Operation procedure

1. Perform one of the following operations.
   - Select <Find by step number> under <Find and jump> in the <Edit> menu.
   - Press the **[Shift]** + **[N]** keys
   - Enter step number (numerical value) while the <Edit lock> is on.
2. In the <Find by step number> dialog, enter a step number and click on the <Search> button.



When the user specifies a step number that is bigger than the END command step number, then the following message is displayed. The END command is displayed at the bottom of the LD program edit window.

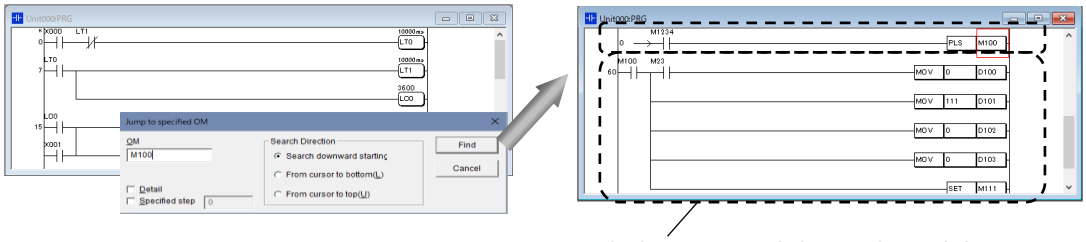


Refer to “10.4.10 Step Number Search (LD Language)” for details on the step number search.

7.7.2 OM Jump

The user can specify an OM and search for circuits where that OM is used. When a circuit is found, then it jumps to that location.
The search history when using the OM jump function does not register to the search results display.

(NOTE) The user cannot specify an OM to conduct a search when the data is not converted.

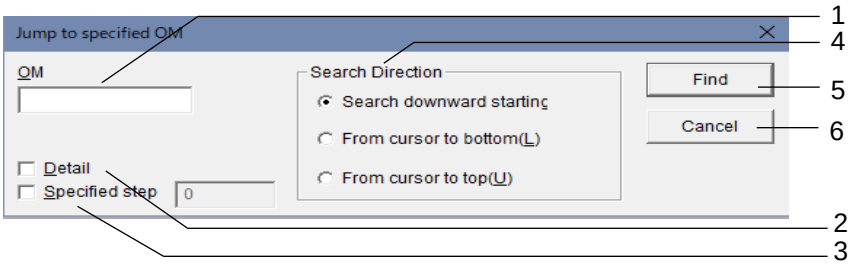


Each time a search is conducted, it searches for locations where the specified OM is used and jumps to that location.

Refer to "10.4.11 OM Jump (LD Language)" for details on the OM jump.

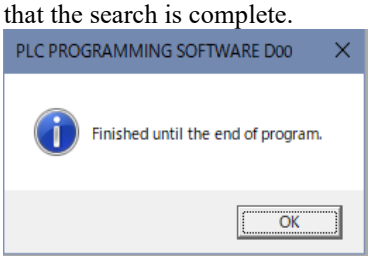
Operation procedure

1. Perform one of the following operations.
   - Select <Find by OM> under <Find and jump> in the <Edit> menu.
   - Press the **[Shift]** + **[H]** keys
2. Set the OM number to search in the <Find and jump by OM> dialog.

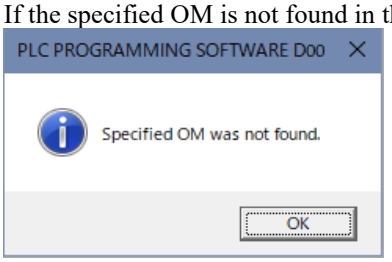


No.	Item	Description
1	OM name	Specifies the OM in the search. When the cursor is at an OM and the dialog is displayed, then the OM (at the cursor position) is automatically inserted.
2	Details	When a bit OM word is specified, tick the checkbox to search for OMs that are not displayed in the LD program.
3	Specified step	Tick the checkbox to search for OMs starting from a specified step number.
4	Search direction	Specifies the search direction. The user can select the following search methods. • Start from top and search down • Start from cursor position and search down • Start from cursor position and search up
5	<Start search> button	Starts the search.
6	<Cancel> button	Ends and exits the search, and the dialog closes.

3. Select the conditions (<Details>, <Specified step> or <Search Direction>) as necessary and click on the <Start search> button. When the user clicks on <Start search> button again, it jumps to the next program.
4. When the search has finished checking all programs, the following message appears indicating that the search is complete.



If the specified OM is not found in the search, then the following message is displayed.

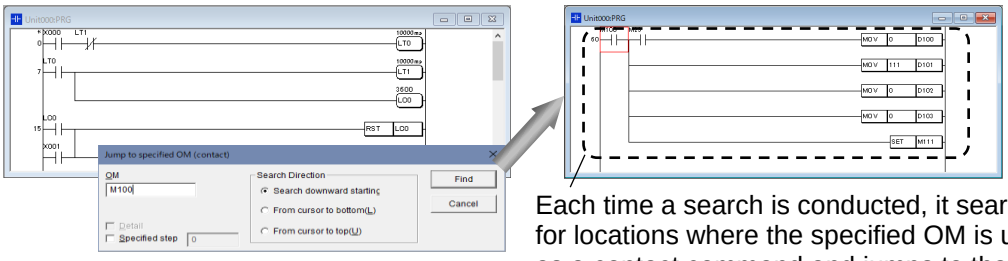


Even when these messages are closed, the <Find and jump by OM> dialog does not automatically close.

7.7.3 OM Jump (Contact)

The user can specify an OM and search for circuits where that OM is used as a contact command. When a circuit is found, then it jumps to that location.
The search history when using the OM jump (contact) function does not register to the search results display.

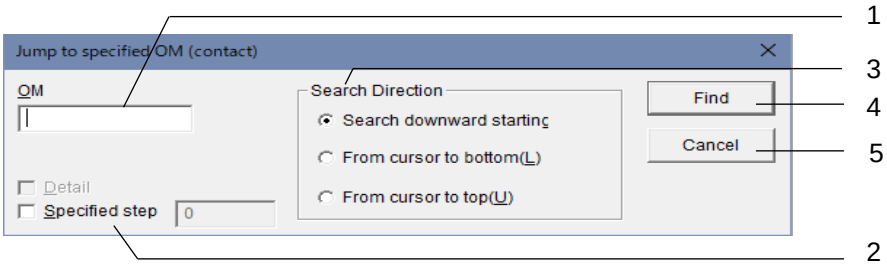
(NOTE 1) The user cannot specify an OM (contact) to jump to when the data is not converted.
(NOTE 2) In the OM jump (contact), the search conditions “Details” cannot be used.



Each time a search is conducted, it searches for locations where the specified OM is used as a contact command and jumps to that location.

Operation procedure

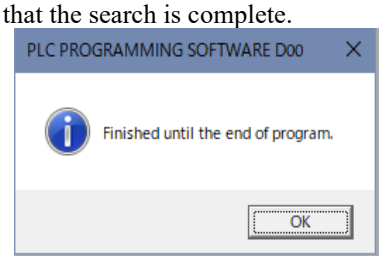
1. Perform one of the following operations.
   - Select <Find by OM (contact)> under <Find and jump> in the <Edit> menu.
   - Press the **[Shift]** + **[Q]** keys
2. Set the OM number to search in the <Jump to specified OM (contact)> dialog.



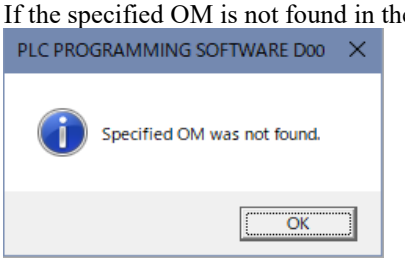
No.	Item	Description
1	OM name	Specifies the OM in the search. When the cursor is at an OM and the dialog is displayed, then the OM (at the cursor position) is automatically inserted.
2	Specified step	Tick the checkbox to search for OMs starting from a specified step number.
3	Search direction	Specifies the search direction. The user can select the following search methods. • Start from top and search down • Start from cursor position and search down • Start from cursor position and search up
4	<Start search> button	Starts the search.
5	<Cancel> button	Ends and exits the search, and the dialog closes.

3. Select the conditions (<Specified step> or <Search Direction>) as necessary and click on the <Start search> button. When the user clicks on <Start search> button again, it searches for the next program.

   When the search has finished checking all programs, the following message appears indicating that the search is complete.



   If the specified OM is not found in the search, then the following message is displayed.



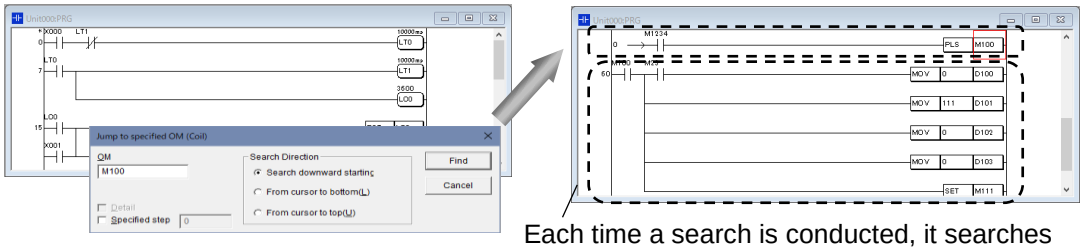
   Even when these messages are closed, the <Jump to specified OM (contact)> dialog does not automatically close.

Refer to “10.4.12 OM Jump (Contact) (LD Language)” for details on the OM jump (contact).

7.7.4 OM Jump (Coil)

The user can specify an OM and search for circuits where that OM is used as an output command. When a circuit is found, then it jumps to that location.
The search history when using the OM jump (coil) function does not register to the search results display.

(NOTE 1) The user cannot specify an OM (coil) to jump to when the data is not converted.
(NOTE 2) In the OM jump (coil), the search conditions "Details" cannot be used.

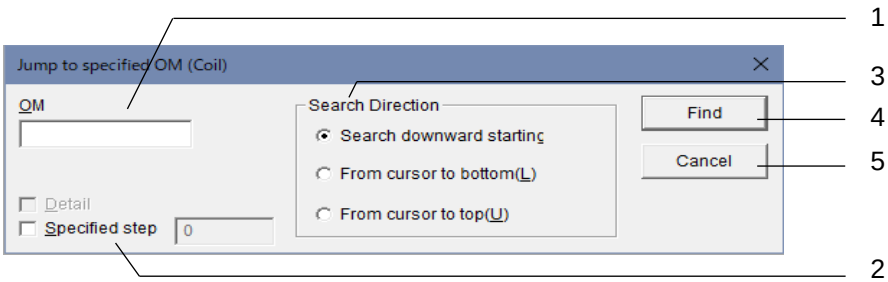


Each time a search is conducted, it searches for locations where the specified OM is used as an output command and jumps to that location.

Operation procedure

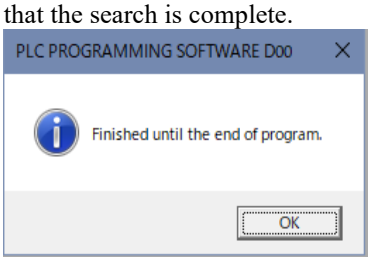
1. Perform one of the following operations.
   - Select <Find by OM (coil)> under <Find and jump> in the <Edit> menu.
   - Press the **[Shift]** + **[G]** keys

2. Set the OM number to search in the <Find and jump OM (Coil)> dialog.

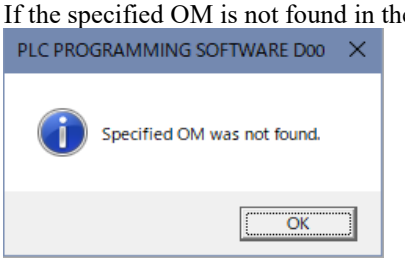


No.	Item	Description
1	OM name	Specifies the OM in the search. When the cursor is at an OM and the dialog is displayed, then the OM (at the cursor position) is automatically inserted.
2	Specified step	Tick the checkbox to search for OMs starting from a specified step number.
3	Search direction	Specifies the search direction. The user can select the following search methods. • Start from top and search down • Start from cursor position and search down • Start from cursor position and search up
4	<Start search> button	Starts the search.
5	<Cancel> button	Ends and exits the search, and the dialog closes.

3. Select the conditions (<Specified step> or <Search Direction>) as necessary and click on the <Start search> button. When the user clicks on <Start search> button again, it searches for the next program.
4. When the search has finished checking all programs, the following message appears indicating that the search is complete.



If the specified OM is not found in the search, then the following message is displayed.



Even when these messages are closed, the <Find and jump by OM (Coil)> dialog does not automatically close.

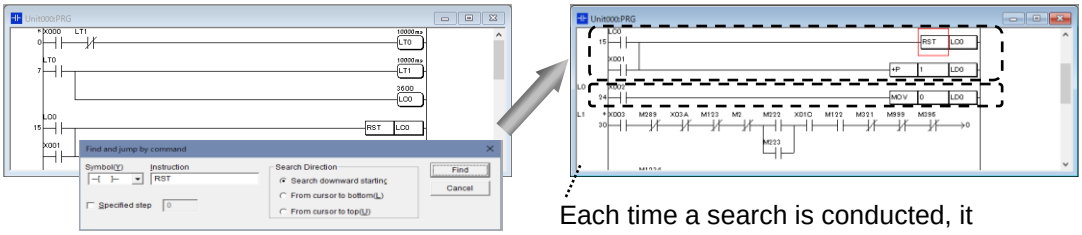
Refer to “10.4.13 OM Jump (Coil) (LD Language)” for details on the OM jump (coil).

7.7.5 Command Jump

The user can specify a command and search for circuits where that command is used. When a circuit is found, then it jumps to that location.
The search history when using the command jump function does not register to the search results display.

(NOTE) The user cannot specify a command to jump to when the data is not converted.

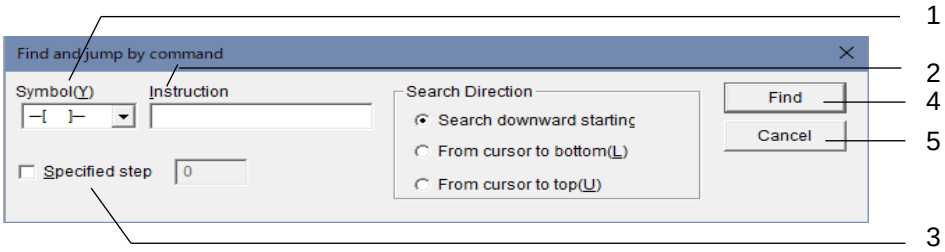
In the command jump, the user can specify an output command and application command.



Each time a search is conducted, it searches following the step number order and displays a circuit block where the specified command is used.

Operation procedure

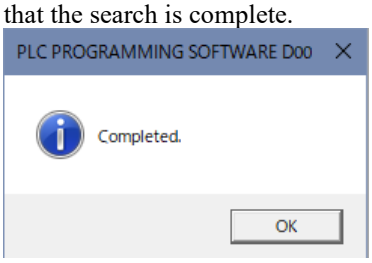
1. Perform one of the following operations.
   - Select <Find and jump by command> under <Find and jump> in the <Edit> menu.
   - Press the **[Shift]** + **[J]** keys
2. Set the command to search in the <Find by command> dialog.



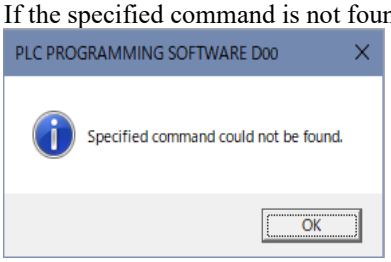
No.	Item	Description
1	Symbol	Sets the command symbol in the search. When the cursor is at a command and the dialog is displayed, then the command symbol (at the cursor position) is automatically inserted.
2	Command	Sets the command in the search. When the cursor is at a command and the dialog is displayed, then the command (at the cursor position) is automatically inserted.
3	Specified step	Tick the checkbox to search for OMs starting from a specified step number.
4	<Start search> button	Starts the search.
5	<Cancel> button	Ends and exits the search, and the dialog closes.

3. Select the conditions (<Specified step> or <Search Direction>) as necessary and click on the <Start search> button. When the user clicks on <Start search> button again, it searches for the next program.

4. When the search has finished checking all programs, the following message appears indicating that the search is complete.



If the specified command is not found in the search, then the following message is displayed.



Even when these messages are closed, the <Find and jump by OM> dialog does not automatically close.

Refer to “10.4.14 Command Jump (LD Language)” for details on the command jump.

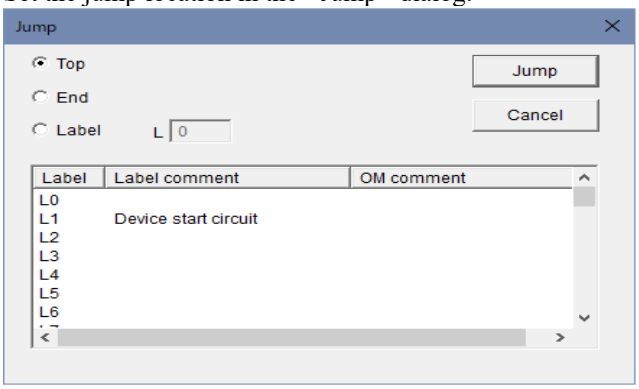
7.8 Jump to Beginning, End & Specified Label in LD Program

When using the step number display, this section describes how to move or to jump to the beginning, the end or a specified label (L) in the LD program.
The user operation to move to a specified label (L) is possible with the OM search function, but the display changes to the search results display.
Refer to “7.6.4 OM search” for details on the OM search. Refer to “7.1.4 Change display order in LD Program” for details about the step number display and search results display.

The following shows the procedure for the jump operation.

Operation procedure

1. Perform one of the following operations.
   - Select <Jump> from the <Edit> menu
   - Press the **[Ctrl]** + **[J]** keys
2. Set the jump location in the <Jump> dialog.



   Select “First” to jump to the first line or top of the LD program.
   Select “End” to jump to the last line or end of the LD program.
   To jump to a label, specify the label and set the label number, or select a label to jump to from the label list.
3. Click on the <Jump> button.
4. Even after jumping to a location, the <Jump> dialog does not close. The user can continue doing more jump operations.
   Click on the <Cancel> button to close and exit the jump operation.

Note that if the user opens the <Jump> dialog and the search results display is shown, the display will change automatically to the step number display.

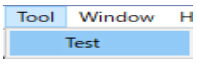
7.9 Launch Simulation

Before running an LD program on the machine, the user can check the operation of the LD program in the PLC Programming software D00 EDITOR.
The following is the procedure for running a simulation of an LD program that was created.

Operation procedure

1. Perform one of the following operations.
   - Select <Simulation> from the <Tool> menu.
   - Click on the button in the tool bar to turn ON the simulation.
2. After the simulation is launched, a checkmark "✓" appears before the item name <Simulation> in the menu.

Checkmark "✓" appears



3. Write the LD program to the simulation.
   Refer to the "5.10.2 Write programs" for details about writing LD programs.

- Exit simulation
  When there is a checkmark "✓" next to the <Simulation> item in the <Tool> menu, select <Simulation> again (to deselect and exit).
  After exiting the simulation, the checkmark "✓" next to the <Simulation> disappears.

(NOTE 1) The user can launch and exit the simulation from the PLC Programming software D00 and the PLC Programming software D00 FBD Programming Editor.
When running an LD program in the simulation, there is no need to re-launch the PLC Programming software D00 LD/ST Programming Editor if the simulation has already been launched with another tool.

(NOTE 2) The POUs set in the <Set program execution> dialog are written to the program. Register the LD program in the <Set program execution> dialog to run the LD program currently being edited in the simulation. Refer to "5.8 Program execution settings" for details on the <Set program execution> dialog.

7.10 NC(BUILT IN PLC) Monitoring

The user can perform the following operations to execute LD program monitoring in the NC(BUILT IN PLC).

Operation procedure

Perform one of the operations below in the PLC Programming software D00 LD/ST Programming Editor.

- Press the **[F2]** key
- Select <Start monitoring> under the <Online> menu
- Click on from the function key tool bar

Refer to "5.12 NC(BUILT IN PLC) Monitoring" for further details on monitoring.
In addition, refer to "7.12 Monitoring operations" and "7.13 DEBUG functions" for details on LD program operations when monitoring.

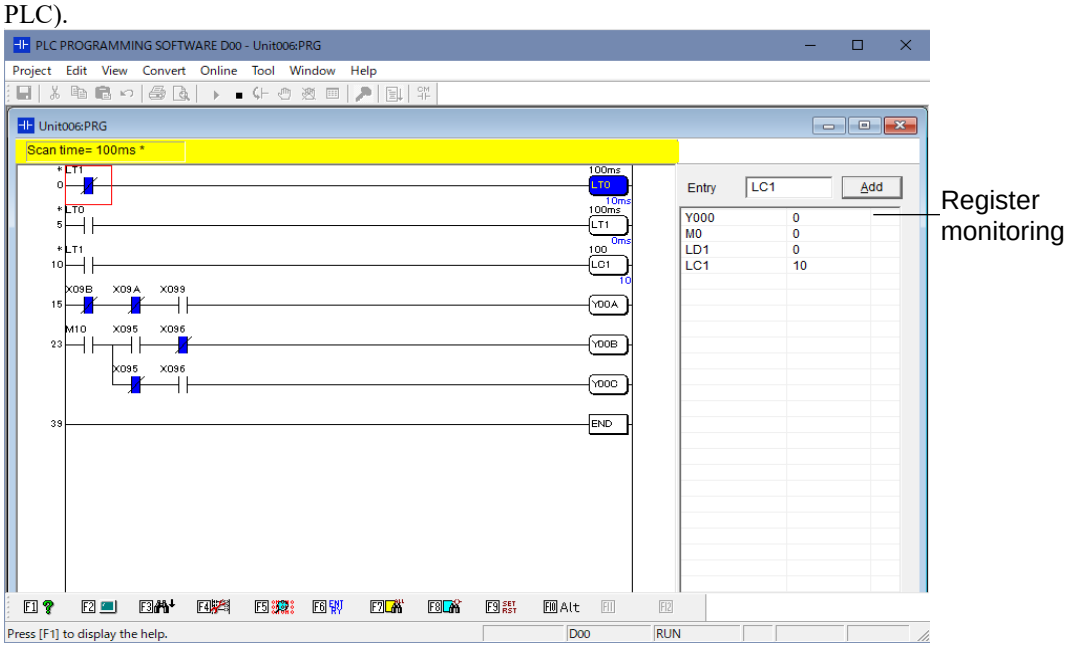
7.11 Run & Stop in NC(BUILT IN PLC)

Refer to "5.13 Run & Stop in NC(BUILT IN PLC)" for details on running and stopping in the NC(BUILT IN PLC).

7.12 Monitoring Operations

This section describes the operations for each function when monitoring an LD program that is running in the NC(BUILT IN PLC).
Refer to “5.12 NC(BUILT IN PLC) Monitoring” for details on changing to monitor mode in the PLC Programming software D00 EDITOR.

The user can check the status of the NC(BUILT IN PLC) and the current value of the OMs in the LD program by connecting the PLC Programming software D00 EDITOR to the NC(BUILT IN PLC).



When the contact is highlighted in blue for a contact command, that contact is conducting.
For contact *a*, the OM is turned ON.
For contact *b*, the OM is turned OFF.
When the command is highlighted in blue for an output command, the OM is turned ON.
The application command displays the current value under the OM.

By registering an OM beforehand in the monitoring registration, the user can constantly check the OM status or value regardless of the display position of the LD program.
Refer to “7.12.2 Monitoring registration” for details on monitoring registration.

The user can change the base number when displaying the current value of the word OM.
Refer to “7.12.1 Change base number for monitoring values” for details on changing the base number.

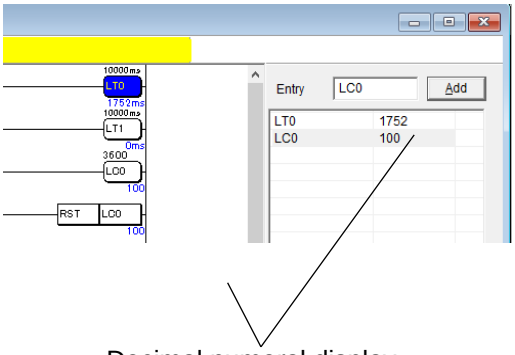
The user can force a change in the current OM status while monitoring.
The following functions can be used to change the OM status.

- To force the bit OM to change to ON or OFF: OM SET/RST
- To force a change in the current value of the word OM: Change current value
- To change the OMs in the table format all at once: Batch monitor

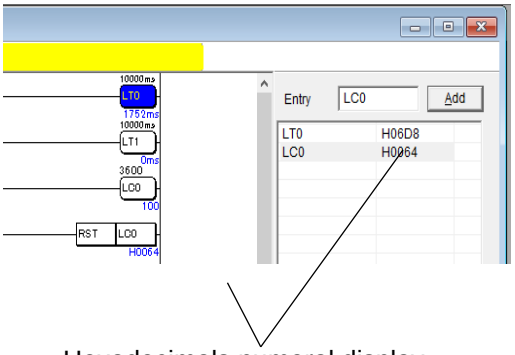
Refer to “7.12.3 SET/RST OM” for setting or resetting the OMs.
Refer to “7.12.4 Change current value” for details about the current monitoring values.
Refer to “7.12.5 Batch monitoring” for details about batch monitoring.

7.12.1 Change Base Number for Monitoring Values

The user can change the current values displayed during monitoring to “Decimals” or “Hexadecimals”.



Decimal numeral display



Hexadecimals numeral display

(NOTE) The base number cannot be changed for the set values and current values of the coil command on the local timers (LT) and local counters (LC).
The values are always displayed in decimals.

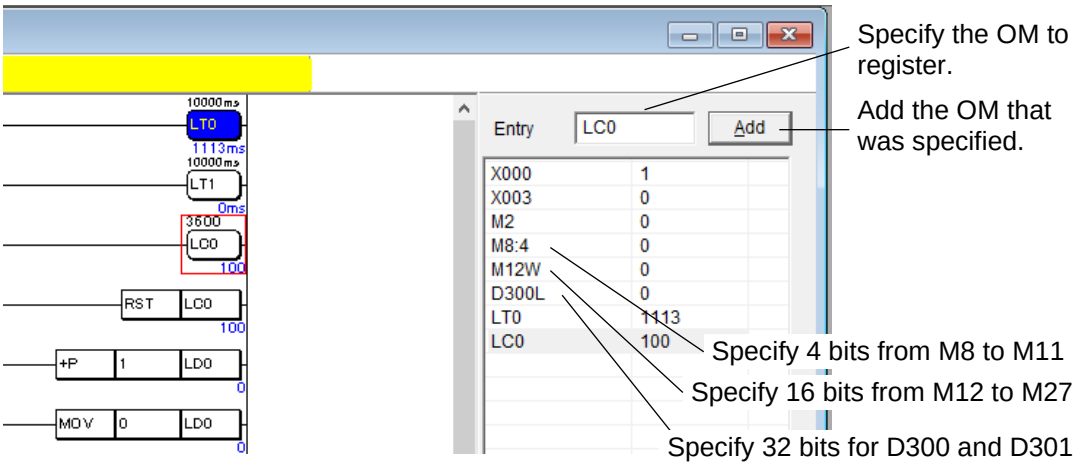
<u>Operation procedure</u>

Perform one of the following operations.

- Select <Change monitoring base number> under <Change display> in the <Display> menu.
- Press the **[Shift]** + **[F8]** keys

7.12.2 Monitoring Registration

By registering an OM beforehand in the monitoring registration, the user can constantly monitor the OM status or value regardless of the display position of the circuit.
The user can also specify the digits for the bit OM in the monitoring registration.
Refer to “Chapter 11 Commands (LD)” in the PLC System Manual for details on specifying the digits of an OM.



In the monitoring registration, the maximum number of OMs that can be registered is 24. If the user enters 25 OMs, the first OM that was registered is cancelled.

Operation procedure

1. Select <Register monitoring> from the <Display> menu to hide the monitoring registration.
2. The user can perform one of the following operations to set an OM in the monitoring registration.
   - Enter an OM into the OM setting field at the top of the monitoring registration, and press the <Add> button or press the **[Enter]** key.
   - Move the cursor to the desired OM, go to the <Online menu> and select <Register (directly)> under <Register monitoring>.
   - Move the cursor to the desired OM, right-click on the mouse to show the menu, and select <Register (directly)> under <Register monitoring>.
   - Move the cursor to the desired OM, and press the **[F6]** key.
   - Move the cursor to the desired OM, go to the <Online menu> and select <Register (command input)> under <Register monitoring>. Then, press the <Add> button or the **[Enter]** key.
   - Move the cursor to the desired OM, right-click on the mouse to show the menu, and select <Register (command input)> under <Register monitoring>. Then, press the <Add> button or the **[Enter]** key.
   - Move the cursor to the desired OM, press the **[Ctrl]** + **[Shift]** + **[E]** keys, and then press the <Add> button or the **[Enter]** key.
3. To unregister one of the OMs in the monitoring registration, select the desired OM, and then press the **[Delete]** key.
4. The user can perform one of the following operations to unregister or clear all of the OMs from the monitoring registration.
   - Select <Clear all entries> under <Register monitoring> in the <Online> menu
   - Right-click on the mouse to show the menu, and select <Clear all entries> under <Register monitoring>
   - Press the **[Shift]** + **[F6]** keys

Each time that the <Register monitoring> menu is selected, the <Register monitoring> item alternates between being shown or hidden.
The user can force a change in the current OM status in the <Register monitoring>.
The following functions can be used to change the OM status.

- To force the bit OM to change to ON or OFF:
  Double-click on the line for the bit OM
- To force a change in the current value of the word OM:
  Double-click on the line for the word OM

Refer to “7.12.3 SET/RST OM” for changing the bit OMs.
Refer to “7.12.4 Change current value” for details about changing the word OMs.

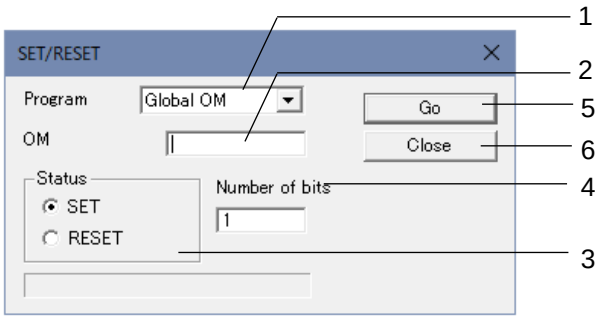
7.12.3 SET/RST OM

The user can force the bit OM status to change from ON to OFF and from OFF to ON.
In addition, the user can reset the current value of the word OM to 0 (zero).

(NOTE 1) When changing OM values online, be sure to take the necessary safety precautions. If an invalid value is set, the device that has the NC(BUILT IN PLC) installed could malfunction.
(NOTE 2) When specifying the number of bits, if the bit number exceeds the last number of the specified OM, then it may adversely affect another OM. Therefore, be careful when specifying the number of bits.
(NOTE 3) If the user specifies too many bits, the processing may take a while.
(NOTE 4) The local timer (LT) coil and the local counter (LC) coil cannot be reset with the OM SET/RST operations.
(NOTE 5) The following word OMs can be reset: BDX, BDY, D, LD, SD, IX, LT and LC.
(NOTE 6) This function does not continuously force the OM to stay ON or OFF. Even if this function is used to change the current value, the OM value will ultimately change based on the program operation or an external input.
(NOTE 7) An OM with an index cannot be specified when changing the current value.
(NOTE 8) This function cannot be used to change the value of a variable.

Operation procedure

1. When monitoring an LD program, perform one of the following operations to display the <SET/RESET> dialog.
   - Double-click on the bit OM in the LD program.
   - Press the **[F9]** keys
   - Select <SET/RST OM> under the <Online> menu.
   - Right-click the mouse and select <SET/RST OM> from the displayed menu.
   - Double-click on a bit OM in <Register monitoring>.
2. In the displayed <SET/RESET> dialog, specify the <Program>, <OM name> and <Number of bits> and select the <Operation>.



No.	Item	Description
1	Program	Specifies the program or global OM. The following selections are available: “Global OM”, “High speed task 1”, “High speed task 2”, “Standard task 1”, “Standard task 2” and “Standard task 3”. A program that is not set in the program execution settings does not appear in the list.
2	OM name	Specifies the bit OM/word OM for changing the status. When the dialog appears with the cursor on a bit OM, that OM (at the cursor position) is automatically inserted.
3	Operation	Select whether to turn ON (SET) or to turn OFF (RESET) the specified OM (in the “OM name” field). RESET is only possible when a word OM is specified in the “OM name” field. In this situation, the user can reset the current value of the word OM to 0 (zero).
4	Number of bits	Specifies how many bits from the bit OM (specified in the “OM name” field) are set to the same status.
5	<Execute> button	Executes the SET or RESET operation.
6	<Close> button	Closes the dialog.

3. Press the <Execute> button or the **[Enter]** key to change the OM status.
4. To change the status of another bit OM, repeat steps 2 and 3.
5. Click on the <Close> button to close the <SET/RESET> dialog.

(NOTE) The execution status of the value change is displayed under the <SET/RESET> dialog.

7.12.4 Change Current Value

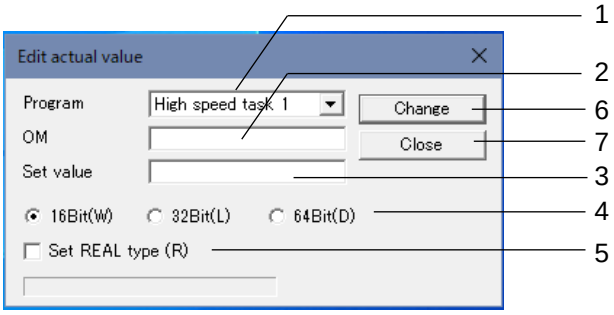
The user can force a change in the current value of a word OM.

(NOTE 1) When changing OM values online, be sure to take the necessary safety precautions. If an invalid value is set, the device that has the NC(BUILT IN PLC) installed could malfunction.
(NOTE 2) The local timer (LT) and local counter (LC) contacts do not turn OFF, even if the current values are forced to reset to “0” on the local timer (LT), where the time expired, and on the local counter (LC), where the count is reached.
(NOTE 3) This function cannot be used to change the set values for the local timer and local counter.
(NOTE 4) An OM with an index cannot be specified when changing the current value.
(NOTE 5) A digit-specified bit OM cannot be specified when changing the current value.
(NOTE 6) This function cannot be used to change the value of a variable.
(NOTE 7) Set a value that does not exceed the last number of the OM when writing a 32-bit value or 64-bit value to the OM.
(NOTE 8) The value cannot be changed for a digit-specified bit OM.

Operation procedure

1. When monitoring an LD program, perform one of the following operations to display the <Change current value> dialog.
   - Double-click on the word OM in the LD program
   - Press the **[Shift]** + **[F9]** keys
   - Select <Change current value> under the <Online> menu
   - Right-click the mouse and select <Change current value> from the displayed menu
   - Double-click on a word OM in <Register monitoring>

2. In the displayed <Change current value> dialog, specify the <Program>, <OM name> and data size to be changed, and enter the <Set value>.
   The user can set the change value to a decimal or hexadecimal. If the value is specified in hexadecimals, add an “H” to the beginning of the numerical value.



No.	Item	Description
1	Program	Specifies the program or global OM. The following selections are available: “Global OM”, “High speed task 1”, “High speed task 2”, “Standard task 1”, “Standard task 2” and “Standard task 3”. A program that is not set in the program execution settings does not appear in the list.
2	OM name	Specifies the word OM for changing the value. Enter an OM name in uppercase letters. When the dialog appears with the cursor on a word OM, that OM (at the cursor position) is automatically inserted.
3	Modified value	Sets the value to change to.
4	Data size	Specifies “16 Bit”, “32Bit” or “64Bit” as the data size for writing. When the checkbox “Set REAL type” is ticked, select either “32Bit” or “64Bit”.
5	Set REAL type	Tick the checkbox to specify a REAL type (real number) for the change value.
6	<Change> button	Executes the current value change.
7	<Close> button	Closes the dialog.

3. Press the <Change> button or the **[Enter]** key to change the current value of the OM.
4. To change the status of another word OM, repeat steps 2 and 3.
5. Click on the <Close> button to close the <Change current value> dialog.

(NOTE) The execution status of the value change is displayed under the <Change current value> dialog.

Executing Completed.

7

7.12.5 Batch Monitoring

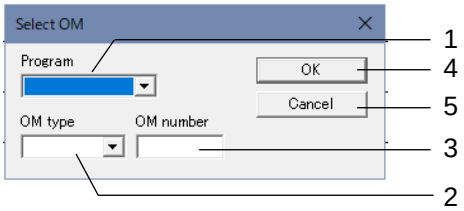
7.12.5.1 Start batch monitoring

In batch monitoring, the user can check the values or status of consecutive OMs in a table format.

Operation procedure

1. Perform one of the following operations to display the <Select OM> dialog for <Batch monitor>.
   - Select <Batch monitor> under the <Online> menu
   - Press the **[F5]** keys
   - Right-click the mouse and select <Batch monitor> from the displayed menu

2. In the displayed <Select OM> dialog, specify the <Program>, <OM name> and <OM number>, and then press the <OK> button or the **[Enter]** key.

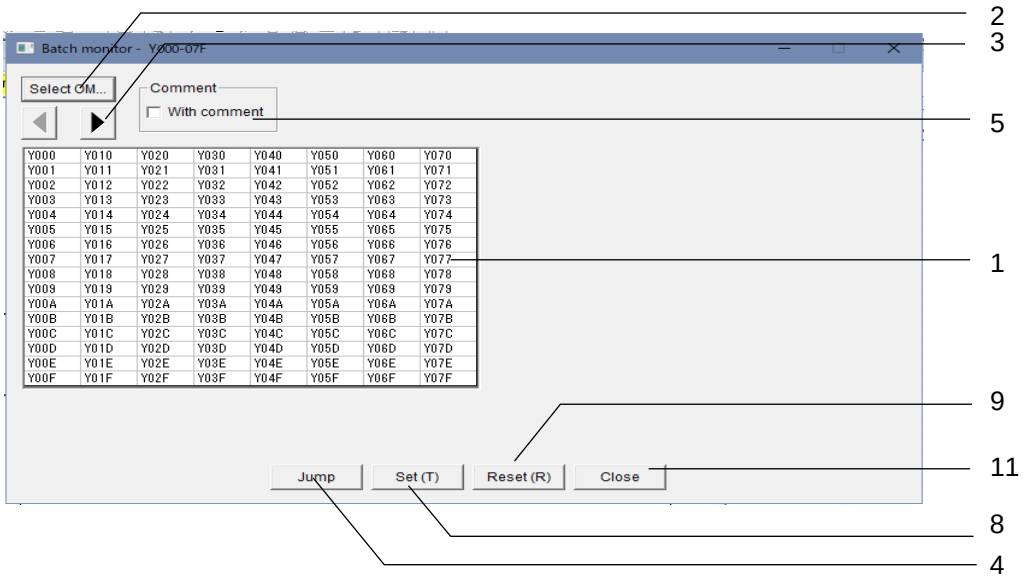


No.	Item	Description
1	Program	Specifies the program or global OM. The following selections are available: “Global OM”, “High speed task 1”, “High speed task 2”, “Standard task 1”, “Standard task 2” and “Standard task 3”. A program that is not set in the program execution settings does not appear in the list.
2	OM name	Selects an OM type from the drop-down list. There is no OM name displayed when no program has been selected.
3	OM number	Specifies the first OM number to start monitoring. When the field is left blank, the number is processed as 0 by default.
4	<OK> button	Sets the selected OM.
5	<Cancel> button	Cancels the OM selection, and the dialog closes.

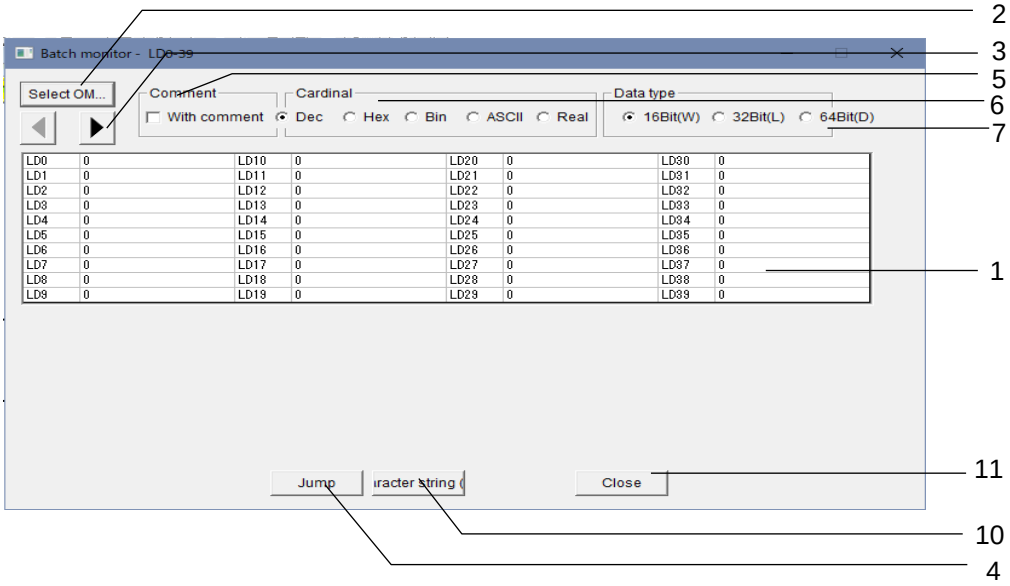
3. Batch monitoring is started and it starts monitoring the specified OMs.

- OM monitoring
  When monitoring OMs, some buttons that display in batch monitoring may be different.

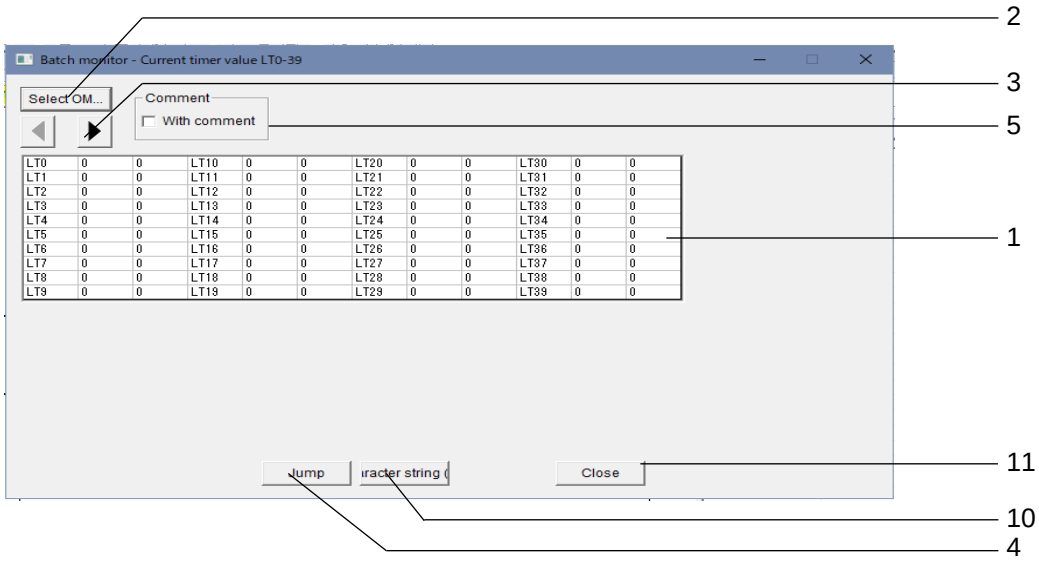
  When monitoring bit OMs



When monitoring word OMs



When monitoring local timers (current values) or local counters (current values)



No.	Item	Description
1	OM display area	Lists the OM names and current values for monitoring. When monitoring bit OMs, the OMs highlighted in yellow are turned ON. When monitoring word OMs, the current values of the OMs are displayed. When monitoring local timers (current values) or local counters (current values), the current values and set values of the OMs are displayed. (Values on the left are the current values, and the ones on the right are the set values.)
2	Select OM	Displays the <Select OM> dialog. Select an OM to display in the list.
3	Change page button	Changes to the next page or previous page in the OM list.
4	<Jump> button	Displays the <Jump> dialog. It designates the first OM number in the list display.
5	Comment	Tick the checkbox to display and show the OM comments.

No.	Item	Description
6	Base number	Changes the base number used to display the word OMs. Select one of the following: Dec (decimal), Hex (hexadecimal), Bin (binary), ASCII (ASCII character string) or Real (real number).
7	Data type	Changes the data type used to display the word OMs. Select either 16-bit, 32-bit or 64-bit.
8	<Set> button	Sets or turns ON the selected bit OM.
9	<Reset> button	Resets or turns OFF the selected bit OM.
10	<Character string> button	Writes the character string data (ASCII character string) to the selected word OM.
11	<Close> button	Closes the batch monitoring.

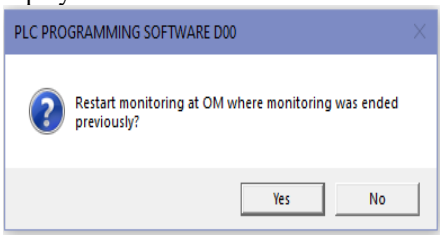
(NOTE 1) The user can display up to four different batches for monitoring at the same time.

(NOTE 2) The PLC Programming software D00 EDITOR can be used and operated even while the batch monitoring is displayed.

(NOTE 3) When the user exits monitor mode in the PLC Programming software D00 EDITOR, the batch monitoring dialog automatically closes. This applies to the following situations.

- When changing from monitor mode to edit mode.
- When closing the edit window while in monitor mode.
- When exiting PLC Programming software D00 EDITOR while in monitor mode.

(NOTE 4) When changing to edit mode while batch monitoring is open, the following dialog is displayed the next time the batch monitor operation is executed.



Selecting <Yes> will automatically display the batch monitoring with the same OMs that were selected for monitoring the previous time.

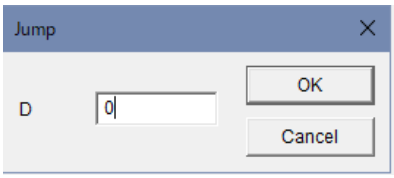
(NOTE 5) The contacts and coils for the local timer and the local counter are processed as bit OMs.

7.12.5.2 Change display

The user can change the display, such as the first address that is shown in the OM display area as well as the base number.

- Next page / Previous page
  Click on the <Change page> button to monitor the addresses on the previous page or following page.

- Jump
  After clicking on the <Jump> button, the <Jump> dialog is displayed.
  Specify the OM number to display on the top, and then click on the <OK> button.
  Click on the <Cancel> button to cancel the jump operation.



- Base number and data type
  When monitoring word OMs, the user can specify the base number and data type.
  When “Hex” is specified for the base number, the display shows the value in hexadecimals with an “H” at the beginning and inserts leading zeros when necessary (such as in “H0123”).
  When “Bin” is specified, the display shows the value in binary and inserts leading zeros when necessary (such as in “00001111000011111”). If “32Bit” or “64Bit” are specified as the data type in the display, the low-order 16 bit data is shown on the upper line.
  When “Real” is specified, the user can only specify either “32Bit” or “64Bit”.

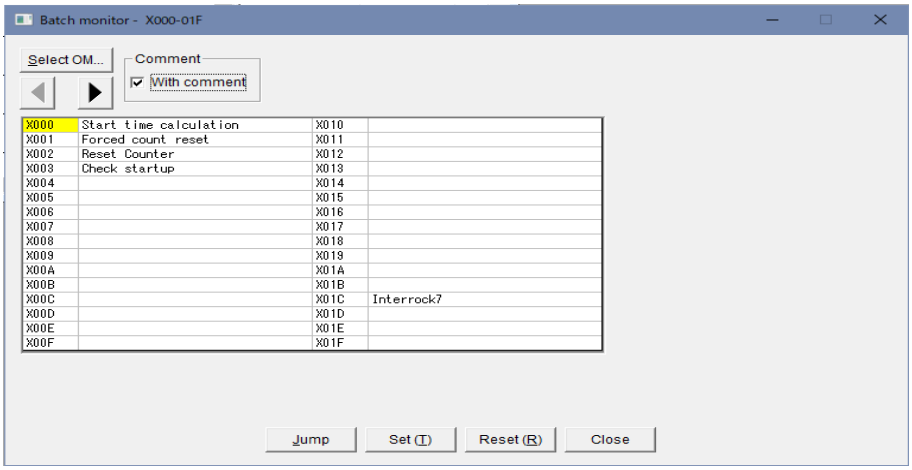
  Display examples

<Dec>, <16bit>		<Dec>, <32bit>		<Dec>, <64bit>	
D0	-1	D0	-1412562945	D0	-6149102338338586625
D1	-21555				
D2	0	D2	-1431699456		
D3	-21846				

<Hex>, <16bit>		<Bin>, <32bit>		<Bin>, <64bit>	
D0	AB	D0	1111 1111 1111 1111	D0	1111 1111 1111 1111
D1	CD		1010 1011 1100 1101		1010 1011 1100 1101
		D2	0000 0000 0000 0000		0000 0000 0000 0000
			1010 1010 1010 1010		1010 1010 1010 1010

<ASCII>, <16bit>		<ASCII>, <32bit>		<Real>, <32bit>	
D0	AB	D0	ABCD	D0	123.45
D1	CD				

- Show with comments
  Tick the checkbox <With comment> to show the OM comments also in the monitoring window.



7.12.5.3 Change OM value

The user can change the OM value from the batch monitoring.

Operation procedure

1. Double-click on the desired OM in the OM display area to change the value.

2. When monitoring bit OMs, the <SET/RESET> dialog displays. When monitoring word OMs, the <Change current value> dialog displays. Specify the new value to change to.
   Refer to "7.12.3 SET/RESET OM" and "7.12.4 Change current value" for details on the dialog operations.

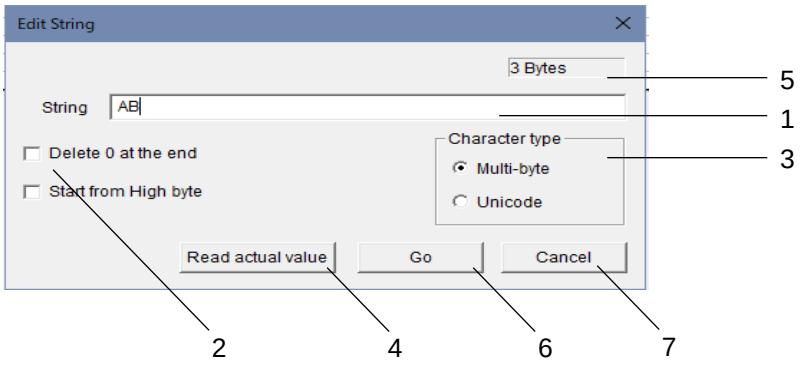
(NOTE) The set values for the local timer and the local counter cannot be changed.

- Character string input
  For word OMs, the user can specify numbers or a character string as the value. When the base number display is changed, the character string can be replaced with an ASCII numerical value.
  Only ASCII characters are compatible for character string input.

Operation procedure

1. Click on an address number for the word OM, and then click on the <Character string> button.

2. The character string change dialog is displayed.



No.	Item	Description
1	Character string	Sets the character string.
2	Delete 0 at the end	Tick the checkbox to not write 0 at the end of the character string. Ex: If "ab"(H6261) is stored in D0 and "c" is written, the following shows the results when using this function. • Checkbox not ticked "H0063"(c) is stored in D0. • Checkbox ticked "H6263"(cb) is stored in D0.
3	Character type	Only <Multibyte character code> can be used.
4	<Read actual value> button	Reads the current value as character string data.
5	Number of bytes	Indicates the number of bytes or the data size that is written to an OM.
6	<Execute> button	Writes the character string data.
7	<Cancel> button	Cancels the character string input, and the dialog closes.

7.12.6 Clear all OMs & Variables

This function clears all the values and status data for the OMs and variables at once (except for the following OMs: X, Y, BX, BY, BDX, BDY, SM and SD). In addition, the user can also reset all the variables to their default values at once.
The set values for the local timer and the local counter cannot be cleared.
The OM/variable all clear operation can only be executed if the NC(BUILT IN PLC) is in a stopped state.

(NOTE) The OM/variable all clear operation applies to all OMs except for the following: X, Y, BX, BY, BDX, BDY, SM and SD. However, when the LD program is running, be aware that the status of Y, BY, BDY, SM and SD will change based on changes in the conditions.

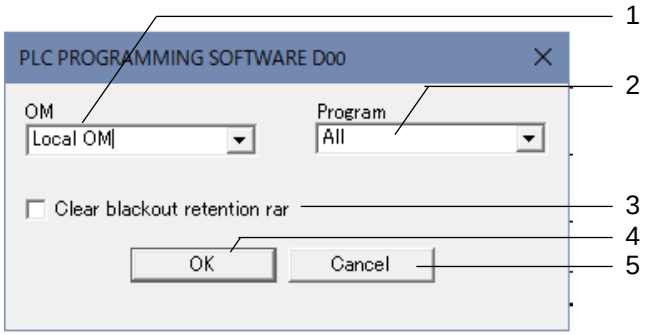
The following shows the procedure for the OM/variable all clear operation.

Operation procedure

1. Change the NC(BUILT IN PLC) to a STOP state.
   Refer to “5.13 Run & Stop in NC(BUILT IN PLC)” for details on changing it to a STOP state.

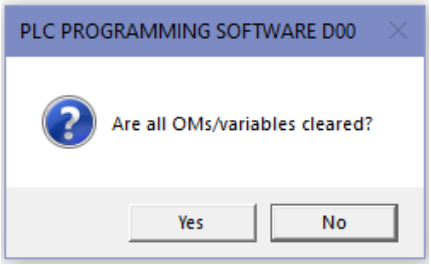
2. Use one of the following operations to “Clear all OMs/variables”.
   - Select <Clear all OMs/variables> under the <Online> menu.
   - Press the **[Ctrl]** + **[F9]** keys

3. In the following dialog, select the <OM>, <Program> and <Clear power failure hold area> as necessary, and then click on the <OK> button.

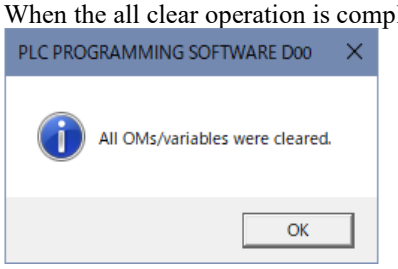


No.	Item	Description
1	OM	Specifies the type of OMs to clear. “All”, “Global OM” and “Local OM” can be selected.
2	Program	Specifies the program. The following selections are available: “All”, “High speed task 1”, “High speed task 2”, “Standard task 1”, “Standard task 2” and “Standard task 3”. A program that is not set in the program execution settings does not appear in the list.
3	Clear power failure hold area	Tick the checkbox to clear the power failure hold area. To keep the data in the power failure hold area, do not tick the checkbox.
4	<OK> button	Enters the settings.
5	<Cancel> button	Cancels the process, and the dialog closes.

4. Select <Yes> when the following message is displayed to clear all the OMs/variables.
   Select <No> to cancel the all clear operation.



5. When the all clear operation is complete, the following message is displayed.

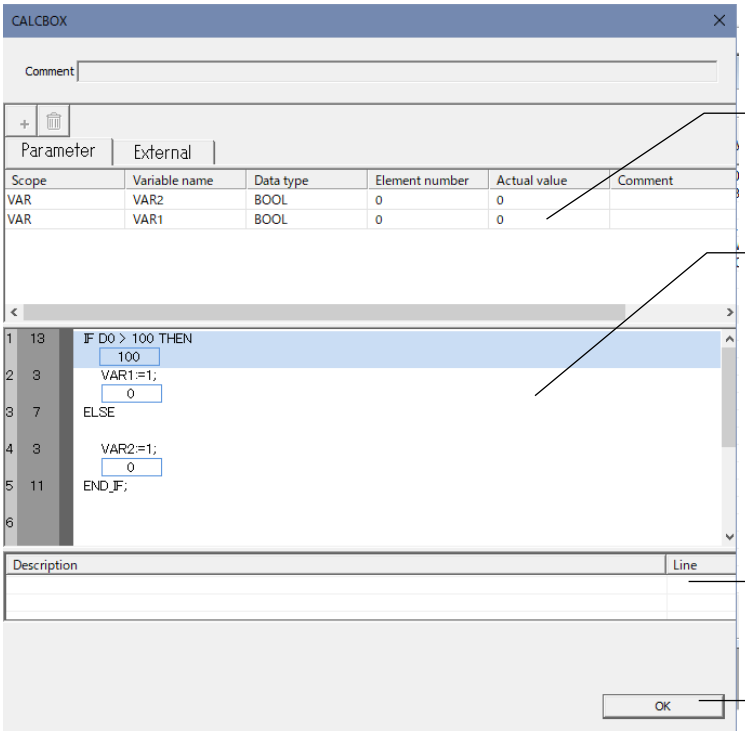


7.12.7 Monitor CALC Commands

When monitoring an LD program, the user can monitor programs that are set to CALC commands in the PLC Programming software D00 LD/ST Programming Editor.

Operation procedure

1. Double-click on a CALC command in the edit window while monitoring.
2. The <CALCBOX> dialog is displayed.



Displays current value of variables being used in CALC command.

Displays program execution status and current value of OMs/variables.

Displays error information when there is an error in the CALC command.

Closes the dialog.

The <CALCBOX> dialog operations during monitoring are the same as the monitor operations for the ST program. Refer to “8.11 Monitoring operations" for further details.

7.13 DEBUG Functions

This section describes the “DEBUG function” used for checking the LD program operation.
The “DEBUG function” can only be used with standard task programs. It is possible to use it on a high speed task only when executing a simulation. As a result, when a high speed task program window is at the forefront, the debug function menu is disabled and cannot be selected. The only exception is when executing a simulation.
The user can perform the following LD program check operations with the “DEBUG function”.

- Step execution
  Executes only one command in the LD program and then stops.

- LD program stop when OM status changes
  LD program stops when the status of the specified OM changes.

- LD program stop at specified circuit block
  LD program stops at the circuit block of the specified step number.

- Monitoring stop by user or by condition
  Monitoring is stopped when the user presses the **[Esc]** key or when a specified OM changes.

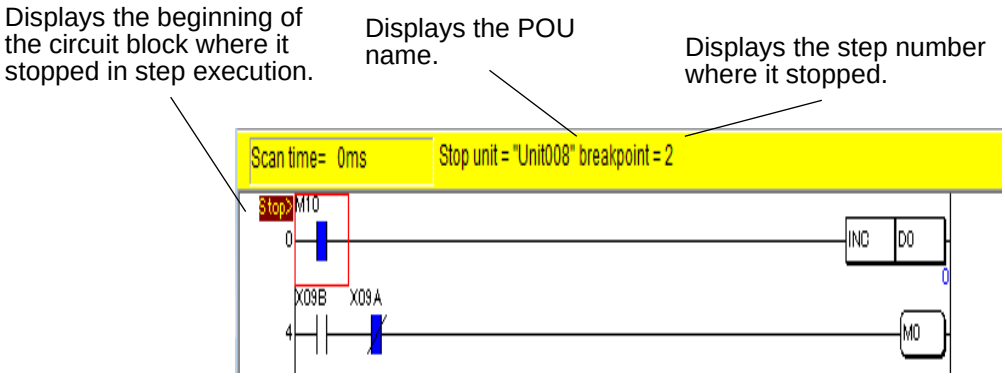
Refer to “11.5 Debug Function” for further details on each comment.

7.13.1 Step Execution

This section describes step execution operations.
The user can execute the program in steps when the LD program is in a stopped state.

(NOTE 1) Step execution is not possible when the LD program monitoring is not executed or when monitoring is stopped.
(NOTE 2) When running multiple high speed or standard control programs in the NC(BUILT IN PLC), step execution is carried out in the program being monitored at the forefront of the PLC Programming software D00 EDITOR. If there are too many processes active for other programs, when the program is executed in DEBUG-RUN and the processing returns to the program being monitored, it will change to a STOP state.

When the <RUN/STOP mode> dialog is used to stop the LD program, it executes from the beginning of the LD program.
When the LD program is stopped based on the conditions that are set in “7.13.2 Stop condition settings and execution”, then it executes from the stop position.



The following shows the step execution procedure.

Operation procedure

1. Start LD program monitoring.

2. Stop the LD program.
   Refer to “5.13 Run & Stop in NC(BUILT IN PLC)” for details on the stop operations using the <RUN/STOP mode> dialog.
   Refer to “7.13.2 Stop condition settings and execution” for details on stopping the LD program based on the condition settings.

3. Use one of the following operations to perform “Step execution”.
   - Select <Execute step> under <Debug> in the <Online> menu
   - Click on the button in the tool bar.

4. Repeat step 3 to execute the LD program per command.

Change the setting in the <RUN/STOP mode> dialog to RUN or DEBUG-RUN in order to cancel the step execution.

7.13.2 Stop Condition Settings and Execution

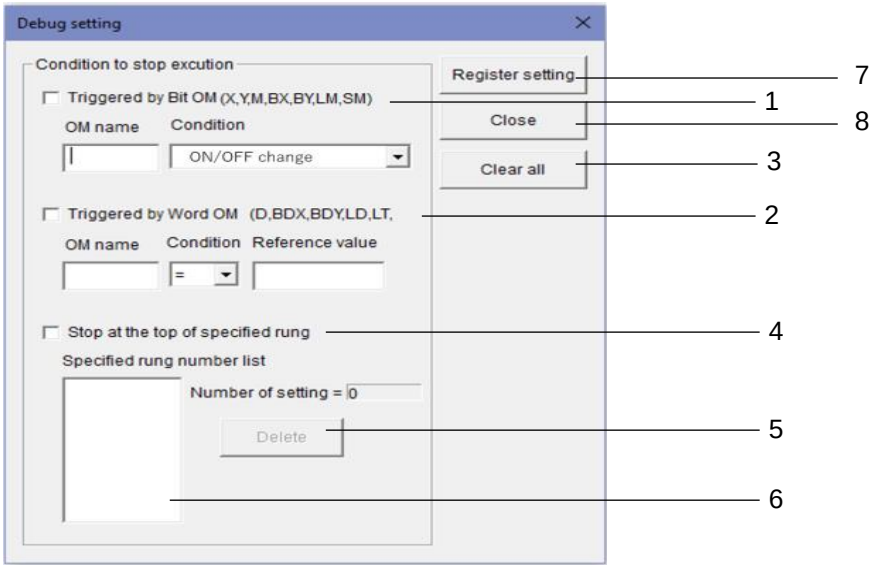
This section describes how to set an OM status change to stop the LD program and how to stop the LD program at a specified circuit block.
The user can set conditions to stop the LD program when the LD program is in a stopped state.
(NOTE) The LD program cannot be stopped using stop conditions when the LD program monitoring is not executed or when monitoring is stopped.

7.13.2.1 Stop condition settings

The user can set the conditions to stop the LD program as follows.

- Stopping LD program when status of bit OM or word OM changes
  Set the conditions in the <Debug condition setting> dialog.

  Operation procedure
  1. Select <Debug condition setting> under <Debug> in the <Online> menu.
  2. The <Debug condition setting> dialog is displayed.



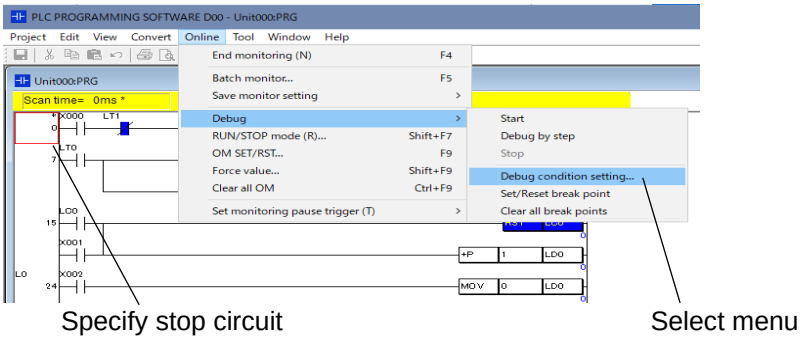
No.	Item	Description
1	Bit OM trigger	Tick the checkbox to use bit OM conditions to stop the LD program. “OM name”: Specifies the bit OM as a stop condition. “Condition”: Selects “OFF to ON”, “ON to OFF” or “ON/OFF Reverse” as the stop condition.
2	Word OM trigger	Tick the checkbox to use word OM conditions to stop the LD program. “OM name”: Specifies the word OM as a stop condition. “Condition” and “Reference value”: Sets a comparison (=, <>, >, >=, < or <=) between the current OM value and a reference value as the stop condition.
3	Stop at the top of specified rung	Tick the checkbox to stop the LD program right before the step execution that is displayed in the “Specified rung number list”. When a stop step is registered, the checkbox is automatically ticked. When the checkbox is not ticked, the LD program does not stop at the stop step. When registering a stop step, refer to “Specify circuit block to stop at beginning of circuit block” (described later).
4	Specified rung number list	Lists the stop steps that are registered. When registering a stop step, refer to “Specify circuit block to stop at beginning of circuit block” (described later).
5	<Delete> button	Deletes the stop step that is selected in the “Specified rung number list”.
6	<Clear all> button	Clears all the settings in the <Debug condition setting> dialog.
7	<Register setting> button	Registers all the settings in the <Debug condition setting> dialog.
8	<Close> button	Closes the <Debug condition setting> dialog.

7

3. Click on the <Register setting> button.
   When the user does not click on the <Register setting> button and closes the <Debug condition setting> dialog, the changed settings become invalid.
4. Click on the <Close> button to close the <Debug condition setting> dialog.

- Specify circuit block to stop at beginning of circuit block

  Operation procedure
  Move the cursor to the desired circuit block, and perform one of the following operations.
  - Move the cursor to the desired circuit block, and select <Set/Reset breakpoint> under <Debug> in the <Online> menu.
  - Double-click at the corresponding step number to register the desired circuit block.

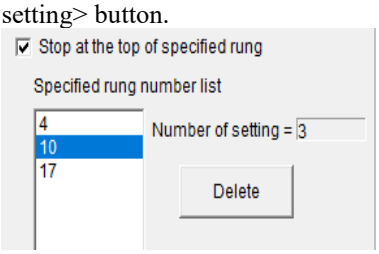


Double-click on the step number

After setting the circuit block, it automatically sets to the <Specified rung number list> in the <Debug condition setting> dialog.

(NOTE) The user cannot add a stop step in the <Specified rung number list> in the <Debug condition setting> dialog.

The user can perform one of the following operations to remove the stop setting at a step number.
To cancel only a specific setting:
- Select <Set/Reset breakpoint> under <Debug> in the <Online> menu.
- Select the step number to remove from the <Specified rung number list> in the <Debug condition setting> dialog, click on the <Delete> button and then click on the <Register setting> button.



To cancel all settings:
- Select <Clear all breakpoints> under <Debug> in the <Online> menu.
- Untick (no check) the checkbox <Stop at the top of specified rung> in the <Debug condition setting> dialog, select all registered step numbers, click on the <Delete> button and then click on the <Register setting> button.
- Click on <Clear all> in the <Debug condition setting> dialog, and then click on the <Register setting> button.
  However, when the <Clear all> button is selected, all other stop conditions are cleared as well.

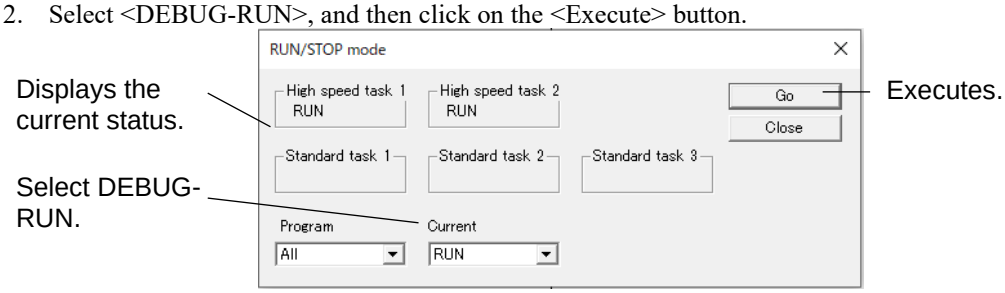
7.13.2.2 LD program debug execution

Use one of the following operations to debug an LD program.
When the set conditions are met, the LD program stops.

Execution using <RUN/STOP mode> dialog

<u>Operation procedure</u>
1. Perform one of the following operations to display the <RUN/STOP mode> dialog.
   - Select <RUN/STOP mode> under the <Online> menu.
   - Press the **[Shift]** + **[F7]** keys

2. Select <DEBUG-RUN>, and then click on the <Execute> button.



Execution without using <RUN/STOP mode> dialog

Operation procedure
Perform one of the following operations to execute debugging when the NC(BUILT IN PLC) or simulation is in a stopped state.
- Select <Execute> under <Debug> in the <Online> menu
- Click on the button in the tool bar.

Perform the same operation again in order to skip to the next location with a stop condition.
Perform step execution to only execute the next command (after the stop position) and then stop.

To cancel the "DEBUG-RUN" state, change the setting in the <RUN/STOP mode> dialog to <RUN>.
All the stop conditions are cancelled, and the LD program executes.

7.13.3 Stop Monitoring

This function stops only the program monitoring but keeps executing the NC(BUILT IN PLC).

7.13.3.1 Pause monitoring

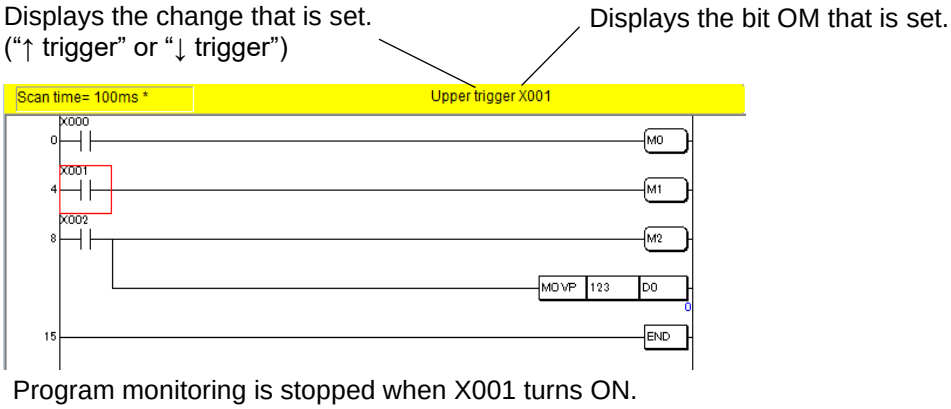
The LD program monitoring is stopped with the following operation.

Operation procedure
Press the **[ESC]** keys

To re-start the monitoring, press the **[F2]** key. The user can also scroll through the program edit window.

7.13.3.2 Stop monitoring using bit OM change setting

The user can stop the program monitoring using a trigger setting when the bit OM value changes.



Program monitoring is stopped when X001 turns ON.

Stop program monitoring when OFF changes to ON (Rising edge trigger)

Operation procedure
1. Move the cursor to the bit OM to set a monitoring stop trigger (during the monitoring execution).
2. Press the **[Ctrl]** + **[F7]** keys while the monitoring is executing.
   Or, select <Rising edge trigger> under <Set monitoring pause trigger> in the <Online> menu.

Stop program monitoring when ON changes to OFF (Falling edge trigger)

Operation procedure

1. Move the cursor to the bit OM to set a monitoring stop trigger (during the monitoring execution).
2. Press the **[Ctrl]** + **[F8]** keys while the monitoring is executing.
   Or, select <Falling edge trigger> under <Set monitoring pause trigger> in the <Online> menu.

When the cursor is moved to a bit OM used to specify digits, the first OM is targeted for monitoring. For example, when the cursor is at “M16:8”, the M16 value change is monitored.
When the cursor is moved to a bit OM used for index addressing, OMs that are not indexed are targeted for monitoring. For example, when the cursor is at “M32I0”, the M32 value change is monitored regardless of the index IX0 value.

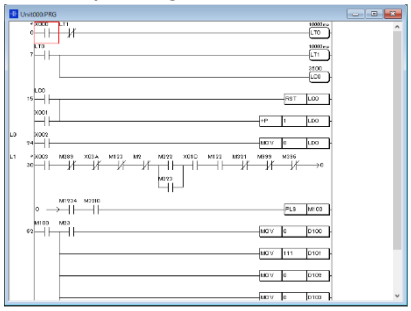
While performing monitoring in the PLC Programming software D00 EDITOR, only one bit OM can be monitored at a time regardless of the language or POU.
For example, after setting one bit OM for monitoring, if the user sets another bit OM as the bit OM for monitoring in another POU, then the last bit OM that was set is enabled for monitoring.

Press the **[F2]** key to cancel the monitoring stop setting (using the bit OM).
In addition, when monitoring is opened again, the bit OM setting for monitoring is cleared.

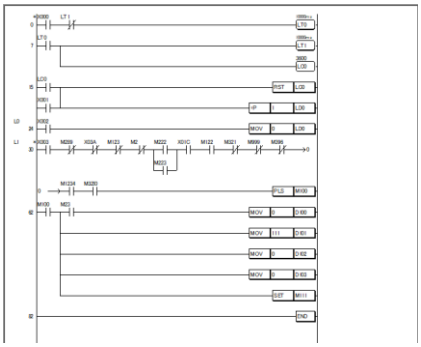
7.14 Print

This section describes the operations to print an LD program.
When printing the LD program, the user can print the program with OM comments, coil remarks, inline comments and label comments.
The display for the PLC Programming software D00 LD/ST Programming Editor changes to a preview of the printing data and then prints.

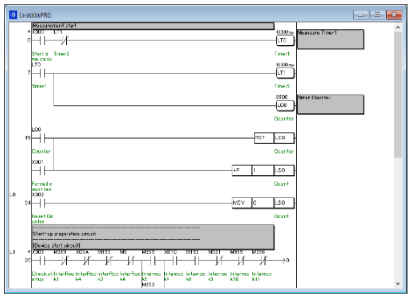
When printing without comments



Print



When printing with all comments



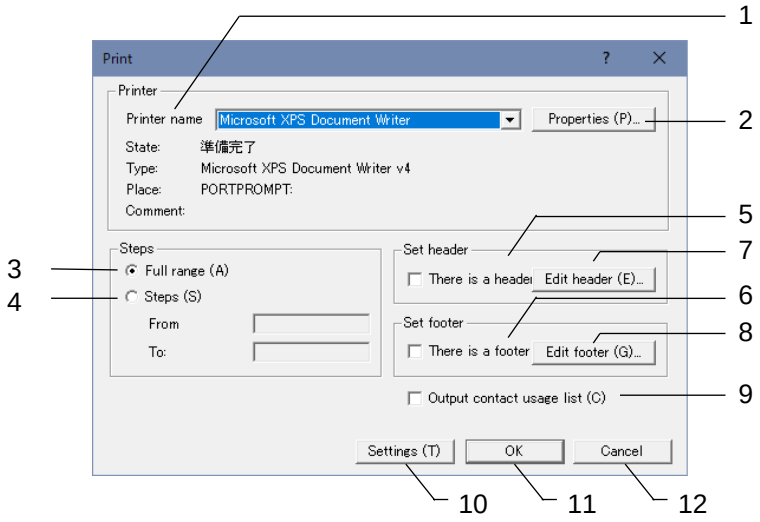
Print

We recommend printing with the paper set to landscape when printing with coil remarks.

In the <Print> dialog, the user can select and set the following when printing: paper size, paper orientation, printing range of the LD program, header and footer settings as well as the contact list.
The user can configure the print settings for each program.

Operation procedure

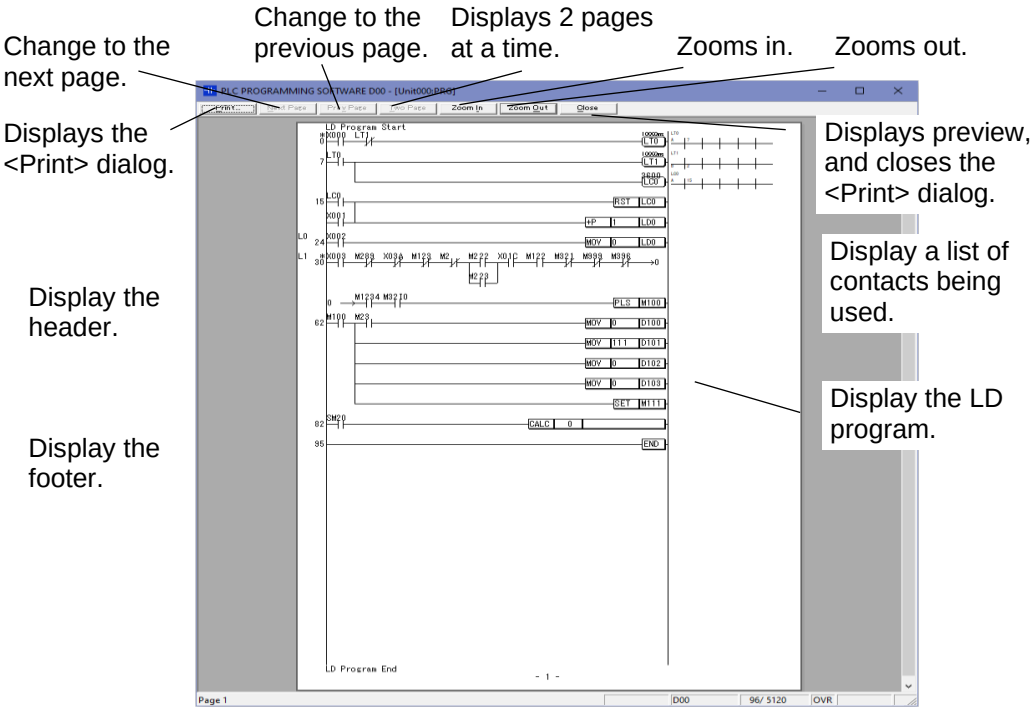
1. Set the LD program display to the desired format for printing (i.e. show comments, etc.).
2. Select <Print> from the <Project> menu.



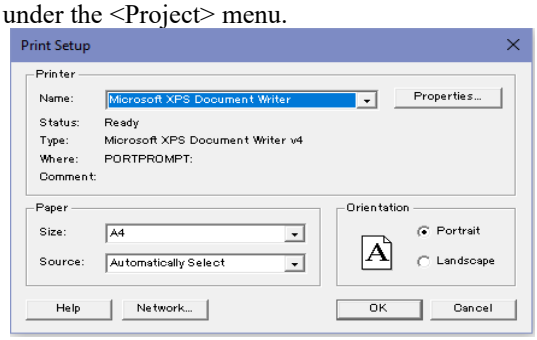
No.	Item	Description
1	Printer name	Selects a printer.
2	<Properties> button	Opens the properties dialog for the printer. (The settings depend on the printer drivers that are installed.)
3	All	Prints the entire range.
4	Steps	Prints the range that is specified by the first step number and last step number.
5	Header	Tick the checkbox to include the header.
6	Footer	Tick the checkbox to include the footer.
7	Edit header	Opens the header edit dialog.
8	Edit footer	Opens the footer edit dialog.
9	Output contact usage list	Tick the checkbox to output the coil usage list on the right side margin of the LD program.
10	Set	Saves the header and/or footer information.
11	<OK> button	Saves the header and/or footer information and starts printing.
12	<Close> button	Closes the dialog.

3. Select a printer in the <Print> dialog.
4. Specify the paper settings and other items under <Properties> in the <Print> dialog.
5. Under the <Specify area> section, select a printing range from the following options: <All> or <Steps>. When <Steps> is selected, specify the first and last step numbers in the range.
6. Set the <Header setting>, the <Footer setting> and/or the <Output contact usage list> when necessary.
7. Click on the <OK> button in the <Print> dialog to start printing.

In the <Project> menu, the user can use <Print preview> to check the image being printed on the paper.



(NOTE 1) The user can also set the printer, paper size and orientation in the <Printer setup> under the <Project> menu.



(NOTE 2) The print preview cannot be used in the search results display. When using the print preview, change to the step number display.

(NOTE 3) When the printing content does not fit on one page, a page break is automatically inserted between circuit blocks so a circuit block is not cut off on the printed page. If one circuit block does not fit onto one page, then a page break is inserted into the middle of the circuit block.

(NOTE 4) If the printing content is too wide to fit onto one page, it will print and the content will cut off on the right edge. The remaining part will be printed on the next page. The user can check the range that is printed onto one page in print preview. If the content does not fit onto one page, adjust the width of the column in the <Option settings> from the <Auxiliary> menu in the PLC Programming software D00.

(NOTE 5) Printing is not possible for a program with data that is not converted.

(NOTE 6) The header and footer settings are configured in the dialog below. The user can enter a maximum of 512 characters. However, any text after 5 lines is ignored when printing. Therefore, do not exceed the line limit when setting. When the user clicks on the <Set> button or the <OK> button in the <Print> dialog, the editing is saved as part of the program files.

When <OK> is selected, the editing is set and it goes back to the <Print> dialog. When <Cancel> is selected, the editing is discarded and it goes back to the <Print> dialog.

7.15 Display Manual

This section describes how to display a manual.
The user can display the different manuals in the PLC Programming software D00.

To display the manuals, the user must install an application software beforehand that can display PDF files, such as Acrobat Reader provided by Adobe.

- User's manual

Operation procedure
Select the <User's manual> under <Manual> in the <Help> menu.

- Operation manual

Operation procedure
Select the <Operation manual> under <Manual> in the <Help> menu.

- Programming manual

Operation procedure
Select the <Programming manual> under <Manual> in the <Help> menu.

CHAPTER 8

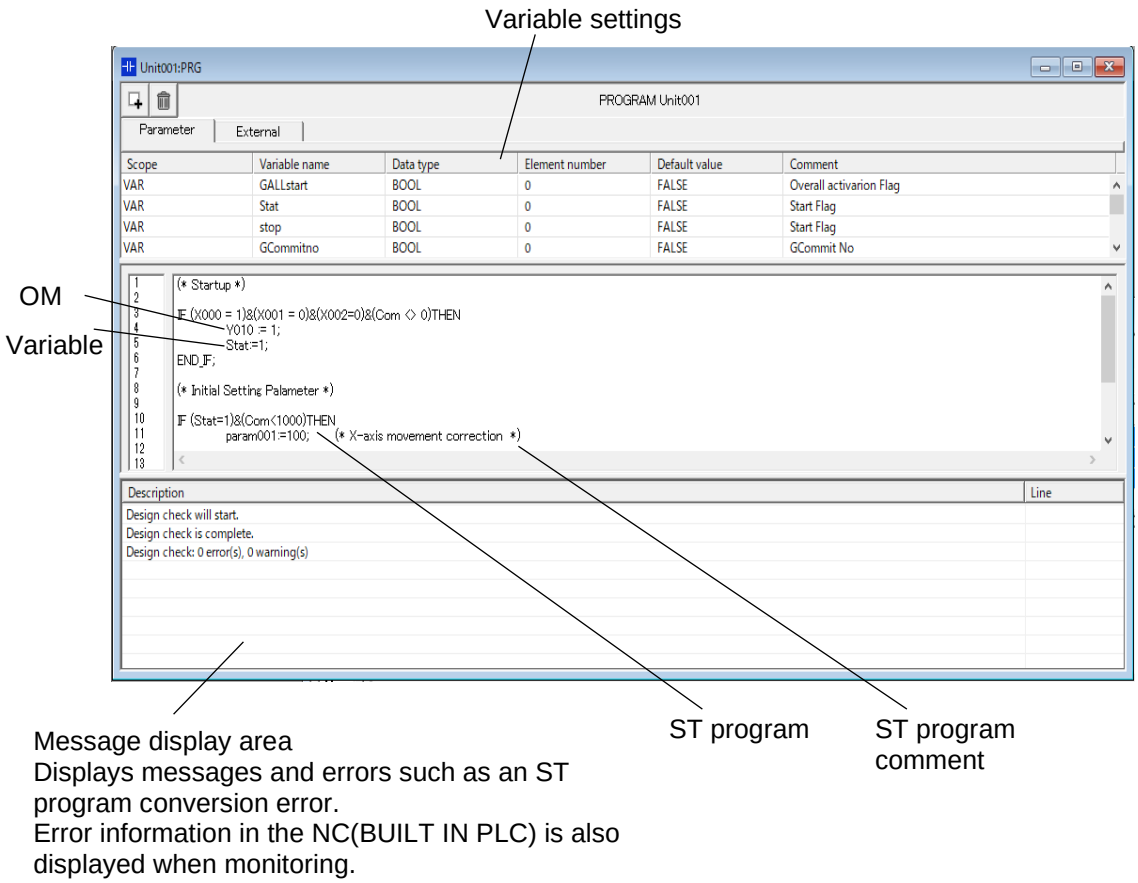
ST EDIT OPERATION PROCEDURE

8.1 ST Program & Edit Screen Layout
8.2 Variable Settings
8.3 Edit ST Programs
8.4 Copy, Cut & Paste Character Strings
8.5 Replace
8.6 Edit Lock
8.7 Search
8.8 Launch Simulation
8.9 NC(BUILT IN PLC) Monitoring
8.10 Run & Stop in NC(BUILT IN PLC)
8.11 Monitoring Operations
8.12 DEBUG Function
8.13 Print
8.14 Display Manuals

8.1 ST Program & Edit Screen Layout

This section describes the ST program, its display and the layout of the edit screen. In addition, this section also describes the common operations and settings.

The screenshot below shows the layout of the ST program display.
Refer to “Chapter 12 Commands (ST)” in the PLC System Manual for further details.



The user can change the font in the ST program display.
Refer to “5.6.2.1 Edit screen adjustment (option settings)” for details on the operations.

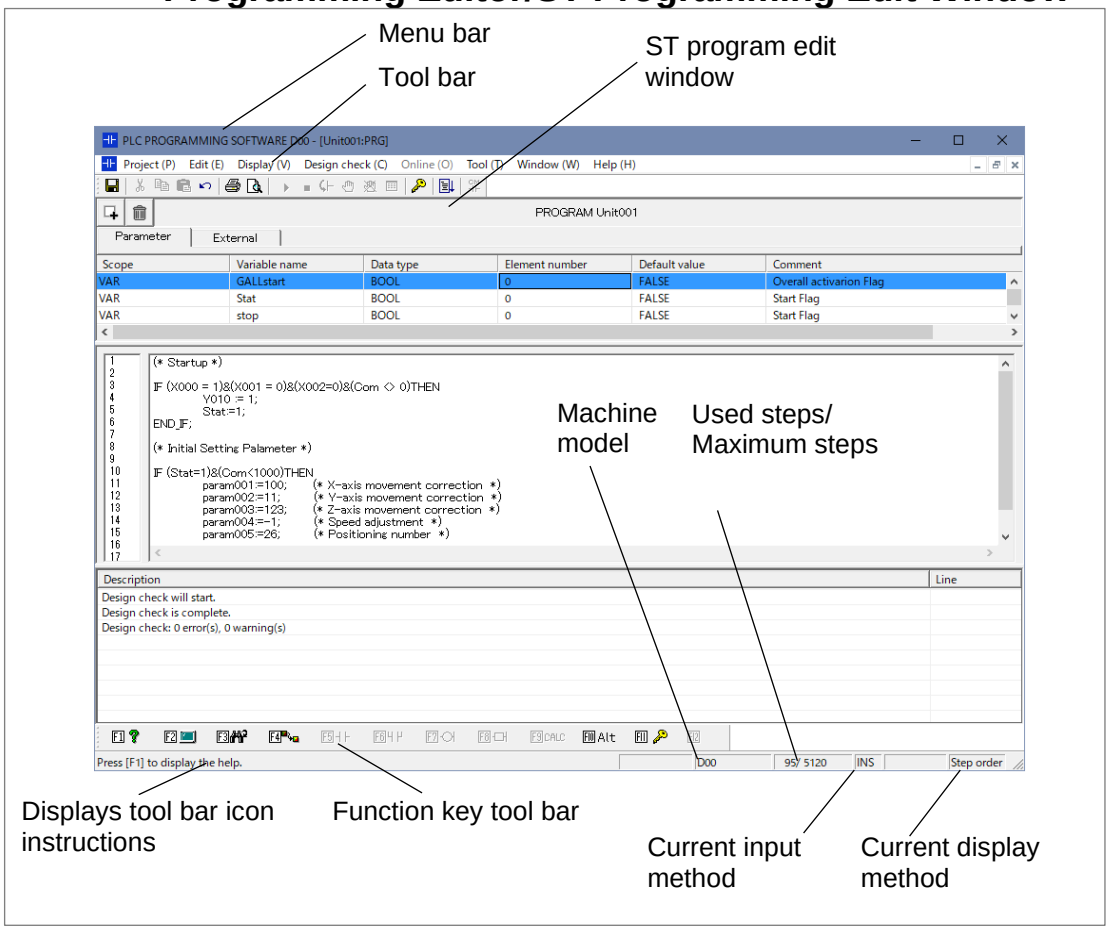
8.1.1 Scroll Through Edit Screen

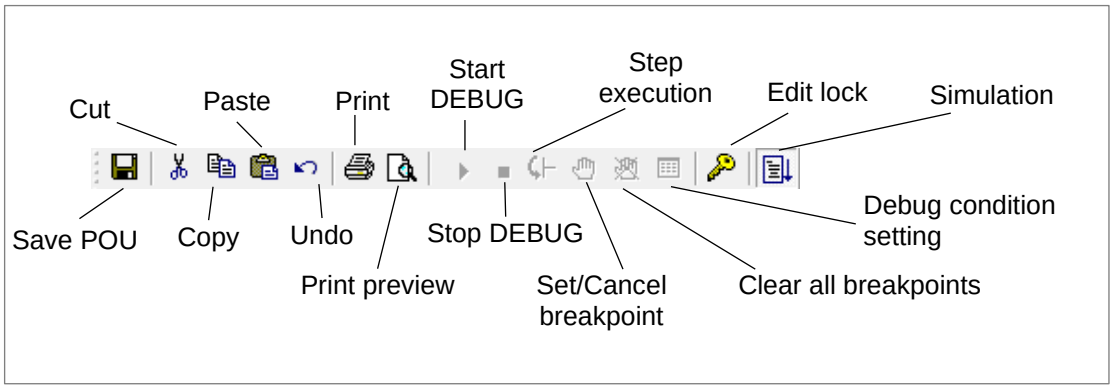
When displaying the ST program, the user can scroll and search through the ST program content that is not displayed in the ST program edit window.
Refer to “8.7 Search” for details on the search operations.

Scrolling through the screen can be executed using the following operations.
- Use mouse to perform scroll bar operations on the right side of the screen
- Use scroll wheel on mouse to perform operations

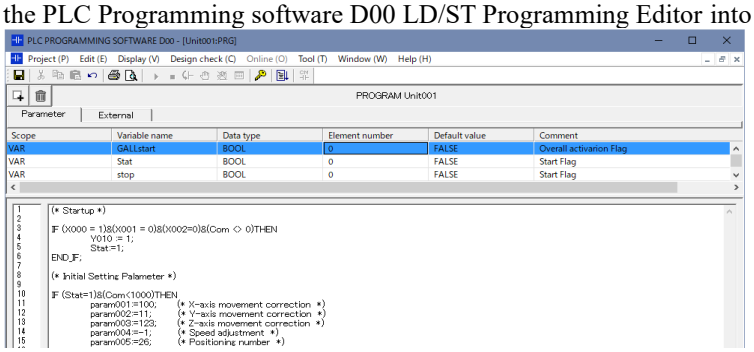
The arrow keys and other keys can also be used to execute the following operations.
- Show lines below current screen display : Moves cursor to last line on current screen and press **[↓]**
- Show lines above current screen display : Moves cursor to first line on current screen and press **[↑]**
- Scroll up, jumping to previous full screen : **[Page Up]**
- Scroll down, jumping to next full screen : **[Page Down]**
- Show screen from first line : **[Home]**, **[Ctrl]**+**[Page Up]**
- Move to last line : **[Ctrl]**+**[Page Down]**

8.1.2 Layout of PLC Programming Software D00 LD/ST Programming Editor/ST Programming Edit Window





When the ST program edit window is shown in the full screen display, the window merges with the PLC Programming software D00 LD/ST Programming Editor into one screen display.



8.2 Variable Settings

This section describes how to create a variable to be used in an ST program.
Variables are set for each POU.

8.2.1 Create & Edit Parameter Variables

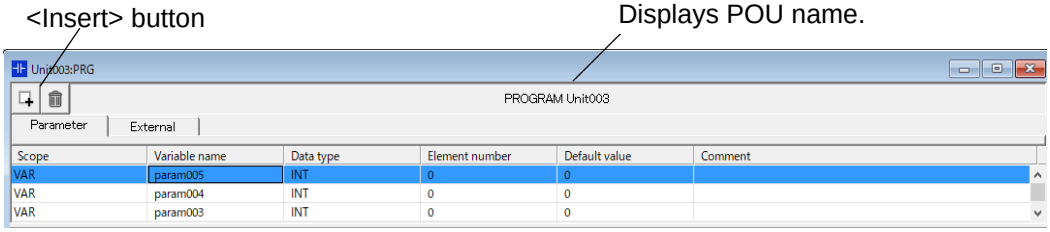
8.2.1.1 Create new parameter variable

The user can perform the following operations in the variable editing area to create a new parameter variable.

Operation procedure

Perform one of the following operations under the <Parameter> tab.

- Click on the <Insert> button.
- Right-click the mouse on the variable setting field, and then select <Insert> from the context menu.



The following number of variables can be set for each POU.

BOOL : 1024 variables

Non-BOOL : 1024 variables (The number of variables that can be used varies according to the type.)

Type	Qty. equal to one BOOL variable	Type	Qty. equal to one BOOL variable	Type	Qty. equal to one BOOL variable
BYTE	1	WORD	1	DWORD	2
INT	1	DINT	2	LINT	4
UINT	1	UDINT	2	ULINT	4
REAL	2	LREAL	4	DATE	4
TEME	4	TOD	4	DT	4
STRING	16				

For example, when only registering WORD type variables, the user can register a maximum of 1024 variables. However, when only registering DWORD type variables, the user can register a maximum of 512 variables

When registering arrays, the total element number for each array that can be registered cannot exceed the maximum number of BOOL type or non-BOOL type variables.

Refer to “12.1.6.1 List of data types” in the PLC System Manual and “8.2.1.2 Parameter variable settings” for further details on data types.

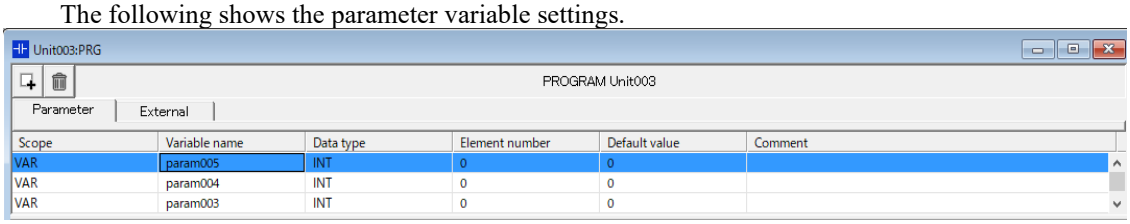
After registering 1024 BOOL type variables, perform the following procedure to register non-BOOL type variables.

Operation procedure

1. Delete a BOOL type variable. Or, change it to another data type.
   (Write down the variable settings before deleting or changing them.)
2. Register a non-BOOL type variable.
3. Re-register the variable that was deleted in step 1. Or, return the data type (that was changed in step 1) back to the BOOL type.

8.2.1.2 Parameter variable settings

The following shows the parameter variable settings.



