brother

W1000Xd1 Series / S300Xd1 Series / S500Xd1 Series
S700Xd1 Series / R450Xd1 Series / R650Xd1 Series
U500Xd1 Series

COMPACT MACHINING CENTER SPEEDIO

CNC-D00
PLC SYSTEM MANUAL
(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.

69A804001(4)

PLC SYSTEM MANUAL

Read this manual before performing work.

brother

Introduction

Thank you for purchasing the SPEEDIO (hereafter referred to as “machine”) made by Brother. Always be sure to read this manual and the machine manual carefully first, in order to use the machine functions properly and safely.

Brother is not responsible or liable for accidents that occur during special machine use or handling that does not follow the general safety usage guidelines.

This machine manual is divided into the following sections.

- Operation Manual
  This manual describes the operation procedure for the machine.
- Installation Manual
  This manual describes the machine’s installation procedure and inspections.
- Programming Manual
  This manual provides a program description.

Keep this manual for future reference.
Attach this manual to the machine if it is resold.
Contact the nearest Brother sales office or Brother approved service dealer if this manual or the safety labels are damaged, lost or missing. (Charges apply)

The re-exporting and resale of this machine is regulated by Japan’s export laws and regulations in accordance with international export management.
When exporting, permission from the exporting country’s government and/or from the Japanese government may be required.
Contact a Brother Industries dealer in advance before re-transferring, reselling or re-exporting this machine.

Copying and reprinting all or part of the content in this manual without permission is illegal.
The content of this manual may be changed without prior notice.
Brother has taken steps to ensure this manual is accurate and complete. However, if you notice or suspect that there is an error, please contact the nearest Brother sales office or Brother approved service dealer.
Ethernet is a registered trademark of Xerox Corp. in the U.S.A.
Any trade names and product names of companies appearing on Brother products, related documents and any other materials are all trademarks or registered trademarks of those respective companies.

How to Read This Manual

This manual is divided into the following sections.

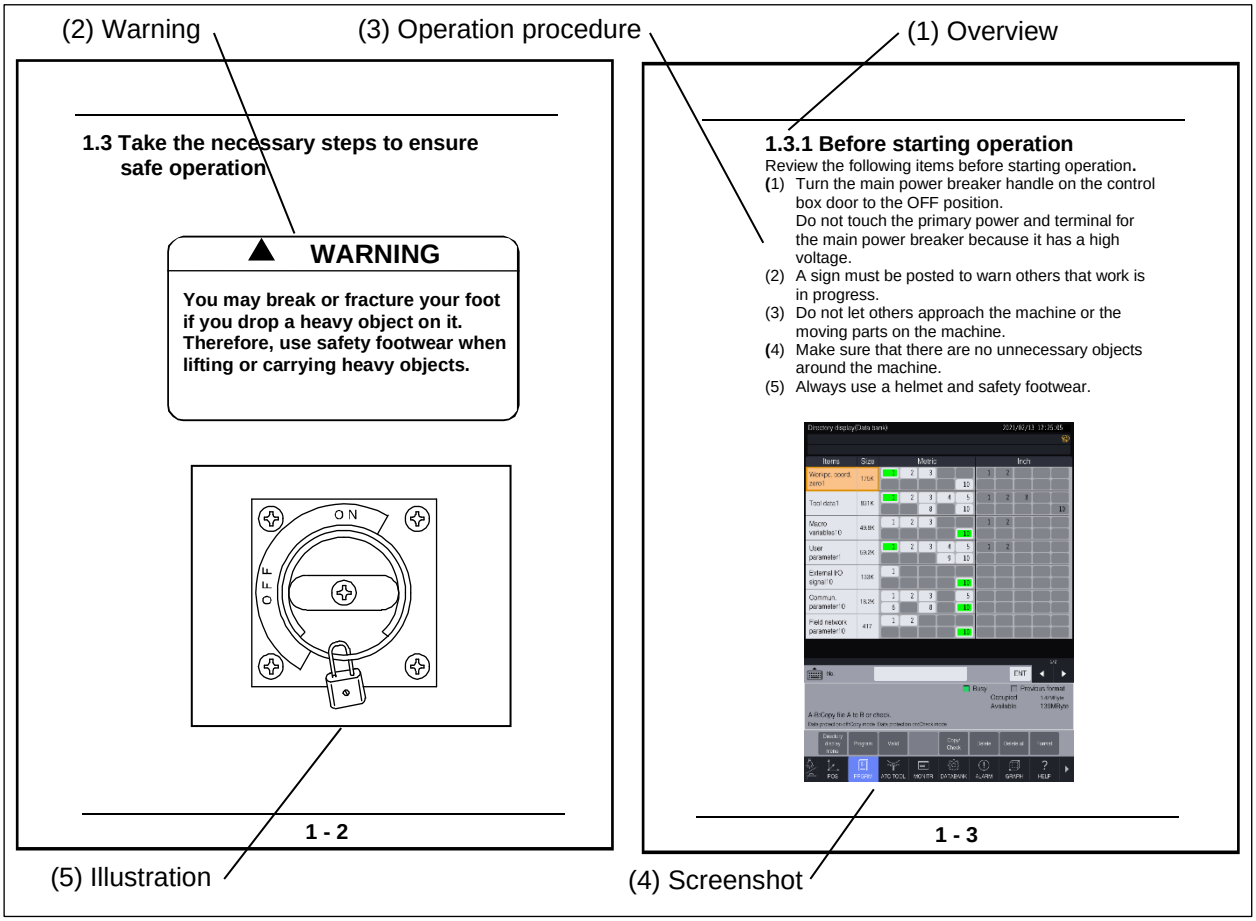
(1) Overview ------ A summary of the content is provided for the corresponding section.

(2) Warning-------- A warning is provided for any hazards that could potentially cause serious bodily injury, death or damage to the machine.
The hazards are described in the following order.
- (2-1) Hazard level
- (2-2) Type of hazard
- (2-3) Potential damage
- (2-4) Safety directions to avoid danger

(3) Operation procedure --- The procedure describes how to operate each function.

(4) Screenshot ---A screenshot is inserted into places to highlight certain points in the operation procedure. The screenshot messages are shown at an approximate position and may differ slightly from the actual position of the line or column. The same applies to the font.

(5) Illustration ---- Illustrations, such as explanatory drawings, diagrams that show the dimensions, positioning, ranges, figures or configurations, are used in certain places where a written explanation alone may be hard to understand.



In this manual, the following symbols are used to differentiate between keys, switches, text displayed on screens and alarm messages.

[ ]	: Keys
[ ]	: Switches
< >	: Text displayed on screens
<< >>	: Alarm messages

(This page was intentionally left blank.)

CHAPTER 1 OVERVIEW ---------- 1-1
1.1 System Overview ---------- 1-2
1.1.1 Project ---------- 1-2
1.1.2 D00 NC Unit ---------- 1-2
1.1.3 PLC Programming Software ---------- 1-2
1.2 Project Overview ---------- 1-3
1.2.1 POU ---------- 1-3
1.2.2 OM (Object Memory) ---------- 1-3
1.3 Overview of Built-in PLC Functions ---------- 1-5
1.3.1 Task ---------- 1-5
1.3.2 System Ladder ---------- 1-6
1.3.3 Standard Control Program 1 ---------- 1-6
1.3.4 Overall View of Built-in PLC Functions ---------- 1-7
1.4 Overview of Programming Functions ---------- 1-8
1.4.1 LD (Ladder) Language ---------- 1-8
1.4.2 ST (Structured Text) Language ---------- 1-8
1.4.3 FBD (Function Block Diagram) Language ---------- 1-8
1.4.4 Programming Language Restrictions ---------- 1-8
1.5 External Device Interface Function ---------- 1-9
1.6 Input/Output Between Projects ---------- 1-9

CHAPTER 2 PROCEDURE FOR USING THE PLC FUNCTION ---------- 2-1
2.1 Procedure for Using the PLC Function ---------- 2-2

CHAPTER 3 BUILT-IN PLC ---------- 3-1
3.1 Built-in PLC Operation ---------- 3-2
3.1.1 At Power ON ---------- 3-2
3.1.2 At Reset ---------- 3-2
3.1.3 Task Operation Check ---------- 3-2
3.2 Time Management for Built-in PLC ---------- 3-3
3.2.1 Task Execution Chart ---------- 3-3
3.2.2 OM Synchronization ---------- 3-3
3.2.3 Watch Dog ---------- 3-5
3.2.3.1 PLC Watch Dog Timeout ---------- 3-5
3.2.3.2 Monitoring in RUN Status ---------- 3-5
3.2.4 Timer ---------- 3-5
3.3 PLC Program Format ---------- 3-6
3.3.1 Operation Procedure ---------- 3-6