Item	Description
Scope	Selects a variable range from the following. • VAR : Variable only used within POU • VAR_OUTPUT : Variable where output value to POU (call source) is assigned • VAR_INPUT : Variable where input value from call source is assigned when POU is called • VAR_IN_OUT : Variable where input value from call source is assigned and where output value to POU (call source) is assigned when POU is called "VAR" is fixed in a standard control program.
Variable name	Sets the variable name. • The user can set a maximum of 32 characters. • Single-byte alphanumeric characters and a half-width underscore (_) can be used. • Numbers cannot be used for the first character. • The name of another parameter variable cannot be duplicated in the same POU. • The variable name cannot be the same as any reserved words.
Data type	Selects the variable data type from the following. BOOL/BYTE/WORD/DWORD/INT/DINT/LINT/UINT/UDINT/ULINT/REAL/LREAL/DATE/TIME/TOD/DT/STRING
Element number	Sets the element number of the array between 1 and 1024 when being used as an array variable. The number is set to 0 when it is not being used as an array variable.
Default value	Sets the default value of the variable when the NC(BUILT IN PLC) is launched. The values that can be set vary depending on the data type. Refer to "12.1.6.1 List of data types" in the PLC System Manual for further details. This is the default value for all elements when using an array.
Comment	No more than 32 characters can be used for a comment.

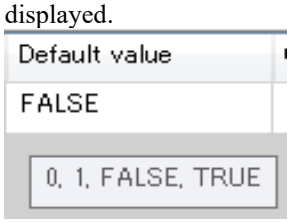
(NOTE 1) After registering a variable, the user can change the settings of a variable being used in an ST program. However, the program can become invalid depending on the content that was changed (for example, when a change is made to a data type that cannot be used in a function).
Therefore, proceed carefully when changing the settings of a variable being used in an ST program.

(NOTE 2) If a variable name is changed for a variable being used in an ST program, that change will not be automatically reflected in the ST program. Change the variable name manually in the ST program.

(NOTE 3) When a default value is set for a REAL or LREAL type variable, then the set value is rounded off to the valid digits of that type (7 digits and 15 digits, respectively).
Examples are shown below.

Data type	Input value	Set value
REAL type	1234567	1234567
	12345678	1.234568E+007
	0.0001234567	0.0001234567
	0.00012345678	0.0001234568
LREAL type	123456789012345	123456789012345
	1234567890123456	1.23456789012346E+015
	0.000123456789012345	0.000123456789012345
	0.0001234567890123456	0.000123456789012346

(NOTE 4) When the mouse cursor is moved to the <Default value> field in the variable editing area, then the valid input range according to the data type of that variable is displayed.



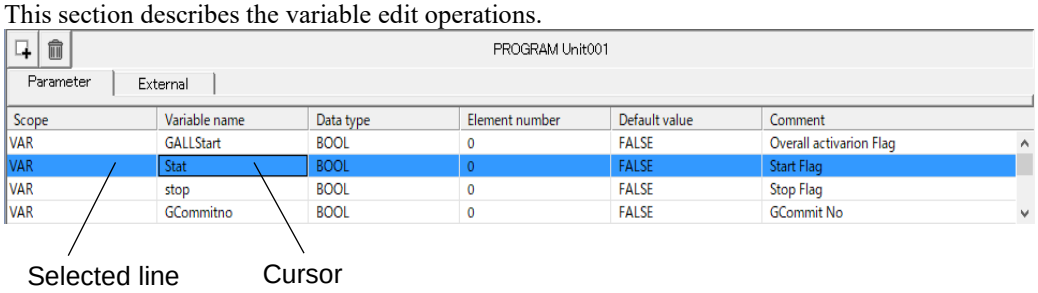
(NOTE 5) The default value of the array variable is the default value for all the elements of that array variable. To set individual values for each element, the user must set them in the program.

(NOTE 6) Refer to "12.1.6.1 List of data types" in the PLC System Manual for further details on data types.
Refer to "Chapter 12 Commands (ST)" in the PLC System Manual for details about reserved words.

(NOTE 7) The user can enter a user or system defined function name for the variable name, but an error will trigger at conversion.

8.2.1.3 Variable edit operations

This section describes the variable edit operations.



8

To edit a variable, move the cursor to the edit location, press the **[Space]** key, or use the mouse to double-click at the cursor position. The position of the cursor becomes editable.

The cursor can be moved using the following operations.

- Single-click on mouse
- Press the **[→]**, **[←]**, **[↑]** or **[↓]** key
- **[Ctrl]+[→]** : Moves to <Comment> field on a line
- **[Ctrl]+[←]** : Moves to <Scope> field on a line
- **[Ctrl]+[↑]** : Moves up to first line in a column
- **[Ctrl]+[↓]** : Moves down to last line in a column
- **[Enter]** : Moves down one line in a column
- **[Tab]** : Moves to next field (column)
  Moves to <Scope> field on next line when at <Comment> field
- **[Shift]+[Tab]** : Moves to previous field (column)
  Moves to <Comment> field on previous line when at <Scope> field
- **[Page Up]** : Scrolls up, jumping to previous full screen
  Moves up to first line if scroll bar is not displayed in variable editing area
- **[Page Down]** : Scrolls down, jumping to next full screen
  Moves down to last line if scroll bar is not displayed in variable editing area
- **[Home]** : Moves to <Scope> field on a line
- **[Ctrl]+[Home]** : Moves to <Scope> field on first line
- **[End]** : Moves to <Comment> field on a line
- **[Ctrl]+[End]** : Moves to <Comment> field on last line

- When <Scope> or <Data type> is in an editing state
  - **[Enter]**, **[Tab]** and **[Shift]** + **[Tab]** keys : Sets selected item in list.
  - **[↓]**, **[↑]** : Selects item in list
  - **[Home]** : Selects first item in list
  - **[End]** : Selects last item in list

- When <Variable name>, <Default value> or <Comment> is in an editing state
  - Enter characters and press the **[Enter]**, **[Tab]** or **[Shift]** + **[Tab]** keys to set the content
  - **[←]** : Moves cursor to previous character on left
  - **[→]** : Moves cursor to next character on right
  - **[Backspace]** : Deletes character before cursor
  - **[Delete]** : Deletes character after cursor
  - **[Ctrl]+[↓]**, **[End]** : Moves cursor to right end
  - **[Ctrl]+[↑]**, **[Home]** : Moves cursor to left end

8.2.1.4 Create using copy

In the variable setting area, the user can copy a variable, and then insert and paste it as another variable.
An underscore (_) is added to the variable name that is pasted.
<Copied variable name>_
If a variable is pasted again, then a number is added after the underscore (_).
<Copied variable name>_1, <Copied variable name>_2, <Copied variable name>_3…
However, if the maximum number of characters (32 characters) is exceeded after adding an underscore (_) and a number in the variable name, then the last character(s) in the original name is deleted so that the underscore (_) and a number can be added.

The user can use the edit operations to change the variable name that was pasted.
The user can copy and paste variables to other ST POUs.

Use the following operations to select a variable to copy.
- Click : Selects the variable that was clicked on
- **[Shift]** + Click : Specifies a range and selects multiple variables
- **[Ctrl]** + Click : Selects multiple variables that were clicked on

The user can perform one of the following operations to copy.

<u>Operation procedure</u>
Perform one of the following operations.
- Select <Copy> from the <Edit> menu
- Press the **[Ctrl]** + **[C]** keys
- Click on the button in the tool bar
- Right-click the mouse and select <Copy> from the displayed menu

The user can perform one of the following operations to paste.

<u>Operation procedure</u>
Perform one of the following operations.
- Select <Paste> from the <Edit> menu
- Press the **[Ctrl]** + **[V]** keys
- Click on the button in the tool bar
- Right-click the mouse and select <Paste> from the displayed menu

When performing a paste operation, the copied variable can be inserted and pasted at the variable position that was last selected.

(NOTE) The copied parameter variable cannot be pasted to the <External> tab in the variable editing area.

8.2.1.5 Cut and delete variables

This function cuts and deletes registered variables.
The cut variables can also be pasted.

The user can perform one of the following operations to cut.

Operation procedure

1. Select a variable to cut.
2. Perform one of the following operations.
   - Select <Cut> from the <Edit> menu.
   - Press the **[Ctrl]** + **[X]** keys
   - Click on the button in the tool bar
   - Right-click the mouse and select <Cut> from the displayed menu.

The user can perform one of the following operations to delete.

Operation procedure

1. Select a variable to delete.
2. Perform one of the following operations.
   - Click on the button
   - Press the **[Delete]** key
   - Right-click the mouse and select <Delete> from the displayed menu.

8.2.2 Create & Edit External Variables

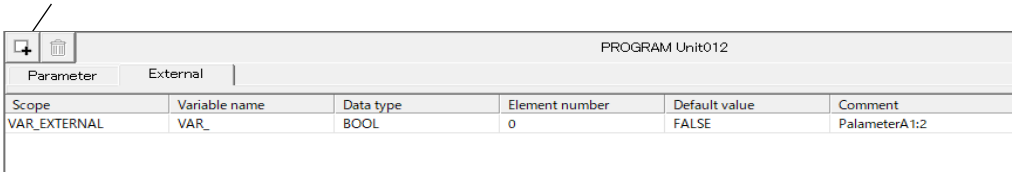
8.2.2.1 Create new external variable

The user can perform the following procedure in the variable editing area to create a new external variable.

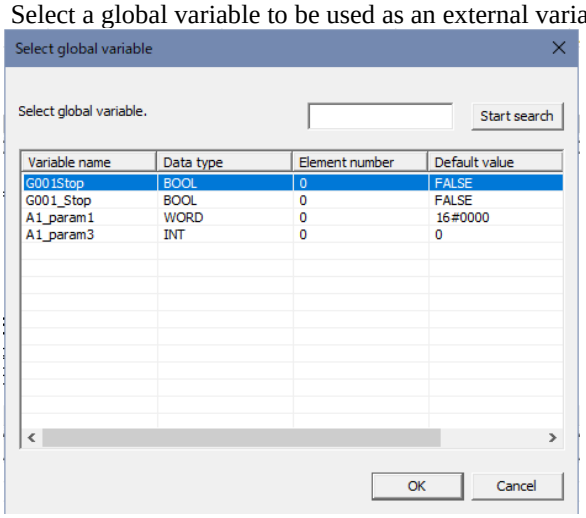
Operation procedure

1. Perform one of the following operations under the <External> tab.
   - Click on the <Insert> button.
   - Right-click the mouse on the variable setting field, and then select <Insert> from the context menu.

<Insert> button



2. The <Select global variable> dialog is displayed.
   The registered global variables are displayed. (Global variables already registered as external variables are not shown.)
   Select a global variable to be used as an external variable. Multiple selections are also possible.



3. Click on the <OK> button.
4. The external variable is added.

PROGRAM Unit004

Parameter | External

Scope	Variable name	Data type	Element number	Default value	Comment
VAR_EXTERNAL	VAR_	BOOL	0	FALSE	PalameterA1;2
VAR_EXTERNAL	G001Stop	BOOL	0	FALSE	StopFlag

When registering external variables, the external variable settings cannot be changed because that variable is registered as a global variable.
"VAR_EXTERNAL" is set for the scope of the external variable.

Click on the <Cancel> button in the <Select global variable> dialog to cancel registering an external variable.

(NOTE 1) In the PLC Programming software D00 <Global variable> dialog, if the user deletes or changes a global variable being used as a POU external variable, that change will not be automatically reflected in the POU external variable settings.
In this situation, re-register the external variable in the POU.

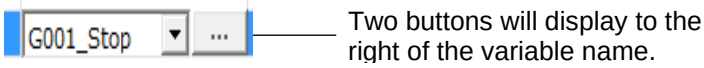
(NOTE 2) In the <Global variable> dialog, the user can specify a character string to search for a variable name that matches that character string.
If there are multiple hits for that variable name, then each time the user clicks on the <Start search> button, the search jumps to the next variable name. The letters in the search are not case sensitive. If there are no hits that match the variable name specified in the character string, then the selection bar will not move.

8.2.2.2 Change external variable

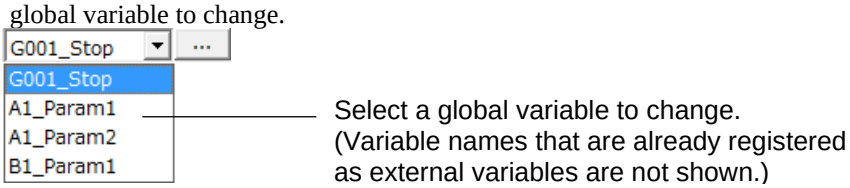
This function changes the source of the registered external variable to another global variable.

Operation procedure

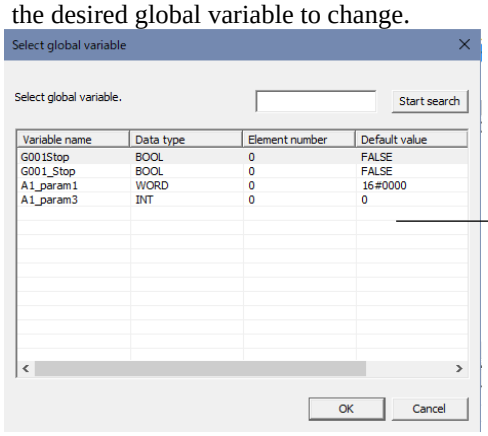
1. Under the <External> tab in the variable editing area, switch the focus to rename the external variable name, and then click on the variable name.



2. After clicking on the button, a list of the global variables is displayed. Select the desired global variable to change.



3. After clicking on the button, the <Select global variable> dialog is displayed. Select the desired global variable to change.



Select a global variable to change.
(Variable names that are already registered as external variables are not shown.)

4. Click on the <OK> button.

Click on the <Cancel> button in the <Select global variable> dialog to cancel the external variable change.

(NOTE) In the <Global variable> dialog, if the user deletes a global variable being used as a POU external variable or changes the settings, that change will not be automatically reflected in the POU external variable settings.
In this situation, re-register the external variable in the POU.

8.2.2.3 Cut and delete external variables

The user can perform one of the following operations to <Cut> a variable.

Operation procedure

1. Select a variable to cut.
2. Perform one of the following operations.
   - Select <Cut> from the <Edit> menu.
   - Press the **[Ctrl]** + **[X]** keys
   - Click on the button in the tool bar
   - Right-click the mouse and select <Cut> from the displayed menu.

The user can perform one of the following operations to <Delete> a variable.

Operation procedure

1. Select a variable to cut.
2. Perform one of the following operations.
   - Click on the button
   - Press the **[Delete]** key
   - Right-click on the mouse and select <Delete> from the displayed menu.

(NOTE 1) If an external variable being used in an ST program is deleted, the external variable name being used in the ST program will not automatically change.
Correct the program manually to change the external variable.

(NOTE 2) Even if the external variable registered in the POU is deleted, the original global variable (source) itself is not deleted.

8.3 Edit ST Programs

This section describes the operations to edit an ST program.

8.3.1 Procedure to Create ST Programs

When the ST program edit window is launched, the ST program is displayed and can be edited. The ST program can be edited by entering character strings and/or numerical values for "Assignment", "Operator" based expressions, "IF statements" or similar statements, "Operation commands" and other items.

Refer to "Chapter 12 Commands (ST)" in the PLC System Manual for details about ST programs.

Refer to the following input example when using variables or OMs in an ST program.

Registered variable

Scope	Variable name	Data type	Element number	Default value	Comment
VAR	GALLStart	BOOL	0	FALSE	Overall activarion Flag
VAR	Stat	BOOL	0	FALSE	Start Flag
VAR	stop	BOOL	0	FALSE	Stop Flag
VAR	GCommitno	BOOL	0	FALSE	GCommit No

Enter OMs

```
IF (X000 = 1)&(X001 = 0)&(X002 = 0)&(GCommitno <> 0)THEN
        Y010 := 1 ;
        Stat := 1 ;
END_IF;
```

Enter registered variable

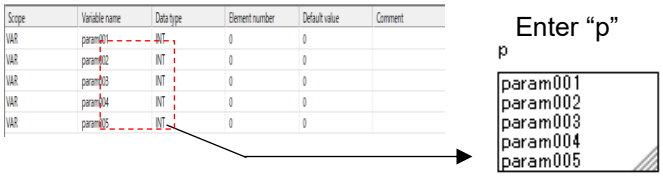
When entering an "OM" in an ST program, use "Half-width uppercase characters" in the OM notation.
If "Half-width uppercase characters" are not used in the OM notation, then it is processed as a variable that is not an OM.
Note that if half-width lowercase characters are used to enter the "OM", then the notation will automatically convert to "Uppercase characters" after inputting the following keys.

Keys	Enter space tab & ( ) * + , - . / : ; < = > [ ] { | }

The conversion function (to uppercase characters) is valid even for OMs where the index or digits are specified.
However, if the OM format is invalid, such as when specifying an out of range OM number, then it will not convert to uppercase characters.

Ex: **[y]→[a]→ [Space]** : Changes to Y00A.
[d]→[0]→[I]→[:] : Changes to D0L.

Make sure the variable matches the variable name that is set.
When inputting the characters for variable input, a list of the variable names is displayed. The user can avoid making an input mistake by selecting a variable from the list.



Use the **[↓]** or **[↑]** key to select a variable and press the **[Enter]** key.
Or, use the mouse to make a selection.

When using external variables, enter “gv.” before the variable name.
For example, when registering the external variable “VAR”, enter “gv.VAR”.

Correct example
gv.VAR:=1;
Incorrect example
VAR:=1; (“gv.” has not been entered)
GV.VAR:=1; (“gv.” has been entered using uppercase letters)

When specifying the element of an array variable, add brackets **[ ]** after the variable name to specify the element. In this case, the beginning element shall be 0. (Ex: Arr**[0]**, Arr**[1]**)
When an element is not specified, it refers to all array variables.
In addition, a variable (excluding arrays) and an OM can also be used to specify an element. (Ex: Arr**[VAR]**, Arr**[D0]**)

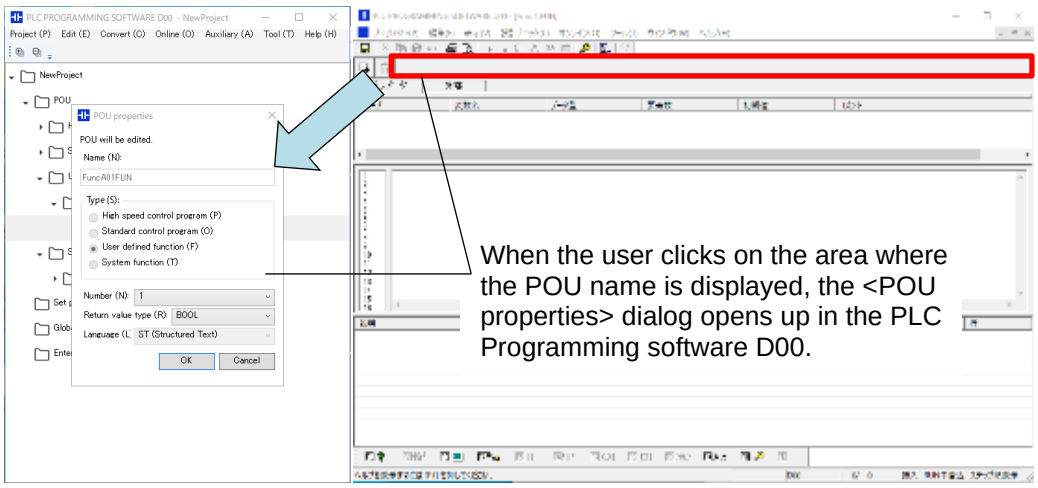
After an ST program is edited, a conversion operation not only checks the ST program created, but converts the program to a format that can be executed in the NC(BUILT IN PLC).
In addition, when the program data is not converted, there are restrictions on operations, such as being unable to save the ST program or use the search function. We recommend converting and saving the ST programs frequently.
Refer to “8.3.4 Conversion” for details about ST program conversion.
Refer to “5.2.3 Save POUs” for details about saving an ST program.

There are two modes for editing an ST program: “Overwrite mode” and “Insert mode”.
In “Overwrite mode”, the specified characters at the cursor position are overwritten.
In “Insert mode”, the specified characters at the cursor position are inserted.
Each time the **[Insert]** key is pressed, the mode changes back and forth between “Overwrite mode” and “Insert mode”.

There is an “Edit lock” function that prohibits edit operations in order to avoid inadvertent operations and changes to an ST program that was created.
When using the “Edit lock” function, the ST program and variables cannot be changed.
Refer to “8.6 Edit lock” for details on the “Edit lock” function.

8

While editing a user or system defined function POU, when the user clicks on the area where the POU name is displayed in the edit window, the <POU properties> dialog opens up in the PLC Programming software D00.
The user can change the return value type in the <POU properties> dialog.



Refer to “5.2.1 Procedure to create new POU” for details on settings and operations in the <POU properties> dialog.

8.3.2 Undo

This section describes the undo operation in the ST program edit window.

The user can perform the undo operation a maximum of 20 times and it applies to the following edit operations.

- Correcting text
- Cutting and pasting
- Replacing
- ST program conversion
- Variable editing (when focus is on the variable edit field)

The undo operation cannot be used for the display position in the ST program, such as returning to the original or previous display.

Operation procedure

The user can perform one of the following operations to carry out an undo operation.

- Select <Undo> from the <Edit> menu
- Press the **[Ctrl]** + **[Z]** keys
- Click on the button in the tool bar

8.3.3 Design Check

The design check function is for making sure there are no invalid statements in the POU that was created.

Refer to “5.7.2 Design check” for details on the design check.

8.3.4 Conversion

Conversion refers to converting the POUs (created in a language in the PLC Programming software D00 EDITOR) to code (intermediate code) that allows execution in the NC(BUILT IN PLC).

Refer to “5.9 Conversion” for further details.

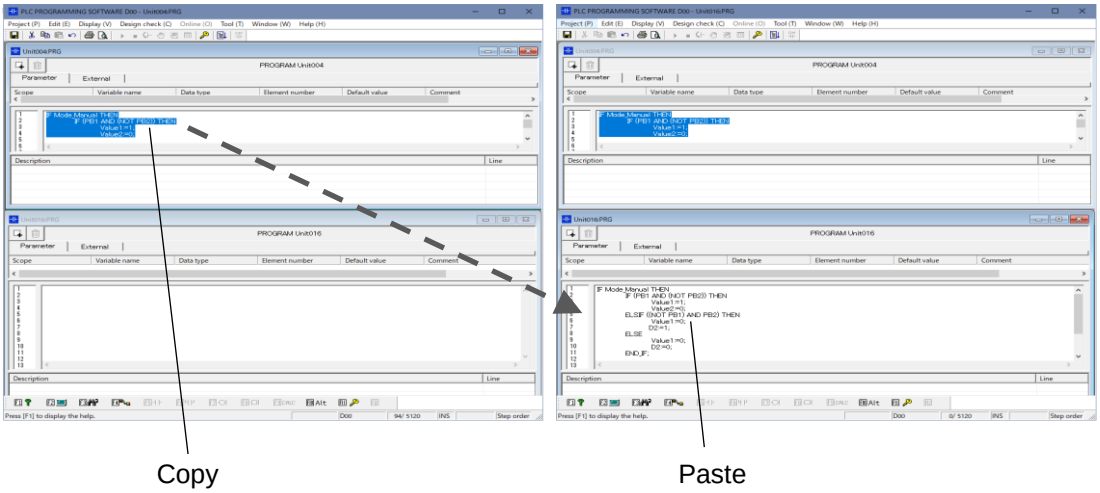
- Write during run
  If the “Write during RUN after conversion” setting is used in the “Conversion mode settings” in the PLC Programming software D00, then write during run operation is executed after a conversion operation is carried out.
  Refer to “5.10.4 Write during run” for details on the write during run operation.

(NOTE) When using the write during run operation, take the necessary safety precautions with the device that has the NC(BUILT IN PLC) and also with the operating environment.

8.4 Copy, Cut & Paste Character Strings

This section describes the copy, cut, paste and delete operations in an ST program that was created. The input format is inserted for the “Paste” operation regardless of whether the mode is in “Overwrite mode” or “Insert mode”.

The ST program “Copy and paste” as well as the “Cut and paste” operations can also be carried out between ST program edit windows.
The user can perform a copy or cut operation in the the ST program edit window that is active, and then the user can switch to another ST program edit window (make that window active) and paste that data into it.



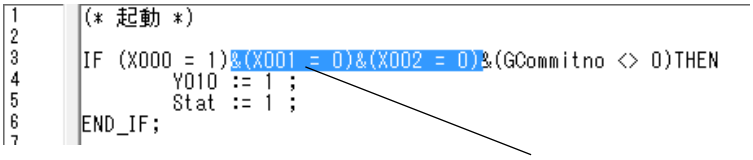
Refer to “8.4.3 Copy, cut, paste & delete operations” for details on how to copy, cut, paste and delete.

8.4.1 Delete 1 Character

Perform one of the following operations to delete a character before or after the cursor position.
- Character before the cursor: **[Back space]** key
- Character after the cursor: Select <Delete> from the <Edit> menu, or use **[Delete]** key

8.4.2 Specify Range

The following shows how to specify the range to perform copy, cut, paste and delete operations.
<u>Operation procedure</u>
Perform one of the following operations.
- Drag operation with mouse
- **[Shift]** + **[→]** keys, **[Shift]** + **[←]** keys, **[Shift]** + **[↓]** keys or **[Shift]** + **[↑]** keys



Selected character string is highlighted in blue.

The user can use the same procedure to select a character string even for character string (or numerical value) settings in the variable setting area.

Or, perform one of the following operations to specify the entire ST program for the range.

<u>Operation procedure</u>
Perform one of the following operations.
- Select <Select all> from the <Edit> menu.
- Press the **[Ctrl]** + **[A]** keys
- Right-click the mouse and select <Select all> from the displayed menu.

8.4.3 Copy, Cut, Paste & Delete Operations

This section describes how to perform copy, cut, paste and delete operations.

The user can perform one of the following operations to copy.

Operation procedure

Perform one of the following operations.

- Select <Copy> from the <Edit> menu
- Press the **[Ctrl]** + **[C]** keys
- Click on the button in the tool bar
- Right-click the mouse and select <Copy> from the displayed menu

The user can perform one of the following operations to cut.

Operation procedure

Perform one of the following operations.

- Select <Cut> from the <Edit> menu
- Press the **[Ctrl]** + **[X]** keys
- Click on the button in the tool bar
- Right-click the mouse and select <Cut> from the displayed menu

The user can perform one of the following operations to paste.

Operation procedure

Perform one of the following operations.

- Select <Paste> from the <Edit> menu
- Press the **[Ctrl]** + **[V]** keys
- Click on the button in the tool bar
- Right-click the mouse and select <Paste> from the displayed menu

The paste operation inserts at the cursor position.
When specifying the range to paste variable items in the ST program and variable editing area, the data is overwritten in the specified range.

The user can perform one of the following operations to delete.

Operation procedure

Perform one of the following operations.

- Select <Delete> from the <Edit> menu
- Press the **[Delete]** key
- Press the **[Back space]** key
- Right-click the mouse and select <Delete> from the displayed menu

8.5 Replace

This section describes how to replace character strings and OMs being used in the ST program.

8.5.1 Replace Character Strings

This function replaces a character string being used in an ST program with another character string.

(NOTE 1) When an OM is specified to replace a character string, the user can replace a bit OM with a word OM or a word OM with a bit OM. Be careful not to replace an OM accidentally.

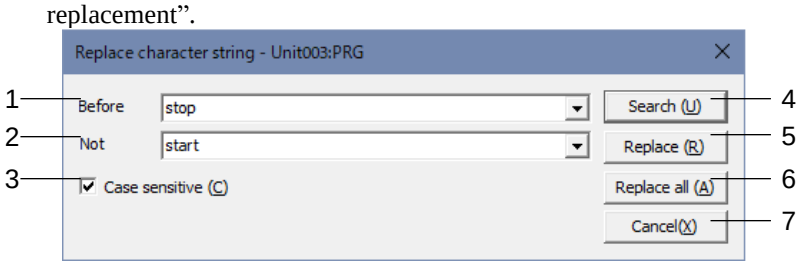
(NOTE 2) If an OM number is out of range after the replacement, an error message is displayed "when the ST program is converted".

(NOTE 3) If the variable name after the replacement is different from the set variable name, an error message is displayed "when the ST program is converted".

(NOTE 4) The character string will not be replaced if it is a "perfect match". Therefore, be careful not to enter similar character strings in the replacement operation, such as replacing "A" with "AA", and then using the "Replace all" function.

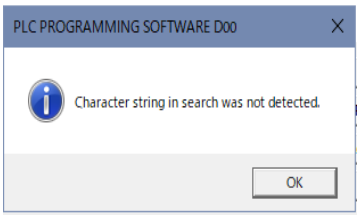
Operation procedure

1. Perform one of the following operations to display the <Replace character string> dialog.
   - Select <Replace character string> under <Replace> in the <Edit> menu
   - Press the **[Ctrl]** + **[R]** keys
2. Tick the checkbox for the <Case sensitive> setting for "Before replacement" and "After replacement".



No.	Item	Description
1	Before replacement	Specifies the character string before replacement.
2	After replacement	Specifies the character string after replacement.
3	Case sensitive	Selects whether or not to differentiate between uppercase and lowercase letters in the character string search. Tick the checkbox to differentiate between uppercase and lowercase letters.
4	<Search> button	Starts from the cursor position and searches for a character string specified in the "Before replacement" field. When the search comes to the end of the ST program, it goes back to the beginning of the ST program and continues the search up to the original cursor position.
5	<Replace> button	Starts searching from the cursor position and replaces the targeted character string. When the replacement search comes to the end of the ST program, it goes back to the beginning of the ST program and continues to replace the targeted character strings up to the original cursor position.
6	<Replace all> button	Replaces the character string specified in the "Before replacement" field that appears throughout the ST program, with the character string specified in the "After replacement" field all at once.
7	<Cancel> button	Closes the <Replace character string> dialog.

3. After clicking on the <Search> or <Replace> button, the first found character string (starting the search from the current cursor position and moving down) becomes selected.
4. Click on the <Replace> button to replace, or click on the <Search> button to skip it and continue the search.
   Click on the <Replace all> button to replace all the locations at once.
5. Repeat step 4 to continue replacing all the specified character strings throughout the ST program.
   When the processing is complete, a dialog appears indicating the processing is complete.



(NOTE) The user can edit an ST program while the <Replace character string> dialog is displayed.

8.5.2 Replace All OMs

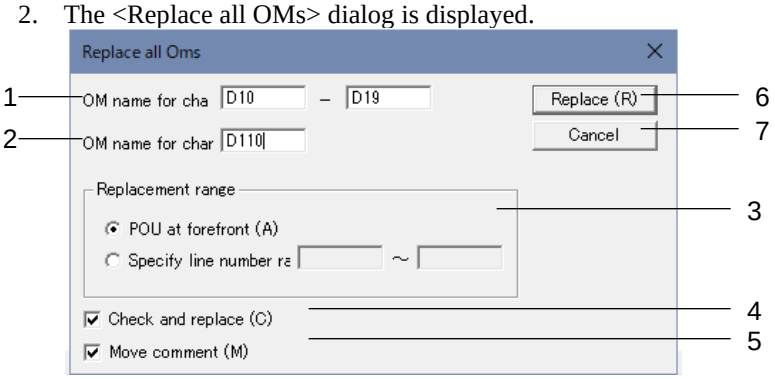
This function replaces an OM being used in an ST program with another OM.
The user can either replace the OMs being used in an ST program all at once, or check and replace each OM one at a time.

When replacing OMs, the user can only replace bit OMs with bit OMs and only replace word OMs with word OMs.
When “X” is specified for the source OM, “X” can only be set for the target OM.
When the user clicks on the <Replace> button and the OM number setting exceeds the upper limit after the replacement, then the error message <<Not possible to replace OM number because new OM number will be out of range.>> is displayed.
When a comment is moved, the OM comment (source) overwrites the target OM comment and then the source OM comment is deleted.
If the same OM is used at multiple locations and one of those OMs is replaced, then that OM comment is also transferred with the data.
After the replacement is completed, the design check is automatically executed.

(NOTE 1) When replacing OMs, the user can only replace bit OMs with bit OMs and only replace word OMs with word OMs.
(NOTE 2) When “X” is specified for the source OM, “X” can only be set for the target OM.
(NOTE 3) When the user clicks on the <Replace> button and the OM number setting exceeds the upper limit after the replacement, then the error message <<Not possible to replace OM number because new OM number will be out of range.>> is displayed.
(NOTE 4) When a comment is moved, the OM comment (source) overwrites the target OM comment and then the source OM comment is deleted.
If the same OM is used at multiple locations and one of those OMs is replaced, then that OM comment is also transferred with the data.
(NOTE 5) After the replacement is completed, the design check is automatically executed.

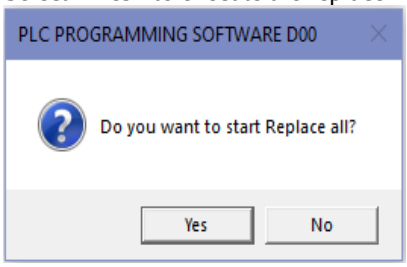
Operation procedure

1. Select <Replace al OMs> under <Replace> in the <Edit> menu.
2. The <Replace all OMs> dialog is displayed.

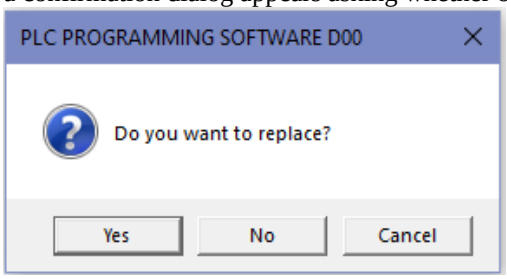


No.	Item	Description
1	OM name for change source	Specifies the first and the last OMs in the source.
2	OM name for change destination	Specifies the first OM in the destination. The range for replacing OMs is specified in the “OM name for change source”.
3	Replacement range	Selects and specifies the replacement range. “POU at forefront”: Specifies the replacement range for the POU at the forefront. “Specify line number range”: Specifies the numbers for the first and last lines in the range.
4	Check and replace	Selects whether to replace all OMs at once or to check and replace each one at a time. When the checkbox is ticked, it will check and replace each OM one at a time.
5	Move comment	Selects whether or not to move the comment from the source OM to the destination when replacing the OM. Tick the checkbox to move the OM comment.
6	<Replace> button	Starts the replacement.
7	<Cancel> button	Closes the <Replace all OMs> dialog.

3. After clicking on the <Replace> button, a confirmation dialog appears asking whether to start the replacement.
   Select <Yes> to execute the replacement operation.



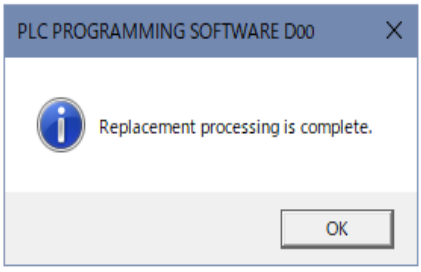
4. When the checkbox <Check and replace> is ticked, the cursor moves to the specified OM and a confirmation dialog appears asking whether or not to replace the OM.



Select <Yes> to replace the OM (at the cursor) and then search for the next OM.
Select <No> to skip replacing the OM (at the cursor) and search for the next OM.
Select <Cancel> to cancel the OM replacement operation.

When the checkbox <Check and replace> is not ticked, the confirmation message does not appear and it replaces the specified OMs all at once.

5. When all the specified OMs have been replaced, a dialog appears indicating the processing is complete.



6. Even if the replacement is complete, it does not close and exit the <Replace all OMs> dialog. The user can continue doing other replacements.
Click on the <Cancel> button to exit replacement and close the <Replace all OMs> dialog.

When all the OMs are replaced at once, the design check is automatically executed. If the program becomes invalid due to the replacement operation, an error is displayed in the message display area. Check the error description and correct the program.

8.6 Edit Lock

This section describes the operations to prohibit or lock editing in the ST program in order to protect the program from an accidental or inadvertent input operation.

The edit lock is a function to prevent an inadvertent operation from writing data to the ST program and specifically prohibits the following edit operations.

- Edit ST program
- Replace
- Edit variable

The user can unlock the editing by performing the edit lock operation again (while the edit lock is on), allowing the user to edit the data.

When the program edit window is closed while the edit lock is on, the edit lock will be released (unlocked) the next time the same POU edit window is opened.

The “Edit lock/edit unlock” can be performed using one of the following operations.

Operation procedure

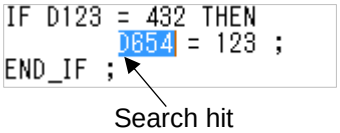
Perform one of the following operations.

- Select <Edit lock> from the <Edit> menu
- Press the **[F11]** key
- Click on the button in the tool bar

8.7 Search

This section describes the operations on how to search for character strings, how to check the OM status and how to search for OMs.

In an ST program search, the character strings or OMs that are found are highlighted in blue.



Note that the search function is only carried out for the POU at the forefront in the PLC Programming software D00 LD/ST Programming Editor. The other POUs are not searched.

8.7.1 Character String Search

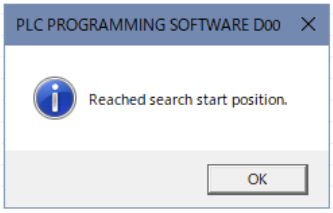
The character string search conducts a search to check if the specified character string is being used or not in the ST program.
The search starts from the cursor position and searches for the specified character string according to the “Conditions”.
When the character string is found, that character string is highlighted in blue.
Each time the <Search> button is pressed, it searches for the next character string.
When the search comes to the end of the ST program, it goes back to the beginning of the ST program and continues the search up to the original cursor position.

Operation procedure

1. Perform one of the following operations.
   - Select <Find character string> under <Search> in the <Edit> menu
   - Press the **[Ctrl]** + **[F]** keys
2. Set the <Conditions> and <Case sensitive> setting in the <Find character string> dialog.

No.	Item	Description
1	Conditions	Specifies the character string in the search.
2	Case sensitive	Selects whether or not to differentiate between uppercase and lowercase letters in the character string search. Tick the checkbox to differentiate between uppercase and lowercase letters.
3	<Search> button	Starts from the cursor position and searches for a character string specified in the “Conditions” field. When the search comes to the end of the ST program, it goes back to the beginning of the ST program and continues the search up to the original cursor position.
4	<Cancel> button	Closes the <Find character string> dialog.

3. When the <Search> button is pressed, it searches for the first character string (in the range).
4. Each time the user clicks on <Search> button, it searches for the next character string.
5. Repeat step 4 to continue searching for all the search hits throughout the ST program.
   When the processing is complete, a dialog appears indicating the processing is complete.



6. Even if the search is complete, it does not close and exit the <Find character string> dialog.
   The user can continue doing other searches.
   Click on the <Cancel> button to close and exit the search.

After conducting a search in the <Find character string> dialog, the user can press the **[F3]** key or the **[Shift] + [F3]** keys to search up or down (from the cursor position) for the same character string (in the last search).
At this time, the search is completed once it reaches the end or the beginning of the ST program.

Refer to "10.4.15 Character String Search (ST Language)" for details on the character string search.

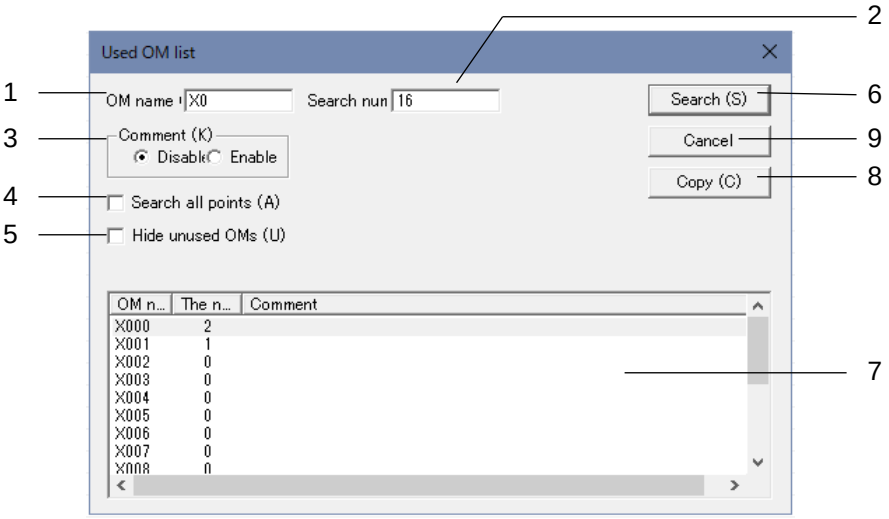
(NOTE 1) The user can edit an ST program while the <Find character string> dialog is displayed.
(NOTE 2) When a replacement operation was executed, the user can press the **[F3]** key or the **[Shift] + [F3]** keys to search for the character string in the "Conditions" field.

8.7.2 OM List

The OM list can be used to conduct a search to check if an OM is being used or not in the ST program by specifying the search range from a certain OM.

Operation procedure

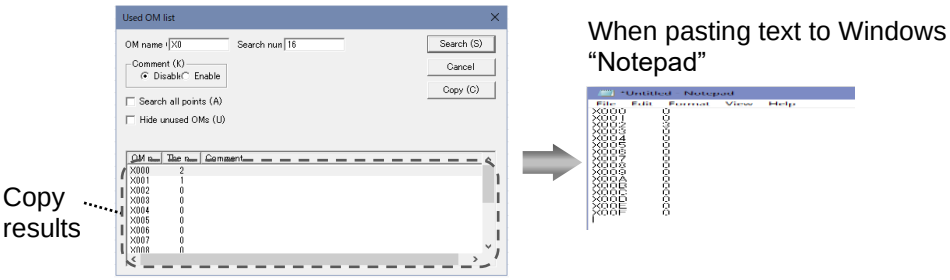
1. Perform one of the following operations.
   - Select <Used OM list> under <Search> in the <Edit> menu
   - Press the **[Shift]** + **[F4]** keys
2. Set the OM and search range in the <Used OM list> dialog.



No.	Item	Description
1	OM name	Specifies the OM search.
2	Search number	Specifies the search range from the set OM.
3	Comment	Select <With> to display the OM comments in the search results.
4	Search all points	Tick the checkbox to search all locations from the specified OM.
5	Hide unused OMs	Tick the checkbox to display the search results excluding the unused OMs.
6	<Search> button	Starts the search.
7	List of search results	Displays the search results in a list. <OM name>: Displays the OMs that are found. <Qty. used>: Displays the quantity of the specified OM that is used. <Comment>: Displays the comment that is set to the OM when <Comment> is set to <With>.
8	<Copy> button	Copies the displayed search results.
9	<Cancel> button	Closes the <Used OM list> dialog.

3. Set the range of the search in the <Search number> field, or tick the checkbox <Search all points> to search all the points after the specified OM.
4. Tick the checkbox <Hide unused OMs> to display only the OMs that are used.
5. Click on the <Search> button.
6. Even if the search is complete, it does not close and exit the <Unused OM list> dialog. The user can continue doing other searches.
   Click on the <Cancel> button to close and exit the search.

- Copy
  The search results can be copied as text to the Windows clipboard.
  The user can paste the copied search results to another application like a commercially available text editor software.



Refer to "10.4.2 OM List (LD Language)" for details on the "Used OM list".

(NOTE) Use the OM list after converting the ST program and when there are no errors.

8.7.3 OM Search

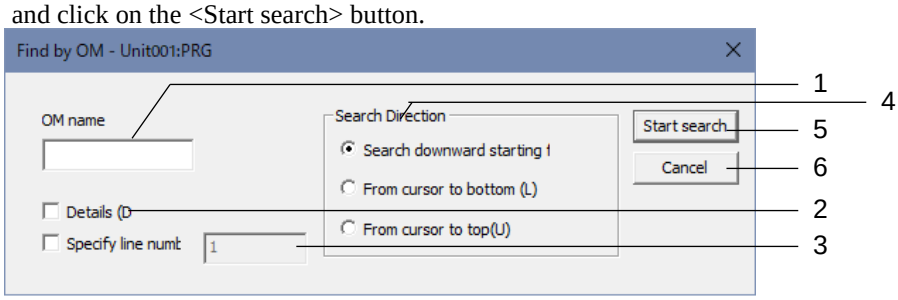
The user can specify an OM and conduct a search to check where it is used.
In an ST program, the search starts from the current cursor position and looks for the specified OM.
However, when the "Specify line number" setting is enabled, the OM search starts from the specified line number.

(NOTE) The user cannot specify an OM to conduct a search when the data is not converted.

The following shows the procedure to specify an OM and conduct a search.

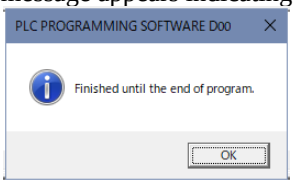
<u>Operation procedure</u>

1. Perform one of the following operations.
   - Select <Find by OM> under <Search> in the <Edit> menu
   - Press the **[Ctrl]** +**[Shift]**+ **[D]** keys
2. Set the <OM number> in the <Find by OM> dialog.
   Select the conditions ("Details", "Specify line number" or "Search direction") as necessary and click on the <Start search> button.



No.	Item	Description
1	OM name	Specifies the OM search.
2	Details	When a bit OM word is specified, tick the checkbox to search for OMs that are not displayed in the ST program.
3	Specify line number	Tick the checkbox to search for OMs starting from the set line number.
4	Search direction	Specifies the search direction. The user can select the following search methods. • Start from top and search down • Start from cursor position and search down • Start from cursor position and search up
5	<Start search> button	Starts the search.
6	<Cancel> button	Closes the <Find by OM> dialog.

3. The OM that is found is highlighted in blue.
   Ex: D0 = 123 ;
4. When the user clicks on <Start search> button again, it searches for the next location where the OM is used.
5. When the search has finished checking up to the last line of the ST program, the following message appears indicating that the search is complete.



6. Even if the search is complete, it does not close and exit the <Find by OM> dialog. The user can continue doing other searches.
   Click on the <Cancel> button to close and exit the search.

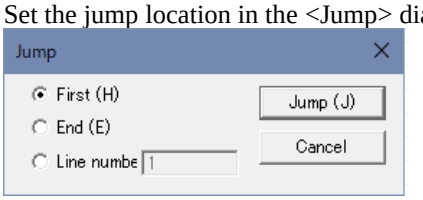
8

8.7.4 Jump

This section describes how to jump to the first line, last line or a specified line in the ST program.

Operation procedure

1. Perform one of the following operations.
   - Select <Jump> from the <Edit> menu.
   - Press the **[Ctrl]** + **[J]** keys
2. Set the jump location in the <Jump> dialog.



   Select <First> to jump to the first line in the ST program.
   Select <End> to jump to the last line in the ST program.
   Select <Line number> and enter a number to jump to a specified line.
3. After clicking on the <Jump> button, the cursor moves to the beginning of the specified line.

When specifying a line number, if the specified number exceeds the actual number of lines, then it jumps to the last line.

Click on the <Cancel> button in the <Jump> dialog to cancel the jump operation.

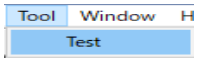
8.8 Launch Simulation

Before running an ST program on the machine, the user can check the operation of the ST program in the PLC Programming software D00 EDITOR.

The following is the procedure for running a simulation of an ST program that was created.

Operation procedure

1. Select <Simulation> from the <Tool> menu.
   After the simulation is launched, a checkmark "✓" appears before the item name in the <Simulation> menu.
   Checkmark "✓" appears.



2. Write the ST program to the simulation.
   Refer to the "5.10.2 Write programs" for details about writing ST programs.

- Exit simulation
  When there is a checkmark "✓" next to the <Simulation> item in the <Tool> menu, select <Simulation> again (to deselect and exit).
  After exiting simulation, the checkmark "✓" next to the <Simulation> disappears.

(NOTE 1) The user can launch and exit simulation from the PLC Programming software D00 or from the FBD Programming Editor.
When running an ST program in the simulation, there is no need to re-launch the PLC Programming software D00 LD/ST Programming Editor if the simulation has already been launched with another tool.

(NOTE 2) The POUs set in the <Program settings> dialog are written to the program. Register the ST program in the <Program settings> dialog to run the ST program currently being edited in the simulation. Refer to "5.8 Program Execution Settings " for details on the <Program settings> dialog.

8.9 NC(BUILT IN PLC) Monitoring

The user can perform following operations to execute ST program monitoring in the NC(BUILT IN PLC) or in the simulation.

Operation procedure

Perform one of the operations below in the PLC Programming software D00 LD/ST Programming Editor.

- Press the **[F2]** key
- Select <Start monitoring> under the <Online> menu
- Click on from the function key tool bar

Refer to "5.12 NC(BUILT IN PLC) Monitoring" for further details n monitoring.
In addition, refer to "8.11 Monitoring operations" and "8.12 DEBUG Function" for details on ST program operations when monitoring.

8.10 Run & Stop in NC(BUILT IN PLC)

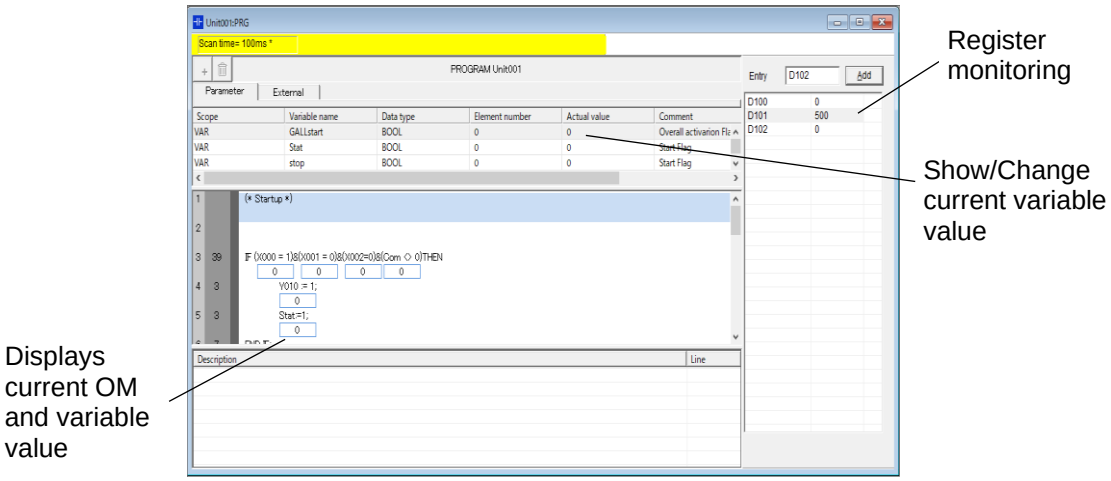
Refer to "5.13 Run & Stop in NC(BUILT IN PLC)" for details on running and stopping in the NC(BUILT IN PLC).

8.11 Monitoring Operations

This section describes the operations for each function when monitoring an ST program in the NC(BUILT IN PLC).

(NOTE) When there is a checkmark "✓" next to the <Simulation> in the <Tool> menu (for each tool), then the program is written to the simulation.

The user can check the current value of the OMs and variables in the ST program by connecting the PLC Programming software D00 EDITOR to the NC(BUILT IN PLC).



The ST program monitoring displays the current values for each OM or variable.

By registering an OM beforehand in the monitoring registration, the user can constantly check the OM status or value regardless of the display position of the ST program.
Refer to "8.11.3 Monitoring registration" for details on monitoring registration.

The user can change the base number when displaying the current value of the word OM.
Refer to "8.11.1 Change base number for monitoring values" for details on changing the base number.

While monitoring is in progress, the "Default value" field in the variable registration area changes to the "Current value", and the current value for each variable is displayed.

The user can force a change in the current OM status while monitoring.

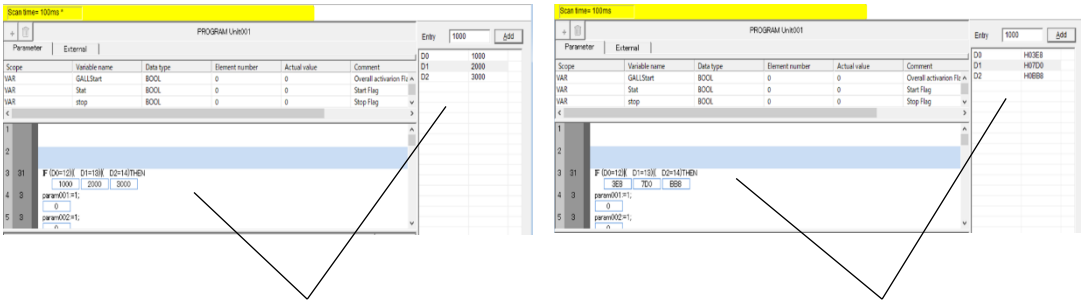
The following functions can be used to change the OM status.
- To force the bit OM to change to ON or OFF :OM SET/RST
- To force a change in the current value of the word OM :Change current value
- To change the OMs in the table format all at once :Batch monitor

Refer to "8.11.4 SET/RST OM" for setting or resetting the OMs.
Refer to "8.11.5 Change current values" for details about changing the current values.
Refer to "8.11.6 Batch monitoring" for details about batch monitoring.

In addition, the user can press the **[ESC]** key to only stop the monitoring and keep executing the NC(BUILT IN PLC).
Press the **[F2]** key to exit monitoring. The user can also scroll through the program edit window.

8.11.1 Change Base Number for Monitoring Values

The user can change the current values displayed during monitoring to “Decimals” or “Hexadecimals”.



Decimal numeral display

Hexadecimals numeral display

Operation procedure

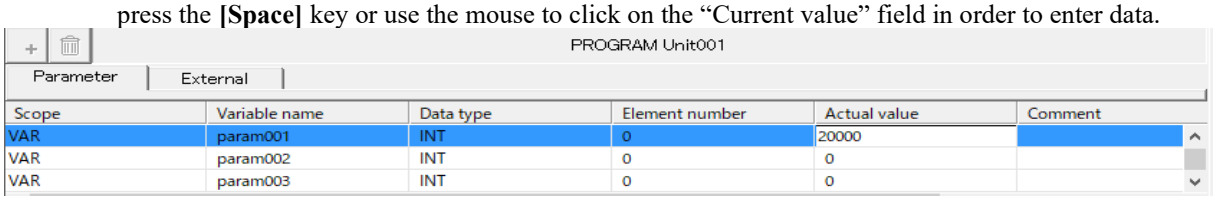
Perform one of the following operations.

- Select <Change monitoring base number> under <Change display> in the <Display> menu
- Press the **[Shift]** + **[F8]** keys

8.11.2 Monitoring & Changing Current Variable Values

8.11.2.1 Monitoring parameter variables and external variables

The current values of the variables are displayed in the variable editing area.
In addition, the user can enter a numerical value or character string into the “Current value” field in the variable editing area in order to change the current value of a variable. The user can also press the **[Space]** key or use the mouse to click on the “Current value” field in order to enter data.



The values (base number) that can be entered for the variable vary depending on the data type.

Data type	Input range/base number
BOOL	0, 1, FALSE, TRUE)
BYTE	2#00000000~2#11111111 8#000~8#377 0~255 16#00~16#FF
WORD	2#0000000000000000~2#1111111111111111 8#000000~8#177777 -32768~32767 16#0000~16#FFFF
DWORD	2#00000000000000000000000000000000~ 2#11111111111111111111111111111111 8#00000000000~8#37777777777 -2147483648~2147483647 16#00000000~16#FFFFFFFF
INT	-32768~32767 16#0000~16#FFFF
DINT	-2147483648~2147483647 16#00000000~16#FFFFFFFF
LINT	-9223372036854775808~9223372036854775807 16#0000000000000000~16#FFFFFFFF FFFFFFFF
UINT	0~65535 16#0000~16#FFFF

Data type	Input range/base number
UDINT	0~4294967295 16#00000000~16#FFFFFFFF
ULINT	0~18446744073709551615 16#0000000000000000~16#FFFFFFFF FFFFFFFF
REAL	-3.402823E+38~-1.401298E-38, 0.0, 1.401298E-38~3.402823E+38 (single-precision real number)
LREAL	-1.79769313486231E+308~-4.94065645841247E-324, 0.0, 4.94065645841247E-324~1.79769313486231E+308 (double-precision real number)
TIME	T#000:00:00~T#255:59:59
DATE	D#1970-01-01~D#2037-12-31
TOD	TOD#00:00:00~TOD#23:59:59
DT	DT#1970-01-01-00:00:00~DT#2037-12-31-23:59:59
STRING	0 to 31 characters (half-width alphanumeric characters and symbols only)

(NOTE 1) When changing variable values online, be sure to take the necessary safety precautions. If an invalid value is set, the device that has the NC(BUILT IN PLC) installed could malfunction.

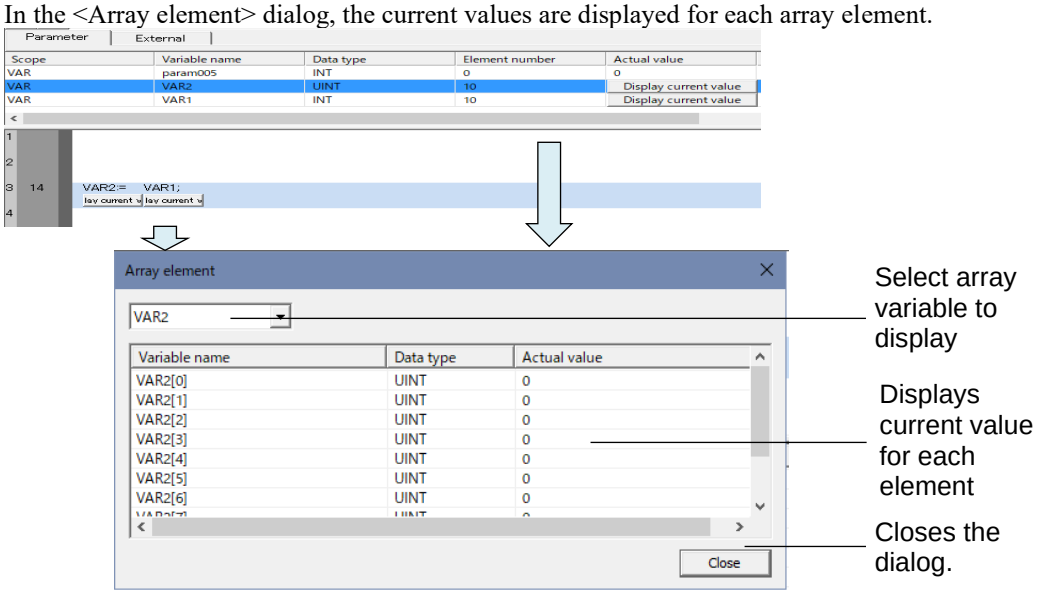
(NOTE 2) When a current value is displayed for a REAL or LREAL type variable, then the set value is rounded off to the valid digits of that type (7 digits and 15 digits, respectively). As a result, the actual value and displayed value may not match. Examples are shown below.
In addition, even when the current value is changed for a REAL or LREAL type variable, the set value is rounded off to the valid digits of that type (7 digits and 15 digits, respectively).

Data type	Current value	Displayed value
REAL type	1234567	1234567
	12345678	1.234568E+007
	0.0001234567	0.0001234567
	0.00012345678	0.0001234568
LREAL	123456789012345	123456789012345
	1234567890123456	1.23456789012346E+015
	0.000123456789012345	0.000123456789012345
	0.0001234567890123456	0.000123456789012346

8

8.11.2.2 Array variable monitoring

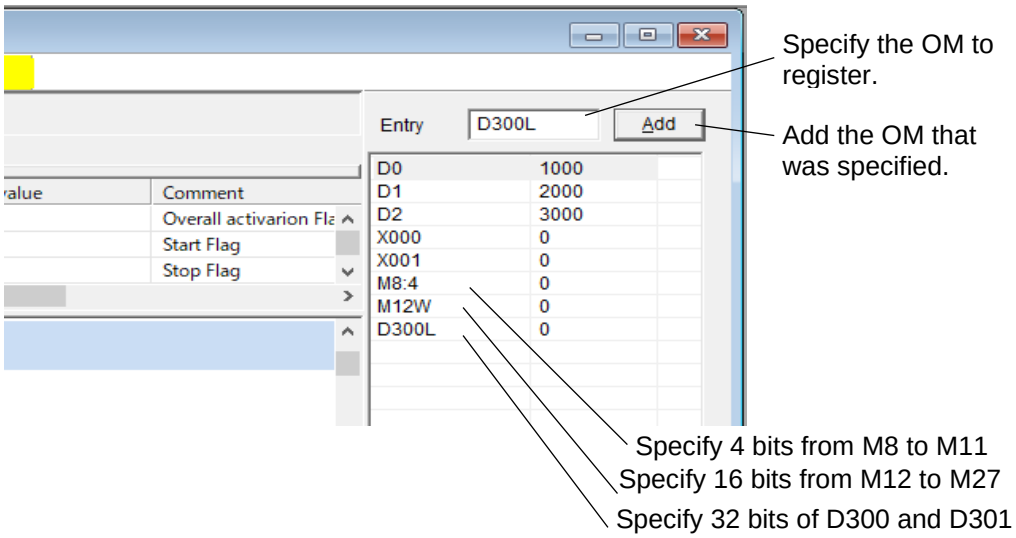
For array variables, the <Display current value> button is displayed in the current value field for the variable editing area.
After clicking on the <Display current value> button, the <Array element> dialog is displayed.
In the <Array element> dialog, the current values are displayed for each array element.



The current value of a variable can be changed by entering a numerical value or character string into the <Current value> field in the <Array element> dialog. The procedure to change the current value is the same as in the variable editing area. Refer to “8.11.2.1 Monitoring parameter variables and external variables”.

8.11.3 Monitoring Registration

By registering an OM beforehand in the monitoring registration, the user can constantly monitor the OM status or value regardless of the display position of the ST program.
The user can also specify the digits for the bit OM in the monitoring registration.
Refer to “Chapter 12 Commands (ST)” in the PLC System Manual for details on specifying the digits of an OM.



In the monitoring registration, the maximum number of OMs that can be registered is 24. If the user enters 25 OMs, the first OM that was registered is cancelled.

The user can switch between showing and hiding the monitoring registration display by performing the following operations.

Operation procedure
Select <Register monitoring> from the <Display> menu to hide the monitoring registration.

The user can force a change in the current OM status in the monitoring registration.
The following functions can be used to change the OM status.
- To force the bit OM to change to ON or OFF: Double-click on the line for the bit OM
- To force a change in the current value of the word OM: Double-click on the line for the word OM

Refer to “8.11.4 OM SET/RST” for setting or resetting the OMs.
Refer to “8.11.5 Change current values” for details about the current monitoring values.

(NOTE) OMs can be registered in the monitoring registration. Variables cannot be registered.

The user can perform one of the following operations to set an OM in the monitoring registration.

Operation procedure
Perform one of the following operations.
- Enter an OM into the OM setting field at the top of the monitoring registration, and press the <Add> button or press the **[Enter]** key.
  The cursor can also be moved to the OM setting field on top of the monitoring registration by selecting <Register (command input)> under <Register monitoring> in the <Online> menu.

The user can perform one of the following operations to unregister or clear an OM from the monitoring registration.

Operation procedure
Select the desired OM in the monitoring registration, and then press the **[Delete]** key.

The user can perform one of the following operations to unregister or clear all of the OMs from the monitoring registration.

Operation procedure
Perform one of the following operations.
- Select <Clear all entries> under <Register monitoring> in the <Online> menu
- Right-click on the mouse, and select <Clear all entries> under <Register monitoring> from the displayed menu
- Press the **[Shift]** + **[F6]** keys

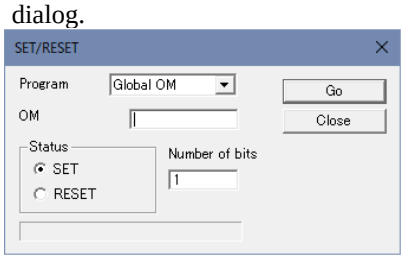
8.11.4 SET/RST OM

The user can force the bit OM status to change from ON to OFF and from OFF to ON.
In addition, the user can reset the current value of the word OM to 0 (zero).

(NOTE 1) When changing OM values online, be sure to take the necessary safety precautions. If an invalid value is set, the device that has the NC(BUILT IN PLC) installed could malfunction.
(NOTE 2) When specifying the number of bits, if the bit number exceeds the last number of the specified OM, then it may adversely affect another OM. Therefore, be careful when specifying the number of bits.
(NOTE 3) If the user specifies too many bits, the processing may take a while.
(NOTE 4) The local timer (LT) coil and the local counter (LC) coil cannot be reset with the OM SET/RST operations.
(NOTE 5) The following word OMs can be reset: BDX, BDY, D, LD, SD, IX, LT and LC.
(NOTE 6) This function does not continuously force the OM to stay ON or OFF. Even if this function is used to change the current value, the OM value will ultimately change based on the program operation or an external input.
(NOTE 7) An OM with an index cannot be specified when changing the current value.
(NOTE 8) This function cannot be used to change the value of a variable.

8

Operation procedure
1. When monitoring an ST program, perform one of the following operations to display the <SET/RESET> dialog.
   - Press the **[F9]** keys
   - Select <OM SET/RST> under the <Online> menu
   - Right-click the mouse and select <OM SET/RST> from the displayed menu
   - Double-click on a bit OM in the monitoring registration
2. In the displayed <SET/RST> dialog, specify the “Program”, “OM name” and “Number of bits” and select the <Operation>.
   Refer to “7.12.3 SET/RST OM” for details about the operations in the <OM SET/RST> dialog.



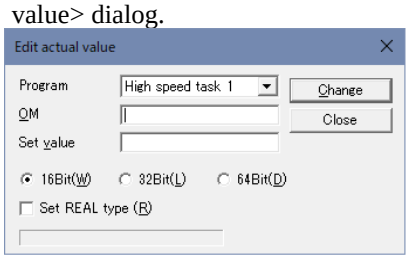
8.11.5 Change Current Values

The user can force a change in the current value of a word OM.

(NOTE 1) When changing OM values online, be sure to take the necessary safety precautions. If an invalid value is set, the device that has the NC(BUILT IN PLC) installed could malfunction.
(NOTE 2) The local timer (LT) and local counter (LC) contacts do not turn OFF, even if the current values are forced to reset to “0” on the local timer (LT), where the time expired, and on the local counter (LC), where the count is reached.
(NOTE 3) This function cannot be used to change the set values for the local timer and local counter.
(NOTE 4) An OM with an index cannot be specified when changing the current value.
(NOTE 5) A digit-specified bit OM cannot be specified when changing the current value.
(NOTE 6) This function cannot be used to change the value of a variable.
(NOTE 7) Set a value that does not exceed the last number of the OM when writing a 32-bit value or 64-bit value to the OM.
(NOTE 8) The value cannot be changed for a digit-specified bit OM.

Operation procedure

1. When monitoring an ST program, perform one of the following operations to display the <Change current value> dialog.
   - Press the **[Shift]** + **[F9]** keys
   - Select <Change current value> under the <Online> menu
   - Right-click the mouse and select <Change current value> from the displayed menu
   - Double-click on a word OM in the monitoring registration
2. In the displayed <Change current value> dialog, specify the “Program”, “OM name” and data size to be changed, and set the “Change value”.
   Refer to “7.12.4 Change current value” for details about operations in the <Change current value> dialog.



8

8.11.6 Batch Monitoring

In batch monitoring, the user can check the values or status of consecutive OMs in a table format. Refer to “7.12.5 Batch monitoring” for details about batch monitoring.

8.11.7 Clear All OMs & Variables

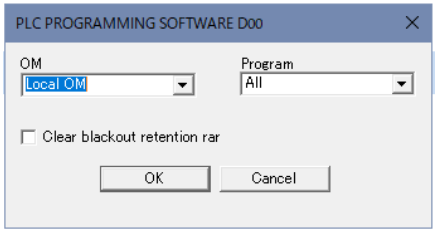
This function clears all the values and status data for the OMs and variables at once (except for the following OMs: X, Y, BX, BY, BDX, BDY, SM and SD). In addition, the user can also reset all the variables to their default values at once.
The set values for the local timer and the local counter cannot be cleared.
The OM/variable all clear operation can only be executed if the NC(BUILT IN PLC) is in a stopped state.

(NOTE) The OM/variable all clear operation applies to all OMs except for the following: X, Y, BX, BY, BDX, BDY, SM and SD. However, when the ST program is running, be aware that the status of Y, BY, BDY, SM and SD will change based on changes in the conditions.

The following shows the procedure for the OM/variable all clear operation.

Operation procedure

1. Change the NC(BUILT IN PLC) to a STOP state.
   Refer to “5.13 Run & Stop in NC(BUILT IN PLC) ” for details on changing it to a STOP state.
2. Use one of the following operations to “Clear all OMs/variables”.
   - Select <Clear all OMs/variables> from the <Online> menu
   - Press the **[Ctrl]** + **[F9]** keys
3. In the following dialog, select the “OM”, “Program” and “Clear power failure hold area” as necessary, and then click on the <OK> button.
   Refer to “7.12.6 Clear all OMs & variables ” for details about the dialog operations.



8

8.12 DEBUG Function

This section describes the “DEBUG function” used for checking the ST program operation.
The “DEBUG function” can only be used with standard task programs. It is possible to use it on a high speed task only when executing a simulation. As a result, when a high speed task program window is at the forefront, the debug function menu is disabled and cannot be selected. The only exception is when executing a simulation.
The user can perform the following ST program check operations with the “DEBUG function”.

- Step execution
  Executes only one command in the ST program and then stops.

- ST program stop based on breakpoint setting
  ST program stops at specified breakpoint.

- Monitoring stop when OM value changes
  Monitoring is stopped when the specified OM value changes.

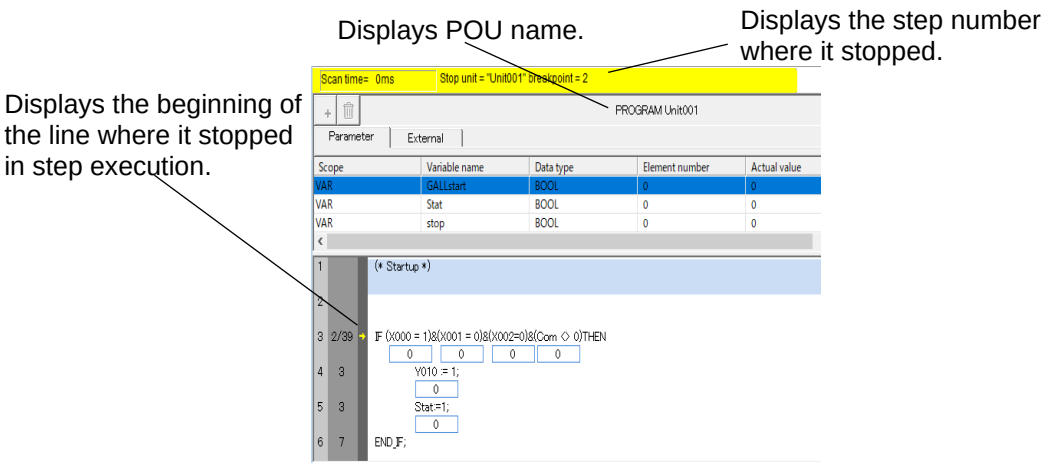
(NOTE) Do not use the DEBUG function on user or system defined functions.

8.12.1 Step Execution

This section describes step execution operations.
The user can execute the program in steps when the ST program is in a stopped state.

(NOTE) Step execution is not possible when the ST program monitoring is not executed or when monitoring is stopped.

When the <RUN/STOP mode> dialog is used to stop the ST program, it executes from the beginning of the ST program.
When the ST program is stopped based on the conditions that are set in “6.14.2 Setting and execution of stop position based on breakpoint setting”, then it executes from the stop position.



The following shows the step execution procedure.

Operation procedure

1. Start ST program monitoring.
2. Stop the ST program.
   Refer to “5.13 Run & Stop in NC(BUILT IN PLC)” for details on the stop operations using the <RUN/STOP mode> dialog.
   Refer to “8.12.2 Setting and execution of stop position based on breakpoint setting” for details on stopping the ST program based on the condition settings.
3. Use one of the following operations to perform “Step execution”.
   - Select <Execute step> under <DEBUG> in the <Online> menu
   - Click on the button in the tool bar
4. Repeat step 3 to execute the ST program per command.

Change the setting in the <RUN/STOP mode> dialog to RUN or DEBUG-RUN in order to cancel the step execution.

8.12.2 Setting and Execution of Stop Position Based on Breakpoint Setting

The section describes operations when stopping the ST program at a specific line.
The user can set breakpoints when the ST program is in a stopped state.

(NOTE) The ST program cannot be stopped using breakpoints when the ST program monitoring is not executed or when monitoring is stopped.

8.12.2.1 Breakpoint settings

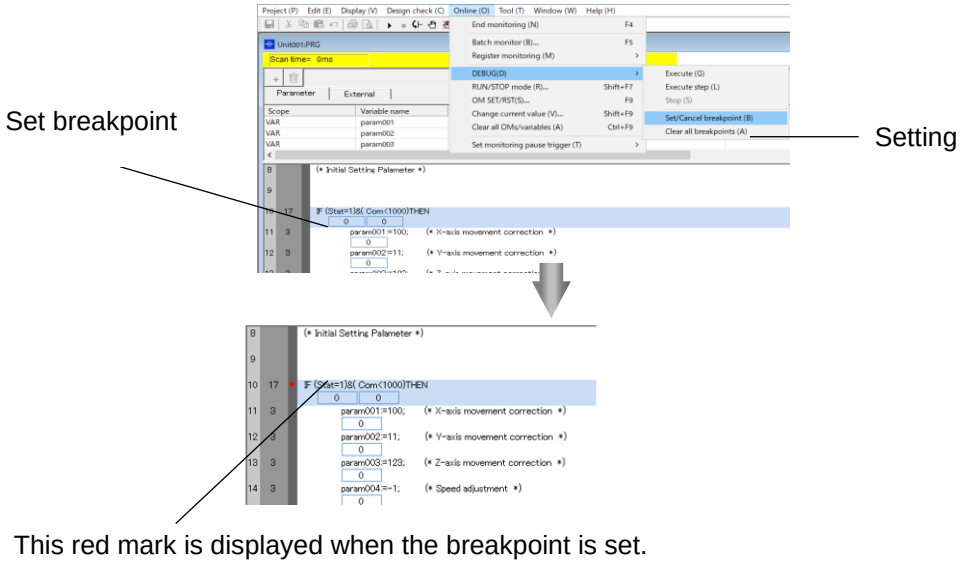
Breakpoints can be set as shown below.

Move the cursor to the desired line to set a stop, and then perform the following operations.

Operation procedure

Perform one of the following operations.

- Select <Set/Cancel breakpoint> under <DEBUG> in the <Online> menu
- Click on the button in the tool bar



The user can perform one of the following operations to cancel the breakpoint.
To cancel only a specific breakpoint:

Operation procedure
Perform one of the following operations.
- Select the desired line where the breakpoint is set, and then select <Set/Cancel breakpoint> under <DEBUG> in the <Online> menu
- Click on the button in the tool bar

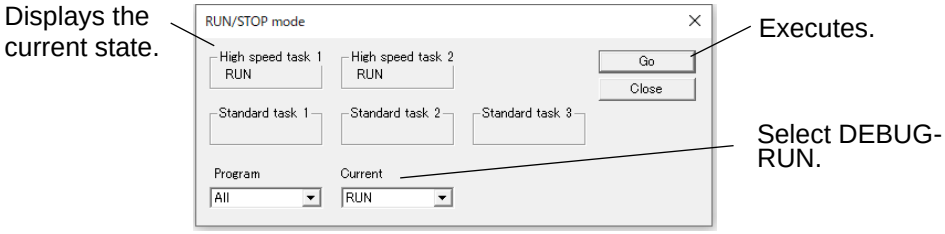
To cancel all breakpoints:
Operation procedure
Perform one of the following operations.
- Select <Clear all breakpoints> under <DEBUG> in the <Online> menu
- Click on the button in the tool bar

8.12.2.2 ST Program debug execution

Use one of the following operations to debug an ST program.
The ST program stops at the line or lines where a breakpoint is set.

Execution using <RUN/STOP mode> dialog

Operation procedure
1. Perform one of the following operations to display the <RUN/STOP mode> dialog.
   Select <RUN/STOP mode> under the <Online> menu
   Press the **[Shift]** + **[F7]** keys
2. Select <DEBUG-RUN>, and then click on the <Execute> button



Execution without using <RUN/STOP mode> dialog

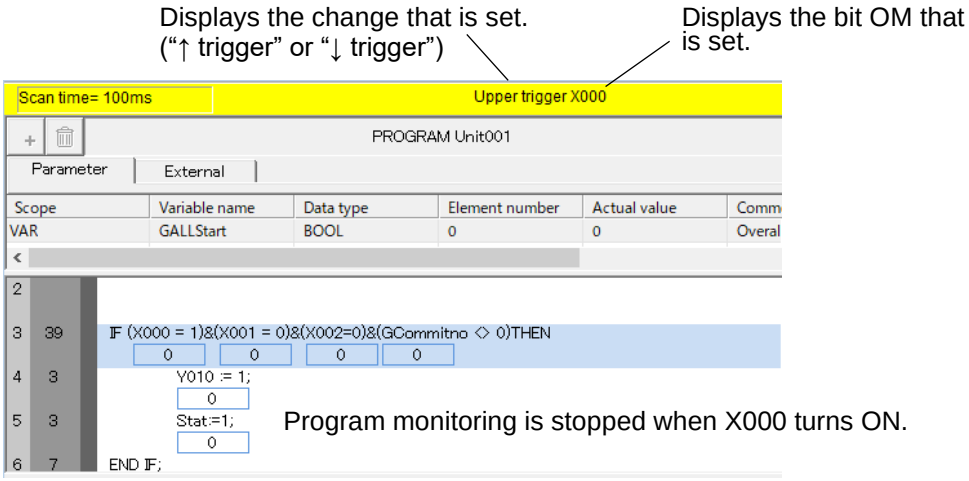
Operation procedure
Perform one of the following operations to execute debugging when the NC(BUILT IN PLC) or simulation is in a stopped state.
- Select <Execute> under <DEBUG> in the <Online> menu
- Click on the button in the tool bar

Perform the same operation again in order to skip to the next breakpoint line.
Perform step execution to only execute the next command (after the stop position) and then stop.

To cancel the “DEBUG-RUN” state, change the setting in the <RUN/STOP mode> dialog to <RUN>.
All the stop conditions are cancelled, and the ST program executes.

8.12.3 Stop Monitoring Using OM Change Setting

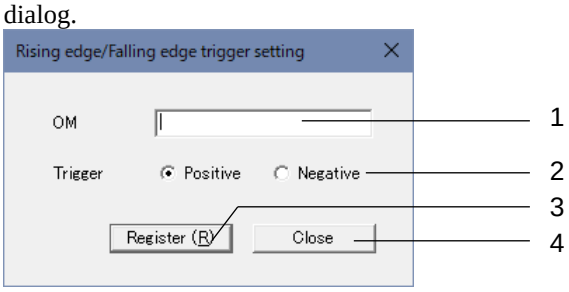
The user can stop the program monitoring using a trigger setting when the bit OM value changes.



Operation procedure

1. Press the **[Ctrl]** + **[F7]** keys or the **[Ctrl]** + **[F8]** keys while the monitoring is executing. Or, select <Rising edge trigger> / <Falling edge trigger> under <Set monitoring pause trigger> in the <Online> menu.

2. Set the bit OM to monitor the value change in the <Rising edge / Falling edge trigger setting> dialog.



No.	Item	Description
1	OM name	Specifies the bit OM to monitor the value change.
2	Trigger	Select <Rising> (Rising edge trigger) to stop the monitoring when OFF changes to ON. Select <Trailing> (Falling edge trigger) to stop the monitoring when ON changes to OFF. Note that if the **[Ctrl]** + **[F7]** keys are pressed to open the dialog, <Rising> is selected. If the **[Ctrl]** + **[F8]** keys are pressed to open the dialog, <Trailing> is selected.
3	<Register> button	Enables the setting, and the dialog closes.
4	<Close> button	Discards the settings, and the dialog closes.

3. Click on the <Register> button.

While performing monitoring in the PLC Programming software D00 EDITOR, only one bit OM can be monitored at a time regardless of the language or POU. After setting one bit OM for monitoring, if the user monitors another POU and sets another bit OM as the bit OM for monitoring, then the last bit OM that was set is enabled for monitoring.

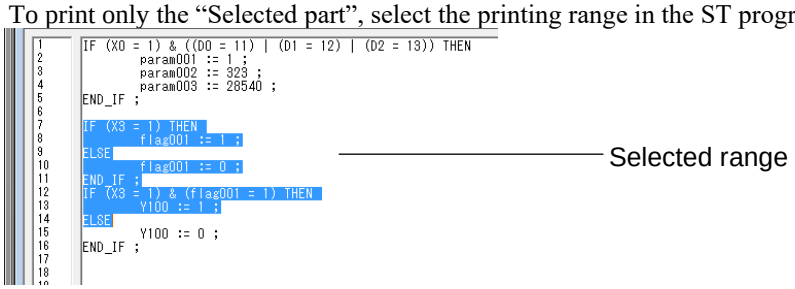
Press the **[F2]** key to cancel the monitoring stop setting (using the bit OM).
In addition, when monitoring is opened again, the bit OM setting for monitoring is cleared.

8.13 Print

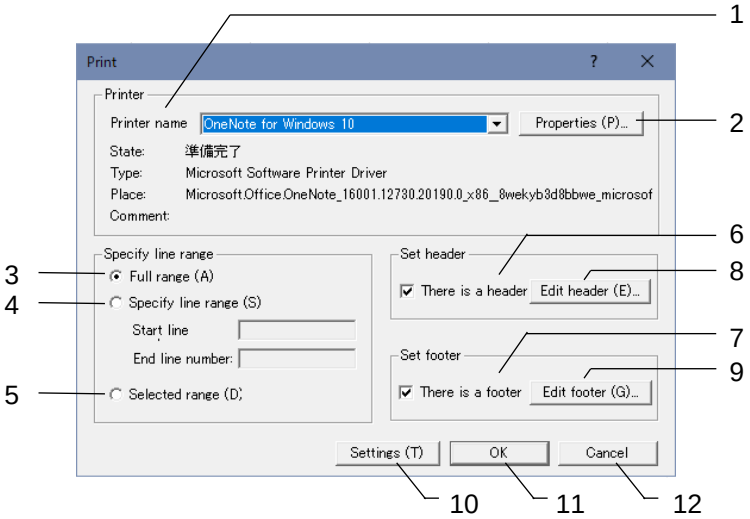
This section describes the ST program print operations.
In the <Print> dialog, the user can select the paper size or paper orientation as well as set the printing range in the ST program when printing. The entire variable area is printed.
The display for the PLC Programming software D00 LD/ST Programming Editor changes to a preview of the printing data and then prints. To change the text font that is printed, change the font setting in the <Option> dialog.

Operation procedure

1. To print only the "Selected part", select the printing range in the ST program display.



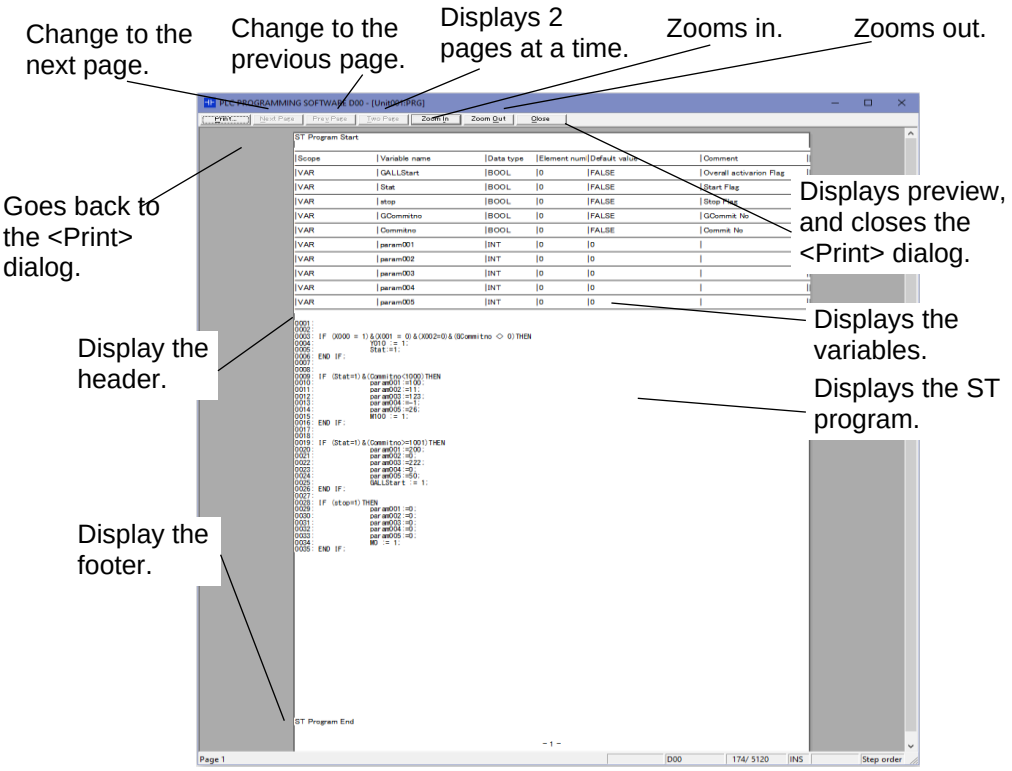
2. Select <Print> under the <Project> menu.



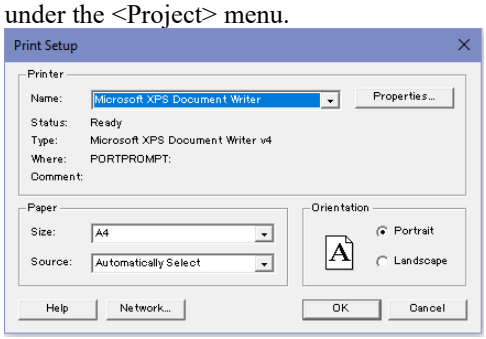
No.	Item	Description
1	Printer name	Selects a printer.
2	<Properties> button	Opens the properties dialog for the printer. (The settings depend on the printer drivers that are installed.)
3	All	Prints the entire range.
4	Specify line range	Prints the range that is specified by the first line number and last line number.
5	Selected part	Prints only the selected part in the ST language edit area.
6	Header	Tick the checkbox to include the header.
7	Footer	Tick the checkbox to include the footer.
8	Edit header	Opens the header edit dialog.
9	Edit footer	Opens the footer edit dialog.
10	Set	Saves the header and/or footer information.
11	<OK> button	Saves the header and/or footer information and starts printing.
12	<Close> button	Closes the dialog.

3. Select a printer in the <Print> dialog.
4. Specify the paper settings under the “Properties” in the <Print> dialog.
5 In the “Specify line range” section, select a printing range from the following options: “All”, “Specify line range” or “Selected part”. When “Specify line range” is selected, specify the first and last lines in the range.
6. Set the “Header settings” and/or “Footer settings” when necessary.
7. Click on the <Yes> button in the <Print> dialog to start printing.

In addition, the user can use “Print preview” to check the image being printed on the paper. Select <Print preview> in the <Project> menu to use print preview.



(NOTE 1) The user can also set the printer, paper size and orientation in the <Printer settings> under the <Project> menu.



(NOTE 2) The settings for the printer properties vary depending on the printer being used. Refer to the printer manual.

(NOTE 3) The header and footer settings are configured in the dialog below. The user can enter a maximum of 512 characters. However, any text after 5 lines is ignored when printing. Therefore, do not exceed the line limit when setting. When the user clicks on the <Set> button or the <OK> button in the <Print> dialog, the editing is saved as part of the program files.

When <OK> is selected, the editing is set and it goes back to the <Print> dialog.
When <Cancel> is selected, the editing is discarded and it goes back to the <Print> dialog.

8.14 Display Manuals

This section describes how to display a manual.
The user can display the different manuals in the PLC Programming software D00.

To display the manuals, the user must install an application software beforehand that can display PDF files, such as Acrobat Reader provided by Adobe.

- User's manual
  Operation procedure
  Select the <User's manual> under <Manual> in the <Help> menu.

- Operation manual
  Operation procedure
  Select the <Operation manual> under <Manual> in the <Help> menu.

- Programming manual
  Operation procedure
  Select the <Programming manual> under <Manual> in the <Help> menu.

(This page was intentionally left blank.)

CHAPTER 9

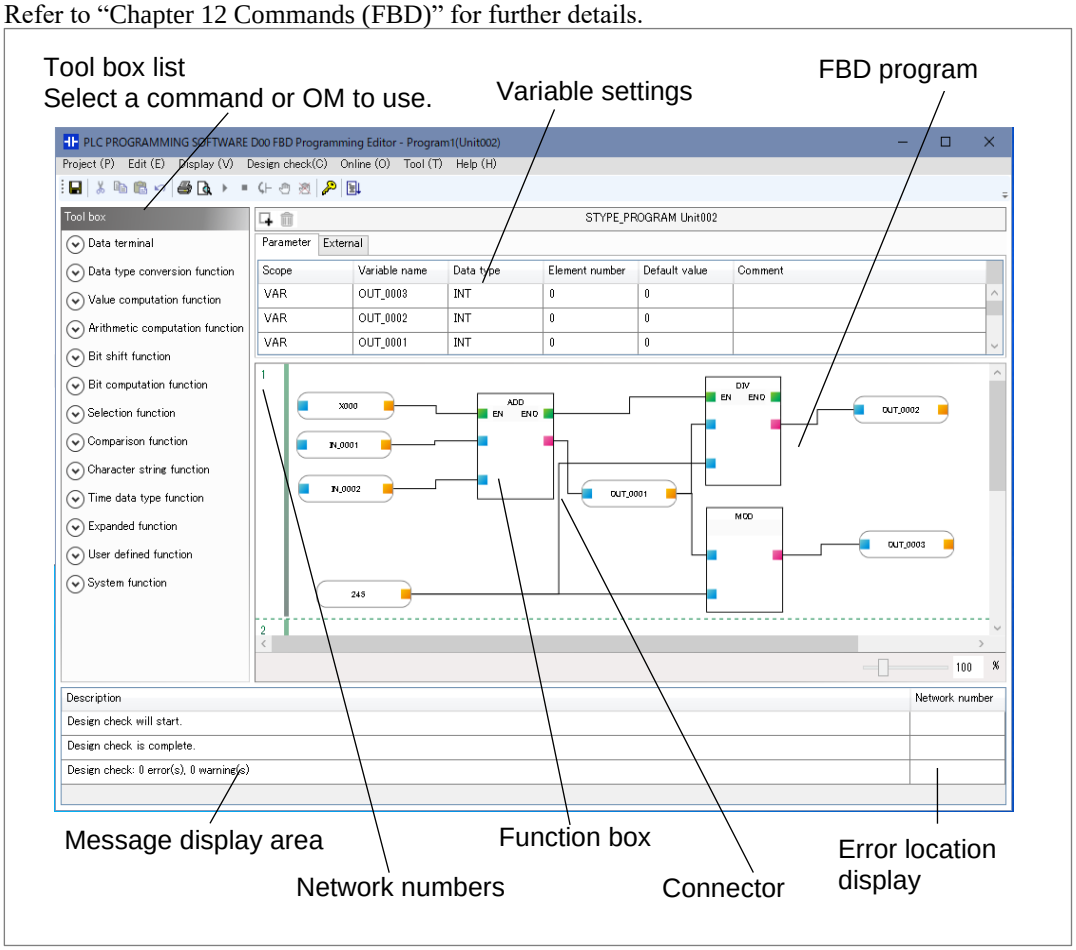
FBD EDIT OPERATION PROCEDURE

9.1 FBD Program & Edit Screen Layout
9.2 Variable Settings
9.3 Edit FBD Programs
9.4 Copy, Cut, Paste & Delete
9.5 Replace
9.6 Search
9.7 Edit Lock
9.8 Launch Simulation
9.9 Monitoring FBD Program in NC(BUILT IN PLC)
9.10 Run & Stop in NC(BUILT IN PLC)
9.11 Monitoring Operations
9.12 DEBUG Function
9.13 Print
9.14 Display Manuals

9.1 FBD Program & Edit Screen Layout

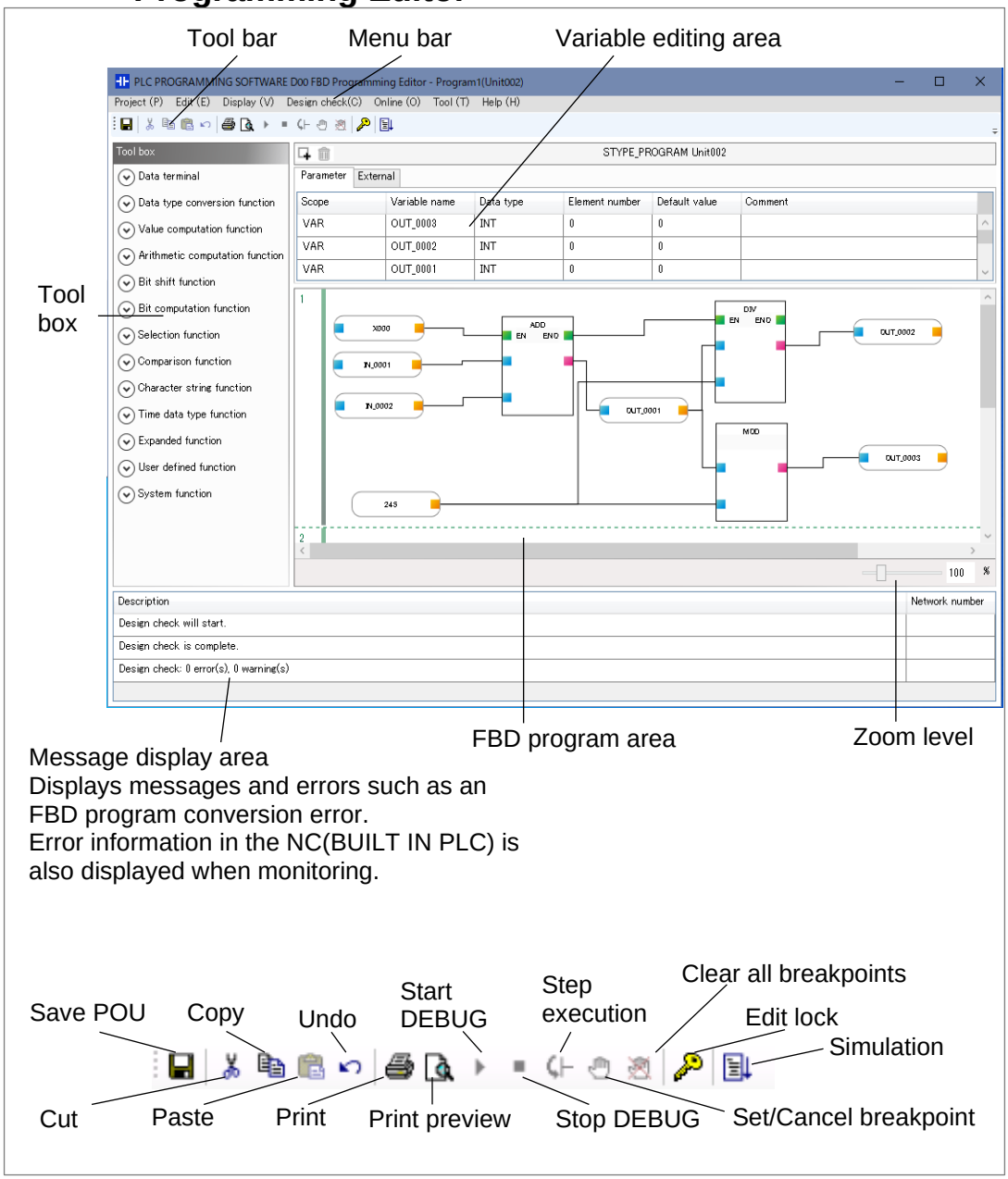
This section describes the FBD program, its display and the layout of the edit screen.

The screenshot below shows the layout of the FBD program display.
Refer to “Chapter 12 Commands (FBD)” for further details.

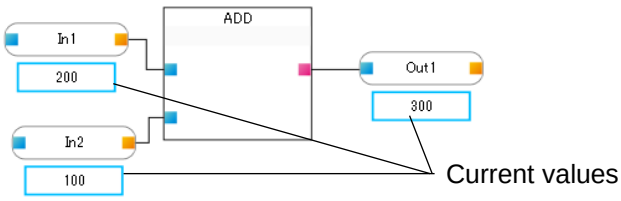


The user can improve the visibility and organization of the FBD program in the PLC Programming software D00 by creating the program by area or what are called “networks” (unit: circuits).
The FBD program is executed following the order of the “Network number”.

9.1.1 Layout of PLC Programming software D00 FBD Programming Editor



(NOTE) When monitoring an FBD program, the current value is displayed under the variable boxes and OM boxes.



Be careful when positioning the boxes, because if the variable box or OM box is too close to another box, or if it is positioned close to the network boundary, then the current value display may get cut off due to overlapping with another box.

9.2 Variable Settings

This section describes variables used in an FBD program.
Variables are set for each POU.

9.2.1 Create & Edit Parameter Variables

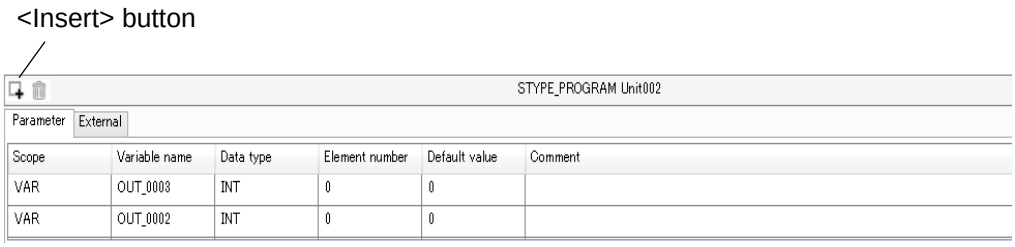
9.2.1.1 Create new parameter variable

The user can perform the following operations in the variable editing area to create a new parameter variable.

Operation procedure

Perform one of the following operations under the <Parameter> tab.

- Click on the <Insert> button.
- Right-click the mouse on the variable setting field, and then select <Insert> from the menu.



The following number of variables can be set in each FBD program edit window.
BOOL :1024 variables
Non-BOOL :1024 variables (The number of variables that can be used varies according to the type.)

Type	Qty. equal to one BOOL variable	Type	Qty. equal to one BOOL variable	Type	Qty. equal to one BOOL variable
BYTE	1	WORD	1	DWORD	2
INT	1	DINT	2	LINT	4
UINT	1	UDINT	2	ULINT	4
REAL	2	LREAL	4	DATE	4
TEME	4	TOD	4	DT	4
STRING	16				

For example, when only registering WORD type variables, the user can register a maximum of 1024 variables. However, when only registering DWORD type variables, the user can register a maximum of 512 variables.
When registering arrays, the total element number for each array that can be registered cannot exceed the maximum number of BOOL type or non-BOOL type variables.
Refer to “12.1.2.1 List of data types” for further details on data types.

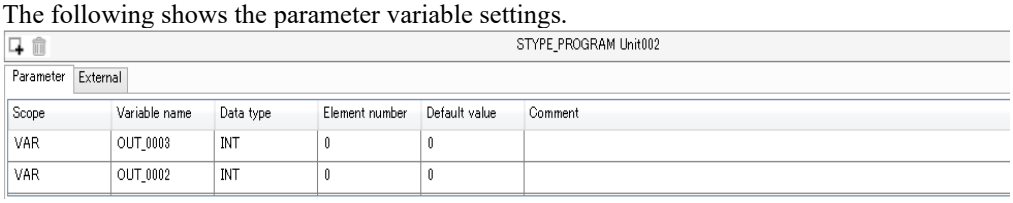
After registering 1024 BOOL type variables, the user must perform the following procedure to register non-BOOL type variables.

Operation procedure

1. Delete a BOOL type variable. Or, change it to another data type.
   (Write down the variable settings before deleting or changing them.)
2. Register a non-BOOL type variable.
3. Re-register the variable that was deleted in step 1. Or, return the data type (that was changed in step 1) back to the BOOL type.

9.2.1.2 Parameter variable settings

The following shows the parameter variable settings.



Scope	Variable name	Data type	Element number	Default value	Comment
VAR	OUT_0003	INT	0	0	
VAR	OUT_0002	INT	0	0	

Item	Description
Scope	Selects a variable range from the following. • VAR : Variable only used within POU • VAR_OUTPUT : Variable where output value to POU (call source) is assigned • VAR_INPUT : Variable where input value from call source is assigned when POU is called • VAR_IN_OUT : Variable where input value from call source is assigned and where output value to POU (call source) is assigned when POU is called "VAR" is fixed in a high speed and standard control program.
Variable name	Sets the variable name. • The user can set a maximum of 32 characters. • Single-byte alphanumeric characters and a half-width underscore (_) can be used. • Numbers cannot be used for the first character. • The name of another parameter variable cannot be duplicated in the same POU. • The variable name cannot be the same as any reserved words.
Data type	Selects the variable data type from the following. BOOL/BYTE/WORD/DWORD/INT/DINT/LINT/UINT/UDINT/ULINT/REAL/LREAL/DATE/TIME/TOD/DT/STRING
Element number	Sets the element number of the array between 1 and 1024 when being used as an array variable. The number is set to 0 when it is not being used as an array variable.
Default value	Sets the default value of the variable when the NC(BUILT IN PLC) is launched. The values that can be set vary depending on the data type. Refer to "12.1.2.1 List of data types" for further details. This is the default value for all elements when using an array.
Comment	No more than 32 characters can be used for a comment.

(NOTE 1) After registering a variable, the user can change the settings of a variable being used in an FBD program. However, the program can become invalid depending on the content that was changed (for example, when a change is made to a data type that cannot be used in a function).
Therefore, proceed carefully when changing the settings of a variable being used in an FBD program.

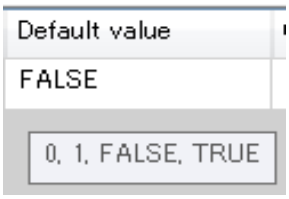
(NOTE 2) When a default value is set for a REAL or LREAL type variable, then a rounded off value is set to the valid digits (7 digits and 15 digits, respectively). Examples are shown below.

Data type	Input value	Set value
REAL type	1234567	1234567
	12345678	1.234568E+07
	0.0001234567	0.0001234567
	0.00012345678	0.0001234568
LREAL type	123456789012345	123456789012345
	1234567890123456	1.23456789012346E+15
	0.000123456789012345	0.000123456789012345
	0.0001234567890123456	0.000123456789012346

9

(NOTE 3) If a name of a variable being used in an FBD program is changed, that name change is automatically reflected in the variable box.

(NOTE 4) When the mouse cursor is moved to the <Default value> field in the variable editing area, then the valid input range according to the data type of that variable is displayed.

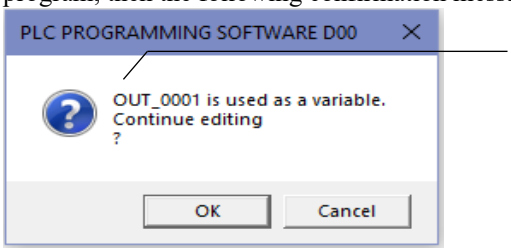


(NOTE 5) The default value of the array variable is the default value for all the elements of that array variable. To set individual values for each element, the user must set them in the program.

(NOTE 6) Refer to “12.1.2.1 List of data types” for further details on data types.
Refer to “Chapter 12 Commands (FBD) ” for details about reserved words.

(NOTE 7) The user can enter a user or system defined function name for the variable name, but an error will trigger at conversion.

If the data type or the element number is changed for a parameter variable used in an FBD program, then the following confirmation message is displayed.



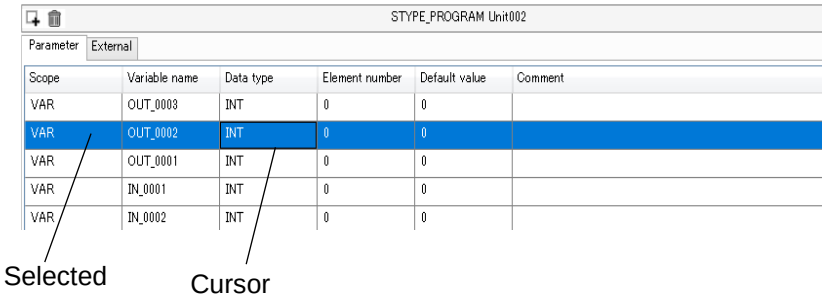
Displays name of variable with setting change.

Select <OK> to proceed with the setting change.
Select <Cancel> to cancel the setting change.

When a setting has been changed, the FBD program needs to be modified. Refer to “9.3.1.10 Change setting for variable, user defined function or system defined function”.

9.2.1.3 Variable edit operations

This section describes the variable edit operations.



To edit a variable, move the cursor to the edit location, press the **[Space]** key, or use the mouse to single-click at the cursor position. The position of the cursor becomes editable.
The cursor can be moved using the following operations.

- Single-click on mouse
- Press the **[→]**, **[←]**, **[↑]** or **[↓]** key
- **[Ctrl]+[→]** : Moves to <Comment> field on a line
- **[Ctrl]+[←]** : Moves to <Scope> field on a line
- **[Ctrl]+[↑]** : Moves up to first line in a column
- **[Ctrl]+[↓]** : Moves down to last line in a column
- **[Enter]** : Moves down one line in a column
- **[Tab]** : Moves to next field (column)
  Moves to <Scope> field on next line when at <Comment> field
- **[Shift]+[Tab]** : Moves to previous field (column)
  Moves to <Comment> field on previous line when at <Scope> field
- **[Page Up]** : Scrolls up, jumping to previous full screen
  Moves up to first line if scroll bar is not displayed in variable editing area
- **[Page Down]** : Scrolls down, jumping to next full screen
  Moves down to last line if scroll bar is not displayed in variable editing area
- **[Home]** : Moves to <Scope> field on a line
- **[Ctrl]+[Home]** : Moves to <Scope> field on first line
- **[End]** : Moves to <Comment> field on a line
- **[Ctrl]+[End]** : Moves to <Comment> field on last line

• When <Scope> or <Data type> is in an editing state
  - Press <▼> to display the list and then select. Press the **[Enter]**, **[Tab]** or **[Shift]** + **[Tab]** keys to set the content.
  - Press **[↓]** or **[Ctrl]** + **[↓]** keys to display the list items below and then select. Press the **[Enter]**, **[Tab]** or **[Shift]** + **[Tab]** keys to set the content.
  - Press **[↑]** or **[Ctrl]** + **[↑]** keys to display the list items above and then select. Press the **[Enter]**, **[Tab]** or **[Shift]** + **[Tab]** keys to set the content.
  - **[Esc]** : Changes the selected data back to previous data (like undo operation).
  - **[Home]** : Selects first item in list.
  - **[End]** : Selects last item in list.

• When <Variable name>, <Default value> or <Comment> is in an editing state
  - Enter characters and press the **[Enter]**, **[Tab]** or **[Shift]** + **[Tab]** keys to set the content.
  - **[←]** : Moves cursor to previous character on left.
  - **[→]** : Moves cursor to next character on right.
  - **[Backspace]** : Deletes character before cursor.
  - **[Delete]** : Deletes character after cursor.
  - **[Ctrl]+[↓]**, **[End]**, **[Ctrl]+[End]** : Moves cursor to right end
  - **[Ctrl]+[↑]**, **[Home]**, **[Ctrl]+[Home]** : Moves cursor to left end
  - **[Insert]** : Switches between inserting a character or overwriting one.

9.2.1.4 Re-use (copy)

In the variable setting area, the user can copy a variable, and then insert and paste it as another variable.
An underscore (_) is added to the variable name that is pasted.
<Copied variable name>_
If a variable is pasted again, then a number is added after the underscore (_).
<Copied variable name>_1, <Copied variable name>_2, <Copied variable name>_3…

The user can use the edit operations to change the variable name that was pasted.

Use the following operations to select a variable to copy.
- Click : Selects the variable that was clicked on
- **[Shift]** + Click : Specifies a range and selects multiple variables
- **[Ctrl]** + Click : Selects multiple variables that were clicked on

The user can perform one of the following operations to copy.
Operation procedure
One of the following operations can be carried out:
- Select <Copy> from the <Edit> menu
- Press the **[Ctrl]** + **[C]** keys
- Click on the button in the tool bar
- Right-click the mouse and select <Copy> from the displayed menu

The user can perform one of the following operations to paste.
Operation procedure
One of the following operations can be carried out:
- Select <Paste> from the <Edit> menu
- Press the **[Ctrl]** + **[V]** keys
- Click on the button in the tool bar
- Right-click the mouse and select <Paste> from the displayed menu

When performing a paste operation, the copied variable can be inserted and pasted at the variable position that was last selected.

(NOTE) The copied parameter variable cannot be pasted to the <External> tab in the variable editing area.

9.2.1.5 Cut and delete variables

This function cuts and deletes registered variables.
The cut variables can also be pasted.

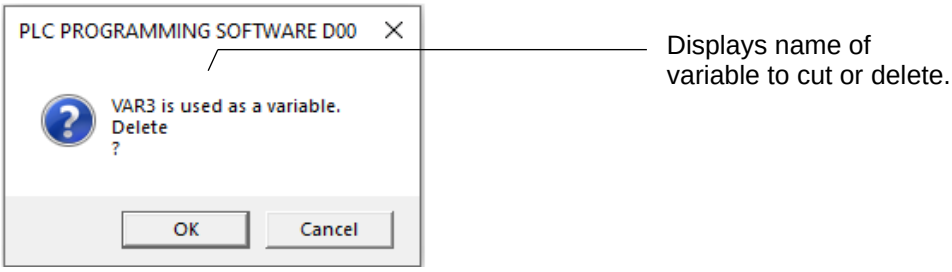
The user can perform one of the following operations to cut.
Operation procedure
1. Select a variable to cut.
2. One of the following operations can be carried out:
   - Select <Cut> from the <Edit> menu.
   - Press the **[Ctrl]** + **[X]** keys
   - Click on the button in the tool bar
   - Right-click the mouse and select <Cut> from the displayed menu.

The user can perform one of the following operations to delete.

Operation procedure

1. Select a variable to delete.
2. One of the following operations can be carried out:
   - Click on the button.
   - Press the **[Delete l]** keys
   - Right-click the mouse and select <Delete> from the displayed menu.

If the user cuts or deletes a parameter variable used in an FBD program, then the following confirmation message is displayed.



Select <OK> to proceed with the cut or delete operation.
Select <Cancel> to cancel the cut or delete operation.

After performing a cut or delete operation, the FBD program needs to be modified. Refer to "9.3.1.10 Change setting for variable, user defined function or system defined function".

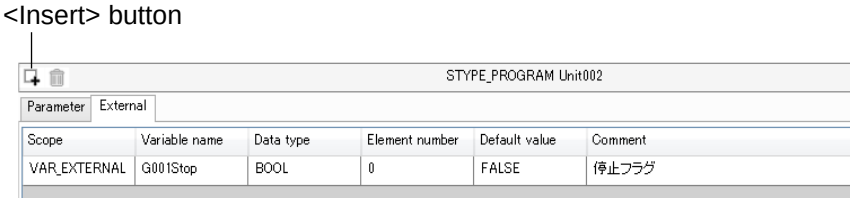
9.2.2 Create & Edit External Variables

9.2.2.1 Create new external variable

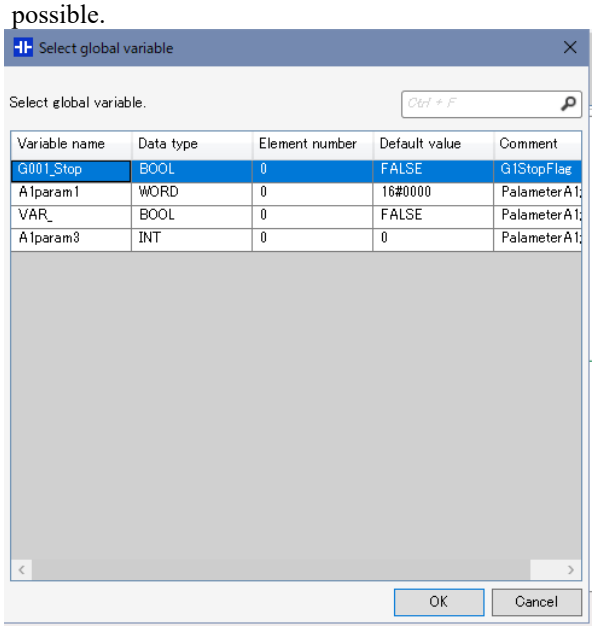
The user can perform the following procedure in the variable editing area to create a new parameter variable.

Operation procedure

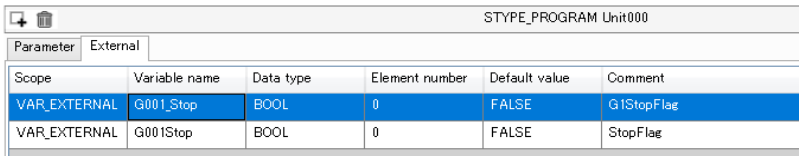
1. Perform one of the following operations under the <External> tag.
   - Click on the <Insert> button.
   - Right-click the mouse on the variable setting field, and then select <Insert> from the context menu.



2. The <Select global variable> dialog is displayed.
   The registered global variables are displayed. (Global variables already registered as external variables are not shown.)
   Select a global variable to be used as an external variable. Multiple selections are also possible.



3. Click on the <OK> button.
4. The external variable is added.



When registering external variables, the external variable settings cannot be changed because that variable is registered as a global variable.
"VAR_EXTERNAL" is set for the scope of the external variable.

Click on the <Cancel> button in the <Select global variable> dialog to cancel registering an external variable.

(NOTE 1) In the PLC Programming software D00 <Global variable> dialog, if the user deletes or changes a global variable being used as a POU external variable, that change will not be automatically reflected in the POU external variable settings.
In this situation, re-register the external variable in the POU.

(NOTE 2) In the <Select global variable> dialog, the user can specify a character string to search for a variable name that matches that character string.
If there are multiple hits for that variable name, then each time the user clicks on the <Start search> button, the search jumps to the next variable name. The letters in the search are not case sensitive. If there are no hits that match the variable name specified in the character string, then the selection bar will not move.

9.2.2.2 Change external variable

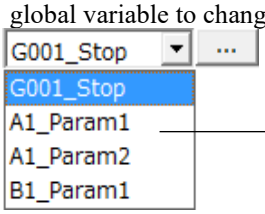
This function changes the source of the registered external variable to another global variable.

Operation procedure

1. Under the <External> tab in the variable editing area, switch the focus to rename the external variable name, and then click on the variable name.

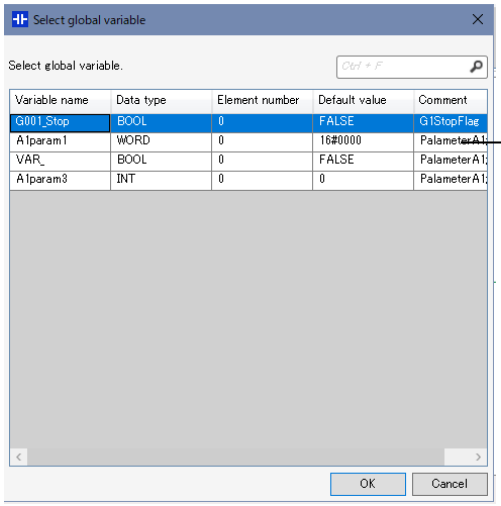
Two buttons will display to the right of the variable name.

2. After clicking on the button, a list of the global variables is displayed. Select the desired global variable to change.



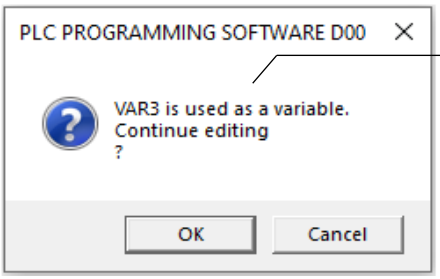
Select a global variable to change.
(Variable names that are already registered as external variables are not shown.)

3. After clicking on the button, the <Select global variable> dialog is displayed. Select the desired global variable to change.



Select a global variable to change.
(Variable names that are already registered as external variables are not shown.)

4. Click on the <OK> button.
   Click on the <Cancel> button in the <Select global variable> dialog to cancel the external variable change.
   If the user makes a change to an external variable used in an FBD program, then the following confirmation message is displayed.



Displays name of variable with setting change.

Select <OK> to proceed with the setting change.
Select <Cancel> to cancel the setting change.

When a setting has been changed, the FBD program needs to be modified. Refer to “9.3.1.10 Change setting for variable, user defined function or system defined function”.

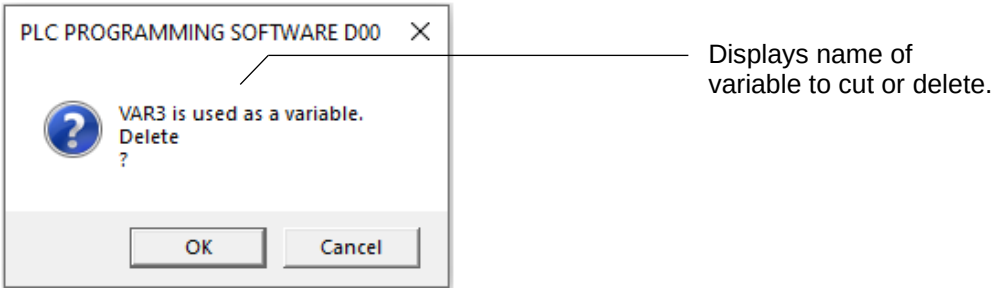
9.2.2.3 Cut and delete external variables

To cut or delete a registered external variable, select the desired variable and then perform one of the following operations.

The user can perform one of the following operations to cut.
Operation procedure
1. Select a variable to cut.
2. One of the following operations can be carried out:
   - Select <Cut> from the <Edit> menu.
   - Press the **[Ctrl]** + **[X]** keys
   - Click on the button in the tool bar
   - Right-click the mouse and select <Cut> from the displayed menu.

The user can perform one of the following operations to delete.
Operation procedure
1. Select a variable to cut.
2. One of the following operations can be carried out:
   - Click on the button.
   - Press the **[Delete]** keys
   - Right-click on the mouse and select <Delete> from the displayed menu.

If the user cuts or deletes an external variable used in an FBD program, then the following confirmation message is displayed.



Select <OK> to proceed with the cut or delete operation.
Select <Cancel> to cancel the cut or delete operation.

After performing a cut or delete operation, the FBD program needs to be modified. Refer to "9.3.1.10 Change setting for variable, user defined function or system defined function".

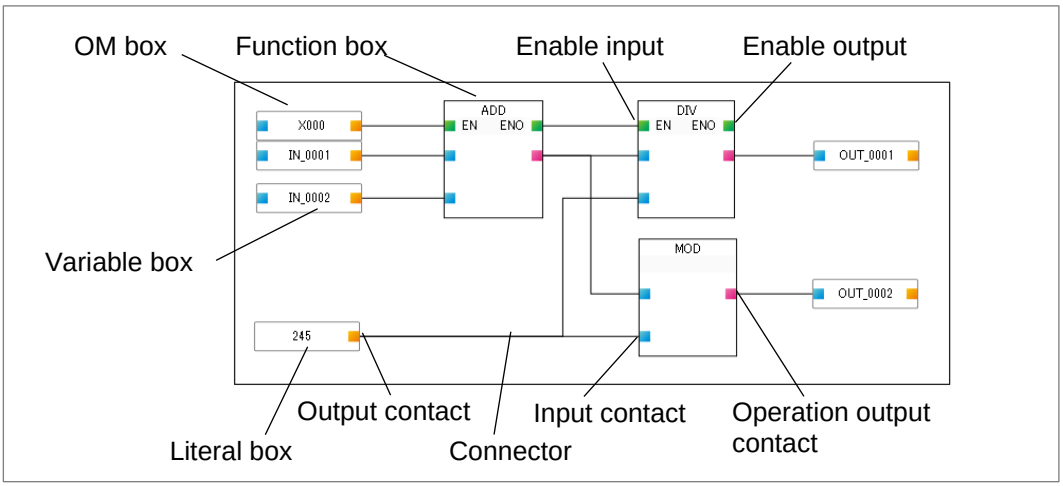
(NOTE) Even if the external variable registered in the POU is deleted, the original global variable (source) itself is not deleted.

9.3 Edit FBD Programs

This section describes the operations to edit an FBD program.
Refer to “Chapter 12 Commands (FBD)” for details on how to create an FBD program.

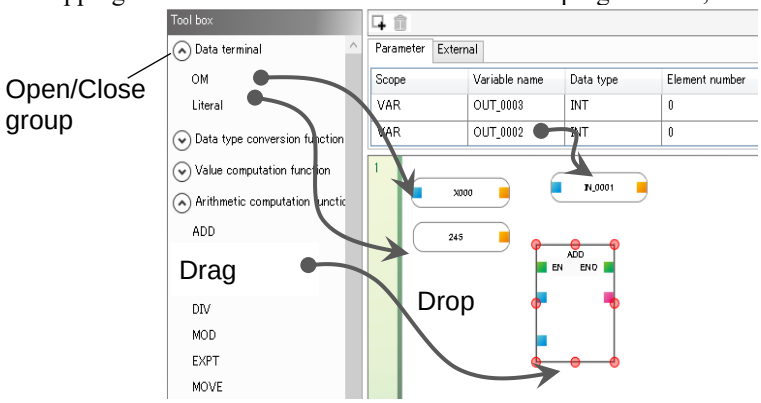
9.3.1 Operations for Creating FBD Programs

An FBD program can be created by setting up different components, such as a <Function box>, <OM box>, <Literal box>, <Variable box> and <Return value box>, and are then connected via a <Connector>.

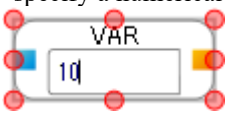


The <Function box>, <OM box>, <Literal box> and <Return value box> can be inserted from the <Tool box list> into the program area by using a drag & drop operation. Similarly, the <Variable box> can be inserted from the variable editing area into the program area by also using a drag & drop operation.

- Insert a <Function box>, <OM box>, <Literal box> and <Variable box>
  After dropping an <OM box> or <Literal box> into the program area, a value needs to be set.



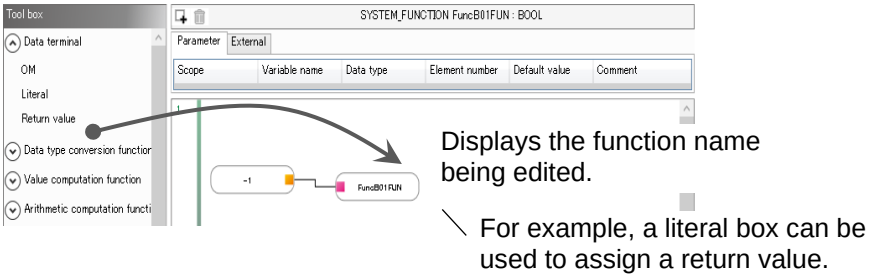
Double-click on the <Variable box> to specify the element for an array variable.
When the data input field appears under the variable name, enter the element. The user can specify a numerical value, a variable or an OM for the element.



Press the **[Enter]** key or click outside of the variable box to set the data.
When the data has been set, the element is automatically enclosed with brackets **[ ]**. There is no need for the user to input them.

There is no need to enter an element when using a variable box that refers to all the array variables.

- Insert a <Return value box>
  The <Return value box> can be used with a user or system defined function.
  This box can be inserted from the <Tool box list> into the program area by using a drag & drop operation.



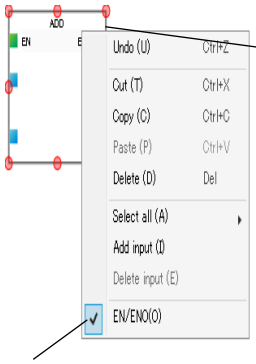
Refer to “Chapter 12 Commands (FBD)” for details on how to create a user defined function and system defined function.

9.3.1.1 Enable Input/Output for Function Box (EN/ENO)

The user can display the enable input/output (EN/ENO) for the function box by using the following operations.

Operation procedure

Use the mouse to specify the function box, and then select <EN/ENO> from the displayed menu.



Enable input/output connector (EN/ENO) is added.

Checkmark "✓" appears in front of item.

Perform the same operation again to hide EN/ENO.

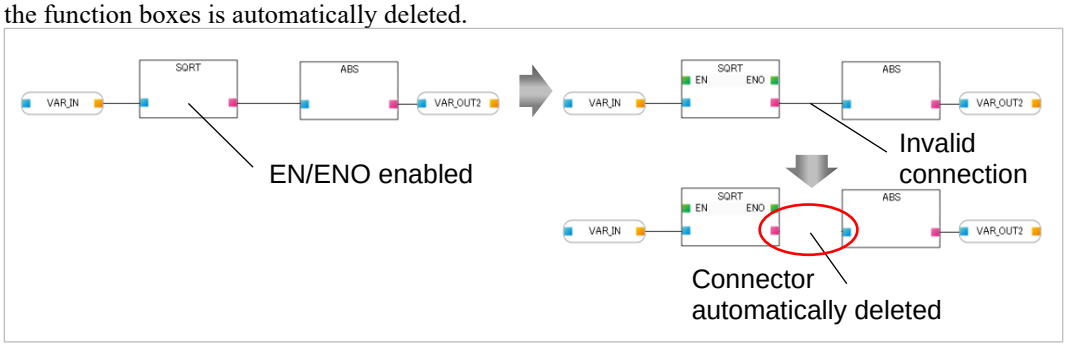
Special notes when using EN/ENO

The output on the function box (where EN/ENO is enabled) cannot be connected directly with the EN/input on another function box. This connection needs to be done through or via a variable or OM box. A direct connection is not possible because the output value becomes undefined when EN is FALSE.

Correct example

Incorrect example

However, when the EN/ENO is enabled and an invalid connection (like the incorrect example) is used, a confirmation message is displayed and then the connector between the input and output of the function boxes is automatically deleted.



Refer to "Chapter 12 Commands (FBD)" for further details about EN/ENO.

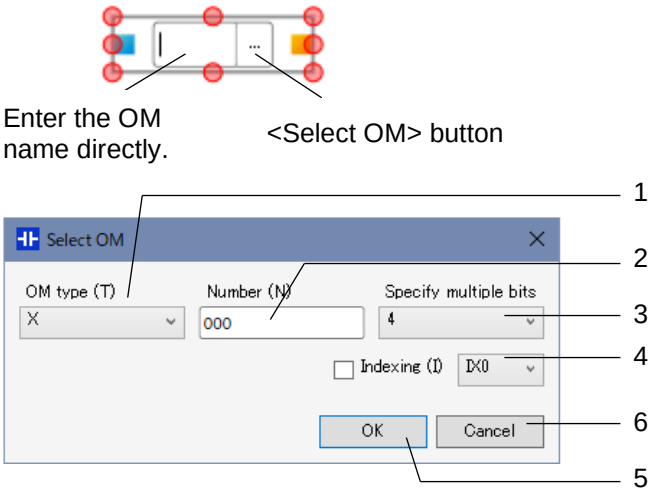
9.3.1.2 OM Settings in OM Box

After dropping the OM box in the program area, the user can perform one of the following operations for the settings.

Operation procedure

Perform one of the following operations to set the OM.

- Enter an OM name directly into the data input field.
- Use the <Select OM> button to set the OM in the <Select OM> dialog.



No.	Item	Description
1	OM type	Selects an OM type. However, T and C cannot be used in an FBD program.
2	No.	Specifies the OM number.
3	Specify multiple bits	Specifies the bit number when specifying multiple bits. The available bit number for the selected OM is displayed. When using the word OM, the user can specify a longword (L(32)).
4	Index addressing	When performing index addressing on a specified OM name, tick the checkbox to select an index number.
5	<OK> button	Applies the setting changes, and the dialog closes.
6	<Cancel> button	Discards the setting changes, and the dialog closes.

When specifying multiple bits for a bit OM, set a multiple of 4 for the number. When setting a word OM, only “L(32)” is displayed for the multiple bit selection. After specifying a multiple bit setting, if the setting accesses an OM number outside of the range, an error is detected at conversion.

To enable the editing on an OM box again, double-click the mouse at a position where the cursor changes to an “I” shape.

Refer to “12.1.2.3 Specifying indexing and digit-specified access for OMs” for details on specifying the index and multiple bit digits.

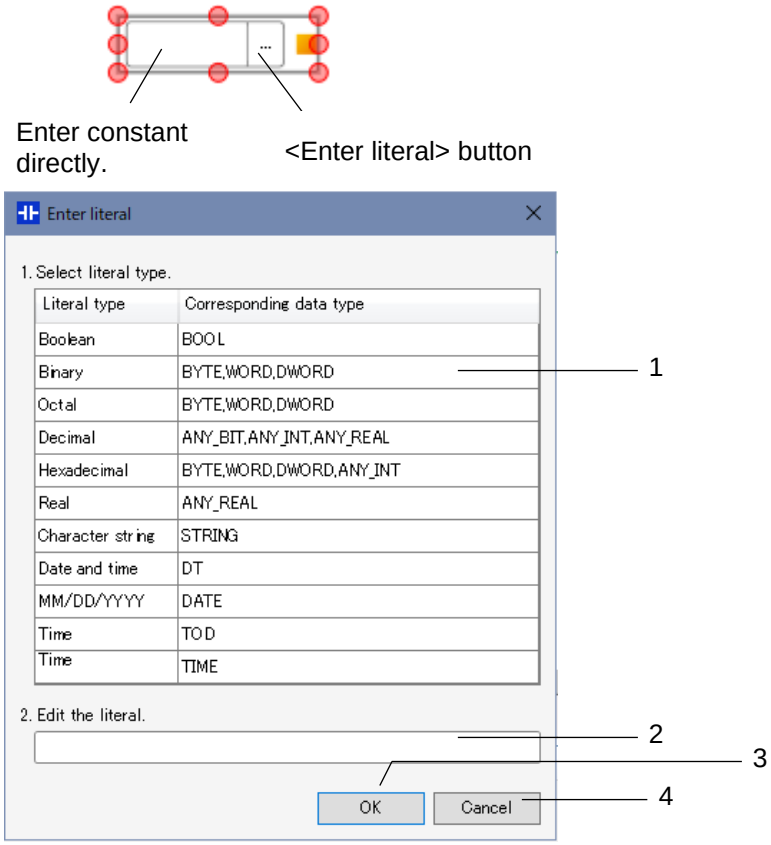
9.3.1.3 Value Setting in Literal (Constant) Box

After dropping the literal box in the program area, the user can perform one of the following operations for the settings.

Operation procedure

Perform one of the following operations to set the literal.

- Enter a literal directly into the data input field. The input values vary depending on the data type.
- Use the <Enter literal> button to set the value in the <Enter literal> dialog.



No.	Item	Description
1	Literal type	Selects the data type.
2	Value	Displays information and a reference about the data types. Follow the reference and set the value.
3	<OK> button	Applies the setting changes, and the dialog closes.
4	<Cancel> button	Discards the setting changes, and the dialog closes.

The values that can be set vary depending on the data type. Refer to “12.1.2.1 List of data types” for further details.

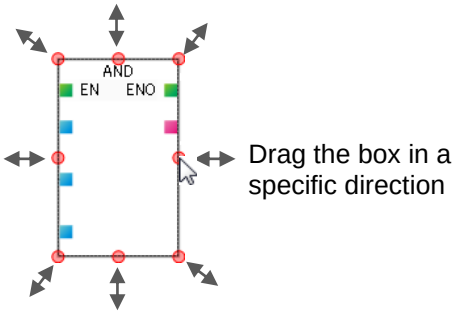
To enable the editing on the literal box again, double-click the mouse at a position where the cursor changes to an “I” shape.

9.3.1.4 Box size adjustment

This section describes how to adjust the display size of a function box, OM box, literal box and variable box.

Operation procedure

1. Select the desired function box, OM box, literal box or variable box to adjust the size.
2. Align the cursor with the red dots on the box and drag it outward (left, right, up or down).



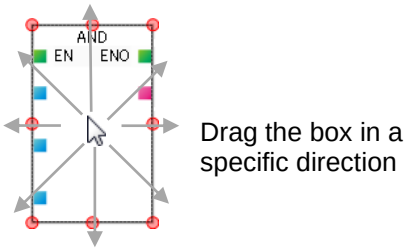
(NOTE) When a variable name or constant does not fit (display) entirely in a box, the data that is hidden due to the size of the box is represented by three dots.
Extend the box outward to show all of the data (for the variable name, etc.).

9.3.1.5 Drag & Move Box

This section describes how to move a function box, OM box, literal box and variable box.

Operation procedure

1. Select the desired function box, OM box, literal box or variable box to move.
2. Align the cursor at a position that is not at the red dots or connector, and drag the box to move it left, right, up or down.

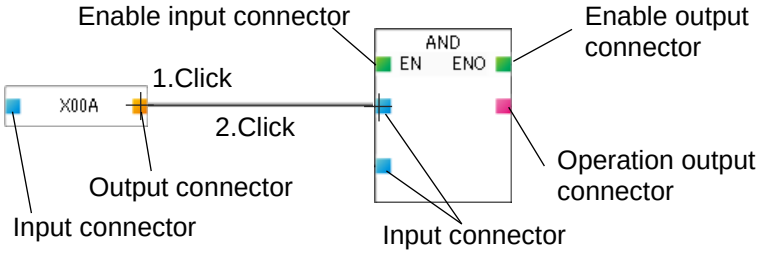


9.3.1.6 Connector

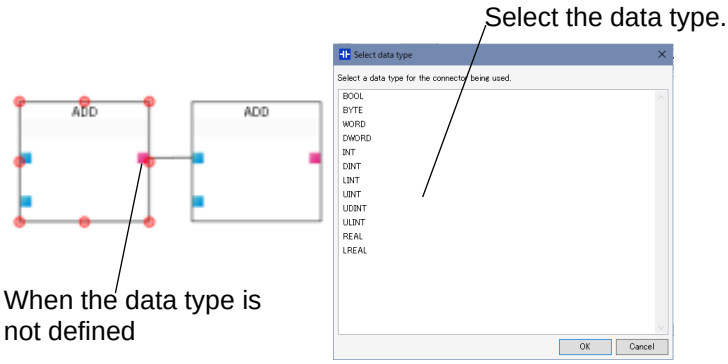
This section describes how to use a connector to link a function box, OM box, literal box and/or variable box.

Operation procedure

1. Align the cursor with a box connector and click on it.
   When the cursor is aligned with the connector, the cursor changes to a plus sign (+).
2. Align the cursor with the other box connector and click on it.
   Right-click on the mouse to cancel the operation before connecting the boxes.



3. When connecting a function box output directly to the input of another function box, the data type may need to be specified. When necessary, the user can select the data type in the <Select data type> dialog to link the connectors.



4. After clicking on the <OK> button in the <Select data type>, the dialog closes and the connector is created.
   After clicking on the <Cancel> button, the dialog closes without creating the connector.

(NOTE 1) Only one connector can be connected for the input connector and the enable input connector. Multiple connecting lines cannot be connected.
Multiple connectors can be connected for the output connector, the operation output connector and the enable output connector.

(NOTE 2) When an attempt is made to connect boxes that cannot be connected, an error is displayed.

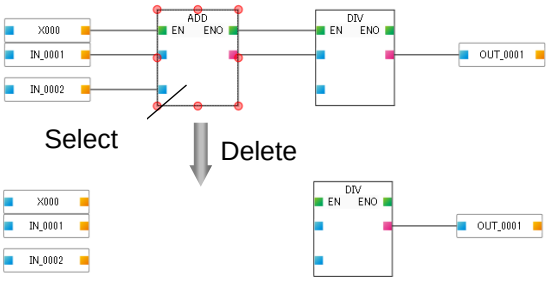
9.3.1.7 Delete boxes and connectors

This section describes how to delete a function box, OM box, literal box, variable box and connector.

(NOTE) When a function box, OM box, literal box or variable box is deleted, the connecting connector is also deleted at the same time.

Operation procedure

Use the mouse to specify the desired box or connector, and then press the **[Delete]** key.



9.3.1.8 Insert and delete networks

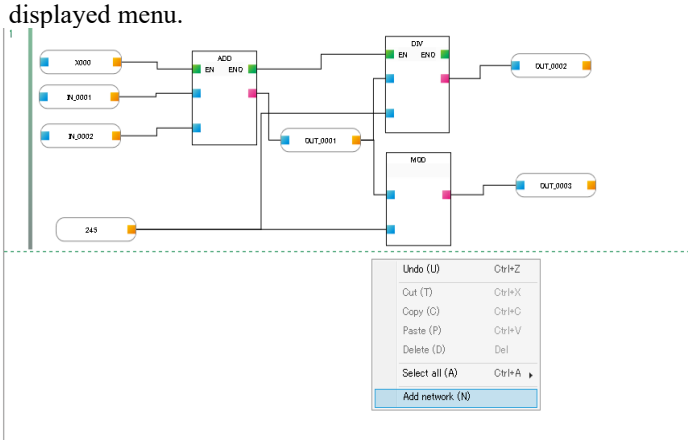
The network operations to insert and delete are as follows.

When inserting a network, it is inserted above the selected network.
When inserting after the last network, perform the operation in the blank space under the last network.
When multiple networks are selected, a network cannot be inserted.

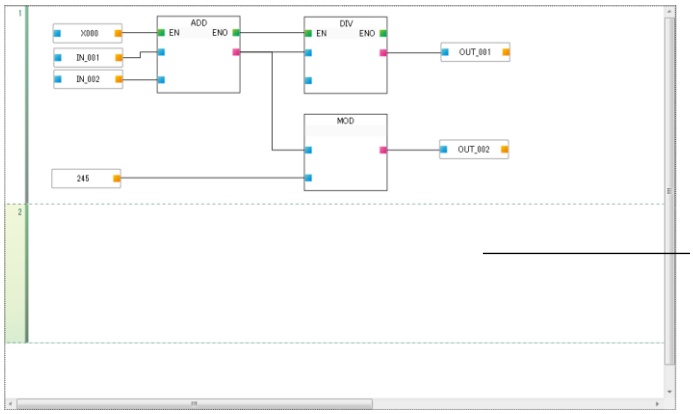
- Insert a network
  The procedure to insert a network is shown below.

Operation procedure

1. Right-click the mouse at the desired position to insert, and select <Add network> from the displayed menu.



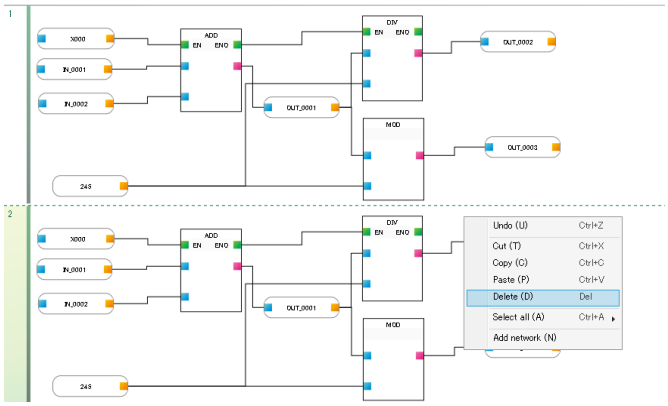
2. The network is added and inserted.



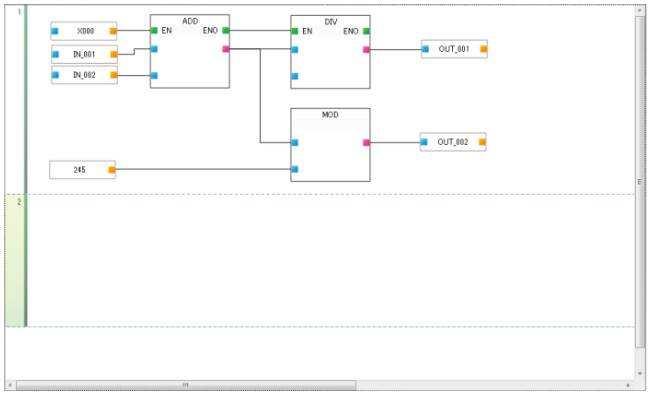
- Delete a network
  The selected network is deleted.
  When multiple networks are selected, all the selected networks are deleted.
  The procedure to delete a network is shown below.

  Operation procedure

  1. Right-click the mouse at the desired network, and select <Delete> from the displayed menu.
     The user can also press the **[Delete]** key, or select <Delete> from the <Edit> menu.



  2. The network is deleted.



9.3.1.9 Change network size

The section below describes how to change the size of a network.

Operation procedure

Drag the boundaries of the network up or down.

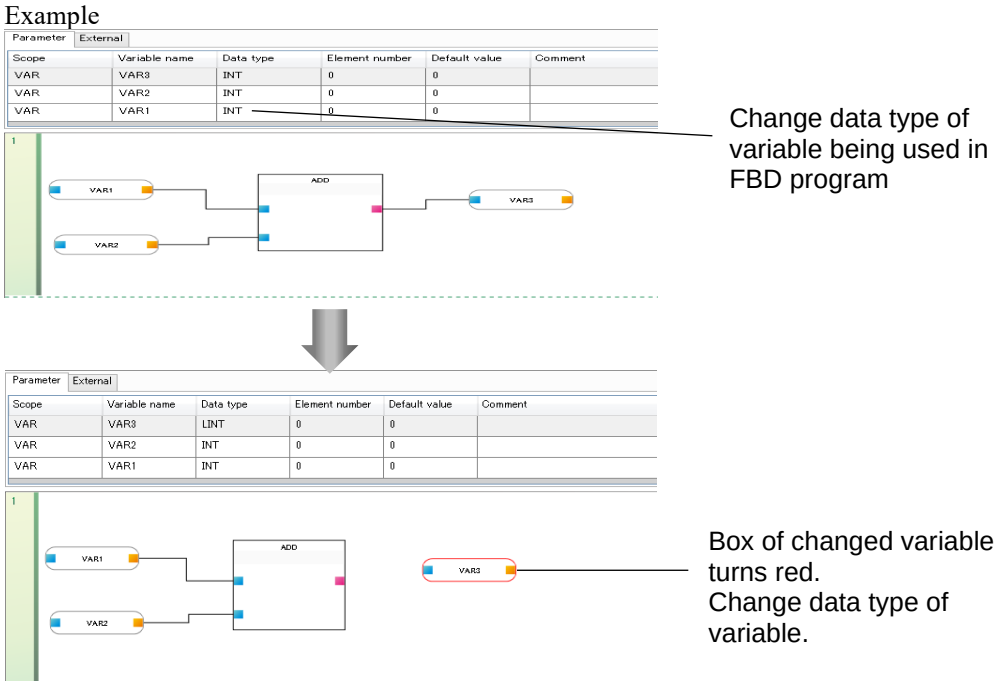
Note that when moving, adding, pasting or changing the size of a box that is close to the bottom boundary of the network, and the position or size of the box protrudes beyond the boundary of the network, then the network size automatically extends out.

9.3.1.10 Change setting for variable, user defined function or system defined function

After inserting a variable, user defined function or system defined function in the program area, if the user performs one of the following operations on that variable, user defined function or system defined function, then the frame of the variable box or function box turns red. In addition, if the user edits a variable, the connector that connects to the variable box is automatically deleted.

- Change variable data type or element number
- Delete variable
- Change properties of user or system defined function (name or return value type)
- Change variable properties of user defined function or system defined function
- Add, delete or change settings of variable (VAR_OUTPUT/VAR_INPUT/VAR_IN_OUT) for user defined function or system defined function

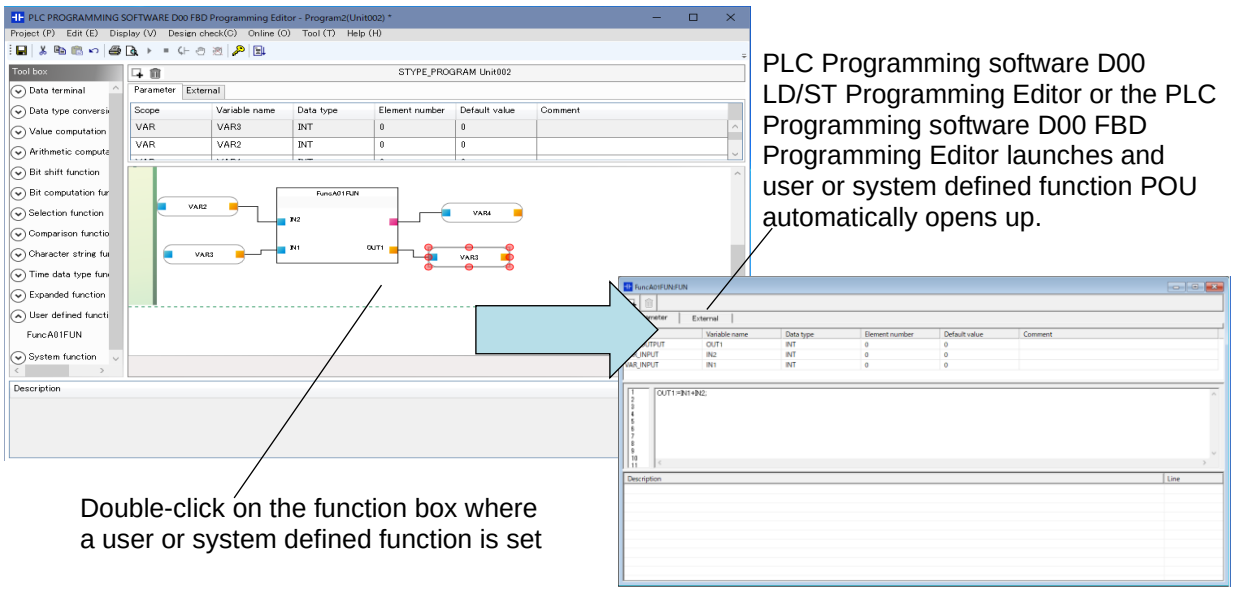
Example



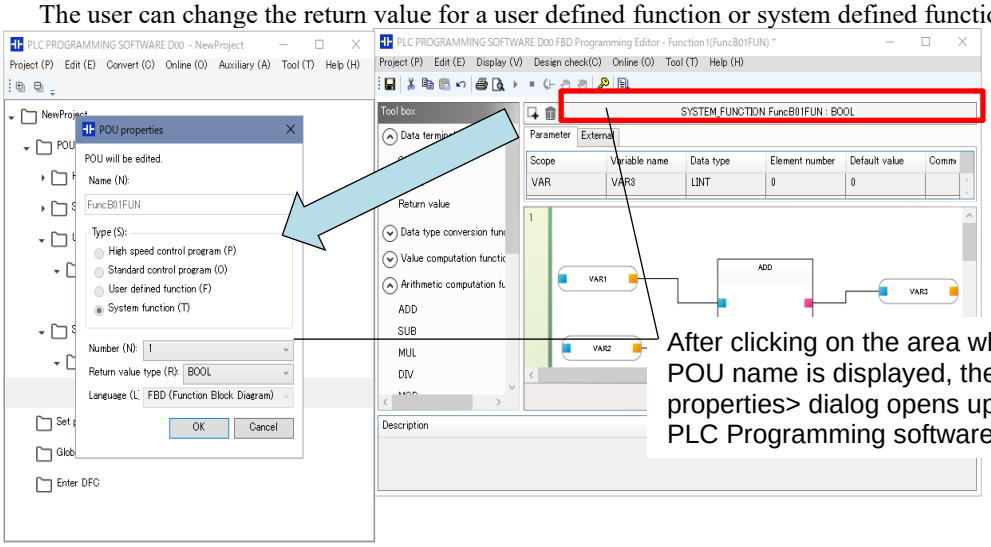
If there is a red box in the FBD program, the conversion or monitoring may not execute correctly. The user must correct the program by deleting the red box from the program area or by revising the variable, user defined function or system defined function. Or, the user can re-assign the box. In addition, refer to “9.3.1.11 Re-assign box” for further details.

(NOTE) If the “variable name” of the variable is changed, that change will automatically be reflected in the variable box.

After double-clicking on a function box (where a user defined function or system defined function) is set), the PLC Programming software D00 LD/ST Programming Editor or the PLC Programming software D00 FBD Programming Editor launches and the user defined function or system defined function POU automatically opens up.
The POU program can be checked and changed.



In addition, when a user or system defined function POU is opened in the PLC Programming software D00 FBD Programming Editor, if the user clicks on the area where the function name is displayed, the <POU properties> dialog displays in the PLC Programming software D00.
The user can change the return value for a user defined function or system defined function.



Refer to “5.2.1 Procedure to Create New POU for details on settings and operations in the <POU properties> dialog.

(NOTE) When the user edits a user or system defined function that is being used in POU in the FBD language, a mismatch may occur with the current program data. In this situation, revise the program, or re-assign the box. Refer to "9.3.1.11 Re-assign box” for further details.

9.3.1.11 Re-assign box

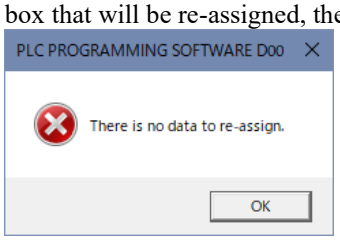
After setting up a variable, user defined function or system defined function box in the program area in the PLC Programming software D00 FBD Programming Editor, if the user makes a change to that variable, user defined function or system defined function, then the box turns red. In addition, the connector connected to that box is deleted.
If there is a red box, conversion is not possible. Therefore, the user must delete the red box and revise the program, or re-assign the box.
When a box is re-assigned, the variable/POU data with the same name as the variable/POU that is set in the current box is reflected in that re-assigned box.

Operation procedure

1. Select one variable box or function box (red box) to re-assign in the PLC Programming software D00 FBD Programming Editor.
2. Select <Re-assign box> from the <Edit> menu.
3. The most recent variable/POU data is reflected in the selected box, and then the box turns black.

(NOTE 1) If a global variable is changed, first update the settings for the external variable registered to the POU, and then re-assign the box.

(NOTE 2) When a box is re-assigned, if there is no variable/POU with the same name as the box that will be re-assigned, then the re-assign operation will not execute.



(NOTE 3) Multiple boxes cannot be re-assigned at the same time. If multiple boxes need to be re-assigned, then re-assign one box at a time.

(NOTE 4) When a box is re-assigned, the deleted connector will not automatically connect. After the box re-assignment, set up and connect the connector again.

Refer to “10.15 Re-assign Boxes (FBD Language)” for details on the box re-assignment.

9.3.2 Undo

This section describes the undo operation when using the PLC Programming software D00 FBD Programming Editor.

The user can perform the undo operation a maximum of 20 times and it applies to the following edit operations.

- Pasting and deleting function boxes
- Adding and deleting networks or changing the size
- Correcting text
- Cutting and pasting
- FBD program conversion
- Editing variable

Operation procedure

The user can perform one of the following operations to carry out an undo operation.

- Select <Undo> from the <Edit> menu
- Press the **[Ctrl]** + **[Z]** keys
- Click on the button in the tool bar

9.3.3 Conversion

Conversion refers to converting the POUs (created in a language in the PLC Programming software D00 EDITOR) to code (intermediate code) that allows execution in the NC(BUILT IN PLC).
Refer to “5.9 Conversion” for further details.

- Write during run
  If the “Write during RUN after conversion” setting is used in the “Conversion mode settings” in the PLC Programming software D00, then write during run operation is executed after a conversion operation is carried out.
  Refer to “5.10.4 Write during run” for details on the write during run operation.

(NOTE) When using the write during run operation, take the necessary safety precautions with the device that has the NC(BUILT IN PLC) and also with the operating environment.

9.4 Copy, Cut, Paste & Delete

This section describes the copy, cut, paste and delete operations for the items below.

- Function box, OM box, literal box, and variable box
- Network

The operation combinations “Copy and paste” and “Cut and paste” in an FBD program can only be done in the same PLC Programming software D00 FBD Programming Editor.

Copy, cut and paste operations cannot be executed for a connector.
However, if the user executes a copy or cut operation while the boxes on both sides of the connector are selected, then the connector and boxes can be pasted together.

When cutting a box, the connector that is connected to that box is deleted. As a result, when pasting, only the box is pasted.

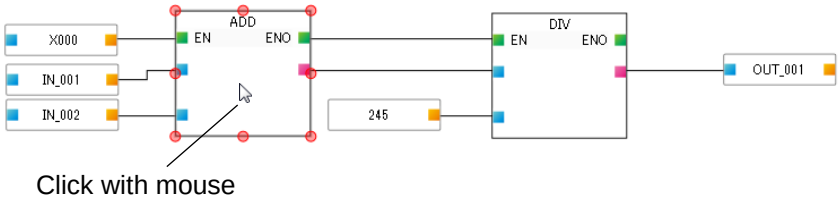
When doing a copy and paste operation in the same network, it does not paste at the copy position but slightly separated away from it.
When doing a cut and paste operation in the same network, it pastes at the same point as the cut position.
When pasting to a different network, it pastes at the same point as the copy and cut position in the original network.
When the network is not set, the paste operation will not execute.

9.4.1 Specify Range

The following shows how to specify the range to perform copy, cut, paste and delete operations.

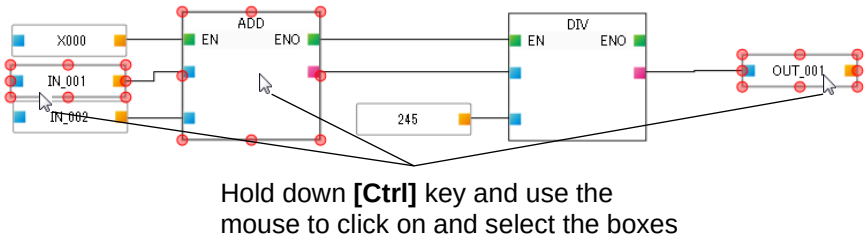
9.4.1.1 When specifying only a function box

Use the mouse to click on the desired function box when only copying, cutting or pasting a single function box.



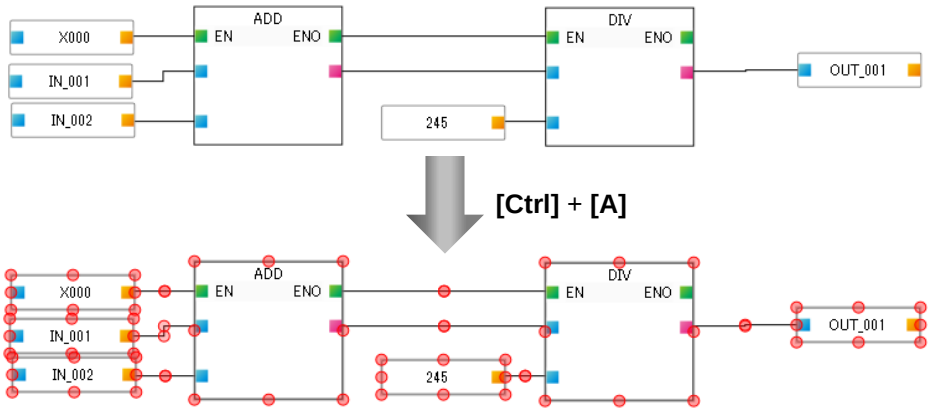
9.4.1.2 When specifying multiple function boxes or connectors

When specifying multiple function boxes to copy, cut and/or paste, hold down the **[Ctrl]** key or **[Shift]** key and use the mouse to click on the desired function boxes.



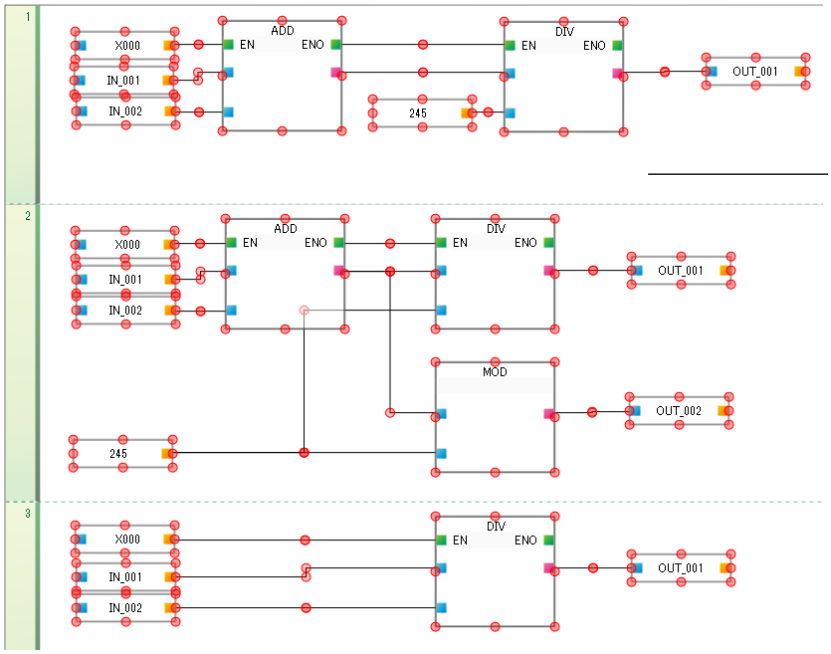
9.4.1.3 When selecting all items in a network

When selecting all the function boxes and connectors in a network, specify the network and then press the keys **[Ctrl]** + **[A]**. Or, the user can select <Control in network> under <Select all> in the <Edit> menu.



9.4.1.4 When selecting networks

The user can click on the network number to select a network.
To select multiple networks, hold down the **[Ctrl]** key and click on the desired network numbers.



When a network is selected, all the controls in that network become selected.

9.4.2 Copy, Cut, Paste & Delete Operations

This section describes how to perform copy, cut, paste and delete operations.

The user can perform one of the following operations to copy.

Operation procedure

One of the following operations can be carried out:

- Select <Copy> from the <Edit> menu.
- Press the **[Ctrl]** + **[C]** keys
- Click on the button in the tool bar
- Right-click the mouse and select <Copy> from the displayed menu.

The user can perform one of the following operations to cut.

Operation procedure

One of the following operations can be carried out:

- Select <Cut> from the <Edit> menu.
- Press the **[Ctrl]** + **[X]** keys
- Click on the button in the tool bar
- Right-click the mouse and select <Cut> from the displayed menu.

The user can perform one of the following operations to paste.

Operation procedure

One of the following operations can be carried out:

- Select <Paste> from the <Edit> menu.
- Press the **[Ctrl]** + **[V]** keys
- Click on the button in the tool bar
- Right-click the mouse and select <Paste> from the displayed menu.

If the user performs a copy and paste operation for the function boxes: input_TO_output, input_TO_BCD or BCD_TO_output, then the “input” and “output” text changes to “???”. After the pasting operation, be sure to re-connect the variable box or OM box.

The user can perform one of the following operations to delete.

Operation procedure

One of the following operations can be carried out:

- Select <Delete> from the <Edit> menu.
- Press the **[Delete]**keys
- Right-click the mouse and select <Delete> from the displayed menu.

9.5 Replace

This section describes how to replace OMs being used in the FBD program.

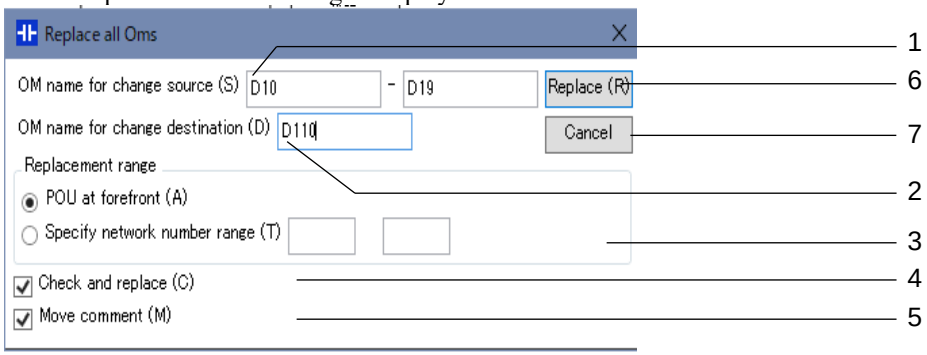
9.5.1 Replace All OMs

This function replaces an OM being used in an FBD program with another OM.
The user can either replace the OMs being used in an FBD program all at once, or check and replace each OM one at a time.
In addition, the user can specify a range in the FBD program and can also transfer the original comment data (that is set to the source OM) when replacing OMs.

(NOTE 1) When replacing OMs, the user can only replace bit OMs with bit OMs and only replace word OMs with word OMs.
(NOTE 2) When "X" is specified for the source OM, "X" can only be set for the target OM.
(NOTE 3) When the user clicks on the <Replace> button and the OM number setting exceeds the upper limit after the replacement, then the error message <<Not possible to replace OM number because new OM number will be out of range.>> is displayed.
(NOTE 4) When a comment is moved or transferred, the OM comment (source) overwrites the target OM comment and then the source OM comment is deleted.
However, if an OM is used at multiple locations and one of those OMs is replaced, then that OM comment is also transferred.
(NOTE 5) After the replacement is completed, the design check is automatically executed.

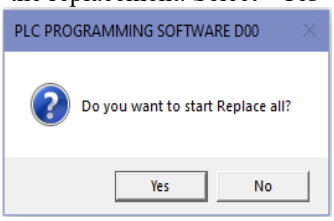
Operation procedure

1. Select <Replace all OMs> under <Replace> in the <Edit> menu.
2. The <Replace all OMs> dialog is displayed.

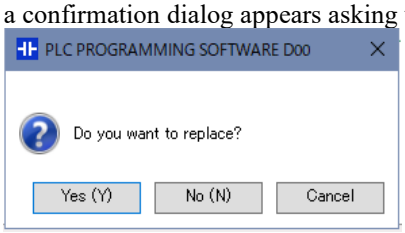


No.	Item	Description
1	OM name for change source	Specifies the first and the last OMs in the source.
2	OM name for change destination	Specifies the first OM in the destination. The range for replacing OMs is specified in the "OM name for change source".
3	Replacement range	Selects and specifies the replacement range. "POU at forefront" : Specifies the replacement range for the POU at the forefront. "Specify network number range" : Specifies the first and last network numbers in the range.
4	Check and replace	Selects whether to replace all OMs at once or to check and replace each one at a time. When the checkbox is ticked, it will check and replace each OM one at a time.
5	Move comment	Selects whether or not to move the comment from the source OM to the destination when replacing the OM. Tick the checkbox to move the OM comment.
6	<Replace> button	Starts the replacement.
7	<Cancel> button	Closes the <Replace all OMs> dialog.

3. After clicking on the <Replace> button, a confirmation dialog appears asking whether to start the replacement. Select <Yes> to execute the replacement operation.



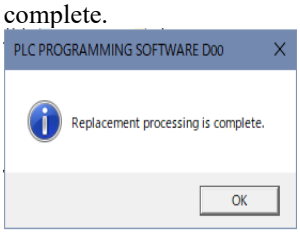
4. When the checkbox <Check and replace> is ticked, the cursor moves to the specified OM and a confirmation dialog appears asking whether or not to replace the OM.



Select <Yes> to replace the OM (at the cursor) and then search for the next OM.
Select <No> to skip replacing the OM (at the cursor) and search for the next OM.
Select <Cancel> to cancel the OM replacement operation.

When the checkbox <Check and replace> is not ticked, the confirmation message does not appear and it replaces the specified OMs all at once.

5. When all the specified OMs have been replaced, a dialog appears indicating the processing is complete.



6. Even if the replacement is complete, it does not close and exit the <Replace all OMs> dialog. The user can continue doing other replacements.
Click on the <Cancel> button to exit replacement and close the <Replace all OMs> dialog.

When all the OMs are replaced at once, the design check is automatically executed. If the program becomes invalid due to the replacement operation, an error is displayed in the message display area. Check the error description and correct the program.

9.6 Search

This section describes how to search for character strings and OMs, and how to jump to a specified network.

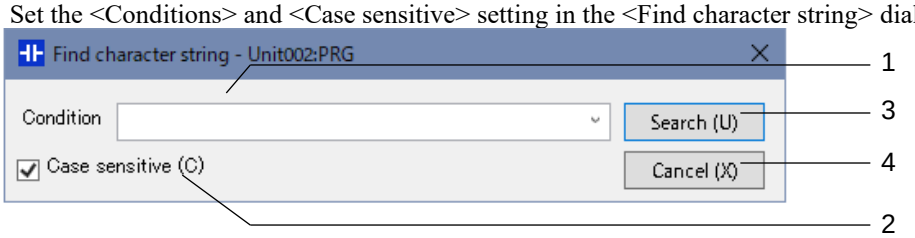
However, this search function is used for POUs that are currently being edited. The other POUs are not searched.

9.6.1 Character String Search

The character string search looks for character strings displayed in each box in an FBD program.
The search starts from the currently selected network and searches for the character string specified in the “Conditions” field.
When a character string is found, the box with the search hit becomes selected.
Each time the <Search> button is pressed, it searches for the next character string.
When the search comes to the end of the FBD program, it goes back to the first network and continues the search up to the original network position.

Operation procedure

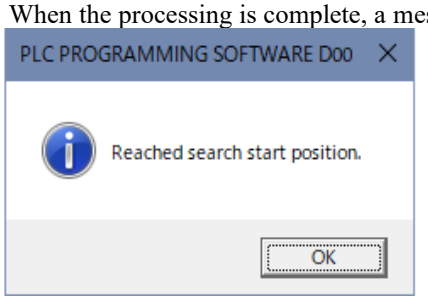
1. Select <Find character string> under <Search> in the <Edit> menu.
2. Set the <Conditions> and <Case sensitive> setting in the <Find character string> dialog.



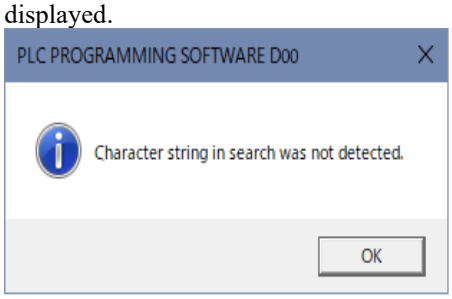
No.	Item	Description
1	Conditions	Specifies the character string in the search. When the user clicks on the <▼> button, a maximum of 32 hits will be displayed search history in the PLC Programming software D00 FBD Programming Editor that is launched.
2	Case sensitive	Selects whether or not to differentiate between uppercase and lowercase letters in the character string search. Tick the checkbox to differentiate between uppercase and lowercase letters.
3	<Search> button	Starts from the currently selected network and searches for a character string specified in the “Conditions” field. When the search reaches the last network, it goes back to the first network and continues the search up to the original search start position.
4	<Cancel> button	Closes the <Find character string> dialog.

3. When the <Search> button is pressed, it searches for the first character string (in the range).
4. Each time the user clicks on <Search> button, it searches for the next character string.
5. Repeat step 4 to continue searching for all the search hits throughout the FBD program.

6. When the processing is complete, a message appears indicating the processing is complete.



In addition, if there are no hits in the character string search, then the following message is displayed.



7. Even if the search is complete, it does not close and exit the <Find character string> dialog.
   The user can continue doing other searches.
   Click on the <Cancel> button to close and exit the search.

9.6.2 Jump

This section describes how to jump to the first network, last network or a specified network in the FBD program.

Operation procedure

1. One of the following operations can be carried out:
   - Select <Jump> from the <Edit> menu.
   - Press the **[Ctrl]** + **[J]** keys
2. Set the jump location in the <Jump> dialog.

   Select <First> to jump to the first network.
   Select <End> to jump to the last network.
   Select <Network number> and enter a network number to jump to a specified network.

3. After clicking on the <Jump> button, it jumps to the specified network.

When specifying a network number, if the specified number exceeds the actual number of networks, then it jumps to the last network.

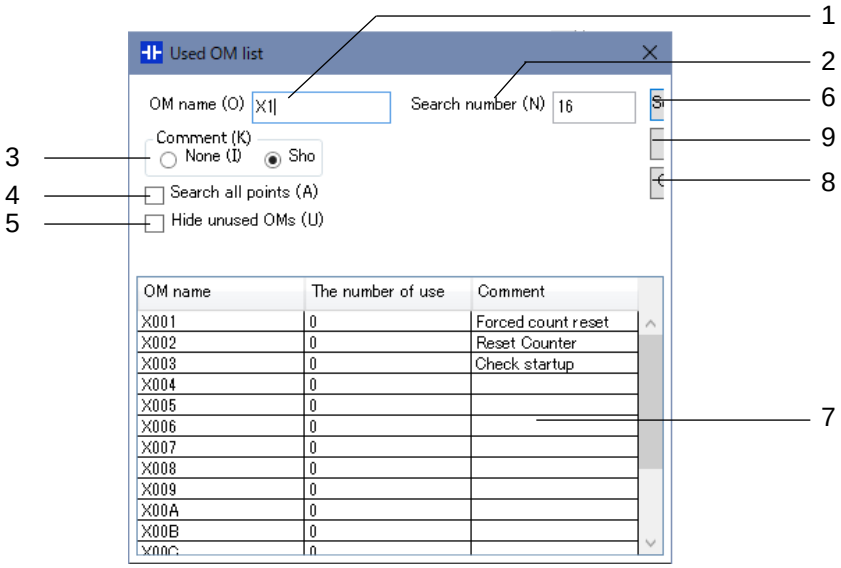
Click on the <Cancel> button in the <Jump> dialog to cancel the jump operation.

9.6.3 OM List

The OM list can be used to conduct a search to check if an OM is being used or not in the FBD program by specifying the search range from a certain OM.

Operation procedure

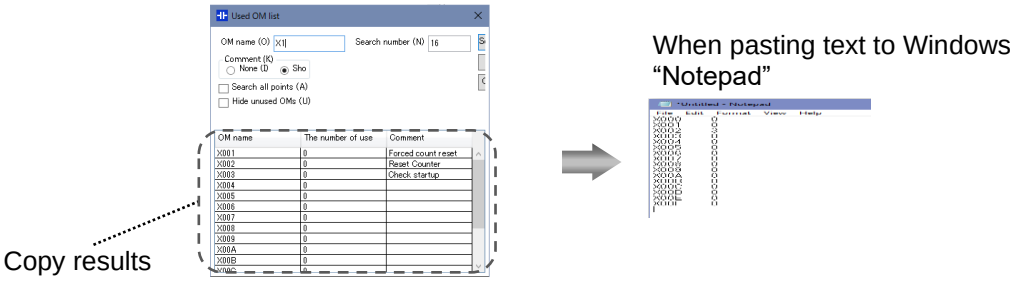
1. One of the following operations can be carried out:
   - Select <Used OM list> under <Search> in the <Edit> menu.
   - Press the **[Shift]** + **[F4]** keys
2. Set the OM and search range in the <Used OM list> dialog.



No.	Item	Description
1	OM name	Specifies the OM in the search.
2	Search number	Specifies the search range from the set OM.
3	Comment	Select <With> to display the OM comments in the search results.
4	Search all points	Tick the checkbox to search all locations from the specified OM.
5	Hide unused OMs	Tick the checkbox to display the search results excluding the unused OMs.
6	<Search> button	Starts the search.
7	List of search results	Displays the search results in a list. <OM name> : Displays the OMs that are found. <Qty. used> : Displays the quantity of the specified OM that is used. <Comment> : Displays the comment that is set to the OM when <Comment> is set to <With>.
8	<Copy> button	Copies the displayed search results.
9	<Cancel> button	Closes the <Used OM list> dialog.

3. Set the range of the search in the <Search number> field, or tick the checkbox <Search all points> to search all the points after the specified OM.
4. Tick the checkbox <Hide unused OMs> to display only the OMs that are used.
5. Click on the <Search> button.
6. Even if the search is complete, it does not close and exit the <Unused OM list> dialog. The user can continue doing other searches.
   Click on the <Cancel> button to close and exit the search.

- Copy
  The search results can be copied as text to the Windows clipboard.
  The user can paste the copied search results to another application like a commercially available text editor software.



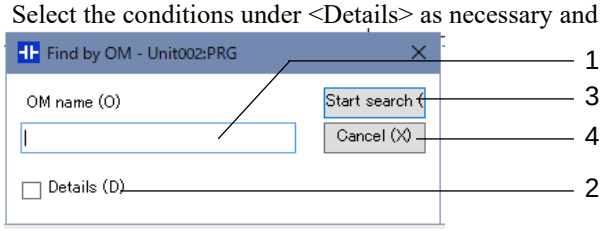
Refer to "10.4.20 OM List (FBD Language)" for details on the "Used OM list".

9.6.4 OM Search

The user can specify an OM and conduct a search to check where it is used.
In an FBD program, the search starts from the currently selected network and looks for the specified OM.
When an OM is found, the box with that OM becomes selected.
Each time the <Search> button is pressed, it searches for the next character string.
When the search comes to the end of the FBD program, it goes back to the first network and continues the search up to the original network position.

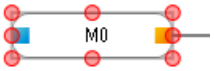
<u>Operation procedure</u>

1. One of the following operations can be carried out:
   - Select <Find by OM> under <Search> in the <Edit> menu.
   - Press the **[Ctrl]** + **[Shift]**+ **[D]** keys
2. Set the OM and search range in the <Find by OM> dialog.
   Select the conditions under <Details> as necessary and click on the <Start search> button.



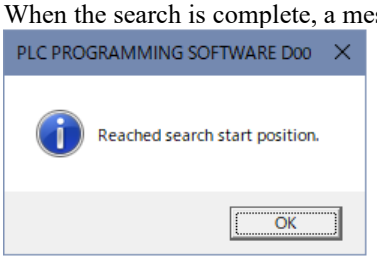
No.	Item	Description
1	OM name	Specifies the OM in the search.
2	Details	When a bit OM word is specified, tick the checkbox to search for OMs that are not displayed in the FBD program.
3	<Start search> button	Starts the search.
4	<Cancel> button	Closes the <Find by OM> dialog.

3. When an OM is found, that OM box becomes selected.

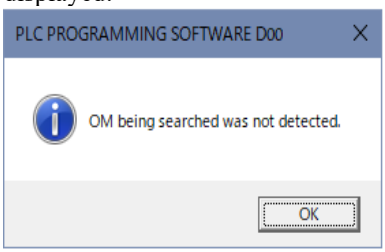


4. When the user clicks on <Start search> button again, it searches for the next location where the OM is used.
5. Repeat step 4 to continue searching for all the hits throughout the FBD program.

6. When the search is complete, a message appears indicating the search is finished.



In addition, if there are no hits in the character string search, then the following message is displayed.



7. Even if the search is complete, it does not close and exit the <Find by OM> dialog. The user can continue doing other searches.
Click on the <Cancel> button to close and exit the search.

9.7 Edit Lock

This section describes the operations to prohibit or lock editing in the FBD program in order to protect the program from an accidental or inadvertent input operation.

The edit lock is a function to prevent an inadvertent operation from writing data to the FBD program and specifically prohibits the following edit operations.
- Edit FBD program
- Edit variable

The user can unlock the editing by performing the edit lock operation again (while the edit lock is on), allowing the user to edit the data.

Even if the PLC Programming software D00 FBD Programming Editor is closed while the edit lock is on, the edit lock will be unlocked the next time the same POU is opened in the PLC Programming software D00 FBD Programming Editor.

The “Edit lock/edit unlock” can be performed using one of the following operations.

Operation procedure
One of the following operations can be carried out:
- Select <Edit lock> from the <Edit> menu.
- Press the **[F11]** key.
- Click on the button in the tool bar

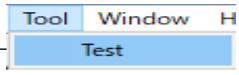
9.8 Launch Simulation

Before running an FBD program on the machine, the user can check the operation of the FBD program in the PLC Programming software D00 EDITOR.

The following is the procedure for running a simulation of an FBD program that was created.

Operation procedure
1. Select <Simulation> from the <Tool> menu.
   After the simulation is launched, a checkmark “✓” appears before the item name <Simulation> in the menu.

   Checkmark “✓” appears.



2. Write the FBD program to the simulation.
   Refer to the “5.10.2 Write programs” for details about writing FBD programs.

- Exit simulation
  When there is a checkmark “✓” next to the <Simulation> item in the <Tool> menu, select <Simulation> again (to deselect and exit).
  After exiting the simulation, the checkmark “✓” next to the <Simulation> disappears.

(NOTE 1) The user can launch and exit the simulation from the PLC Programming software D00 and the PLC Programming software D00 LD/ST Programming Editor.
When running an FBD program in the simulation, there is no need to re-launch the PLC Programming software D00 LD/ST Programming Editor if the simulation has already been launched with another tool.

(NOTE 2) The POUs set in the <Program settings> dialog are written to the program. Register the FBD program in the <Program settings> dialog to run the FBD program currently being edited in the simulation.
Refer to “5.8 Program Execution Settings ” for details on the <Program settings> dialog.

9.9 Monitoring FBD Program in NC(BUILT IN PLC)

The user can perform the following operations to execute FBD program monitoring in the NC(BUILT IN PLC).

Operation procedure
Perform one of the operations below in the PLC Programming software D00 FBD Programming Editor.
- Press the **[F2]** key
- Select <Start monitoring> under the <Online> menu

Refer to “5.12 NC(BUILT IN PLC) Monitoring ” for further details on monitoring.
In addition, refer to “9.11 Monitoring operations” and “9.12 DEBUG Function” for details on FBD program operations when monitoring.

(NOTE) When an FBD program is converted, an ST program is automatically created based on the same processes as the FBD program and then it is displayed in PLC Programming software D00 LD/ST Programming Editor.
When monitoring in the PLC Programming software D00 FBD Programming Editor, display the ST program at the forefront in the PLC Programming software D00 LD/ST Programming Editor.

9.10 Run & Stop in NC(BUILT IN PLC)

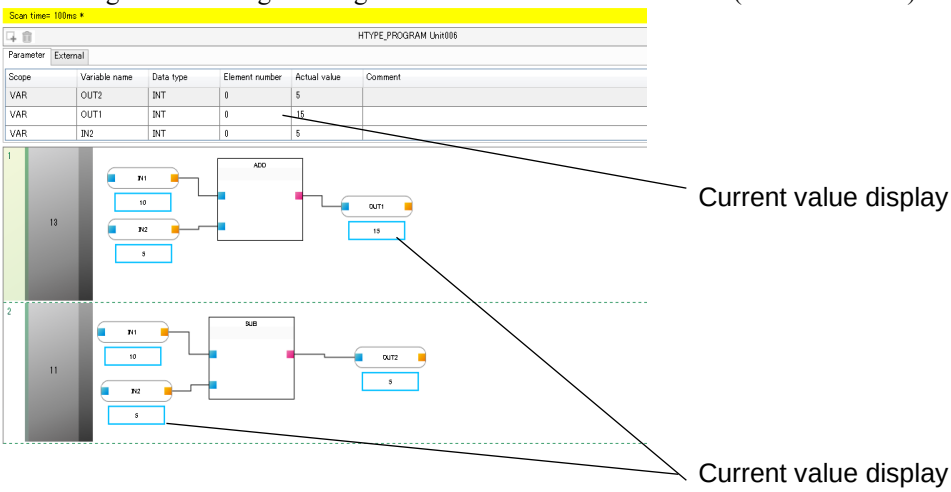
Refer to “5.13 Run & Stop in NC(BUILT IN PLC)” for details on running and stopping in the NC(BUILT IN PLC).

9.11 Monitoring Operations

This section describes the operations for each function when monitoring an FBD program in the NC(BUILT IN PLC).

(NOTE) When there is a checkmark "✓" next to the <Simulation> in the <Tool> menu (for each tool), then the program is written to the simulation.

The user can check the operation status and the current value of the OMs in the FBD program by connecting the PLC Programming software D00 EDITOR to the NC(BUILT IN PLC).



In the FBD program, the current value is displayed for the connectors between OMs, variables and function boxes.
The values 0 and 1 or FALSE and TRUE are displayed for BOOL type variables, following the same format as the default value in the variable settings.

The user can change the base number when displaying the current value of the word OM.
Refer to "9.11.1 Change base number for monitoring values" for details on changing the base number.

While monitoring is in progress, the "Default value" field in the variable registration area changes to the "Current value", and the current value for each variable is displayed.

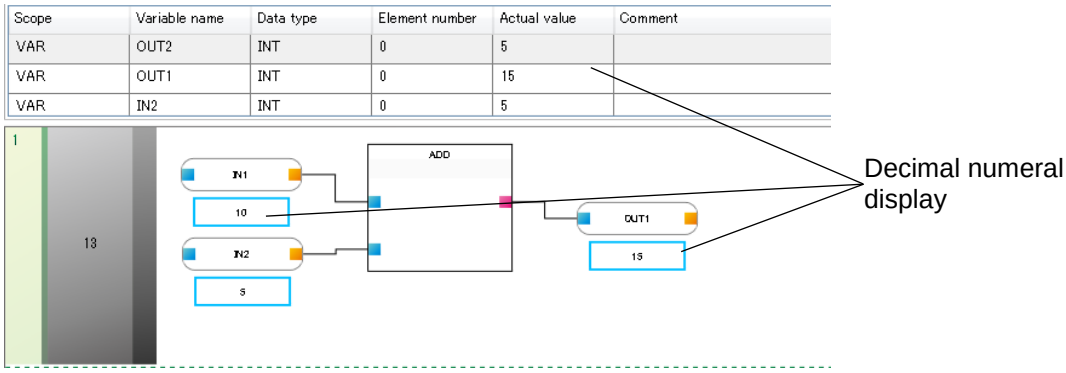
The user can force a change in the current value for the variable and OM while monitoring.
In addition, the following functions can be used to change the OM status.

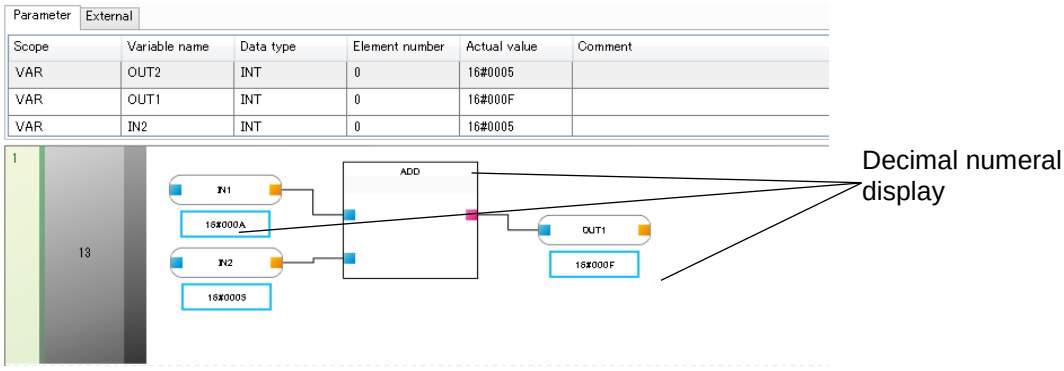
- Enter directly into the current value display field under the OM box
- To force the bit OM to change to ON or OFF :OM SET/RST
- To force a change in the current value of the word OM :Change current OM value

Refer to "9.11.3 SET/RST OM & change current OM value" for setting or resetting the OMs and changing their current value.
In addition, the user can press the **[ESC]** key to only stop the monitoring and keep executing the NC(BUILT IN PLC).
To re-start the monitoring, press the **[F2]** key. The user can also scroll through the program edit window.

9.11.1 Change Base Number for Monitoring Values

In the FBD program monitoring, the user can change the current values displayed during monitoring to “Decimals” or “Hexadecimals” for data type variables that can be expressed in hexadecimals.





(NOTE) The user cannot change the current OM value display when the data type variable has a value that cannot be expressed as a hexadecimal.

<u>Operation procedure</u>

One of the following operations can be carried out:

- Select <Change monitoring base number> under <Change display> in the <Display> menu.
- Press the **[Shift]** + **[F8]** keys

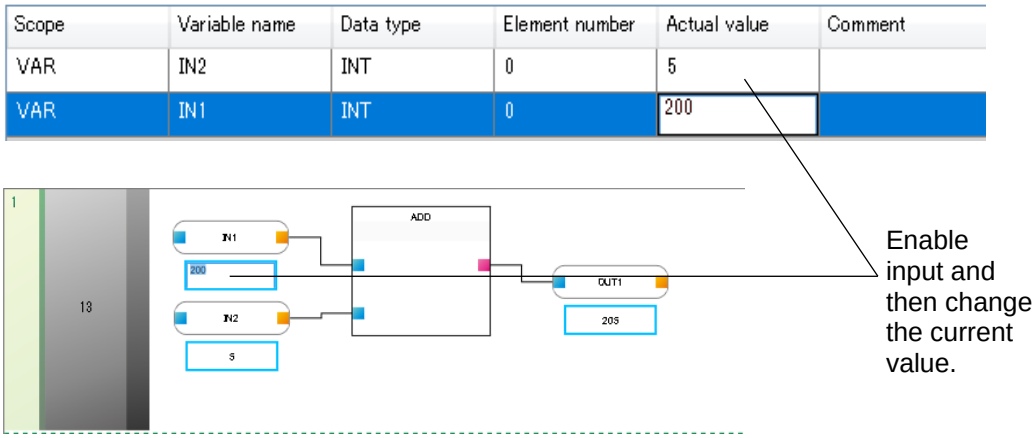
9.11.2 Monitoring & Changing Current Variable Values

9.11.2.1 Monitoring parameter variables and external variables

The current values of the variables are displayed in the variable editing area and in the FBD program.

The user can enter a numerical value or character string into the “Current value” field in the variable editing area in order to change the current value of a variable. The user can also press the **[Space]** key or use the mouse to click on the “Current value” field in order to enter data.

When changing a current value in the FBD program, double-click on the current value box under the function box to enable input. Enter a current value for the variable.



The values (base number) that can be entered for the variable vary depending on the data type.

Data type	Input range/base number
BOOL	0, 1, FALSE, TRUE
BYTE	2#00000000~2#11111111 8#000~8#377 0~255 16#00~16#FF
WORD	2#0000000000000000~2#1111111111111111 8#000000~8#177777 -32768~32767 16#0000~16#FFFF
DWORD	2#00000000000000000000000000000000~ 2#11111111111111111111111111111111 8#00000000000~8#37777777777 -2147483648~2147483647 16#00000000~16#FFFFFFFF
INT	-32768~32767 16#0000~16#FFFF
DINT	-2147483648~2147483647 16#00000000~16#FFFFFFFF
LINT	-9223372036854775808~9223372036854775807 16#0000000000000000~16#FFFFFFFF FFFFFFFF
UINT	0~65535 16#0000~16#FFFF
UDINT	0~4294967295 16#00000000~16#FFFFFFFF
ULINT	0~18446744073709551615 16#0000000000000000~16#FFFFFFFF FFFFFFFF
REAL	-3.402823E+38~-1.401298E-38, 0.0, 1.401298E-38~3.402823E+38 (single-precision real number)
LREAL	-1.79769313486231E+308~-4.94065645841247E-324, 0.0, 4.94065645841247E-324~1.79769313486231E+308 (double-precision real number)
TIME	T#000:00:00~T#255:59:59

Data type	Input range/base number
DATE	D#1970-01-01~D#2037-12-31
TOD	TOD#00:00:00~TOD#23:59:59
DT	DT#1970-01-01-00:00:00~DT#2037-12-31-23:59:59
STRING	0 to 31 characters (half-width alphanumeric characters and symbols only)

(NOTE 1) When changing variable values online, be sure to take the necessary safety precautions. If an invalid value is set, the device that has the NC(BUILT IN PLC) installed could malfunction.

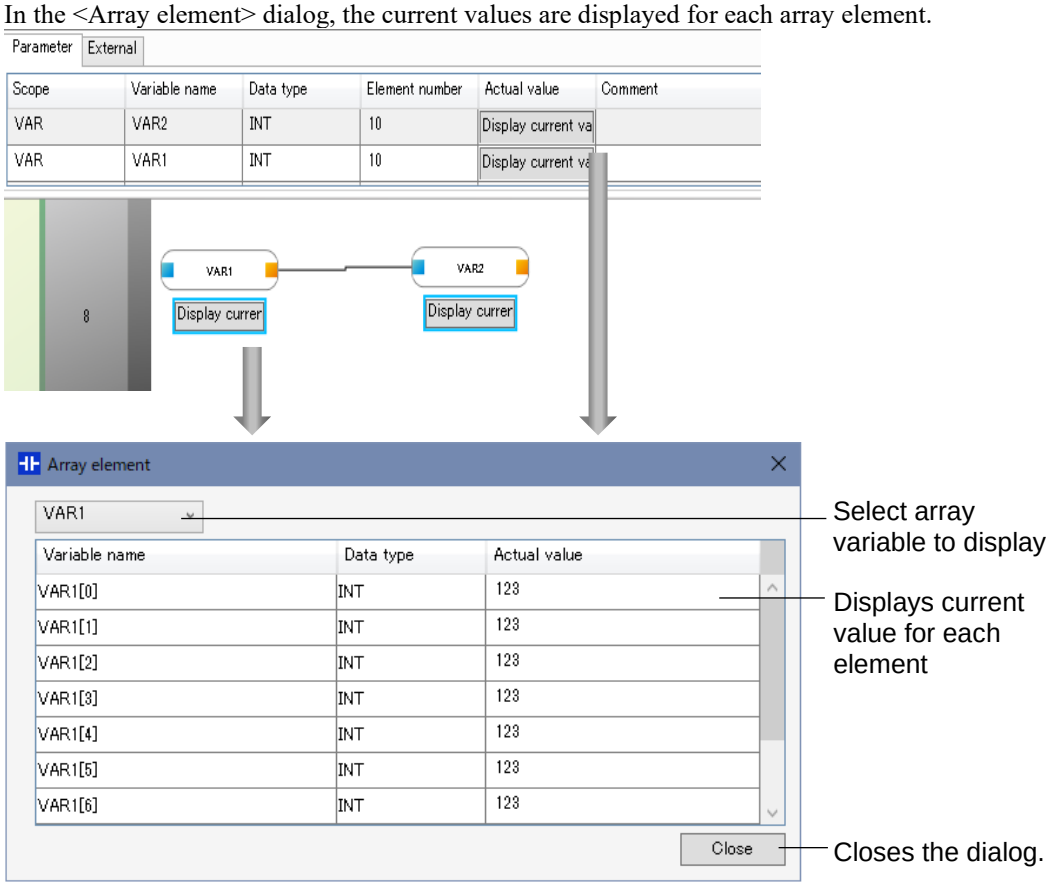
(NOTE 2) When a current value is displayed for a REAL or LREAL type variable, then the set value is rounded off to the valid digits of that type (7 digits and 15 digits, respectively). As a result, the actual value and displayed value may not match. Examples are shown below.
In addition, even when the current value is changed for a REAL or LREAL type variable, the set value is rounded off to the valid digits of that type (7 digits and 15 digits, respectively).

Data type	Current value	Displayed value
REAL type	1234567	1234567
	12345678	1.234568E+007
	0.0001234567	0.0001234567
	0.00012345678	0.0001234568
LREAL	123456789012345	123456789012345
	1234567890123456	1.23456789012346E+015
	0.000123456789012345	0.000123456789012345
	0.0001234567890123456	0.000123456789012346

(NOTE 3) Monitoring is not possible for the current value in the variable box displayed in red. Refer to “9.3.1.10 Change setting for variable, user defined function or system defined function” and then modify the program.

9.11.2.2 Array variable monitoring

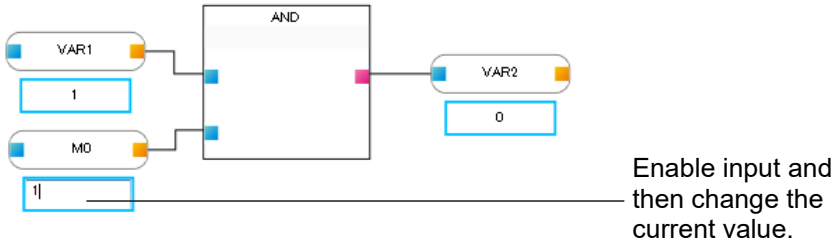
For array variables, the <Display current value> button is displayed in the current value field for the variable editing area.
After clicking on the <Display current value> button, the <Array element> dialog is displayed.
In the <Array element> dialog, the current values are displayed for each array element.