CHAPTER 4 OM ---------- 4-1
4.1 OM List ---------- 4-2
4.1.1 Global OM ---------- 4-2
4.1.2 Local OM ---------- 4-2
4.2 Global OM Description ---------- 4-3
4.2.1 External Input Relay (External→PLC, X000 to X7FF) ---------- 4-3
4.2.2 External Output Relay (PLC→External, Y000 to Y7FF) ---------- 4-3
4.2.3 Internal Input (NC→PLC, BX000 to BX7FF) ---------- 4-4
4.2.3.1 Overview ---------- 4-4
4.2.3.2 Signal List ---------- 4-4
4.2.4 Internal Output (PLC→NC, BY000 to BY7FF) ---------- 4-35
4.2.4.1 Overview ---------- 4-35
4.2.4.2 Signal List ---------- 4-35
4.2.5 Internal Input Data (NC→PLC, BDX0 to BDX1023) ---------- 4-61
4.2.5.1 Overview ---------- 4-61
4.2.5.2 Signal List ---------- 4-61
4.2.6 Internal Output Data (PLC→NC, BDY0 to BDY1023) ---------- 4-71
4.2.6.1 Overview ---------- 4-71

4.2.6.2 Data List ---------- 4-71
4.2.7 Internal Relay (M0 to M10239) ---------- 4-77
4.2.7.1 Reserved Internal Relay ---------- 4-77
4.2.8 Data Register (D0 to D8191) ---------- 4-77
4.2.8.1 Reserved Data Register ---------- 4-77
4.3 Local OM Description ---------- 4-78
4.3.1 Local Relay (LM0 to LM4095) ---------- 4-78
4.3.1.1 Reserved Local Relay ---------- 4-78
4.3.2 Local Data Register (LD0 to LD8191) ---------- 4-78
4.3.2.1 Reserved Local Data Register ---------- 4-78
4.3.3 Local Timer (LT0 to LT511) ---------- 4-78
4.3.3.1 Reserved Local Timer ---------- 4-78
4.3.4 Local Counter (LC0 to LC511) ---------- 4-79
4.3.4.1 Reserved Local Counter ---------- 4-79
4.3.5 Special Relay (SM0 to SM255) ---------- 4-79
4.3.5.1 List ---------- 4-79
4.3.5.2 Special Relay Description ---------- 4-80
4.3.6 Special Register (SD0 to SD255) ---------- 4-86
4.3.6.1 List ---------- 4-86
4.3.6.2 Special Register Description ---------- 4-87
4.3.7 Index Register (IX0 to IX7) ---------- 4-90
4.3.8 Label (L0 to L1023) ---------- 4-90
4.4 PLC Data Input ---------- 4-91
4.4.1 Workpiece Coordinate Zero Writing ---------- 4-91
4.4.2 Character String ---------- 4-91
4.5 PLC Data Output ---------- 4-93
4.5.1 Data list ---------- 4-93
4.5.2 PLC output No.0 (None) ---------- 4-94
4.5.3 PLC output No.1 (Current alarm/operator message screen 1) ---------- 4-94
4.5.4 PLC output No.2 (Current alarm/operator message screen 2) ---------- 4-95
4.5.5 PLC output No.3 (Current alarm/operator message screen 3) ---------- 4-96
4.5.6 PLC Output No.2001 (Production monitor － Time display screen 1) ---------- 4-96
4.5.7 Output Data No.201x (Production Monitor - Workpiece Counter Screen) ---------- 4-97
4.5.8 Output Data No.202x
(Production Monitor - Automatic Thermal Distortion Compensation Screen) ---------- 4-97
4.5.9 PLC Output No.4xx1 (Input/Output Servo (*-axis) Screen 1) ---------- 4-98
4.5.10 PLC Output No.4xx2 (Input/Output Servo (*-axis) Screen 2) ---------- 4-99
4.5.11 PLC Output No.4xx3 (Input/Output Servo (*-axis) Screen 3) ---------- 4-100
4.5.12 Output Data No.500x (Workpiece Coordinate Zero Screen) ---------- 4-101
4.5.13 Output Data No.51xx (Workpiece Coordinate Zero (Extended) Screen) ---------- 4-102

CHAPTER 5 I/O INFORMATION ---------- 5-1
5.1 <PLC (I/O Control)> Screen ---------- 5-2
5.2 Layout and Operations on OM Monitor Screen ---------- 5-3
5.2.1 Description of Screen Display ---------- 5-3
5.2.2 Description of Screen Operations ---------- 5-4
5.2.3 Special Notes ---------- 5-4
5.3 Global OM ---------- 5-5
5.4 Local OM ---------- 5-6
5.4.1 Local Timer Screen ---------- 5-7
5.4.2 Local Counter Screen ---------- 5-8
5.5 Registration OM Monitor/Test ---------- 5-9
5.5.1 Description of Screen Display ---------- 5-9
5.5.2 Registered Monitor Setting Screen ---------- 5-9
5.5.3 Registered Monitor Setting Screen ---------- 5-10
5.6 Status Screen ---------- 5-11

CHAPTER 6 PROGRAM ORGANIZATION UNIT ---------- 6-1
6.1 Runtime System Settings ---------- 6-2

6.1.1 Edit POU (High Speed / Standard Programs) ----------6-2
6.1.2 File Operation ----------6-4
6.1.3 Program Edit----------6-4
6.1.4 Program Execution Settings ----------6-5
6.2 Conversion ---------- 6-6
6.2.1 Conversion Mode Settings----------6-6
6.3 Global Variable Settings ---------- 6-8
6.3.1 Available Data Capacity ----------6-9
6.4 Edit POU (Functions) ----------6-10
6.4.1 Edit user defined function---------- 6-10
6.4.2 Edit System Defined Functions ---------- 6-11
6.4.3 Number of Registered Variables ---------- 6-12
6.4.4 File Operations ---------- 6-12
6.5 Check POU ----------6-13
6.5.1 Double Coil Check Result ---------- 6-13
6.5.2 Results from Power Failure Hold Area / Writing Range Check ---------- 6-14
6.5.3 Result from Writing Check of OMs Outside Writing Range---------- 6-15
6.5.4 Check Lines with Errors ---------- 6-16
6.6 Parameter Settings ----------6-17
6.6.1 <PLC (Parameter setting)> Screen ---------- 6-17
6.6.2 Setting Items---------- 6-18

CHAPTER 7 PROGRAMMING FUNCTION LD---------- 7-1
7.1 <PLC (Program edit - LD)> Screen ---------- 7-2
7.1.1 Overview----------7-2
7.1.2 View Mode ----------7-3
7.1.2.1 Screen layout ----------7-3
7.1.2.2 Write program ----------7-4
7.1.2.3 Read program----------7-5
7.1.2.4 Save ----------7-5
7.1.2.5 New program ----------7-5
7.1.2.6 Cancel new program ----------7-5
7.1.2.7 Search----------7-6
7.1.2.8 Comment ----------7-7
7.1.3 Edit Mode----------7-8
7.1.3.1 Screen layout ----------7-8
7.1.3.2 Key assignment----------7-9
7.1.3.3 Symbol input----------7-9
7.1.3.4 Multiple bit access for bit OM---------- 7-10
7.1.3.5 32-bit access for word OM---------- 7-10
7.1.3.6 Insert / Overwrite mode ---------- 7-10
7.1.3.7 Rung operations ---------- 7-11
7.1.3.8 Delete symbol---------- 7-11
7.1.3.9 Undo edit operation ---------- 7-12
7.1.3.10 Convert ---------- 7-12
7.1.3.11 Write during run ---------- 7-12
7.1.3.12 Edit comment ---------- 7-13
7.1.4 Monitor Mode ---------- 7-14
7.1.4.1 Screen layout ---------- 7-14
7.1.4.2 Symbol display ---------- 7-15
7.1.4.3 Set / Reset ---------- 7-16
7.1.4.4 Change current value ---------- 7-17
7.1.4.5 Execute / Stop ---------- 7-17
7.1.4.6 Clear all local OMs ---------- 7-18
7.1.4.7 Clear all global OMs---------- 7-18
7.1.5 Calcbox Edit Mode ---------- 7-18
7.1.5.1 Create new calcbox ---------- 7-19
7.1.5.2 Edit calcbox---------- 7-19
7.1.5.3 Delete calcbox ---------- 7-19