The current value of a variable can be changed by entering a numerical value or character string into the <Current value> field in the <Array element> dialog. The procedure to change the current value is the same as in the variable editing area. Refer to “9.11.2.1 Monitoring parameter variables and external variables”.

9.11.3 SET/RST OM & Change Current OM Value

To change the current OM value, double-click on the current value box under the OM box. This allows the user to enter the desired current value.



In addition, when an FBD program is converted, an ST program is automatically created based on the same processes as the FBD program and then it is displayed in PLC Programming software D00 LD/ST Programming Editor.
When the user changes the mode to monitoring mode in the PLC Programming software D00 FBD Programming Editor, the ST program also automatically changes to monitoring mode in the PLC Programming software D00 LD/ST Programming Editor too.

Use the <SET/RESET> dialog or the <Change current value> dialog to change the value for the OM from the PLC Programming software D00 LD/ST Programming Editor. Refer to “8.11.4 SET/RST OM” and “8.11.5 Change current value” for details on the change procedure.

(NOTE) When changing OM values online, be sure to take the necessary safety precautions. If an invalid value is set, the device that has the NC(BUILT IN PLC) installed could malfunction.

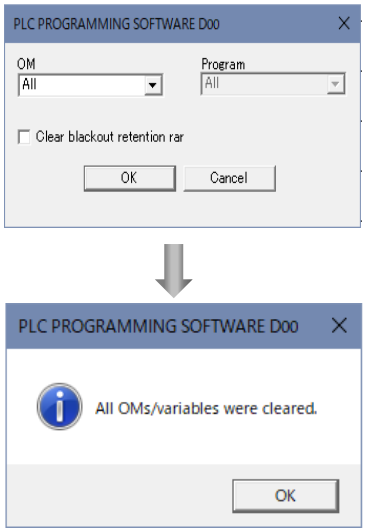
9.11.4 Clear All OMs & Variables

This function clears all the values and status data for the OMs and variables at once (except for the following OMs: X, Y, BX, BY, BDX, BDY, SM and SD). In addition, the user can also reset all the variables to their default values at once.
The set values for the local timer and the local counter cannot be cleared.
The OM/variable all clear operation can only be executed if the NC(BUILT IN PLC) is in a stopped state.

(NOTE) The OM/variable all clear operation applies to all OMs except for the following: X, Y, BX, BY, BDX, BDY, SM and SD. However, when the FBD program is running, be aware that the status of Y, BY, BDY, SM and SD will change based on changes in the conditions.

The following shows the procedure for the OM/variable all clear operation.
Operation procedure

1. Change the NC(BUILT IN PLC) to a STOP state.
   Refer to “5.13 Run & Stop in NC(BUILT IN PLC)” for details on changing it to a STOP state.
2. Use one of the following operations to “Clear all OMs/variables”.
   - Select <Clear all OMs/variables> from the <Online> menu.
   - Press the **[Ctrl]** + **[F9]** keys
3. In the following dialog, select the <OM>, <Program> and <Clear power failure hold area> as necessary, and then click on the <OK> button.
   Refer to “7.12.6 Clear all OMs & variables ” for details about the dialog operations.



9.12 DEBUG Function

This section describes the “DEBUG function” used for checking the FBD program operation.
The user can perform the following FBD program check operations with the “DEBUG function”.

- Step execution
  Executes only one command in the FBD program and then stops.

- FBD program stop based on breakpoint setting
  FBD program stops at specified breakpoint.

- Monitoring stop when OM value changes
  Monitoring is stopped when the specified OM value changes.

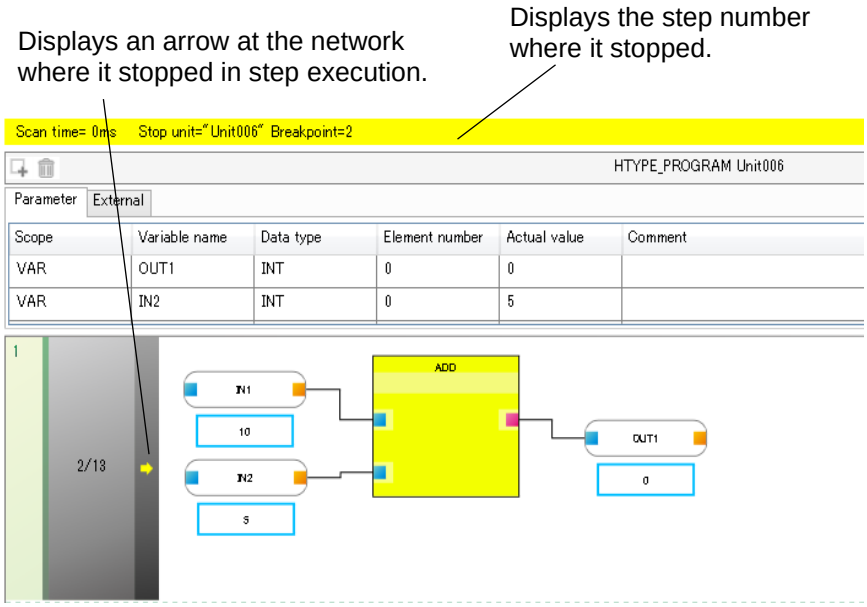
(NOTE) Do not execute the DEBUG function on user or system defined functions.

9.12.1 Step Execution

This section describes step execution operations.
The user can execute the program in steps when the FBD program is in a stopped state.

(NOTE) Step execution is not possible when the FBD program monitoring is not executed or when monitoring is stopped.

When the <RUN/STOP mode> dialog is used to stop the FBD program, it executes from the beginning of the FBD program.
When the FBD program is stopped based on the conditions that are set in “9.12.2 Setting and execution of stop position based on breakpoint setting”, then it executes from the stop position.



The following shows the step execution procedure.

Operation procedure
Start FBD program monitoring.

Stop the FBD program.
Refer to “5.13 Run & Stop in NC(BUILT IN PLC) ” for details on the stop operations using the <RUN/STOP mode> dialog.
Refer to “9.12.2 Setting and execution of stop position based on breakpoint setting” for details on stopping the FBD program based on the condition settings.

Use one of the following operations to perform “Step execution”.
- Select <Execute step> under <DEBUG> in the <Online> menu
- Click on the button in the tool bar

Repeat step 3 to execute the FBD program per command.

Change the setting in the <RUN/STOP mode> dialog to RUN or DEBUG-RUN in order to cancel the step execution.

9.12.2 Setting and Execution of Stop Position Based on Breakpoint Setting

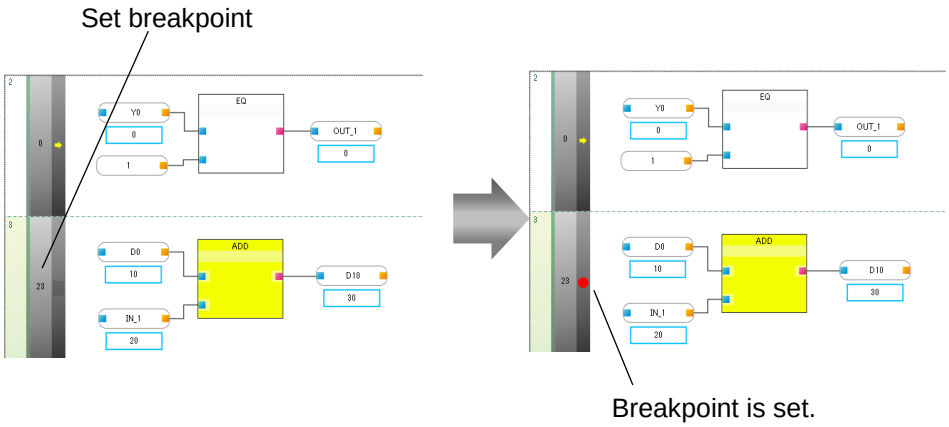
The section describes operations when stopping the FBD program at a specific line.
The user can set breakpoints when the FBD program is in a stopped state.

(NOTE) The FBD program cannot be stopped using breakpoints when the FBD program monitoring is not executed or when monitoring is stopped.

9.12.2.1 Breakpoint settings

Breakpoints can be set as shown below.

Specify the desired network to set a stop, and then perform one of the following operations.
- Select <Set/Cancel breakpoint> under <DEBUG> in the <Online> menu
- Click on the button in the tool bar



The user can perform one of the following operations to remove the stop setting at a step number.
To cancel only a specific setting:
Select <Set/Cancel breakpoint> under <DEBUG> in the <Online> menu
To cancel all settings:
Select <Clear all breakpoints> under <DEBUG> in the <Online> menu

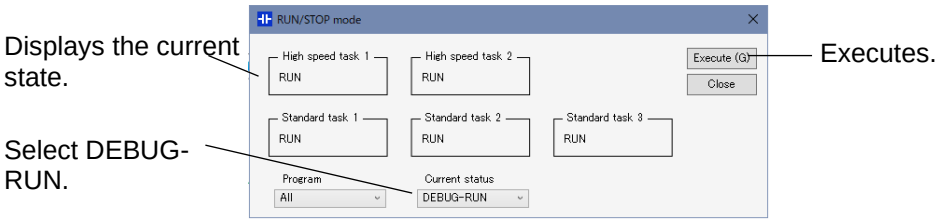
9.12.2.2 FBD program debug execution

Use one of the following operations to debug an FBD program.
The FBD program stops at the line or lines where a breakpoint is set.

Execution using <RUN/STOP mode> dialog

Operation procedure

1. Perform one of the following operations to display the <RUN/STOP mode> dialog.
   - Select <RUN/STOP mode> under the <Online> menu.
   - Press the **[Shift]** + **[F7]** keys
2. Select <DEBUG-RUN>, and then press the <Execute> button.



Execution without using <RUN/STOP mode> dialog

Operation procedure

Perform one of the following operations to execute debugging when the NC(BUILT IN PLC) or simulation is in a stopped state.

- Select <Execute> under <DEBUG> in the <Online> menu
- Click on the button in the tool bar

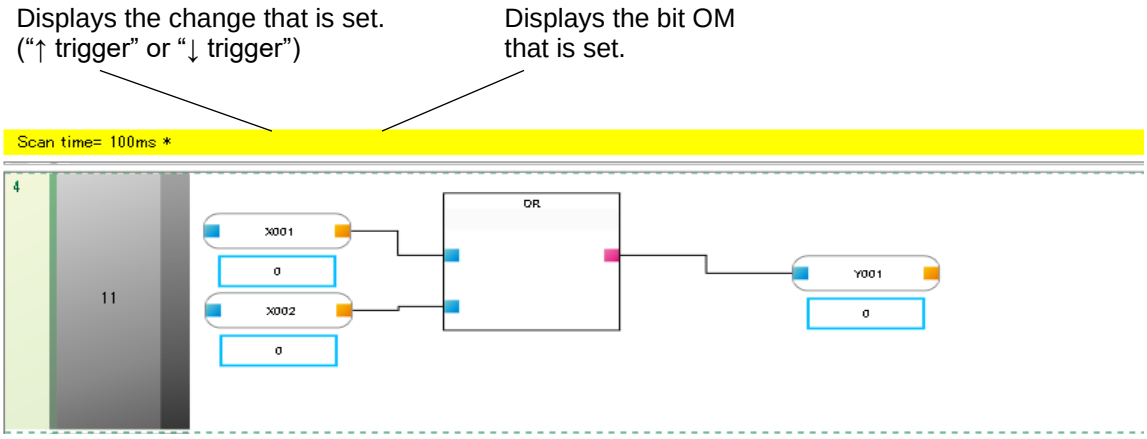
Perform the same operation again in order to skip to the next breakpoint line.
Perform step execution to only execute the next command (after the stop position) and then stop.

To cancel the “DEBUG-RUN” state, change the setting in the <RUN/STOP mode> dialog to <RUN>.
All the stop conditions are cancelled, and the FBD program executes.

9

9.12.3 Stop Monitoring Using OM Change Setting

The user can stop the program monitoring using a trigger setting when the bit OM value changes.



Program monitoring is stopped when X000 turns ON.

Stop program monitoring when OFF changes to ON (Rising edge trigger)

Operation procedure

Select an OM box to set a monitoring stop trigger (during the monitoring execution) at the bit OM. Press the **[Ctrl]** + **[F7]** keys while the monitoring is executing. Or, select <Rising edge trigger> under <Set monitoring pause trigger> in the <Online> menu.

Stop program monitoring when ON changes to OFF (Falling edge trigger)

Operation procedure

Select an OM box to set a monitoring stop trigger (during the monitoring execution) at the bit OM. Press the **[Ctrl]** + **[F8]** keys while the monitoring is executing. Or, select <Falling edge trigger> under <Set monitoring pause trigger> in the <Online> menu.

When a bit OM used to specify digits is selected, the first OM is targeted for monitoring. For example, when an OM box with a “M16:8” setting is selected, the M16 value change is monitored. When a bit OM used for index addressing is selected, OMs that are not indexed are targeted for monitoring. For example, when an OM box with a “M32I0” setting is selected, the M32 value change is monitored regardless of IX0 index value.

While performing monitoring in the PLC Programming software D00 EDITOR, only one bit OM can be monitored at a time regardless of the language or POU. After setting one bit OM for monitoring, if the user monitors another POU and sets another bit OM as the bit OM for monitoring, then the last bit OM that was set is enabled for monitoring.

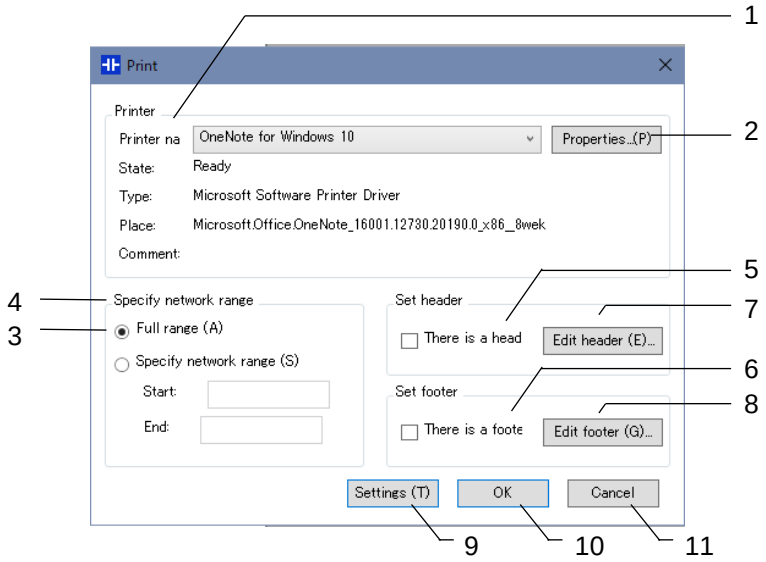
Press the **[F2]** key to cancel the monitoring stop setting (using the bit OM).
In addition, when monitoring is opened again, the bit OM setting for monitoring is cleared.

9.13 Print

This section describes the operations to print an FBD program.
In the <Print> dialog, the user can set the printer, the printer properties (paper size, orientation, etc.) as well as set the printing range in the FBD program when printing. The entire variable area is printed.
The data that is printed out is the POU that is currently being edited in the PLC Programming software D00 FBD Programming Editor.

Operation procedure

1. Select <Print> under the <Project> menu.
2. In the <Print> dialog, set the printer, the printer properties (paper size, orientation, etc.) and the printing range in the FBD program.



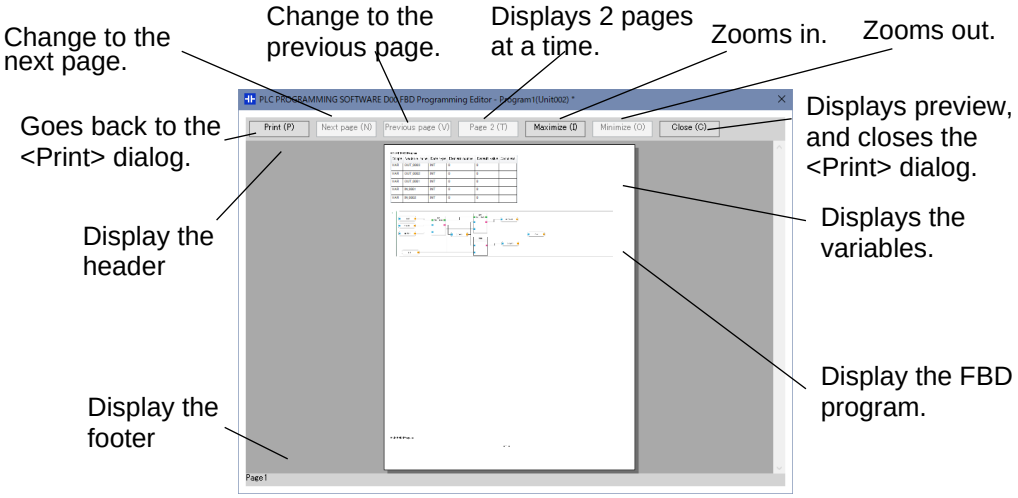
No.	Item	Description
1	Printer name	Selects a printer.
2	<Properties> button	Opens the properties dialog for the printer. (The settings depend on the printer drivers that are installed.)
3	All	Prints the entire range.
4	Specify network range	Prints the range that is set from the first network number to the last network number.
5	Header	Tick the checkbox to include the header.
6	Footer	Tick the checkbox to include the footer.
7	Edit header	Opens the header edit dialog.
8	Edit footer	Opens the footer edit dialog.
9	Set	Saves the header and/or footer information.
10	<OK> button	Saves the header and/or footer information and starts printing.
11	<Close> button	Closes the dialog.

When printing the entire FBD program, select “All” for the range.
To set the network range to print, select “Specify network range” and enter the “Start” and “End” of the range.

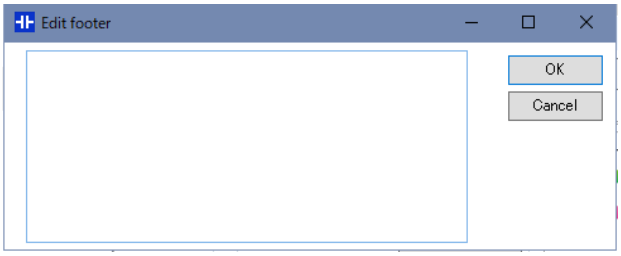
When printing the FBD program, set the paper size to A4 and set the orientation to portrait in the printer properties.

3. Set the “Header settings” and/or “Footer settings” when necessary.

4. Click on the <Yes> button in the <Print> dialog to start printing.
   In addition, the user can use “Print preview” to check the image being printed on the paper.
   Select <Print preview> in the <Project> menu to use print preview.



(NOTE) The header and footer settings are configured in the dialog below. The user can enter a maximum of 512 characters. However, any text after 5 lines is ignored when printing. Therefore, do not exceed the line limit when setting. When the user clicks on the <Set> button or the <OK> button in the <Print> dialog, the editing is saved as part of the program files.



When <OK> is selected, the editing is set and it goes back to the <Print> dialog.
When <Cancel> is selected, the editing is discarded and it goes back to the <Print> dialog.

9.14 Display Manuals

This section describes how to display a manual.
The user can display the different manuals in the PLC Programming software D00.

To display the manuals, the user must install an application software beforehand that can display PDF files, such as Acrobat Reader provided by Adobe.

- User's manual
  Operation procedure
  Select the <User's manual> under <Manual> in the <Help> menu.

- Operation manual
  Operation procedure
  Select the <Operation manual> under <Manual> in the <Help> menu.

- Programming manual
  Operation procedure
  Select the <Programming manual> under <Manual> in the <Help> menu.

(This page was intentionally left blank.)

CHAPTER 10

FUNCTIONS FOR CREATING PROGRAM

10.1 Undo
10.2 Copy, Cut & Paste
10.3 Delete
10.4 Search & Jump
10.5 Replace
10.6 Jump
10.7 Create Comments & Drawings
10.8 Edit Lock
10.9 Program Conversion Function
10.10 Program Check Functions
10.11 Display Function
10.12 Print Function
10.13 Verify Projects
10.14 B00/C00 Project Conversion
10.15 Re-assign Boxes (FBD Language)
10.16 Import & Export

10.1 Undo

This function is the undo operation in PLC Programming software D00 EDITOR.
The undo function can be used on the following operations.
However, when editing a global variable, the user operation history is cleared.

Tool / Language	Applicable operations	Maximum execution times
Common: When editing a variable	• Edit cells • Cut or paste • Insert lines • Delete lines	The “Undo” operation can be executed a maximum of 20 times in one editing window.
PLC Programming software D00 LD/ST Programming Editor When editing an LD program	• Cut or paste • Write, insert or delete circuits • Replace (all *a/b* contacts or all OMs) • Enter, delete or edit inline comments • Enter, delete or edit coil remarks • Enter, delete or edit OM comments • Enter, delete or edit label comments • OM number increase (increment) or decrease (decrement) • Design check • Convert	The “Undo” operation can be executed a maximum of 20 times in one editing window.
	When editing a program in the CALC command edit dialog • Enter and delete characters • Line break • Paste • Cut	The “Undo” operation can be executed a maximum of 20 times until the dialog is closed.
PLC Programming software D00 LD/ST Programming Editor When editing an ST program	• Enter and delete characters • Line break • Paste • Cut	The “Undo” operation can be executed a maximum of 20 times in one editing window.
PLC Programming software D00 LD/ST Programming Editor When editing an FBD program	• Edit operations on network, function box, connector, variable box, literal box and OM box • Enable/Disable EN/ENO	The “Undo” operation can be executed a maximum of 20 times in one editing window.

Refer to “Chapter 7 LD edit operation procedure”, “Chapter 8 ST edit operation procedure” or “Chapter 9 FBD edit operation procedure” for details on how to use the PLC Programming software D00 EDITOR.

10.2 Copy, Cut & Paste

In the PLC Programming software D00 EDITOR, the user can perform copy, cut and paste operations with POUs, variables and programs.
The following items can be copied, cut and pasted.

Tool / Language	Applicable operations
Common Variable editing area	• Variable • Character string in a cell
PLC Programming software D00	POU *1
PLC Programming software D00 LD/ST Programming Editor When editing an LD program	• Circuit symbol • Circuit line • Circuit block
	When editing a program in the CALC command edit dialog • Character string
PLC Programming software D00 LD/ST Programming Editor When editing an ST program	• Character string
PLC Programming software D00 LD/ST Programming Editor When editing an FBD program	• Function box, variable box, literal box, OM box and return value box • Network

*1 Only copying and pasting are possible for the same type of POUs

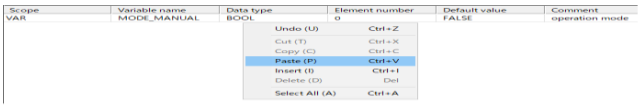
10.2.1 Copy, Cut & Paste Variables (LD, ST & FBD Languages)

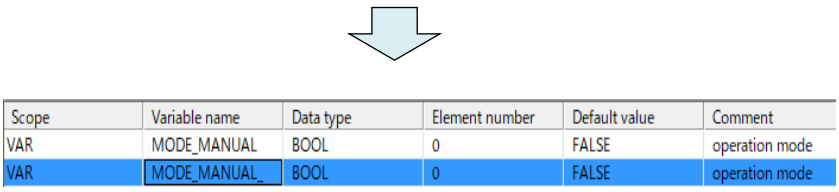
When editing a CALC command in ST, FBD and LD programs, the user can copy, cut and paste the following in the variable editing area.

- Variables (by line)
- Character string in a cell (only for cells where character strings are input)

The user can copy and cut variables and paste them to other POUs.

<u>Copy, cut and paste variables (by line)</u>
The user can copy, cut and paste a variable selected in the variable editing area.
Ex: The variable “MODE_MANUAL” is copied and pasted.





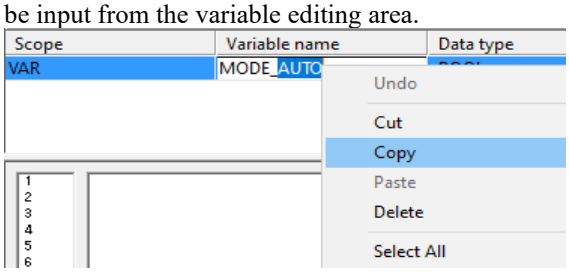
When pasting a variable that was copied or cut, the same settings (scope, data type, element number, default value and comment) are pasted with the variable.

After a variable is pasted, the variable name automatically changes when the same name already exists.

No.	Operation	Example
1	An underscore (_) is added to the end of the item name (copy source).	When the variable “Value” is copied, the name changes to “Value_” after being pasted.
2	“_x” is added when the variable name above (No.1) already exists. (x = positive integer; lowest number that is not being used)	When the variable “Value” is copied and pasted, if the names “Value” and “Value_” already exist, the name changes to “Value_1” after being pasted. (In this case, when an underscore (_) is added to the end of the variable name (copy source), the name “Value_” is already being used. Therefore, “_1” is added to the end of the name.) Thereafter, if the variable is pasted again, the name changes to “Value_2” after being pasted. (When an underscore (_) or “_1” is added to the end of the variable name (copy source), those names already exist. Therefore, “ 2” is added.)
3	If the maximum number of characters (32) is exceeded in the variable name in the above operations (No.1 or No.2), then one or more characters are deleted from the original name before adding an underscore (_) or “_x”. The number of characters that is deleted is determined by how many are added.	After the variable “XXXXXXXXXXYYYYYYYYYYZZZZZZZZZZAB” is copied and pasted, the variable name becomes: “XXXXXXXXXXYYYYYYYYYYZZZZZZZZZZA_”. (In this case, when an underscore (_) is added to the end of the variable name (copy source), the last character “B” is deleted first before adding an underscore (_) to the name. Otherwise, the variable name will exceed the maximum number of characters.) Thereafter, if the variable is pasted again, the name changes to “XXXXXXXXXXYYYYYYYYYYZZZZZZZZZZ_1”. (In this case, when “_1” is added to the end of the variable name (source), the last characters “AB” are deleted first before adding “_1” to the name. Otherwise, the variable name will exceed the maximum number of characters.)

The same global variable in the source cannot be pasted multiple times as an external variable in the same POU.
When editing a CALC command in an LD program, an external variable cannot be copied, cut or pasted.
A parameter variable cannot be pasted as an external variable. An external variable cannot be pasted as a parameter variable.

<u>Copy, cut and paste character strings in a cell</u>
It is possible to copy, cut and paste a selected character string in cells where a character string can be input from the variable editing area.



(NOTE) If the variable name or default value becomes invalid after a character string cut or paste operation, correct the error manually.

Refer to “Chapter 7 LD edit operation procedure”, “Chapter 8 ST edit operation procedure” or “Chapter 9 FBD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

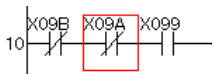
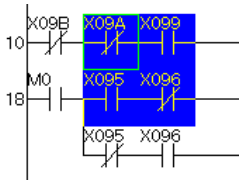
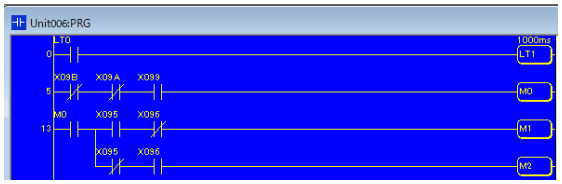
10.2.2 Copy, Cut & Paste POUs

POUs can be copied and pasted in the PLC Programming software D00. However, the user and system defined function POUs can only be cut.
Refer to “Chapter 6 Basic Operation Procedure for Projects” for details on how to use PLC Programming software D00 EDITOR.

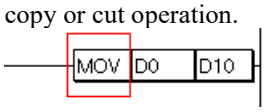
(NOTE 1) The user cannot copy or paste POUs that are open in the PLC Programming software D00 LD/ST Programming Editor and the FBD Programming Editor. Unsaved POUs cannot be copied or cut.
(NOTE 2) The maximum number of POUs that can be created for a single project are as follows: 10 high speed control programs, 10 standard control programs, 924 user defined functions and 100 system defined functions. The user cannot paste any POUs beyond said limits.
(NOTE 3) POUs that are copied cannot be pasted as a different type of POU.
(NOTE 4) POUs that are copied or cut can be pasted multiple times.
(NOTE 5) After pasting a POU, save the project data in the PLC Programming software D00.

10.2.3 Copy, Cut & Paste Circuit Symbols (LD Language)

Copy / Paste
The following shows three ranges and selection methods for copying and cutting circuit symbols.

No.	Range / Method	Example
1	When selecting a single circuit symbol, move the cursor to the desired circuit symbol.	
2	When selecting multiple circuit symbols, use the mouse to drag the cursor to specify the range. The user can specify multiple rows and/or multiple columns.	 Selected ranged is highlighted in blue.
3	To select all the circuits, choose <Select all> under the <Edit> menu in the PLC Programming software D00 LD/ST Programming Editor.	

(NOTE) When the cursor is on a command, the entire command becomes selected for the copy or cut operation.
For example, the entire MOV command in the figure below becomes selected for the copy or cut operation.



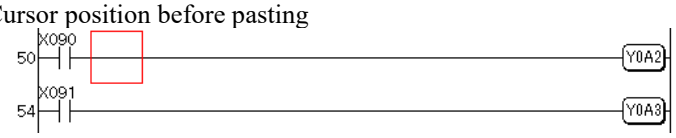
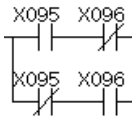
A circuit symbol that has been copied or cut can be pasted.
When a circuit symbol has been cut, the circuit block that includes that circuit symbol becomes unconverted.

Refer to “Chapter 7 LD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

Paste

A circuit symbol that has been copied or cut can be pasted. After it is pasted, the ladder circuit becomes unconverted.

The paste operation is different for “Overwrite mode” and “Insert mode” in the PLC Programming software D00 LD/ST Programming Editor.
When using overwrite mode, a circuit symbol (that has been copied or cut) overwrites and pastes at the cursor position.
When using insert mode, a line is inserted above the cursor position and a circuit symbol (that has been copied or cut) is pasted there.
Refer to the following examples.

Circuit symbol being pasted	Mode	Operation
		Cursor position before pasting 
	Overwrite mode	After pasting
	Insert mode	After pasting
	Overwrite mode	After pasting
	Insert mode	After pasting

Refer to “Chapter 7 LD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

(NOTE 1) While the PLC Programming software D00 LD/ST Programming Editor is launched, the user can also paste a circuit symbol (that has been copied or cut) to another POU (LD language).
At this time, if a coil remark is set to the circuit (copy source), then that coil remark is also pasted to the copy destination. In addition, if a label (L) with an OM comment or label comment is used, then the user can choose whether or not to paste the OM comment or label comment to the POU (copy destination).

(NOTE 2) When the cursor is at the END line, the copy and cut operations cannot be executed.

(NOTE 3) The paste operation is not possible in the following situations.
- When the cursor is outside the rail
- When the pasted circuit goes beyond the right rail
- When a contact touches or overlaps the right rail after being pasted
- When an output does not overlap or touch the right rail after being pasted

(NOTE 4) When a paste operation is performed with the cursor at the END line, a line is inserted above the END line (even in overwrite mode) and then the paste operation is executed.

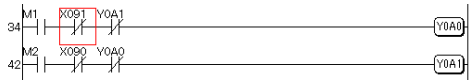
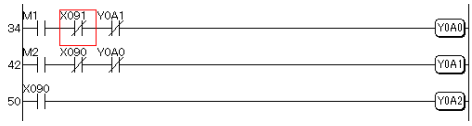
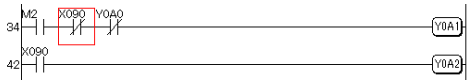
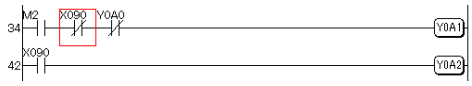
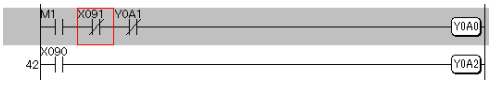
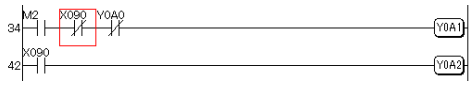
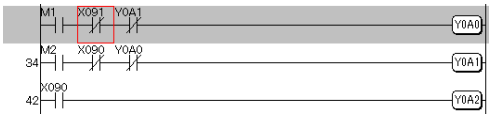
(NOTE 5) If the program becomes invalid after a cut or paste operation, correct the error manually.

10.2.4 Copy, Cut & Paste One Line (LD Language)

This function copies, cuts or pastes the line where the cursor is.
The user can paste the line that was copied or cut.
If a line is cut or if the whole circuit block is cut, then the data is automatically converted. If part of the circuit block is cut, then that circuit block becomes unconverted.
When a line is pasted, that line becomes unconverted.

The paste operation is different for “Overwrite mode” and “Insert mode” in the PLC Programming software D00 LD/ST Programming Editor.
When using overwrite mode, a single line (that is copied or cut) overwrites and pastes at the line with the cursor.
When using insert mode, a line is inserted above the cursor position and then the single line (that is copied or cut) overwrites and pastes there.

Operation examples are shown below.

Function	Before execution	After execution
Copy a line		(There is no change in the circuit because it is a copy operation.)
Cut a line		 The line at the cursor position is cut and the data shifts up.
Paste a line (Overwrite mode)		 The line at the cursor position is overwritten.
Paste a line (Insert mode)		 A line is inserted above the cursor position.

Refer to “Chapter 7 LD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

(NOTE 1) Even if multiple lines are selected, the line with the cursor is the line that is copied, cut or pasted.

(NOTE 2) While the PLC Programming software D00 LD/ST Programming Editor is launched, the user can also paste a single circuit line (that has been copied or cut) to another POU (LD language).
At this time, if an inline comment or coil remark is set to the circuit block (copy source), then that inline comment or coil remark is also pasted to the copy destination. In addition, if a label (L) with an OM comment or label comment is used, then the user can choose whether or not to paste the OM comment or label comment to the POU (copy destination).

(NOTE 3) The <Paste> selection under the <Edit> menu cannot be used to paste a single line that has been copied or cut.

(NOTE 4) When the cursor is at the END line, a single line copy or cut operation cannot be executed.

(NOTE 5) A single line cannot be pasted at the END line when in overwrite mode.

(NOTE 6) If the program becomes invalid after a single line is cut or pasted, correct the error manually.

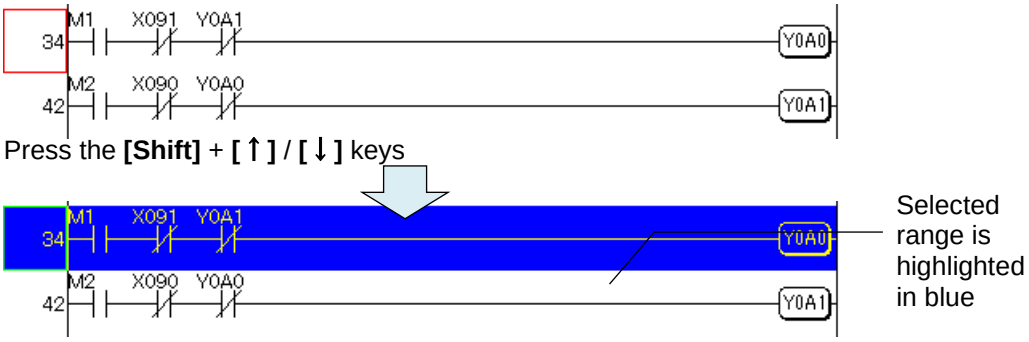
10.2.5 Copy, Cut & Paste Circuit Blocks (LD Language)

The user can copy, cut and paste the selected circuit block.
Circuit blocks that are copied or cut can be pasted.

Circuit block selection

When selecting a circuit block, move the cursor to the area where the step number is displayed and then press the **[Shift]** + **[↑]** keys or the **[Shift]** + **[↓]** keys.
Note that multiple circuit blocks can be selected by pressing the **[Shift]** + **[↑]** keys or the **[Shift]** + **[↓]** keys repeatedly.

Ex:



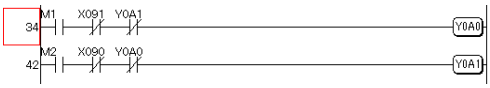
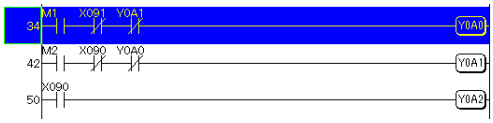
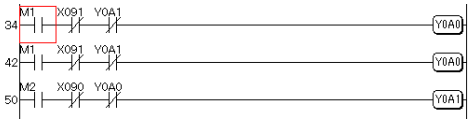
Copy, cut and paste circuit blocks

The user can copy and cut a circuit block that is selected. Circuit blocks that are copied or cut can be pasted.
When a circuit block is pasted, a line is inserted above the cursor position and then the data is pasted.
When a circuit block is cut or pasted, it is automatically converted.

However, a pasted circuit block becomes unconverted in the following situations.

- When data is pasted in the middle of a circuit block
- When a circuit block with a CALC command is pasted

Operation examples are shown below.

Function	Before execution	After execution
Copy		 (There is no change in the circuit because it is a copy operation.)
Cut		The selected circuit block is cut, the data shifts up.
Paste		 A circuit block is inserted above the cursor position.

Refer to "Chapter 7 LD edit operation procedure" for details on how to use PLC Programming software D00 EDITOR.

(NOTE 1) While the PLC Programming software D00 LD/ST Programming Editor is launched, the user can also paste a circuit block (that has been copied or cut) to another POU (LD language).
At this time, if an inline comment or coil remark is set to the circuit block (copy source), then that inline comment or coil remark is also pasted to the copy destination. In addition, if a label (L) with an OM comment or label comment is used, then the user can choose whether or not to paste the OM comment or label comment to the POU (copy destination).

(NOTE 2) When the END line is selected, the copy and cut operations cannot be executed.

(NOTE 3) If the program capacity will be exceeded, the paste operation cannot be executed.

(NOTE 4) When pasting a circuit block, a line is inserted above the cursor position regardless of the overwrite or insert mode setting.

(NOTE 5) If the program becomes invalid after a cut or paste operation, correct the error manually.

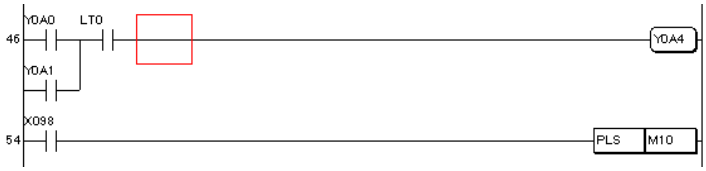
(NOTE 6) When the PLC Programming software D00 LD/ST Programming Editor is equipped with the “Reuse circuit“ function, the user can copy a circuit block with this function. Refer to “10.2.6 Reuse circuit (LD language)” for further details.

10.2.6 Reuse Circuit (LD Language)

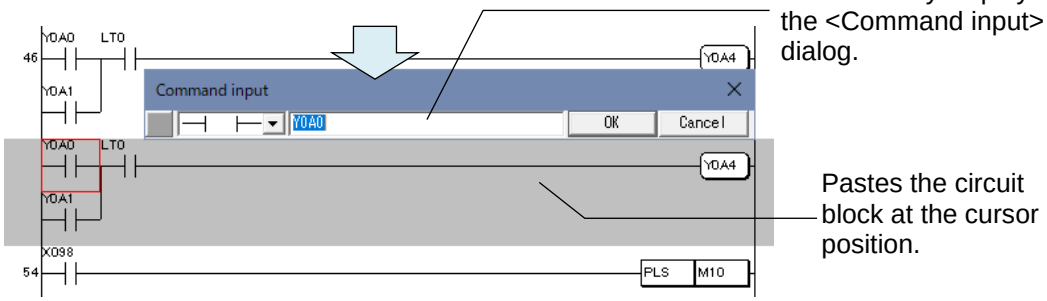
This function copies the circuit block at the cursor position.
When the reuse circuit function is executed, the following operations are carried out.

- The circuit block at the cursor position is copied, and the data is pasted (inserted) at the next line.
- The <Command input> dialog is automatically displayed. The user can change OMs or commands consecutively for each command that is pasted.

Example



Reuse circuit



A pasted circuit block becomes unconverted.

Refer to “Chapter 7 LD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

(NOTE 1) When the cursor is at the END line, the reuse circuit function cannot be executed.

(NOTE 2) The circuit block is inserted regardless of the overwrite or insert mode setting.

10.2.7 Copy, Cut & Paste (Character String) in ST Program (ST Language)

The user can copy and cut a selected character string in the ST language and for CALC commands in the LD language. In addition, the copied or cut character strings can also be pasted.

Refer to " Chapter 8 ST edit operation procedure" for details on how to use PLC Programming software D00 EDITOR.

(NOTE 1) While the CALC command is being edited in the LD language, the menu cannot be selected in the PLC Programming software D00 LD/ST Programming Editor.
Select <Copy>, <Cut> or <Paste> from the context menu, or use the shortcut keys.

(NOTE 2) If the program becomes invalid after a cut or paste operation, correct the error manually.

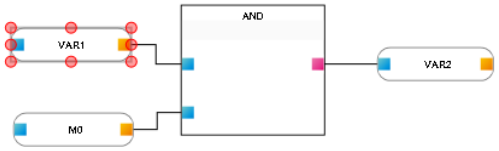
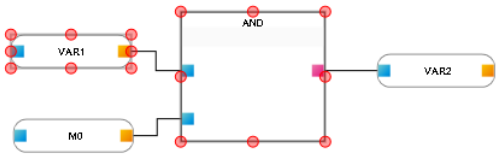
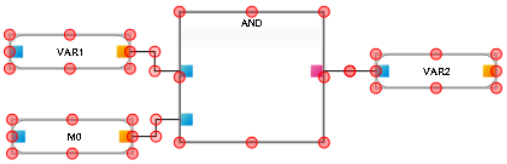
(NOTE 3) The user can paste a copied or cut character string to a commercially available text editor, because the data being copied, cut and pasted is text data.

10.2.8 Copy, Cut & Paste Boxes (FBD Language)

Selected function boxes, variable boxes, literal boxes, OM boxes and return value boxes can be copied and cut in an FBD program. In addition, the copied or cut boxes can also be pasted. It is also possible to copy and cut multiple (selected) boxes at the same time.

Copy / Paste

The following shows three selection methods for copying and cutting boxes.

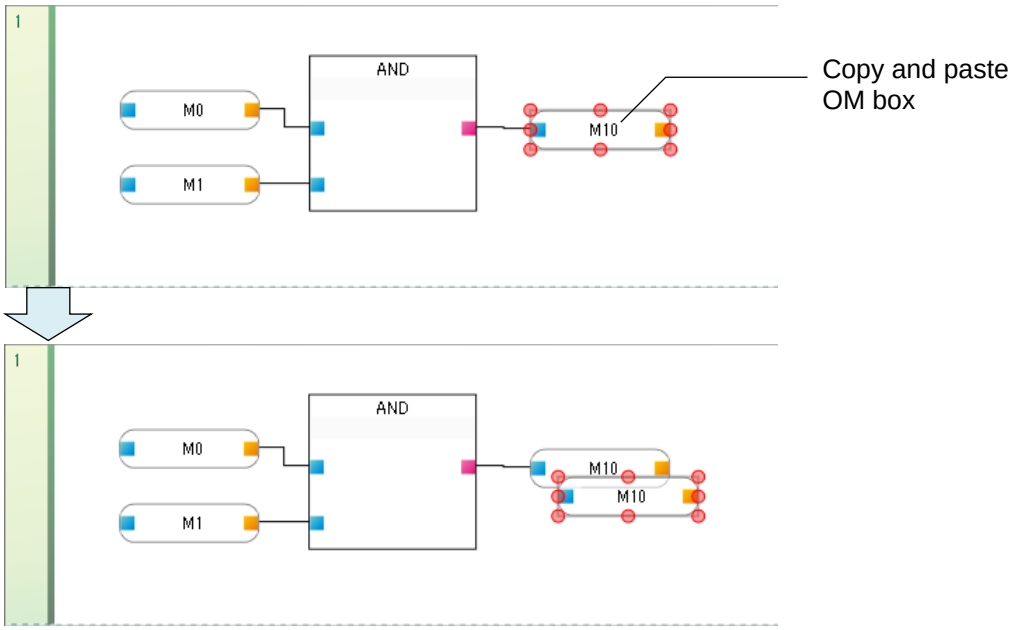
No.	Range / Method	Example
1	To select a single box, click on the desired box.	Click on the variable box "VAR1" 
2	To select multiple boxes, hold down the **[Shift]** key or the **[Ctrl]** key and click on the desired boxes.	Hold down the **[Shift]** key and then click on the variable box "VAR1" and the function box "AND" 
3	To select all the boxes in a selected network, choose <Control in network> under <Select all> in the <Edit> menu in the PLC Programming software D00 FBD Programming Editor.	

(NOTE) The user can use the above method to select connectors, but connectors cannot be copied, cut or pasted.
It is possible to delete selected connectors.
In addition, when a box is cut or deleted, the connectors connected to that box are also automatically deleted.

Paste

When performing a paste operation, the data can be pasted in the selected network.

Example



Refer to "Chapter 9 FBD edit operation procedure" for details on how to use PLC Programming software D00 EDITOR.

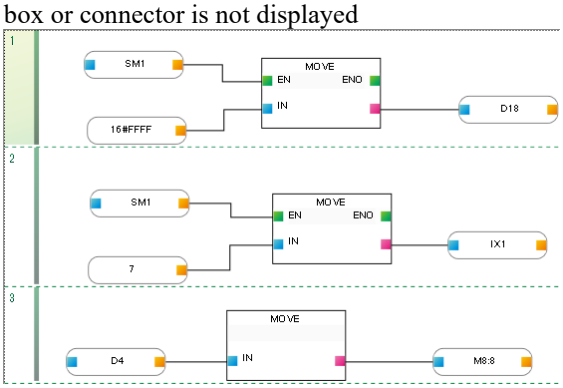
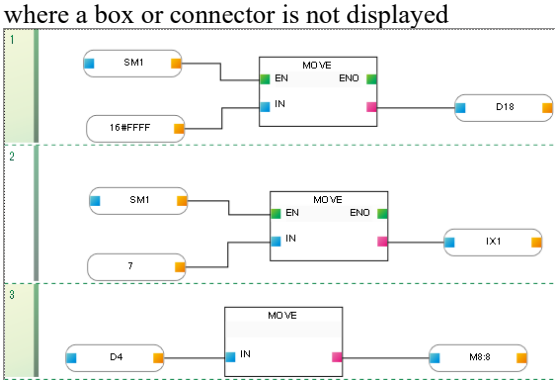
(NOTE 1) When a copy or cut operation is performed and a box is not selected, then the selected network is copied or cut. Refer to "10.2.9 Copy, cut & paste networks (FBD language)" for details on copying networks.

(NOTE 2) When a network is not selected, the paste operation will not execute.

10.2.9 Copy, Cut & Paste Networks (FBD Language)

The user can copy or cut a selected network in an FBD program. In addition, the copied or cut network can also be pasted.
If a network is selected but no box in that network is selected, then the network is copied or cut.

Copy / Paste
The following shows two selection methods for copying and cutting networks.

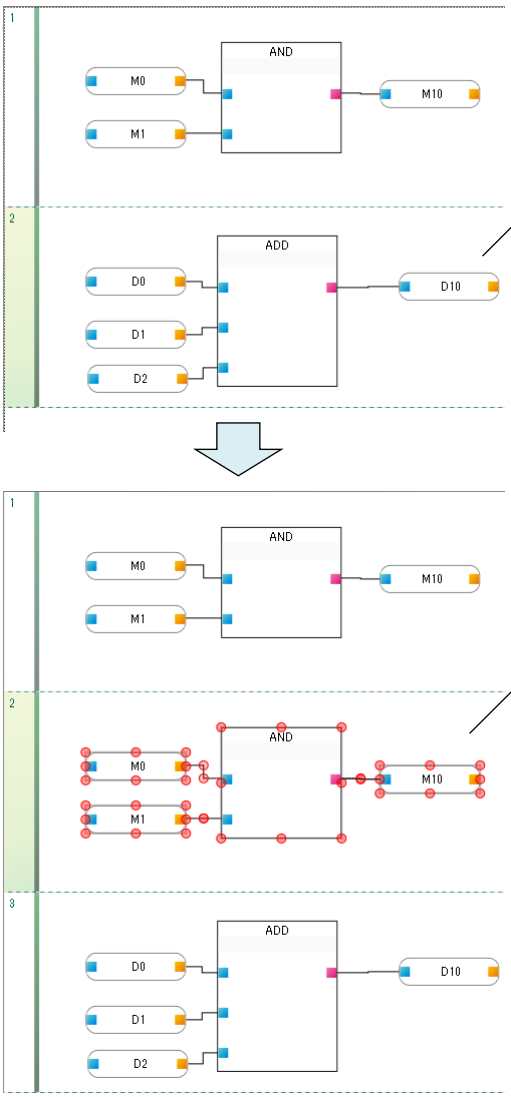
No.	Range / Method	Example
1	To select one network, click on an empty space in the desired network where a box or connector is not displayed.	Click on an empty space in the No.1 network where a box or connector is not displayed  The network No. is highlighted in green for the selected network.
2	To select multiple networks, hold down the **[Ctrl]** key and click on an empty space in the desired network where a box or connector is not displayed.	Click on an empty space in the No.1 and No.3 network where a box or connector is not displayed 

(NOTE) When <Network> is selected under <Select all> in the <Edit> menu in the PLC Programming software D00 FBD Programming Editor, all the boxes and connectors in the network become selected. In this situation, if a copy or cut operation is performed, the boxes, not the network, are copied or cut.

Paste

When performing a paste operation, the copied or cut network is inserted in the selected network. When there is no selected network, the copied or cut network is inserted at the last line.

Ex: Paste when a network is selected

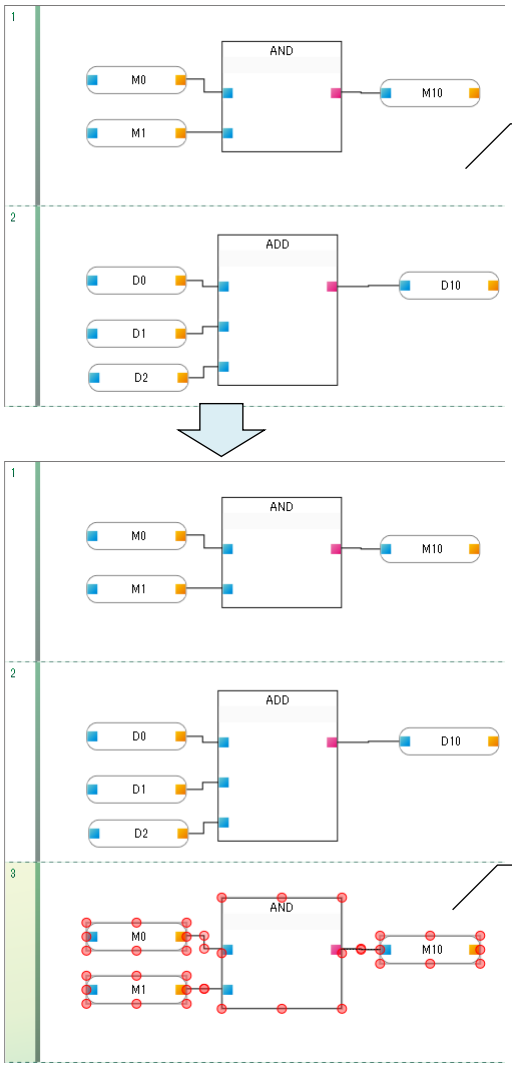


Copy network No.1, and paste when network No.2 is selected

Inserted copied network No.1 above the selected network

10

Ex: Paste when a network is not selected



Copy network No.1, and paste when a network is not selected

Copied network No.1 is inserted at the bottom row (network)

Refer to “Chapter 9 FBD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

10.3 Delete

In the PLC Programming software D00 EDITOR, it is possible to delete POUs, variables, circuit symbols and character strings.
In contrast with the “Cut” operation, deleted data cannot be pasted.
The following can be deleted.

Tool / Language	Applicable operations
Common Variable editing area	• Variable • Character string
PLC Programming software D00	• POU • Folders (only folders created by the user can be deleted)
PLC Programming software D00 LD/ST Programming Editor When editing an LD program	• Circuit symbol • Circuit line • Circuit block
	When editing a program in the CALC command edit dialog • Character string
PLC Programming software D00 LD/ST Programming Editor When editing an ST program	• Character string
PLC Programming software D00 LD/ST Programming Editor When editing an FBD program	• Function boxes, variable boxes, literal boxes and OM boxes • Network

Refer to “Chapter 7 LD edit operation procedure”, “Chapter 8 ST edit operation procedure” or “Chapter 9 FBD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

(NOTE 1) When a POU is deleted, it cannot be restored. Therefore, exercise caution when deleting POUs.

(NOTE 2) When a folder is deleted, all of the POUs in that folder are deleted. The deleted folder and POU(s) cannot be restored. Therefore, exercise caution when deleting a folder.

(NOTE 3) A POU cannot be deleted when it is currently being edited in the PLC Programming software D00 LD/ST Programming Editor or in the FBD Programming Editor.

10.4 Search & Jump

OMs, variables, character strings and OM comments being used in an LD, ST or FBD program can be searched.
There are various search methods when using the PLC Programming software D00 EDITOR and each varies depending on the language being used. The following is a list of the search methods.

Tool / Language	Search method	Description
PLC Programming software D00 LD /ST Programming Editor When editing an LD program	Contact/Coil list	Displays a list of where a specified OM is being used.
	OM list	Lists how many times a given OM is used.
	Timer/Counter list	Lists the set values for the local timer (LT) and the local counter (LC). The user can also change all the set values at once.
	OM search	Searches for circuit blocks that use the specified OM and displays the search results.
	OM search (contact)	Searches for circuit blocks that use the specified OM as a contact and displays the search results.
	OM search (coil)	Searches for circuit blocks that use the specified OM as an output and displays the search results.
	OM search (all)	Searches all circuits that use the specified OM and displays the search results.
	Command search	Searches for circuit blocks that use the specified command and displays the search results.
	Search up	Starts from the cursor position and moves up in the ladder circuits searching for an OM. The display is based on the search results.
	Search down	Starts from the cursor position and moves down in the ladder circuits searching for an OM. The display is based on the search results.
	Comment search	Searches for a comment (character string) that is set in the ladder circuits.
	Step number search	Moves the cursor to the beginning of the circuit block with the specified step number, and the display is sorted according to the step number.
	OM jump	Jumps or moves the cursor to the circuit block that uses the specified OM, and the display is sorted according to the step number.
	OM jump (contact)	Jumps or moves the cursor to the circuit block that uses the specified OM as a contact, and the display is sorted according to the step number.
	OM jump (coil)	Jumps or moves the cursor to the circuit block that uses the specified OM as an output, and the display is sorted according to the step number.
	Command jump	Jumps or moves the cursor to the circuit block that uses the specified command, and the display is sorted according to the step number.
PLC Programming software D00 LD /ST Programming Editor When editing an ST program	Character string search	Searches for the specified character string.
	OM list	Lists how many times a given OM is used.
	OM search	Searches for the specified OM.
PLC Programming software D00 FBD Programming Editor	Character string search	Searches for the specified character string.
	OM list	Lists how many times a given OM is used.
	OM search	Searches for the specified OM.

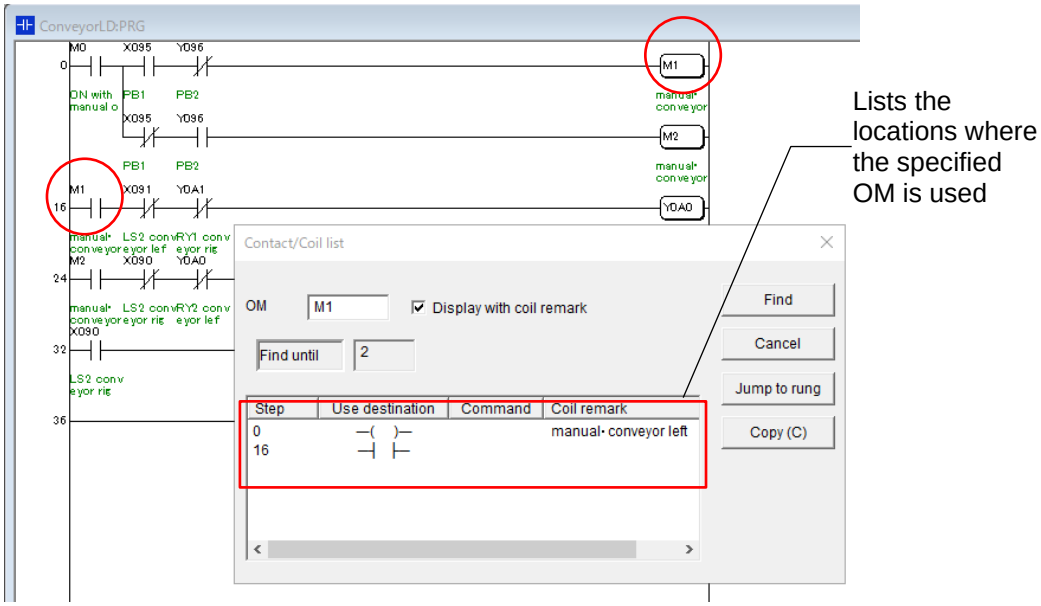
(NOTE 1) The search function can also be used when the PLC Programming software D00 EDITOR is in monitoring mode (excluding timer/counter list). This function is not only available during program editing but can also be used for debugging.

(NOTE 2) Refer to “10.11.2 Step number display (LD language)” and “10.11.3 Search results display (LD language)” for details about the step number display or search results display.

10.4.1 Contact/Coil List (LD Language)

This function displays a list of where the specified OM is used in a POU.

Ex: Search locations where M1 is used



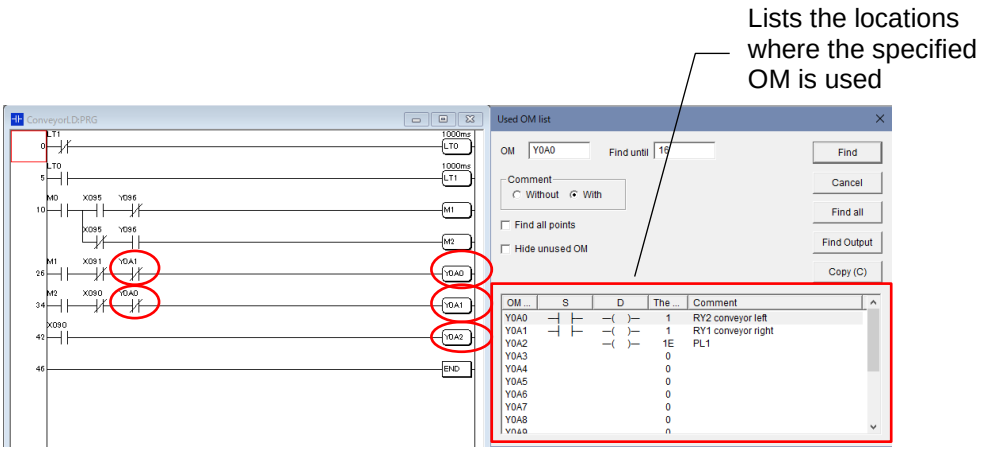
Item	Description	
Search range	POU	
Search target	Specified OM.	
Search result	Displays the OM search results in the <Contact/Coil list> dialog. The following items are displayed.	
	Step	Displays the step number for the circuit block where the specified OM is used.
	Use destination	Displays whether the specified OM is being used in a contact, output or command. ─┤ ├─ : Used in a contact ─( )─ : Used in an output ─[ ]─ : Used in an application command
	Command	Displays the command name when the specified OM is not being used in a contact or OUT command.
	Coil remark	Displays a coil remark that is set to a circuit block with the specified OM. (User can show or hide this remark.)

The search and display for the contact/coil list is carried out in the <Contact/Coil list> dialog. Refer to “Chapter 7 LD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

10.4.2 OM List (LD Language)

This function lists the how many times the specified OM is used in a POU.

Ex: Search how many times the OM is used from YA0 to YAF



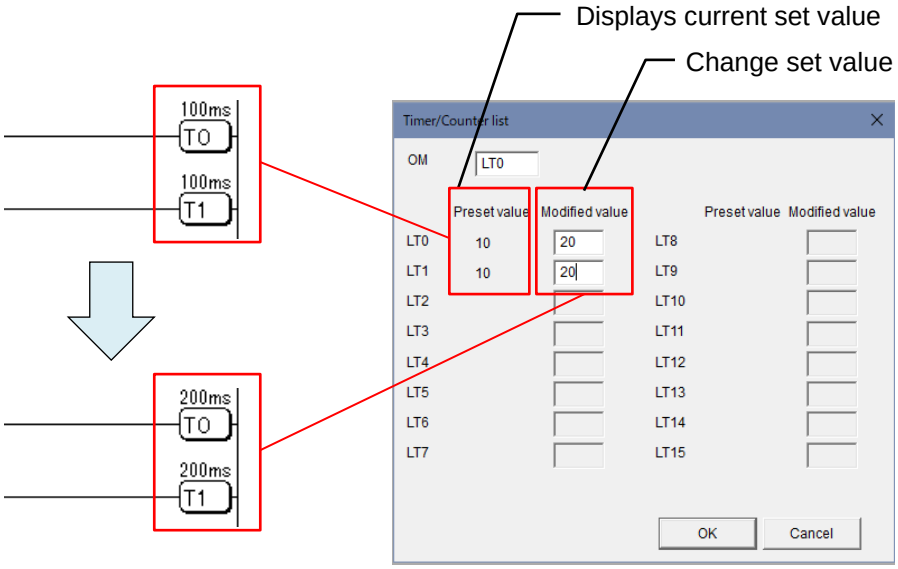
Item	Description	
Search range	POU	
Search target	Specifies the beginning address in the OM search and the search range. For example, if YA0 is specified as the beginning address and the range is set to 16, then the addresses YA0 to YAF are searched. It is also possible to search all locations for the same type as the specified OM.	
Search result	Displays the OM search results in the <Used OM list> dialog. The following items are displayed.	
	OM name	Displays the OM names in the search range. (When the checkbox for <Hide unused OMs> is ticked, only the OMs used in the target POU are displayed.)
	S	Displays ⊣ ⊢ when the specified OM is being used as the source for a contact or an application command.
	D	Displays ⊣( )⊢ when the specified OM is being used as the destination for a contact or an application command.
	Quantity being used	Displays the quantity or how many times the specified OM is being used as the destination for a contact or an application command. When the OM is only being used as either a source for a contact/application command or a destination for an output/application command, then an "E" is displayed after the quantity number. For example, if there was an OM that is only being used in a contact, then the "E" acts as a warning indicating that there may be a problem with the LD program, such as having no circuit that turns ON that OM, etc. Correct or revise the LD program when necessary.
	Comment	Displays the OM comment that is set to the OM in the search. (When <Display comment> is set to <Show>, the OM comment is displayed.)

The search and display for the OM list is carried out in the <Used OM list> dialog.
Refer to "Chapter 7 LD edit operation procedure" for details on how to use PLC Programming software D00 EDITOR.

10.4.3 Timer/Counter List (LD Language)

This function lists the set values for the local timer (LT) and the local counter (LC) that are used in the LD program.
The user can also change all the set values at once.

Ex: Display and change the LT0 and LT1 set values



Item	Description
Search range	POU
Search target	All local timers (LT) and local counters (LC).
Search result	Lists the set values for the local timer and the local counter in the <Timer/Counter list>. The fields are blank for the local timer (LT) and the local counter (LC) when they are not used. If an OM is specified for the set value instead of a constant, then that OM is displayed.
Change set value	The user can change all the set values at once for the local timers and local counters being used. 0 to 32767 and D (OM) can be specified for the set value.

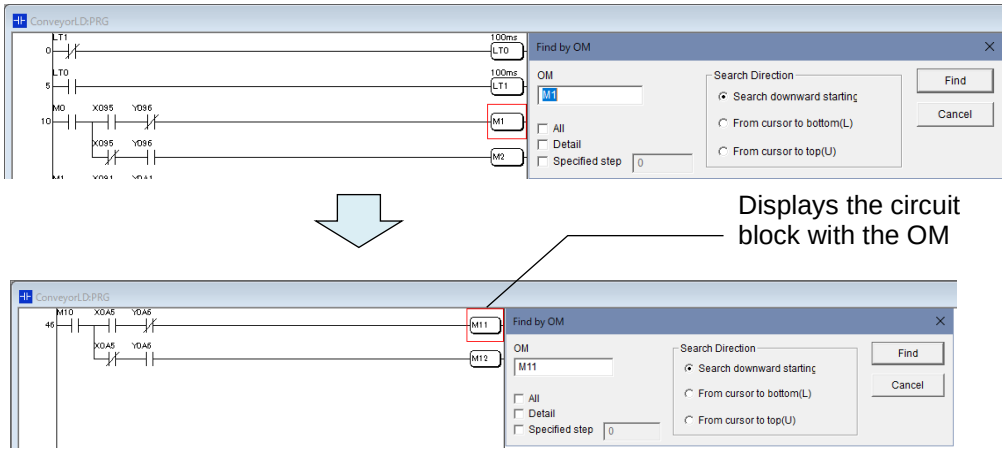
The settings and display for the timer/counter list is carried out in the <Timer/Counter list> dialog. Refer to “Chapter 7 LD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

(NOTE) This operation cannot be carried out when the data is not converted.

10.4.4 OM Search (LD Language)

This function searches for circuit blocks that use the specified OM and displays the search results.

Ex: Search for M11



The OM search function is carried out from the <Find by OM> dialog. The <Find by OM> dialog does not close even after conducting a search, and the user can continue doing searches. Once the search has gone through all the hits, the dialog closes.
When this OM search is carried out, the ladder circuits are displayed based on the search results.
Note that when one of the following keys is entered (**[a]** to **[z]** and **[A]** to **[Z]**, **[+]**, **[-]**, **[*]** or **[/]** key) while the edit lock is enabled, the OM search is carried out.

(NOTE) Refer to "10.11.2 Step number display (LD language)" and "10.11.3 Search results display (LD language)" for details about the ladder circuit display.

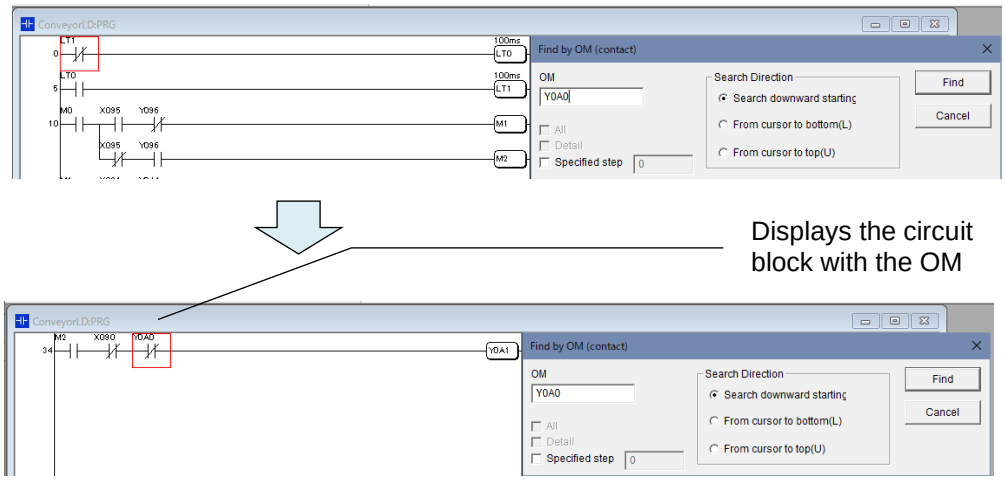
Item	Description	
Search range	POU	
Search target	Specified OM。	
Search options	The user can specify the following search options: "All", "Details" or "Specified step".	
	All	When this option is specified, the search is conducted on all circuit blocks that use the specified OM and displays the search results.
	Details	When this option is specified, the "Digit specified bit OM" or "Longword specified word OM" are also searched. Ex: M17 search when "M16:8" is used in the ladder circuits. Option not used: M17 is not found in search. Option used: M17 is found in search. M16:8 is displayed as a hit in the search. D1 search when "D07" is used in the ladder circuits. Option not used: D1 is not found in search. Option used: D1 is found in search. D0L is displayed as a hit in the search.
	Specified step	When this option is used, it specifies the step number. The search range starts from the specified step number.
Search result	Starts from the beginning of the ladder circuit and searches for the specified OM. When an OM is found (hit), the cursor moves to the location where that OM is used.	

Refer to "Chapter 7 LD edit operation procedure" for details on how to use PLC Programming software D00 EDITOR.

10.4.5 OM Search (Contact) (LD Language)

This function searches for circuit blocks that use the specified OM as a contact and displays the search results.

Ex: Search for YA0



The OM search (contact) function is carried out from the <Find by OM (contact)> dialog. The <Find by OM (contact)> dialog does not close even after conducting a search, and the user can continue doing searches. Once the search has gone through all the hits, the dialog closes.
When this OM search (contact) is carried out, the ladder circuits are displayed based on the search results.

(NOTE) Refer to “10.11.2 Step number display (LD language)” and “10.11.3 Search results display (LD language)” for details about the ladder circuit display.

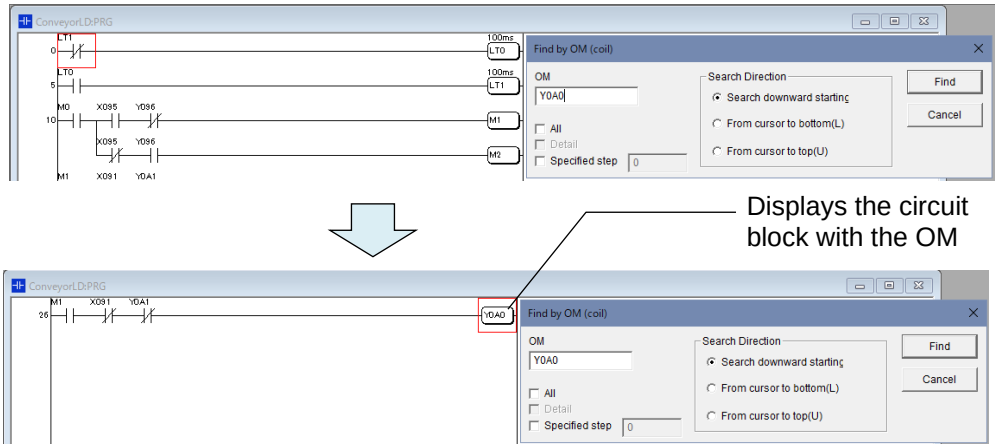
Item	Description	
Search range	POU	
Search target	Specified OM	
Search options	The user can set the search option: “Specified step”.	
	Specified step	When this option is used, it specifies the step number. The search range starts from the specified step number.
Search result	Starts from the beginning of the ladder circuit and searches for locations where the specified OM is used as a contact. When an OM is found (hit), the cursor moves to the location where that OM is used. If there are multiple locations where the same OM is used in the ladder circuit, the user can continue searching the locations for the same OM.	

Refer to “Chapter 7 LD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

10.4.6 OM Search (Coil) (LD Language)

This function searches for circuit blocks that use the specified OM as an output and displays the search results.

Ex: Search for YA0



This OM search (coil) function searches for all OMs that are used in a CALC command.
For example, when the following process is noted for the CALC command and the OM search (coil) is conducted for the D0 and D1, this CALC command is found (hit) in the search for both cases.

```
1   D0:=D1+1;
2
3
```

The OM search (coil) function is carried out from the <Find by OM (coil)> dialog. The <Find by OM (coil)> dialog does not close even after conducting a search, and the user can continue doing searches. Once the search has gone through all the hits, the dialog closes.
When this OM search (coil) is carried out, the ladder circuits are displayed based on the search results.

(NOTE) Refer to “10.11.2 Step number display (LD language)” and “10.11.3 Search results display (LD language)” for details about the ladder circuit display.

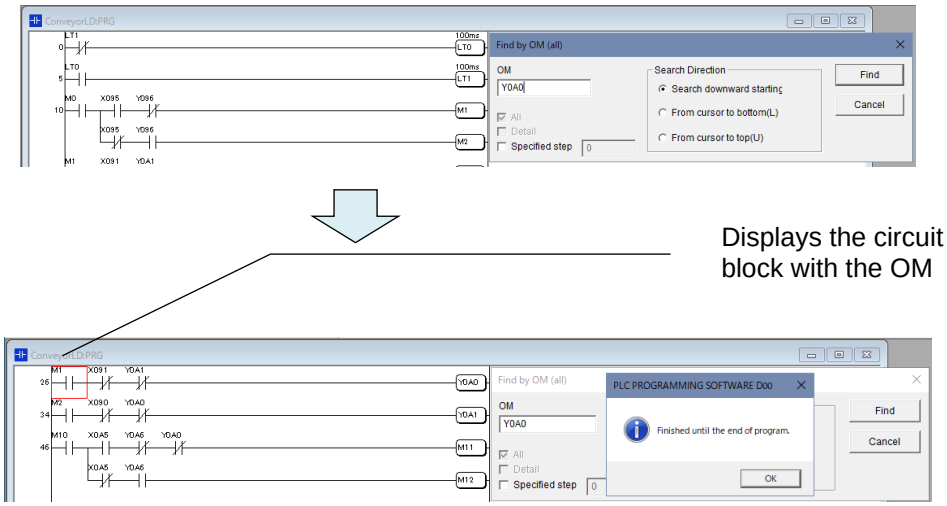
Item	Description	
Search range	POU	
Search target	Specified OM。	
Search options	The user can specify the following search options: “All” or “Specified step”.	
	All	When this option is specified, the search is conducted on all circuit blocks where the specified OM is used as an output.
	Specified step	When this option is used, it specifies the step number. The search range starts from the specified step number.
Search result	Starts from the beginning of the ladder circuit and searches for locations where the specified OM is used as an output. When an OM is found (hit), the cursor moves to the location where that OM is used. If there are multiple locations where the same OM is used in the ladder circuit, the user can continue searching the locations for the same OM.	

Refer to “Chapter 7 LD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

10.4.7 OM Search (All) (LD Language)

This function searches all the circuit blocks that use the specified OM.

Ex: Search for YA0



When this OM search (all) is carried out, the ladder circuits are displayed based on the search results.

Item	Description	
Search range	POU	
Search target	Specified OM	
Search options	The user can set the search option: “Specified step”.	
	Specified step	When this option is used, it specifies the step number. The search range starts from the specified step number.
Search result	Searches all locations for the specified OM and displays all the circuit blocks that are found in the search. When an OM is found (hit), the cursor moves to the first location where that OM is used.	

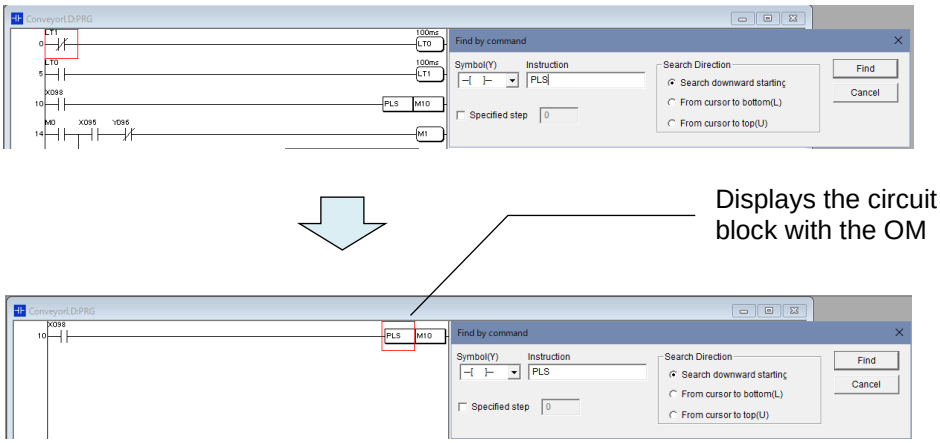
The OM search (all) function is carried out from the <Find by OM (all)> dialog.
Refer to “Chapter 7 LD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

(NOTE) Refer to “10.11.2 Step number display (LD language)” and “10.11.3 Search results display (LD language)” for details about the ladder circuit display.

10.4.8 Command Search (LD Language)

This function searches for circuit blocks that use the specified command and displays the search results.

Ex: Search for PLS command



The command search function is carried out from the <Find by command> dialog. The <Find by command> dialog does not close even after conducting a search, and the user can continue doing searches. Once the search has gone through all the hits, the dialog closes.
When this command search is carried out, the ladder circuits are displayed based on the search results.

Item	Description	
Search range	POU	
Search target	Specified command.	
Search options	The user can set the search option: “Specified step”.	
	Specified step	When this option is used, it specifies the step number. The search range starts from the specified step number.
Search result	Starts from the beginning of the ladder circuit searching for the specified command and displays the circuit block where the command is found. The cursor moves to the location where the command is used.	

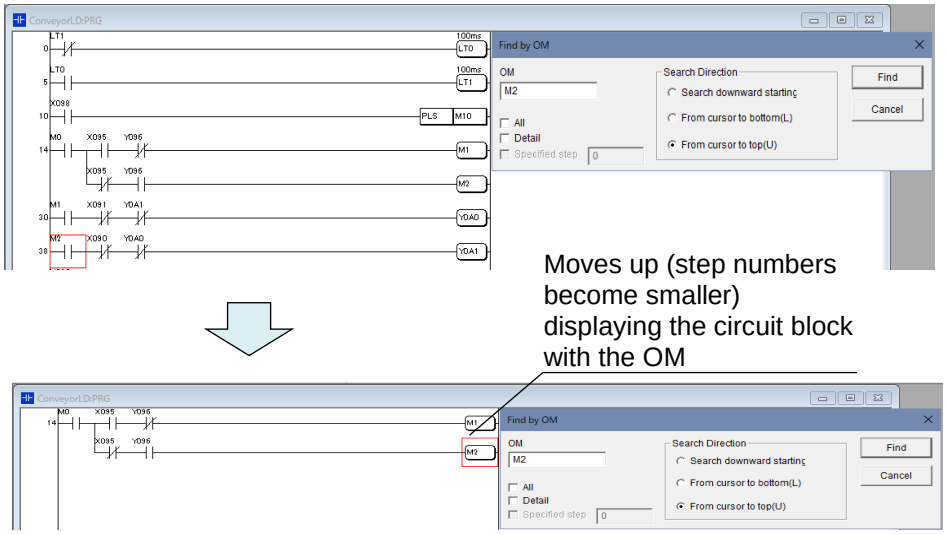
Refer to “Chapter 7 LD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

(NOTE) Refer to “10.11.2 Step number display (LD language)” and “10.11.3 Search results display (LD language)” for details about the ladder circuit display.

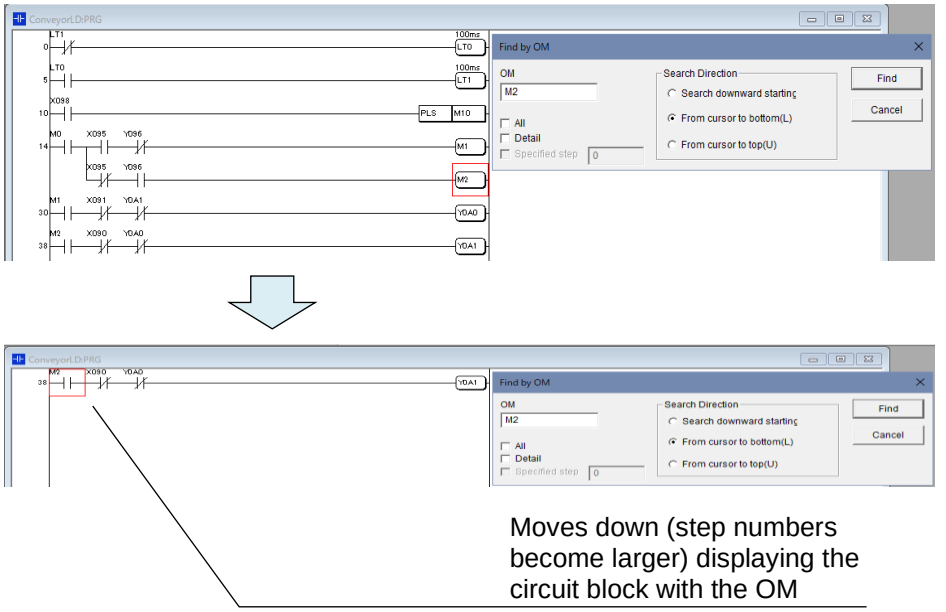
10.4.9 Search Up & Search Down (LD Language)

An OM search is conducted starting from the cursor position and moves up or down in the ladder circuits.

Example 1: Search for M2 starting at cursor position and moving up



Example 2: Search for M2 starting at cursor position and moving down



Once the top or bottom is reached, the search is finished.
When this OM search is carried out, the ladder circuits are displayed based on the search results.

Item	Description
Search range	POU
Search target	OM at cursor position
Search result	In the search up operation, it starts from the cursor position and searches up (step numbers become smaller). The first circuit block that is found (hit) is displayed. In the search down operation, it starts from the cursor position and searches down (step numbers become larger). The first circuit block that is found (hit) is displayed.

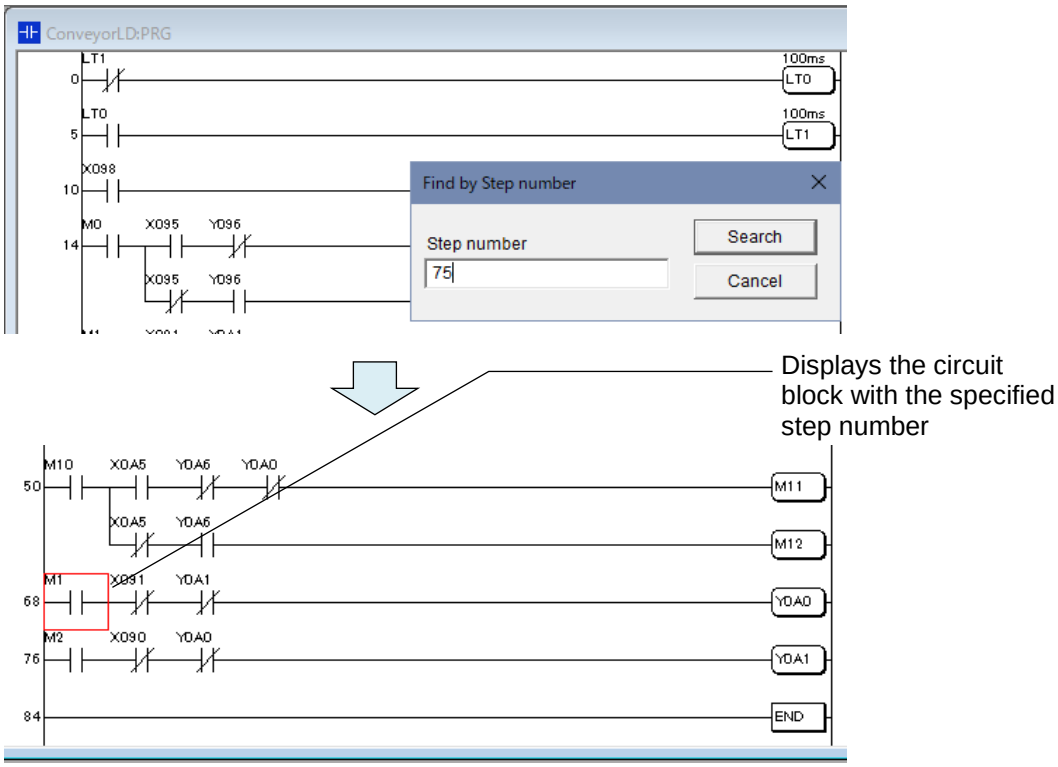
Refer to “Chapter 7 LD edit operation procedure” for details on searching up and searching down.

(NOTE 1) When the cursor is not displayed on the OM, the search up and search down operations cannot be executed.
(NOTE 2) Refer to “10.11.2 Step number display (LD language)” and “10.11.3 Search results display (LD language)” for details about the ladder circuit display.

10.4.10 Step Number Search (LD Language)

This function searches for the specified step number and displays the circuit block with that step.

Ex: Search for step No.75



Item	Description
Search range	POU
Search result	Searches for the specified step number and displays the circuit block with that step. The cursor moves to the beginning of that circuit block.

When this step number search is carried out, the ladder circuits are displayed based on the step number order.
Note that when a key between **[0]** and **[9]** is entered while the edit lock is enabled, the step search is carried out.

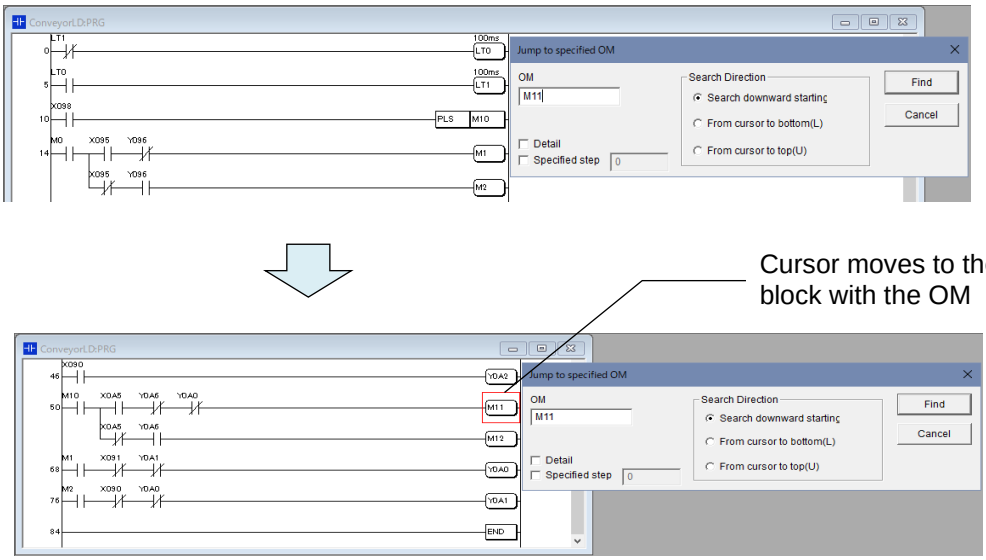
The step number search is carried out from the <Find by step number> dialog.
Refer to “Chapter 7 LD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

(NOTE 1) Refer to “10.11.2 Step number display (LD language)” and “10.11.3 Search results display (LD language)” for details about the ladder circuit display.
(NOTE 2) Refer to “10.8 Edit lock” for details about the edit lock.

10.4.11 OM Jump (LD Language)

This function searches for the circuit block that uses the specified OM, and jumps to the location where the OM is found (hit).

Ex: Search for M11



The OM jump function is carried out from the <Find and jump by OM> dialog. The <Find and jump by OM> dialog does not close even after conducting a search, and the user can continue doing searches.
When this OM jump is carried out, the ladder circuits are displayed based on the step order.

(NOTE) Refer to “10.11.2 Step number display (LD language)” and “10.11.3 Search results display (LD language)” for details about the ladder circuit display.

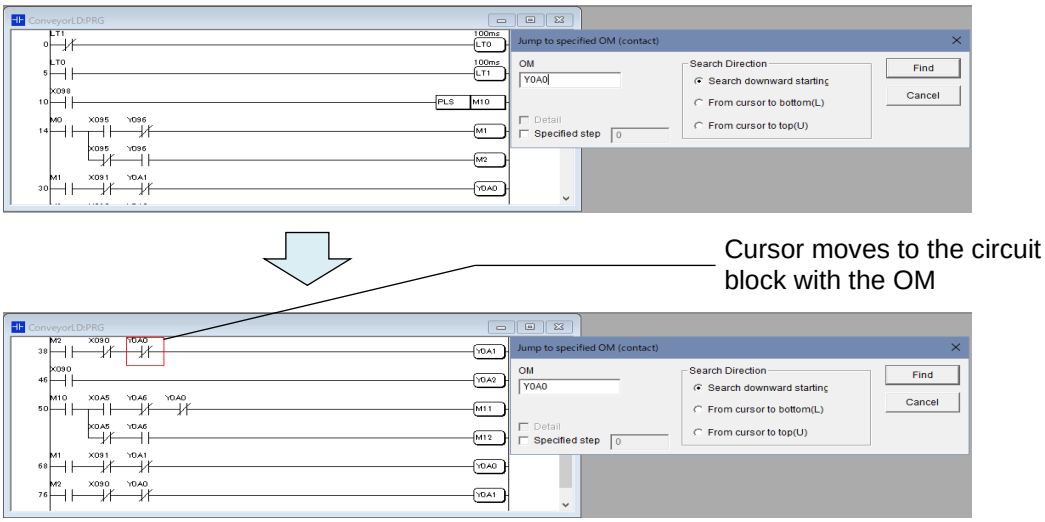
Item	Description	
Search range	POU	
Search target	Specified OM	
Search options	The user can specify the following search options: “Details” or “Specified step”.	
	Details	When this option is specified, the “Digit specified bit OM” or “Longword specified word OM” are also searched. Ex: M17 search when “M16:8” is used in the ladder circuits. Option not used: M17 is not found in search. Option used: M17 is found in search. M16:8 is displayed as a hit in the search. D1 search when “D07” is used in the ladder circuits. Option not used: D1 is not found in search. Option used: D1 is found in search. D0L is displayed as a hit in the search.
	Specified step	When this option is used, it specifies the step number. The search range starts from the specified step number.
Search result	Starts from the beginning of the ladder circuit and searches for the specified OM. When an OM is found (hit), the cursor moves to the location where that OM is used.	

Refer to “Chapter 7 LD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

10.4.12 OM Jump (Contact) (LD Language)

This function searches for the circuit block where the specified OM is used as a contact, and jumps to the location where the OM is found (hit).

Ex: Search for YA0



The OM jump function (contact) is carried out from the <Jump to specified OM (contact)> dialog. The <Jump to specified OM (contact)> dialog does not close even after conducting a search, and the user can continue doing searches.
When this OM jump (contact) is carried out, the ladder circuits are displayed based on the step order.

(NOTE) Refer to “10.11.2 Step number display (LD language)” and “10.11.3 Search results display (LD language)” for details about the ladder circuit display.

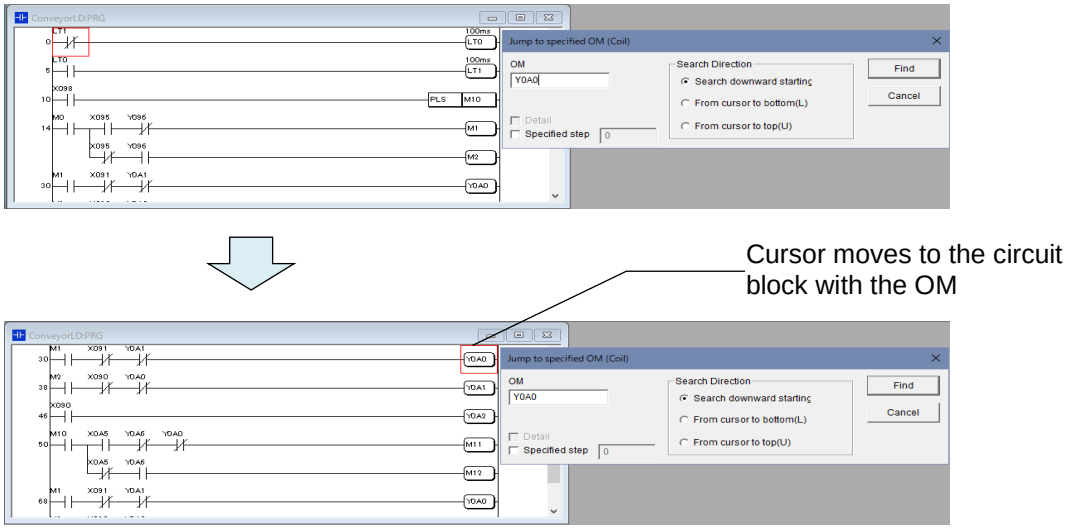
Item	Description	
Search range	POU	
Search target	Specified OM	
Search options	The user can set the search option: “Specified step”.	
	Specified step	When this option is used, it specifies the step number. The search range starts from the specified step number.
Search result	Starts from the beginning of the ladder circuit and searches for locations where the specified OM is used as a contact. When an OM is found (hit), the cursor moves to the location where that OM is used. If there are multiple locations where the same OM is used in the ladder circuit, the user can continue searching the locations for the same OM.	

Refer to “Chapter 7 LD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

10.4.13 OM Jump (Coil) (LD Language)

This function searches for the circuit block where the specified OM is used as an output, and jumps to the location where the OM is found (hit).

Ex: Search for YA0



This OM jump (coil) function searches for all OMs that are used in a CALC command.
For example, when the following process is noted for the CALC command and the OM search (coil) is conducted for the D0 and D1, this CALC command is found (hit) in the search for both cases.

```
1  D0:=D1+1;
2
3
```

The OM jump function (coil) is carried out from the <Find and jump by OM (Coil)> dialog. The <Find and jump by OM (Coil)> dialog does not close even after conducting a search, and the user can continue doing searches.
When this OM jump (coil) is carried out, the ladder circuits are displayed based on the step order.

(NOTE) Refer to “10.11.2 Step number display (LD language)” and “10.11.3 Search results display (LD language)” for details about the ladder circuit display.

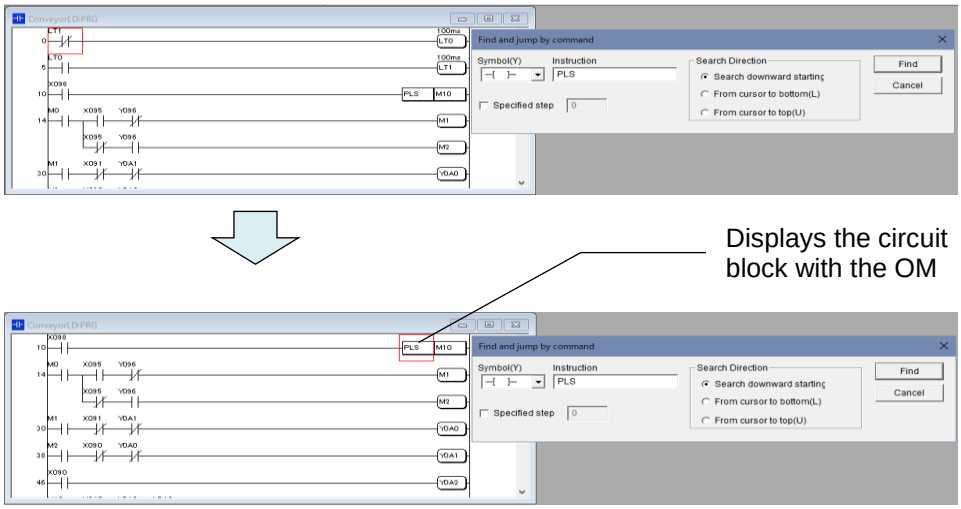
Item	Description	
Search range	POU	
Search target	Specified OM	
Search options	The user can set the search option: “Specified step”.	
	Specified step	When this option is used, it specifies the step number. The search range starts from the specified step number.
Search result	Starts from the beginning of the ladder circuit and searches for locations where the specified OM is used as an output. When an OM is found (hit), the cursor moves to the location where that OM is used. If there are multiple locations where the same OM is used in the ladder circuit, the user can continue searching the locations for the same OM.	

Refer to “Chapter 7 LD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

10.4.14 Command Jump (LD Language)

This function searches for the circuit block that uses the specified command, and jumps to the location where the OM is found (hit).

Ex: Search for PLS command



The command jump function is carried out from the <Find and jump by command> dialog. The <Find and jump by command> dialog does not close even after conducting a search, and the user can continue doing searches.
When this command jump is carried out, the ladder circuits are displayed based on the step order.

Item	Description	
Search range	POU	
Search target	Specified command.	
Search options	The user can set the search option: “Specified step”.	
	Specified step	When this option is used, it specifies the step number. The search range starts from the specified step number.
Search result	Starts from the beginning of the ladder circuit searching for the specified command and displays the circuit block where the command is found. The cursor moves to the location where the command is used.	

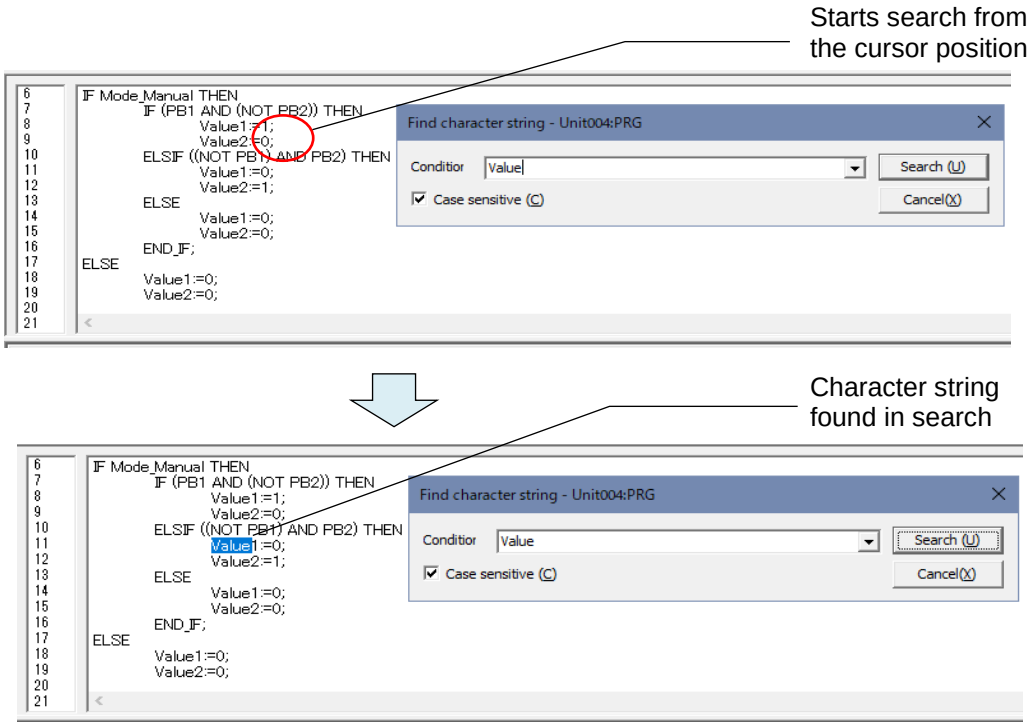
Refer to “Chapter 7 LD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

(NOTE) Refer to “10.11.2 Step number display (LD language)” and “10.11.3 Search results display (LD language)” for details about the ladder circuit display.

10.4.15 Character String Search (ST Language)

This function searches for character strings that are used in the ST program.

Ex: Search for the character string “Value”



The character string search is carried out from the <Find character string> dialog. The <Find by character string> dialog does not close even after conducting a search, and the user can continue doing searches.

Item	Description	
Search range	POU	
Search target	Specified character string	
Search options	The user can set the search option: “Case sensitive”.	
	Case sensitive	When this option is used, the search is conducted and distinguishes between uppercase and lowercase letters in the search.
Search result	Starts from the current cursor position and searches down (line number becomes larger) for a matching character string. A maximum of 64 characters can be specified for the character string search. When the search reaches the last line of the ST program, the search returns to the fist line and continues. Once the search reaches the initial cursor position (start of search), the search is finished.	

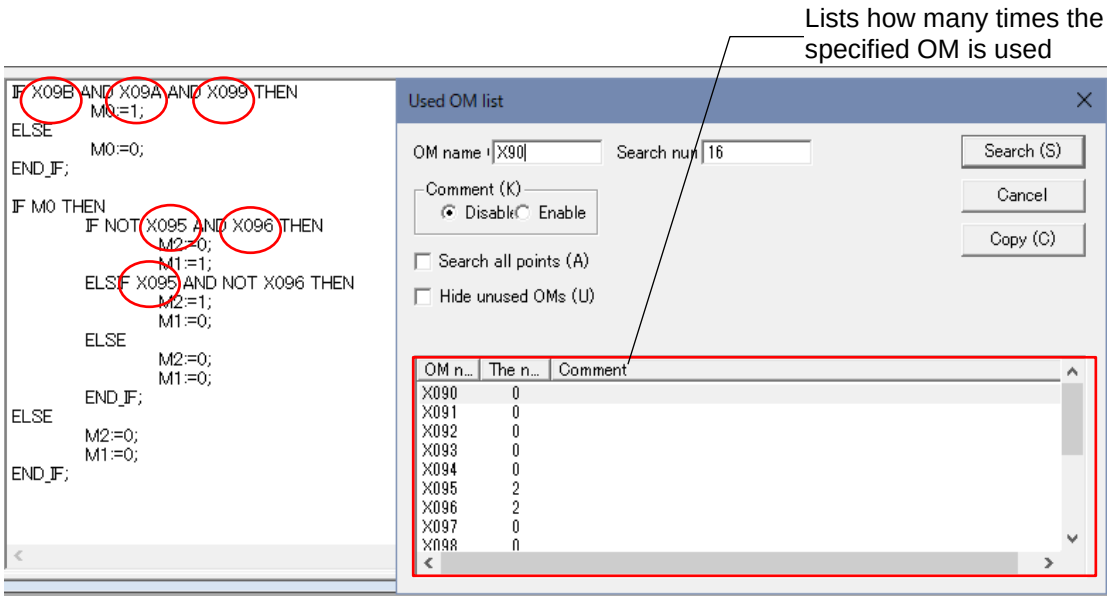
In addition, the user can search up or down again for the character string that was just searched.

Refer to “Chapter 8 ST edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

10.4.16 OM List (ST Language)

This function lists the how many times the specified OM is used in a POU.

Ex: Search how many times the OM is used from X90 to X9F



This search is conducted from the <Used OM list> dialog. The <Used OM list> dialog does not close even after conducting a search, and the user can continue doing searches.

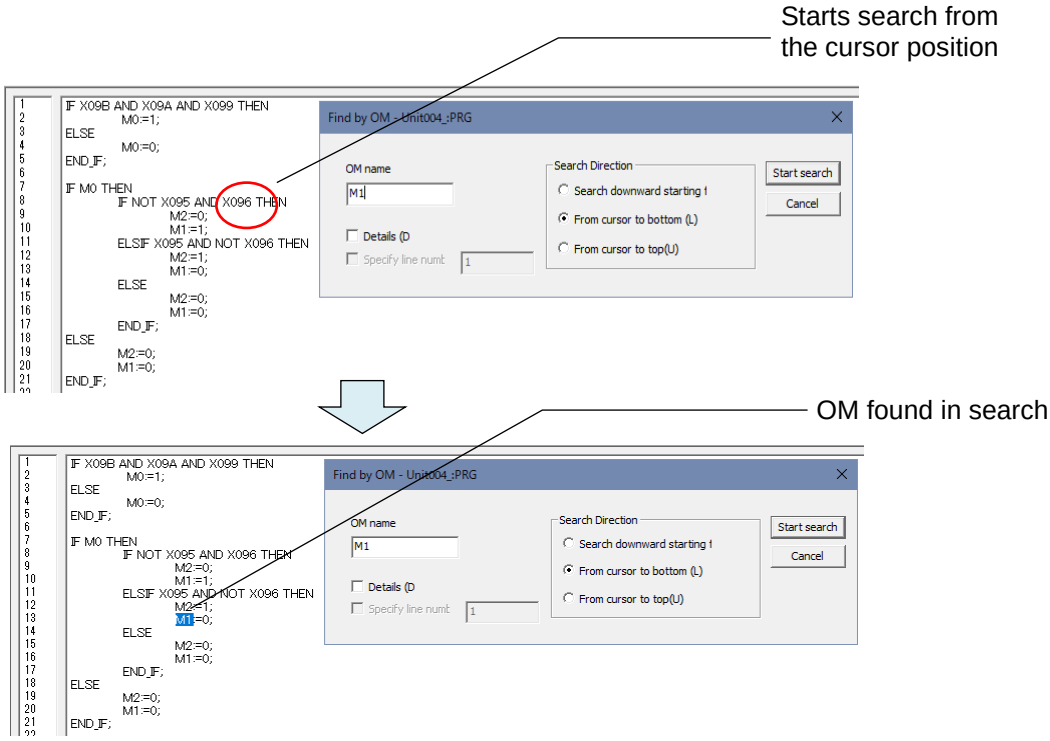
Item	Description	
Search range	POU	
Search target	Specifies the beginning address in the OM search and the search range. For example, if X90 is specified as the beginning address and the range is set to 16, then the addresses X90 to X9F are searched. OMs in a comment are not counted in the search.	
Search result	Displays the OM search results in the <Used OM list> dialog. The following items are displayed.	
	OM NAME	Displays the OM names in the search range. (User can show or hide unused OMs.)
	Quantity being used	Displays how many times an OM is used.
	Comment	Displays the OM comment that is set to the OM in the search. (User can show or hide this comment.)

Refer to “Chapter 8 ST edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

10.4.17 OM Search (ST Language)

This function starts from the current cursor position and moves down searching for the specified OM.

Ex: Search for M1



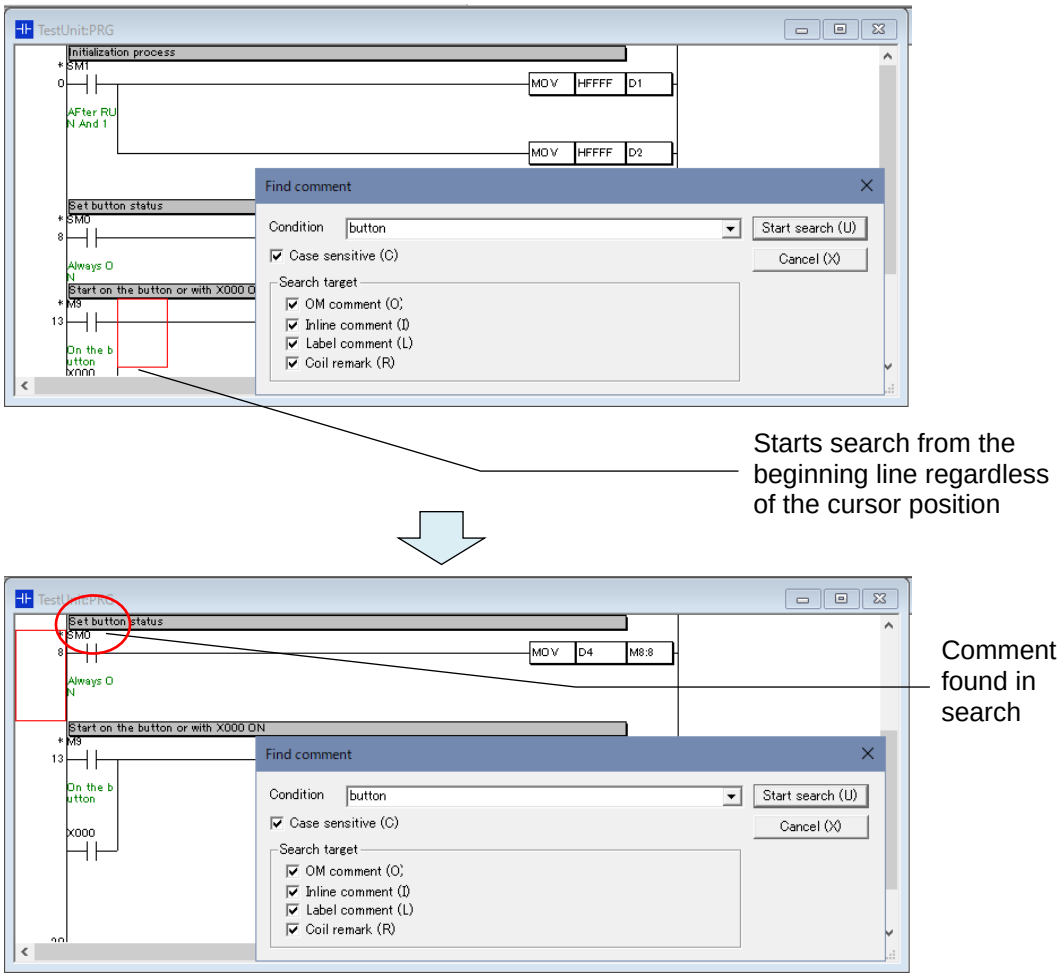
The OM search is carried out from the <Find by OM> dialog. The <Find by OM> dialog does not close even after conducting a search, and the user can continue doing searches. Once the search reaches the last line, the search is finished.

Item	Description	
Search range	POU	
Search target	Specified OM	
Search options	The user can specify the following search options: “Details” or “Specified step”.	
	Details	When this option is specified, the “Digit specified bit OM” or “Longword specified word OM” are also searched. Ex: M17 search when “M16:8” is used in the program. Option not used: M17 is not found in search. Option used: M17 is found in search. M16:8 is displayed as a hit in the search. D1 search when “D0L” is used in the program. Option not used: D1 is not found in search. Option used: D1 is found in search. D0L is displayed as a hit in the search.
	Specified step	When this option is used, it specifies the step number. The search range starts from the specified step number.
Search result	The cursor moves to the location where the OM is used. The search starts from the cursor position and moves down (line number becomes larger) searching for the specified OM.	

Refer to “Chapter 8 ST edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

10.4.18 Comment Search (LD Language)

This function searches for a comment (character string) that is set in the LD program.



Item	Description
Search range	POU
Search target	Searches for a comment where the specified character string is used. The user can select the following comments to conduct a search. • OM comment • Inline comment • Label comment • Coil remark
Search result	The cursor moves to the position where the comment setting is found.

When performing a comment search, if the comment targeted in the search is not shown in the editing window, the display changes automatically to show it. When performing an OM comment search, a 32 character display is used for the OM comment.

The comment search is carried out from the <Find comment> dialog. The <Find comment> dialog does not close even after conducting a search, and the user can continue doing searches. The comment search starts from the beginning line in the ladder circuit (regardless of the cursor position at the start of the search) and continues searching until the END line.

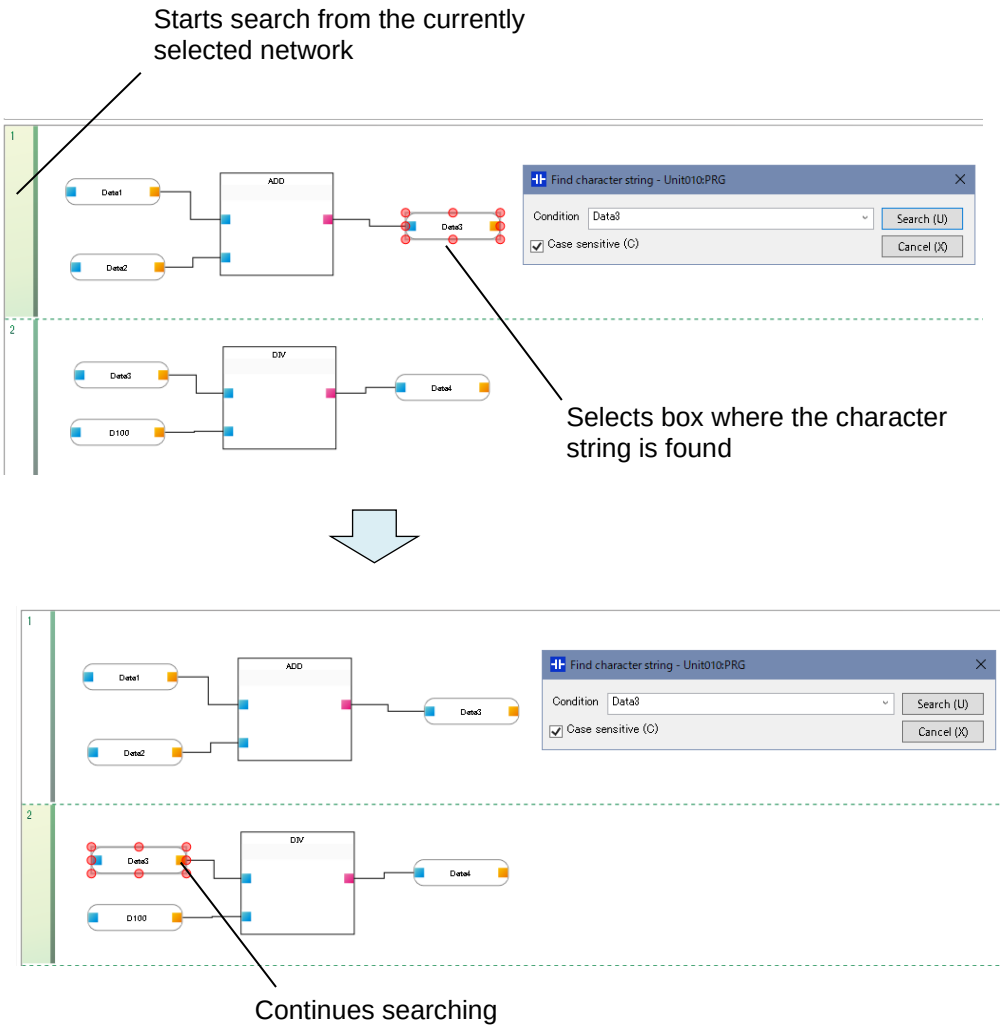
The comment search is carried out in the step number display. If the comment search is started and the search results display is shown, the display will change automatically to the step number display.

Refer to “Chapter 7 LD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

10.4.19 Character String Search (FBD Language)

This function searches for character strings that are used in the FBD program.

Ex: Search for character string “Data3”



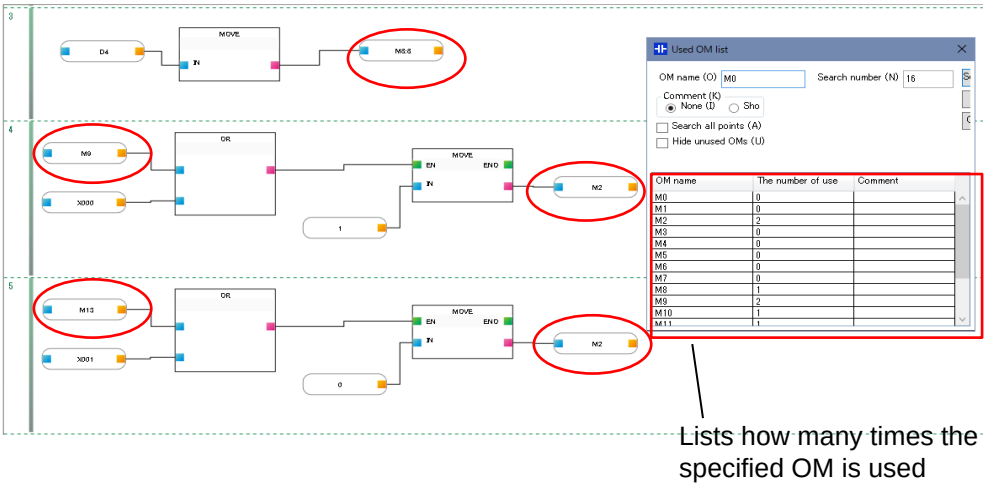
Item	Description	
Search range	POU	
Search target	Searches for a box where the specified character string is used.	
Search options	The user can set the search option: “Case sensitive”.	
	Case sensitive	When this option is used, the search is conducted and distinguishes between uppercase and lowercase letters in the search.
Search result	Starts from the currently selected network and searches down (network number becomes larger) for a matching character string. A maximum of 64 characters can be specified for the character string search. When the search reaches the last line of the FBD program, the search returns to the fist network and continues. Once the search reaches the initial network (start of search), the search is finished.	

The character string search is carried out from the <Find character string> dialog.
Refer to “Chapter 9 FBD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

10.4.20 OM List (FBD Language)

This function lists how many times the specified OM is used in a POU.

Ex: Search how many times the OM is used from M0 to M15



This search is conducted from the <Used OM list> dialog. The <Used OM list> dialog does not close even after conducting a search, and the user can continue doing searches.

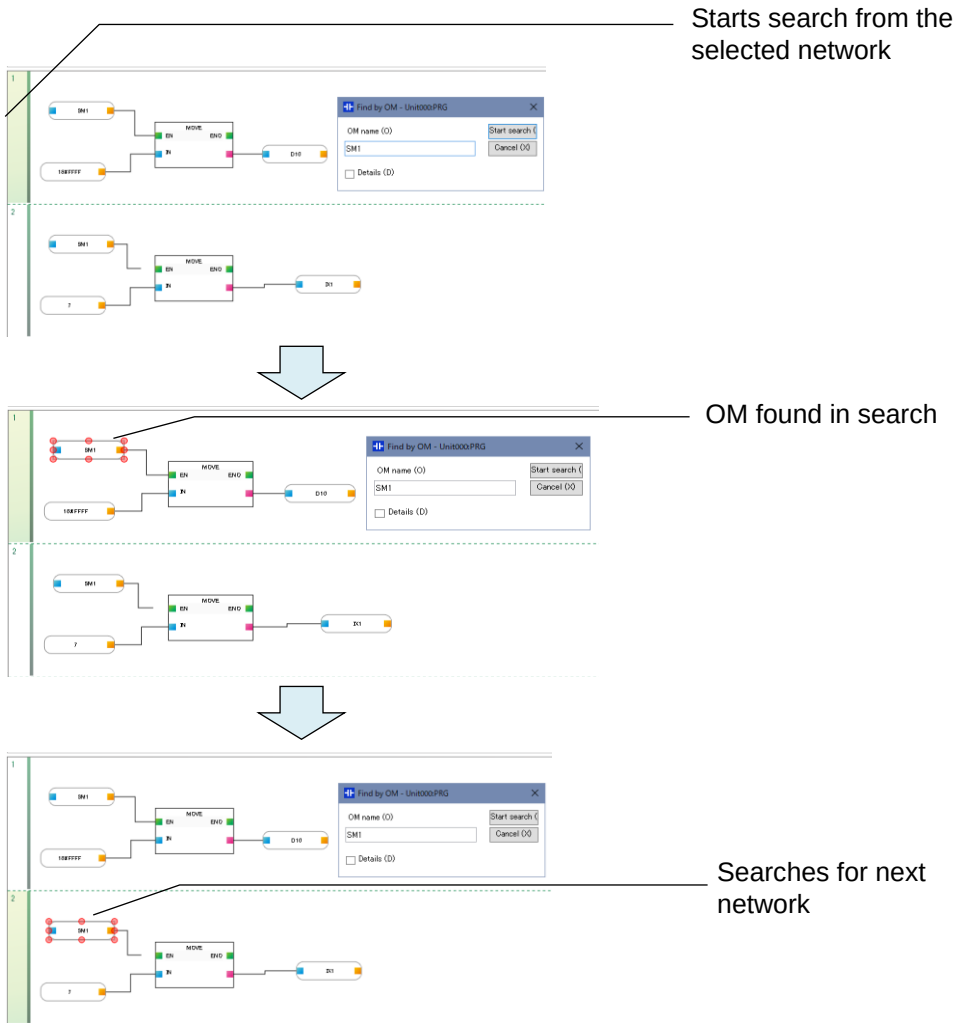
Item	Description	
Search range	POU	
Search target	Specifies the beginning address in the OM search and the search range. For example, if X90 is specified as the beginning address and the range is set to 16, then the addresses X90 to X9F are searched.	
Search result	Displays the OM search results in the <Used OM list> dialog. The following items are displayed.	
	OM NAME	Displays the OM names in the search range. (User can show or hide unused OMs.)
	Quantity being used	Displays how many times an OM is used.
	Comment	Displays the OM comment that is set to the OM in the search. (User can show or hide this remark.)

Refer to “Chapter 9 FBD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

10.4.21 OM Search (FBD Language)

This function searches for the specified OM.

Ex: Search for SM1



The OM search is carried out from the <Find by OM> dialog. The <Find by OM> dialog does not close even after conducting a search, and the user can continue doing searches.

Item	Description	
Search range	POU	
Search target	Specified OM	
Search options	The user can specify the following search options: “Details” or “Specified step”.	
	Details	When this option is specified, the “Digit specified bit OM” or “Longword specified word OM” are also searched. Ex: M17 search when “M16:8” is used in the program. Option not used: M17 is not found in search. Option used: M17 is found in search. M16:8 is displayed as a hit in the search. D1 search when “D0L” is used in the program. Option not used: D1 is not found in search. Option used: D1 is found in search. D0L is displayed as a hit in the search.
	Specified step	When this option is used, it specifies the step number. The search range starts from the specified step number.
Search result	OM found in search becomes selected. The search starts from the currently selected network and moves down (network number becomes larger) searching for the specified OM. When the search reaches the last line, the search returns to the fist network and continues. Once the search reaches the initial network (start of search), the search is finished.	

Refer to “Chapter 9 FBD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

10.5 Replace

This function replaces the OMs, variables or character strings used in an LD or ST program. There are different ways to use this replace function in the PLC Programming software D00 LD/ST Programming Editor.

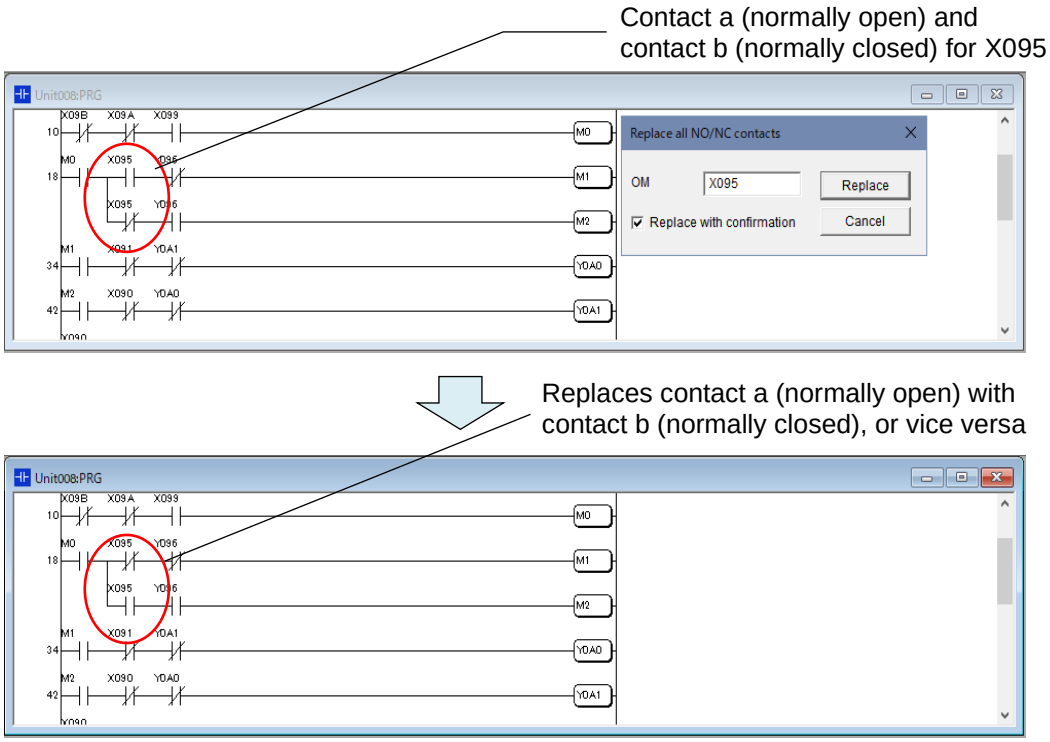
Language	Search method	Description
When editing an LD program	Replace all contacts a/b	Replaces contact *a* (normally open) where the specified OM is used, with contact *b* (normally closed), or vice versa.
	Replace all OMs	Replaces the specified OM with another OM.
	Replace all IXs	Replaces an address number for an OM that is indexed with an address number where the specified index value is set.
When editing an ST program	Replace character string	Replaces the specified character string with another one.
	Replace all OMs	Replaces the specified OM with another OM.
When editing an FBD program	Replace all OMs	Replaces the specified OM with another OM.

(NOTE) Exercise caution when using the replace function, because this function can replace all the items at once throughout the LD or ST program, which may inadvertently replace an item that the user did not intend to replace. In addition, carefully check that there are no unintentional changes made to the LD or ST program logic after the replacement operation.

10.5.1 Replace All Contacts a/b (LD Language)

This function replaces contact *a* (normally open) where the specified OM is used, with contact *b* (normally closed), or vice versa.

Ex: Replace contact *a* (normally open) with contact *b* (normally closed) for X095



(NOTE 1) The replace function for all a / b contacts (normally open/closed) is carried out from the <Replace all contacts a/b> dialog.

(NOTE 2) The replacement operation cannot be carried out when the data is not converted. When the replacement processing is all finished, the data is automatically converted. However, when a replacement is made in a POU that is not registered in the program execution settings, the design check is automatically carried out.

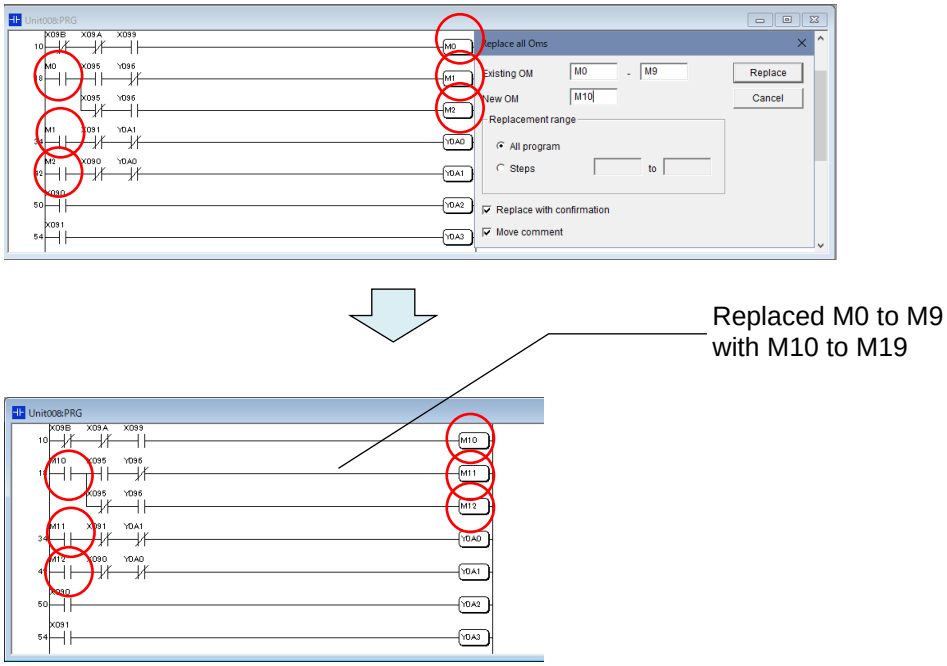
Item	Description
Search range	POU
Replace target	Contact *a* (normally open) and contact *b* (normally closed) where the specified OM is used.
Replace	Replaces contact *a* (normally open) with contact *b* (normally closed) where the specified OM is used, or vice versa.
	The user can replace all the corresponding locations at once, or check and replace each location one at a time.

Refer to “Chapter 7 LD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

10.5.2 Replace All OMs (LD, ST and FBD Languages)

This function replaces an OM being used in a program with another OM.
When replacing the OMs here, the user specifies the replacement range and the new name of the first OM in that range.

Ex: Replace M0 to M9 with M10 to M19 in an LD program



(NOTE 1) The OM replacement is carried out from the <Replace all OMs> dialog. The <Replace all OMs> dialog does not close even after a replacement, and the user can continue doing more replacement operations.

(NOTE 2) The replacement operation cannot be carried out when the data is not converted in the LD language.

(NOTE 3) When the replacement processing is all finished, the data is automatically converted. However, when a replacement is made in a POU that is not registered in the program execution settings, the design check is automatically carried out.

The replacement operation is not carried out in the following situations.

- The replacement operation will not execute if the maximum number of characters will be exceeded after the replacement in an ST program or for a CALC command in an LD program. In this case, the replacement operation is carried out partially (not for the full range) so that the maximum number of characters is not exceeded.
  Ex: When D0 is replaced with D100 in the ST program below and 16380 characters are being used in that program before the replacement:
  Before replacement (semicolon (;) represents the 16380th character)
  D0:=D0+1;
  After replacement
  D100:=D0+1;

  The second D0 cannot be replaced with D100 because the total number of characters in the program will be 16384.
- The replacement operation will not execute if the address number (after replacement) becomes out of range.
  Ex: When an attempt is made to replace D81912 with D8192 for D8191L
- The replacement operation will not execute if the address number for the bit OM (with digit-specified access) is no longer a multiple of 4.
  Ex: When an attempt is made to replace M0 with M1 for M0:4

Item	Description
Search range	POU
Replace target	Specified OM
Replace	Replaces the specified OM being used in a program with another OM. When replacing the OMs here, the user specifies the replacement range and the new name of the first OM in that range. When executing the replacement, the OMs are also replaced where the index or digits are specified.
	The range of the replacement can be set for the entire POU, or it can be set using step numbers, line numbers or network numbers to define the range.
	The OM comment set to the OM before replacement can be transferred to the OM after replacement as well.
	The user can replace all the corresponding locations at once, or check and replace each location one at a time. However, even if the user checks and replaces each OM one at a time, the OMs in one CALC command are replaced all at once.

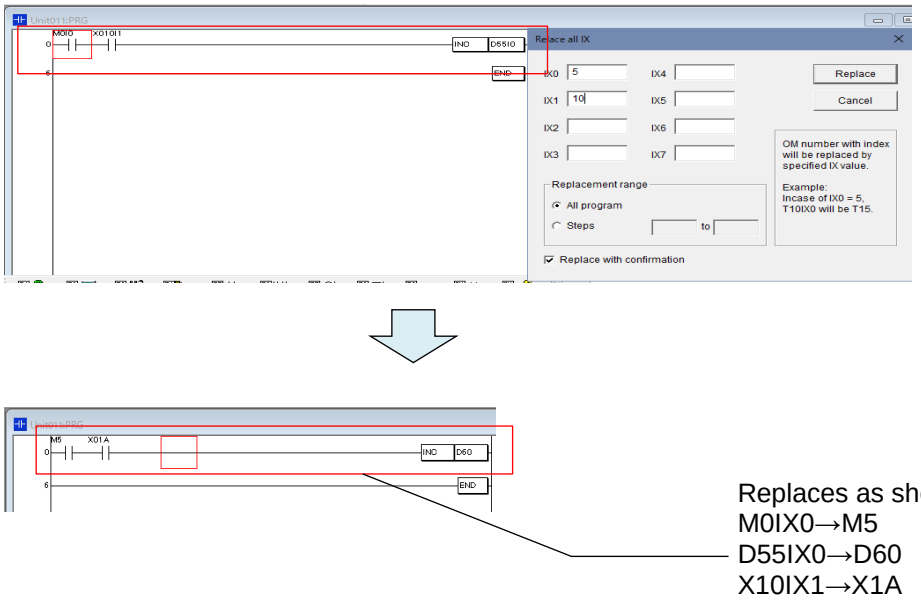
Refer to “ Chapter 7 LD edit operation procedure”, “Chapter 8 ST edit operation procedure” or “Chapter 9 FBD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

10.5.3 Replace All IXs (LD Language)

This function replaces an address number for an OM that is indexed with an address number where the specified index value is set.
When a bit OM (digit-specified access) is indexed, the value equal to specified digits is added to the OM number.

Example

OM before replacement	All replacement IX setting	OM after replacement	Description of OM replacement
M0IX0	IX0=5	M5	5 is added to the address number 0 (before replacement). Therefore, the address number after replacement becomes 5.
D55IX0	IX0=5	D60	5 is added to the address number 55 (before replacement). Therefore, the address number after replacement becomes 60.
X10IX1	IX1=10	X1A	10 (decimal) is added to the address number 10 (hexadecimal before replacement). Therefore, the address number after replacement becomes 1A (hexadecimal).
M0IX2:4	IX2＝1	M4:4	4 (index value 1 × digit number 4) is added to the address number 0 (before replacement). Therefore, the address number after replacement becomes 4.
D25IX3L	IX3＝10	D35L	10 is added to the address number 25 (before replacement). Therefore, the address number after replacement becomes 35.



The all IX replacement function can be a convenient tool for the user, for example, when making repeated changes only to the OM numbers on the same circuit block or when making changes to the OM number from an LD program (as a template).

(NOTE 1) The all IX replacement operation is carried out from the <Replace all IX> dialog.
(NOTE 2) The replacement operation cannot be carried out when the data is not converted.
(NOTE 3) When the replacement processing is all finished, the data is automatically converted. However, when a replacement is made in a POU that is not registered in the program execution settings, the design check is automatically carried out.

The replacement operation is possible if the maximum number of characters is not exceeded after the replacement for a CALC command in an LD program. Replacement is not possible when the maximum number of characters is exceeded. 1024 characters can be entered for a CALC command.
The replacement operation will not execute if the address number (after replacement) becomes out of range.

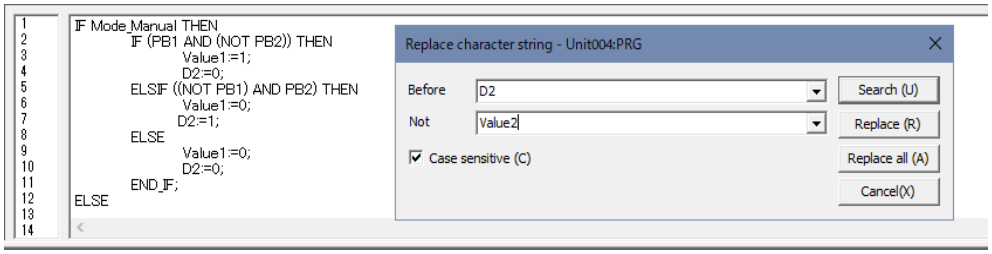
Item	Description
Replace range	POU
Replace target	OM where the specified index is set.
Replace	Replaces the specified OM being used in an LD program with another OM.
	The range of the replacement can be set for all POUs, or it can be set using step numbers to define the range.
	The user can replace all the corresponding locations at once, or check and replace each location one at a time. However, even if the user checks and replaces each OM one at a time, the OMs in one CALC command are replaced all at once.

Refer to “Chapter 7 LD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

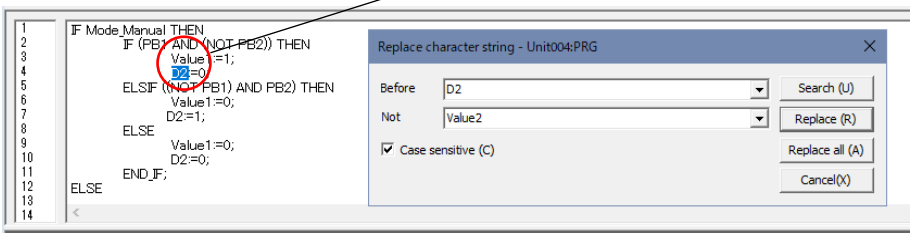
10.5.4 Replace Character String (ST Language)

The user can replace a character string being used in an ST program with another character string.

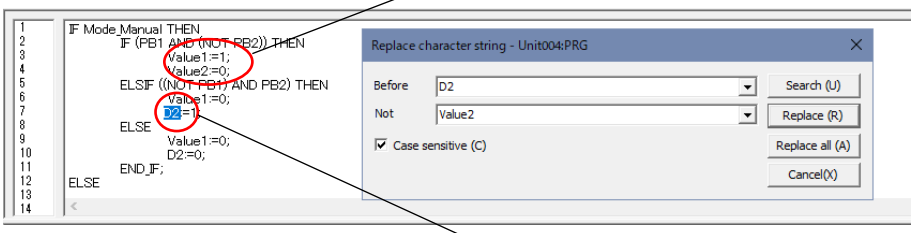
Ex: Replace “D2” with “Value2”



Searches for character string to replace



Replaces “D2” with “Value2”



Searches for next character string to replace

The character string replacement is carried out from the <Replace> dialog. The <Replace> dialog does not close even after a replacement, and the user can continue doing more replacement operations.
The replacement starts from the cursor position and moves down (line number becomes larger) searching for the replacement. When the replacement search reaches the last line, it continues the replacement search from the first line. Thereafter, once the replacement search reaches the initial cursor position (start position), the replacement operation is finished.

Special notes when replacing character strings

- The character string replacement does not replace comments or variable names in the variable editing area.
- When the character string specified in the search (text before replacement) is found within the name of an item, then that text is also replaced. For example, when replacing the OM name “M1” with the OM name “X2”, if the variable name “NUM1” is being used in the ST program, it is replaced with “NUM2”. Therefore, be careful not to make unnecessary replacements.
- If the ST program becomes invalid after a replacement operation, correct the error manually.
- The replacement operation is possible if the maximum number of characters is not exceeded after the replacement in an ST program. Replacement is not possible when the maximum number of characters is exceeded. 16383 characters can be entered into an ST program.
- When replacing a character string for an OM, make sure that the OM can be still be used and is valid with the new name (after the replacement).

Item	Description	
Replace range	POU (excluding variable editing area)	
Replace target	Specified character string.	
Search options	When searching for a character string to replace, the following option can be set.	
	Case sensitive	When this option is used, the search is conducted and distinguishes between uppercase and lowercase letters in the search.
Replace	Replaces a character string being used in the program with the specified character string.	
	The character strings in the search can be replaced one at a time or all at once.	
	The user can specify whether or not to distinguish between uppercase and lowercase letters when searching and replacing character strings.	

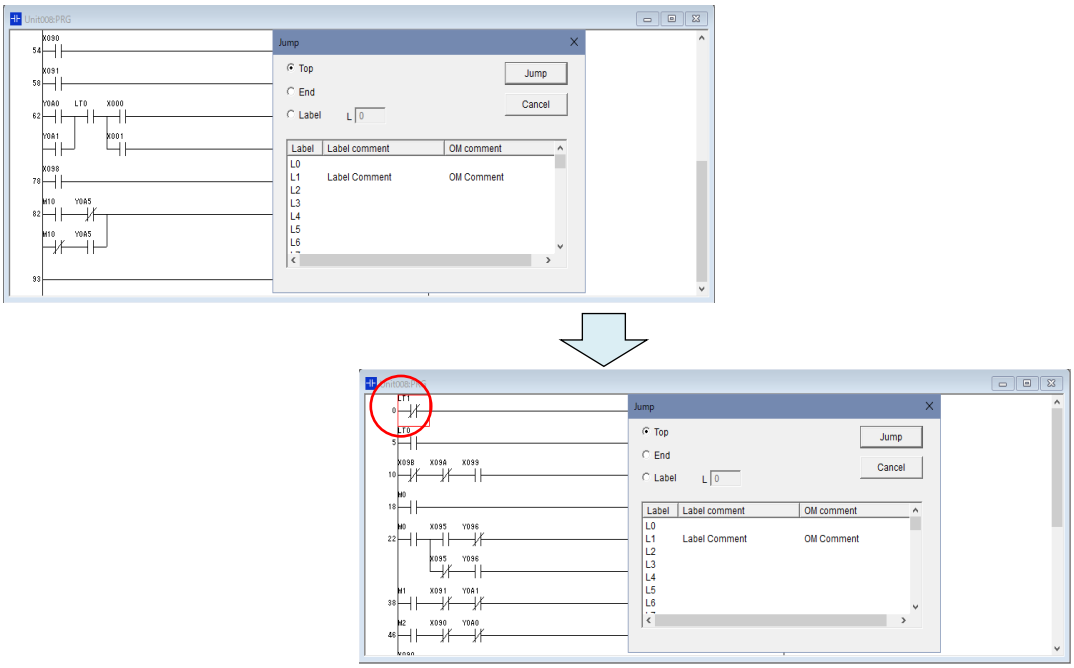
Refer to “Chapter 8 ST edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

10.6 Jump

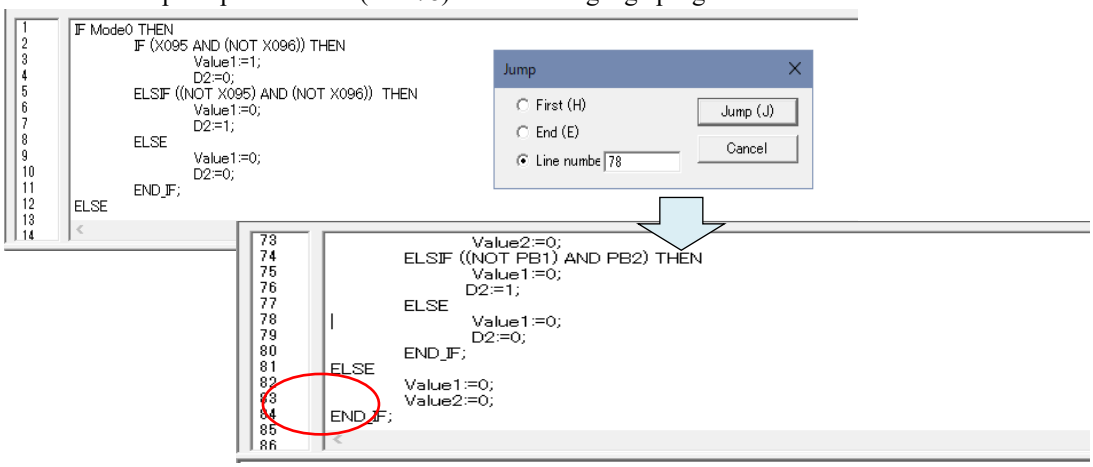
The user can jump or move the cursor to a specified location in the LD, ST or FBD program currently being edited in the PLC Programming software D00 LD/ST Programming Editor and FBD Programming Editor.
The possible jump locations vary depending on which language is being used to edit the program.

Language	Possible jump locations
LD language	First step, last step or a circuit block where a specific label is set
ST language	First line, last line or specified line number
FBD language	First network, last network or specified network number

Ex: Jump to first step in an LD language program

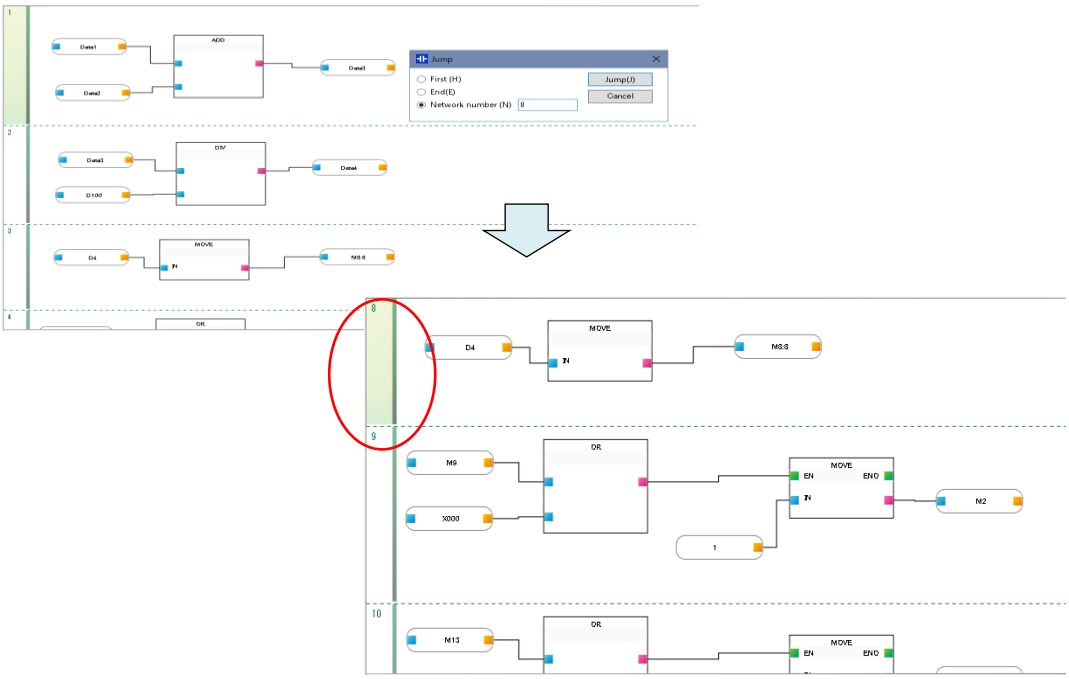


Ex: Jump to specified line (line 78) in an ST language program



10

Ex: Jump to specified network (network number 8) in an FBD language program



The jump function is carried out from the <Jump> dialog.
In an LD program, the <Jump> dialog does not close even after a jump, and the user can continue doing more jump operations.
Another POU cannot be specified as the jump location.

Refer to “Chapter 7 LD edit operation procedure”, “Chapter 8 ST edit operation procedure” or “Chapter 9 FBD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

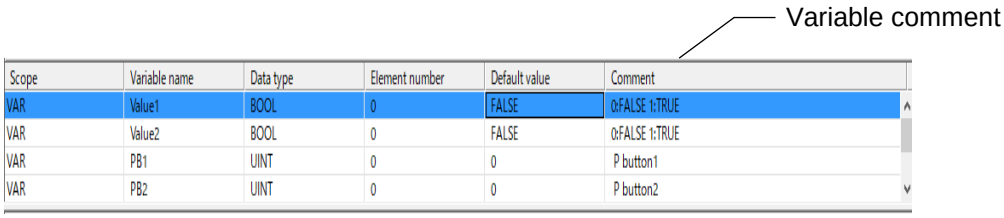
10.7 Create Comments & Drawings

This function sets comments for OMs, variables, and LD or ST programs.

Variable comment

The user can set a comment in an ST or FBD program and in a CALC command for an LD program.

Ex:

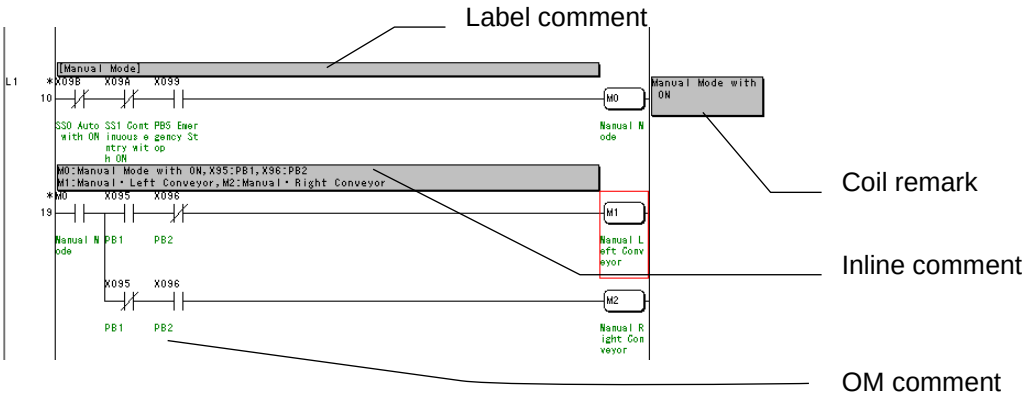


Type of comment	Description
Comment	This comment is set to a variable. The user can set a maximum of 32 characters.

LD program comment

OM comments, inline comments, label comments or coil remarks (excluding variables) can be set while editing an LD program.

Ex:



Type of comment	Description
OM comment	This comment is set to an OM. There is a 32 character (half-width) limit for one OM comment. (One full-width character is counted as two characters.) Once an OM comment is set to an OM, the content is the same throughout the project. However, an OM comment cannot be set for indexes (IX) and nesting (N).
Inline comment	The user can edit a comment between circuit blocks. There is a 64 character (half-width) limit per line, with a maximum of 5 lines, for one inline comment. (One full-width character is counted as two characters.) For example, this comment can be added to check whether processing has been carried out or not.
Label comment	This comment is set to a circuit label. The comment can only bet set for a circuit where a label has been set. There is a 64 character (half-width) limit for one label comment. (One full-width character is counted as two characters.) This comment can be added, for example, as a delimiter in a program.

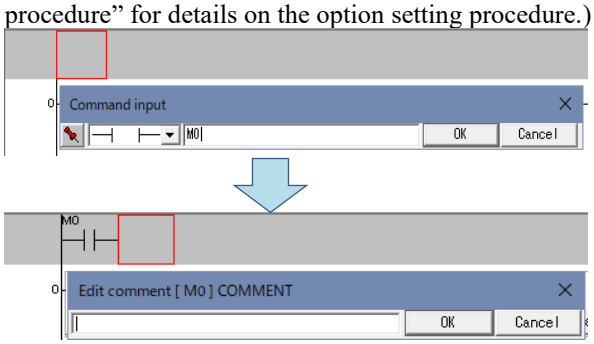
Type of comment	Description
Coil remark	This comment is set to an output command and application command. There is a 32 character (half-width) limit for one coil remark. (One full-width character is counted as two characters.) For example, this comment can be added to check what type of processing has been carried out in the output command.

(NOTE 1) "Create drawing" refers to a function that creates a comment when editing in the LD language.

(NOTE 2) The comment on the ladder circuit can be shown or hidden. Refer to "10.11.1 Display comments (LD language)" for details on the comment display function.

(NOTE 3) An OM comment is shared throughout a project, but a label (L) shows the position in the POU. Therefore, only an OM comment for a label is set per POU.

(NOTE 4) When the option setting "Perform continuous input for comment" is enabled in the PLC Programming software D00, after the command input, the user can enter an OM comment for the OM being used in that command. However, this option is used with OMs where an OM comment can be set. (Refer to "Chapter 7 LD edit operation procedure" for details on the option setting procedure.)



After the command input dialog, the comment input dialog is displayed.
If a digit-specified OM is included in one command or if a command with two or more arguments is edited, the OM command input applies to all OMs included in that command.

<u>ST program comment</u>

When entering a comment into an ST program, the character string for the comment is enclosed by (* *).

Example

```
(* MODE_MANUAL:Manual Mode with ON
   GO_LEFT:AUTO・Left Conveyor,GO_RIGHT:AUTO・Right Conveyor *)
IF Mode_Manual THEN
            IF PB1 AND NOT PB2 THEN
                    GO_RIGHT:=0;
                    GO_LEFT:=1;
            ELSIF NOT PB1 AND PB2 THEN
                    GO_LEFT:=0;
                    GO_RIGHT:=1;
```

Comment

Type of comment	Description
Comment	When entering a comment into an ST program, the character string for the comment is enclosed by (* *). As long as the ST program character limit is not exceeded, there are no limits to the number of characters in a comment or the number of comments used.

Refer to "Chapter 8 ST edit operation procedure" for details on each comment setting procedure.

10.8 Edit Lock

In order to prevent inadvertent or mistaken changes in the PLC Programming software D00 EDITOR, the user can change to a mode where editing is prohibited or locked and only displays the program.
This mode is called edit lock.
The following operations are not possible while editing is locked.

Item	Description
Unavailable functions while editing is locked	Program edit
	Add, change and delete variables
	Change timer/counter values (when editing in LD)
	Create drawings (when editing in LD)

Editing is possible after opening a POU in the PLC Programming software D00 EDITOR.
When multiple POUs are opened in PLC Programming software D00 EDITOR, the edit lock is set for each POU. If multiple editing windows are opened for the same POU, when the editing is locked in one of the windows, then editing for all of the windows become locked.
While the PLC Programming software D00 EDITOR is in monitoring mode, the program cannot be edited.

Refer to “Chapter 7 LD edit operation procedure”, “Chapter 8 ST edit operation procedure” or “Chapter 9 FBD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

10.9 Program Conversion Function

10.9.1 Conversion

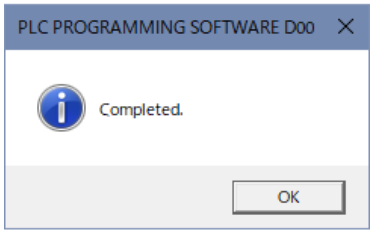
Conversion refers to converting the programs (created in a language in the PLC Programming software D00 EDITOR) to code that allows execution in the NC(BUILT IN PLC).
POUs that run in the NC(BUILT IN PLC) need to be converted before writing a program to the NC(BUILT IN PLC), regardless of the language (LD, ST or FBD) that is used.
Conversion is carried out in the PLC Programming software D00.

This conversion is for high speed or standard control programs that execute in the NC(BUILT IN PLC) and the user or system defined functions that are called by those high speed or standard control programs. (High speed and standard control programs that execute are set in the <Set program execution> dialog in the PLC Programming software D00 EDITOR.)

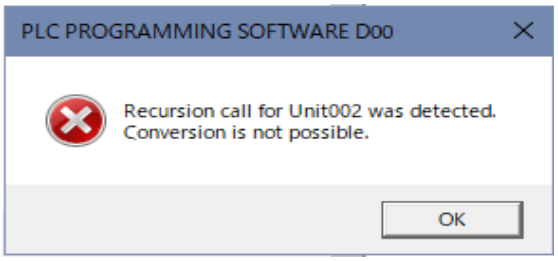
If the design check (Refer to 10.10.2) is carried out before conversion and an invalid program is created, then a conversion error is detected and the conversion will be cancelled. The invalid content is different in each program language. Refer to the “PLC Programming Software D00 Programming Manual” for further details.

The following is an example of program conversion results.

- When there is no conversion error, design check error or warning

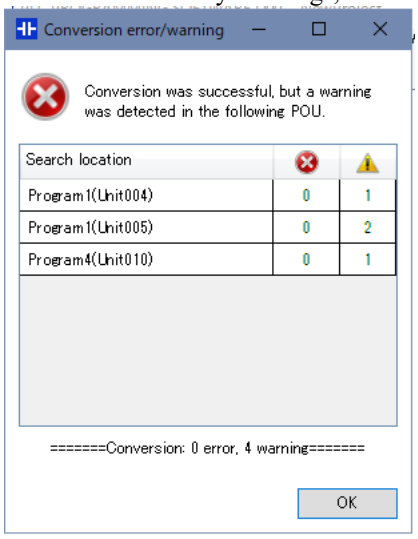


- When there is a conversion error

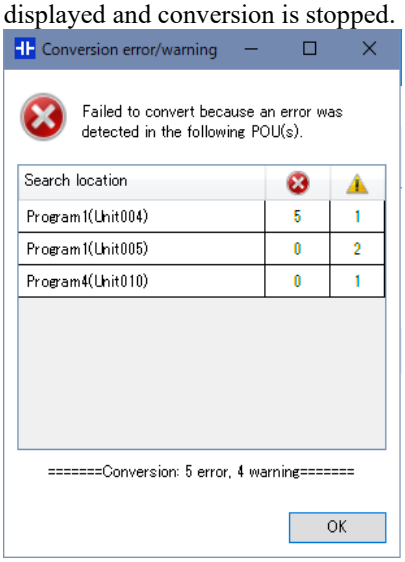


Even if there is no error in the design check, a conversion error is triggered in the following situations.

- When a POU is not set in the <Set program execution> dialog
- When a recursive call for a function is executed (function itself called in that function)
- When there are only warnings found in the design check
  When there are only warnings found in the design check (before conversion), the number of warnings is displayed.
  When there are only warnings, the conversion is still executed.



- When there are only warnings found in the design check
  When there are errors found in the design check (before conversion), the number of errors is displayed and conversion is stopped.



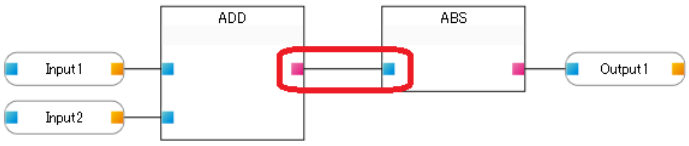
After double-clicking on a POU in the <Conversion error/warning>, the edit screen for that POU is displayed in PLC Programming software D00 EDITOR.
Check the design check results displayed in each edit screen, and correct any errors or warnings.

FBD program conversion

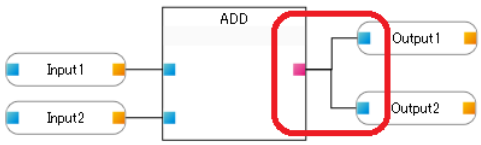
When converting an FBD program, an ST program is automatically created using the same logic as the FBD program.

At this time, variables may be added automatically.

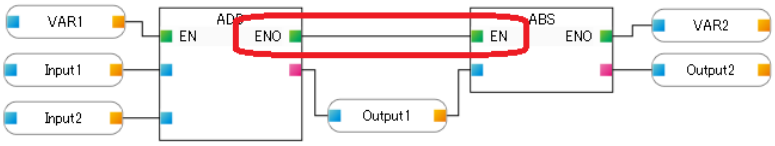
- When connecting a function box output directly to the input of another function box
  Ex:



- When connecting a function box output to multiple variable or OM boxes
  Ex:



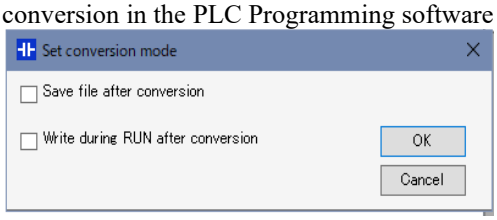
- When connecting a function box ENO directly to the EN of another function box
  Ex:



There is an upper limit on the number of variables that can registered to one POU. A check is carried out at conversion to make sure that the number of registered variables does not exceed that upper limit. At this time, the total number of variables automatically added and registered by the user is checked.

10.9.2 Conversion Mode Settings

The conversion mode settings are used to set the operation that automatically executes after conversion in the PLC Programming software D00 EDITOR.



The user can set the following operations after conversion is executed.

Item	Description
Save project after conversion	The user can select whether or not to automatically save the whole project after executing the conversion operation.
Write during run after conversion	The user can select whether or not to automatically write the program (while the program is running) after executing the conversion operation.

The conversion mode settings are common settings in the PLC Programming software D00 EDITOR regardless of the project.

(NOTE 1) If the POU data to be written is not converted, then the data is automatically converted. However, even if the checkbox “Write during run after conversion” is ticked, the write during run operation will not execute right away. A conversion confirmation message is displayed. After starting the conversion, if there is no conversion error, then the write during run operation is executed.

(NOTE 2) When a “Design check” is carried out, this setting does not apply. Refer to 10.10.2 for details about the design check.

(NOTE 3) Refer to “11.1 Write & Read Programs” for details on the write during run operation.

Refer to “5.9.2 Conversion Mode Settings” for details on the settings.

10.10 Program Check Functions

In addition to the conversion function, the PLC Programming software D00 EDITOR is also equipped with other functions that check the program and display warnings.

10.10.1 Double Coil Check

The double coil check refers checking whether there are multiple locations in the program being written for the same OM. The PLC Programming software D00 EDITOR has a function that checks whether there is a double coil.

(NOTE) In the PLC Programming software D00, executing a program with a double coil is prohibited.
Note, a program could malfunction depending on the program content. Therefore, the user should check the program content carefully in the event that a program with a double coil is executed in the NC(BUILT IN PLC).

The double coil check is executed for the following items.

Check item	Description
Double coil check for LD program	Displays a warning for the POU being edited, identifying an OM that is already being used when an OUT command (including local timers and local counters) is input.
Double coil check for all POUs	Checks all POUs being written to the NC(BUILT IN PLC). If there is a double coil that is detected, a list is displayed showing all of the locations. This check is performed when accessing the <Double coil check> menu under <Edit> in the PLC Programming software D00, or when executing the program write operation*.

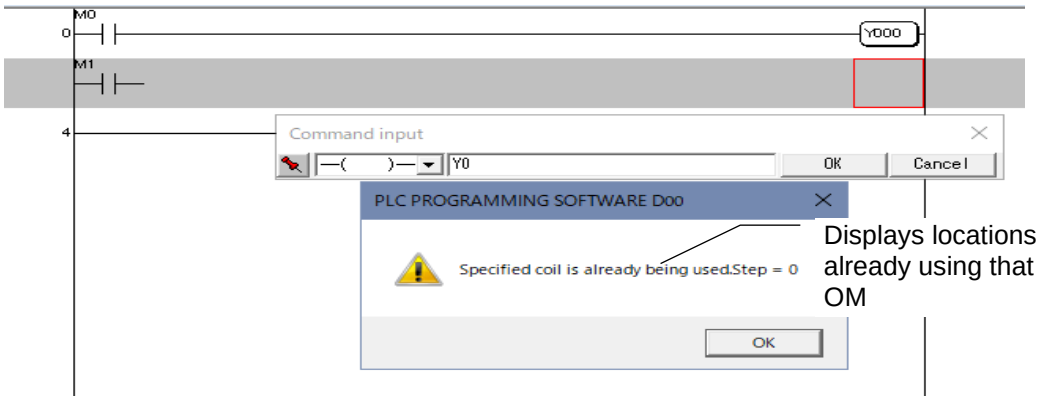
* Including the write during run operation. Refer to “11.1 Write & Read Programs” for details on the program write and write during run operations.

(NOTE) Refer to “11.1 Write & Read Programs” for details on the program write and write during run operations.

<u>Double coil check for LD program</u>
When an OUT command (including local timers and local counters) is input, if an OM is set that is already used in the output, then a warning is displayed.
Correct the OM when necessary.

(NOTE 1) OMs being used in other commands (besides OUT commands) are not checked.
(NOTE 2) When an OM with an index is used in an OUT command, then that OM (with the same content) is also targeted in the check.
Ex: When “OUT Y0IX0” is used at multiple locations, that is targeted in the check. “OUT Y0IX0” and “OUT Y0IX1” are not targeted in the check because another OM is being used.

Ex: Y0 that is already used in an OUT command is reused in an OUT command



When the same OM is used in multiple OUT commands, the lowest step number where it is used is displayed. If the OM is being used on a circuit block that is unconverted, then "?" is displayed at the locations where it is used.

The double coil check is carried out when the user clicks on the <OK> button in the <Command input> dialog. The check is not carried out at other times, such as for a circuit paste operation or for a conversion operation.

Double coil check for all POUs

The double coil check is carried out for all POUs, which are set in the "Set program execution". This also includes user and system defined functions called from the POUs.
In this situation, a double coil check is carried out for the following.

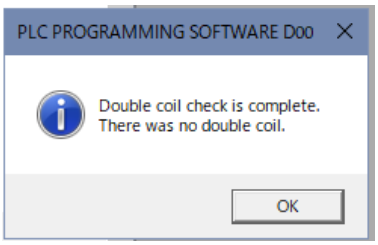
POU	Description
LD program	Corresponding OMs :Y, BY (*1) Corresponding commands : OUT, SET, RST, PLS, PLSN and SFT(P) commands, application command destination and ST language processing in CALC commands
ST program FBD program	Corresponding OMs : Y, BY(*1) Corresponding command : Assignment, function output argument and function output/input argument

*1 OMs with an index are not differentiated. For example, a double coil is detected for "OUT Y0IX0" and "OUT Y0IX1".

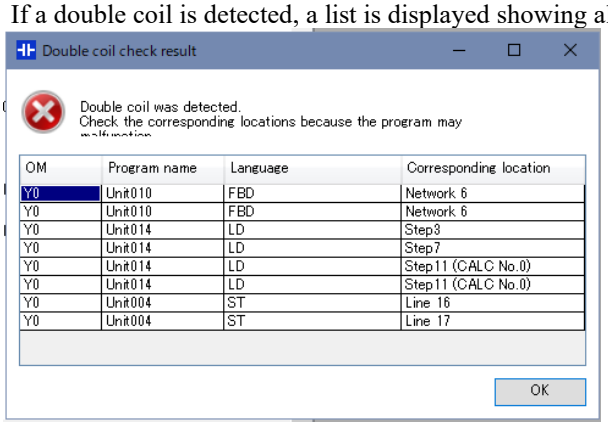
(NOTE) Refer to "Chapter 11 Commands (LD)" or "Chapter 12 Commands (ST)" in the PLC System Manual, or "Chapter 12 Commands (FBD)" in this manual for details on commands in each language.

The check results are displayed as shown below.

- When a double coil is not detected



- When a double coil is detected
  If a double coil is detected, a list is displayed showing all of the locations.



OM	Program name	Language	Corresponding location
Y0	Unit010	FBD	Network 6
Y0	Unit010	FBD	Network 6
Y0	Unit014	LD	Step3
Y0	Unit014	LD	Step7
Y0	Unit014	LD	Step11 (CALC No.0)
Y0	Unit014	LD	Step11 (CALC No.0)
Y0	Unit004	ST	Line 16
Y0	Unit004	ST	Line 17

The <Double coil check result> displays the OM with the double coil, the POU, the language and the double coil location.
After double-clicking on a line displayed in the <Double coil check result>, it jumps to that location.
A maximum of 1023 double coil locations can be displayed in the <Double coil check result>. If there are 1024 double coils or more, then any data on those coils (that exceed the maximum) is not displayed.

This double coil check is executed at the following times.

- Executes when the <Double coil check> menu is selected under <Edit> in the PLC Programming software D00
- Executes automatically when the program write operation is executed or when the write during run operation is executed

Refer to “5.7.1 Double Coil Check” for details on how to use PLC Programming software D00 EDITOR.

10.10.2 Design Check

The design check function is for making sure there are no invalid notations in the POU that was created.
The design check is executed for the POU that is displayed at the forefront in the PLC Programming software D00 LD/ST Programming Editor and FBD Programming Editor.
However, when a conversion is executed, the design check is carried out for the user/system defined functions and high speed/standard control programs set in the <Set program execution> dialog.

Timing of check

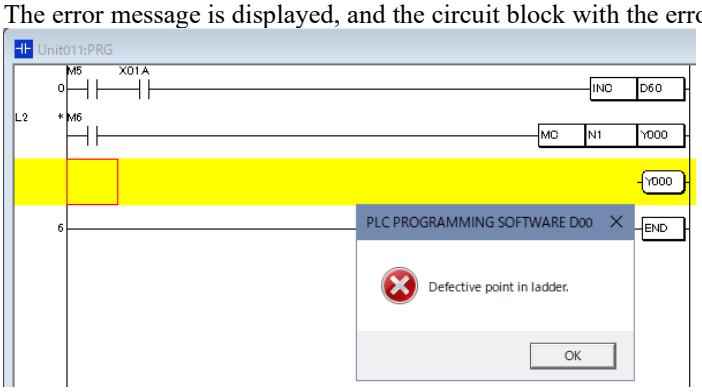
This design check is executed at the following times.

- When executed from the <Design check> menu in the PLC Programming software D00 LD/ST Programming Editor and FBD Programming Editor
- When conversion is carried out
- When one of the following operations in performed while editing a POU in the LD language
- When the **[End]** key is pressed
- When the **[Shift]** + **[End]** keys are pressed
- When replacement is executed
- When all set values are changed using the <Timer/Counter list>

Display of check results

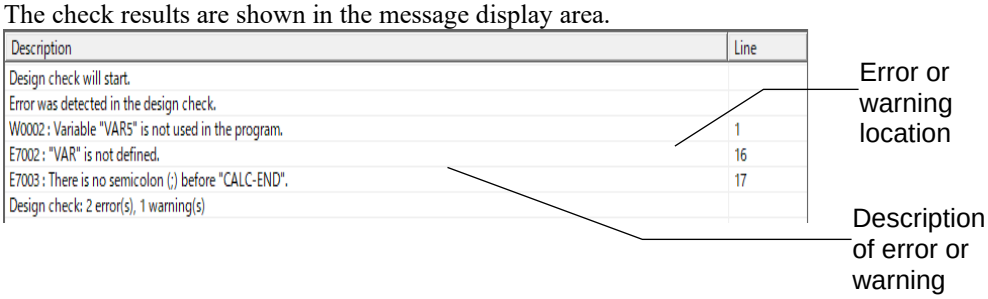
When the design check is carried out and an error or a warning is detected, an error message is displayed, or an error or warning description is displayed in the message display area for the PLC Programming software D00 LD/ST Programming Editor or FBD Programming Editor. The invalid content is different in each program language. Refer to “Chapter 11 Commands (LD)” or “Chapter 12 Commands (ST)” in the PLC System Manual, or “Chapter 12 Commands (FBD)” in this manual for further details.
In addition, the error or warning display method varies depending on the language.

- When using the LD language
  The error message is displayed, and the circuit block with the error is highlighted in yellow.



After clicking on the <OK> button in the <CALCBOX> dialog, the design check is carried out for the CALC command. Refer to “When using the ST/FBD language” (described below) for further details.

- When using the ST/FBD language
  The check results are shown in the message display area.

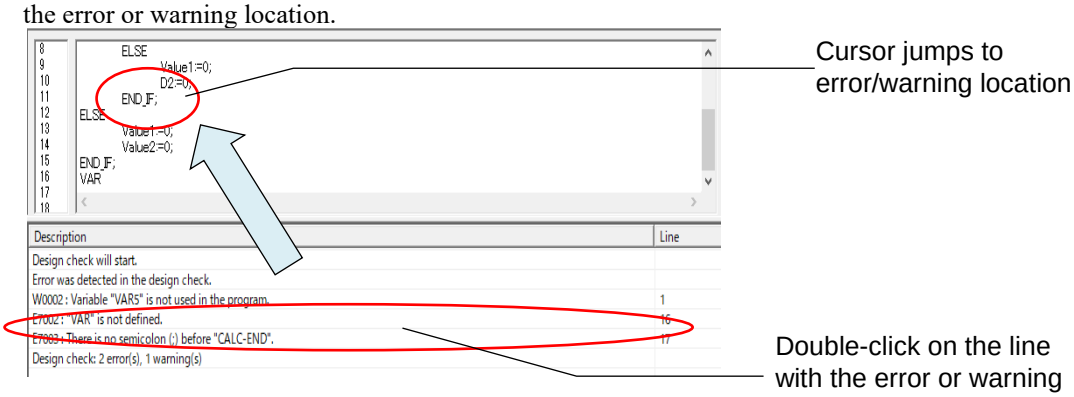


If the description begins with an “E” such as “Exxxx”, then it is an error. Check the message description and correct the error(s). Otherwise, the program will be invalid.
If the description begins with an “W” such as “Wxxxx”, then it is a warning. Check the message and only correct the program when necessary, since the program syntax is correct.
Refer to “Chapter 12 Commands (ST)” in the PLC System Manual or “Chapter 12 Commands (FBD)” for details on errors and warnings shown in the message display area.

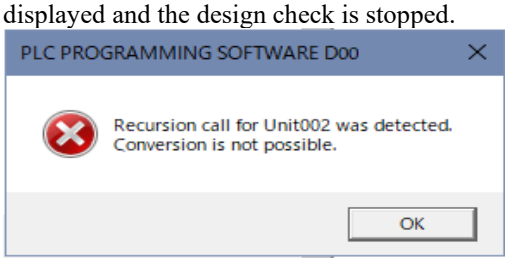
The following information is displayed for the error and warning locations.

- When a registered variable has an error or warning: Line number in variable editing area
- When an ST program has an error or warning: Line number in ST program
- When an FBD program has an error or warning: Network number in FBD program

After double-clicking on the line with the error or warning in the message display area, it jumps to the error or warning location.



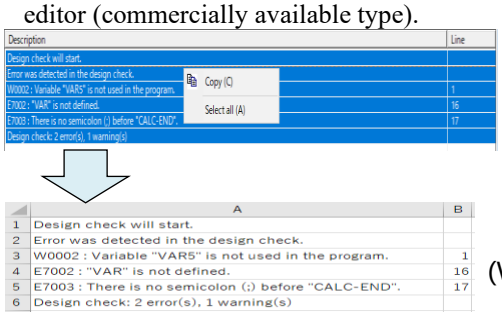
If a recursive call for the user or system defined function is executed, the following message is displayed and the design check is stopped.



(NOTE 1) The following is not detected in the design check. However, it is detected as an error when conversion is executed.

- Number of variables used exceeds the upper limit (when checking all POUs that execute in the NC(BUILT IN PLC))
  Refer to “10.9 Program Conversion Function” for further details about the conversion.

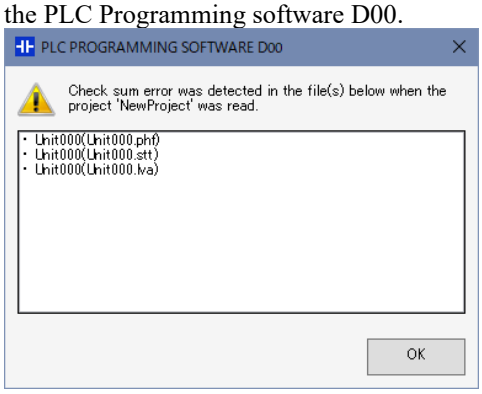
(NOTE 2) The user can copy the text data in the message display area and paste it to a text editor (commercially available type).

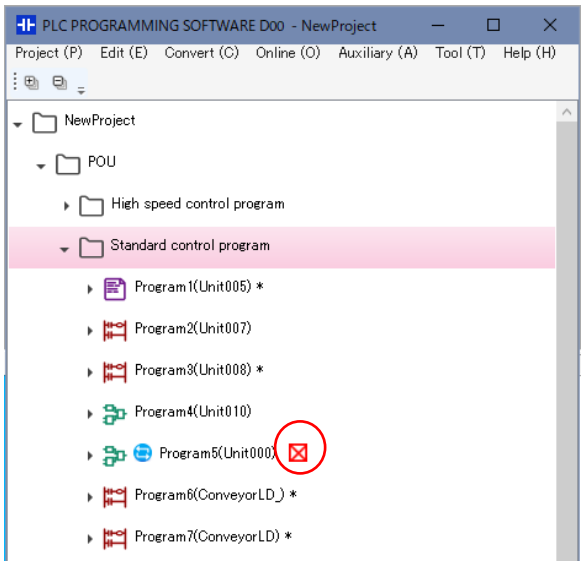


(When pasting to Microsoft Excel)

10.10.3 File Checksum Function

In order to confirm whether or not project data has been corrupted or altered while using the PLC Programming software D00 EDITOR, a checksum is used when the project is opened to verify the integrity of the files that contain project data.
When a checksum error has been detected, a data list is displayed and an "×" symbol appears in the PLC Programming software D00.





The checksum verification executes in the following operations.

- Open project
- Read program
- Write program

A program that contains a checksum error cannot be written to the NC(BUILT IN PLC).
Check and correct the program and settings where the checksum error was detected, and then save the data.

(NOTE) Even when there are differences in the program content and/or settings, the checksum values may still match. When writing the program, be sure to check the data before executing the writing operation.

10.11 Display Function

This function changes the display of the PLC Programming software D00 EDITOR.
The user can change to the following displays.

Language	Item	Description
LD program	Show/Hide comments	The user can show or hide the display for OM comments, inline comments, label comments and coil remarks. In addition, the user can change the number of characters in the display to 16 characters or 32 characters for an OM comment.
	Step number display	Displays the LD program based on the step number order.
	Search results display	Changes the LD program to show the search results. The search results display shows a search history. The display is based on order of the search, and the most recent search result is displayed on the bottom.
	Change monitoring base number	The user can change the base number between decimals and hexadecimals for the OM and variable values displayed when monitoring in the NC(BUILT IN PLC).
ST program	Change monitoring base number	The user can change the base number between decimals and hexadecimals for the OM and variable values displayed when monitoring in the NC(BUILT IN PLC).
FBD program	Change monitoring base number	The user can change the base number between decimals and hexadecimals for the OM and variable values displayed when monitoring in the NC(BUILT IN PLC).

(NOTE 1) “Change monitoring base number” is a function used while in monitoring mode in the PLC Programming software D00 EDITOR. Refer to “11.4 Monitoring function during program operation” for details on the monitoring mode.

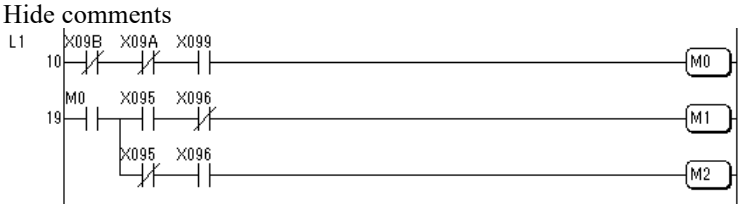
(NOTE 2) The display font can be changed in the <Option setting> in the PLC Programming software D00 for the PLC Programming software D00 LD/ST Programming Editor and FBD Programming Editor.

10.11.1 Display Comments (LD Language)

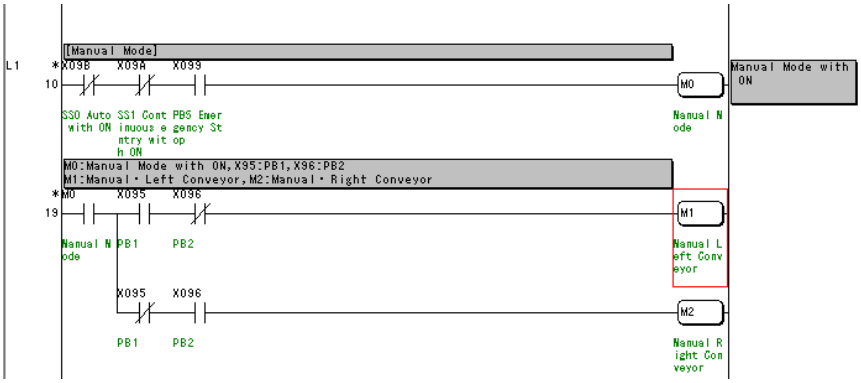
When the ladder circuit is displayed in the PLC Programming software D00 EDITOR, the user can specify whether to show or hide the OM comments, inline comments, label comments or coil remarks.

Ex:

Hide comments



Show comments



In addition, when displaying the OM comments, the user can change the number of characters in the display to 16 characters or 32 characters.

Ex: 16 character display 32 character display

However, when using full-width characters in an OM comment, a single full-width character takes up the space of two half-width characters. Therefore, when using both half-width and full-width characters at the same time, the number of characters that can be displayed becomes less.

Ex: OM comment using full-width characters in a 16 character display

Refer to “Chapter 7 LD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

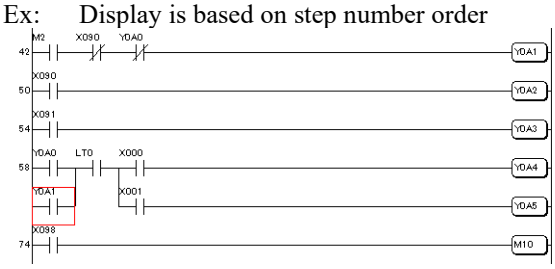
Refer to “10.7 Create comments & drawings” for details on each comment.

10

10.11.2 Step Number Display (LD Language)

When the ladder circuit is displayed in the PLC Programming software D00 EDITOR, the display is based on the step number order.

Ex: Display is based on step number order



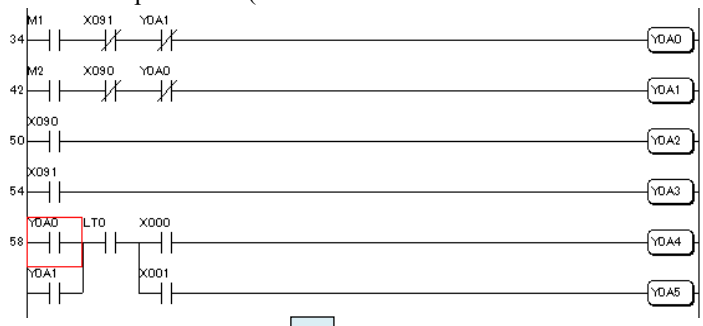
Refer to “Chapter 7 LD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

10.11.3 Search Results Display (LD Language)

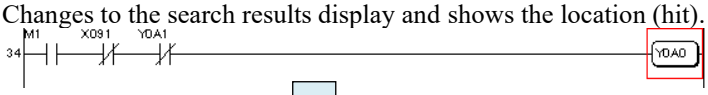
When performing an OM search or command search in an LD program using the PLC Programming software D00 EDITOR, the search results are displayed in a search history that includes previous searches.

If an OM search or command search is started and the step number display is shown, it automatically changes to the search results display.

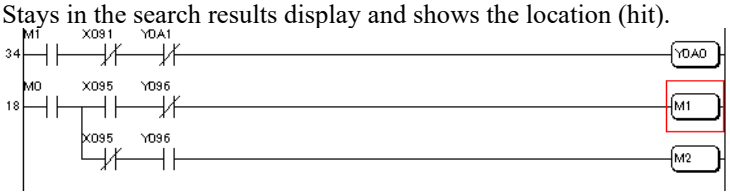
Ex: Search up for YA0 (Searches for the OM YA0 that turns on YA4)



Changes to the search results display and shows the location (hit).



Then, search for M1 (Searches for the OM M1 that turns on YA0).
Stays in the search results display and shows the location (hit).



If another operation (besides an OM or command search) is performed and the search results display is shown, it automatically changes to the step number display.
In the following situations, the search results in the history are cleared.

- When the ladder edit screen is closed
- When the ladder circuit is edited
- When the **[Ctrl]** + **[W]** keys are pressed (in the step number display)

Refer to “Chapter 7 LD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

10.12 Print Function

This function prints programs created in the PLC Programming software D00 EDITOR.
Refer to “Chapter 7 LD edit operation procedure”, “Chapter 8 ST edit operation procedure” or “Chapter 9 FBD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

10.12.1 Print Content

The function prints the POU that is displayed at the forefront in the PLC Programming software D00 LD/ST Programming Editor or FBD Programming Editor.
The print function prints the following items.

Language	Item
LD program	Ladder program (*1)
	OM comment, inline comment, label comment and coil remark (*2)
	Header and footer
	List of contact locations
ST program	Variable (Scope, variable name, data type, element number, default value and comment)
	ST program
	Header and footer
FBD program	Variable (Scope, variable name, data type, element number, default value and comment)
	FBD program
	Header and footer

*1 Any program in the CALC command will not print.
*2 When a comment is displayed in the PLC Programming software D00 LD/ST Programming Editor, it will be printed. Refer to “10.11.1 Display comments (LD language)” for details on the comment display.
(NOTE) When printing, the page number at the bottom of the page is automatically added.

10.12.2 Print Range

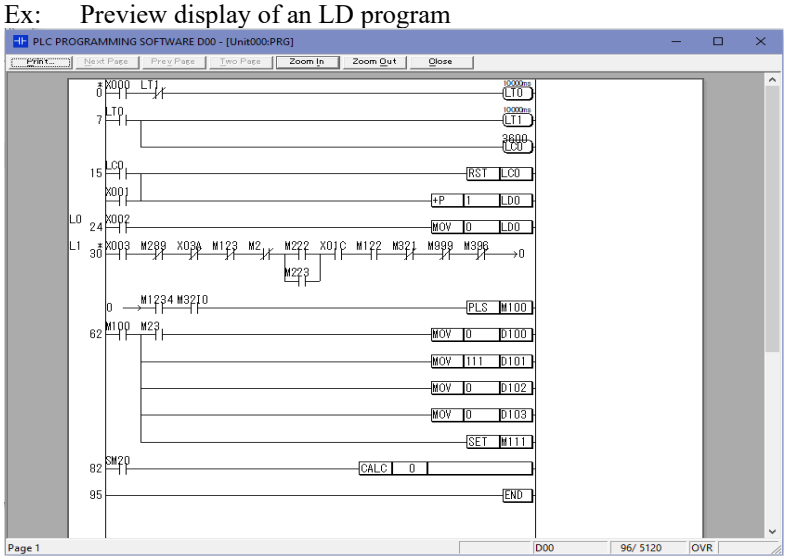
The user can set the printing range from the following selections.

Language	Item
LD program	All POUs
	Range set using step numbers
ST program	All POUs
	Range set using line numbers
FBD program	All POUs
	Range set using network numbers

10.12.3 Print Preview Screen

The user can preview the content before printing it.

Ex: Preview display of an LD program



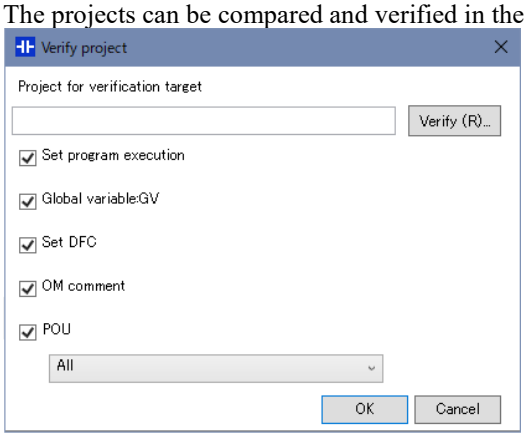
10.13 Verify Projects

The user can compare two projects and check whether the content matches or not.
The project that is currently opened in the PLC Programming software D00 EDITOR can be compared with another project that is saved.
The following items are compared.

- Program execution settings
- Global variables
- POU programs
- Variables registered to POUs
- Comments registered to POUs (LD only)
- Folder configuration

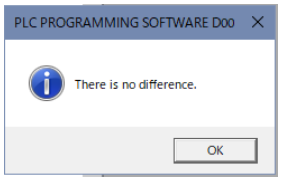
Variables with the same name are compared.
The user can select whether to compare all of the POUs in the project or only compare specific POUs.
POUs with the same name and same type are compared.

The projects can be compared and verified in the PLC Programming software D00.

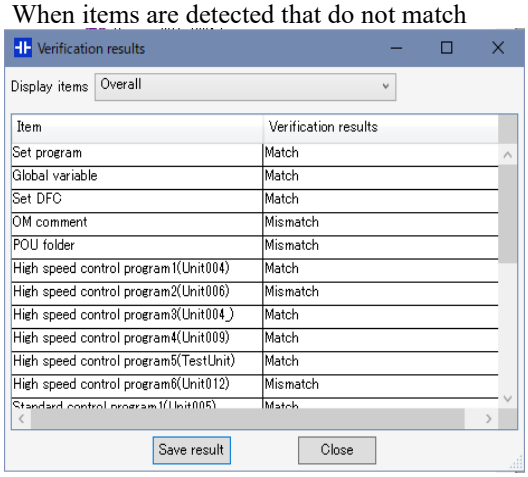


The comparison and verification results are displayed as shown below.

- When all items match

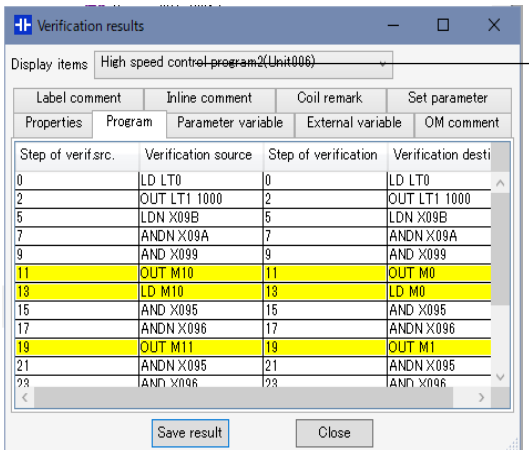


- When items are detected that do not match



10

The user can select the following data under the the <Display items> in the <Verification results>.
Overall, program settings, parameter settings, global variable, DFC settings, OM comment, POU name and POU name - CALC number
When an item is selected (except “Overall”), the user can check the specific details that do not match for each item.



User can change the display items and check the details in the verification results.

Refer to the following for those items where the details can be checked.
Only those items with mismatches are displayed in the <Verification results>. However, if there are more than 100 mismatches for each category, the user can only check the first 100 mismatches listed from the beginning, excluding the program.

Item		Description
Program settings		POUs that are set in the execution sequence
Global variable (Compares variables with the same name)		Scope, data type, element number, default value and comment
OM comment data		Comment set for each global OM
POU*1	Property	Type, return value data type and language
(Compares POUs with the same name and type)	Program	Program description • Uses an abbreviated code to refer to the LD/FBD program, and displays the steps with mismatches • Displays the ST program, and displays the line numbers with mismatches
	Parameter variable (Compares variables with the same name)	Scope, data type, default value and comment
	External variable (Compares variables with the same name)	Scope, data type, default value and comment
	OM comment (LD program only)	Local OM comment
	Label comment (LD program only)	Label comment
	Inline comment (LD program only)	Inline comment
	Coil remark (LD program only)	Coil remark
	Parameter settings	Scan time, monitor WDT, WDT monitor time, WDT error, calculation error, command execution error, output method when mode is changed from STOP to RUN and power failure hold area setting
CACL command (LD program only. Compares CALC commands with same name and number)	Program	Program description • Displays program at line number with mismatch
Folder configuration		Folder structure created by the user for storing user / system defined functions

*1 The details about the mismatches cannot be checked when a POU is missing in the verification source or target.

The data that is displayed in the <Verification results> dialog can be saved to a CSV file.
Before verifying projects, both projects (comparison source and target) need to be converted and saved.
Refer to “6.12 Verify Projects” for details on how to use PLC Programming software D00 EDITOR.

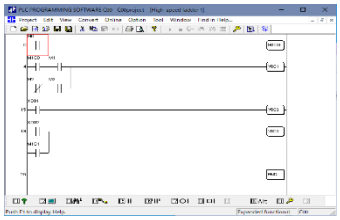
(NOTE 1) This function is not for comparing projects that run in the NC(BUILT IN PLC). To compare projects that run in the NC(BUILT IN PLC), first use the “Read program” function to save the projects that run in the NC(BUILT IN PLC), and then verify the projects. Refer to “11.1.2 Read programs” for details on the program read operation.

(NOTE 2) When comparing POUs, the projects may not be compared properly if a conversion error is included in a POU.

10.14 B00/C00 Project Conversion

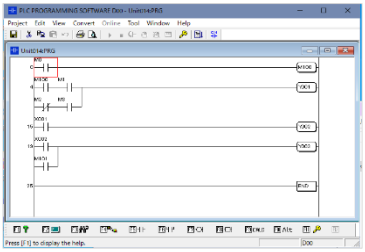
B00 and C00 project conversion refers to the converter in the PLC Programming software D00 EDITOR that converts B00 or C00 project data into project data for the PLC Programming software D00.

B00 or C00



Convert the project

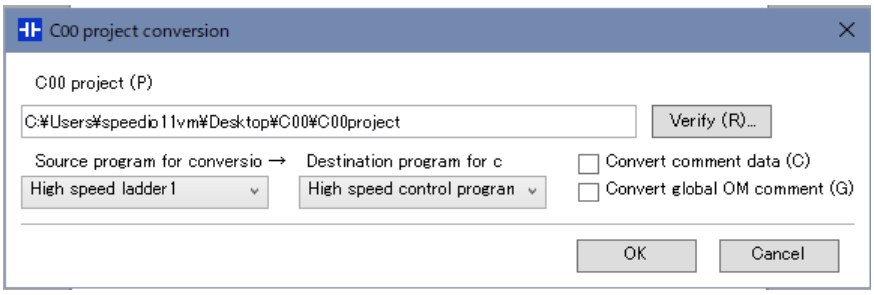
PLC Programming software D00



The following items can be selected.

- Convert comment data
- Convert global OM comment

B00/C00 project conversion is carried out in the PLC Programming software D00.



(NOTE 1) In the PLC Programming software D00, the maximum number of steps in one POU varies.

- High :5K steps
- Standard :40K steps

When registering POUs, be sure that the maximum number of steps is not exceeded in the PLC Programming software D00.

(NOTE 2) When the checkbox is not ticked for the <Enable function> - <B00 and C00 project> (under the <Auxiliary> menu, go to <Option setting> and then the <Customize settings> tab to access), this function displays “Disable”.