CHAPTER 8 PROGRAMMING FUNCTION ST ---------- 8-1
8.1 <PLC (Program edit - ST)> Screen ---------- 8-2
8.1.1 Overview ---------- 8-2
8.1.2 View Mode ---------- 8-3
8.1.2.1 Screen layout ---------- 8-3
8.1.2.2 Write program ---------- 8-4
8.1.2.3 Read program ---------- 8-5
8.1.2.4 Save ---------- 8-5
8.1.2.5 New program ---------- 8-5
8.1.2.6 Cancel new program ---------- 8-5
8.1.2.7 Find character string ---------- 8-5
8.1.3 Edit Mode ---------- 8-6
8.1.3.1 Screen layout ---------- 8-6
8.1.3.2 Multiple bit access for bit OM ---------- 8-8
8.1.3.3 32-bit access for word OM ---------- 8-8
8.1.3.4 Specify OM indirectly ---------- 8-8
8.1.3.5 Undo edit operation ---------- 8-9
8.1.3.6 Write during RUN ---------- 8-9
8.1.3.7 Conversion ---------- 8-9
8.1.4 Monitor Mode ---------- 8-10
8.1.4.1 Screen layout ---------- 8-10
8.1.4.2 Change Current Value ---------- 8-11
8.1.4.3 Execute / Stop ---------- 8-11
8.1.4.4 Clear all local OMs ---------- 8-11
8.1.4.5 Clear all global OMs ---------- 8-11
8.2 Edit Variables ---------- 8-12
8.2.1 New variable ---------- 8-13
8.2.2 Copy & Paste ---------- 8-14
8.2.3 Delete Variable ---------- 8-14

CHAPTER 9 PROGRAMMING FUNCTION FBD ---------- 9-1
9.1 <PLC (Program edit - FBD)> Screen ---------- 9-2
9.1.1 Overview ---------- 9-2
9.1.2 View mode ---------- 9-3
9.1.2.1 Screen layout ---------- 9-3
9.1.2.2 Write program ---------- 9-5
9.1.2.3 Read program ---------- 9-5
9.1.2.4 Save ---------- 9-5
9.1.2.5 Search ---------- 9-5
9.1.3 Edit Mode ---------- 9-6
9.1.3.1 Screen layout ---------- 9-6
9.1.3.2 Program edit ---------- 9-7
9.1.3.3 Multiple bit access for bit OM ---------- 9-7
9.1.3.4 32-bit access for word OM ---------- 9-7
9.1.3.5 Specify OM indirectly ---------- 9-8
9.1.3.6 Undo edit operation ---------- 9-8
9.1.3.7 Write during RUN ---------- 9-8
9.1.3.8 Convert ---------- 9-8
9.1.4 Monitor mode ---------- 9-9
9.1.4.1 Screen layout ---------- 9-9
9.1.4.2 Change current value ---------- 9-10
9.1.4.3 Execute / Stop ---------- 9-10
9.1.4.4 Clear all local OMs ---------- 9-10
9.1.4.5 Clear all global OMs ---------- 9-11
9.2 Edit Variables ---------- 9-12
9.2.1 New variable ---------- 9-14
9.2.2 Copy / Paste ---------- 9-14
9.2.3 Delete Variable ---------- 9-14

CHAPTER 10 PLC AXIS ---------- 10-1

10.1 Overview----------10-2
10.2 Parameters----------10-3
10.2.1 Machine Parameter---------- 10-3
10.2.2 User Parameter ---------- 10-7
10.2.3 Special Notes When Changing Parameters ---------- 10-8
10.3 Door Interlock Specifications----------10-9
10.3.1 Special Notes When Changing Door Interlock Mode ---------- 10-9
10.3.2 Operation Status in Each Door Interlock Mode ---------- 10-9
10.3.3 Restricted Operation in Each Door Interlock Mode ---------- 10-9
10.3.4 Limits on Operation When Doors are Open ---------- 10-9
10.3.5 Door Lock---------- 10-9
10.4 Screen Display ---------- 10-10
10.5 Manual Operation Mode ---------- 10-11
10.5.1 High-speed Travel----------10-11
10.5.2 Low-speed Travel ----------10-11
10.5.3 Step Travel ----------10-11
10.5.4 Handle Travel Operation ----------10-12
10.5.5 Other Operations ----------10-12
10.5.6 Special Notes ----------10-12
10.6 PLC Axis Control Using Built-in PLC ---------- 10-13
10.6.1 Command Method Between PLC-NC----------10-13
10.6.2 PLC Axis Control Data Register Details----------10-14
10.6.3 Special Notes ----------10-22
10.7 Clamp Mechanism---------- 10-23
10.8 Special Notes ---------- 10-23

CHAPTER 11 COMMANDS (LD) ----------11-1
11.1 LD Language Specifications----------11-2
11.1.1 Commands ---------- 11-2
11.1.2 Summary of Commands---------- 11-2
11.2 LD Basic Commands ----------11-4
11.2.1 List of LD Basic Commands---------- 11-4
11.2.2 LD Basic Command Description---------- 11-6
11.2.2.1 Contact commands ---------- 11-6
11.2.2.2 Connection commands---------- 11-8
11.2.2.3 Output commands ----------11-11
11.2.2.4 Program control commands ----------11-17
11.2.2.5 End commands----------11-20
11.3 LD Application Commands---------- 11-22
11.3.1 List of LD Application Commands ----------11-22
11.3.2 LD Application Command Description ----------11-25
11.3.2.1 Data comparison----------11-25
11.3.2.2 Arithmetic operation ----------11-26
11.3.2.3 BCD-BIN conversion ----------11-40
11.3.2.4 Transfer----------11-42
11.3.2.5 Branch ----------11-47
11.3.2.6 Logic operation ----------11-51
11.3.2.7 Rotation----------11-60
11.3.2.8 Shift ----------11-65
11.3.2.9 Data processing----------11-76
11.3.2.10 FIFO----------11-85
11.3.2.11 Loop ----------11-89
11.3.2.12 Carry flag ----------11-91
11.3.2.13 Date/Time----------11-92
11.3.2.14 Palletize ----------11-93
11.3.2.15 User function ----------11-96
11.4 Error Message List During LD Program Conversion ---------- 11-97

CHAPTER 12 COMMANDS (ST)----------12-1

12.1 ST Language Specifications ---------- 12-2
12.1.1 ST Program Components ---------- 12-2
12.1.2 Usable Characters ---------- 12-2
12.1.2.1 Reserved words and reserved characters ---------- 12-2
12.1.3 Operators ---------- 12-3
12.1.4 Control Statements ---------- 12-4
12.1.5 Functions ---------- 12-4
12.1.6 Variables and OMs ---------- 12-4
12.1.6.1 List of data types ---------- 12-4
12.1.6.2 Usable OMs ---------- 12-5
12.1.6.3 Specifying indexing and digit-specified access for OMs ---------- 12-5
12.1.6.4 External variables ---------- 12-6
12.1.6.5 Array variables ---------- 12-6
12.1.7 Constants ---------- 12-8
12.1.8 Delimiters ---------- 12-9
12.1.9 Comments ---------- 12-9
12.2 Control Statements in ST Language ---------- 12-10
12.2.1 List of Control Statements in ST Language ---------- 12-10
12.2.2 Detailed Descriptions of Control Statements in ST Language ---------- 12-11
12.2.2.1 Assignment ---------- 12-11
12.2.2.2 Function call ---------- 12-12
12.2.2.3 IF ---------- 12-13
12.2.2.4 CASE ---------- 12-14
12.2.2.5 FOR DO ---------- 12-15
12.2.2.6 WHILE DO ---------- 12-16
12.2.2.7 REPEAT UNTIL ---------- 12-17
12.2.2.8 RETURN ---------- 12-18
12.2.2.9 EXIT ---------- 12-18
12.2.2.10 Conditional expressions (BOOL expression) ---------- 12-19
12.3 ST Language Functions ---------- 12-20
12.3.1 List of ST Language Functions ---------- 12-20
12.3.2 Detailed Description of ST Language Functions ---------- 12-22
12.3.2.1 Data type conversion function ---------- 12-23
12.3.2.2 Binary copy function ---------- 12-27
12.3.2.3 Arithmetic operation function ---------- 12-29
12.3.2.4 Numerical value operation function ---------- 12-35
12.3.2.5 Bit shift function ---------- 12-46
12.3.2.6 Bit operation function ---------- 12-52
12.3.2.7 Selection function ---------- 12-56
12.3.2.8 Comparison function ---------- 12-62
12.3.2.9 Character string function ---------- 12-68
12.3.2.10 Time data type function ---------- 12-78
12.3.2.11 Expanded function ---------- 12-90
12.3.3 Create & Use User Defined Function / System Defined Function ---------- 12-93
12.3.3.1 Create POU ---------- 12-93
12.3.3.2 Register variable ---------- 12-94
12.3.3.3 Programming and return value ---------- 12-94
12.3.3.4 How to use user defined function / system defined function ---------- 12-95
12.3.3.5 Restrictions and special notes ---------- 12-95
12.4 Error Message List During ST Program Conversion ---------- 12-96

CHAPTER 1	OVERVIEW	1
CHAPTER 2	PROCEDURE FOR USING THE PLC FUNCTION	2
CHAPTER 3	BUILT-IN PLC	3
CHAPTER 4	OM	4
CHAPTER 5	I/O INFORMATION	5
CHAPTER 6	PROGRAM ORGANIZATION UNIT	6
CHAPTER 7	PROGRAMMING FUNCTION LD	7
CHAPTER 8	PROGRAMMING FUNCTION ST	8
CHAPTER 9	PROGRAMMING FUNCTION FBD	9
CHAPTER 10	PLC AXIS	10
CHAPTER 11	COMMANDS (LD)	11
CHAPTER 12	COMMANDS (ST)	12

(This page was intentionally left blank.)

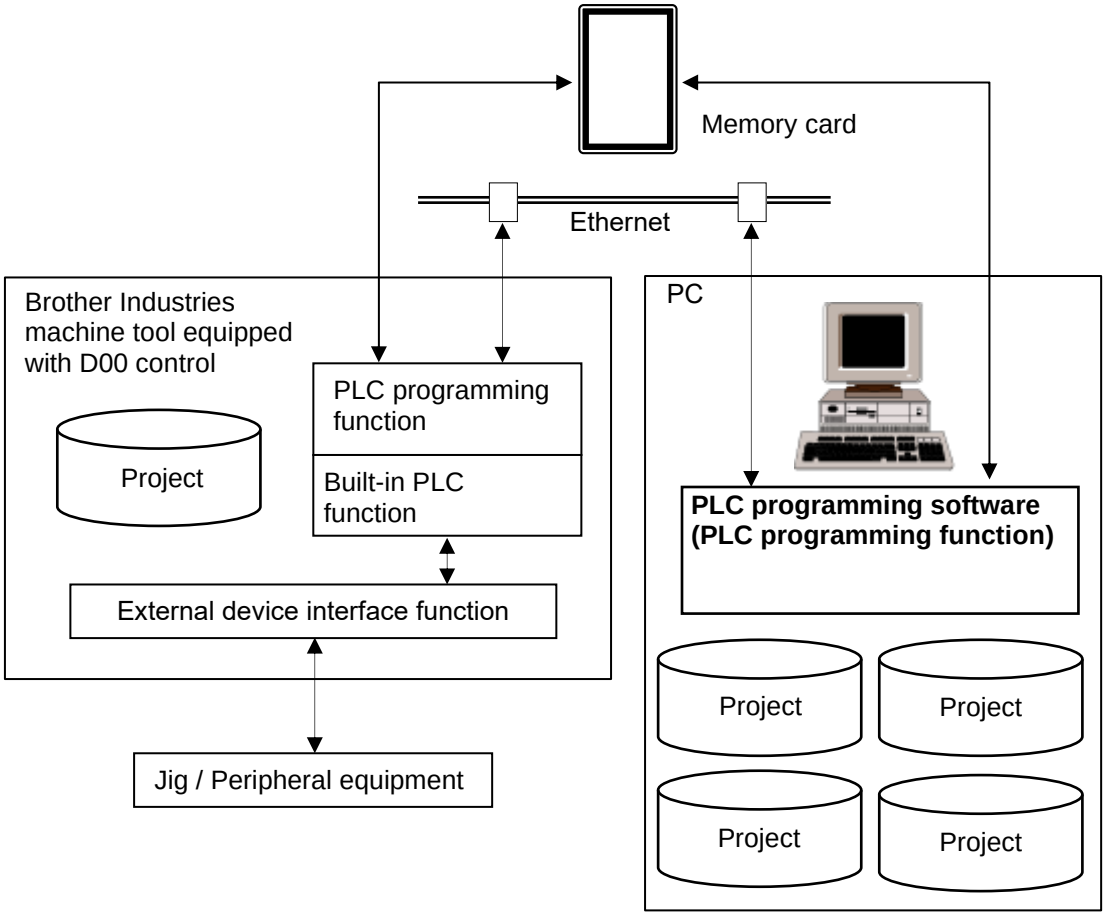
CHAPTER 1

OVERVIEW

1.1 System Overview
1.2 Project Overview
1.3 Overview of Built-in PLC Functions
1.4 Overview of Programming Functions
1.5 External Device Interface Function
1.6 Input/Output Between Projects

1.1 System Overview

The PLC system is built into the numerical control (NC) on the Brother Industries machine tool equipped with D00 control, and it can be used for linking with jig control and peripheral equipment. In addition, as an option, we also provide PLC programming software that can be used on computers.



1.1.1 Project

The overall PLC system saves or holds the data as a unit that is referred to as a project.

1.1.2 D00 NC Unit

The PLC system in the NC holds one project.
In addition, the NC also features a programming function, a built-in PLC function and an external device interface function.

1.1.3 PLC Programming Software

There is a computer application software with a programming function for the PLC system. This software product is provided as an option for users.
Users can store multiple projects and select and then edit a given project.
In addition, the user can connect a PC with the NC via Ethernet and overwrite or monitor the project stored in the NC.
However, the programming function on the machine and the PLC programming software operations (via Ethernet) cannot be performed at the same time.

If the communication parameter (Ethernet/FTP) <Restrict Ethernet access> is set to <1: Yes>, set the user parameter (switch 1: system) <PLC program protection> to <0:Invalid> in order to perform a writing operation or monitoring from the PLC programming software.
Refer to the PLC System Manual (Option) for further details.

1.2 Project Overview

1.2.1 POU

Inside a given project, there are POUs (Program Organization Units), which refer to the program components.
There are two types of the POUs: control programs and functions.

Name	Application
Control program	This POU is used to directly specify an execution. A given project can hold 10 high speed control programs and 10 standard control programs. The user can register control programs to execute tasks (described later) for the built-in PLC.
Function	This unit can be called from a control program. There are two types of functions: system defined function and user defined function. • System defined function : Function can only be viewed or used • User defined function : Function can be created, edited, deleted and used When variables are created in functions, they are reset each time the function is called.

We do not recommend running a mix of control programs or functions between different projects, because it will cause a number of operating conditions to be in conflict.
To change a POU on this machine, set the user parameter (switch 1: system) <PLC program protection> to <0: Invalid>.

1.2.2 OM (Object Memory)

There is an OM (object memory), which refers to the memory used by the POU for computation. Different memory areas are included in that memory, such as external I/O and timers. Refer to “Chapter 4 OM” for further details.

Global OM (OM used commonly for all tasks)

OM name	Address	Points	Description
X Input relay	X000~X7FF	2048	External input relay
	BX000~BX7FF	2048	Internal input relay (NC → PLC)
	BDX000~BDX1023	1024	Internal input data (NC → PLC)
Y Output relay	Y000~Y7FF	2048	External output relay
	BY000~BY7FF	2048	Internal output relay (PLC → NC)
	BDY000~BDY1023	1024	Internal output data (PLC → NC)
M Internal relay	M0000~M10239	10240	Internal auxiliary relay
D Data register	D0000~D8191	8192	Data register

Local OM (OM unique to each task)

OM name	Address	Points	Description
LM Local relay	LM0000~LM4095	4096	Internal auxiliary relay
LD Local data register	LD0000~LD8191	8192	Data register
LT Local timer	LT000~LT511	512	Timer
LC Local counter	LC000~LC511	512	Counter
SM Special relay	SM00~SM255	256	Special relay
SD Special register	SD00~SD255	256	Special register
IX Index register	IX0~IX7	8	Index register
L Label	L0~L1023	1024	Label
GB Bit range used for variables	GB0~8191	8192	Used with ST/FBD languages
GW Word range used for variables	GW0~8191	8192	Used with ST/FBD languages