10

10.15 Re-assign Boxes (FBD Language)

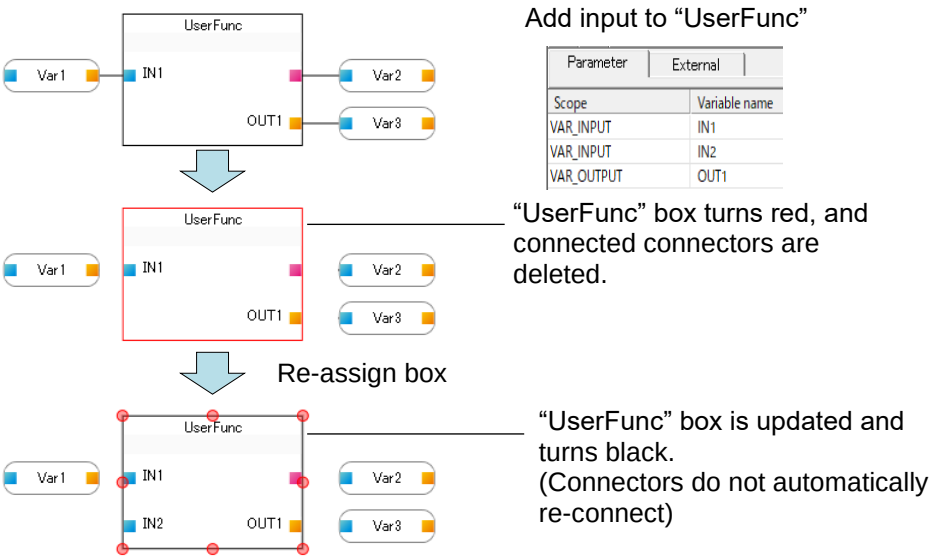
After setting up a variable, user defined function or system defined function box in the program area in the PLC Programming software D00 FBD Programming Editor, if the user performs one of the following operations, then the box turns red. In addition, the connector connected to that box is deleted.

- Change variable data type or element number
- User and system defined functions:
  - Change data type of return value.
  - Add or delete variables in the scope: VAR_IN/VAR_OUT/VAR_IN_OUT.
  - Change data type or element number for the variable in the scope: VAR_IN/VAR_OUT/VAR_IN_OUT.

Conversion is not possible when there is a red box.
If there is a red box, perform one of the following operations.

- Delete the red box and correct the program
- Re-assign the red box

When a box is re-assigned, the settings for the variable, user defined function or system defined function with the same name as the variable name or POU name that is set in the current box is reflected in that re-assigned box.
Operation examples are shown below.



Refer to “Chapter 9 FBD edit operation procedure” for details on how to use PLC Programming software D00 EDITOR.

10.16 Import & Export

Import refers to the project data (bpp format) that was exported beforehand and is imported into the project currently being edited.
Export refers to selecting an item to export from the project currently being edited. (bpp format, file name: PLCPRJ.bpp)

The following items can be selected to export.

- High speed control program
- Standard control program
- User defined function
- Program execution settings
- Global variable
- Global OM comment

Refer to “6.18 Import & Export” for details on how to use PLC Programming software D00 EDITOR.

(NOTE 1) Before importing or exporting, save the project data.
(NOTE 2) An error is triggered if the POU limit or the folder limit is exceeded after importing data. (10 high speed control programs, 10 standard control programs, 924 user defined functions, 100 system defined functions and 64 folders)

CHAPTER 11

PROGRAM EXECUTION FUNCTIONS

11.1 Write & Read Programs
11.2 Program Execution
11.3 RUN & STOP Mode
11.4 Monitoring Function of Program Operation
11.5 Debug Function
11.6 Simulation Function

11.1 Write & Read Programs

11.1.1 Write programs

After creating a program in PLC Programming software D00 EDITOR, the user must write the project data into the NC(BUILT IN PLC) in order to run the program in it.
The PLC Programming software D00 is used to write the data into NC(BUILT IN PLC).
The following data is written into NC(BUILT IN PLC).

Project data	Writes? (○: Writes ×: Does not write)
POU (*1)	○
Parameter settings (*4)	○
Variable default values (*4)	○
Project data set (*2) (*3) (*4)	○

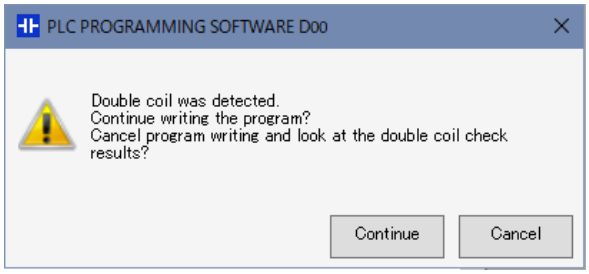
*1 Programs set in the "Set program execution" and the functions used in those programs are written. Code created with the conversion function is written.
*2 All files created in the PLC Programming software D00 EDITOR (included in the project data to be written) are compressed and sent to the NC(BUILT IN PLC). This data is not referenced when executing a program in the NC(BUILT IN PLC), but it is used in the "Read program" operation.
*3 This does not write in the program write operation to the simulation function.
*4 When performing a write during run operation, this does not write to the NC(BUILT IN PLC).

The NC(BUILT IN PLC) must be in a stopped state before writing a program. The write during run operation is not possible.
After writing a program, the NC(BUILT IN PLC) must be in a run state to execute the program.
When a program is written, all the local OM values in the NC(BUILT IN PLC) are cleared. The global OMs are not cleared. All variables are reset to their default values.

Refer to the "5.10 Write/Read Programs" for details on how to use the PLC Programming software D00 to write programs to NC(BUILT IN PLC).

(NOTE 1) When writing a program from the PLC Programming software D00 EDITOR to the NC(BUILT IN PLC), take the necessary safety precautions when using the device equipped with the NC(BUILT IN PLC) and ensure the surroundings are safe as well.

(NOTE 2) When a program write operation is executed, a double coil check is automatically carried out. If a double coil is detected, then the following message is displayed.



Select <Continue> to proceed with the program write operation.
Select <Cancel> to cancel the write operation in order to check the double coil check results and correct the program.
Refer to "10.10.1 Double coil check" for details on the double coil check.

(NOTE 3) When a checksum error is detected in a POU or in the settings, the program will not write. Check and correct the settings and program where the checksum error was detected. Then, after saving the data, write the program.

(NOTE 4) The data needs to be converted beforehand when writing a program to the NC(BUILT IN PLC). Refer to "10.9 Program conversion function" for further details about the conversion.
Note that when an attempt is made to write an unconverted program, the user will be prompted to convert the program.

(NOTE 5) If the simulation function is enabled in the PLC Programming software D00 EDITOR, the program writes to the simulation function.

(NOTE 6) Before writing a program if the project is not saved in the PLC Programming software D00 EDITOR, save it.
(NOTE 7) Refer to “11.1.2 Read programs” and “11.1.3 Write during run” for details about the program read and the write during run operations.
(NOTE 8) When writing to the simulation function, the user can also write the <Option setting>: <Perform the count up operation on the timer per scan> and/or <Count up time per scan>. Refer to “11.6 Simulation function” for further details.

11.1.2 Read programs

This function in the PLC Programming software D00 EDITOR reads the project data that is written to the NC(BUILT IN PLC).

The program is read from the PLC Programming software D00.
Refer to “5.10 Write/Read Programs” for details on how to use the PLC Programming software D00.

(NOTE 1) When reading a program, specify the save location for the project data.
(NOTE 2) A project written to the simulation function cannot be read.
(NOTE 3) If a project is not written to the NC(BUILT IN PLC), then this function cannot be executed.

11.1.3 Write During Run

11.1.3.1 What is write during run?

The write during run operation is a function that allows a program to be written while the NC(BUILT IN PLC) is in a RUN state or DEBUG-RUN state.
The write during run operation is only possible for changes to the logic of a high speed or standard control program.

Change	Write during run? (○: Possible ×: Not possible)
High speed and standard control program changes	○ However, the following restrictions apply. • Multiple POUs cannot be changed at the same time. • Multiple circuit blocks cannot be changed at the same time for POUs in the LD language.
Program changes in user or system defined functions	×
Variables editing (adding and deleting variables or variable setting changes)	×
Program execution setting changes	×
Parameter setting changes	×
Other setting changes not noted above	×

The write during run operation is processed during the END processing in the NC(BUILT IN PLC). As a result, the size of the program that is written affects the scan time.

In a write during run operation, the state before the writing operation is maintained for the following.
- OM and variable values
- Status of rising edge and falling edge commands

However, when the LD program is changed by the write during run operation, then the status of the rising and falling edge commands (included in the edited circuit block) is cleared.

(NOTE) When modifying a circuit block with a rising or falling edge command, take into consideration that the commands are cleared when performing the write during run operation.

11.1.3.2 Conditions for write during run

All of the following conditions must be met to perform a write during operation.

- Projects must match in the PLC Programming software D00 EDITOR and in the NC(BUILT IN PLC).
- The high speed or standard control program and the POUs registered in the “Set program execution” must both be changed. (User or system defined functions are not possible)
- Only one POU program is changed.
- Only one circuit block is changed when a POU in the LD language is the target of the write during run operation.
- NC(BUILT IN PLC) must be in a RUN or DEBUG-RUN state.
- Write location from PLC Programming software D00 EDITOR must be the NC(BUILT IN PLC). (Simulation function cannot be the write location.)

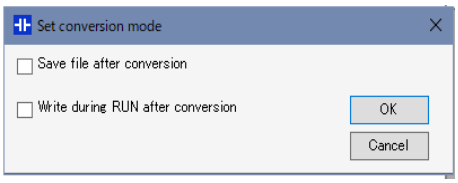
When all of the conditions above have been met, use the “Write program” function to write all programs for running in the NC(BUILT IN PLC).

11.1.3.3 Execute write during run operation

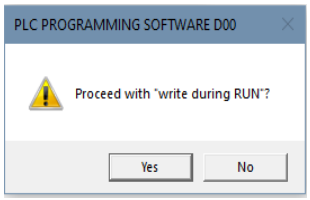
(NOTE) When using the write during run operation, take the necessary safety precautions with the device that has the NC(BUILT IN PLC) and also with the operating environment.

The following is the basic procedure for executing the write during run operation.

1. Display the <Conversion mode settings> dialog (from the <Auxiliary> - <Conversion mode settings> menu) in the PLC Programming software D00, and then set the <Write during RUN after conversion>.



2. Use the PLC Programming software D00 to open the same project that is running in the NC(BUILT IN PLC).
3. Use the PLC Programming software D00 LD/ST Programming Editor or FBD Programming Editor to change a high speed or standard control program. When changing the POU in the LD language, modify or edit the program for one circuit block.
4. Select <Convert> from the <Convert> menu in the PLC Programming software D00.
   When the data is compared with the project stored in the NC(BUILT IN PLC) and that data matches, then a confirmation message appears asking whether or not to perform the write during run operation.
   Select <Yes> to execute the write during run operation.



5. When the write during run operation is complete, the message <Completed.> is displayed.

11.1.3.4 Special notes and restrictions

(NOTE 1) When using the write during run operation, take the necessary safety precautions with the device that has the NC(BUILT IN PLC) and also with the operating environment.

(NOTE 2) Check the following special notes and restrictions first before using the write during run operation.

The following shows the special notes and restrictions for the write during run operation.

Item	Description
Read program	A program written with the write during run operation cannot be read using the “read program” function. The project data, before the write during run operation, is read. Therefore, we recommend ultimately using “write program” to write the program.
Exiting NC(BUILT IN PLC)	When a program is written using the write during run operation, after exiting and then re-launching the NC(BUILT IN PLC), the project data from before the write during run operation is used for operation. Therefore, we recommend ultimately using “write program” to write the program.
Rising edge/Falling edge command	When the LD program is changed by the write during run operation, the rising and falling edge command (in the edited circuit block) status is cleared.
Double coil	When a double coil is detected in a program during the write during run operation, the PLC Programming software D00 EDITOR displays a confirmation message asking whether or not to proceed with the write during run operation. The user must select whether to proceed or cancel the write during run operation.
When an error or warning is detected at conversion	The write during run operation is not possible when an error is detected at conversion. The write during run operation is carried out when only a warning is detected at conversion.
When there are no changes to the program	The write during run operation is not carried out, even after conversion, if there are no changes to the program (after the last conversion).
When making changes to multiple POUs	The write during run operation is not carried out. Only conversion is carried out.
When making changes to multiple circuit blocks in an LD program	The write during run operation is not carried out. Only conversion is carried out.
When making changes to a user defined function or system defined function	The write during run operation is not carried out. Only conversion is carried out.
When the NC(BUILT IN PLC) is in a STOP state	The write during run operation is not carried out. Only conversion is carried out.
Write during run operation during simulation	The write during run operation is not carried out. Only conversion is carried out.
When a file checksum error is detected for writing	The write during run operation is not carried out. Correct and save the program, and then re-execute it.
When making a change to the program execution settings, parameter settings, global variables or DFC settings	In addition to a POU program change, when a change is made to the program execution settings, parameter settings, global variables or DFC settings, only the POU program change is written to NC(BUILT IN PLC) in a write during run operation.

11.2 Program Execution

To execute a written program, the NC(BUILT IN PLC) must be in a RUN state.
The mode changes to a RUN state in the NC(BUILT IN PLC) for the following situations.

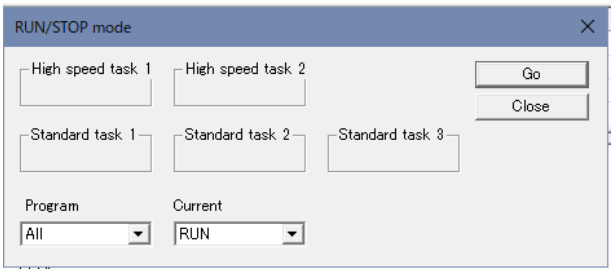
- When the program is written and the NC(BUILT IN PLC) is launched
- When the PLC Programming software D00 EDITOR is used to change NC(BUILT IN PLC) to a RUN state

(NOTE) The launch method of the NC(BUILT IN PLC) depends on the porting method.

11.3 RUN & STOP Mode

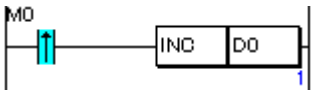
The user can use the PLC Programming software D00 EDITOR to remotely change between RUN and STOP mode in the NC(BUILT IN PLC).

Select the program to execute the mode change and the corresponding operation from the dialog below.



(NOTE) Proceed carefully by checking the status of the device that has the NC(BUILT IN PLC) and ensuring the operating environment is safe. Otherwise, a user mistake in operation may cause an accident or damage to the machine.

<u>Operation when changing from RUN to STOP</u>
When the mode is changed from RUN to STOP, the Y outputs all turn OFF.
In addition, the previous values for the pulse are cleared. As a result, when changing from a STOP to a RUN state the next time, the mode change will not execute because there are no pulse conditions to carry out the operation. For example, in the figure below, the “INC D0” does not execute when the mode of NC(BUILT IN PLC) changes from RUN to STOP to RUN.



Refer to “5.13 Run & Stop in NC(BUILT IN PLC)” for details on executing and changing between RUN and STOP.

11.4 Monitoring Function of Program Operation

The PLC Programming software D00 LD/ST Programming Editor and FBD Programming Editor can be used to check the following items, such as the operation status of the NC(BUILT IN PLC) as well as the current values of OMs and variables.

Item	Description	Display location in tree
Status of NC(BUILT IN PLC)	State of NC(BUILT IN PLC) (RUN, STOP or DEBUG-RUN)	Status bar
Program operation status	Program operation status such as status of contacts and coils, etc.	Program editing area
Scan time	Current scan time (timed value)	Monitoring status bar
Current value of OM	Current value of OM	Program editing area, batch monitoring and monitoring registration
Current value of variable	Current value of variable	Variable editing area
Error code (when error is triggered)	Error code	Monitoring status bar
Error step (when error is triggered)	Step with error	Monitoring status bar
Error information (only ST and FBD when error is triggered)	Description and location of error	Message display area

Refer to “11.4.2 Display in monitor mode” for further details.

The PLC Programming software D00 LD/ST Programming Editor and FBD Programming Editor have the following monitoring functions.

Function	Overview
Start, stop and end monitoring	PLC Programming software D00 LD/ST Programming Editor and FBD Programming Editor are used to start, stop and end monitoring in the NC(BUILT IN PLC).
Batch monitoring	Lists the current values of the OMs for each OM type.
Monitoring registration	Constantly displays the current value of the specified OM.
SET/RESET OM	Changes current value of bit OM.
Change current OM value	Changes current value of word OM.
Change current variable value	Changes current value of variable.
Clear all OMs/variables	Resets all OMs or variables to their default values.
Rising edge and falling edge triggers (LD only)	Stops monitoring at the rising or falling edge of the specified OM.
Change monitoring base number	Changes the base number (decimal/hexadecimal) in the current value display for the OM or variable.

Refer to the sections from “11.4.3 Batch monitoring (LD and ST languages)” to “11.4.10 Change monitoring base number” for further details about each operation.

(NOTE 1) When changing OM or variable values online, be sure to take the necessary safety precautions. If an invalid value is set, the device that has the NC(BUILT IN PLC) installed could malfunction.

(NOTE 2) In the PLC Programming software D00 EDITOR, the user can change to monitor mode in multiple POU windows at the same time. However, the current value of the OM or variable is updated only in the POU displayed at the forefront.

(NOTE 3) Refer to “11.5 Debug function” for details on the debug function that stops the operation of the NC(BUILT IN PLC) at the specified step and when the conditions specified by the user are met.

11.4.1 Start, Stop & End Monitoring

11.4.1.1 Monitoring location

The PLC Programming software D00 EDITOR monitors the NC(BUILT IN PLC) and the simulation function.
The monitoring locations are specified using the following methods.

Monitoring location	How to specify
NC(BUILT IN PLC)	When <Simulation> under the <Tool> menu in the PLC Programming software D00, PLC Programming software D00 LD/ST Programming Editor and FBD Programming Editor is turned OFF, then the NC(BUILT IN PLC) becomes the monitoring location. It monitors the NC(BUILT IN PLC) that is connected to the communication port specified in the PLC Programming software D00 EDITOR.
Simulation function	When <Simulation> under the <Tool> menu in the PLC Programming software D00, PLC Programming software D00 LD/ST Programming Editor and FBD Programming Editor is turned ON, then the simulation function becomes the monitoring location.

Operation of multiple POUs is possible in the NC(BUILT IN PLC), but only one POU can be monitored at the same time in the PLC Programming software D00 EDITOR. This function monitors the POU that is displayed at the forefront in the edit window of the PLC Programming software D00 EDITOR.
A high speed control program, standard control program, user defined function and system defined function can also be monitored.

Refer to the “5.12 NC(BUILT IN PLC) Monitoring” for details on the PLC Programming software D00 EDITOR menu and operations.

11.4.1.2 Start, stop & end monitoring

The PLC Programming software D00 LD/ST Programming Editor and FBD Programming Editor are used to start, stop and end monitoring.

Item	Description
Start monitoring	Starts monitoring the operation status of the NC(BUILT IN PLC). “Monitor mode” refers to when the PLC Programming software D00 LD/ST Programming Editor or FBD Programming Editor is monitoring the NC(BUILT IN PLC).
Stop monitoring	Stops only the monitoring while the NC(BUILT IN PLC) is executing a program. The user can check the status of the NC(BUILT IN PLC) at the time when the monitoring is stopped.
End monitoring	Ends and exits monitor mode. The PLC Programming software D00 LD/ST Programming Editor or FBD Programming Editor changes to edit mode, that is, the mode for editing a program.

Refer to the “5.12 NC(BUILT IN PLC) Monitoring” for details on how to start, stop and end monitoring.

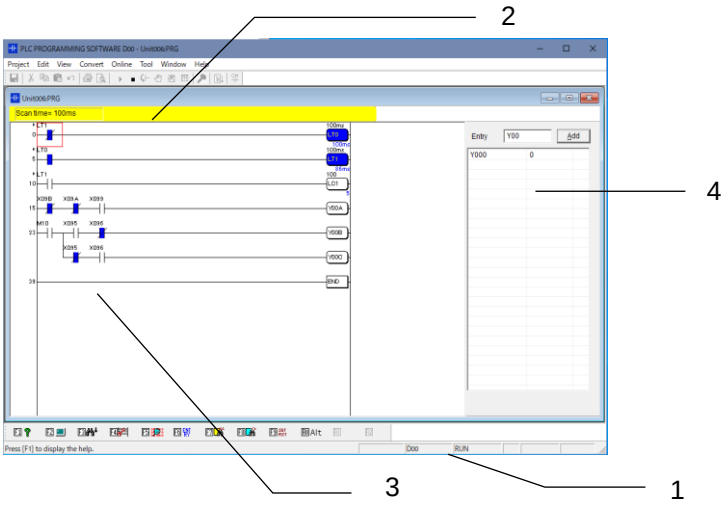
(NOTE 1) Program editing, variable editing and program writing are not possible in monitor mode.
(NOTE 2) Monitor the program that is written to the NC(BUILT IN PLC). The user can use the PLC Programming software D00 LD/ST Programming Editor and FBD Programming Editor to open a program that is not written to the NC(BUILT IN PLC) and switch to monitor mode. However, the displayed content and the actual operation will not match.
(NOTE 3) Even if the monitoring is started, stopped or ended in the PLC Programming software D00 EDITOR, the operation status of the NC(BUILT IN PLC) does not change.

11.4.2 Display in Monitor Mode

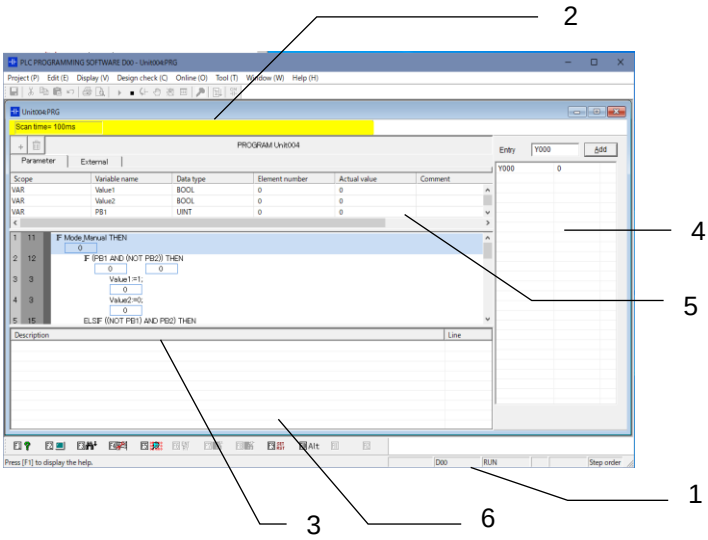
When the PLC Programming software D00 LD/ST Programming Editor or FBD Programming Editor is in monitor mode, the operation status of the NC(BUILT IN PLC) is displayed.

No.	Item	Display description
1	Status bar	Displays the state of the NC(BUILT IN PLC) (RUN, STOP or DEBUG-RUN).
2	Monitoring status bar	Displays the scan time, monitoring status, error code, POUs with an error and error steps.
3	Program editing area	Displays the operation status of the program and the current value of an OM.
4	Monitoring registration	Displays the current value of the specified OM.
5	Variable editing area	Displays the current value of a variable.
6	Message display area	Displays the error information in a program for ST and FBD languages.

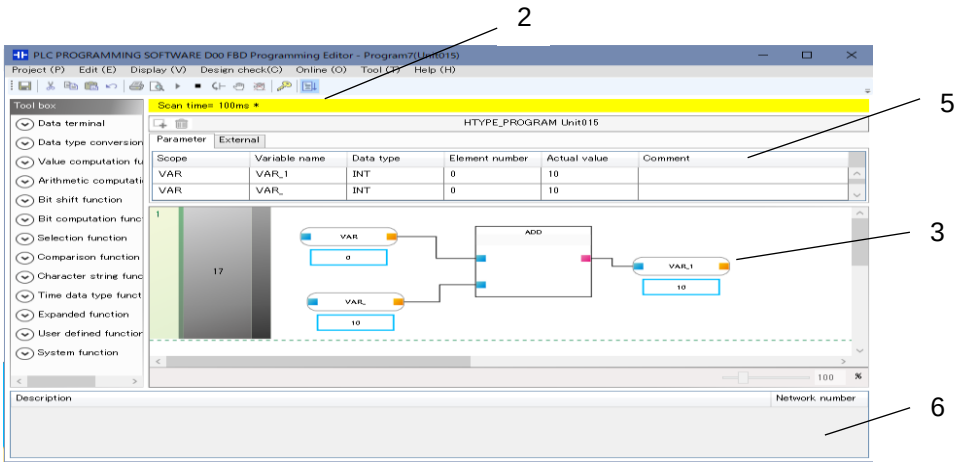
LD/ST Programming Editor (when monitoring an LD program)



LD/ST Programming Editor (when monitoring an ST program)



FBD Programming Editor (when monitoring an FBD program)

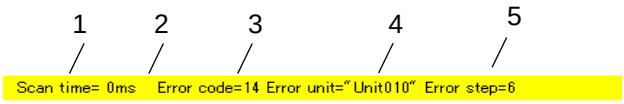


11.4.2.1 Status bar

The state of the NC(BUILT IN PLC) (RUN, STOP or DEBUG-RUN) is displayed.

11.4.2.2 Monitoring status bar

The scan time, monitoring status, error code, error unit and error step are displayed.

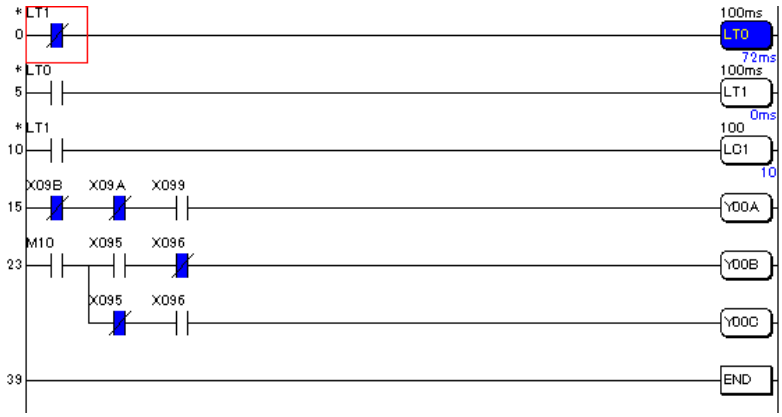


No.	Item	Display description
1	Scan time	Displays the timed value (unit: ms) of the scan time. Note that "0 ms" is displayed when one of the following situations applies to the NC(BUILT IN PLC). • When a program is not written • When in a STOP state (including when stopped due to an error) • When in DEBUG-RUN state
2	Monitoring status	Asterisk (*) flashes while monitoring is executing. Asterisk (*) does not flash while monitoring is stopped.
3	Error code	Displays the error code when an error is triggered in the NC(BUILT IN PLC). This item does not display when there is no error.
4	Error unit	Displays the POU name with the error only when the data is written in the NC(BUILT IN PLC). When there is an error in a user or system defined function, the POU name that calls the user defined function or system defined function is displayed. This item does not display when there is no error.
5	Error step	Displays the step number where the error occurred. This item does not display when there is no error.

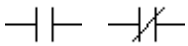
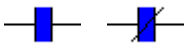
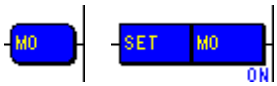
(NOTE) When multiple programs are running in the NC(BUILT IN PLC), the error step is displayed starting with step 0 for the beginning of each POU.

11.4.2.3 Program editing area (LD language)

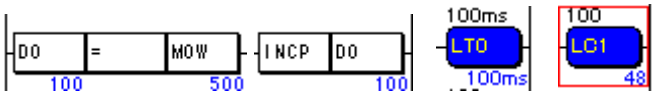
The execution status of the LD program and the current values of the OMs are displayed.



The circuit symbols are displayed as follows.

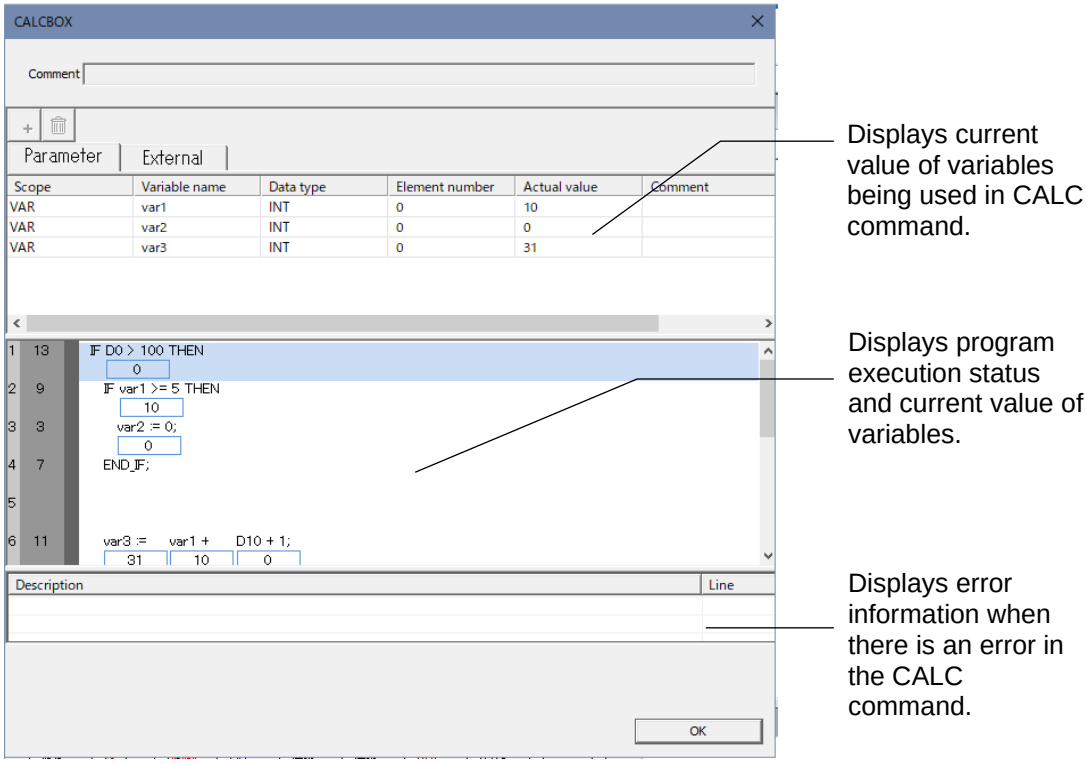
Circuit symbol	OFF	ON
Contact		
Output		

The current value of an OM being used in a circuit symbol (except contacts) is displayed below the circuit symbol.



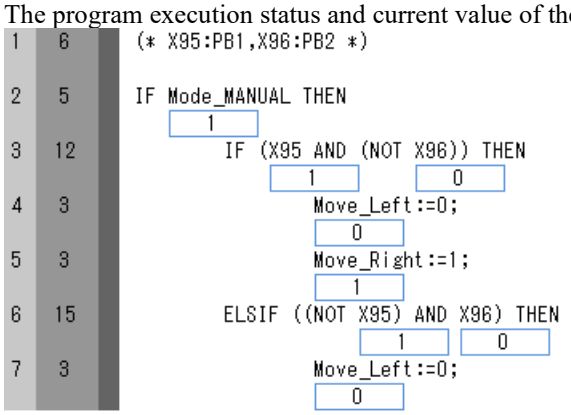
(NOTE) The user can change the base number display for the current value between a decimal or hexadecimal. Refer to “11.4.10 Change monitoring base number” for details about changing the displayed base number.

When monitoring a program that is set to the CALC command, double-click on the CALC command.
The current value is displayed for the program or variable being used in the CALC command.
Refer to “11.4.2.4 Program editing area (ST language)” and “11.4.2.7 Variable editing area (ST and FBD languages)” for details about the display.



11.4.2.4 Program editing area (ST language)

The program execution status and current value of the OMs and variables are displayed.



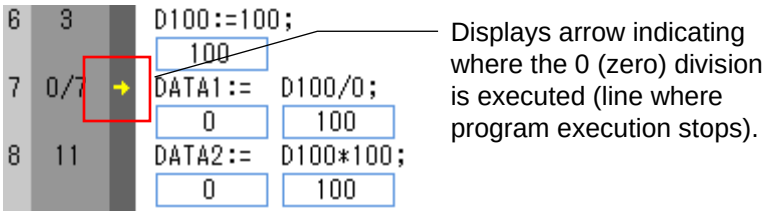
The current value is displayed under the OM or variable in the program.
The current value of a BOOL data type is displayed as 0 or 1.
The current value for the data types REAL, LREAL, DATE, TIME, TOD, DT and STRING is displayed in the format of that data type.
The current value of other data types is displayed as a decimal or hexadecimal.

(NOTE 1) The user can change the base number display for the current value between a decimal or hexadecimal. Refer to “11.4.10 Change monitoring base number” for details about changing the displayed base number.

(NOTE 2) Refer to “11.4.2.8 Array variable monitoring” for details about monitoring array variables.

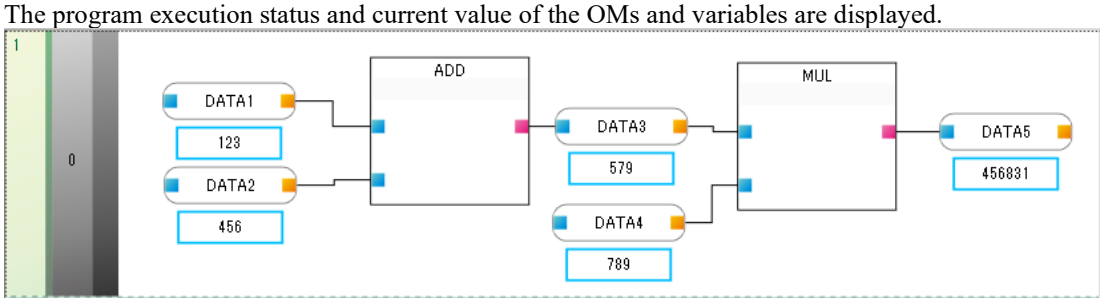
In addition, when there is an error and the program execution stops, an arrow is displayed indicating where the program stopped.

Ex: When 0 (zero) division is executed and an error is triggered



11.4.2.5 Program editing area (FBD language)

The program execution status and current value of the OMs and variables are displayed.



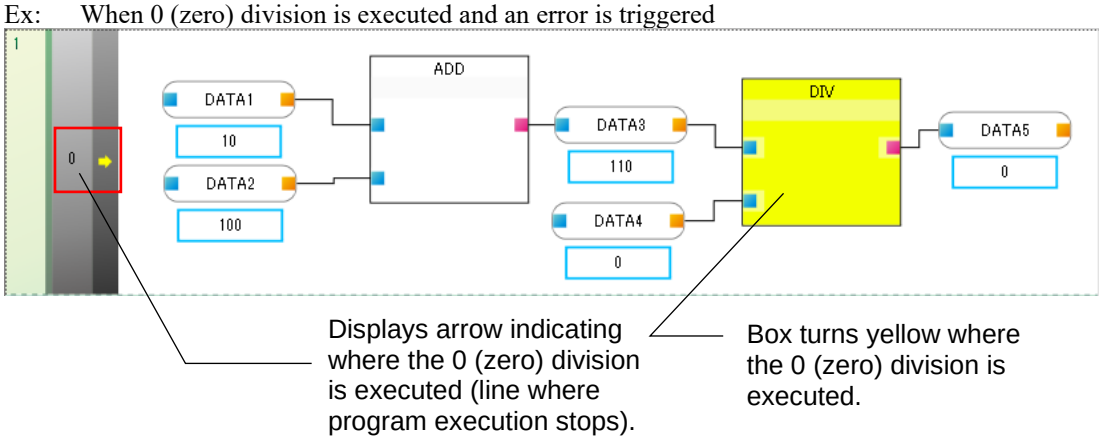
The current value is displayed under the OM or variable in the program.
The current value of a BOOL data type is displayed as 0 or 1.
The current value for the data types REAL, LREAL, DATE, TIME, TOD, DT and STRING is displayed in the format of that data type.
The current value of other data types is displayed as a decimal or hexadecimal.

(NOTE 1) The user can change the base number display for the current value between a decimal or hexadecimal. Refer to “11.4.10 Change monitoring base number” for details about changing the displayed base number.

(NOTE 2) Refer to “11.4.2.8 Array variable monitoring” for details about monitoring array variables.

When there is an error and the program execution stops, an arrow is displayed indicating where the program stopped and the function box turns yellow.

Ex: When 0 (zero) division is executed and an error is triggered



After setting up variable boxes in the program area, if a variable is deleted or a setting is changed, the variable box turns red. Monitoring is not possible for the current value in the variable box displayed in red. In this situation, correct the program. Refer to “Chapter 9 FBD edit operation procedure” for further details.

11.4.2.6 Monitoring registration (LD and ST languages)

The current value of the specified OM is displayed per POU.
Refer to “11.4.4 Monitoring registration (LD and ST languages)” for further details.

11.4.2.7 Variable editing area (ST and FBD languages)

The current values are displayed for the variables registered to the program being monitored.
The current value of a variable can be changed by entering a numerical value or character string into the <Current value> field.

Scope	Variable name	Data type	Element number	Actual value	Comment
VAR	var_BOOL	BOOL	0	0	
VAR	var_WORD	WORD	0	192	
VAR	var_INT	INT	0	125	
VAR	var_REAL	REAL	0	1.15718E-038	
VAR	var_DATE	REAL	0	0	
VAR	var_STRING	STRING	0	''	
VAR	var_ARRAY	INT	10	Display current value	

The current value of a BOOL data type is displayed as 0 or 1.
The current value for the data types REAL, LREAL, DATE, TIME, TOD, DT and STRING is displayed in the format of that data type.
The current value of other data types is displayed as a decimal or hexadecimal.

(NOTE 1) Check the current OM values in the program display, or in the batch monitoring and monitoring registration. Refer to “11.4.3 Batch monitoring (LD and ST languages)” and “11.4.4 Monitoring registration (LD and ST languages)” for details on batch monitoring and monitoring registration.

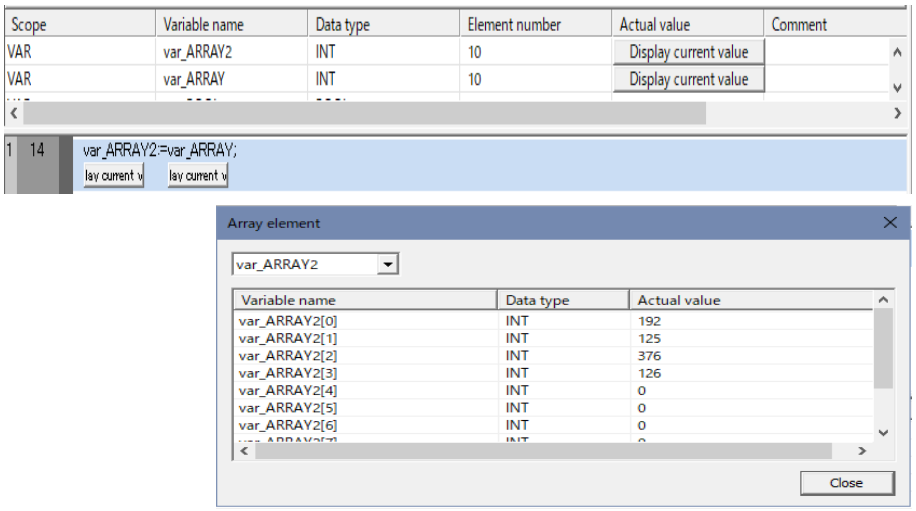
(NOTE 2) Refer to “11.4.7 Change current variable value” for details about changing the current variable values.

(NOTE 3) The user can change the base number display for the current value between a decimal or hexadecimal. Refer to “11.4.10 Change monitoring base number” for details about changing the base number in monitoring.
However, the display cannot be changed for data types (BOOL, REAL, LREAL, DATE, TIME, TOD, DT or STRING) that cannot be expressed in decimals or hexadecimals.

(NOTE 4) Refer to “11.4.2.8 Array variable monitoring” for details about monitoring array variables.

11.4.2.8 Array variable monitoring

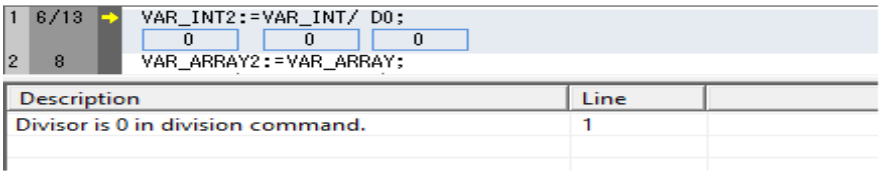
The <Display current value> button is shown in the current value field for the array variables.
After clicking on the <Display current value> button, the <Array element> dialog is displayed and shows the current values for each element in the array.
The current value of a variable can be changed by entering a numerical value or character string into the <Current value> field in the <Array element> dialog.



(NOTE) The user can change the base number display for the current value between a decimal or hexadecimal. Refer to “11.4.10 Change monitoring base number” for details about changing the base number in monitoring.
However, the display cannot be changed for data types (BOOL, REAL, LREAL, DATE, TIME, TOD, DT or STRING) that cannot be expressed in decimals or hexadecimals.

11.4.2.9 Message display area (ST and FBD languages)

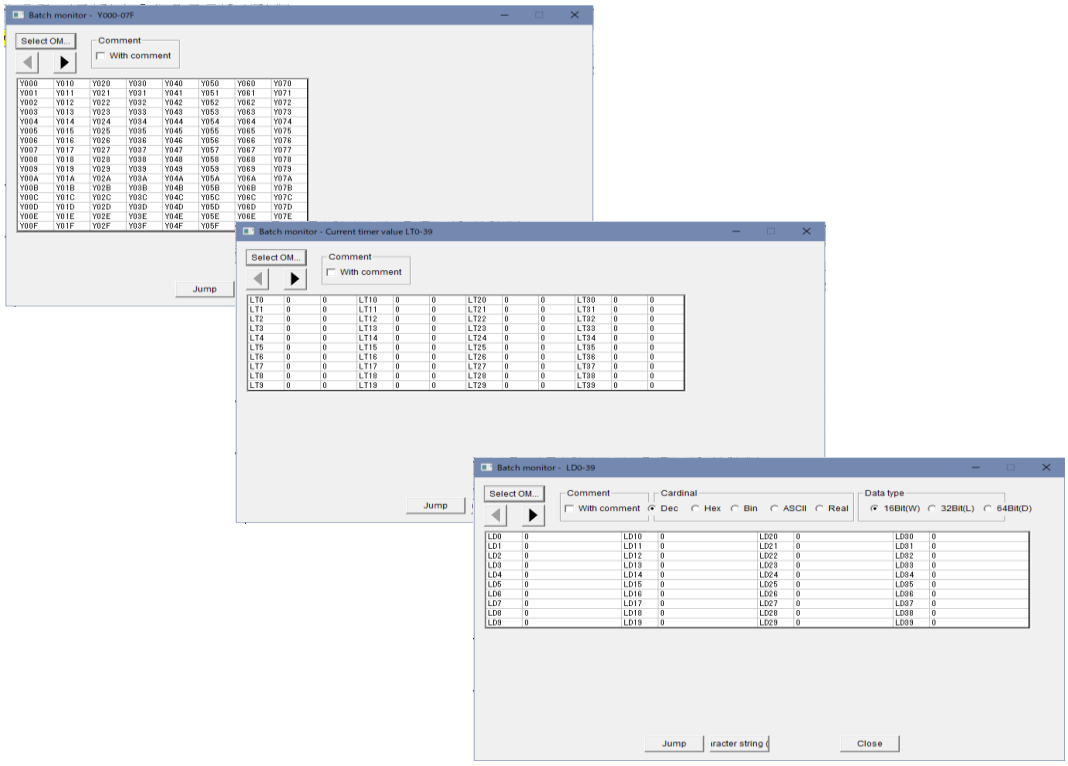
When there is an error during program execution, an error description and location are displayed.



No.	Item	Description
1	Description	Displays a description of the error.
2	Line / Network number	Displays the line number or network number for the error.

11.4.3 Batch monitoring (LD and ST languages)

The current OM values for each OM type are listed.



In batch monitoring, the following OMs can be monitored.

Item	Program	Description
OMs that can be monitored	Global OM	X, Y, M, D, BX, BY, BDX, BDY
	High speed / Standard task	LM, LD, SM, SD, LTS (contact), LTC (coil), LT (current value), LCS (contact), LCC (coil) and LC (current value)

11

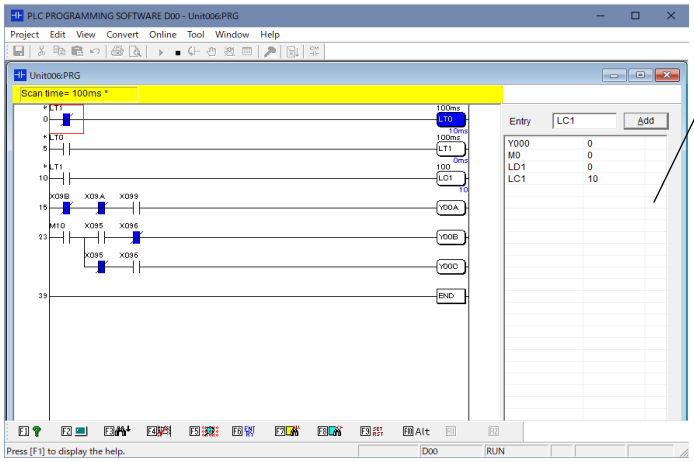
The following shows the functions and specifications for batch monitoring.

Item		Description
Start	Display Qty.	Four OMs can be displayed at the same time for batch monitoring.
	Previous monitoring	When batch monitoring is started, the user can monitor OMs that were previously displayed the last time. Refer to “7.12.5 Batch Monitoring” for details on the batch monitoring operations.
Current OM value display	Bit OM	Displays the current value (ON/OFF).
	Word OM	When the current value for the word OM is displayed, the user can select the following base numbers for monitoring. (Excluding timers and counters) • DEC (Decimal numeral) • HEX (Hexadecimal numeral) • BIN (Binary numeral) • ASCII (Character code is used for the current OM value, and displays compatible character strings) • Real (Real number) In addition, the user can select the following data sizes for monitoring. (Excluding timers and counters) • 16bit • 32bit (Displays 2 consecutive OMs as one piece of data) • 64bit (Displays 4 consecutive OMs as one piece of data)
	Show comments	The user can display an OM comment together with the current value.
	Jump	Changes the display or jumps to the OM for the specified number.
Change current value	Bit OM	The user can change the current value for the bit OM.
	Word OM	The user can change the current value for the word OM. The value can be changed to a decimal, hexadecimal, real number or character string. In addition, the user can also combine two or four consecutive OMs as one piece of data by specifying 32-bit or 64-bit.

Refer to “7.12.5 Batch Monitoring” for details on how to start batch monitoring and related operations.

11.4.4 Monitoring registration (LD and ST Languages)

When monitoring an LD or ST program, this function constantly monitors the specified OM value.



Monitoring registration list

The following OMs can be monitored.

Item	Description
OMs that can be monitored	X, BX, BDX, Y, BY, BDY, M, D, LM, LD, LT, LC, SM, SD, IX Contacts, coils, current values and set values can be monitored for timers and counters. LTS (timer contact), LTC (timer coil), LTV (current timer value), LTP (timer set value), LCS (counter contact), LCC (counter coil), LCV (counter current value) and LCP (counter set value) In addition, an LT and LC are registered, the current values for each can be monitored.

The following shows the functions for monitoring registration.

Item	Description
Display Qty.	Displays one per POU in the monitoring registration area.
Maximum registered OMs	The user can display the current values for a maximum of 24 OMs.
Clear registration	Deletes or clears a registered OM from the monitoring registration list. The user can delete one OM at a time or clear all registered OMs.
Change current value	The user can change the current value of a registered OM. After double-clicking on the OM display area, a dialog appears for changing the current value. Refer to “11.4.5 SET/RST OM (LD and ST languages)” and “11.4.6 Change current OM value” for details on how to change the current OM value.
Save registration data	The registered OM data is automatically saved when ending and exiting monitor mode. Next time when monitoring is started again for the same program, the OMs registered the last time appears in the monitoring registration list.

Refer to “7.12.2 Monitoring Registration” or “8.11.3 Monitoring Registration ” for details on the monitoring registration operations.

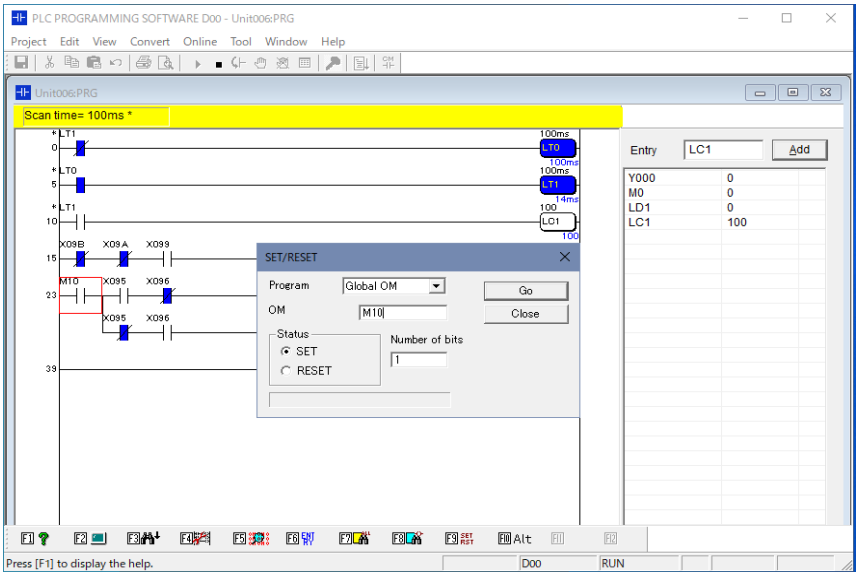
(NOTE 1) When monitor mode is started in the PLC Programming software D00 LD/ST Programming Editor, the monitoring registration list is automatically displayed.

(NOTE 2) Monitoring registration does not support registering variables.

11.4.5 SET/RESET OM (LD and ST Languages)

This function changes the current value of the selected bit OM in the program. In addition, the user can reset the current value of the word OM to 0 (zero).
The user can specify a range of consecutive OMs and change the current values for them at the same time.

Function	Description
SET	Sets the current value of the bit OM to ON (1).
RESET	Sets the current value of the bit OM to OFF (0). The value can be reset to 0 (zero) for the following word OMs: D, BDX, BDY, LD, SD, IX, LT and LC.



The current value can be changed for the following OMs.

Item	Description
OM	X(*1), Y, BX, BY, M, SM, LTS (timer contact), LTC (timer coil), LCS (counter contact), LCC (counter coil), D (*2), BDX (*2), BDY (*2), LD (*2), SD (*2), IX (*2), LT (timer current value) (*2) (*3) and LC (counter current value) (*2) (*3)

*1 Even after changing the current value with this function, the value is written back to the external input state for OMs that are refreshed.
*2 Only reset is possible.
*3 The current value becomes 0 (zero), and the timer/counter contacts and coils turn OFF.

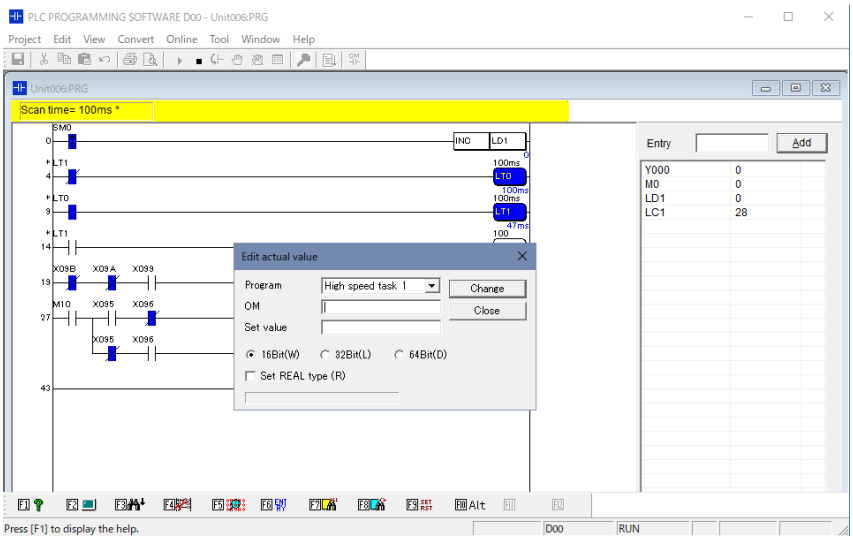
Refer to “7.12.3 SET/RST OM” or “8.11.4 SET/RST OM” for details on setting and resetting OMs.

(NOTE 1) When changing OM values online, be sure to take the necessary safety precautions. If an invalid value is set, the device that has the NC(BUILT IN PLC) installed could malfunction.
(NOTE 2) This function does not continuously force the OM to stay ON or OFF. Even if this function is used to change the current value, the OM value will ultimately change based on the program operation or an external input.
(NOTE 3) An OM with an index cannot be specified when changing the current value.
(NOTE 4) This function cannot be used to change the value of a variable.

11.4.6 Change Current OM Value

This function changes the current value of the selected word OM in the program.

Function	Description
Change current value	Changes current value of word OM. The value can be changed to a decimal, hexadecimal or real type (real number). In addition, the user can also combine two or four consecutive OMs as one piece of data by specifying 32-bit or 64-bit.



The current value can be changed for the following word OMs.

Item	Description
OM	Changes each current value when D, BDX, BDY, LD, SD, IX, LT (timer), LC (counter), LT and LC are set.

Refer to “7.12.4 Change Current Value”, “8.11.5 Change Current Values” or “9.11.3 SET/RST OM & Change Current OM Value” for details on how to change the current OM value.

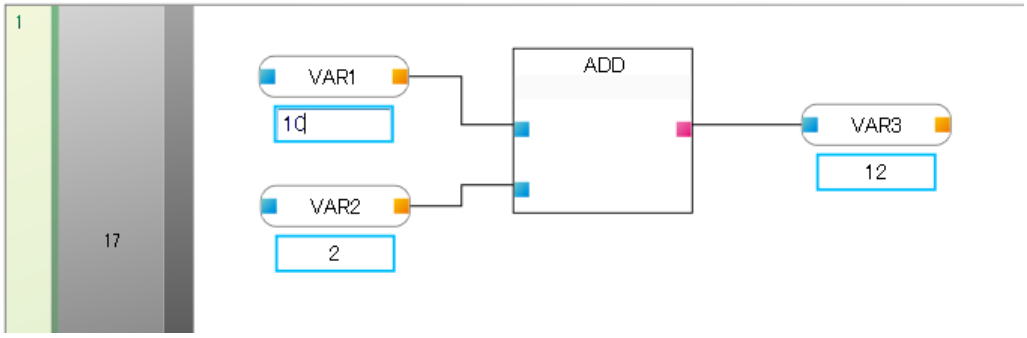
(NOTE 1) When changing OM values online, be sure to take the necessary safety precautions. If an invalid value is set, the device that has the NC(BUILT IN PLC) installed could malfunction.
(NOTE 2) This function cannot be used to change the set values for the timer and counter.
(NOTE 3) An OM with an index cannot be specified when changing the current value.
(NOTE 4) A digit-specified bit OM cannot be specified when changing the current value.
(NOTE 5) This function cannot be used to change the value of a variable.
(NOTE 6) When entering a real type (real number), the user can select 32-bit or 64-bit for the data size.

11.4.7 Change Current Variable Value

This function changes the current variable value.
The user can enter a numerical value or character string into the <Current value> field in the variable editing area in order to change the current value of a variable.

Scope	Variable name	Data type	Element number	Actual value	Comment
VAR	var_BOOL	BOOL	0	0	
VAR	var_WORD	WORD	0	192	
VAR	var_INT	INT	0	125	
VAR	var_REAL	REAL	0	1.15718E-038	
VAR	var_DATE	REAL	0	0	
VAR	var_STRING	STRING	0	''	
VAR	var_ARRAY	INT	10	Display current value	

In the FBD Programming Editor, the current variable value can be changed from the program editing area.



The values (base number) that can be entered for the variable vary depending on the data type.

Data type	Input range/base number
BOOL	0, 1, FALSE, TRUE
BYTE	2#00000000~2#11111111 8#000~8#377 0~255 16#00~16#FF
WORD	2#0000000000000000~2#1111111111111111 8#000000~8#177777 -32768~32767 16#0000~16#FFFF
DWORD	2#00000000000000000000000000000000~ 2#11111111111111111111111111111111 8#00000000000~8#37777777777 -2147483648~2147483647 16#00000000~16#FFFFFFFF
INT	-32768~32767 16#0000~16#FFFF
DINT	-2147483648~2147483647 16#00000000~16#FFFFFFFF
LINT	-9223372036854775808~9223372036854775807 16#0000000000000000~16#FFFFFFFF FFFFFFFF
UINT	0~65535 16#0000~16#FFFF
UDINT	0~4294967295 16#00000000~16#FFFFFFFF
ULINT	0~18446744073709551615 16#0000000000000000~16#FFFFFFFF FFFFFFFF
REAL	-3.402823E+38~-1.401298E-45, 0.0, 1.401298E-45~3.402823E+38 (Valid digits: 7 digits)
LREAL	-1.79769313486231E+308~-4.94065645841247E-324, 0.0, 4.94065645841247E-324~1.79769313486231E+308 (Valid digits: 15 digits)
TIME	T#000:00:00~T#255:59:59
DATE	D#1970-01-01~D#2037-12-31

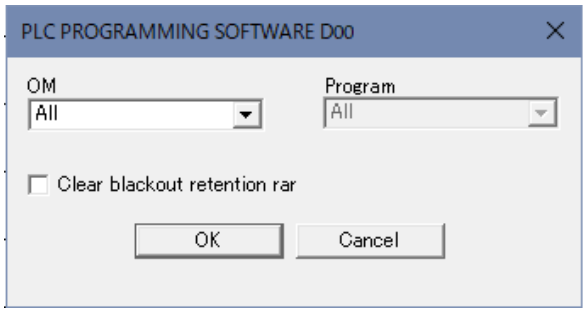
Data type	Input range/base number
TOD	TOD#00:00:00~TOD#23:59:59
DT	DT#1970-01-01-00:00:00~DT#2037-12-31-23:59:59
STRING	0 to 31 characters (half-width alphanumeric characters and symbols only)

Refer to “9.11.2 Monitoring & Changing Current Variable Values” for details on the operations in the variable editing area.

(NOTE 1) When changing variable values online, be sure to take the necessary safety precautions. If an invalid value is set, the device that has the NC(BUILT IN PLC) installed could malfunction.
(NOTE 2) Even if this function is used to change the current value, the variable value will ultimately change based on the program operation.
(NOTE 3) Refer to “11.4.6 Change current OM value” for details about changing the current OM values.

11.4.8 Clear all OMs & Variables

This function resets the current value back to the default value for all variables and OMs (except X, Y, BX, BY, BDX, BDY, SM, SD and L).
In the Clear all OMs/variables dialog below, the user can select whether to clear the global OMs or to clear the local OMs in a particular control program.



When this function is executed, the OM resets to the default value 0 (zero). The variable default value is the value that is set in the PLC Programming software D00 EDITOR.
The set values for the timer and the counter cannot be cleared.

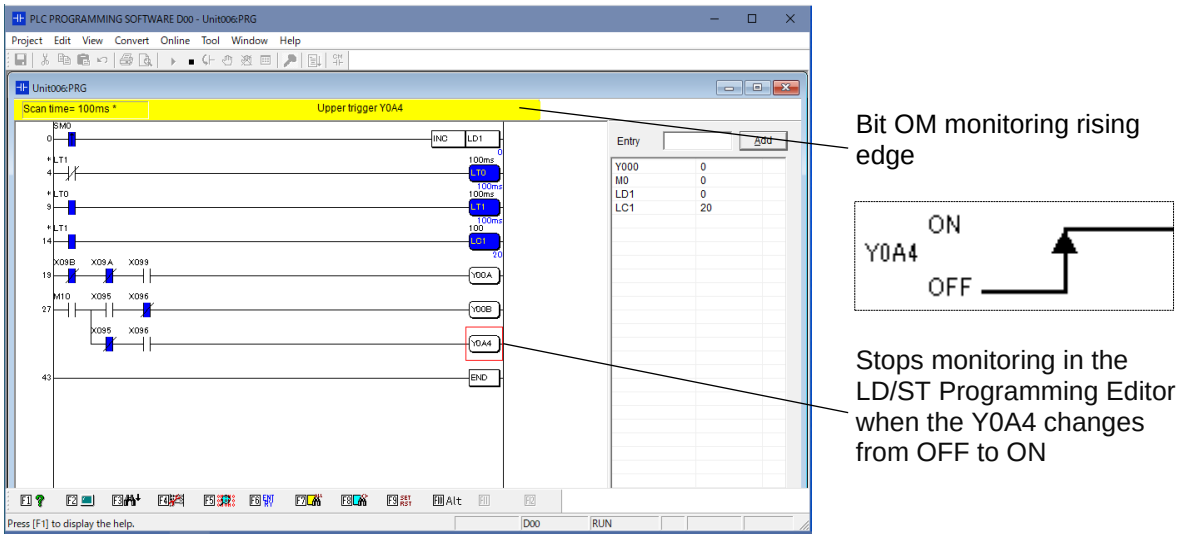
(NOTE) When the NC(BUILT IN PLC) is in a RUN state, the all clear operation cannot be executed. When the user needs to use the clear all operation, change the mode to a STOP state before executing the operation.

Refer to “7.12.6 Clear all OMs & Variables”, “8.11.7 Clear All OMs & Variables” or “9.11.4 Clear All OMs & Variables” for details on clearing all OMs/variables.

11.4.9 Rising Edge Trigger & Falling Edge Trigger

This function stops the monitoring in the PLC Programming software D00 LD/ST Programming Editor or in the FBD Programming Editor when the specified bit OM rises (OFF → ON) or when it falls (ON → OFF). The user can check the status of the NC(BUILT IN PLC) when the specified OM changes.

Ex: Set rising edge trigger to Y0A4 when LD program is monitored



This function temporarily stops or pauses monitoring in the PLC Programming software D00 LD/ST Programming Editor or in the FBD Programming Editor. This function does not end or exit monitoring, nor does it stop the operation itself in the NC(BUILT IN PLC). When the monitoring is re-opened, the user can check the most recent operation status of the NC(BUILT IN PLC).

The rising and falling edge trigger settings are shared settings throughout the PLC Programming software D00 EDITOR, regardless of the monitoring POUs. For example, after setting a trigger in the monitoring of one POU, when a trigger is set in another POU, the latter trigger setting becomes valid. In the same way, after setting one trigger and then clearing the trigger in another POU, the former trigger setting is cleared.

The following operations clear the rising and falling edge trigger settings.

- Starting / Re-opening monitoring
- Pressing the **[ESC]** key / **[F2]** key
  (Pressing the **[F3]** key when "Change monitoring" in the option setting is set to the **[F3]** key)
- End monitoring
- When monitoring is temporarily stopped or paused with this function

Refer to "7.13 DEBUG Functions", "8.12 DEBUG Function" or "9.12 DEBUG Function" for details on setting the rising and falling edge triggers and related operations like re-opening the monitoring.

(NOTE 1) In an LD or FBD program, the user can set a bit OM being used in the program as a monitoring OM.
In an ST program, the user can set a given bit OM as a monitoring OM.

(NOTE 2) The monitoring OM cannot be specified with this function for word OMs and variables.

(NOTE 3) This function is valid when the NC(BUILT IN PLC) is in a STOP or DEBUG-RUN state.

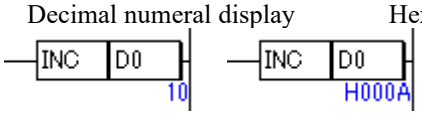
11.4.10 Change Monitoring Base Number

The current value for the OM or variable is displayed when the PLC Programming software D00 EDITOR is monitoring the NC(BUILT IN PLC).
The user can change that current value to display as a decimal or hexadecimal.
The base number can be changed for the current OM or variable value that is displayed in the following areas.

Tool / Language	Applicable areas
PLC Programming software D00 LD/ST Programming Editor When monitoring an LD program	Program editing area Registration monitoring
PLC Programming software D00 LD/ST Programming Editor When monitoring an ST program	Variable editing area Program editing area Registration monitoring
FBD Programming Editor When monitoring an FBD program	Variable editing area Program editing area

When using an LD program and displaying in hexadecimals, there is an “H” that is added at the beginning of the current value in the display.

Ex: Decimal numeral display Hexadecimal number display



Note that decimals are displayed for the following regardless of the base number setting for monitoring.

- Timer current value and timer set value
- Counter current value and counter set value
- Current variable value for data types that cannot be expressed in decimals or hexadecimals (BOOL/REAL/LREAL/TIME/TOD/DATE/DT/STRING)

11.4.11 Restrictions

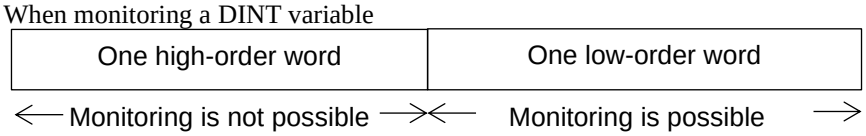
The following restrictions apply to the monitoring function in the PLC Programming software D00 EDITOR.
When the PLC Programming software D00 EDITOR is used to monitor the NC(BUILT IN PLC), up to 256 OMs and variables can be monitored at the same time. (*1) (*2)
0 (zero) is displayed for the current value, regardless of the actual value, for any variables or OMs that exceed that maximum. (*3) (*4)
These restrictions only apply to the display, and the operation results are correctly stored in the OMs and variables.

*1 The monitoring points are counted in word units. The count number varies depending on the OM and variable data type.

OM/Variable data type	Count
Bit OM	Counts the range from the beginning address and next consecutive 15 addresses as one unit. Ex: When setting the monitoring to M0 and M15, all addresses from M0 to M15 are counted as one unit. When setting the monitoring to M0 and M16, all addresses from M0 to M15 (first set) and M16 to M31 (next set) are counted as two units.
Word OM	Counts each OM as one unit.
BOOL	Counts 16 variables as one unit.
BYTE, WORD, INT, UINT	Counts each variable as 1 units.
DWORD, DINT, UDINT, REAL	Counts each variable as 2 units.
LINT, ULINT, LREAL, TIME, DATE, TOD, DT	Counts each variable as 4 units.
STRING	Counts each variable as 16 units.

*2 The number of items being monitored in batch monitoring is not included in this count. The correct values are displayed in batch monitoring for OMs that display 0 (zero) in the PLC Programming software D00 EDITOR.

*3 When monitoring data for 2 words or more, only part of the data unit may be monitored depending on the usage conditions of the OM or variable.
Ex: When monitoring a DINT variable



When this variable value is 16#AAAABBBB, the current value is displayed as 16#0000BBBB in the PLC Programming software D00 EDITOR.

*4 Data type variables that handle non-numerical values are displayed below.
DATE type: D#1970-01-01, TIME type: T#00:00:00, TOD type: TOD#00:00:00,
DT type: DT#1970-01-01-00:00:00, STRING type: "" (No character string)

11.5 Debug Function

When the specified conditions are met, program execution can be stopped in the NC(BUILT IN PLC). This function is called the debug function.
The debug function is equipped with the following functions.

Function	Description
Step execution	This function executes 1 command at a time in the program. This can be executed when the NC(BUILT IN PLC) is in a STOP state.
Program execution stop using breakpoint/stop step	This function executes the program until the specified location and then it switches the NC(BUILT IN PLC) to a STOP state. This can be executed when the NC(BUILT IN PLC) is in a DEBUG-RUN state.
Program execution stop using specified condition (LD program only)	When the specified OM values changes, this function switches the NC(BUILT IN PLC) to a STOP state. This can be executed when the NC(BUILT IN PLC) is in a DEBUG-RUN state.

11.5.1 Start & End Debug Function

This debug function can be executed when the NC(BUILT IN PLC) is in a STOP or DEBUG-RUN state.

Mode of NC(BUILT IN PLC)	Function that can be executed
STOP state	Step execution
DEBUG-RUN state	Program execution stop using breakpoint/stop step
	Program execution stop using specified condition (LD program only)

To end the debug function, the NC(BUILT IN PLC) must be set to a RUN state. At this time, the set breakpoints, stop steps and the program execution stop conditions are all cleared.

When this debug function is used, the operation mode of the NC(BUILT IN PLC) can be switched remotely in the PLC Programming software D00 EDITOR.
Refer to “7.13 DEBUG Functions”, “8.12 DEBUG Function” or “9.12 DEBUG Function” for details on how to change the operation mode or state of the NC(BUILT IN PLC).

(NOTE) To use the debug function, use the PLC Programming software D00 EDITOR to monitor a program that is written to the NC(BUILT IN PLC).
The user can use the PLC Programming software D00 LD/ST Programming Editor and FBD Programming Editor to open a program that is not written to the NC(BUILT IN PLC) and switch to monitor mode. However, the displayed content and the actual operation will not match.

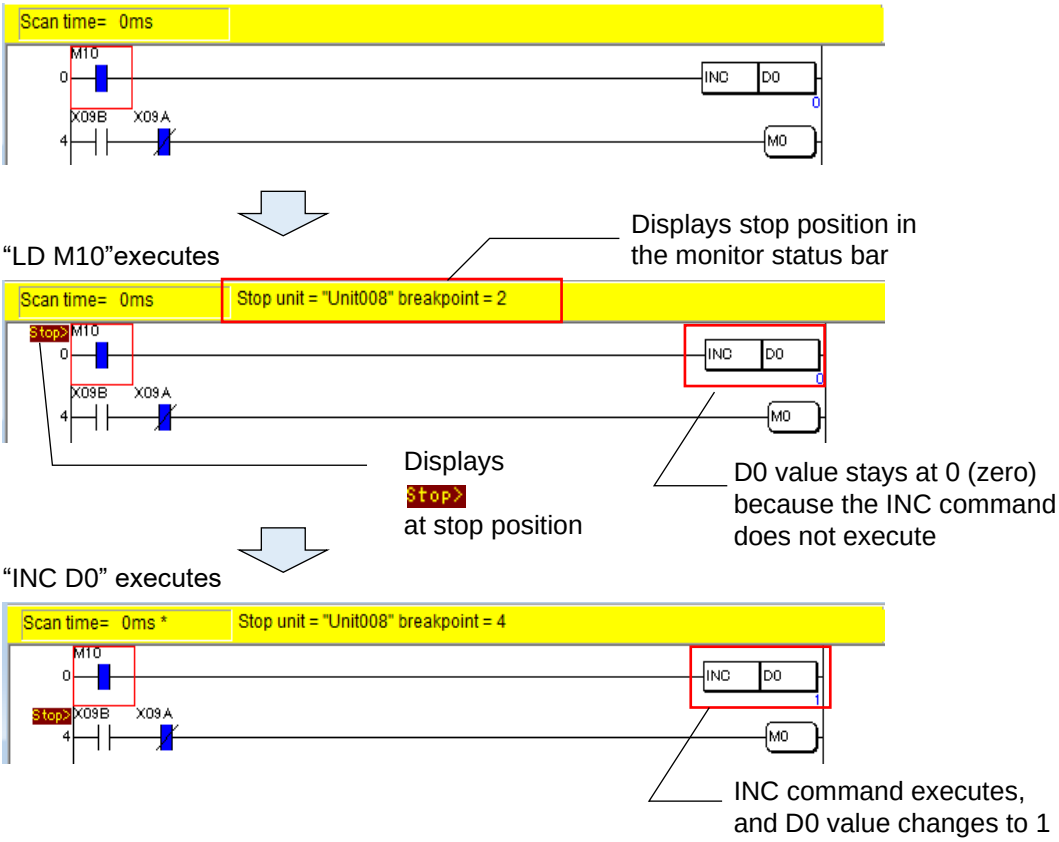
11.5.2 Step Execution

In step execution, a program can be executed using the following units or intervals.
The user can check the OM or variable change per process.

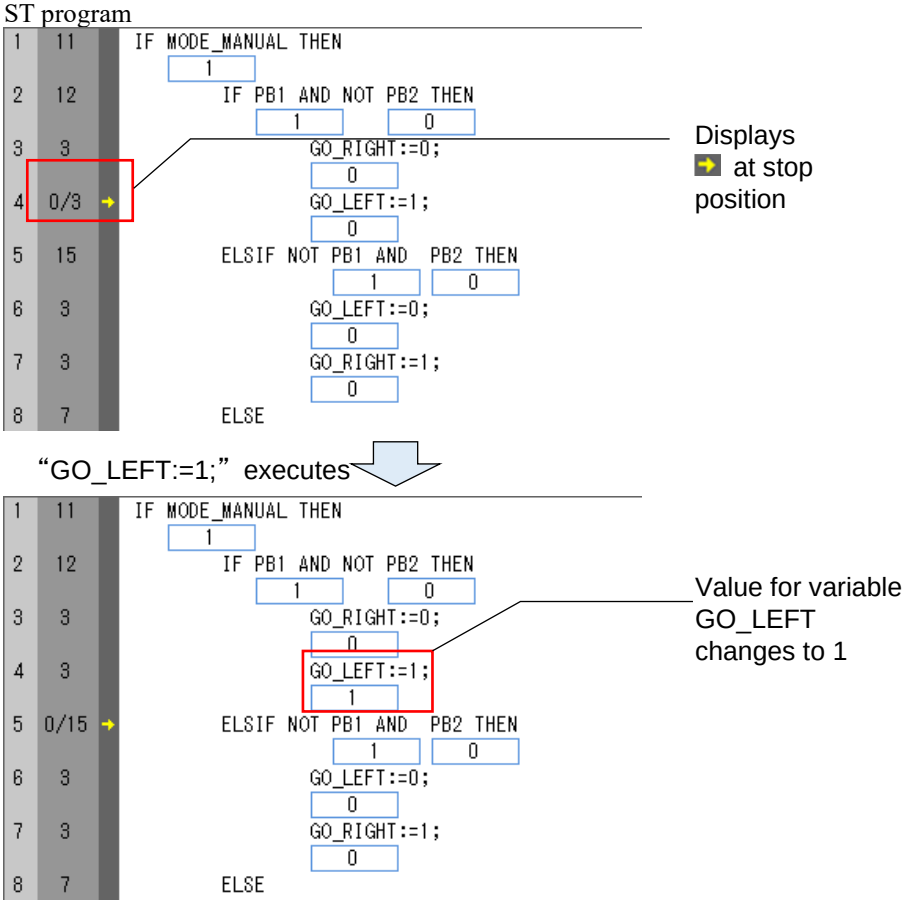
Language	Process carried out with one step execution
LD program	Executes one command at a time in the program.
ST program	Executes one command at a time in the program.
FBD program	Executes one command at a time in the program.

Ex: LD program

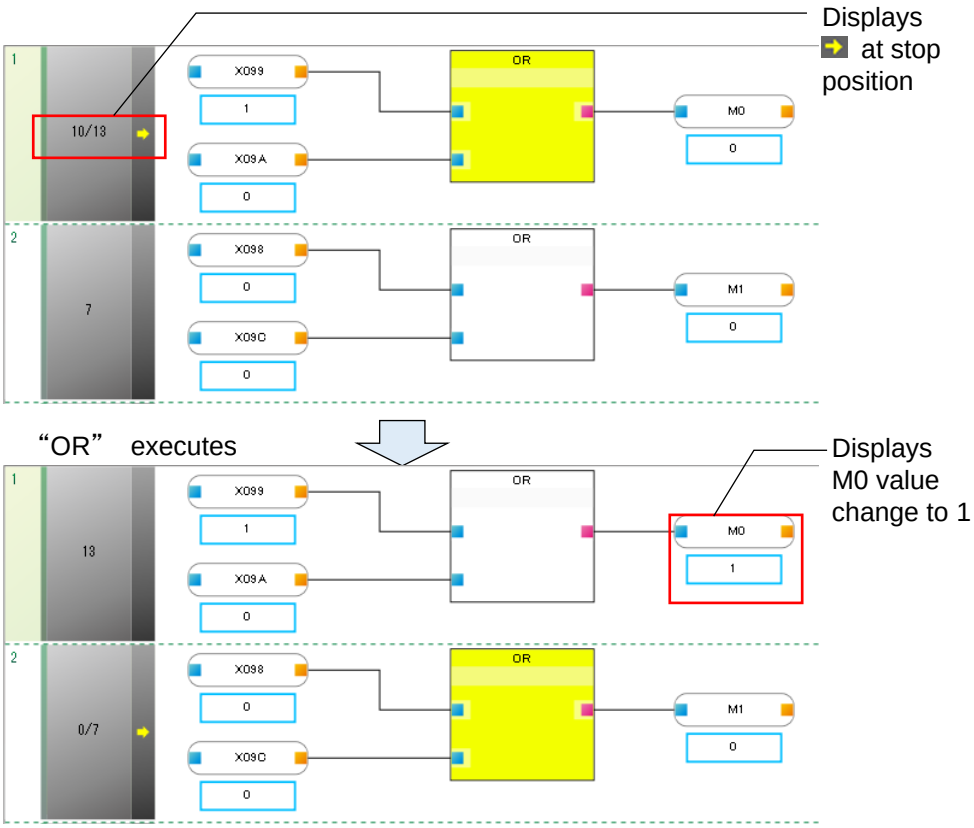
Start step execution



Ex: ST program



Ex: FBD program



Step execution is possible when the NC(BUILT IN PLC) is in a STOP state.
The PLC Programming software D00 EDITOR is used to carry out step execution.
Refer to “7.13 DEBUG Functions”, “8.12 DEBUG Function” or “9.12 DEBUG Function” for details on how to use the PLC Programming software D00 EDITOR.

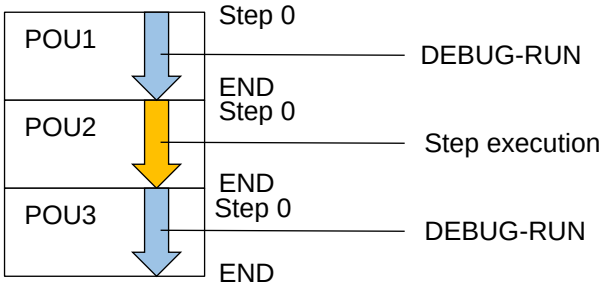
(NOTE 1) Step execution is carried out from the following positions.
- When the user changes the NC(BUILT IN PLC) to a STOP state manually, execution starts from step 0.
- If the NC(BUILT IN PLC) is already in a STOP state as a result from other stop conditions, then execution starts from that position. However, when a bit trigger or word trigger (described later) is used to change it to a STOP state, then step execution is not possible.

(NOTE 2) When multiple programs are running in the NC(BUILT IN PLC), the stop step or breakpoint in the monitor status bar is displayed starting with step 0 for the beginning of the program (where it is currently stopped).

(NOTE 3) Step execution is not possible on the monitoring screen for a CALC command.

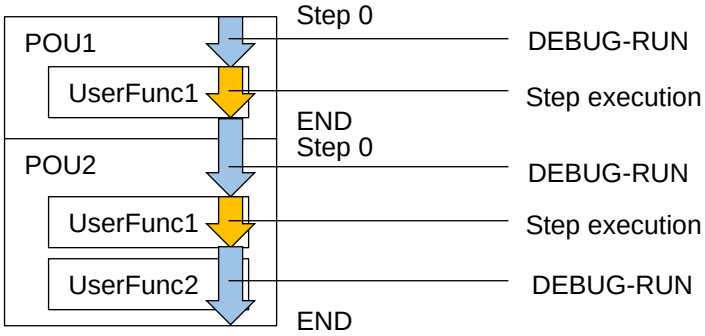
When step execution is used with multiple programs running in the NC(BUILT IN PLC), step execution is carried out in the program being monitored at the forefront of the PLC Programming software D00 EDITOR.
If there are too many processes active for other programs, when the program is executed in DEBUG-RUN, it changes to a STOP state if the processing goes back to the program being monitored.

Ex: Programs are executed in the order POU1, POU2 and then POU3. Step execution when POU2 is being monitored in the PLC Programming software D00 EDITOR.



When step execution is used for a user or system defined function, the processing for other POUs is carried out in DEBUG-RUN until the user or system defined function is called.
When step execution is used for the processing of a user or system defined function, after the processing returns back to the caller POU (calling source), it operates in DEBUG-RUN until the user or system defined function is called again.

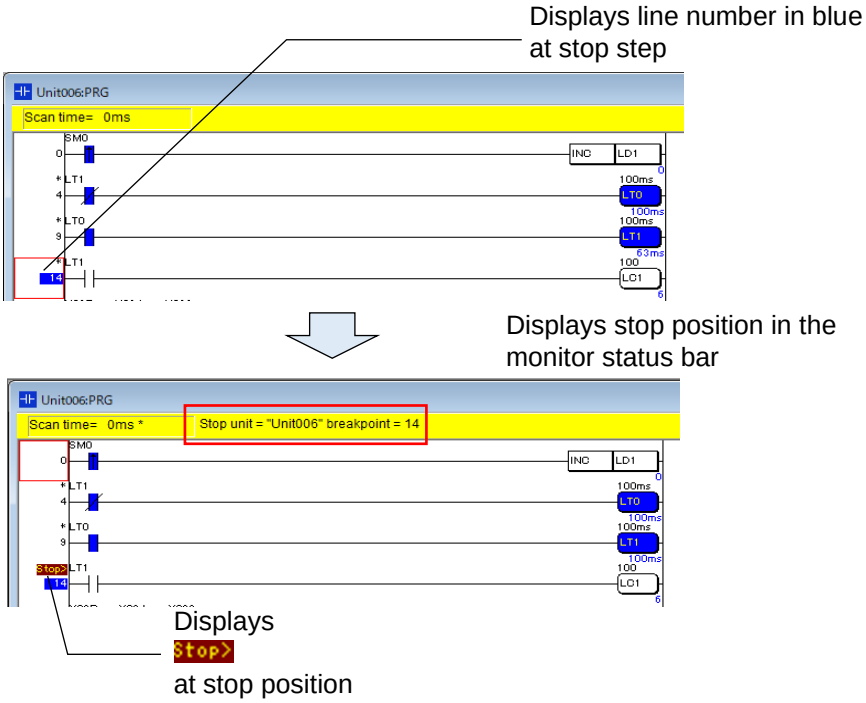
Ex: There is POU1 and POU2. User defined function “UserFunc1” is called. Step execution is used when monitoring UserFunc1 in PLC Programming software D00 EDITOR.



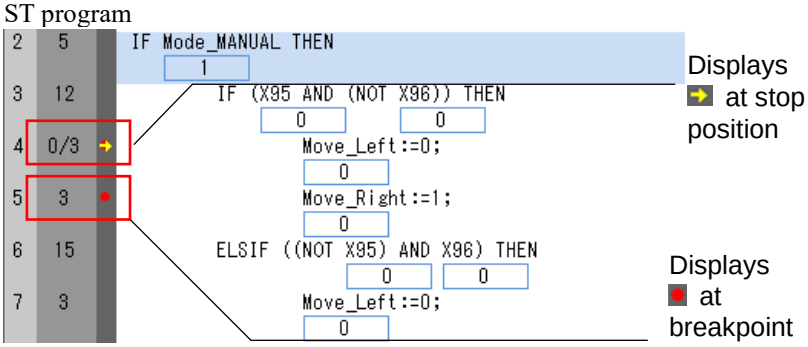
11.5.3 Program Execution Stop Using Breakpoint/Stop Step

If the NC(BUILT IN PLC) is in a DEBUG-RUN state, when a program is executed up to a specified location, the NC(BUILT IN PLC) changes to a STOP state.
The user can check the results when a program is executed up to a specified location.
The position where this program stops is called a breakpoint or stop step (LD program).
A breakpoint or stop step can be set at the beginning of a line (ST), at the beginning of a network (FBD) or at the beginning of a circuit block (LD).
The user can set up to 16 breakpoints or stop steps in the program running in the NC(BUILT IN PLC).

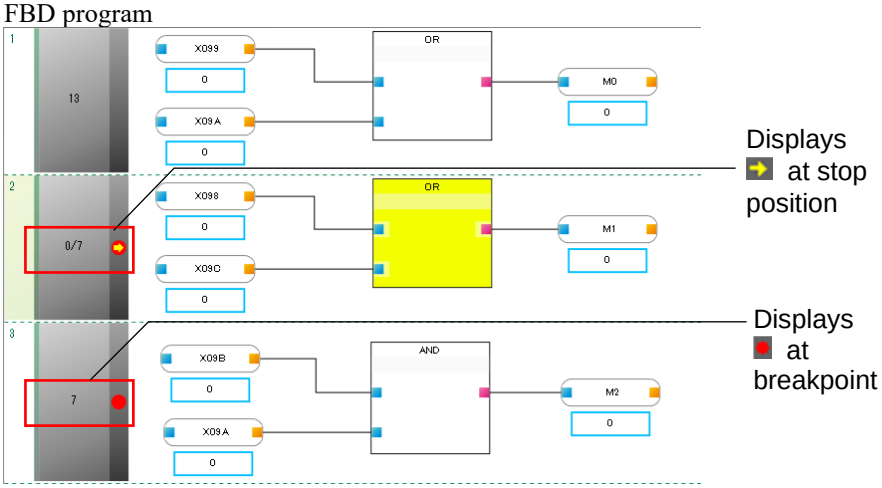
Ex: LD program



Ex: ST program

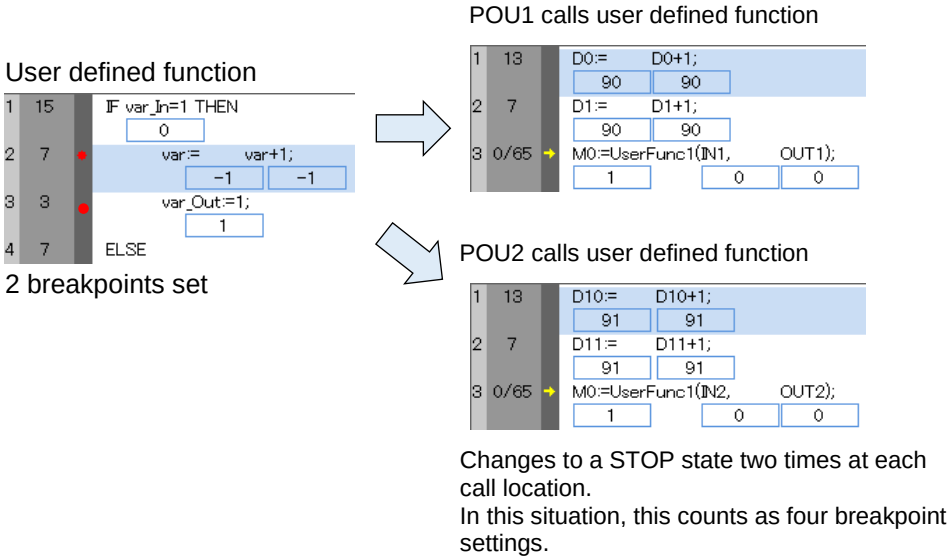


Ex: FBD program



When a breakpoint is set to the user or system defined function, the NC(BUILT IN PLC) changes to a STOP state every time that the user or system defined function is called. Even when the user or system defined function is used from multiple POUs, the NC(BUILT IN PLC) changes to a STOP state every time that the user or system defined function is called.
When multiple breakpoints are set to the user or system defined function, the NC(BUILT IN PLC) changes to a STOP state every time that the user or system defined function is called and the breakpoint location is processed.

Ex:



The PLC Programming software D00 EDITOR is used to set or clear a breakpoint and stop step. Refer to “7.13 DEBUG Functions”, “8.12 DEBUG Function” or “9.12 DEBUG Function” for details on how to use the PLC Programming software D00 EDITOR.

11.5.4 Stop Using Specified Condition (LD Language)

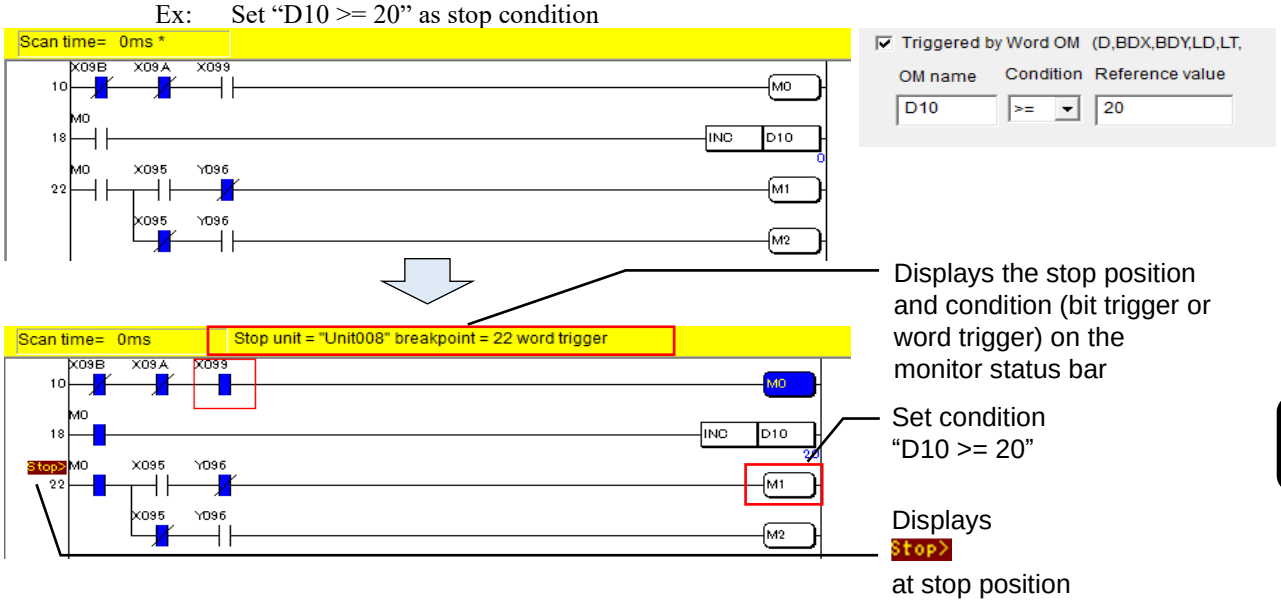
When an LD program is being executed and the NC(BUILT IN PLC) is in a DEBUG-RUN state, an OM value change can be used as a trigger to change the NC(BUILT IN PLC) to a STOP state. This function is valid when executing an LD program. It cannot be used when executing an ST or FBD program.
The following conditions can be set in the PLC Programming software D00 EDITOR.

Setting		Timing when NC(BUILT IN PLC) changes to a STOP state
Bit OM trigger	OFF → ON	When the specified bit OM changes from OFF to ON
	ON → OFF	When the specified bit OM changes from ON to OFF
	ON or OFF change	When the specified bit OM value changes
Word OM trigger	=	When the specified word OM and compared value match
	< >	When the specified word OM and compared value do not match
	>	When the specified word OM is greater than the compared value
	>=	When the specified word OM is equal to or greater than the compared value
	<	When the specified word OM is less than the compared value
	<=	When the specified word OM is equal to or less than the compared value

A stop condition can be set for the following OMs, etc.

Item	Description
Bit OM that can be set	X, Y, BX, BY, LM, LTS, LTC, LCS, LCC, SM
Word OM that can be set	D, LD, LT, LC, SD, IX
Numerical value that can be set as a compared value	-32768~32767 (Decimal numeral)
Quantity that can be set	Only one OM and one condition can be set as a “Bit OM trigger” and “Word OM trigger”.

Ex: Set “D10 >= 20” as stop condition



The PLC Programming software D00 EDITOR is used to set the stop condition.
Refer to “7.13 DEBUG Functions” for details on how to use the PLC Programming software D00 EDITOR.

(NOTE 1) The following OMs cannot be set as a bit OM trigger or word OM trigger.
- OM where an index is specified
- Bit OM where the digits are specified
- Word OM where a longword is specified

(NOTE 2) A variable change cannot be set as a stop condition.
(NOTE 3) The stop condition setting can be executed when the NC(BUILT IN PLC) is in a STOP or DEBUG-RUN state.

11.6 Simulation Function

The simulation function checks the operation of programs (created in the PLC Programming software D00 EDITOR) on a computer, without connecting to the NC(BUILT IN PLC).
The user can test offline whether the program operation follows the designed logic or not.

(NOTE) The simulation function can be used to confirm whether the logic for the program is correct or not. However, there is a difference between the function and actual operation in the NC(BUILT IN PLC), and therefore the scan time, timing and/or operation may not necessarily match. Before starting the operation of a device with the NC(BUILT IN PLC), always be sure to thoroughly check the operation on the actual machine.

Similarly to when the PLC Programming software D00 EDITOR is used to monitor the NC(BUILT IN PLC), the following functions are also available during simulation execution.

Function	Description
Write program	Writes the program to the simulation function in order to check operation.
Program execution	Executes the program that is written to the simulation function.
RUN/STOP mode	The operation mode (RUN/STOP/DEBUG-RUN) can be changed in the simulation function.
Batch monitor	Lists the current values of the OMs for each OM type.
Monitoring registration	Constantly displays the current value of the specified OM.
OM SET/RESET	Changes current value of bit OM.
Change current OM value	Changes current value of word OM.
Change current variable value	Changes current value of variable.
Clear all OMs	Resets current OM values back to their default values.
Rising edge/Falling edge trigger	Stops monitoring at the rising or falling edge of the specified OM.
Debug function	The following function can be used when the simulation function is in a STOP state. • Step execution The following functions can be used when the simulation function is in a DEBUG-RUN state. • Breakpoint stop • Stop using specified condition

Note that the following shows some differences between the NC(BUILT IN PLC) and the simulation function.

Item	NC(BUILT IN PLC)	Simulation function
Scan time	Follows the specified scan time from the parameter settings in the PLC Programming software D00 EDITOR.	Fixed at 100 ms.
WDT monitor time	Follows the specified WDT monitor time from the parameter settings in the PLC Programming software D00 EDITOR.	Fixed at 5000 ms.
Operation when there is an error	Follows the specified operation from the parameter settings in the PLC Programming software D00 EDITOR, and changes the PLC Programming software D00 EDITOR to a STOP state or RUN state.	When an error occurs, simulation stops (STOP state).
Input refresh and output refresh	Input from an external device and output to an external device is carried out in the scan processing.	The input and output between external devices is not carried out because it is offline. “Change current OM value” is used to turn “X” ON or OFF (Refer to “11.4.6 Change current OM value”). When the value is changed, that change is reflected in the program operation.
Read program and write during run	PLC Programming software D00 EDITOR can be used to read a program or perform a write during run operation.	Read program and write during run operation are not supported.
Counting up current timer value	Current timer value counts up (increases) based on the actual measured scan time after the input time is carried out.	Selects from one of the following operations. • Counts up based on the actual measured scan time after the input time is carried out. • User sets the time count per scan. (Refer to “11.6 Simulation function” for further details.)

(NOTE) The scan time, WDT monitor time and error operation settings are configured in the “Parameter settings” in the PLC Programming software D00 EDITOR. The parameter settings are written to the NC(BUILT IN PLC) when the program is written.
However, the parameter settings are not written to the simulation function. The simulation function uses fixed values for the scan time, WDT monitor time and error operation, regardless of the parameter settings.

In the simulation function, the user can specify the following count up methods for the current timer value.

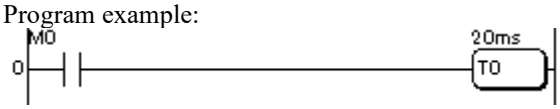
- Counts up based on the actual measured scan time after the input time is carried out.
- User sets the time count per scan.

The former is the same operation as the NC(BUILT IN PLC).
To use the latter, access the <Option setting> in the PLC Programming software D00 EDITOR and enable the <Perform the count up operation on the timer per scan> setting. Then, specify between 1 and 100 ms for the <Count up time per scan>.

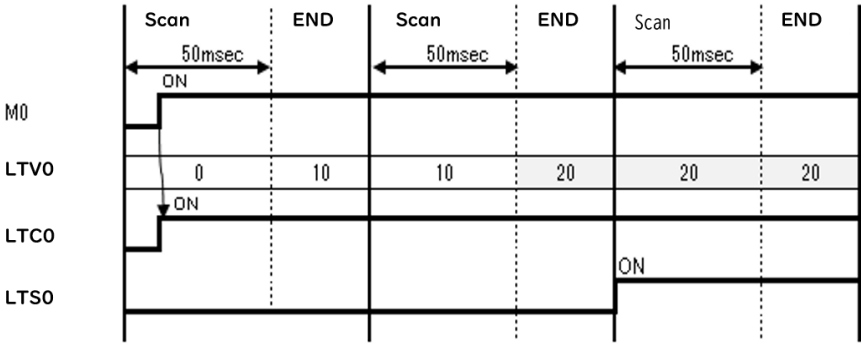
(NOTE) The option settings (<Perform the count up operation on the timer per scan> and <Count up time per scan>) are enabled when a program is written to the simulation function.
The option settings are not written when a program is written to the NC(BUILT IN PLC).