1.3 Overview of Built-in PLC Functions

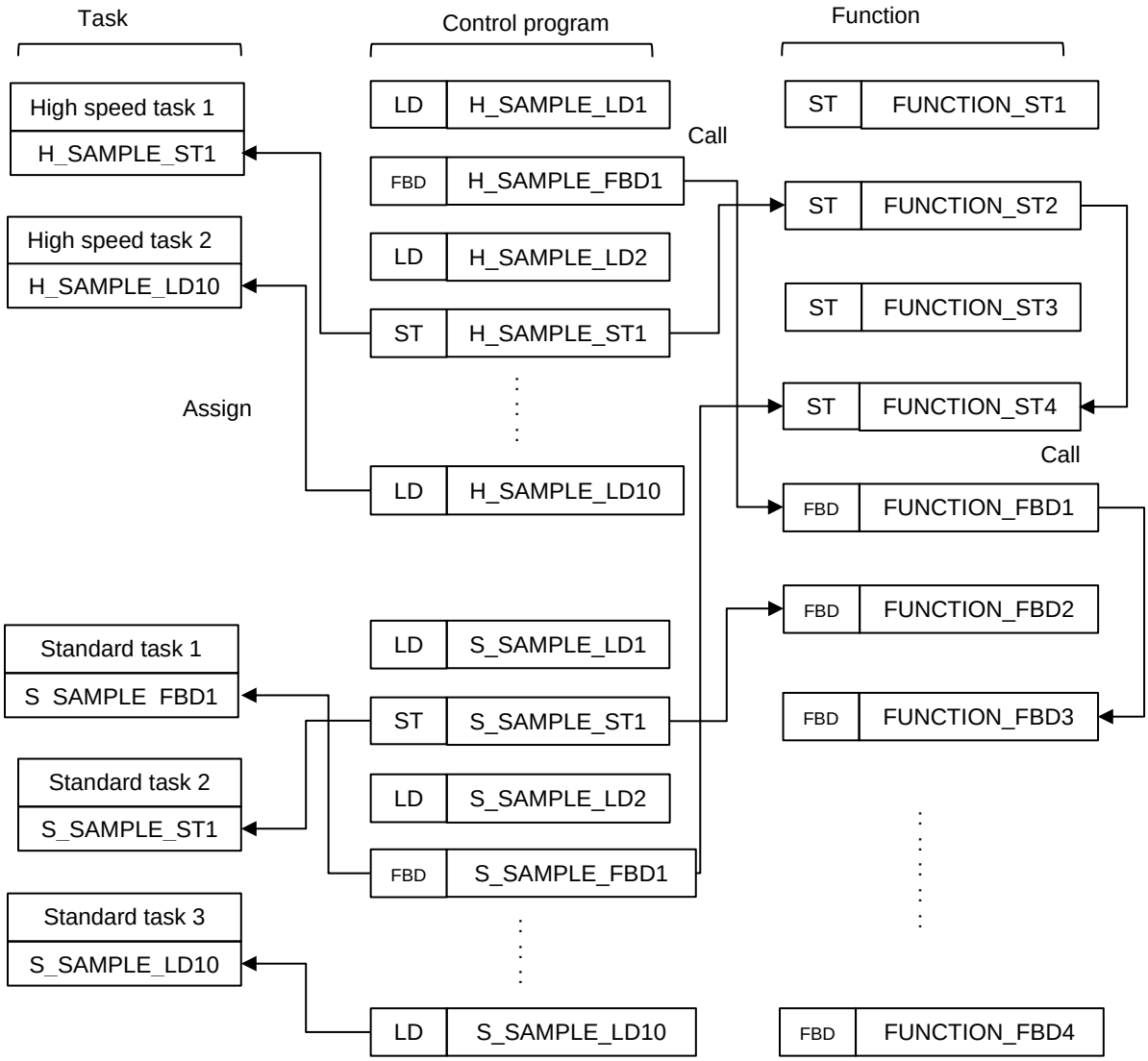
1.3.1 Task

The PLC system can execute 2 high speed control programs and 3 standard control programs at the same time.
Control programs are assigned and the part that is executing is referred to as a task. There are high speed tasks and standard tasks depending on the control program that is assigned.

	High speed runtime system	Standard runtime system
Minimum scan time	1 msec	10 msec
Maximum steps (NOTE 1)	Approx. 5,000 steps	Approx. 40,000 steps
Maximum number of simultaneous executions	2 (NOTE 2)	3 (NOTE 3)

(NOTE 1) The number of steps is displayed on the bottom of the edit screen and is the same for the LD, ST and FBD languages.
(NOTE 2) A maximum of 2 motions can be carried out from the 10 POUs.
There are two tasks: high speed task 1 and high speed task 2.
(NOTE 3) A maximum of 3 motions can be carried out from the 10 POUs.
Standard task 1, standard task 2 and standard task 3 refer to the three execution systems for operation.

There is a flow chart below showing the assignment of the control program tasks and the functions called from the control programs.



1.3.2 System Ladder

The No.10 control program is pre-registered to high speed task 2 and standard task 3, and this program provides control of the peripheral equipment (in the table below) that comes standard with the Brother Industries machine tools. This control program is referred to as the system ladder. Do not change the execution settings for the system ladder nor the assigned control program.
In addition, do not perform writing operations to the global OMs used in the factory-default settings.

Start switch / External start switch
Chip conveyor
Automatic door
CTS
Coolant
Chip shower
Spindle air blow (tool wash)
Jig shower
Cyclone filter
Chip auger

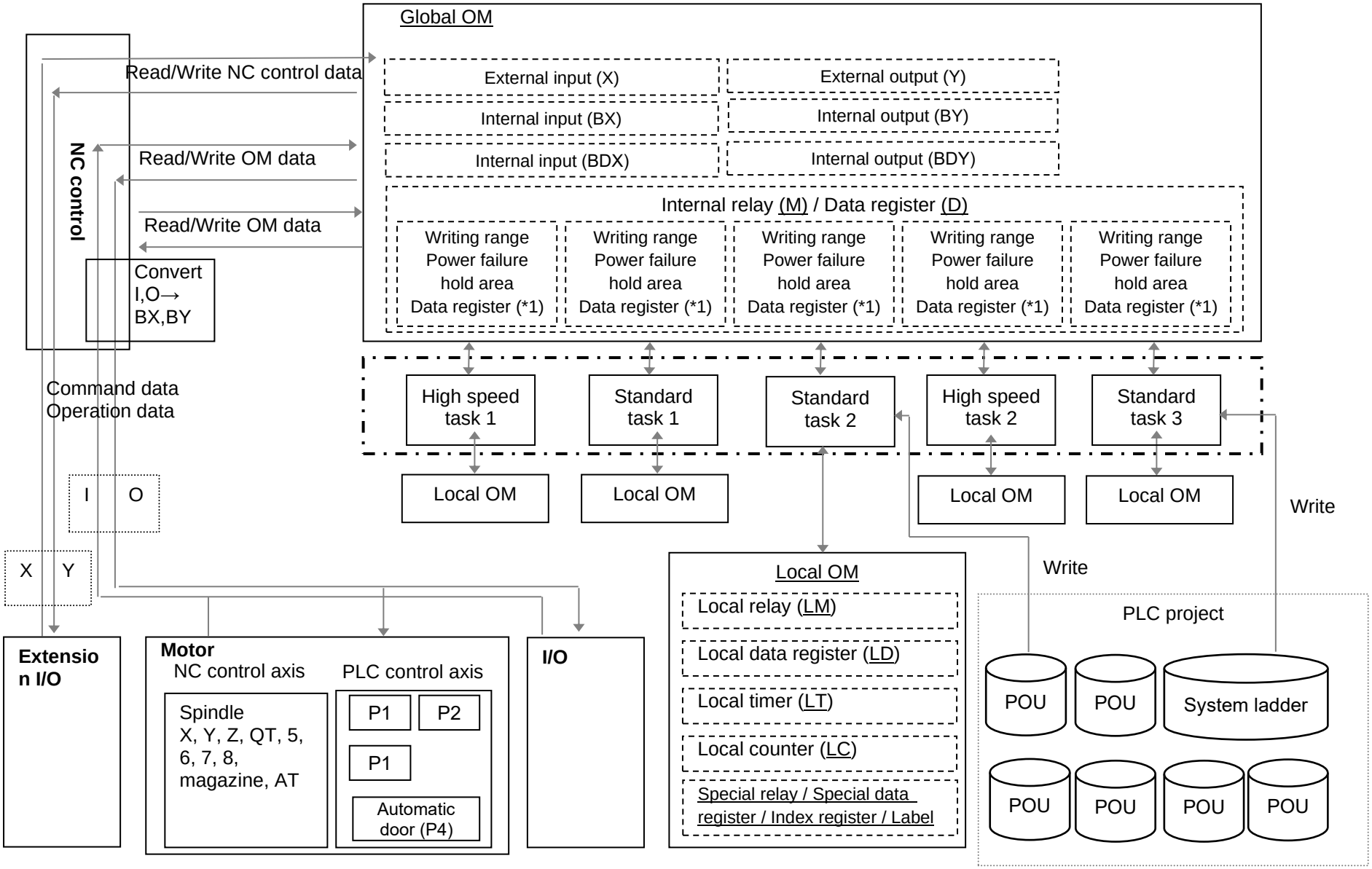
When making a change on the machine in high speed or standard control program 10, the <Parameter change> on the I/O <Extension maintenance> screen must be set to <1: Writable>. And, the <PLC data protection> needs to be set to <0: Invalid>.
When the version of the Brother Industries machine tool is updated, the version is automatically updated and overwritten regardless if there is a change in the factory-default settings of the high speed or standard control program 10.
In addition, execute the system ladder for high speed task 2 and standard task 3. If high speed task 2 and standard task 3 are stopped, the system assumes that the peripheral equipment (genuine devices of Brother Industries machine tools) is not operating properly, and a forced output is carried out on the safe side logic of the prescribed I/O port, regardless of the control program content.

1.3.3 Standard Control Program 1

In the factory-default settings, standard control program 1 controls the following signals.

Indicators (green, yellow and red)
Operation preparation
PLC rapid traverse override
PLC feedrate override
PLC spindle override
Internal light
Table light
Cycle time measurement

1.3.4 Overall View of Built-in PLC Functions

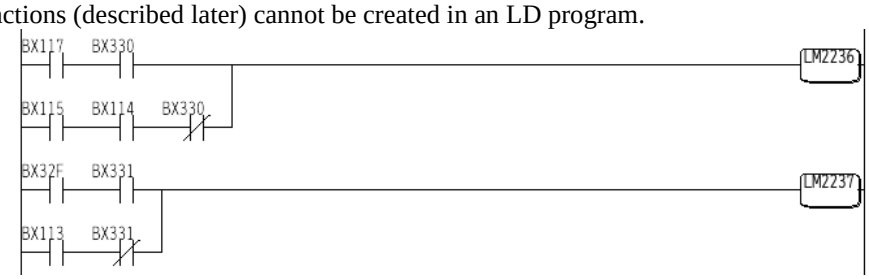


1.4 Overview of Programming Functions

There are 3 different programming languages that can be used with this PLC system. The programs that are created with these languages are referred to as control programs in this manual. Refer to each language programming chapter for an explanation of each language.

1.4.1 LD (Ladder) Language

This language is used for programming to express using the wiring as shown in the figure below. This language expresses the execution of a function and uses what is called a rail (left/right vertical lines) and establishes conditions on whether to energize or not from the left to the right (across a rung). This type of program is also called a sequence program because it runs from the top in a sequenced order.
A program created by the LD (ladder) language is referred to as an “LD program” in this manual. Functions (described later) cannot be created in an LD program.



1.4.2 ST (Structured Text) Language

This language uses a text format like C and BASIC programming languages.
Computation processing can be noted easily and it is good for visualizing a program.
A program created by the ST language is referred to as an “ST program” in this manual.

1.4.3 FBD (Function Block Diagram) Language

This language creates a program by using function blocks - which carry out an established process- and connects those blocks with connection lines in accordance with the process flow.
Since programs are expressed using graphics, the language is visually easy-to-understand.
This program created by the FBD language is referred to as an “FBD program” in this manual.

(NOTE) When using the programming function in the NC, all the editing functions cannot be used with the FBD language. When creating an FBD program, use the PLC programming software.

1.4.4 Programming Language Restrictions

There are some language restrictions with certain functions. The table below shows when calling is available or unavailable among the different language programs.

Available: Calling is possible Unavailable: Calling is not possible		Being called		
		LD program	ST program	FBD program
Calling runtime system/function	LD program	Not available	Available (NOTE 1)	Available (NOTE 1)
	ST program	Not available	Available (NOTE 2)	Available
	FBD program	Not available	Available	Available (NOTE 2)

(NOTE 1) The calcbox function can be used to call.
Refer to “Chapter 7 Programming function LD” for further details.
(NOTE 2) A program can call itself.
Refer to “Chapter 7 Programming function LD” for further details.

1.5 External Device Interface Function

There are inputs and outputs available for signals that link with an external device.

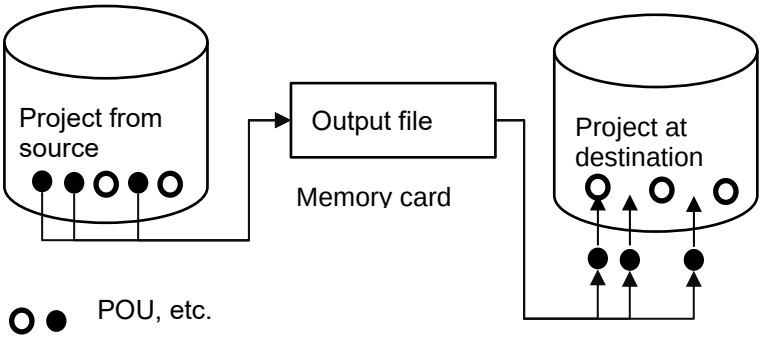
1. NC unit interface
   This NC unit interface is used for the input and output of signals and data that are used with NC unit and external devices. When using the NC unit itself for the input and output, the BD/BDX and BY/BDY are used, respectively, for the input and output in the OM.
   Refer to “Chapter 4 OM” for further details on the individual signals and data.

2. General I/O interface (standard terminal block/extension terminal block)
   There are 2048 input and output points available for the interface signals between external devices. These inputs and outputs are used to assign signals on the terminal block. The X and Y in the OM are used for the input and output, respectively, in the control program.
   Refer to “Chapter 5 External I/O signals” in the Installation Manual for details on how to assign signals to the terminal block.

3. Fieldbus interface
   This interface is available to use with the fieldbus to assign signals. The X and Y (or D) in the OM are used for the input and output, respectively, in the control program.
   Refer to “Chapter 9 (15) Fieldbus network” in Operation Manual I for further details.

1.6 Input/Output Between Projects

An external device and the PLC system can input and output POUs and other data between different projects.
The user can only select a POU for input and output during output.



However, the system ladder (high speed control program 10, standard control program 10) is excluded during input except for the following situations.

- When all data is input during a maintenance data transfer
- When the data protection for the system ladder is disabled

Refer to “3.2 Master station communication” in Operation Manual II for further details about the input and output procedure and about the available selections when outputting.

(NOTE) Do not change the PLC program name that is output because input and output will no longer be possible. In addition, do not edit a PLC program that is output in a software other than the PLC programming software, because the PLC program may be unable to open properly thereafter.

(This page was intentionally left blank.)

CHAPTER 2

PROCEDURE FOR USING THE PLC FUNCTION

2.1 Procedure for Using the PLC Function

2.1 Procedure for Using the PLC Function

This section describes the procedure when using the PLC function.

1. Identify the I/O signal that is required.
   Check the available I/O points and OM signal information.
2. Edit the POU (control program / function).
   - When editing on a computer
     use the PLC programming software (option).
   - When editing on the screen of the machine
     use the PLC programming function.

   It is possible to modify some POUs using the PLC programming function on the the machine, even without all editing functions being available. However, the PLC programming function on the machine is not suitable when creating a new program or when making modifications that cover a wide range. In this situation, use the PLC programming software (option).

3. Assign the edited control program to a task.
   On the <PLC (Program execution settings)> screen, the user must register the edited control program to a task and perform the <Write> operation.
   When performing the <Write> operation, all control programs that are automatically running may stop.
   (Ex: The alarm <<Control program stopped (standard task 1)>> is triggered.)
   Thereafter, when the **[RST]** key is pressed, the alarm is cancelled and the changed control program starts running.
   When using Ethernet to connect a computer with the NC in order to write a POU from the PLC programming software (option), be sure to also change the operation status (STOP → RUN).
4. Assign the external I/O signals.
   Set the <External input signal> and the <External output signal> from the <External I/O signal> in the data bank.
5. Set up the wiring and cables for the external I/O signals.
6. Check the motion while also checking the circuit (ladder).
   - When using the PLC programming function on the machine to check
     Use the monitor function from the PLC programming functions to perform the operation check.
   - When using the PLC programming software to perform this check
     Use the PLC programming software (option) to check the operation. Connect the PC and NC using Ethernet, and then perform this check.

(NOTICE 1) When performing PLC editing, press the **[EMERGENCY]** switch and then carry out the editing.

(NOTICE 2) Do not change the system ladder (high speed / standard control program 10). In addition, do not perform writing operations to the OMs that are used by the system ladder.

(NOTICE 3) If a POU (control program / function) is changed, print out a copy of the changed POU circuit and always make sure to post it near the machine. In addition, send the modified circuit diagram (printout using the PLC programming software) or the PLC system project file to the nearest Brother sales office or Brother approved service dealer. When using our repair services, etc., send or present the changed POU circuit beforehand.

(NOTICE 4) The PLC project comes pre-saved inside the machine when it is shipped out. Therefore, when changing a POU, the user must first read (load) the PLC project that is inside the NC, and then perform the procedure to make additions or changes to the PLC project.

(NOTE 1) The programming function on the machine and the PLC programming software (option) operations (via Ethernet) cannot be performed at the same time.

(NOTE 2) If the communication parameter (Ethernet/FTP) <Restrict Ethernet access> is set to <1: Yes>, set the user parameter (switch 1: system) <PLC program protection> to <0:Invalid> in order to perform a writing operation or check the motion from the PLC programming software (connected via Ethernet).