The differences in the operations are shown in the timing charts below.
When using the LD language

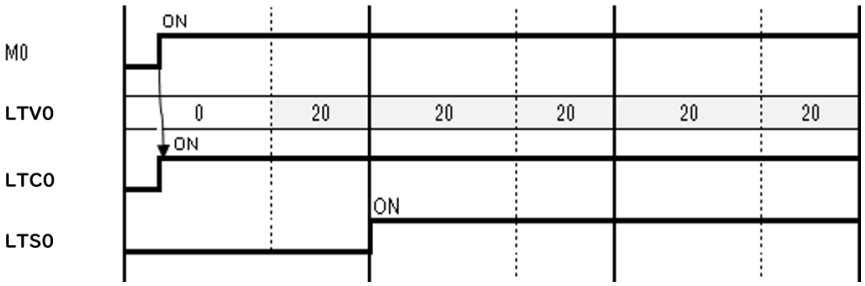
Program example:



When user sets the time count per scan.
(Count up time is set to 10 ms)



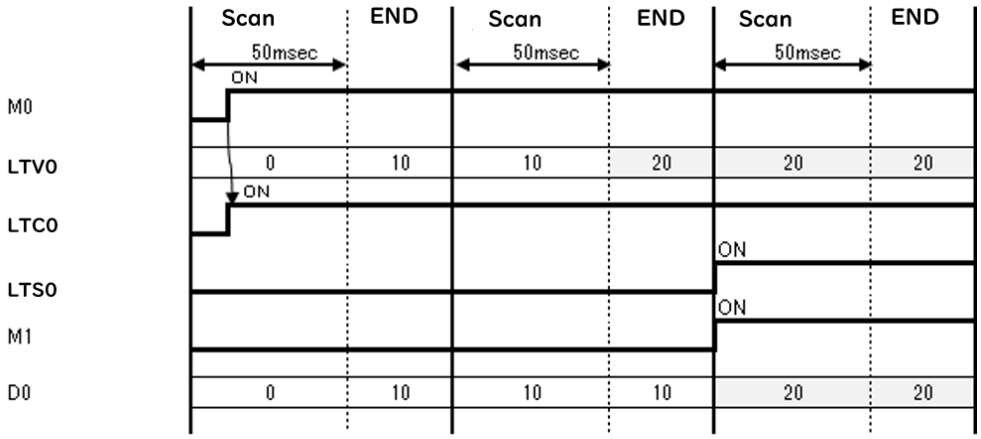
When counting up based on the actual measured scan time after the input time is carried out.



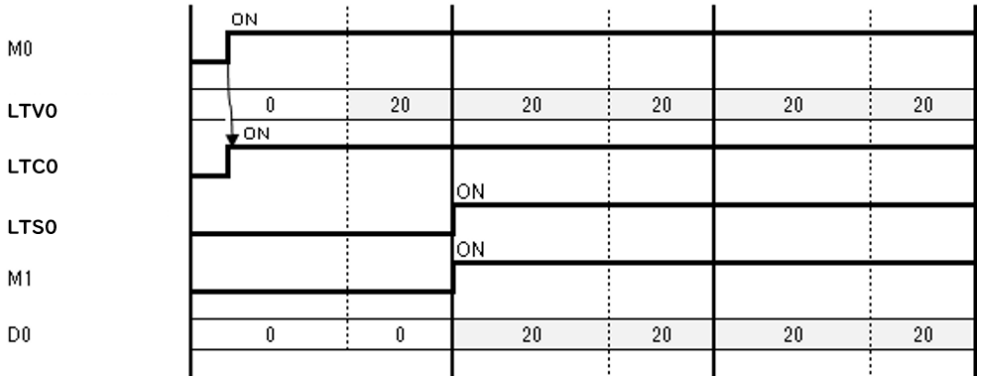
When using the ST/FBD language

Program example:
M1 := TON(M0,2,0,D0);

When user sets the time count per scan.
(Count up time is set to 10 ms)



When counting up based on the actual measured scan time after the input time is carried out.



CHAPTER 12

COMMANDS (FBD)

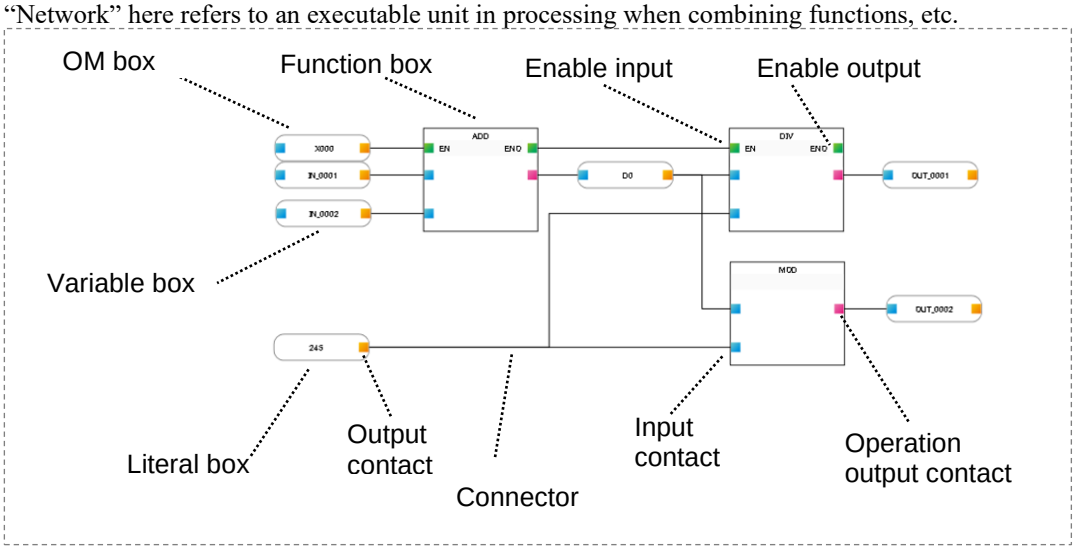
12.1 FBD Language Specifications
12.2 FBD Language Standard Functions
12.3 Error Message List During FBD Program Conversion

12.1 FBD Language Specifications

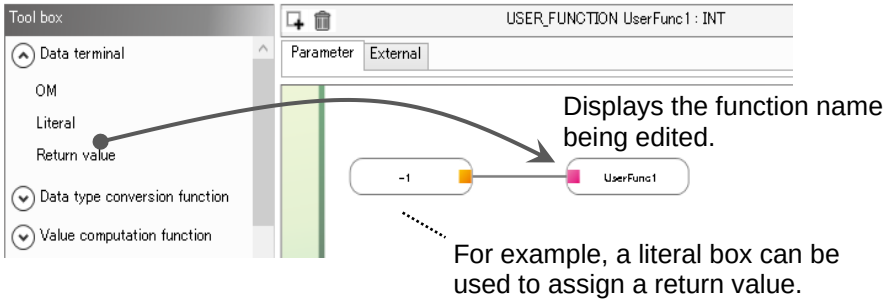
The section here describes the requirements when using the PLC Programming software DOO for programming in the FBD language.

12.1.1 Create FBD Programs

An FBD program can be created by setting up different components on a network, such as a "Function box", "Variable box", "OM box", "Literal box" and "Return value box", and their inputs and outputs are connected via a "Connector".
"Network" here refers to an executable unit in processing when combining functions, etc.



The function box specifies the executing process. The user or system defined function created by the user can also be used.
The variables, OMs and literals (constants) used for the function input and output are set in the variable box, OM box and literal box.
The return value box is used to specify the return value in a user or system defined function.

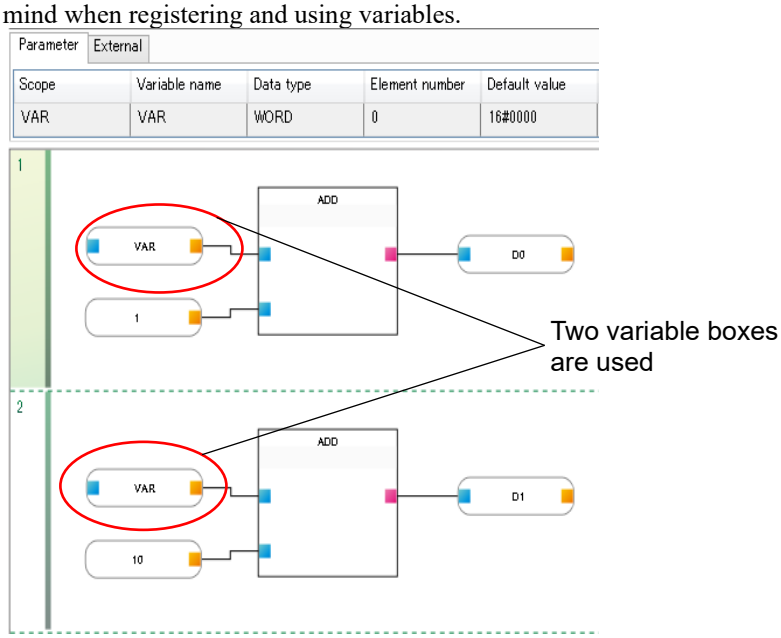


The following table shows the maximum number of networks, boxes or connectors that can be input into one POU.

Item	Maximum quantity
Network	256
Function box	256
Variable box	1024
OM box	1024
Literal box	1024
Return value box	256
Connector	4096

Refer to "Chapter 9 FBD Edit Operation Procedure" for details on how to use the FBD Programming Editor.

(NOTE 1) For example, when an FBD program is created like in the figure shown below, there is one variable being used ("VAR") and there are two variable boxes being used.
In one POU, the user can register a maximum of 1024 BOOL type variables as well as a maximum of 1024 non-BOOL type variables. However, the user can only use a maximum of 1024 variable boxes in an FBD program. Therefore, keep this point in mind when registering and using variables.

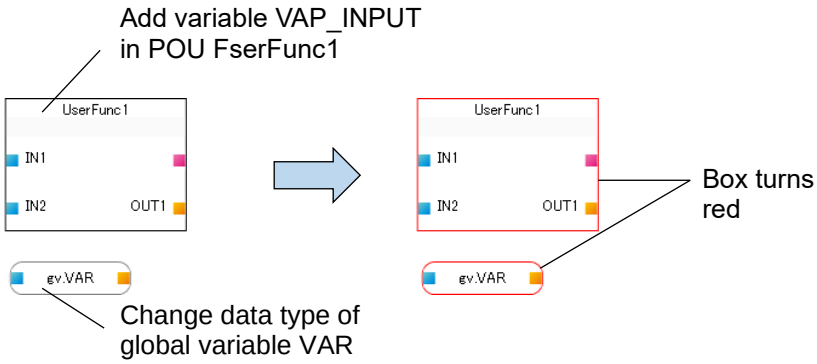


The same applies to the OM box.

(NOTE 2) After setting up a variable, user defined function or system defined function box in the program area, if the user performs one of the following operations, then the box turns red.

- Change variable data type or element number
- Setting change or deletion of a global variable that an external variable references
- User and system defined functions:
  - Change data type of return value
  - Add or delete variables in the scope: VAR_INPUT/VAR_OUTPUT/VAR_IN_OUT
  - Change data type or element number for the variable in the scope: VAR_INPUT/VAR_OUTPUT/VAR_IN_OUT
  - Delete user defined function / system defined function

Ex



Conversion is not possible when there is a red box.
If there is a red box, perform one of the following operations.
- Delete the red box and correct the program.
- Re-assign the red box.

When a box is re-assigned, the settings for the variable, user defined function or system defined function with the same name as the variable name or POU name in the current box are reflected in that re-assigned box.
Refer to “9.3.1.11 Re-assign box” for details on re-assigning a box.

12.1.2 Variables & OMs

In an FBD program, variables and OMs can be used as a “container” to store or hold data.

12.1.2.1 List of data types

The following data types can be used in an FBD program.

Type	Data type	Bit size	Range	Default value
Boolean	BOOL	1	FALSE(0), TRUE(1)	FALSE
Bit stream	BYTE	8	16#00~16#FF	16#00
	WORD	16	16#0000~16#FFFF	16#0000
	DWORD	32	16#00000000~16#FFFFFFFF	16#00000000
Signed integer	INT	16	-32768~32767	0
	DINT	32	-2147483648~2147483647	0
	LINT	64	-9223372036854775808~ 9223372036854775807	0
Unsigned integer	UINT	16	0~65535	0
	UDINT	32	0~4294967295	0
	ULINT	64	0~18446744073709551615	0
Single-precision real number	REAL	32	-3.402823E+38~ -1.401298E-45, 0.0, 1.401298E-45~3.402823E+38	0.0
Double-precision real number	LREAL	64	-1.79769313486231E+308~ -4.94065645841247E-324, 0.0, 4.94065645841247E-324~ 1.79769313486231E+308	0.0
Time	TIME	64	T#000:00:00~T#255:59:59	T#000:00:00
Date	DATE	64	D#1970-01-01~D#2037-12-31	D#1970-01-01
Time of day	TOD	64	TOD#00:00:00~TOD#23:59:59	TOD#00:00:00
Date and time	DT	64	DT#1970-01-01-00:00:00~ DT#2037-12-31-23:59:59	DT#1970-01-01-00:00:00
Character string	STRING	256	0 to 31 characters	(No characters)

Hexadecimal numbers, unsigned decimal numbers, binary numbers and octal numbers can be used to store numerical values for BYTE, WORD and DWORD variables.
Signed decimal numbers and hexadecimal numbers can be used to store numerical values for INT, DINT and LINT variables.
Unsigned decimal numbers and hexadecimal numbers can be used to store numerical values for UINT, UDINT and ULINT variables.

12.1.2.2 Usable OMs

The following OMs can be used in an FBD program.

Bit									Word								Label	Nesting
X	Y	M	BX	BY	LM	SM	LT*1	LC*1	D	BDX	BDY	LD	LT*1	LC*1	SD	IX	L	N
○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-

*1 LTS, LTC, LTV, LTP, LCS, LCC, LCV and LCP also cannot be used.

The following data types are used by the corresponding OMs.

OM	Data type
X,Y, M, BX,BY,LM,SM	BOOL
X,Y, M, BX, BY, LM, SM for specifying 4-, 8-, 12-, 16-bit digits (Ex: M0:8 and M0W)	WORD
D,BDX,BDY,LD,SD,IX	WORD
X,Y, M, BX, BY, LM and SM for specifying 16-bit digits (Ex: M0W)	WORD
D, BDX, BDY, LD and SD for specifying 32-bit digits (Ex: D0L)	DWORD
X,Y, M, BX, BY, LM and SM for specifying 20-, 24-, 28-, 32-bit digits (Ex: M0:20 and M0L)	DWORD

12.1.2.3 Specifying indexing and digit-specified access for OMs

Specifying indexing refers to directly specifying an OM number. When using an index, the index is added after the OM number.
Ex M0IX0 refers to M10 when IX0 = 10.

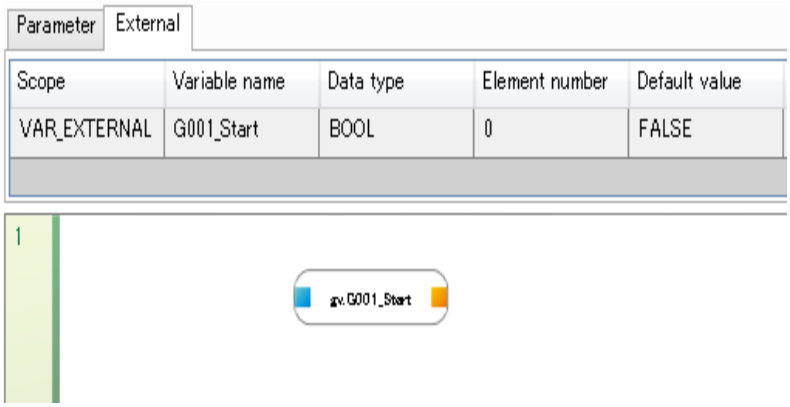
Digit-specified access refers to accessing multiple consecutive OMs at one time and using a bigger data size than what can be used with one OM point. When using digit-specified access, the symbols “W”, “L” or “:n” (n refers to the bit number) are added after the OM number.
Ex M0:8… Accesses an 8-bit segment all at once from 0 to M7.
D10L…Accesses D10 and D11.

(NOTE) Set the IX value so that the OM number does not exceed the maximum value based on index addressing.
When specifying an OM that is out of range, proceed with caution because a different type of OM may be read and written. However, if another OM value is read and written, it is read and written following the order below.
Global OM range: X → Y → M → D → BX → BY → BDX → BDY → Variable
Local OM range: LM → LD → LT → LTS → LTC → LTV (current value) → LTP (set value) → LC → LCS → LCC → LCV (current value) → LCP (set value) → SM → SD → IX
For example, when IX0 = 1, the Y7FFIX0 becomes out of range (by 1 point), and designates M0 as the OM’s first point after Y when following the sequence above.

In this way, the program created may malfunction because it accesses a different OM value. When index addressing, exercise caution when assigning the index values to be set.
Set the IX value so that the OM number does not exceed the maximum value based on index addressing.

12.1.2.4 External variable

When an external variable is set up in a network, “gv.” is automatically added before the variable name.



12.1.2.5 Array variables

Array variables in an FBD program can be used to assign other array variables or can be used in function arguments that correspond to array variables.
When assigning to array variables, the data type must be compatible for assigning to array variables and it must be assigned at an array variable with the same element number. Refer to “MOVE” under “12.2.2.3 Arithmetic operation function” for details about data types that can be used for assignment.
When using array variables in a function argument, the array variable must match with the element number and data type that are defined in that argument.

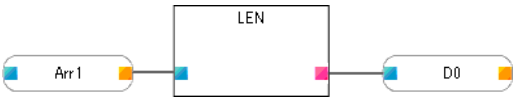
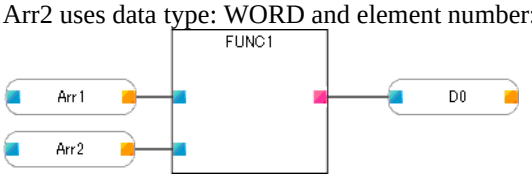
When specifying the element of an array variable, add brackets **[ ]** after the variable name to specify the element. In this case, the beginning element shall be 0.
For example, when specifying element 0 and element 1 for the array variable Arr1, it is expressed as shown in the figure below.

In addition, a variable (excluding arrays) and an OM can also be used to specify an element (in the figure below, VAR and D0 are specified). In this case, store the integer value for 0 to element number -1 in the variable and OM.

When specifying an element for an array variable, it can be used in an FBD program just like other variables (that are not array variables).

(NOTE) The standard functions available in the PLC Programming software D00 do not support array variables. If a function to process or use arrays is needed, create a user defined function or system defined function.

An array variable cannot be used in the following cases.
(In the notation examples below, Arr1, Arr2 and Arr3 are array variables, and VAR is a variable that is not an array.)

Item	Notation example (*)
Assigning items other than array variables to array variables	
Assigning array variables to data that is not an array	
Assigning array variables that include data types that cannot be assigned	(Data type of Arr1: INT, Data type of Arr2: STRING)
Assigning array variables with element numbers that do not match	(Element number of Arr1: 5, Element number of Arr2: 10)
Using array variables in arguments for functions that do not support arrays	
Using array variables with data types or element numbers that do not match, in arguments for functions that support arrays	(In FUNC1, argument No.1 is defined by data type UINT and element number 5, and argument No.2 is defined by data type WORD and element number 8. Arr1 uses data type: INT and element number: 5, and Arr2 uses data type: WORD and element number: 10.) 
Using another item besides constants, variables or OMs to specify array elements Or, using array variables to specify array elements	
Using constants or variables for other data types not listed below to specify array elements BOOL, BYTE, WORD, DWORD, INT, DINT, LINT, UINT, UDINT, ULINT	 (VAR_TIME is a TIME type variable)
Specifying out of range numerical values for elements of variable arrays	 (Element number of Arr1 is 5.)

* An error is triggered during input, conversion or program execution when the program is invalid.

The default value of an array variable is set in the PLC Programming software D00 EDITOR, but the set value becomes the default value for all elements of that array variable.
To set individual values for each element, the user must set them in the program.

12.1.3 Constants

The following table shows the notation of constants.

Item	Valid data type	Notation method	Notation example
Boolean	BOOL	TRUE/FALSE or 1/0	TRUE
Bit stream (binary numeral)	BYTE, WORD, DWORD	[2][#][Numerical value]	2#0101
Octal numeral	BYTE, WORD, DWORD	[8][#][Numerical value]	8#4723
Decimal numeral	BOOL, BYTE, WORD, DWORD, INT, DINT, LINT, UINT, UDINT, ULINT	Numerical value is noted	1234
Hexadecimal numeral	BYTE, WORD, DWORD, INT, DINT, LINT, UINT, UDINT, ULINT	[16][#][Numerical value] Enter a letter A through F in uppercase letters. (*)	16#A0F5
Real	REAL, LREAL	Numerical value is noted. To express “10 to nth power”, E is used.	12.34, 2.1E+10 (=21000000000) (+ after E can be omitted)
Character string	STRING	' ' are used to enclose the string.	'ABCDE'
Date and time	DT	[DT][#][Year-Month-Day-Hour:Minute:Second]	DT#2017-10-05-15:51:00
Date	DATE	[D][#][Year-Month-Day]	D#2017-10-05
Time of day	TOD	[TOD][#][Hour:Minute:Second]	TOD#15:51:00
Boolean	BOOL	TRUE/FALSE or 1/0	TRUE

* For example, a conversion error is triggered when a letter is entered in lowercase letters like in “16#abcd”.

The following symbols and character codes use $ in the display.

Input method	Corresponding characters/function
$$	$
$'	'
$L or $l	Line feed
$N or $n	Line break
$P or $p	Change page
$R or $r	Return
$T or $t	Tab

When using hexadecimal, octal or binary constants in an FBD program, the data is converted into unsigned decimal numerals to carry out operations.

When a constant is input that exceeds the range of the data type (ULINT type, LINT type and LREAL type) in the FBD program, then the value and operation are processed as follows in NC(BUILT IN PLC)NC(NC(BUILT IN PLC)).

- When a value that is larger than 18446744073709551615 is input using decimal numerals, the maximum value (18446744073709551615) for the ULINT type data is input and processed for the operation.
- When a value that is smaller than -9223372036854775808 is input using decimal numerals, the minimum value (-9223372036854775808) for the LINT type data is input and processed for the operation.
- When a numerical value that is outside of LREAL data type range is input using real numbers, a value of 0 is input and processed for the operation.

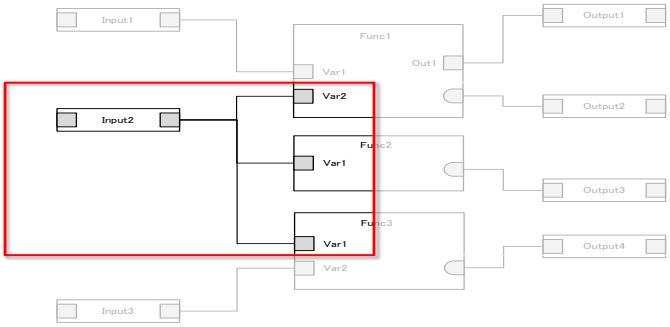
Ex When “DATA1” is a constant in ULINT type data
After the program below is executed, 18446744073709551615 is assigned to DATA1.

(NOTE) Refer to “MOVE” in “12.2.2.3 Arithmetic operation function” for details about operations when the data size does not match on the right side and left side for an assignment operation.
Refer to “12.2.2 Detailed descriptions of FBD language standard functions” for details about data types that can be used with each function when using the FBD language.

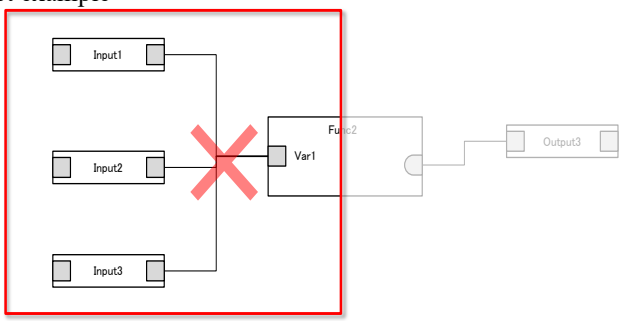
12.1.4 Input & Output Connections

The user can branch out and connect one output to multiple inputs.
However, multiple outputs cannot be merged into one input.
Examples are shown below.

Correct example



Incorrect example



Always connect the input and output for each function box to the input and output of another function box, variable box, OM box or literal box.

Correct example

Incorrect example

However, input and output connections can be omitted in user and system defined functions.

In addition, EN/ENO connections can also be omitted.

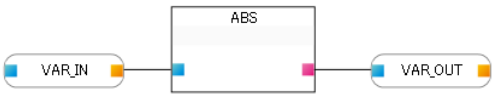
12.1.5 EN/ENO

"EN/ENO" can be used with each function box.
EN inputs the execution conditions of a function.
TRUE: Executes the function.
FALSE: Does not execute the function.
If EN/ENO is not used, the function will execute at every scan.

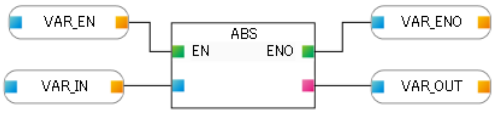
ENO outputs the output status of a function.
TRUE: Operation results of a function are output.
FALSE: Operation results of a function are not output.

Program example:

When EN/ENO is not used in ABS



When EN/ENO is used in ABS



EN/ENO and the operation result of a function are as follows.

EN	Function execution	ENO	Function output
TRUE(*2)	Function executes and no error is triggered.	TRUE	Operation result
	Function executes and error is triggered.	FALSE	Undefined (*1)
FALSE	Function does not execute.	FALSE	Undefined (*1)

*1 The output value of a function is the same as the previous time and does not change.
*2 If an EN input is omitted, the operation is normally processed as TRUE.

When EN and ENO are used, the program size increases about 15 steps per location (where EN/ENO is used).

The user can specify a variable or OM for EN and ENO. A constant can also be specified for EN. Any of the following data types can be used for EN and ENO. In addition, the table also shows the TRUE and FALSE values for EN and ENO.

	Usable data type	TRUE	FALSE
EN	BOOL	TRUE(1)	FALSE(0)
	BYTE, WORD, DWORD, INT, DINT, LINT, UINT, UDINT, ULINT, REAL, LREAL	Value other than 0 *1	0
ENO	BOOL	TRUE(1)	FALSE(0)
	BYTE, WORD, DWORD, INT, DINT, LINT, UINT, UDINT, ULINT, REAL, LREAL	Value other than 0 *2	0

*1 When REAL or LREAL is used, infinity and non-numbers are also included.
*2 ENO stores the EN value.

Refer to "MOVE" under "12.2.2.3 Arithmetic operation function" for details on the stored values when EN and ENO data types are different.

Special notes when using EN/ENO

The output on the function box (where EN/ENO is enabled) cannot be connected directly with the EN/input on another function box. This connection needs to be done through or via a variable or OM box. A direct connection is not possible because the output value becomes undefined when EN is FALSE.

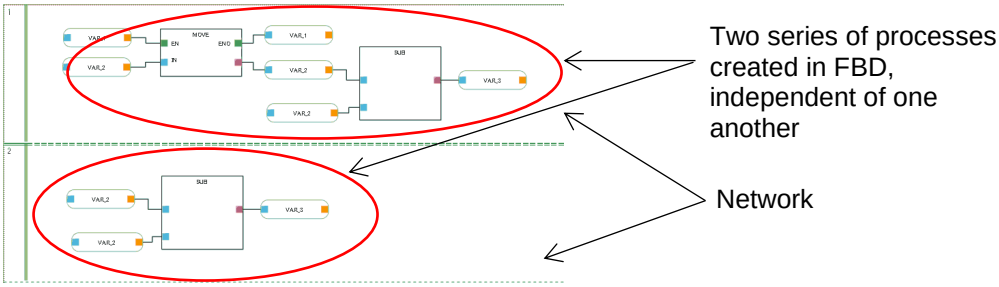
Correct exampl

Incorrect example

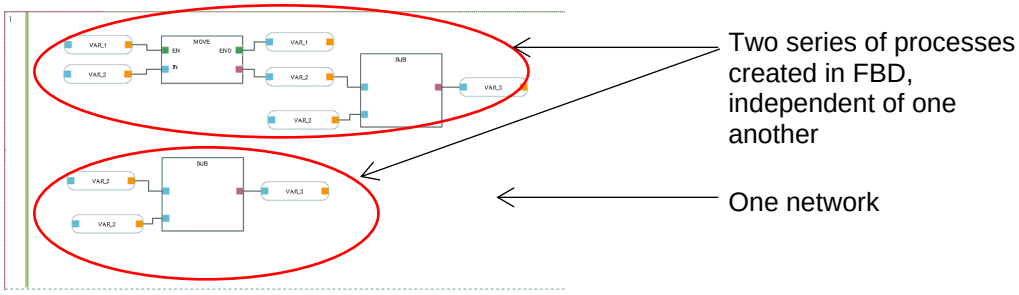
12.1.6 FBD Notation in Networks

The user can only create one independent series of processes per network in the FBD language. This means that all the "boxes are connected in one or more blocks without any break in the connections", showing a series of processes in the FBD language.

Correct example

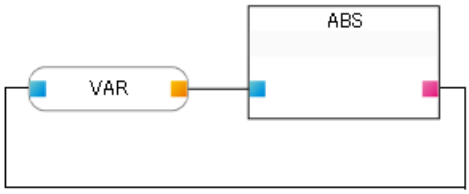


Incorrect example

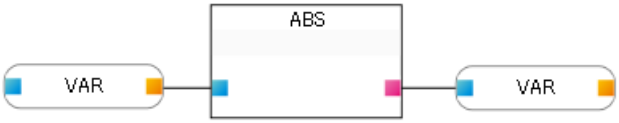


12.1.7 Feedback Paths

Do not create programs with loops like the one shown below. In this situation, conversion is not possible.



When that type of processing is needed in the program, create a program like the following.



12.1.8 Reserved Words and Reserved Characters

Reserved words and reserved characters are characters and combinations of characters that follow the rules of syntax. In addition, letters are not case sensitive.
In the FBD language, the following words and characters are reserved.

Reserved characters	:=, (,), *, **, /, +, -, <, >, <=, >=, =, <>, ｜, &, ;, ¥, **[,]**, ., ,, :, #, e, E, $, ', d, D, h, H, m, M, s, S, %, "	
Reserved words	Same character strings as control statements	IF, THEN, END_IF, ELSE, ELSIF, CASE, OF, END_CASE, FOR, TO, BY, DO, END_FOR, WHILE, END_WHILE, REPEAT, END_REPEAT, UNTIL, RETURN, EXIT, TRUE, FALSE
	Same character strings as function names and method names	Refer to “12.3.1 List of ST language functions” in the PLC System Manual. Ex SHR, ROR User defined function and system defined function names
	Same character strings as OM names	X, Y, BX, BY, BDX, BDY, M, LM, SM, LT, LTS, LTC, LTP, LTV, LC, LCS, LCC, LCP, LCV, D, LD, IX, SD, N, L, GB, GW, SAIX, AIX, B, W and R (including character strings used in the PLC Programming software D00) Character strings with the same format as an OM , such as combinations with the aforementioned character strings and numbers, as well as combinations with characters used for indexing or characters for specifying multiple bits (W and L) (*1)(*2) (Ex: X0, D100, M0IX0, M16W, D100L, etc.)

*1 When the beginning character is an X or Y, the character strings that include numbers and letters (between A and F) are also reserved words.
(Ex: XA and Y7FF)

*2 Number combinations outside of the OM number range are also reserved words.
(Ex: XFFFF, D99999, M1W, etc.)

Reserved words cannot be used for variable names. If an attempt is made to set a reserved word for a variable name in the PLC Programming software D00 FBD Programming Editor, then an error will appear regardless of the case of the letter. For example, neither “IF” nor “if” can be used for a variable name.

12.2 FBD Language Standard Functions

This section provides a list of FBD language functions and detailed descriptions.

12.2.1 List of FBD Language Standard Functions

The following table shows a list of FBD language functions that come standard in the PLC Programming software D00.

Category	Command	Process	Step number (*2)
Convert data type	input_TO_output	Convert data type	4
	input_TO_BCD	Convert numerical value → BCD	4
	BCD_TO_output	Convert BCD → numerical value	4
Copy binary	input_TO_output	Copy bit stream	4
Arithmetic operation	ADD	Addition	4*(N-1)
	SUB	Subtraction	4
	MUL	Multiplication	4*(N-1)
	DIV	Division	4
	MOD	Remainder	12
	EXPT	Exponentiation	4
	MOVE	Transfer	3
Numerical operation	ABS	Absolute value	4
	SQRT	Square root	4
	LN	Natural logarithm	4
	LOG	Common logarithm	4
	EXP	Exponential function	4
	SIN	Sine value (sin)	4
	COS	Cosine value (cos)	4
	TAN	Tangent value (tan)	4
	ASIN	Arcsine value ($\sin^{-1}$)	4
	ACOS	Arccosine value ($\cos^{-1}$)	4
	ATAN	Arctangent ($\tan^{-1}$)	4
Bit shift	SHR	Bit shift to the right	4
	SHL	Bit shift to the left	4
	ROR	Bit rotation to the right	4
	ROL	Bit rotation to the left	4
Bit operation	AND	Logical conjunction	4*(N-1)
	OR	Logical disjunction	4*(N-1)
	XOR	Exclusive disjunction	4*(N-1)
	NOT	Negation	3
Selection	SEL	Select one from IN0 and IN1	19
	MAX	Select maximum value	4+(N-1)*10
	MIN	Select minimum value	4+(N-1)*10
	LIMIT	Limit using upper/lower limit value	5
	MUX	Multiplexer	21+(N-1)*7
Comparison	GT	Size comparison (>)	11+(N-1)*6
	GE	Size comparison (>=)	11+(N-1)*6
	LT	Size comparison (<)	11+(N-1)*6
	LE	Size comparison (<=)	11+(N-1)*6
	EQ	Equal (=)	11+(N-1)*3
	NE	Not equal (<>)	4

Category	Command	Process	Step number (*2)
Character string	LEN	Character string length detection	4
	LEFT	Extract character string data from the left	5
	RIGHT	Extract character string data from the right	5
	MID	Extract character string data from specified position	6
	CONCAT	Concatenate character strings	5
	INSERT	Insert character string	6
	DELETE	Delete character string	6
	REPLACE	Replace character string	7
	FIND	Find character string	4
Time data type	ADD_TIME	Add time	4
	ADD_DT_TIME	Add date and time	4
	ADD_TOD_TIME	Add time of day	4
	SUB_TIME	Subtract time	4
	SUB_DATE_DATE	Subtract date	4
	SUB_TOD_TIME	Subtract time from time of day	4
	SUB_TOD_TOD	Subtract time of day from time of day	4
	SUB_DT_TIME	Subtract time from date and time	4
	SUB_DT_DT	Subtract date and time from date and time	4
	MUL_TIME	Multiply time	4
	DIV_TIME	Divide time	4
	CONCAT_DATE_TOD	Concatenate date and time of day	5
Extension (such as OM operations)	MCPY	Copy data from OM/variable to OM	4
	TON	Timer OM count	14
User definition	Depends on the function data created by the user. (*1)		7 or greater Varies depending on number of arguments.
System definition	Depends on the function data created by the user. (*1)		7 or greater Varies depending on number of arguments.

*1 Refer to “12.2.3 Create & Use user defined function / system defined function” for details about user and system defined functions.

*2 The following is not considered for the step number that is indicated.
- When using an array variable for a function argument
- When using a character string constant for a function argument
- When using EN/ENO

12.2.2 Detailed Descriptions of FBD Language Standard Functions

The following provides detailed descriptions about the FBD language standard functions.
In those detailed descriptions, WL designation and INDEX refer to the following.

WL designation:
Indicates whether multiple bit access for the bit OM is possible.
Ex M0:4 : Accesses M0 to M3 simultaneously.
Indicates whether 32-bit (long) access for the word OM is possible.
Ex D0L: Accesses D0 and D1 simultaneously with 32-bit data.

INDEX:
Indicates whether specifying indirectly for the OM is possible.
Ex M0IX0: Specifies OM with the IX0 value for the OM number.

However, an array variable can be used only with “MOVE”.
When specifying an element for an array variable, if the data types match, they can be used with each function.

(NOTE 1) Use constants within the range of the data types that can be used with the function parameters.

(NOTE 2) Set the IX value so that the OM number does not exceed the maximum value based on index addressing.

(NOTE 3) When specifying an OM that is out of range, proceed with caution because a different type of OM may be read and written.
If another OM value is read and written, it is read and written following the order below.

Global OM range: X → Y → M → D → BX → BY → BDX → BDY → Variable

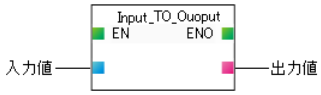
Local OM range: LM → LD → LT → LTS → LTC → LTV (current value) → LTP (set value) → LC → LCS → LCC → LCV (current value) → LCP (set value) → SM → SD → IX

For example, when IX0 = 1, the Y7FFIX0 becomes out of range (by 1 point), and designates M0 as the OM’s first point after Y when following the sequence above. In this way, the program created may malfunction because it accesses a different OM value.

12.2.2.1 Data type conversion function

input_TO_output

- Function

Function name	Notation	Description
input_TO_output		Converts the data type. The data type before conversion is used at input, and the data type after conversion is used at output.

- Conversion table for input_TO_output
  The following is a conversion table for input_TO_output. The symbol ○ indicates that conversion is possible, and the symbol – indicates that conversion is not possible.

Data type before conversion	Data type after conversion																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
BOOL	-	-	-	-	○	○	○	○	○	○	-	-	-	-	-	-	-
BYTE	-	-	-	-	○	○	○	○	○	○	-	-	-	-	-	-	○
WORD	-	-	-	-	○	○	○	○	○	○	-	-	-	-	-	-	-
DWORD	-	-	-	-	○	○	○	○	○	○	-	-	-	-	-	-	-
INT	○	○	○	○	-	○	○	○	○	○	○	○	-	-	-	-	-
DINT	○	○	○	○	○	-	○	○	○	○	○	○	-	-	-	-	-
LINT	○	○	○	○	○	○	-	○	○	○	○	○	-	-	-	-	-
UINT	○	○	○	○	○	○	○	-	○	○	○	○	-	-	-	-	○
UDINT	○	○	○	○	○	○	○	○	-	○	○	○	-	-	-	-	○
ULINT	○	○	○	○	○	○	○	○	○	-	○	○	-	-	-	-	○
REAL	-	-	-	-	○	○	○	○	○	○	-	○	-	-	-	-	-
LREAL	-	-	-	-	○	○	○	○	○	○	○	-	-	-	-	-	-
TIME	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
DATE	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
TOD	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
DT	-	-	-	-	-	-	-	-	-	-	-	-	-	○	○	-	-
STRING	-	○	-	-	-	-	-	○	○	○	-	-	-	-	-	-	-

- When converting from a non-BOOL type to a BOOL type, if the value before conversion is 0, then the value is converted to 0. If the value is not 0, then the value is converted to 1.
- When converting from a BOOL type to a non-BOOL type, if the value before conversion is 0, then the least significant bit is assigned 0. If the value before conversion is 1, then the least significant bit is assigned 1. Other bits are assigned 0.
- When converting from a STRING type to a non-STRING type, the character string (0 to 9) is converted to a numerical value.
  (Ex: Character string “1” is converted to “1”)
  When the beginning of the character string is a plus (+) or minus (-), then it is processed as a plus or minus sign.
  If an invalid character is included in the conversion source, the numerical value 0 is stored in the conversion destination.
  (Ex: Character string “2A3” is converted to the numerical value “0”)
  When a numerical value after conversion exceeds the upper limit for the data type at the stored location, then the value wraps to the lower limit. In the same way, when a numerical value after conversion exceeds the lower limit for the data type at the stored location, then the value wraps to the upper limit. For example, in STRING_TO_BYTE, the character string “256” is converted to the numerical value “0”, and the character string “-1” is converted to the numerical value “255”.
- When converting from a non-STRING type to a STRING type, the numerical value (decimal numeral) is converted to a character string.
  (Ex: Numerical value “4” is converted to character string “4”)
- When converting from a DT type to a DATE type or to a TOD type, the date and time at the conversion source are stored at the conversion destination.
- When converting from a REAL/LREAL type to another type, it is converted to a value with the decimal point rounded off.

(NOTE) An error is triggered at conversion (POU conversion) if the combination of data types before and after conversion is invalid.

- Description of parameter

input	Input data type
Output	Data type after output
Input value	Input data
Output value	Data after converting type

- Usable data type
  "○" indicates that the data type can be used. "-" indicates that the data type cannot be used.

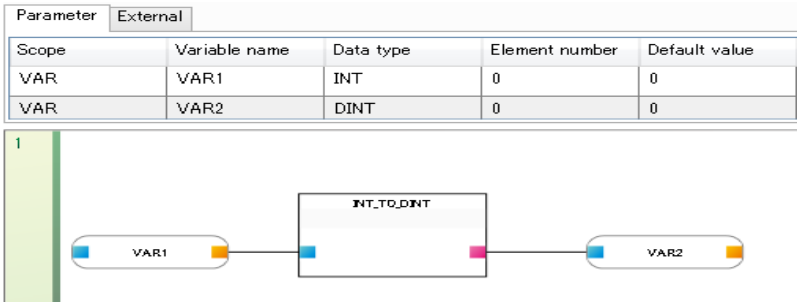
	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	○	○
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	○	○	-	○

- Usable OM / Constant

	Bit									Word								Bit	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○*1	○
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

*1 8-, 16- and 32-bit can only be specified.

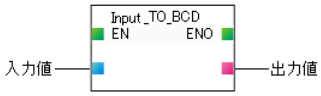
- Usage example
  INT type variable (VAR1) is changed to DINT type and is assigned to output value (VAR2).



- NC(BUILT IN PLC) execution error
  When the value, converted from a real number to an integer, exceeds the range of the data type where the data is assigned. (Error code 15)

input_TO_BCD

- Function

Function name	Notation	Description
input_TO_BCD		Converts an integer (input value) to a BCD value.

(NOTE 1) Store values that can be converted to BCD for the input value.
Ex 0 to 9999 for an INT type

(NOTE 2) Refer to "12.2.2.1 Assignment" in the PLC System Manual for details about how to store the output value for the operation result.

- Description of parameter

input	Input data type
Input value	Value before conversion
Output value	Value after converted to BCD

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

- Usable OM / Constant

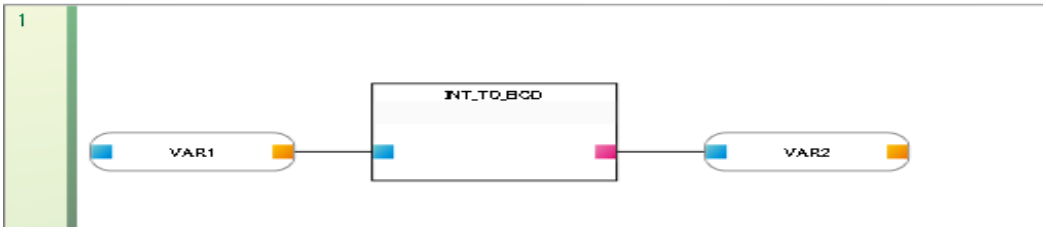
	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○*1	○
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

* 1 8-, 16- and 32-bit can only be specified.

- Usage example
  INT type is converted to a BCD type and is assigned to a WORD type variable. When 1985 is assigned to the input value, 16#1985 is assigned to the output value.

Parameter | External

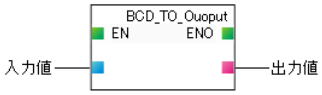
Scope	Variable name	Data type	Element number	Default value
VAR	VAR1	INT	0	1985
VAR	VAR2	WORD	0	16#0000



- NC(BUILT IN PLC) execution error
  When a numerical value that cannot be converted to BCD is specified at the conversion source. (Error code 10)

BCD_TO_Output

- Function

Function name	Notation	Description
BCD_TO_Output		Converts a BCD value (input value) to another integer value.

(NOTE 1) Store the BCD value in the input value.
(NOTE 2) Refer to "12.2.2.1 Assignment" in the PLC System Manual for details about how to store the output value for the operation result.

- Description of parameter

input	Data type after conversion
Input value	Input data
Output value	Data after converting to BCD

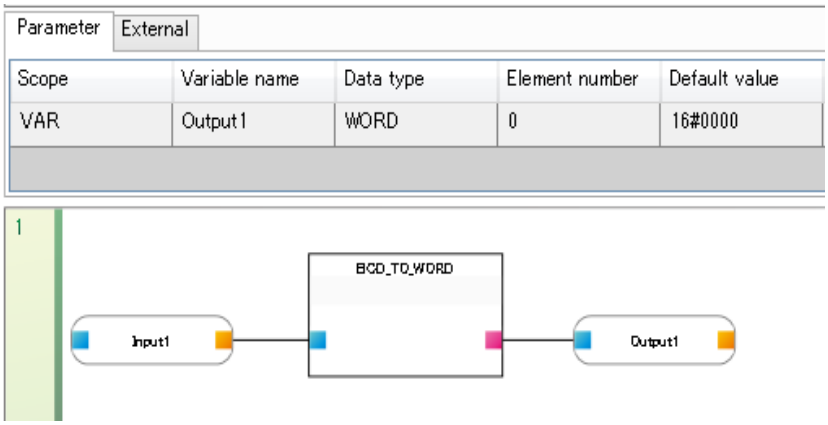
- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

- Usage example
  The BCD value, stored in the INT type variable Input1, is first converted to an integer value and then assigned to the WORD type variable Output1.
  For example, when 16#1234 is assigned to the input value, 1234 is then assigned to the output value.

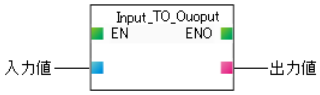


- NC(BUILT IN PLC) execution error
  When a value that is not a BCD value is specified for the conversion source. (Error code 10)

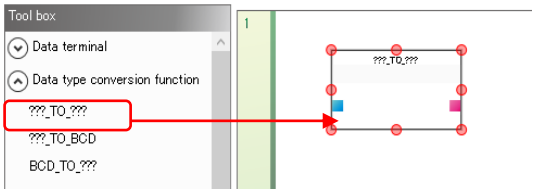
12.2.2.2 Binary copy function

input_TO_output

- Function

Function name	Notation	Description
input_TO_output		Copies the data from the copy source to the copy destination. The data type for the copy source is entered for input, and the data type for the copy destination is entered for output. The data at the copy source is “In”, and the data at the copy destination is “Out”.

When copying binary, use the tool box in the FBD Programming Editor to insert “???_TO_???” under the “Data type conversion function” into the program editing area.



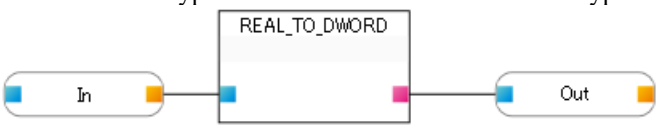
- Conversion table for input_TO_output
  The following is a conversion table for input_TO_output. The symbol ○ indicates that conversion is possible, and the symbol – indicates that conversion is not possible.

Data type before conversion	Data type after conversion																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
BOOL	-	○	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-
BYTE	○	-	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-
WORD	○	○	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-
DWORD	○	○	○	-	-	-	-	-	-	-	○	-	-	-	-	-	-
INT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
DINT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
LINT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
UINT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
UDINT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ULINT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
REAL	-	-	-	○	-	-	-	-	-	-	-	-	-	-	-	-	-
LREAL	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
TIME	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
DATE	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
TOD	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
DT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
STRING	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

(NOTE) An error is triggered at conversion (POU conversion) if the combination of data types before and after conversion is invalid.

- Operation
  The data from the copy source is copied to the copy destination.

  Ex "In" is a REAL type variable and "Out" is a DWORD type variable.



  If the "In" value is 3.5 (2#0100 0000 0110 0000 0000 0000 0000 0000), then the "Out" value is 1080033280 (2#0100 0000 0110 0000 0000 0000 0000 0000).

- Description of parameter

input	Data type at copy source
output	Data type at copy destination
Input value	Data at copy source
Output value	Data at copy destination

- Usable data type
  "○" indicates that the data type can be used. "-" indicates that the data type cannot be used.

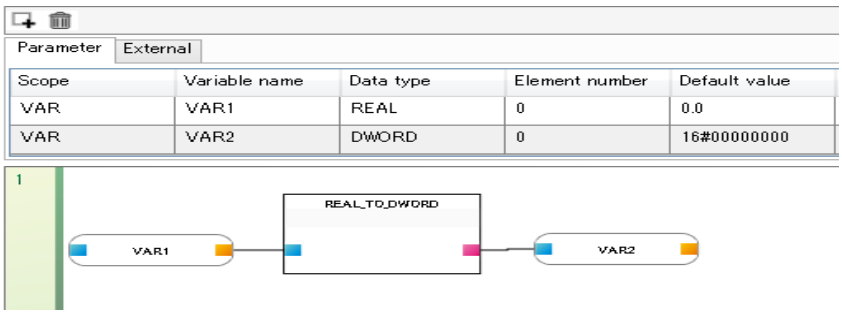
	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	○	○	○	-	-	-	-	-	-	-	○	-	-	-	-	-	-
Output value	○	○	○	○	-	-	-	-	-	-	○	-	-	-	-	-	-

- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○*1	○
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

* 1 8-, 16- and 32-bit can only be specified.

- Usage example
  The bit stream for the REAL type variable (VAR1) is copied to the DWORD type variable (VAR2).

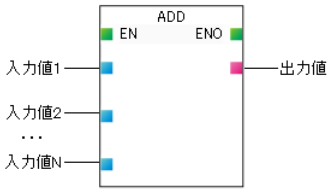


- NC(BUILT IN PLC) execution error
  None

12.2.2.3 Arithmetic operation function

ADD

- Function

Function name	Notation	Description
ADD	 (N:2~15)	Adds the input values.

(NOTE 1) An error is not triggered even if an overflow occurs from the addition operation. In this case, the value is expressed as one that can be used with the "Return value" data type.
For example, if the input value 1 is 32767 (INT), the input value 2 is 1 (INT) and the output value is an INT type, then the result is 32768 (16#8000) and "-32768" expressed as an INT type is output.

(NOTE 2) Refer to "12.2.2.1 Assignment" in the PLC System Manual for details when the data types for the operation result and the "Return value" data are different.

- Description of parameter

Input value1~N	Add value
Output value	Value after adding

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

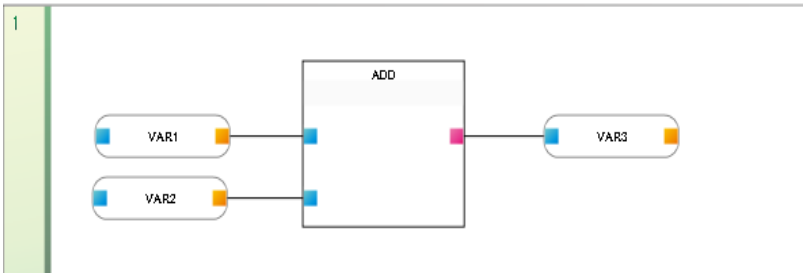
- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	-	○
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	-	○

- Usage example
  Input values VAR1 and VAR2 are added and then assigned to the output value VAR3.
  Ex When the input values are 1.2 and 0.2, then 1.4 is assigned to the output value.

Parameter | External

Scope	Variable name	Data type	Element number	Default value	Comment
VAR	VAR1	REAL	0	1.2	
VAR	VAR2	REAL	0	0.2	
VAR	VAR3	REAL	0	0.0	



- NC(BUILT IN PLC) execution error
  None

SUB

- Function

Function name	Notation	Description
SUB		Performs input subtraction. A value (input value 2) is subtracted from another value (input value 1), and that result is output to the output value.

(NOTE 1) An error is not triggered even if an overflow occurs from the subtraction operation. In this case, the value is expressed as one that can be used for a bit number in the output value data type.
For example, if the input value 1 is -32768 (INT), the input value 2 is 1 (INT) and the output value is an INT type, then after subtracting the values, 32767 is output.

(NOTE 2) Refer to "12.2.2.1 Assignment" in the PLC System Manual for details when the data types for the operation result and the "Return value" data are different.

- Description of parameter

Input value1	Subtraction value (minuend)
Input value2	Subtraction value (subtrahend)
Output value	Subtraction result (difference)

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

- Usable OM / Constant

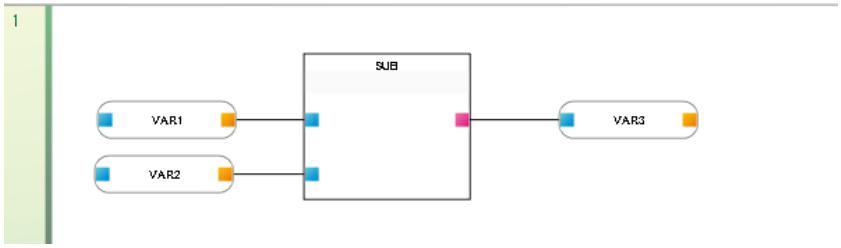
	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

12

- Usage example
  VAR2 is subtracted from VAR1, and the operation result is assigned to VAR3.
  Ex When 1 is subtracted from 3 (minuend), then 2 is assigned to the output value.

Parameter | External

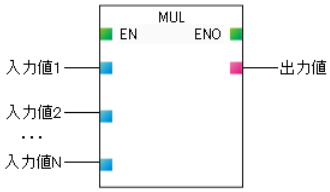
Scope	Variable name	Data type	Element number	Default value	Comment
VAR	VAR1	INT	0	3	
VAR	VAR2	INT	0	1	
VAR	VAR3	INT	0	0	



- NC(BUILT IN PLC) execution error
  None

MUL

- Function

Function name	Notation	Description
MUL	 N:2~15	Multiplies the input values and outputs the operation result.

(NOTE 1) An error is not triggered even if an overflow occurs from the multiplication operation. In this case, the value is expressed as one that can be used for a bit number in the output value data type.
For example, when the operation result for 32767 (INT type) × 2 (INT type) is output to an INT type, the operation is as follows:
32767 × 2 = 65534 (16#FFFE)
16#FFFE is expressed as an INT type and -2 becomes the operation result.

(NOTE 2) Refer to "12.2.2.1 Assignment" in the PLC System Manual for details when the data types for the operation result and the "Return value" data are different.

- Description of parameter

Input value1~N	Multiplication value
Output value	Data after multiplication

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

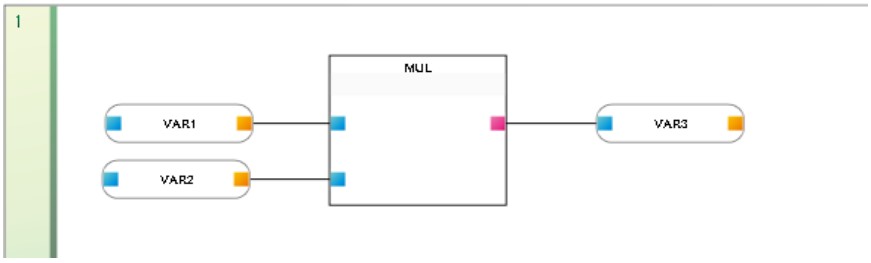
- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

- Usage example
  VAR1 and VAR2 are multiplied, and the operation result is assigned to the variable VAR3.
  Ex When the input values are 3 and 2, then 6 is assigned to the output value.

Parameter | External

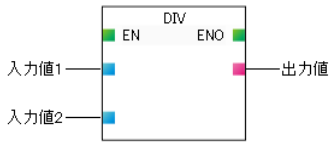
Scope	Variable name	Data type	Element number	Default value	Comment
VAR	VAR1	INT	0	3	
VAR	VAR2	INT	0	2	
VAR	VAR3	INT	0	0	



- NC(BUILT IN PLC) execution error
  None

DIV

- Function

Function name	Notation	Description
DIV		Performs input division. The value (input value 1) is divided by another value (input value 2), and that result is output to the output value.

(NOTE 1) It is OK even if different data types are used for input value 1 and input value 2. However, in this situation, the data types are unified to the type that has the greater data size to perform the calculation.

(NOTE 2) The “Remainder” is not included in the operation result. If there is a remainder, then it is rounded off.

(NOTE 3) An error is not triggered even if an operation result produces an overflow. In this case, the value is expressed as one that can be used for a bit number in the “return value” data type.

(NOTE 4) Refer to “12.2.2.1 Assignment” in the PLC System Manual for details when the data types for the operation result and the “Return value” data are different.

- Description of parameter

Input value1	Division value (dividend)
Input value2	Division value (divisor)
Output value	Division result (quotient)

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-
Input value2	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

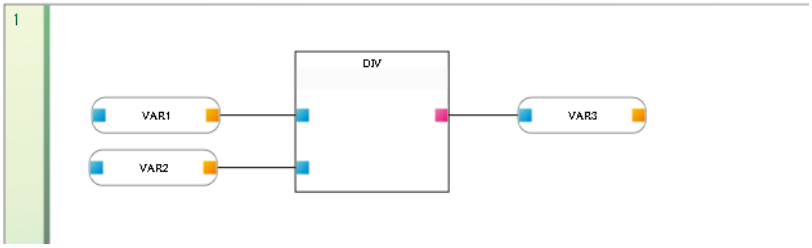
- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Input value2	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

- Usage example
  VAR1 is divided by VAR2, and the operation result is assigned to the variable VAR3.
  For example, when the dividend is 16 and the divisor is 4, then 4 is assigned to the output value.

Parameter | External

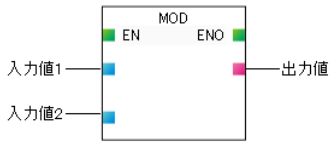
Scope	Variable name	Data type	Element number	Default value	Comment
VAR	VAR1	INT	0	16	
VAR	VAR2	INT	0	4	
VAR	VAR3	INT	0	0	



- NC(BUILT IN PLC) execution error
  When the divisor is 0 in the division command. (Error code 11)

MOD

- Function

Function name	Notation	Description
MOD		Calculates the remainder. A value (input value 1) is divided by another value (input value 2), and that result (remainder) is output to the output value.

(NOTE) Refer to "12.2.2.1 Assignment" in the PLC System Manual for details when the data types for the operation result and the "Return value" data are different.

- Description of parameter

Input value1	Division value (dividend)
Input value2	Division value (divisor)
Output value	Remainder

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Input value2	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

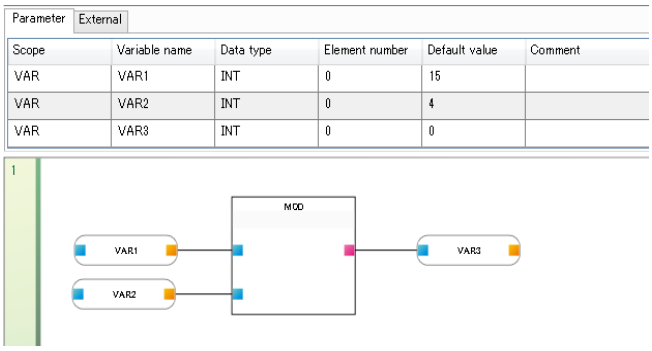
- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Input value2	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

- Usage example

The remainder from dividing variable VAR1 by VAR2 is assigned to variable VAR3.

Ex When the dividend (VAR1) is 15 and the divisor (VAR2) is 4, then 3 is assigned to the output value.

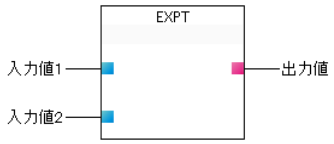


- NC(BUILT IN PLC) execution error

When the divisor is 0 in the division command. (Error code 11)

EXPT

- Function

Function name	Notation	Description
EXPT		Performs exponentiation with a base value (input value 1) and an exponent (input value 2), and the result is output to the output value.

(NOTE 1) An error is not triggered even if an overflow occurs from the operation.
(NOTE 2) Refer to "12.2.2.1 Assignment" in the PLC System Manual for details when the data types for the operation result and the "Return value" data are different.

- Description of parameter

Input value1	Base
Input value2	Exponent
Output value	Operation result of exponentiation

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-
Input value2	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

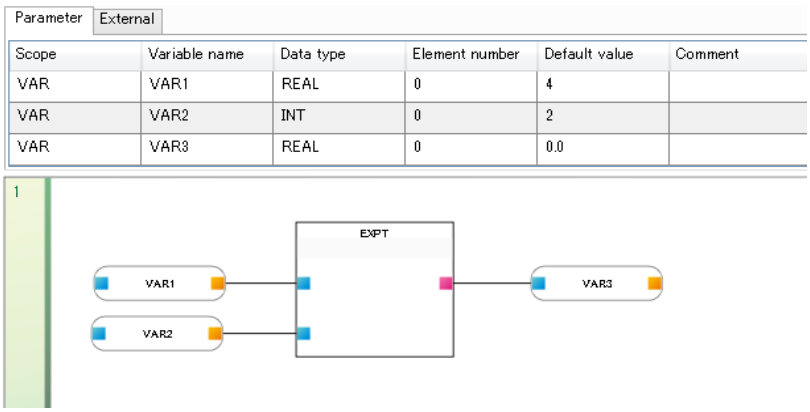
- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
Input value2	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

*1 Real numerals can only be specified.

- Usage example

Variable VAR1 to the power of VAR2 is calculated and the result is assigned to VAR3.
Ex When 4 is assigned as the base and 2 is assigned as the exponent, then 16 is assigned to the output value.

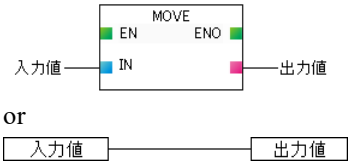


- NC(BUILT IN PLC) execution error

None

MOVE

- Function

Function name	Notation	Description
MOVE		Assigns input value to output value.

(NOTE) Never assign for input X.

- Description of parameter

Input value	Value assigned
Output value	Output

- Usable data type
  The following data types for variables can be used for the input value and output value, but the data type needs to be able to store data for the input value. Refer to “Data type combinations that can be assigned” (described later) for details on data type combinations that can be used for input and output values.

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Output value	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	○	○	○
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

- Data type combinations that can be assigned
  The following describes the data type combinations that can be assigned.

  The following motions are carried out depending on the data type size.
  - When the bit number is the same on the left and right sides, the value is assigned as is.
  - If the bit number on the left side is larger than the bit number on the right side, when right side is a signed value and the most significant bit is a 1, then the remainder bit (where it is assigned) is filled with a 1. For example, when assigning from 16-bit to 32-bit, if the input value is 16#8000, then 16#FFFF8000 is carried out as the assignment.
  - If the bit number on the left side is larger than the bit number on the right side, when the right side is a signed value and the most significant bit is a 0, or when the right side is an unsigned value, then the remainder bit (where it is assigned) is filled with a 0. For example, when assigning from 16-bit to 32-bit, if the input value is 16#7FFF, then 16#00007FFF is carried out as the assignment.
  - When the bit number on the left side is smaller than the bit number on the right side, then the number of digits (for the number of bits on the left side) is assigned from the low-order on the right side. For example, when assigning from 32-bit to 16-bit, if the input value is 16#FFFF1111, then 16#1111 for the low-order 16-bit is assigned.

Output value data type	Input value data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	DATE	TOD	DT	TIME	STRING
BOOL	○	○*1	○*1	○*1	○*1	○*1	○*1	○*1	○*1	○*1	○*1	○*1	-	-	-	-	-
BYTE	○*3	○	○	○	○	○	○	○	○	○	○*2	○*2	-	-	-	-	-
WORD	○*3	○	○	○	○	○	○	○	○	○	○*2	○*2	-	-	-	-	-
DWORD	○*3	○	○	○	○	○	○	○	○	○	○*2	○*2	-	-	-	-	-
INT	○*3	○	○	○	○	○	○	○	○	○	○*2	○*2	-	-	-	-	-
DINT	○*3	○	○	○	○	○	○	○	○	○	○*2	○*2	-	-	-	-	-
LINT	○*3	○	○	○	○	○	○	○	○	○	○*2	○*2	-	-	-	-	-
UINT	○*3	○	○	○	○	○	○	○	○	○	○*2	○*2	-	-	-	-	-
UDINT	○*3	○	○	○	○	○	○	○	○	○	○*2	○*2	-	-	-	-	-
ULINT	○*3	○	○	○	○	○	○	○	○	○	○*2	○*2	-	-	-	-	-
REAL	○*3	○*2	○*2	○*2	○*2	○*2	○*2	○*2	○*2	○*2	○	○	-	-	-	-	-
LREAL	○*3	○*2	○*2	○*2	○*2	○*2	○*2	○*2	○*2	○*2	○	○	-	-	-	-	-
DATE	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
TOD	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
DT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
TIME	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-
STRING	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○

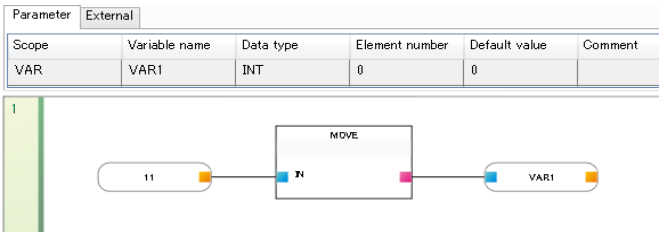
*1 When the input value is a 0, then 0 is assigned. 1 is assigned if the input value is another value besides 0.

*2 When a floating point and integer are converted, that result is assigned.
When a real number is assigned to an integer, the value after the decimal point is rounded off.
However, an error is triggered if the value (converted from a real number to an integer) exceeds the range of the data type (where the value is assigned). When infinity and non-numbers are assigned to an integer, 0 is assigned.

*3 If the input value is 1 (true), the least significant 1 bit (where it is assigned) turns ON and the other bits turn OFF.
If the input value is 0 (false), the least significant 1 bit (where it is assigned) turns OFF and the other bits turn ON.

(NOTE) An error is triggered at conversion if the combination of data types for the input and output values is invalid.

- Usage example
  Constant 11 (INT) is assigned to VAR1.

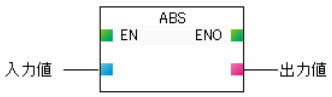


- NC(BUILT IN PLC) execution error
  When real number data is assigned to an integer type and a value outside of the range (for the data type where it is being assigned) is assigned. (Error code 15)

12.2.2.4 Numerical value operation function

ABS

- Function

Function name	Notation	Description
ABS		Outputs the absolute value of the input value.

(NOTE 1) Refer to "12.2.2.1 Assignment" in the PLC System Manual for details when the data types for the operation result and the "Return value" data are different.

(NOTE 2) When the minimum values -32768/-2147483648/-9223372036854775808 for the data types INT/DINT/LINT are entered as input values, then the values -32768/-2147483648/-9223372036854775808 are output.

- Description of parameter

Input value	Input value before converting to an absolute value
Output value	Absolute value

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	-	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

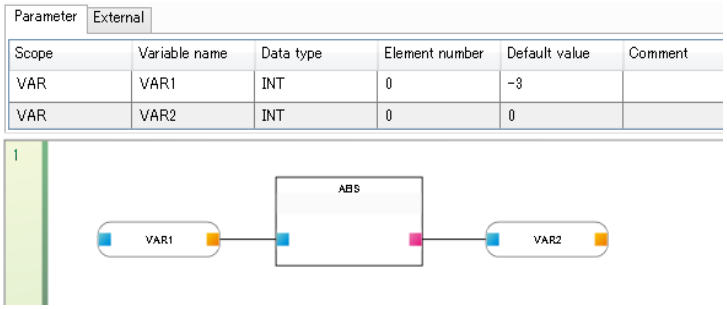
- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value	○*1	○*1	○*1	○*1	○*1	○*1	○*1	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

*1 Possible only when digits are specified.

- Usage example

The absolute value of variable VAR1 is assigned to VAR2.
For example, when the input value is -3, then 3 is assigned to the output value.

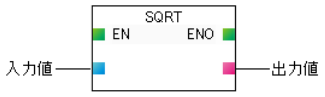


- NC(BUILT IN PLC) execution error

None

SQRT

- Function

Function name	Notation	Description
SQRT		Outputs the square root of the input value.

(NOTE 1) When the input value is negative, -1.#QNAN is output to the output value.
(NOTE 2) Refer to "12.2.2.1 Assignment" in the PLC System Manual for details when the data types for the operation result and the "Return value" data are different.

- Description of parameter

Input value	Value used for operation
Output value	Square root of input value

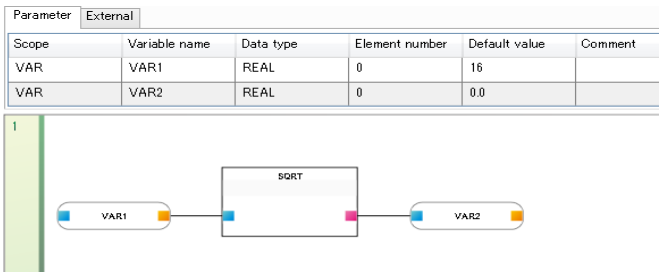
- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

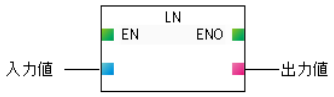
- Usage example
  The square root of variable VAR1 is assigned to variable VAR2.
  Ex When the input value is 16, then 4 is assigned to the output value.



- NC(BUILT IN PLC) execution error
  None

LN

- Function

Function name	Notation	Description
LN		Outputs the natural logarithm of the input value.

(NOTE 1) When the input value is a negative number, 0, infinity, or a non-number, the output value is as follows:
- Negative number: Non-number (-1.#QNAN)
- 0: Non-number (-1.#QNAN)
- Infinity (+∞): Infinity (+∞) (1.#INF)
- Infinity (-∞): Non-number (-1.#QNAN)
- Non-number: Non-number (±1.#QNAN)

(NOTE 2) Refer to "12.2.2.1 Assignment" in the PLC System Manual for details when the data types for the operation result and the "Return value" data are different.

- Description of parameter

Input value	Value used for operation
Output value	Natural logarithm

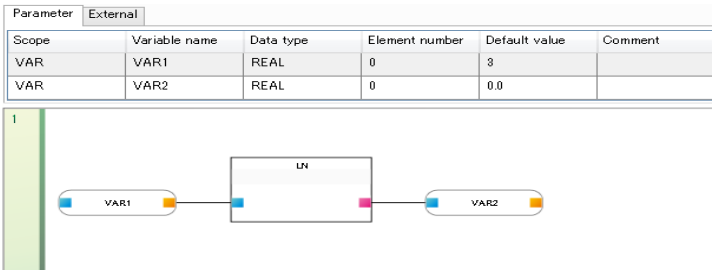
- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

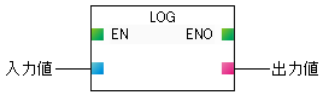
- Usage example
  The natural logarithm of variable VAR1 is assigned to variable VAR2.
  Ex When 3 is assigned to the input value, then 1.098612 is assigned to the output value.



- NC(BUILT IN PLC) execution error
  None

LOG

- Function

Function name	Notation	Description
LOG		Calculates the floating-point natural logarithm of the input value.

(NOTE 1) When the input value is a negative number, 0, infinity, or a non-number, the output value is as follows:
- Negative number: Non-number (-1.#QNAN)
- 0: Non-number (-1.#QNAN)
- Infinity (+∞): Infinity (+∞) (1.#INF)
- Infinity (-∞): Non-number (-1.#QNAN)
- Non-number: Non-number (±1.#QNAN)

(NOTE 2) Refer to "12.2.2.1 Assignment" in the PLC System Manual for details when the data types for the operation result and the "Return value" data are different.

- Description of parameter

Input value	Input value (value used for operation)
Output value	Floating-point natural logarithm of the input value

- Usable data type

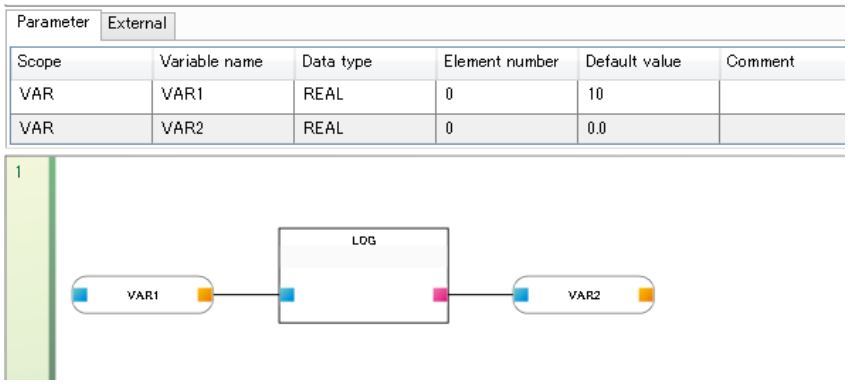
	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

- Usage example

The floating-point natural logarithm of variable VAR1 is assigned to variable VAR2.
For example, when the input value is 10, then 1 is assigned to the output value.

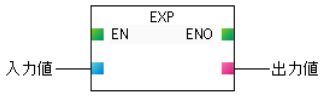


- NC(BUILT IN PLC) execution error

None

EXP

- Function

Function name	Notation	Description
EXP		Calculates the natural exponent ($e^{Input\ value}$) of the input value.

(NOTE 1) When the input value is infinity or a non-number, the output value is as follows:
- Infinity (+∞): Infinity (+∞) (1.#INF)
- Infinity (-∞): 0
- Non-number: Non-number (±1.#QNAN)

(NOTE 2) Refer to "12.2.2.1 Assignment" in the PLC System Manual for details when the data types for the operation result and the "Return value" data are different.

- Description of parameter

Input value	Input value (value used for operation)
Output value	Natural exponent of the input value

- Usable data type

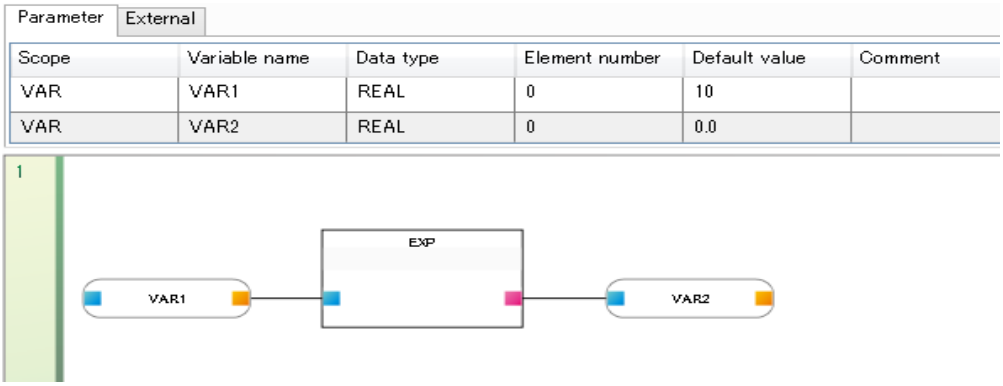
	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

- Usable OM / Constant

	Bit						Word					Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	SM	T	C	D	SD	T	C	IX	L	N	Value	Character string		
Input value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Output value	-	○	○	○	-	-	○	○	-	-	○	-	-	-	-	○	○

- Usage example

The natural exponent of variable VAR1 is assigned to variable VAR2.
When the input value is 3, then e^3 is assigned to the output value.



- NC(BUILT IN PLC) execution error

None

SIN

- Function

Function name	Notation	Description
SIN		Outputs the sine (sin) of the input value.

(NOTE 1) Enter the input value in radians.
(NOTE 2) When the input value is infinity or a non-number, the output value is a non-number.
(NOTE 3) Refer to "12.2.2.1 Assignment" in the PLC System Manual for details when the data types for the operation result and the "Return value" data are different.

- Description of parameter

Input value	Value used for operation
Output value	Sine value (sin)

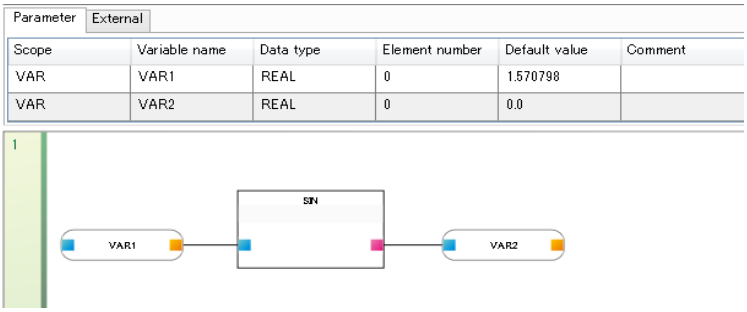
- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

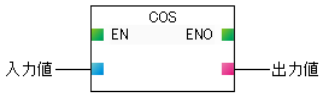
- Usage example
  The sine value of variable VAR1 is assigned to VAR2.
  When the input value is 1.570796(π/2), then 1 is assigned to output value.



- NC(BUILT IN PLC) execution error
  None

COS

- Function

Function name	Notation	Description
COS		Outputs the cosine (cos) of the input value.

(NOTE 1) Enter the input value in radians.
(NOTE 2) When the input value is infinity or a non-number, the output value is a non-number.
(NOTE 3) Refer to "12.2.2.1 Assignment" in the PLC System Manual for details when the data types for the operation result and the "Return value" data are different.

- Description of parameter

Input value	Value used for operation
Output value	Cosine value (cos)

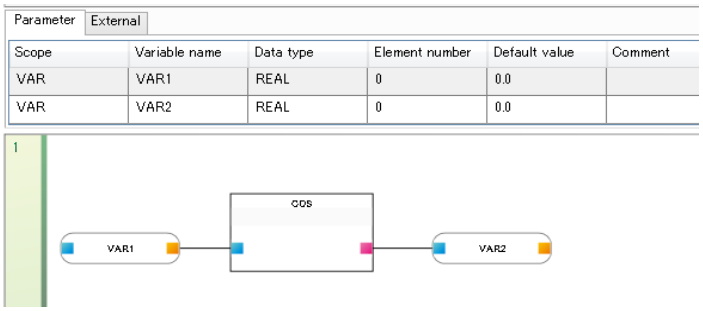
- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

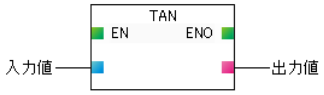
- Usage example
  The cosine (cos) of VAR1 is assigned to VAR2.
  For example, when the input value is 0, then 1 is assigned to the output value.



- NC(BUILT IN PLC) execution error
  None

TAN

- Function

Function name	Notation	Description
TAN		Calculates the tangent (tan) of the input value.

(NOTE 1) Enter the input value in radians.
(NOTE 2) When the input value is 0, the output value is 0.
(NOTE 3) Refer to "12.2.2.1 Assignment" in the PLC System Manual for details when the data types for the operation result and the "Return value" data are different.

- Description of parameter

Input value	Value used for operation
Output value	Tangent value (tan)

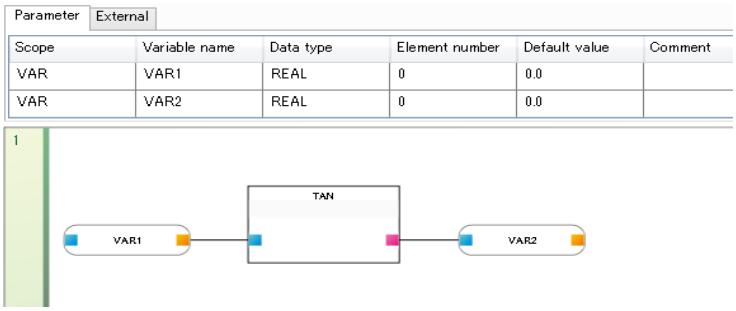
- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

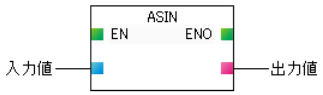
- Usage example
  The tangent (tan) of VAR1 is assigned to VAR2.
  For example, when the input value is 0, then 0 is assigned to the output value.



- NC(BUILT IN PLC) execution error
  None

ASIN

- Function

Function name	Notation	Description
ASIN		Outputs the arcsine ($\sin^{-1}$) of the input value.

(NOTE 1) When the input value range is between -1.0 and 1.0, the output result is a non-number (±1.#QNAN).
(NOTE 2) The output value is in radians.
(NOTE 3) Refer to "12.2.2.1 Assignment" in the PLC System Manual for details when the data types for the operation result and the "Return value" data are different.

- Description of parameter

Input value	Value used for operation
Output value	Arcsine value (arcsin)

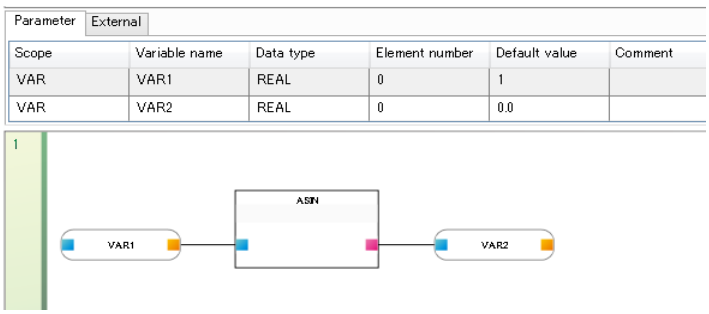
- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

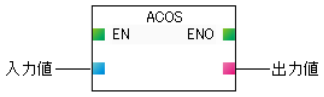
- Usage example
  The arcsine of variable VAR1 is assigned to variable VAR2.
  For example, when the input value is 1, then 1.570796(π/2) is assigned to the output value.



- NC(BUILT IN PLC) execution error
  None

ACOS

- Function

Function name	Notation	Description
ACOS		Calculates the arccosine ($\cos^{-1}$) of the input value.

(NOTE 1) When the input value range is between -1.0 and 1.0, the output result is a non-number (±1.#QNAN).
(NOTE 2) The output value is in radians.
(NOTE 3) Refer to "12.2.2.1 Assignment" in the PLC System Manual for details when the data types for the operation result and the "Return value" data are different.

- Description of parameter

Input value	Value used for operation
Output value	Arccosine value (arccos)

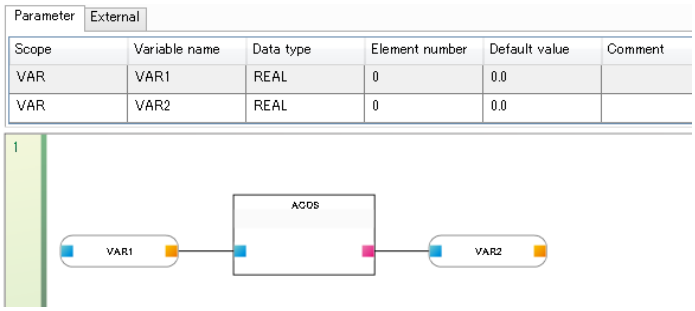
- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

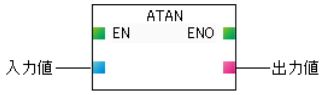
- Usage example
  The arccosine ($\cos^{-1}$) of VAR1 is assigned to VAR2.
  For example, when the input value is 0, then 1.570796(π/2) is assigned to the output value.



- NC(BUILT IN PLC) execution error
  None

ATAN

- Function

Function name	Notation	Description
ATAN		Calculates the arctangent ($\tan^{-1}$) of the input value.

(NOTE 1) When the input value is a non-number, the output value is a non-number.
(NOTE 2) The output value is in radians.
(NOTE 3) Refer to "12.2.2.1 Assignment" in the PLC System Manual for details when the data types for the operation result and the "Return value" data are different.

- Description of parameter

Input value	Value used for operation
Output value	Arctangent ($\tan^{-1}$) of the input value

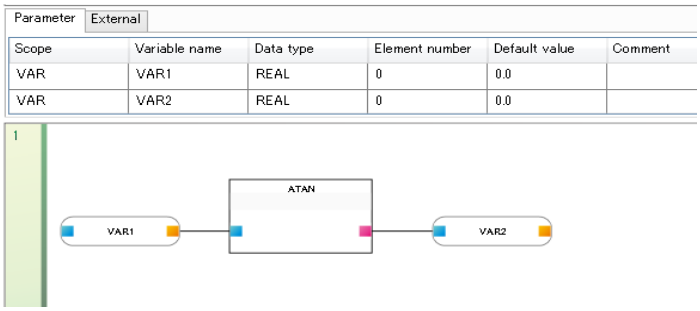
- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

- Usage example
  The arctangent ($\tan^{-1}$) of VAR1 is assigned to VAR2.
  For example, when the input value is 0, then 0 is assigned to the output value.

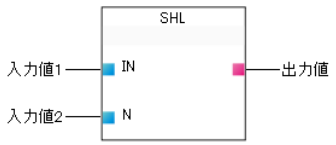


- NC(BUILT IN PLC) execution error
  None

12.2.2.5 Bit shift function

SHL

- Function

Function name	Notation	Description
SHL		Shifts input value 1 to the left only for the specified bit number (input value 2), and that value is output to the output value.

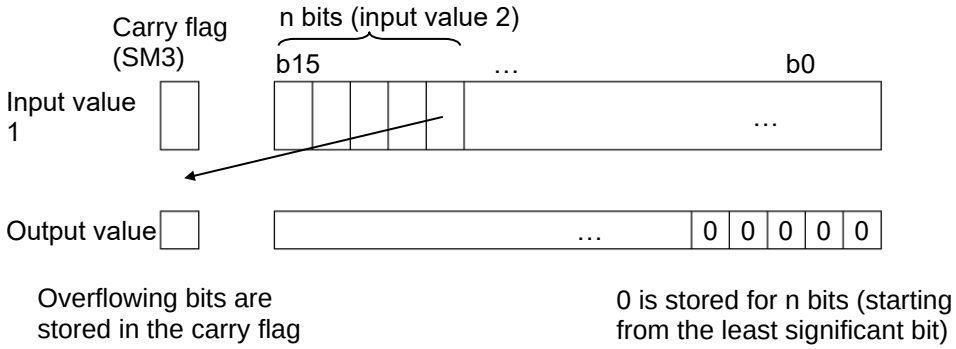
(NOTE 1) Specify a positive integer for the bit number that shifts.
However, when 0 is specified as the bit number that shifts, the input value is output.
When a negative number is specified as the bit number that shifts, 0 is output.

(NOTE 2) In contrast with the SHL command in the LD language, the variable / OM value specified for input value 1 does not change.

(NOTE 3) Similar to the SHL command in the LD language, in order to shift the variable / OM value specified for input value 1, the same variable / OM is specified to the output value.

- Operation

When SHL is executed, D is shifted n bits to the right from the low-order to the high-order, and that value is output.



- Description of parameter

Input value1	Data for shift
Input value2	Bit number that shifts
Output value	Result after the left bit shift

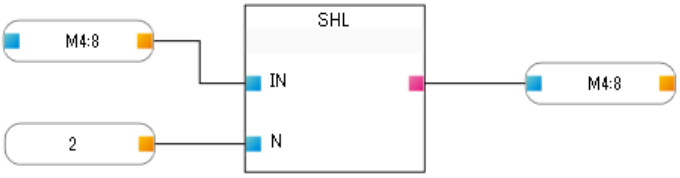
- Usable data type

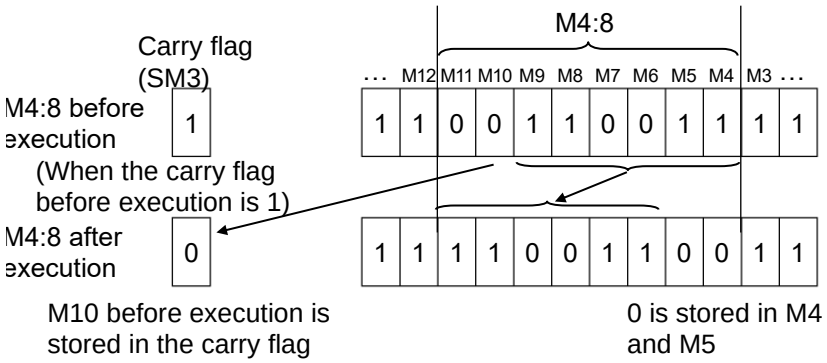
	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Input value2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○
Output value2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

- Usage example
  The value for OM M4:8 is shifted 2 bits to high-order from low-order.



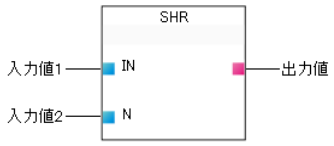


- NC(BUILT IN PLC) execution error
  None

12

SHR

- Function

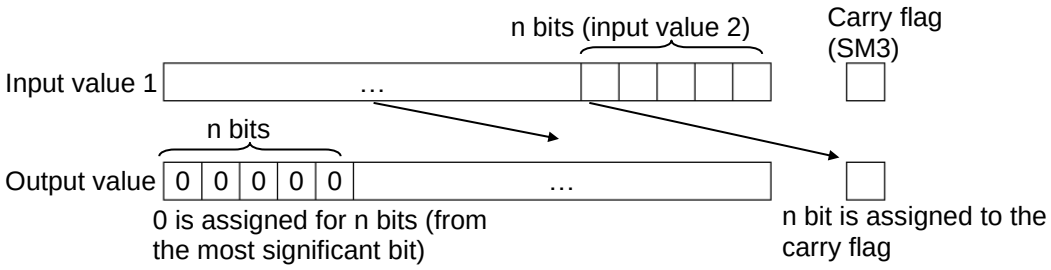
Function name	Notation	Description
SHR		Shifts input value 1 to the right only for the specified bit number (input value 2), and that value is output to the output value.

(NOTE 1) Specify a positive integer for the bit number that shifts.
However, when 0 is specified as the bit number that shifts, the input value is output.
When a negative number is specified as the bit number that shifts, 0 is output.

(NOTE 2) In contrast with the SHR command in the LD language, the variable / OM value specified for input value 1 does not change.

(NOTE 3) Similar to the SHR command in the LD language, in order to shift the variable / OM value specified for input value 1, the same variable / OM is specified to the output value.

- Operation
  When the SHL command is executed, the n bits are shifted to the right (from the high-order to the low-order), and that value is output.



- Description of parameter

Input value1	Data for shift
Input value2	Bit number that shifts
Output value	Result after the left bit shift

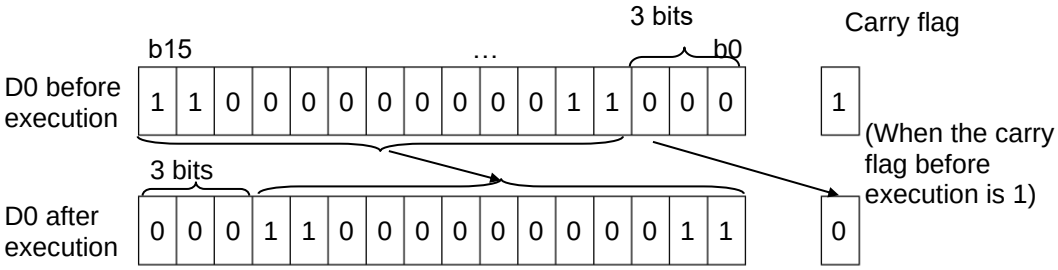
- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Input value2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○
Output value2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

- Usage example
  Value for OM D0 shifts 3 bits to the right.

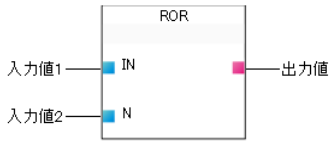


0 is assigned for 3 bits (from the most significant bit)

- NC(BUILT IN PLC) execution error
  None

ROR

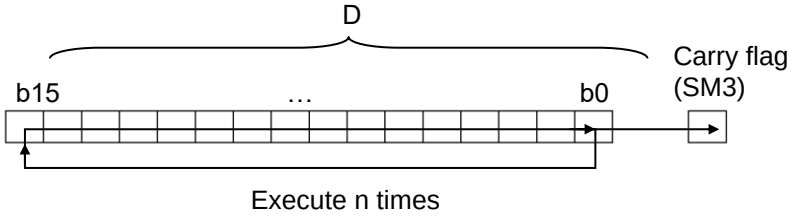
- Function

Function name	Notation	Description
ROR		Rotates the input value 1 data to the right only for the specified bits (input value 2), and that value is output.

(NOTE 1) Specify a positive integer for the bit number that shifts.
(NOTE 2) When the bit number that shifts is 0, then the input value 1 is output.
(NOTE 3) When the bit number is a negative number, the value that rotates to the left is output.
(NOTE 4) In contrast with the RORcommand in the LD language, the variable / OM value specified for input value 1 does not change.
(NOTE 5) Similar to the ROR command in the LD language, in order to shift the variable / OM value specified for input value 1, the same variable / OM is specified to the output value.

- Operation

Each bit for D, not including the carry flag, is rotated to the right when ROR is executed.



- Description of parameter

Input value1	Data for bit rotation
Input value2	Bit number that rotates
Output value	Data after bit rotation to the right

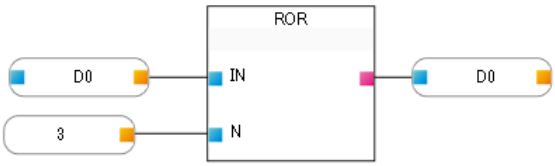
- Usable data type

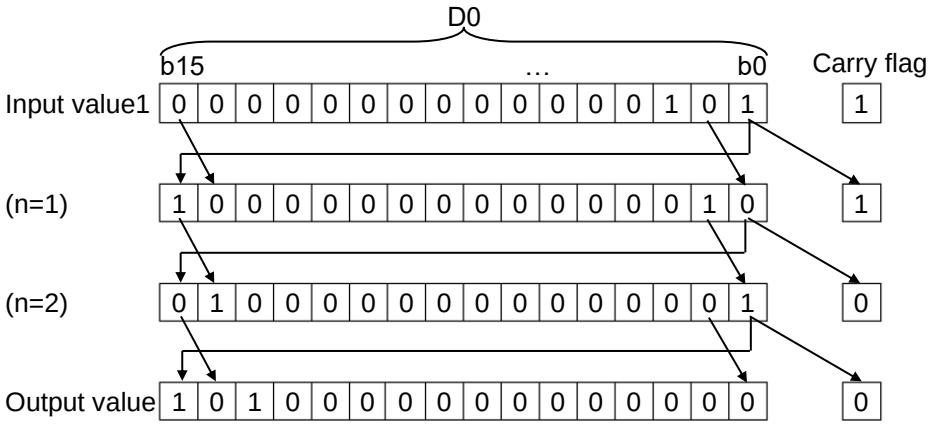
	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1	-	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Input value2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1	-	-	-	-	-	-	-	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○
Output value2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

- Usage example
  D0 is rotated 3 bits to the right and that result is assigned to D0.



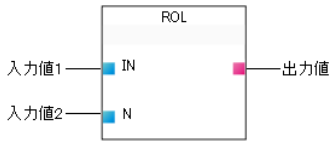


(n=3, Input value5)

- NC(BUILT IN PLC) execution error
  None

ROL

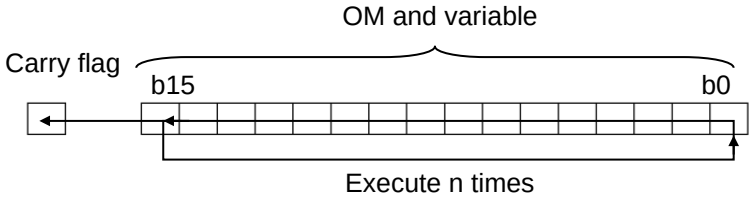
- Function

Function name	Notation	Description
ROL		Rotates the input value 1 data to the left only for the specified bits (input value 2), and that value is output.