CHAPTER 3

BUILT-IN PLC

3.1 Built-in PLC Operation
3.2 Time Management for Built-in PLC
3.3 PLC Program Format

3.1 Built-in PLC Operation

3.1.1 At Power ON

The POUs (control program / function) are saved even when the machine power is turned OFF. The control programs assigned to tasks are automatically executed when the NC is started up.

When the power is turned ON, the OM data area (counter, timer and data register values) is cleared except for the OMs that are specified in the power failure hold area.

However, in the following situations, the control programs assigned to tasks are not automatically executed when the power is turned ON.

- When the alarm <<PLC program could not be written.>> is triggered by one of the following
  - When a control program assigned to a task still has not been converted
  - When there is an error in the control program assigned to a task

3.1.2 At Reset

The executing control programs are not affected even if the **[RST]** key on the machine is pressed.

When a control program assigned to a task is stopped and the **[RST]** key is pressed in one of the following situations, the program then begins to execute operation.

- When inputting/outputting a PLC program via an external I/O (data bank, maintenance, slave communication)

(NOTE) In the following situations, the alarm <<PLC program could not be written.>> is triggered and the control program assigned to a task will not be executed.

- When communicating with the PLC programming software
- When the PLC program cannot be converted
- When the power failure hold or writing range setting is not correct

3.1.3 Task Operation Check

In the following situations, the machine checks the operation of a control program assigned to a task.

- When starting up in memory operation or MDI operation

Alarm description	Stop	Cancel	Overview
There is a control program not running (****)	1	1	When at least one control program assigned to a task is not running (special relay SM8 for each task is OFF)

- When memory operation or MDI operation is in progress

Alarm description	Stop	Cancel	Overview
Control program stopped during operation (****)	5	2	When at least one control program assigned to a task is stopped (special relay SM8 for each task is OFF)

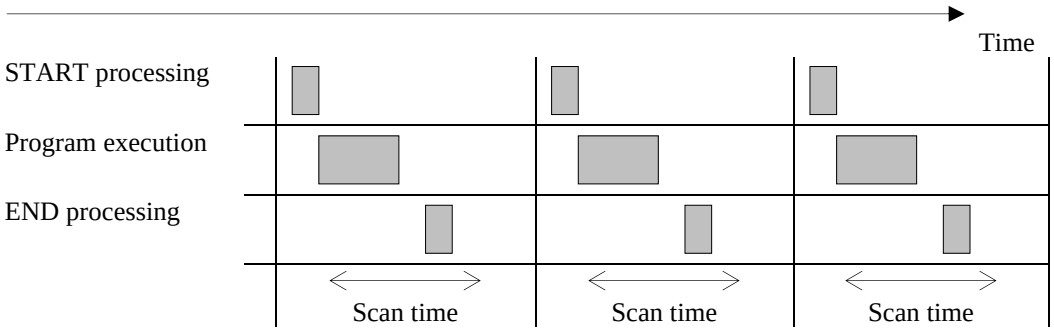
- When check is set to always-on (constant monitoring)

Alarm description	Stop	Cancel	Overview
Control program stopped (****)	4	2	When a control program assigned to a task stopped (special relay SM8 for each task is OFF)
Control program execution setting error (****)	5	2	When a control program has not been assigned to high speed task 2 or standard task 3

3.2 Time Management for Built-in PLC

3.2.1 Task Execution Chart

The task execution chart for the built-in PLC is shown below.



Each task executes a control program using time slices in simultaneous parallel processing. There is no priority order in this processing.
When one task cycle ends, the processing waits for the set time in the control program parameter <Scan time>.

When the scan time is longer than the set time in the control program parameter <Scan time>, the cycle ends and the next cycle is executed immediately. As a result, the execution cycle becomes longer.
The user can confirm the actual scan time on the <PLC (Status)> screen.
In addition, when changing the execution setting of a task, it impacts the scan times for the unchanged tasks, and as a consequence, the execution timing and scan times may change.

3.2.2 OM Synchronization

The writing area is fixed for each task in the global OMs M and D.
The OM for all areas can be read, but the writing operation can only be performed in the area specified for the writing range of the assigned task.

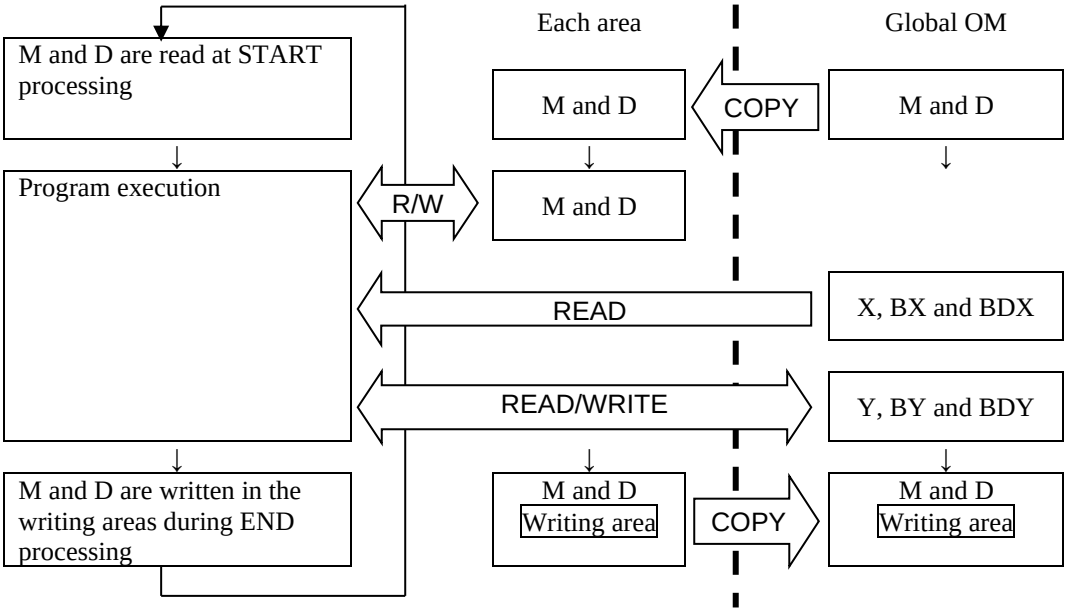
In the built-in PLC processing, the data for M and D is copied to the built-in PLC area during the START processing, and the reading or writing operation is performed in the built-in PLC area (where it was copied while the program is being executed). Thereafter, during the END processing, writing operations are performed to the global OM only for the area that is specified as the writing range. Accordingly, other parts that are not part of area specified as the writing range are not written to the global OM.

For global OMs (excluding M and D), direct reading or writing operations are performed while the program is being executed.
Accordingly, M and D are delayed when being written from other tasks, so be aware of this when creating control programs.

Writing to the global OM can be carried out on the <PLC (I/O control)> screen.

Refer to “5.2 Layout and operations on OM monitor screen” for details on the <PLC (I/O control)> screen.

OM name	Reading		Writing	
	At START	During execution processing	During execution processing	END processing
X		○		
Y		○	○	
BX		○		
BY		○	○	
BDX		○		
BDY		○	○	
M	○			○
D	○			○

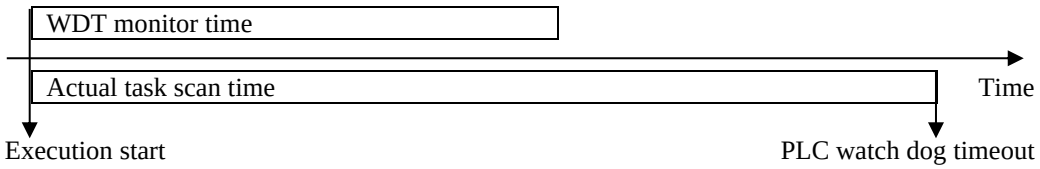


Local OMs are cleared for program writing.

3.2.3 Watch Dog

3.2.3.1 PLC Watch Dog Timeout

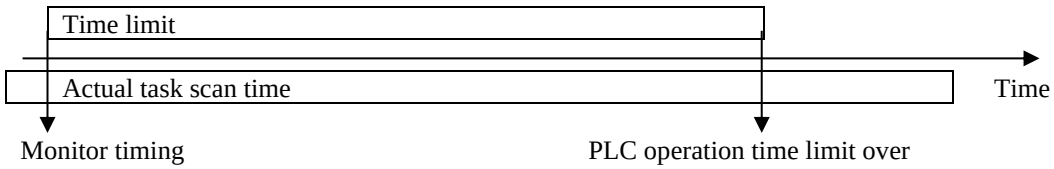
A task with an assigned control program monitors every scan time and checks whether the <WDT monitor time> has been exceeded.
When a watch dog error occurs, it triggers an NC alarm and stops the operation.