(NOTE 1) Specify an integer for the bit number that shifts.
(NOTE 2) When the bit number that shifts is 0, then the input value 1 is output.
(NOTE 3) When the bit number is a negative number, the value that rotates to the right is output.
(NOTE 4) In contrast with the ROL command in the LD language, the variable / OM value specified for input value 1 does not change.
(NOTE 5) Similar to the ROL command in the LD language, in order to shift the variable / OM value specified for input value 1, the same variable / OM is specified to the output value.

- Operation

When the ROL command is executed, each bit is rotated to the left from the low-order to the high-order.
The value for the least significant bit is assigned to the carry flag.



- Description of parameter

Input value1	Data for bit rotation
Input value2	Bit number that rotates to the left
Output value	Data after bit rotation to the left

- Usable data type

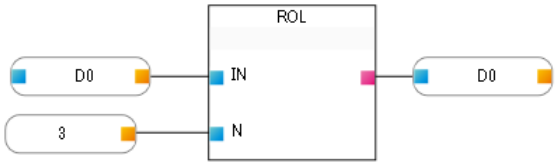
	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1	-	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Input value2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

12

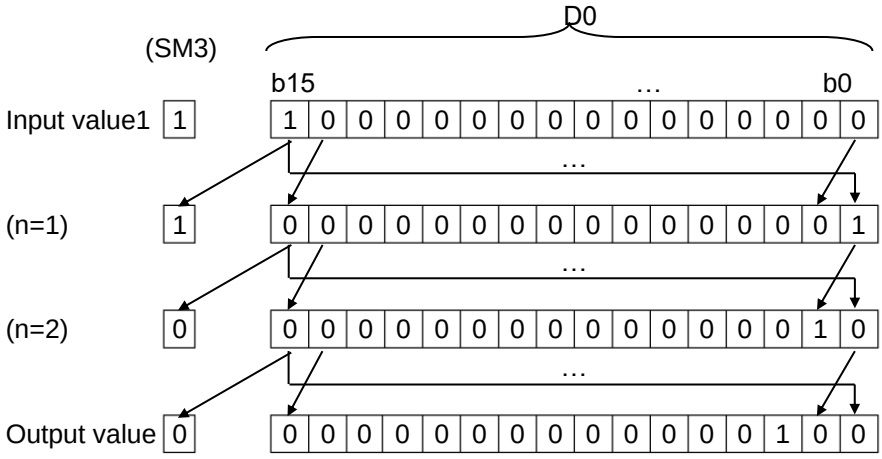
- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1	-	-	-	-	-	-	-	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○
Output value2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

- Usage example
  D0 is rotated 3 bits to the left and that result is assigned to D0.



Carry flag (when the carry flag before execution is 1)



(n=3, Input value-32768)

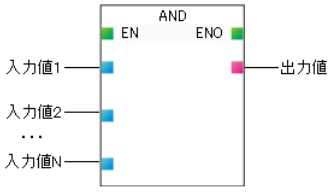
- NC(BUILT IN PLC) execution error
  None

12

12.2.2.6 Bit operation function

AND

- Function

Function name	Notation	Description
AND	 N=2~15	Performs a logical conjunction (AND) operation for the input values.

(NOTE) Refer to "12.2.2.1 Assignment" in the PLC System Manual for details when the data types for the operation result and the "Return value" data are different.

- Description of parameter

Input value1~N	Input value
Output value	Result from logical conjunction operation

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

- Usable OM / Constant

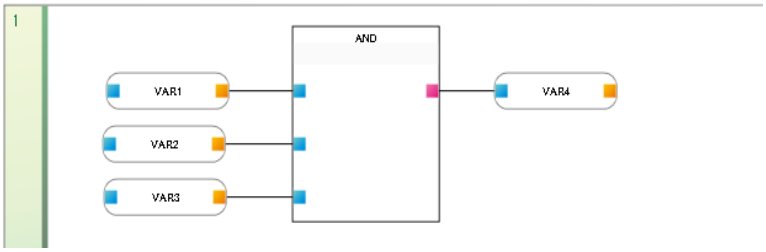
	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

- Usage example

The logical conjunction operation of VAR1, VAR2 and VAR3 is assigned to VAR4.
For example, when the values for VAR1, VAR2 and VAR3 are 2#1111111111111111, 2#0000111111111111 and 2#1111111111110000, respectively, 2#0000111111110000 is assigned to VAR4.

Parameter | External

Scope	Variable name	Data type	Element number	Default value	Comment
VAR	VAR1	WORD	0	2#1111111111111111	
VAR	VAR2	WORD	0	2#0000111111111111	
VAR	VAR3	WORD	0	2#1111111111110000	
VAR	VAR4	WORD	0	16#0000	

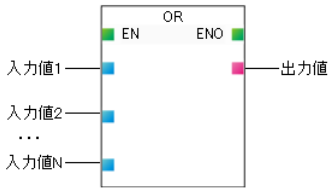


- NC(BUILT IN PLC) execution error

None

OR

- Function

Function name	Notation	Description
OR	 N=2 to 15	Performs a logical disjunction (OR) operation for the input values.

(NOTE) Refer to "12.2.2.1 Assignment" in the PLC System Manual for details when the data types for the operation result and the "Return value" data are different.

- Description of parameter

Input value1~N	Input value
Output value	Result of OR operation

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Input value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

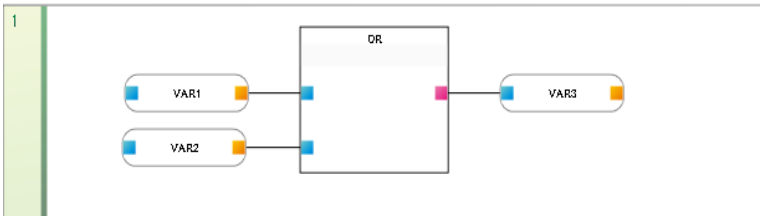
- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

- Usage example
  The logical disjunction (OR) of variables VAR1 and VAR2 is assigned to VAR3.
  For example, when VAR1 and VAR2 are 2#0000111111111111 and 2#1111111111110000, respectively, then 2#1111111111111111 is assigned to VAR3.

Parameter | External

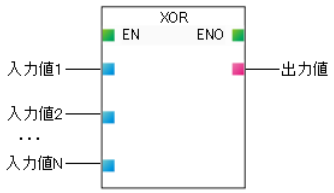
Scope	Variable name	Data type	Element number	Default value	Comment
VAR	VAR1	WORD	0	2#0000111111111111	
VAR	VAR2	WORD	0	2#1111111111110000	
VAR	VAR3	WORD	0	16#0000	



- NC(BUILT IN PLC) execution error
  None

XOR

- Function

Function name	Notation	Description
XOR	 N=2 to 15	Performs an exclusive disjunction (XOR) operation for the input values.

(NOTE) Refer to "12.2.2.1 Assignment" in the PLC System Manual for details when the data types for the operation result and the "Return value" data are different.

- Description of parameter

Input value1~N	Input value
Output value	Result of exclusive disjunction

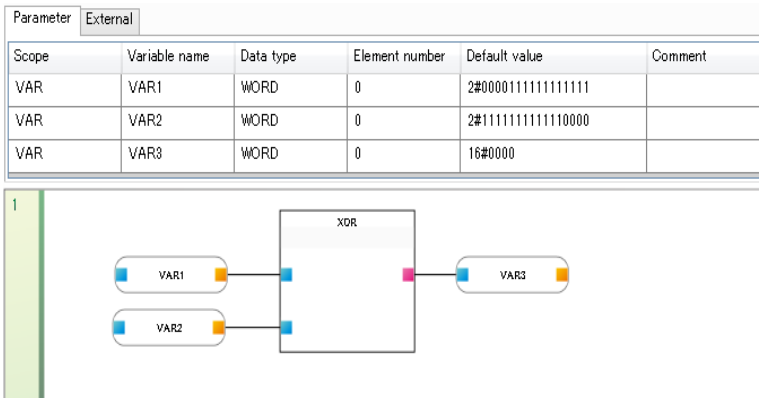
- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Input value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

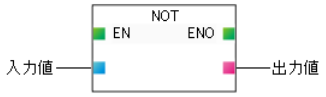
- Usage example
  The exclusive disjunction (XOR) of variables VAR1 and VAR2 is assigned to VAR3.
  For example, when VAR1 and VAR2 are 2#0000111111111111 and 2#1111111111110000, respectively, then 2#1111000000001111 is assigned to VAR3.



- NC(BUILT IN PLC) execution error
  None

NOT

- Function

Function name	Notation	Description
NOT		Outputs the negation operation of the input value.

(NOTE) Refer to "12.2.2.1 Assignment" in the PLC System Manual for details when the data types for the operation result and the "Return value" data are different.

- Description of parameter

Input value	Input value
Output value	Result of negation operation

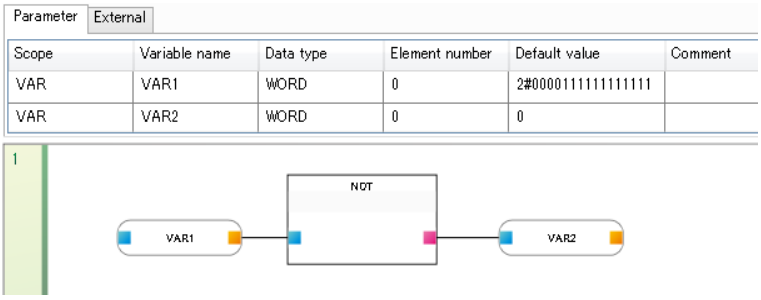
- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Input value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

- Usage example
  The result of the negation operation for variable VAR1 is assigned to variable VAR2.
  For example, when VAR1 is 2#0000111111111111, then 2#1111000000000000 is assigned to VAR2.

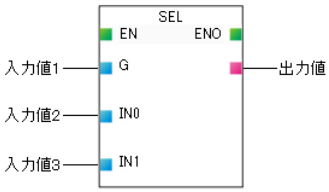


- NC(BUILT IN PLC) execution error
  None

12.2.2.7 Selection function

SEL

- Function

Function name	Notation	Description
SEL		Selects one of two options (input 2 or 3) based on the value (0 or another value that is not 0) of input value 1. When the input value is 0, input value 2 is selected. When the input value is 1, input value 3 is selected.

(NOTE 1) The selections input values 2 and 3 need to be set to the same data type. An error is triggered at conversion if the combination of data types is invalid.

(NOTE 2) Refer to "12.2.2.1 Assignment" in the PLC System Manual for details when the data types for the operation result and the "Return value" data are different.

- Description of parameter

Input value1	Value that determines selection
Input value2	Selected when input value 1 is 0
Input value3	Selected when input value 1 is not 0
Output value	Selection result

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Input value2	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Input value3	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Output value	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

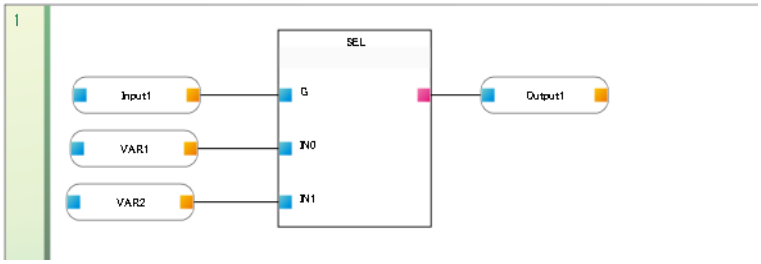
- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1	○	○	○	○	○	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
Input value2	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	○	○	○
Input value3	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	○	○	○
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

- Usage example
  When variable Input1 is FALSE, then 100 (VAR1 value) is assigned to Output1.
  When variable Input1 is TRUE, then 200 (VAR2 value) is assigned to Output1.

Parameter | External

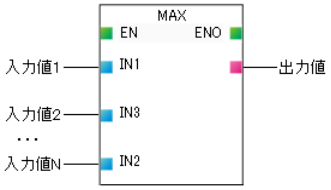
Scope	Variable name	Data type	Element number	Default value	Comment
VAR	Input1	BOOL	0	FALSE	
VAR	VAR1	INT	0	100	
VAR	VAR2	INT	0	200	
VAR	Output1	INT	0	0	



- NC(BUILT IN PLC) execution error
  None

MAX

- Function

Function name	Notation	Description
MAX	 N=2 to 15	Outputs the maximum value from the input values.

(NOTE 1) The selections, for input value 1 to input value N, need to be set to the same data type. An error is triggered at conversion if the combination of data types is invalid.

(NOTE 2) When non-numbers are included in the values for comparison, the input values besides the non-numbers are compared. However, when the last argument of the function is a non-number, then the non-number is output.
When all the input values are non-numbers, then a non-number is output.

(NOTE 3) When the values for comparison are not decimal numerals, then they are converted to signed decimal numerals for comparison.

(NOTE 4) Refer to "12.2.2.1 Assignment" in the PLC System Manual for details when the data types for the operation result and the "Return value" data are different.

- Description of parameter

Input value1~N	Selections
Output value	Maximum value from the selections

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Output value	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

- Comparison of TIME, DATE, TOD and DT types of data
  TIME, DATE, TOD and DT types of data are compared as follows.
  - TIME type: A large amount of time is assessed as large.
  - DATE type: A later date is assessed as large.
  - TOD type: A later time of day is assessed as large.
  - DT type: A later date and time of day are assessed as large.

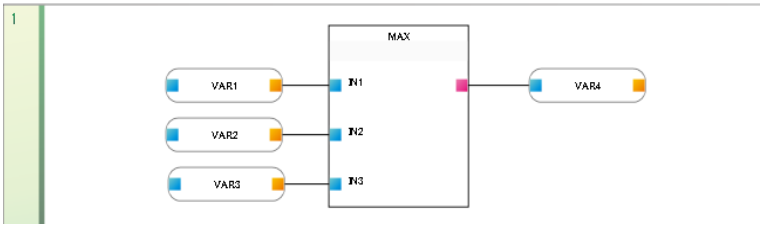
- Comparison of STRING type data
  When comparing the size or amount in STRING type data, the character code is used for the assessment.
  First, the character code for the first character in the character string is compared. Then, if the character code is different, the STRING type data is assessed as large or small based on whether the size or amount of that character code is large or small.
  If the character code for the first character is the same, then the next character is used for the comparison, and this process is repeated from the second character onward until a different character code appears.
  If the number of characters is different, then “16#00” (NULL) is added to the end of the shorter character string for the comparison.

  Ex
  - B' is assessed as larger between 'A' (16#41) and 'B'(16#42).
  - Based on the comparison of the second character, 'AC' is assessed as larger between 'AB' (16#41,16#42) and 'AC'(16#41,16#43).
  - Based on the comparison of the second character, 'AB' is assessed as larger between 'AB' (16#41,16#42) and 'A'(16#41,16#00 is added as the second character).

- Usage example
  The maximum value out of the variables VAR1, VAR2 and VAR3 is assigned to VAR4.
  For example, when variable VAR1:=16#0000000F(15), variable VAR2:=16#FFFFFFFF(-1) and variable VAR3:=16#00000000(0), then 16#0000000F is assigned to variable VAR4.

Parameter | External

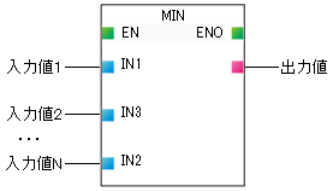
Scope	Variable name	Data type	Element number	Default value	Comment
VAR	VAR1	DWORD	0	16#0000000F	
VAR	VAR2	DWORD	0	16#FFFFFFFF	
VAR	VAR3	DWORD	0	16#00000000	
VAR	VAR4	INT	0	0	



- NC(BUILT IN PLC) execution error
  None

MIN

- Function

Function name	Notation	Description
MIN	 N=2 to 15	Outputs the data for the minimum value out of the input values.

(NOTE 1) The selections, for input value 1 to input value N, need to be set to the same data type. An error is triggered at conversion if the combination of data types is invalid.

(NOTE 2) When non-numbers are included in the values for comparison, the input values besides the non-numbers are compared. However, when the last argument of the function is a non-number, then the non-number is output.
When all the input values are non-numbers, then a non-number is output.

(NOTE 3) When the values for comparison are not decimal numerals, then they are converted to signed decimal numerals for comparison.

(NOTE 4) Refer to "12.2.2.1 Assignment" in the PLC System Manual for details when the data types for the operation result and the "Return value" data are different.

(NOTE 5) Refer to "MAX" for further details on comparing the data types TIME, DATE, TOD, DT and STRING.

- Description of parameter

Input value1~N	Selections
Output value	Minimum value from the selections

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Output value	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

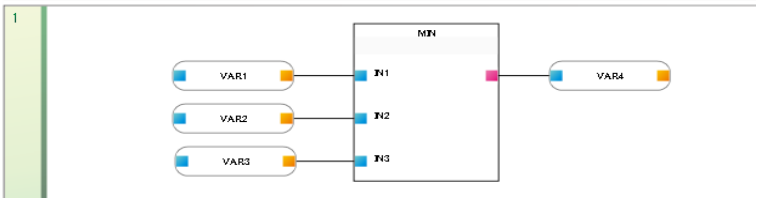
- Usable data type

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

- Usage example
  The minimum value out of the variables VAR1, VAR2 and VAR3 is assigned to VAR4.
  For example, when variable VAR1:=16#0000000F(15), variable VAR2:=16#FFFFFFFF(-1) and variable VAR3:=16#00000000(0), then 16#FFFFFFFF is assigned to variable VAR4.

Parameter | External

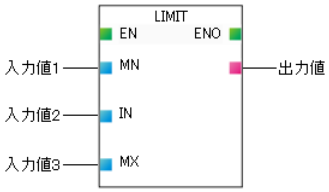
Scope	Variable name	Data type	Element number	Default value	Comment
VAR	VAR1	DWORD	0	16#0000000F	
VAR	VAR2	DWORD	0	16#FFFFFFFF	
VAR	VAR3	DWORD	0	16#00000000	
VAR	VAR4	INT	0	0	



- NC(BUILT IN PLC) execution error
  None

LIMIT

Function

Function name	Notation	Description
LIMIT		Assesses the upper limit and lower limit. Input value 1 is the lower limit and input value 3 is the upper limit. Input value 2 is the value that is compared. If input value 2 does not exceed the upper or lower limits, then input value 2 value is assigned to the output value. If input value 2 exceeds the upper limit, or falls below the lower limit, then the upper limit and lower limit are assigned accordingly.

(NOTE 1) The input values 1 to 3 need to be set to the same data type. An error is triggered at conversion if the combination of data types is invalid.

(NOTE 2) If the upper and lower limits are reversed, the following operations apply.
- When the comparison value is smaller than the upper limit, then the lower limit is output.
- When the comparison value is larger than the upper limit, then the upper limit is output.

(NOTE 3) When the comparison value is a non-number, then a non-number is output.
When the lower limit is a non-number, if the comparison value is less than the upper limit, then the comparison value is output.

(NOTE 4) When the values for comparison are not decimal numerals, then they are converted to signed decimal numerals for comparison.

(NOTE 5) Refer to “12.2.2.1 Assignment” in the PLC System Manual for details when the data types for the operation result and the “Return value” data are different.

(NOTE 6) Refer to “MAX” for further details on comparing the data types TIME, DATE, TOD, DT and STRING.

- Description of parameter

Input value1	Lower limit (minimum)
Input value2	Comparison value
Input value3	Upper limit (maximum)
Output value	Result of comparison (when the comparison value exceeds upper/lower limit, then the result is the upper/lower limit; when the comparison value is within the specified range, then the result is the comparison value)

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Input value2	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Input value3	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Output value	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

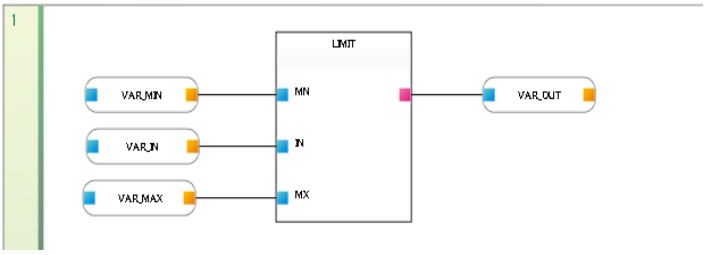
- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	○	○	○
Input value2	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○
Input value3	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	○	○	○
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

- Usage example
  The lower limit is VAR_MIN, the upper limit is VAR_MAX and the comparison value is VAR_IN.
  For example, when VAR_MIN is 16#0000 and VAR_MAX is 16#000F, the processing below applies.
  When VAR_IN is smaller than 16#0000, then 16#0000(VAR_MIN) is assigned to VAR_OUT.
  When VAR_IN is greater than 16#0000 and less than 16#000F, then the VAR_IN value is assigned to VAR_OUT.
  When VAR_IN is larger than 16#000F, then 16#000F(VAR_MAX) is assigned to VAR_OUT.

Parameter | External

Scope	Variable name	Data type	Element number	Default value	Comment
VAR	VAR_MIN	WORD	0	16#0000	
VAR	VAR_MAX	WORD	0	16#0000	
VAR	VAR_IN	WORD	0	16#0000	
VAR	VAR_OUT	WORD	0	16#0000	

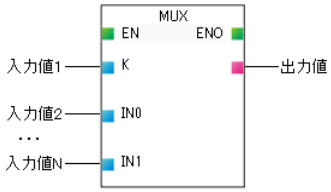


- NC(BUILT IN PLC) execution error
  None

12

MUX

- Function

Function name	Notation	Description
MUX	 N=2 to 15	Selects the value to be output based on input value 1. Input value 1 = 0: Outputs input value 2 Input value 1 = 1: Outputs input value 3 … Input value 1 = N-1: Outputs input value N If input value 1 is not within this range, then 0 is output.

(NOTE 1) The selections, for input value 2 to input value N, need to be set to the same data type. An error is triggered at conversion if the combination of data types is invalid.

(NOTE 2) Refer to "12.2.2.1 Assignment" in the PLC System Manual for details when the data types for the operation result and the "Return value" data are different.

- Description of parameter

Input value1	Specifies selection
Input value2~N	Selections
Output value	Selected value

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Input value2~N	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Output value	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

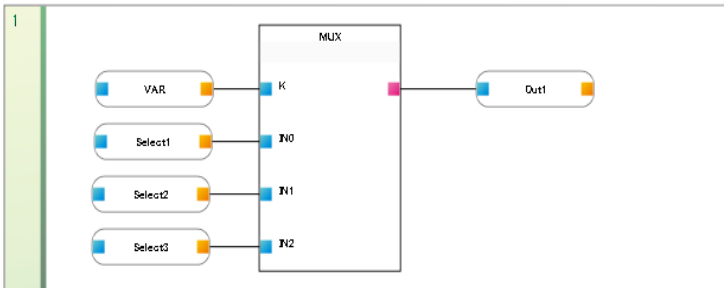
- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○
Input value2~N	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	○	○	○
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

- Usage example
  The values for the variables from Select1 to Select3 are assigned to Out1 depending on the variable VAR value.
  For example, when VAR = 0, then Select1 value (111) is assigned to Out1.

Parameter | External

Scope	Variable name	Data type	Element number	Default value	Comment
VAR	VAR	INT	0	0	
VAR	Select1	INT	0	111	
VAR	Select2	INT	0	222	
VAR	Select3	INT	0	333	
VAR	Out1	INT	0	0	

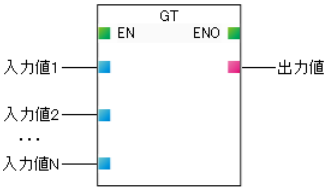


- NC(BUILT IN PLC) execution error
  None

12.2.2.8 Comparison function

GT

- Function

Function name	Notation	Description
GT	 N=2 to 15	Compares the input values and when the following applies: Input value 1 > Input value 2 >…> Input value N, then 1 is output. In all other situations, 0 is output.

(NOTE 1) The input values need to be set to the same data type. An error is triggered at conversion if the combination of data types is invalid.
(NOTE 2) When non-numbers are included in the values for comparison, 0 is output.
(NOTE 3) When an integer and a real number are compared, the integer is converted to a real number for comparison.
(NOTE 4) When the values for comparison are not decimal numerals, then they are converted to signed decimal numerals for comparison.
(NOTE 5) Refer to "MAX" for further details on comparing the data types TIME, DATE, TOD, DT and STRING.

- Description of parameter

Input value1~N	Input value
Output value	0: When the conditions are false 1: When the conditions are true

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

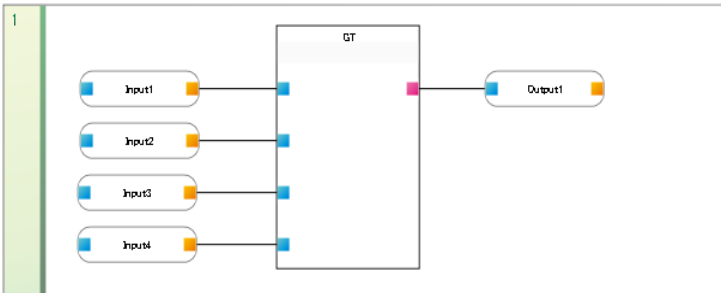
- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	○	○	○
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

- Usage example
  The input values from Input1 to Input4 are compared, and the result is assigned to Output1.
  Input1 to Input4 are all INT variables.
  For example, when Input1 = 10, Input2 = 5, Input3 = 0 and Input4 = -1, then the following condition is true: Input1 > Input2 > Input3 > Input4. 1 is stored in Output1.

Parameter | External

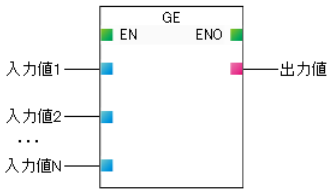
Scope	Variable name	Data type	Element number	Default value	Comment
VAR	Input1	INT	0	10	
VAR	Input2	INT	0	5	
VAR	Input3	INT	0	0	
VAR	Input4	INT	0	-1	
VAR	Output1	INT	0	0	



- NC(BUILT IN PLC) execution error
  None

GE

- Function

Function name	Notation	Description
GE	 N=2 to 15	Compares the input values and when the following applies: Input value 1 ≥ Input value 2 ≥…≥ Input value N, then 1 is output. In all other situations, 0 is output.

(NOTE 1) The input values need to be set to the same data type. An error is triggered at conversion if the combination of data types is invalid.
(NOTE 2) When non-numbers are included in the values for comparison, 0 is output.
(NOTE 3) When an integer and a real number are compared, the integer is converted to a real number for comparison.
(NOTE 4) When the values for comparison are not decimal numerals, then they are converted to signed decimal numerals for comparison.
(NOTE 5) Refer to "MAX" for further details on comparing the data types TIME, DATE, TOD, DT and STRING.

- Description of parameter

Input value1~N	Input value
Output value	0: When the conditions are false 1: When the conditions are true

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

- Usable OM / Constant

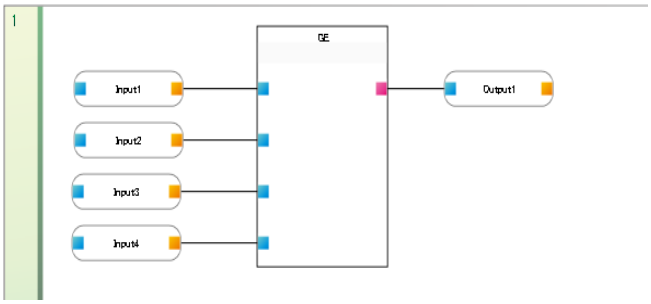
	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	○	○	○
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

12

- Usage example
  The input values from Input1 to Input4 are compared, and the result is assigned to Output1.
  Input1 to Input4 are all INT variables.
  For example, when Input1 = 10, Input2 = 5, Input3 = 5 and Input4 = -1, then the following condition is true: Input1 >= Input2 >= Input3 >= Input4. 1 is stored in Output1.

Parameter | External

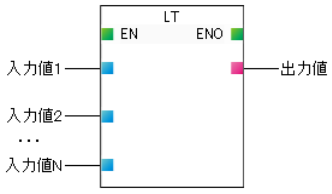
Scope	Variable name	Data type	Element number	Default value	Comment
VAR	Input1	INT	0	10	
VAR	Input2	INT	0	5	
VAR	Input3	INT	0	5	
VAR	Input4	INT	0	-1	
VAR	Output1	INT	0	0	



- NC(BUILT IN PLC) execution error
  None

LT

- Function

Function name	Notation	Description
LT	 N=2 to 15	Compares the input values and when the following applies: Input value 1 < Input value 2 <…< Input value N, then 1 is output. In all other situations, 0 is output.

(NOTE 1) The input values need to be set to the same data type. An error is triggered at conversion if the combination of data types is invalid.
(NOTE 2) When non-numbers are included in the values for comparison, 0 is output.
(NOTE 3) When an integer and a real number are compared, the integer is converted to a real number for comparison.
(NOTE 4) When the values for comparison are not decimal numerals, then they are converted to signed decimal numerals for comparison.
(NOTE 5) Refer to "MAX" for further details on comparing the data types TIME, DATE, TOD, DT and STRING.

- Description of parameter

Input value1~N	Input value
Output value	0: When the conditions are false 1: When the conditions are true

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

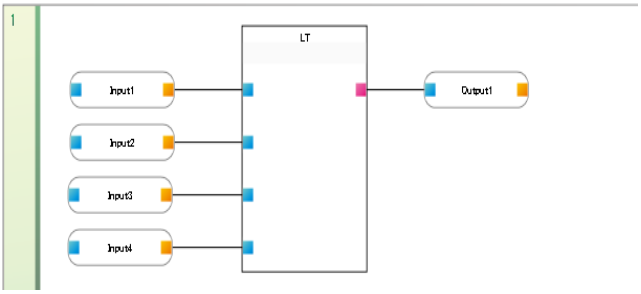
- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	○	○	○
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

- Usage example
  The input values from Input1 to Input4 are compared, and the result is assigned to Output1.
  Input1 to Input4 are all INT variables.
  For example, when Input1 = -10, Input2 = -5, Input3 = 0 and Input4 = 1, then the following condition is true: Input1 < Input2 < Input3 < Input4. 1 is stored in Output1.

Parameter | External

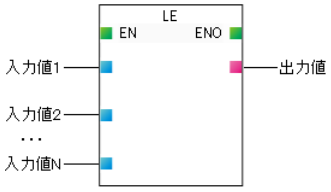
Scope	Variable name	Data type	Element number	Default value	Comment
VAR	Input1	INT	0	-10	
VAR	Input2	INT	0	-5	
VAR	Input3	INT	0	0	
VAR	Input4	INT	0	1	
VAR	Output1	INT	0	0	



- NC(BUILT IN PLC) execution error
  None

LE

- Function

Function name	Notation	Description
LE	 N=2 to 15	Compares the input values and when the following applies: Input value 1 ≤ Input value 2 ≤…≤ Input value N, then 1 is output. In all other situations, 0 is output.

(NOTE 1) The input values need to be set to the same data type. An error is triggered at conversion if the combination of data types is invalid.
(NOTE 2) When non-numbers are included in the values for comparison, 0 is output.
(NOTE 3) When an integer and a real number are compared, the integer is converted to a real number for comparison.
(NOTE 4) When the values for comparison are not decimal numerals, then they are converted to signed decimal numerals for comparison.
(NOTE 5) Refer to "MAX" for further details on comparing the data types TIME, DATE, TOD, DT and STRING.

- Description of parameter

Input value1~N	Input value
Output value	0: When the conditions are false 1: When the conditions are true

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

- Usable OM / Constant

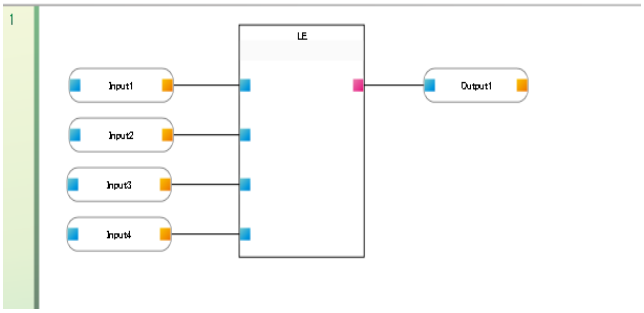
	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	○	○	○
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

12

- Usage example
  The input values from Input1 to Input4 are compared, and the result is assigned to Output1.
  Input1 to Input4 are all INT variables.
  For example, when Input1 = -10, Input2 = 0, Input3 = 0 and Input4 = 1, then the following condition is true: Input1 <= Input2 <= Input3 <= Input4. 1 is stored in Output1.

Parameter | External

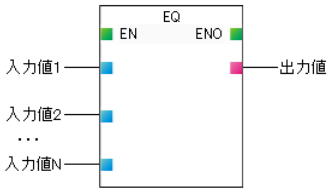
Scope	Variable name	Data type	Element number	Default value	Comment
VAR	Input1	INT	0	-10	
VAR	Input2	INT	0	0	
VAR	Input3	INT	0	0	
VAR	Input4	INT	0	1	
VAR	Output1	INT	0	0	



- NC(BUILT IN PLC) execution error
  None

EQ

- Function

Function name	Notation	Description
EQ	 N=2 to 15	Outputs 1 when all the input values are equal. In all other situations, 0 is output.

(NOTE 1) The input values need to be set to the same data type. An error is triggered at conversion if the combination of data types is invalid.
(NOTE 2) When non-numbers are included in the values for comparison, 0 is output.
(NOTE 3) When an integer and a real number are compared, the integer is converted to a real number for comparison.
(NOTE 4) When the values for comparison are not decimal numerals, then they are converted to signed decimal numerals for comparison.
(NOTE 5) Refer to "MAX" for further details on comparing the data types TIME, DATE, TOD, DT and STRING.

- Description of parameter

Input value1~N	Input value
Output value	0: When the conditions are false 1: When the conditions are true

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

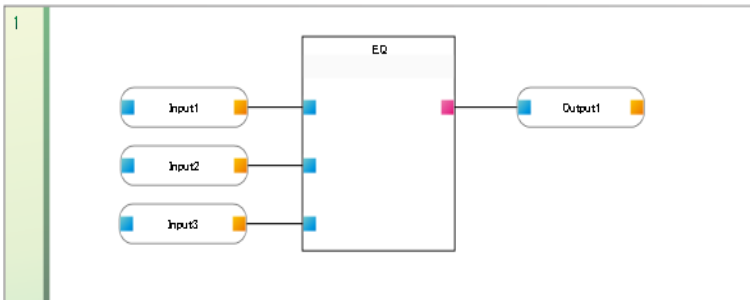
- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	○	○	○
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

- Usage example
  The input values from Input1 to Input3 are compared, and the result is assigned to Output1.
  Input1 to Input4 are all INT variables.
  For example, when Input1 = 10, Input2 = 10 and Input3 = 10, then the following condition is true: Input1 = Input2 = Input3. 1 is stored in Output1.

Parameter | External

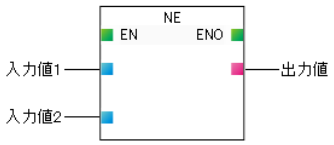
Scope	Variable name	Data type	Element number	Default value	Comment
VAR	Input1	INT	0	10	
VAR	Input2	INT	0	10	
VAR	Input3	INT	0	10	
VAR	Output1	INT	0	0	



- NC(BUILT IN PLC) execution error
  None

NE

- Function

Function name	Notation	Description
NE		Outputs 1 when the input values are different. In all other situations, 0 is output.

(NOTE 1) The input values need to be set to the same data type. An error is triggered at conversion if the combination of data types is invalid.
(NOTE 2) When non-numbers are included in the values for comparison, 1 is output.
(NOTE 3) When an integer and a real number are compared, the integer is converted to a real number for comparison.
(NOTE 4) When the values for comparison are not decimal numerals, then they are converted to signed decimal numerals for comparison.
(NOTE 5) Refer to "MAX" for further details on comparing the data types TIME, DATE, TOD, DT and STRING.

- Description of parameter

Input value	Input value
Output value	0: When the conditions are false 1: When the conditions are true

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

- Usable OM / Constant

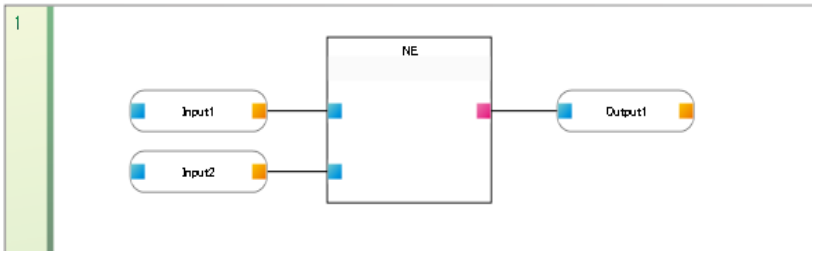
	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	○	○	○
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

12

- Usage example
  The variables Input1 and Input2 are compared, if the values are not equal, then 1 is assigned to the output value Output1. If the values are equal, then 0 is assigned.
  For example, when Input1 = 1 and input2 = -1, then Input1 < > Input2 is true and 1 is assigned to Output1.

Parameter | External

Scope	Variable name	Data type	Element number	Default value	Comment
VAR	Input1	INT	0	1	
VAR	Input2	INT	0	-1	
VAR	Output1	INT	0	0	

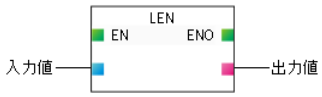


- NC(BUILT IN PLC) execution error
  None

12.2.2.9 Character string function

LEN

- Function

Function name	Notation	Description
LEN		Calculates the number of characters in the character string (input value).

(NOTE 1) When the input value is NULL, the number of characters is processed as 0.
(NOTE 2) Use ASCII code for the character code of the input value.

- Description of parameter

Input value	Input value (character string)
Output value	Number of characters in character string

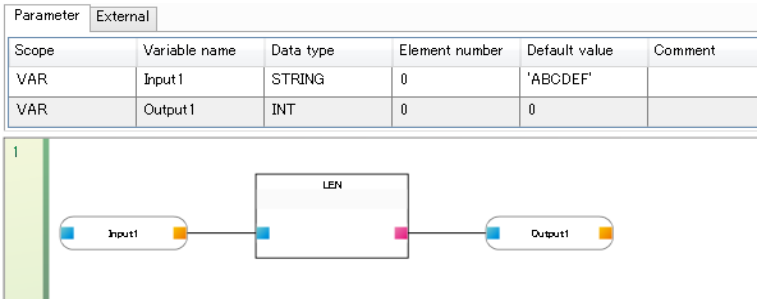
- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
Output value	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

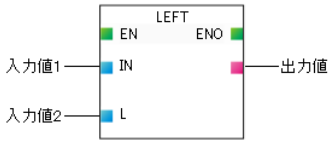
- Usage example
  The number of characters in the character string, stored in the STRING type variable Input1, is assigned to variable Output1.
  For example, when the character string 'ABCDEF' is stored in Input1, then 6 is assigned to the variable Output1.



- NC(BUILT IN PLC) execution error
  None

LEFT

- Function

Function name	Notation	Description
LEFT		Acquires the specified number of characters (input value 2), starting from the left, in the character string for the input value.

(NOTE 1) When input value 1 is NULL, then the NULL characters are output.

(NOTE 2) When 0 or a negative number is entered for input value 2, then the NULL characters are output.

(NOTE 3) When input value 2 is greater than the number of characters, then all characters entered for input value 1 are acquired. In addition, any difference between input value 2 and the number of characters is ignored.
For example, when the character string “STRING” is assigned to input value 1 and input value 2 is 7, since “STRING” is 6 characters long, all the characters “STRING” are assigned to the output value.

- Description of parameter

Input value1	Character string
Input value2	Specifies number of characters to acquire
Output value	Acquired character string

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
Input value2	-	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Output value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○

- Usable OM / Constant

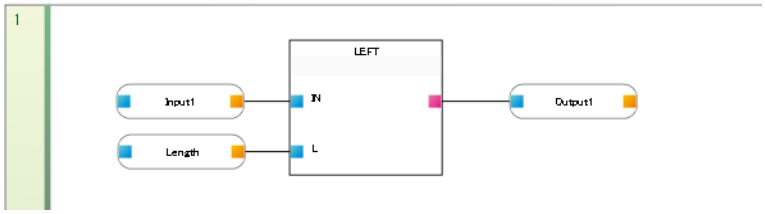
	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
Input value2	○*1	○*1	○*1	○*1	○*1	○*1	○*1	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Output value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*1 Possible only when digits are specified.

- Usage example
  A specified number of characters (specified length starting from the left) in the character string Input1 is assigned to Output1.
  For example, when Input1 is 'ABCDE' and the Length is 2, then the two characters 'AB' (starting at the left of 'ABCDE') are assigned to Output1.

Parameter | External

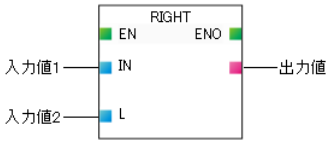
Scope	Variable name	Data type	Element number	Default value	Comment
VAR	Input1	STRING	0	'ABCDE'	
VAR	Length	INT	0	2	
VAR	Output1	STRING	0	''	



- NC(BUILT IN PLC) execution error
  None

RIGHT

- Function

Function name	Notation	Description
RIGHT		Acquires the specified number of characters (input value 2), starting from the right, in the character string for input value 1.

(NOTE 1) When the input value is NULL, then the NULL characters are output.

(NOTE 2) When 0 or a negative number is entered for input value 2, then the NULL characters are output.

(NOTE 3) When input value 2 is greater than the number of characters, then all characters entered for input value 1 are acquired. In addition, any difference between input value 2 and the number of characters is ignored.
For example, when the character string “STRING” is assigned to input value 1 and input value 2 is 7, since “STRING” is 6 characters long, all the characters “STRING” are assigned to the output value.

- Description of parameter

Input value1	Character string
Input value2	Specifies number of characters to acquire
Output value	Acquired character string

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
Input value2	-	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Output value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○

- Usable OM / Constant

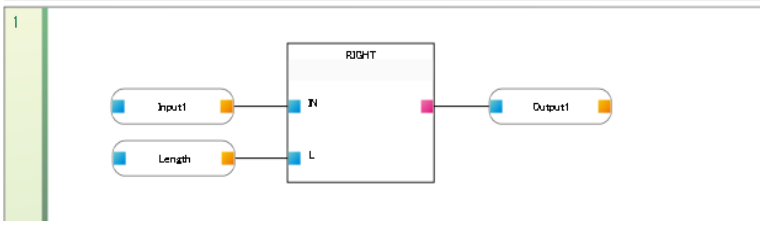
	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
Input value2	○*1	○*1	○*1	○*1	○*1	○*1	○*1	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Output value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*1 Possible only when digits are specified.

- Usage example
  A specified number of characters (specified length starting from the right) in the character string Input1 is assigned to Output1.
  For example, when Input1 is 'ABCDE' and the Length is 2, then the two characters 'DE' (starting at the right of 'ABCDE') are assigned to Output1.

Parameter | External

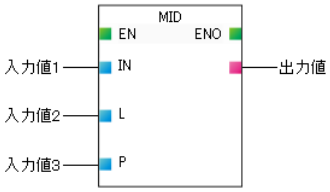
Scope	Variable name	Data type	Element number	Default value	Comment
VAR	Input1	STRING	0	'ABCDE'	
VAR	Length	INT	0	2	
VAR	Output1	STRING	0	''	



- NC(BUILT IN PLC) execution error
  None

MID

- Function

Function name	Notation	Description
MID		Acquires the specified number (input value 3) of characters in the character string of input value 1, starting from the specified position (input value 2) moving to the right.

(NOTE 1) The position of the beginning character in input value 1 is processed as 1.
(NOTE 2) When the input value is NULL, then the NULL is output.
(NOTE 3) When 0 or a negative number is specified for input value 2, then NULL is output.
(NOTE 4) When input value 2 is greater than the number of characters from the specified position (moving to the right), then the entire character string from the specified position is acquired.
(NOTE 5) When the start position (input value 3) is greater than the number of characters (input value 1), or when it is a 0 or negative number, then NULL is output.

- Description of parameter

Input value1	Character string
Input value2	Number of characters to be acquired
Input value3	Start position for acquiring characters
Output value	Acquired character string

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
Input value2	-	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Input value3	-	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Output value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○

- Usable OM / Constant

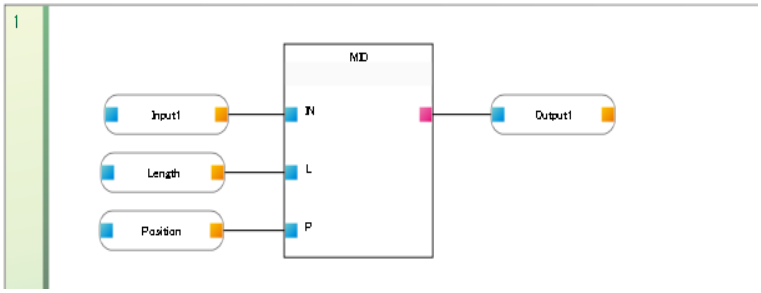
	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
Input value2	○*1	○*1	○*1	○*1	○*1	○*1	○*1	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Input value3	○*1	○*1	○*1	○*1	○*1	○*1	○*1	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Output value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*1 Possible only when digits are specified.

- Usage example
  The number of characters specified in the Length starting from the indicated Position are acquired in the character string Input1 and assigned to Output1.
  For example, the character string 'STRING' is assigned to Input1. When Length = 2 (number of characters) and Position = 2 are specified, then the character string 'TR' is assigned to Output1.

Parameter | External

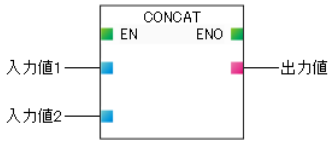
Scope	Variable name	Data type	Element number	Default value	Comment
VAR	Input1	STRING	0	'STRING'	
VAR	Length	INT	0	2	
VAR	Position	INT	0	2	
VAR	Output1	STRING	0	''	



- NC(BUILT IN PLC) execution error
  None

CONCAT

- Function

Function name	Notation	Description
CONCAT		Joins or concatenates the character strings input value 1 and input value 2.

(NOTE 1) The character string for input value 2 is joined to the right of the character string for input value 1.

(NOTE 2) If the concatenated character string is equal to or longer than 32 characters, any character from the 32nd character is ignored.
For example, if input value 1 is "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA" (30 As) and input value 2 is "BC", then the concatenated character string becomes "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAB" (31 characters).

- Description of parameter

Input value1,2	Character strings that are joined (concatenated)
Output value	Acquired character string

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1,2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
Output value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○

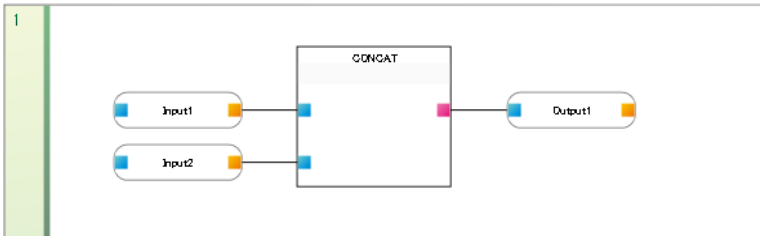
- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1,2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
Output value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

- Usage example
  The concatenated character string that joins Input1 and Input2 is assigned to Output1.
  For example, when ‘STRING’ is stored for Input1 and ‘TEXT’ is stored for Input2, then the character string ‘STRINGTEXT’ is assigned to Output1.

Parameter | External

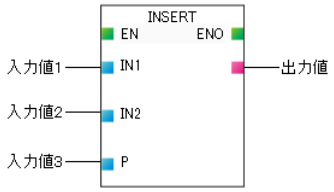
Scope	Variable name	Data type	Element number	Default value	Comment
VAR	Input1	STRING	0	'STRING'	
VAR	Input2	STRING	0	'TEXT'	
VAR	Output1	STRING	0	''	



- NC(BUILT IN PLC) execution error
  None

INSERT

- Function

Function name	Notation	Description
INSERT		Inserts one character string (input value 2) at the specified position (input value 3) of another character string (input value 1).

(NOTE 1) Input value 2 character string is inserted to the right of the character position specified by input value 3.
The position of the beginning character in input value 1 is processed as 1.

(NOTE 2) When input value 3 is greater than the number of characters for input value 1, then the input value 1 character string is output.

(NOTE 3) When input value 3 is 0 or a negative number, then the input value 2 character string is inserted to the left of the input value 1 character string.

(NOTE 4) If the output value is more than 31 characters, any character from the 32nd character is ignored.

- Description of parameter

Input value1	Character string
Input value2	Inserted character string
Input value3	Insertion position
Output value	Character string after insertion

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
Input value2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
Input value3	-	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Output value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○

- Usable OM / Constant

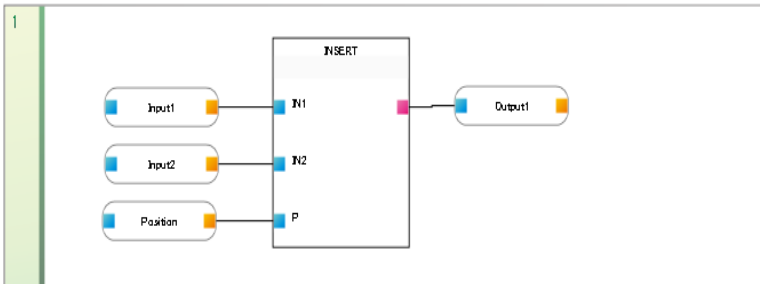
	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
Input value2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
Input value3	○*1	○*1	○*1	○*1	○*1	○*1	○*1	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Output value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*1 Possible only when digits are specified.

- Usage example
  The character string Input2 is inserted at the specified Position for the character string Input1. After the insertion, the character string is assigned to Output1.
  For example, when Input1 is ‘ABCDEF’, Input2 is ‘XYZ’ and the Position is 2, then ‘ABXYZCDEF’ is assigned to Output1.

Parameter | External

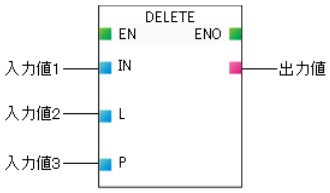
Scope	Variable name	Data type	Element number	Default value	Comment
VAR	Input1	STRING	0	'ABCDEF'	
VAR	Input2	STRING	0	'XYZ'	
VAR	Position	INT	0	2	
VAR	Output1	STRING	0	''	



- NC(BUILT IN PLC) execution error
  None

DELETE

- Function

Function name	Notation	Description
DELETE		Deletes only the specified number of characters (input value 2) from the specified position (input value 3) in the character string (input value 1).

(NOTE 1) The position of the beginning character in input value 1 is processed as 1.
(NOTE 2) When input value 2 is greater than the number of characters from the specified position (moving to the right), then all the characters from the specified position are deleted.
(NOTE 3) When input value 3 is greater than the number of characters for input value 1, then the input value 1 character string is output.
(NOTE 4) When 0 or a negative number is entered for input value 2 and 3, then the NULL characters are output.

- Description of parameter

Input value1	Input value (character string)
Input value2	Number of characters to be deleted
Input value3	Delete start position
Output value	Character string after deletion

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
Input value2	-	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Input value3	-	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Output value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○

- Usable OM / Constant

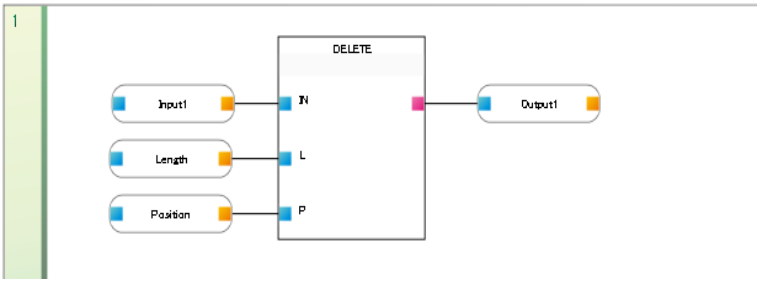
	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
Input value2	○*1	○*1	○*1	○*1	○*1	○*1	○*1	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Input value3	○*1	○*1	○*1	○*1	○*1	○*1	○*1	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Output value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*1 Possible only when digits are specified.

- Usage example
  The number of characters specified by the Length is deleted at the specified Position for the character string Input1. After the deletion, the character string is assigned to Output1.
  For example, when Input1 is ‘ABCDEF’, the Length is 3 and the Position is 2, then ‘AEF’ is assigned to the output value.

Parameter | External

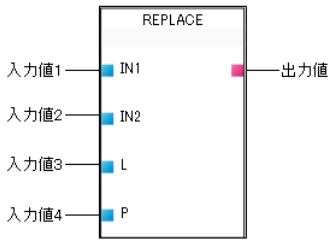
Scope	Variable name	Data type	Element number	Default value	Comment
VAR	Input1	STRING	0	'ABCDEF'	
VAR	Length	INT	0	3	
VAR	Position	INT	0	2	
VAR	Output1	STRING	0	''	



- NC(BUILT IN PLC) execution error
  None

REPLACE

- Function

Function name	Notation	Description
REPLACE		Replaces part of a character string with a specific character string. The specified number of characters (input value 3) from the specified position (input value 4) is replaced in the input value 1 character string. That section is replaced with the input value 2 character string.

(NOTE 1) The position of the beginning character in input value 1 is processed as 1.

(NOTE 2) When input value 4 is greater than the number of characters for input value 1, then the input value 2 character string is inserted behind the input value 1 character string.

(NOTE 3) When input value 3 is greater than the number of characters from the position specified by input value 4 (moving to the right), then the entire character string to the right of the position specified by input value 4 is replaced.

(NOTE 4) When 0 or a negative number is specified for input value 3 and 4, then NULL is output.

(NOTE 5) If the character string after the replacement is equal to or longer than 32 characters, any character from the 32nd character is ignored.

- Description of parameter

Input value1	Character string where replacement is carried out
Input value2	Character string inserted for replacement
Input value3	Number of characters used in replacement
Input value4	Replacement start position
Output value	Character string after replacement

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
Input value2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
Input value3	-	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Input value4	-	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Output value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○

- Usable OM / Constant

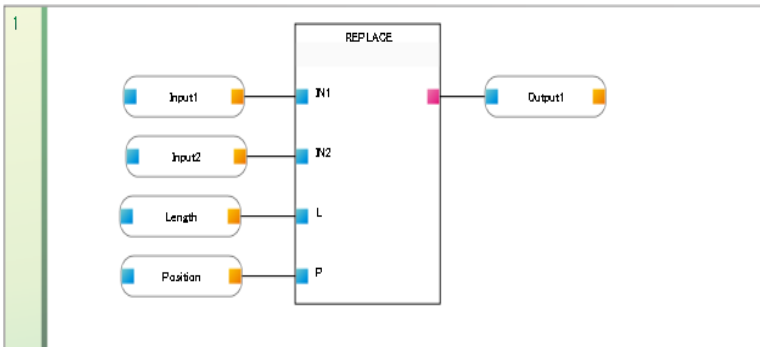
	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BD X	BD Y	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
Input value2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
Input value3	○*1	○*1	○*1	○*1	○*1	○*1	○*1	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Input value4	○*1	○*1	○*1	○*1	○*1	○*1	○*1	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Output value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*1 Possible only when digits are specified.

- Usage example
  The number of characters in the Length at the specified Position for the character string Input1 is replaced with the Input2 character string. After the replacement, the character string is assigned to Output1. For example, when Input1 is ‘ABCDEF’, Input2 is ‘XYZ’, the Length is 3 and the Position is 2, then ‘AXYZEF’ is assigned to the output value.

Parameter | External

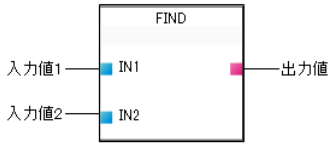
Scope	Variable name	Data type	Element number	Default value	Comment
VAR	Input1	STRING	0	'ABCDEF'	
VAR	Input2	STRING	0	'XYZ'	
VAR	Length	INT	0	3	
VAR	Position	INT	0	2	
VAR	Output1	STRING	0	''	



- NC(BUILT IN PLC) execution error
  None

FIND

- Function

Function name	Notation	Description
FIND		Assigns the specified position in the character string to the output value.

(NOTE 1) If there are multiple search results, the first matching place from the beginning is output.
(NOTE 2) The position of the beginning character in input value 1 is counted as 1.
(NOTE 3) The letters are case sensitive when carrying out a search.
(NOTE 4) If a character string is not specified for input value 1 or input value 2, then 0 is output.

- Description of parameter

Input value1	Character string where search is carried out
Input value2	Character string used in search
Output value	Search result 1 or thereafter: Position of character string found in search 0: When there is no search conducted

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
Input value2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
Output value	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

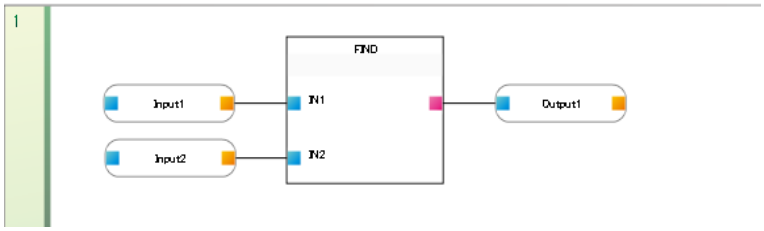
- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
Input value2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	○	○	○

- Usage example
  A search for character string Input2 is conducted in the character string Input1, and the search result is assigned to Output1.
  For example, when Input1 is ‘ABCDEF’ and Input2 is ‘BC’, then 2 is assigned to Output1.

Parameter | External

Scope	Variable name	Data type	Element number	Default value	Comment
VAR	Input1	STRING	0	'ABCDEF'	
VAR	Input2	STRING	0	'BC'	
VAR	Output1	INT	0	0	

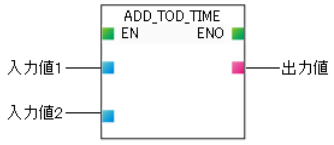


- NC(BUILT IN PLC) execution error
  None

12.2.2.10 Time data type function

ADD_TIME

- Function

Function name	Notation	Description
ADD_TIME		Adds the time (input values).

(NOTE) If the operation result exceeds the TIME type range, an error is triggered.

- Description of parameter

Input value1,2	Time added (TIME type)
Output value	Value after adding (TIME type)

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
Input value2	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
Output value	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-

- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
Input value2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
Output value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

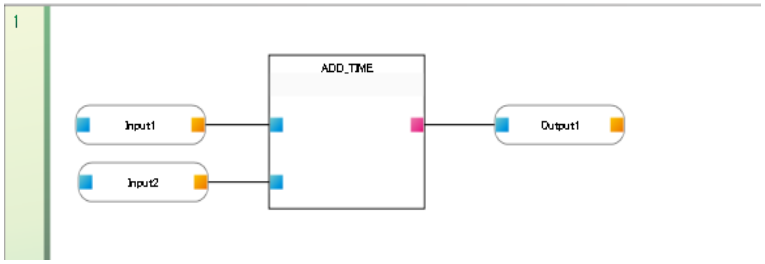
*1 Possible only with TIME type.

- Usage example

The TIME type variables Input1 and Input2 are added, and the result is assigned to Output1.
For example, when T#100:20:30 and T#011:11:11 are stored for Input1 and Input2, respectively, then T#111:31:41: is stored for Output1.

Parameter | External

Scope	Variable name	Data type	Element number	Default value	Comment
VAR	Input1	TIME	0	T#100:20:30	
VAR	Input2	TIME	0	T#011:11:11	
VAR	Output1	TIME	0	T#000:00:00	

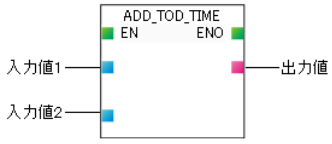


- NC(BUILT IN PLC) execution error

When the operation result exceeds the TIME type range. (Error code 14)

ADD_TOD_TIME

- Function

Function name	Notation	Description
ADD_TOD_TIME		Adds time (input value 2) to time of day (input value 1).

(NOTE) If the operation result exceeds the TOD type range, an error is triggered.

- Description of parameter

Input value1	Time of day that is added (TOD type)
Input value2	Time that is added (TIME type)
Output value	Time of day after addition (TOD type)

- Usable data type

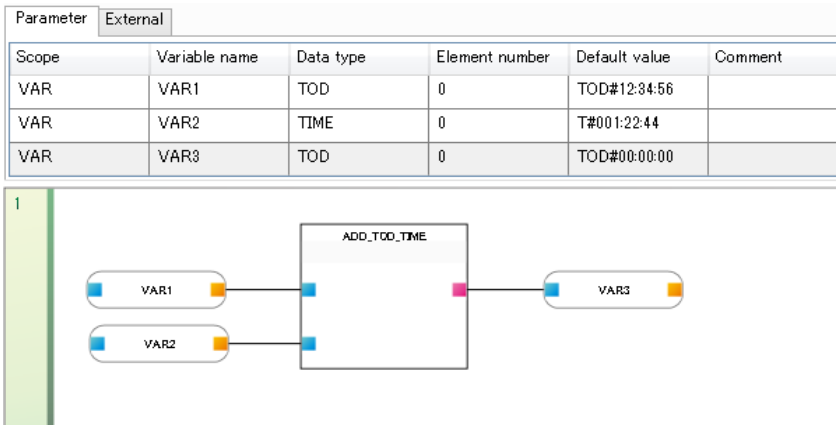
	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
Input value2	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
Output value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-

- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
Input value2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*2	-	-	-
Output value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*1 Possible only with TOD type.
*2 Possible only with TIME type.

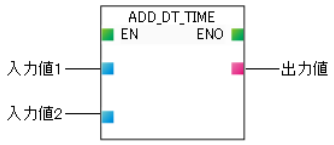
- Usage example
  The TIME type VAR1 is added to the TIME type VAR2, and the result is assigned to TOD type VAR3.
  Ex When the time T#001:22:44 is added to TOD#12:34:56, then TOD#13:57:40 is assigned to the output value.



- NC(BUILT IN PLC) execution error
  When the operation result exceeds the TOD type range. (Error code 14)

ADD_DT_TIME

- Function

Function name	Notation	Description
ADD_DT_TIME		Adds time (input value 2) to date and time (input value 1).

(NOTE 1) Leap year is also taken into account in the add operation.
For example, the following shows how it applies: DT#2020-02-28-23:59:59 + T#000:00:01 = DT#2020-02-29-00:00:00

(NOTE 2) If the operation result exceeds the DT type range, an error is triggered.

- Description of parameter

Input value1	Date and time that is added to (DT type)
Input value2	Time that is added (TIME type)
Output value	Date and time after addition (DT type)

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-
Input value2	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
Output value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-

- Usable OM / Constant

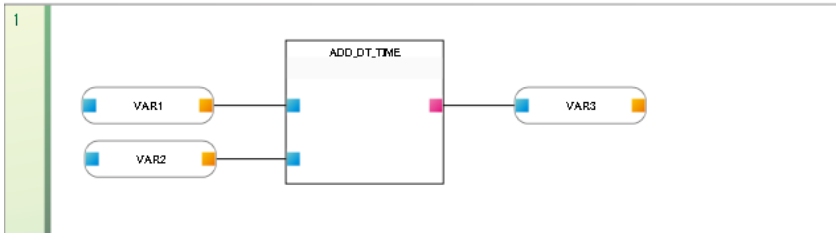
	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
Input value2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*2	-	-	-
Output value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*1 Possible only with DT type.
*2 Possible only with TIME type.

- Usage example
  The DT type VAR1 is added to the TIME type VAR2, and the result is assigned to the DT type output value VAR3.
  For example, whenT#01:11:11 is added to DT#2017-10-19-13:43:30, then DT#2017-10-19-14:54:41 is assigned to the output value.

Parameter | External

Scope	Variable name	Data type	Element number	Default value	Comment
VAR	VAR1	DT	0	DT#2017-10-19-13:43:30	
VAR	VAR2	TIME	0	T#001:11:11	
VAR	VAR3	DT	0	DT#1970-01-01-00:00:00	



- NC(BUILT IN PLC) execution error
  When the operation result exceeds the DT type range. (Error code 14)

SUB_TIME

- Function

Function name	Notation	Description
SUB_TIME		Subtracts time values. A value (input value 2) is subtracted from another value (input value 1), and that result is output to the output value.

(NOTE) If the operation result exceeds the TIME type range, an error is triggered.

- Description of parameter

Input value 1	Subtraction value (minuend) (TIME type)
Input value 2	Subtraction value (subtrahend) (TIME type)
Output value	Subtraction result (TIME type)

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
Input value2	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
Output value	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-

- Usable OM / Constant

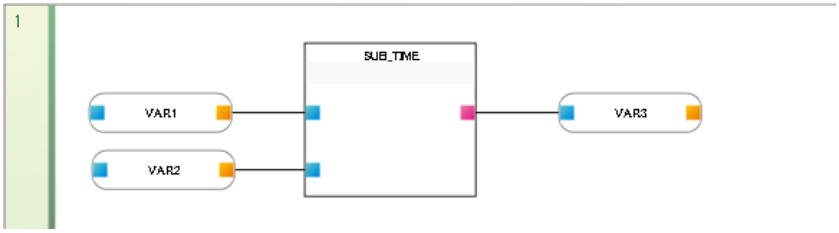
	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
Input value2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
Output value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*1 Possible only with TIME type.

- Usage example
  The TIME type VAR2 is subtracted from the TIME type VAR1, and the operation result is assigned to VAR3.
  For example, when T#001:23:46 is subtracted from T#023:16:54, then T#021:53:08 is assigned to the output value.

Parameter | External

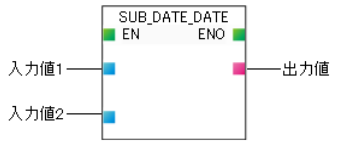
Scope	Variable name	Data type	Element number	Default value	Comment
VAR	VAR1	TIME	0	T#023:16:54	
VAR	VAR2	TIME	0	T#001:23:46	
VAR	VAR3	TIME	0	T#000:00:00	



- NC(BUILT IN PLC) execution error
  When the operation result exceeds the TIME type range. (Error code 14)

SUB_DATE_DATE

- Function

Function name	Notation	Description
SUB_DATE_DATE		Subtracts date values. A value (input value 2) is subtracted from another value (input value 1), and that result is output to the output value.

(NOTE) If the operation result exceeds the TIME type range, an error is triggered.

- Description of parameter

Input value1	Subtraction value (minuend) (DATE type)
Input value2	Subtraction value (subtrahend) (DATE type)
Output value	Subtraction result (TIME type)

- Usable data type

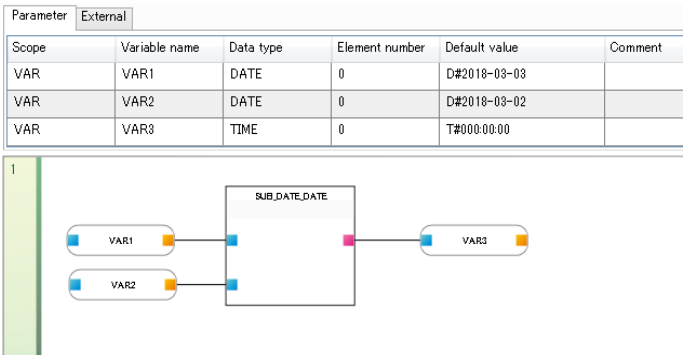
	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
Input value2	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
Output value	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-

- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
Input value2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
Output value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*1 Possible only with DATE type.

- Usage example
  The DATE type VAR2 is subtracted from the DATE type VAR1, and the operation result is assigned to VAR3.
  Ex.: When D#2018-03-02 is subtracted from D#2018-03-03, then T#024:00:00 is assigned to the output value.



- NC(BUILT IN PLC) execution error
  When the operation result exceeds the TIME type range. (Error code 14)

SUB_TOD_TIME

- Function

Function name	Notation	Description
SUB_TOD_TIME		Subtracts time from time of day. A value (input value 2) is subtracted from another value (input value 1), and that result is output to the output value.

(NOTE) If the operation result exceeds the TOD type range, an error is triggered.

- Description of parameter

Input value1	Subtraction value (minuend) (TOD type)
Input value2	Subtraction value (subtrahend) (TIME type)
Output value	Subtraction result (difference) (TOD type)

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
Input value2	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
Output value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-

- Usable OM / Constant

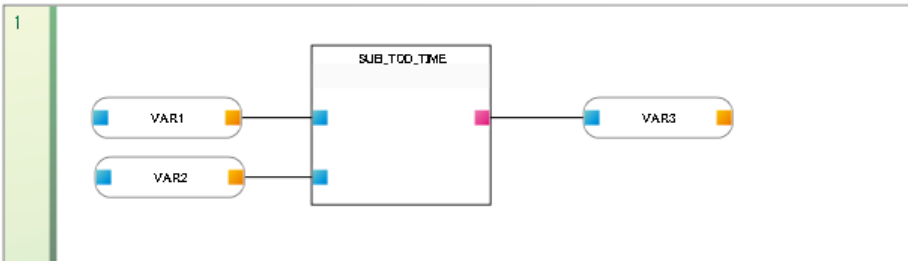
	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
Input value2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*2	-	-	-
Output value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*1 Possible only with TOD type.
*2 Possible only with TIME type.

- Usage example
  The TIME type variable VAR2 is subtracted from the TOD type variable VAR1, and the operation result is assigned to VAR3.
  For example, when the T#003:21:11 is subtracted from the time of day TOD#12:34:56, then TOD#09:13:45 is assigned.

Parameter | External

Scope	Variable name	Data type	Element number	Default value	Comment
VAR	VAR1	TOD	0	TOD#12:34:56	
VAR	VAR2	TIME	0	T#003:21:11	
VAR	VAR3	TOD	0	TOD#00:00:00	



- NC(BUILT IN PLC) execution error
  When the operation result exceeds the TOD type range. (Error code 14)

SUB_TOD_TOD

- Function

Function name	Notation	Description
SUB_TOD_TOD		Subtracts time of day values. A value (input value 2) is subtracted from another value (input value 1), and that result is output to the output value.

(NOTE) If the operation result exceeds the TIME type range, an error is triggered.

- Description of parameter

Input value1	Subtraction value (minuend) (TOD type)
Input value2	Subtraction value (subtrahend) (TOD type)
Output value	Subtraction result (TIME type)

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
Input value2	-	-	-	-	-	-	-	-	-	-	-	-		-	○	-	-
Output value	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-

- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
Input value2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
Output value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

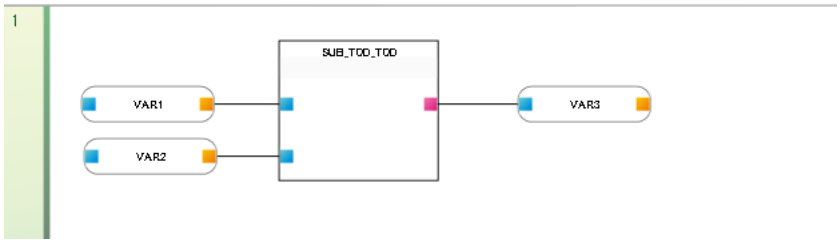
*1 Possible only with TOD type.

- Usage example

The TOD type variable VAR2 is subtracted from the TOD type variable VAR1, and the operation result is assigned to the TIME type variable VAR3.
For example, when TOD#11:33:44 is subtracted from the time TOD#12:34:56, then T#01:01:12 is assigned to the output value.

Parameter | External

Scope	Variable name	Data type	Element number	Default value	Comment
VAR	VAR1	TOD	0	TOD#12:34:56	
VAR	VAR2	TOD	0	TOD#11:33:44	
VAR	VAR3	TIME	0	T#000:00:00	



- NC(BUILT IN PLC) execution error

When the operation result exceeds the TIME type range. (Error code 14)

SUB_DT_TIME

- Function

Function name	Notation	Description
SUB_DT_TIME		Subtracts time from date and time. A value (input value 2) is subtracted from another value (input value 1), and that result is output to the output value.

(NOTE 1) If the operation result exceeds the DT type range, an error is triggered.
(NOTE 2) Leap year is also taken into account in the operation.

- Description of parameter

Input value1	Subtraction value (minuend) (DT type)
Input value2	Subtraction value (subtrahend) (TIME type)
Output value	Subtraction result (difference) (DT type)

- Usable data type

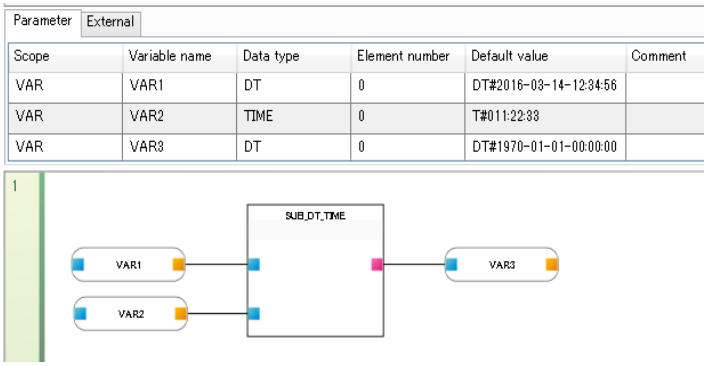
	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-
Input value2	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
Output value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-

- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
Input value2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*2	-	-	-
Output value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*1 Possible only with DT type.
*2 Possible only with TIME type.

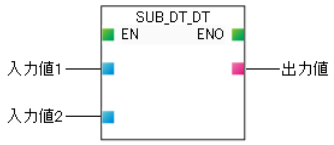
- Usage example
  The TIME type VAR2 is subtracted from the DT type variable VAR1, and the operation result is assigned to DT type VAR3.
  For example, when T#011:22:33 is subtracted from DT#2016-03-14-12:34:56, then DT#2016-03-14-01:12:23 is assigned to the output value.



- NC(BUILT IN PLC) execution error
  When the operation result exceeds the DT type range. (Error code 14)

SUB_DT_DT

- Function

Function name	Notation	Description
SUB_DT_DT		Subtracts the date and time values. A value (input value 2) is subtracted from another value (input value 1), and that result is output to the output value.

(NOTE 1) If the operation result exceeds the TIME type range, an error is triggered.
(NOTE 2) Leap year is also taken into account in the operation.

- Description of parameter

Input value1	Subtraction value (minuend) (DT type)
Input value2	Subtraction value (subtrahend) (DT type)
Output value	Subtraction result (TIME type)

- Usable data type

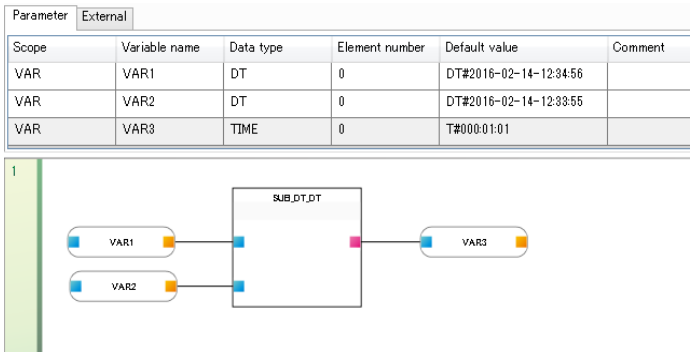
	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-
Input value2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-
Output value	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-

- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
Input value2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
Output value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*1 Possible only with DT type.

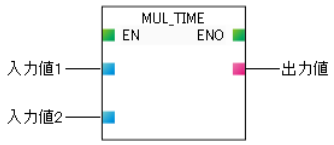
- Usage example
  The DT type variable VAR1 is subtracted from DT type variable VAR2, and the operation result is assigned to VAR3.
  For example, when DT#2016-02-14-12:33:55 is subtracted from DT#2016-02-14-12:34:56, then T#000:01:01 is assigned to the output value.



- NC(BUILT IN PLC) execution error
  When the operation result exceeds the TIME type range. (Error code 14)

MUL_TIME

- Function

Function name	Notation	Description
MUL_TIME		Multiplies the time (input values).

(NOTE 1) If the operation result exceeds the TIME type range, an error is triggered.
(NOTE 2) When a negative number is specified as a multiplication value, the value is converted to an unsigned 64-bit integer and then used for multiplication.

- Description of parameter

Input value1	Time value used in multiplication (TIME type)
Input value2	Multiplication value
Output value	Value after multiplication (TIME type)

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
Input value2	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Output value	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-

- Usable OM / Constant

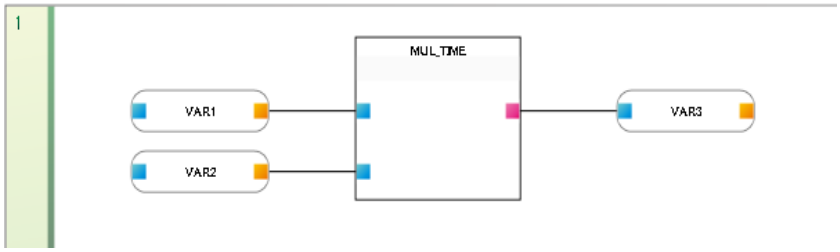
	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
Input value2	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Output value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*1 Possible only with TIME type.

- Usage example
  The TIME type variables VAR2 and VAR1 are multiplied, and the operation result is assigned to the TIME type variable VAR3.
  Ex When T#009:03:00 and 4 are multiplied, then T#36:12:00 is assigned to the output value.

Parameter | External

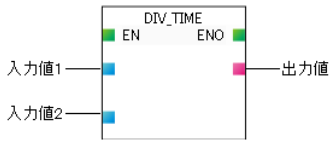
Scope	Variable name	Data type	Element number	Default value	Comment
VAR	VAR1	TIME	0	T#009:03:00	
VAR	VAR2	INT	0	4	
VAR	VAR3	TIME	0	T#000:00:00	



- NC(BUILT IN PLC) execution error
  When the operation result exceeds the TIME type range. (Error code 14)

DIV_TIME

- Function

Function name	Notation	Description
DIV_TIME		Divides time values. A value (input value 1) is divided by another value (input value 2), and that result is output to the output value.

(NOTE 1) The "Remainder" is not included in the operation result. If there is a remainder, then it is rounded off.

(NOTE 2) When a negative number is specified as a division value, the value is converted to an unsigned 64-bit integer and then used for division.

- Description of parameter

Input value1	Time value for division (TIME type)
Input value2	Division value (divisor)
Output value	Time value after division (TIME type)

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
Input value2	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Output value	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-

- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
Input value2	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Output value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*1 Possible only with TIME type.

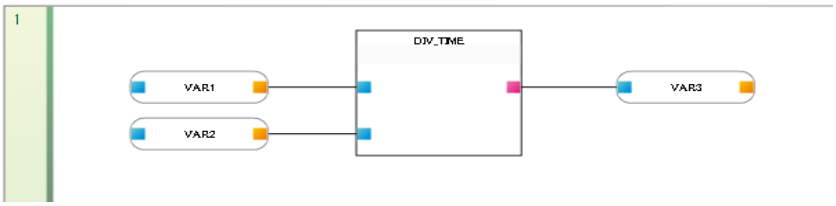
- Usage example

The TIME type variable VAR1 is divided by the variable VAR2, and the operation result is assigned to VAR3.

Ex When the time T#100:50:55 is divided by the numerical value 5, then T#20:10:11 is assigned to the output value.

Parameter | External

Scope	Variable name	Data type	Element number	Default value	Comment
VAR	VAR1	TIME	0	T#100:50:55	
VAR	VAR2	INT	0	5	
VAR	VAR3	TIME	0	T#000:00:00	

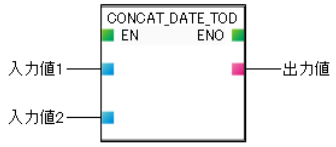


- NC(BUILT IN PLC) execution error

When the divisor is 0 in the division command. (Error code 11)

CONCAT_DATE_TOD

- Function

Function name	Notation	Description
CONCAT_DATE_TOD		Joins or concatenates the date (input value 1) and time of day (input value 2).

- Description of parameter

Input value1	Date
Input value2	Time of day
Output value	Concatenated date and time (western calendar)

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
Input value2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
Output value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-

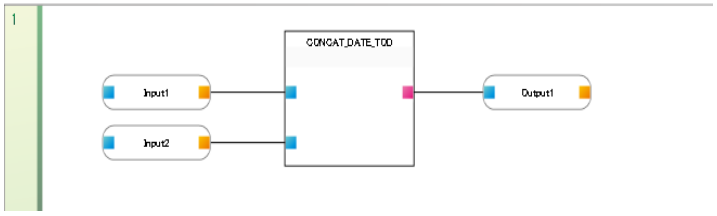
- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Input value2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Output value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

- Usage example
  The DATE type variable Input1 and the TOD type variable Input2 are concatenated, and the result is assigned to the DT type variable Output1.
  For example, when D#2018-03-01 is assigned to Input1 and TOD#01:02:03 is assigned to Input2, then DT#2018-03-01-01:02:03 is assigned to Output1.

Parameter | External

Scope	Variable name	Data type	Element number	Default value	Comment
VAR	Input1	DATE	0	D#2018-03-01	
VAR	Input2	TOD	0	TOD#01:02:03	
VAR	Output1	DT	0	DT#1970-01-01-00:00:00	

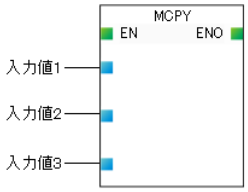


- NC(BUILT IN PLC) execution error
  None

12.2.2.11 Expanded function

MCPY

- Function

Function name	Notation	Description
MCPY		Transfers only the specified number of word points, from the specified variable / OM to another variable / OM.

(NOTE 1) When input value 1 (source) is a STRING type variable, ASCII code is stored in the destination OM.

(NOTE 2) When the transfer value exceeds the valid range of the OM, a numerical value that can be stored in the OM is assigned to the OM.

(NOTE 3) Set a positive integer to input value 3 (transfer points). If the setting is invalid, an error is triggered at conversion.
If the number of points exceeds the range of the OM specified for input value 1 (source) and input value 2 (destination) and the number of points is set to input value 3 (transfer points), then an error is triggered at conversion.

- Description of parameter

Input value1	Variable / OM for source
Input value2	Variable / OM for destination
Input value3	Word points to transfer

When the variable data is transferred to an OM, specify a number of word points for “In2” that matches the data size of the variable.

Data type	Points specified for In2
WORD, INT, UINT	1
DWORD, DINT, UDINT, REAL	2
LINT, ULINT, LREAL, TIME, DATE, TOD, DT	4
STRING	16

When the data size of the variable does not match the “In2” value, the variable data cannot transfer to the OM properly.
For example, when a DINT type variable is transferred to D0 and 1 is specified for “In2” (transfer points), then only 1 low-order word for the DINT type variable is transferred to D0.
When 3 is specified for “In2” (transfer points), the DINT type variable data is transferred to D0 and D1, but other variable data is stored to D2.

- Usable parameter

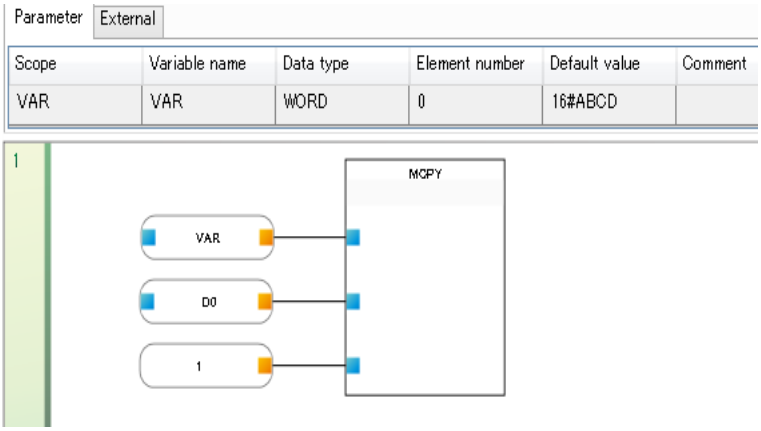
	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1	-	-	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Input value2	-	-	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Input value3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1	○*1	○*1	○*1	○*1	○*1	○*1	○*1	-	-	○*2	○*2	○*2	○*2	-	-	○*2	○*2	-	-	-	-	○*1*2	○
Input value2	-	-	-	-	-	-	-	-	-	○	○	○	○	-	-	○	○	-	-	-	-	-	○
Input value3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-

*1 Possible only when 16-bit digits are specified.
*2 Longwords cannot be specified for the word OM.

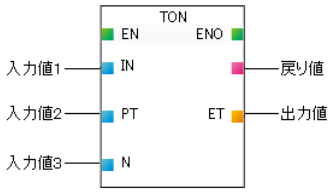
- Usage example
  The value for the variable VAR is 1 word and is transferred to D0.



- NC(BUILT IN PLC) execution error
  When an attempt is made to access an out of range OM. (Error code 23)

TON

- Function

Function name	Notation	Description
TON		Counts up in the timer OM for the number specified in DATA3 when DATA1 is true. After counting up to the time specified in DATA2 (set value), the return value becomes true. When DATA1 is false, the count (time) is reset and the return value becomes false.

(NOTE 1) TON uses the same range as LT in the LD language. Therefore, make sure that the number of the timer OM specified in DATA3 does not exceed the setting range of the timer OM.
If an out of range value is set, the time count is not carried out.

(NOTE 2) When the timer setting specified in DATA2 is a 0, a negative number or greater than 32768, then the count is not performed for the current timer value and the return value will not become true.

- Description of parameter

Input value1	Input. Current timer value counts up when it is true. Current timer value resets when it is false.
Input value2	Timer set value (only integers are possible, units of 10 ms). Set a value between 1 and 32767.
Input value3	Timer number (0 to 511)
Output value	Current timer value
Return value	True when timer counts up. False in all other situations.

- Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input value1	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Input value2	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Input value3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Output value	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Return value	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-

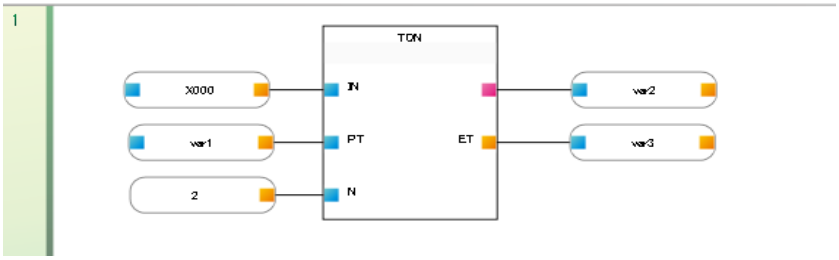
- Usable OM / Constant

	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
Input value1	○	○	○	○	○	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
Input value2	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Input value3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
Output value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○
Return value	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

- Usage example
  When X000 is true, the No.2 timer OM counts up. The current timer value is stored in var3. After counting up to the time specified in var1, var2 becomes true. When X000 is false, the count (var3) is reset and var2 becomes false.

Parameter | External

Scope	Variable name	Data type	Element number	Default value	Comment
VAR	var1	INT	0	100	
VAR	var2	BOOL	0	FALSE	
VAR	var3	INT	0	0	



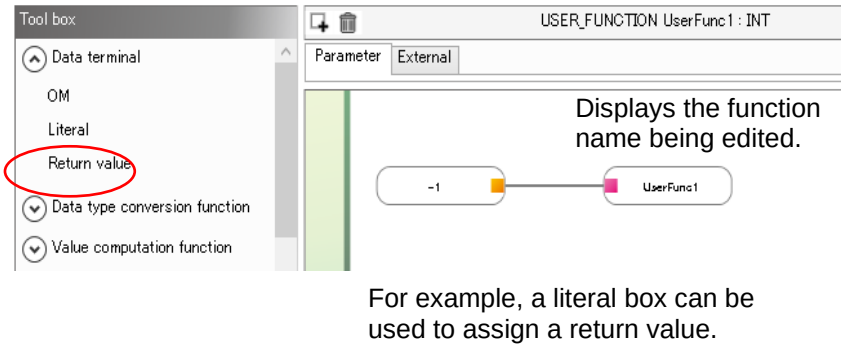
- NC(BUILT IN PLC) execution error
  None

12.2.3 Create & Use User Defined Function / System Defined Function

The basic method for creating and using user and system defined functions in the FBD language is the same as when using the ST language. Therefore, the description here covers the FBD Programming Editor operations only when using the FBD language. Refer to “12.2.3 Create & Use user defined function / system defined function” for further details.

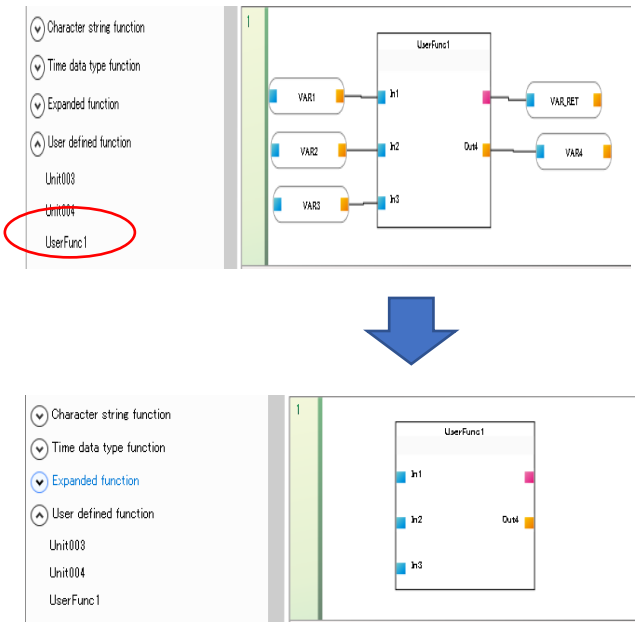
12.2.3.1 Return value box

The return value for the user or system defined function is specified in the return value box. Drag the <Return value> (found in the <Data terminal> under the <Tool box> in the FBD Programming Editor) and drop it into the program area.



12.2.3.2 User defined function and system defined function

The user defined functions or system defined functions that are created are listed under <User defined function> or <System defined function> in the <Tool box> (FBD Programming Editor). The user defined functions or system defined functions can be dragged and dropped into the program area to use them.



(NOTE) After double-clicking on a function box (where a user defined function or system defined function is set), the PLC Programming software D00 LD/ST Programming Editor or the FBD Programming Editor launches. When the POU being used as the user or system defined function automatically opens up, the user can check the settings or change the program.
However, if the following change is made for that POU, the function box turns red.

- Change data type of return value
- Add or delete variables in the scope:
  VAR_INPUT/VAR_OUTPUT/VAR_IN_OUT
- Change data type or element number for the variable in the scope:
  VAR_INPUT/VAR_OUTPUT/VAR_IN_OUT

Conversion is not possible when there is a red box.
If there is a red box, perform one of the following operations.

- Delete the red box and correct the program.
- Re-assign the red box.

When a box is re-assigned, the settings for the variable, user defined function or system defined function with the same name as the variable name or POU name in the current box are reflected in that re-assigned box. Refer to “9.3.1.11 Re-assign box” for details on re-assigning a box.

12.3 Error Message List During FBD Program Conversion

The following shows a list of error messages during FBD program conversion.

Code	Error message	Cause	Countermeasure
E0001	Data size used in the BOOL type parameter variables exceeds the limit: 1024 Data size used in the non-BOOL parameter variables exceeds the limit: 1024	The number of parameter variables registered to the POU exceeds the allowable number that can be registered (BOOL type: 1024, Non-BOOL type: 1024).	Correct the program so that the number of parameter variables used does not exceed the maximum. (There is no need to decrease the number of OMs used in the program.)
	Data size used in the BOOL type external variables exceeds the limit: 1024 Data size used in the non-BOOL external variables exceeds the limit: 1024	The number of external variables registered to the POU exceeds the allowable number that can be registered (BOOL type: 1024, Non-BOOL type: 1024).	Correct the program so that the number of external variables used does not exceed the maximum. (There is no need to decrease the number of OMs used in the program.)
	Number of OMs used for BOOL type parameter variables and return values in functions exceeds the limit: 4096. Number of OMs used for non-BOOL parameter variables and return values in functions exceeds the limit: 4096.	The parameter variables and return values registered to the user defined functions exceed the allowable number that can be registered (BOOL type: 4096, Non-BOOL type: 4096).	Correct the program so that the number of parameter variables used does not exceed the maximum. (There is no need to decrease the number of OMs used in the program.)
E0002	There is an error in the <User defined function name>.	There is an error in the user defined function called by the POU.	Check the error description for the user defined function with the error, and correct the error location.
E0004	<Character string> is defined in both the variable name and function name.	The same name is defined in both the variable name and function name.	Change one of the names to avoid duplicating a name.
E3009	Number of characters after the ST language conversion exceeds the upper limit: 16383 characters.	When an FBD program is converted, it is automatically converted and creates an ST program. However, the number of characters in the ST program has exceeded the limit of 16383 characters.	Split up the FBD program into multiple POUs, for example, and reduce the program size of one POU.
E7001	Not compatible with this data type.	Incompatible data types are being used in an assignment process or value comparison.	Use compatible data types when assigning or comparing data.
E7002	"%s" is not defined.	An unregistered variable is being used in the FBD program. (For example, a variable being used in the FBD program was deleted in the variable editing area.)	Register that variable being used in the FBD program, or use another variable.
E7007	External variable "%s" is defined as a global variable.	The global variable that the external variable references is not registered.	Re-set the global variable that the external variable references.
E7008	No. %d argument type for "%s" is invalid.	The variable or OM is a data type that cannot be used in a function argument.	Specify a variable or OM type that can be used as an argument.
	Bit OM cannot be used for the No. %d argument for "%s".	An OM that cannot be used was specified for a function argument.	
	No. %d argument type for "%s" does not match the other argument.	The function argument types do not match.	

Code	Error message	Cause	Countermeasure
E7015	Subscript for the array variable is invalid.	The character string that is set cannot be used as the subscript for specifying the element of the array variable. Or, the input field is blank.	Set one of the following as the subscript for specifying the element of the array variable. • Constant or variable that has one of the following data types: BOOL, BYTE, WORD, DWORD, INT, DINT, LINT, UINT, UDINT or ULINT • OM
E7016	Input step number exceeds the maximum value: %d steps.	The number of input steps exceeds the maximum number of steps (5120 steps when using a high speed control program. 40960 steps when using a standard control program and function).	Correct the program so that the maximum number of steps is not exceeded.
E7033	Number of arguments for "%s" is invalid.	An invalid FBD program has been created. (For example, an input is not connected to a function box.)	Check the function box with the error, and correct it.
E7038	Variable %s is not used in the program.	An invalid FBD program has been created. (For example, an input is not connected to a function box.)	Check the function box with the error, and correct it.
E7042	If %d words are forwarded from %s, the word number %d for %s will be exceeded. Data across multiple OMs cannot be copied and forwarded.	There is a setting for transferring data with a data size that exceeds the maximum OM (destination) number for the MCPY.	Correct the OM (destination) number so it does not exceed the maximum OM number, or correct the transfer data size.
E7045	No. %d argument value for "%s" is invalid.	The range of the valid setting values for the function argument has been exceeded.	Check the function specification and re-set the argument correctly.
W0001	There is a warning in the <User defined function name> or <System defined function name>.	There is a warning in the user defined or system defined function called by the POU.	Check the warning description for the user defined or system defined function with the warning, and correct when necessary.
W0002	Variable %s is not used in the program.	The registered variable is not used in the program.	When necessary, delete variables that are not needed.
W0003	Definition for %s has changed. Delete the box and create it again.	After a box for a variable, system defined function or user defined function was set up in the program area, that variable, user defined function or system defined function was changed.	After setting up a box in the program area and then changing the corresponding variable, user defined function or system defined function, that box turns red. Delete the red box and correct the program. Or, re-assign the box. Refer to “9.3.1.11 Re-assign box ” for details on how to re-assign a box.

(This page was intentionally left blank.)

(This page was intentionally left blank.)

©2021 BROTHER INDUSTRIES, LTD.
All Rights Reserved.
Printed in Japan.
This is the original instructions.

69A809001
2109 (3)