Triggered alarm	Time limit	Built-in PLC operation
<<PLC watch dog timeout (**)>> ** is the task name	<WDT monitor time>	Operation will stop, or continue, depending on the <Operation method in case of WDT error> in the parameter settings.

3.2.3.2 Monitoring in RUN Status

Apart from the PLC watchdog timeout, the NC monitors the RUN counter (SD255) for each task. An NC alarm is triggered and operation stops when the RUN counter (SD255) does not change in the time period below, when the NC is in memory operation mode , or in MDI operation mode, or during the RUN operation (SM8 is 1).

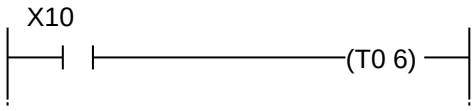


Triggered alarm	Time limit	Built-in PLC operation
<<PLC operation time limit over (****)>> ****: Task name	High speed task 1: 1000 ms Standard task 1: Standard task 2: 10000 ms High speed task 2: Standard task 3: 100 ms	Operation continues

3.2.4 Timer

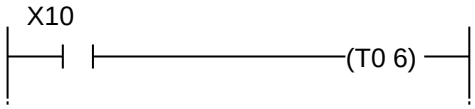
The minimum timer unit is 1 ms in high speed task control programs.
The minimum timer unit is 10 ms in standard task control programs.
Timer refresh processing is performed during the END processing. As a result, the scan time is added to the timer's count up operation. Accordingly, the accuracy of the timer depends on the scan time. If the execution time exceeds the scan time, the count up operation is performed during the END processing. In this way, the time expires during the execution time.

For control programs in the high speed tasks



After X10 turns ON for timer T0, the time up operation is performed in 6ms.

For control programs in the standard speed tasks



After X10 turns ON for timer T0, the time up operation is performed in 60ms.

3.3 PLC Program Format

If the alarm <<Check sum error (PLC: common file)>> is triggered, a format operation is required for restoring the PLC program. When using this operation to format, all the PLC programs including the default PLC programs are deleted.

3.3.1 Operation Procedure

(1) Set the **[DATA PROTECTION]** to “Disable”.
(2) Move the cursor to <Parameter change> on the <Extension maintenance> screen, and select <Writable>.
(3) Set the user parameter (switch 1: system) <PLC program protection> to <0:Invalid>.
(4) Enter **[P]**, **[L]**, **[C]**, **[F]**, **[M]** and **[T]** in the <Maintenance input>, and press and **[ENT]** key.
(5) The alarms <<All control programs are stopped>> and <<Turn power off.>> are triggered.
(6) Turn OFF the power.

(NOTE 1) If another screen besides the <PLC (Menu)> screen is displayed in the support application (PLC), the alarm <<There is a PLC file still being edited/communicating>> is triggered and the format operation is not possible.

(NOTE 2) If there is checksum error in a high speed / standard control program or user / system defined function, recovery is possible by deleting the corresponding program.

CHAPTER 4

OM

4.1 OM List
4.2 Global OM Description
4.3 Local OM Description
4.4 PLC Data Input
4.5 PLC Data Output

4.1 OM List

OM stands for “Object Memory” and is a generic term for the memory used for PLC operations. These OMs contain various memory areas, such as for the external I/O as well as the timer.

4.1.1 Global OM

This OM is common or shared among each control program.

OM name	Address	Points	Description
X Input relay	X000 to X7FF	2048	External input relay
	BX000 to BX7FF	2048	Internal input relay (NC→PLC)
	BDX000 to BDX1023	1024	Internal input data (NC→PLC)
Y Output relay	Y000 to Y7FF	2048	External output relay
	BY000 to BY7FF	2048	Internal output relay (PLC→NC)
	BDY000 to BDY1023	1024	Internal output data (PLC→NC)
M Internal relay	M0000 to M10239	10240	Internal auxiliary relay
D Data register	D0000 to D8191	8192	Data register

The writing range can be set in the parameter for M (internal relay) and D (data register). When the writing range is not set, the write operation cannot be carried out. Refer to “6.6 Parameter Settings ” for further details about setting.

4.1.2 Local OM

This OM is specific to each control program.

OM name	Address	Points	Description
LM Local relay	LM0000 to LM4095	4096	Internal auxiliary relay
LD Local data register	LD0000 to LD8191	8192	Data register
LT Local timer	LT000 to LT511	512	Timer
LC Local counter	LC000 to LC511	512	Counter
SM Special relay	SM00 to SM255	256	Special relay
SD Special register	SD00 to SD255	256	Special register
IX Index register	IX0 to IX7	8	Index register
L Label	L0 to L1023	1024	Label

4.2 Global OM Description

The global OM is used for the following purposes:

- The global variables are used in the data communication between the built-in PLC and the external devices / NC.
- The global variables are used in the data communication between control programs.
- Maintains desired data even when the power is turned OFF.

4.2.1 External Input Relay (External→PLC, X000 to X7FF)

This input signal is from an external source.
Set the terminal block assignment using the external I/O signals (external input signal).

OM name			X000 to X7FF (hexadecimal shown)
Total number of signals			2048
Type			Bit
Output specification			Not available
Double coil check			-
Record when power is OFF			-
Index addressing			Available
Signal assignment	IO PCB	PCB quantity	1
		Number of signals per PCB	16
		Terminal name	IO_PI_000 to 015
	EXIO PCB (Extension terminal block)	PCB quantity	32
		Number of signals per PCB	32
		Terminal name	EXIO** INPUT0 to 31
	FC PCB	PCB quantity	1
		Number of signals per PCB	PROFIBUS DP: 1024 DeviceNet: 1024 CC-Link: 128
		Terminal name	-

(NOTE 1) The output setting indicates whether it is possible or not to use the coil or the application command D (Destination) as an output.

(NOTE 2) 32 signals per PCB are displayed on the screen for an EXIO PCB even if there are only 16 signals per PCB.

4.2.2 External Output Relay (PLC→External, Y000 to Y7FF)

This output signal is used to an external source. It indicates the internal status of the PLC in bits.
Set the assignment to the terminal block using the external I/O signals (external output signal).

OM name			Y000 to Y7FF (hexadecimal shown)
Total number of signals			2048
Type			Bit
Output specification / Output range specification			Available / None
Double coil check			Yes
Record when power is OFF			Volatile
Index addressing			Available
Signal assignment	IO PCB	PCB quantity	1
		Number of signals per PCB	16
		Terminal name	IO_PO_100 to 115
	EXIO PCB (Extension terminal block)	PCB quantity	32
		Number of signals per PCB	32
		Terminal name	EXIO**_INPUT0 to 31
	FC PCB	PCB quantity	1
		Number of signals per PCB	PROFIBUS DP: 1024 DeviceNet: 1024 CC-Link: 128
		Terminal name	-

4.2.3 Internal Input (NC→PLC, BX000 to BX7FF)

4.2.3.1 Overview

This signal is from the NC→PLC.
It indicates the internal status of the NC in bits.

OM name	BX000 to BX7FF (hexadecimal shown)
Total number of signals	2048
Type	Bit
Output specification	Not available
Double coil check	-
Record when power is OFF	-
Index addressing	Available

4.2.3.2 Signal List

The list of signals from the NC→PLC is shown below.

OM name : This column is used for specifying the signal in the PLC program description.
Symbol : This symbol is for supporting the OM. When the OM comment displayed is enabled, this symbol is displayed on the screen.
Use : This column shows whether it is provided as a factory default setting. The following symbols indicate the control program being used.
High speed control program 10 : H10
Standard control program 1 : S1
Standard control program 10 : S10

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BX000 to BX008	Zero point position output X-axis Zero point position output Y-axis Zero point position output Z-axis Zero point position output spindle Zero point position output No 4-axis Zero point position output No 5-axis Zero point position output No 6-axis Zero point position output No 7-axis Zero point position output No 8-axis	ZPX ZPY ZPZ ZPSP ZP4 ZP5 ZP6 ZP7 ZP8	Common	When the machine zero return operation is completed, ZPX, ZPY, ZPZ, ZP4, ZP5, ZP6, ZP7 and ZP8 turn ON. The signals are separate for each axis. In addition, the signal turns OFF when traveling along the axis from the machine zero point, and it turns ON when the machine returns back to the zero point again. The ZPSP turns ON when the spindle orientation operation has been completed. It turns OFF with the spindle rotation, the reset signal and the **[SP.STOP]** key. (NOTE 1) When the stop level 5 (servo OFF level) alarm occurs, the signal turns OFF. (NOTE 2) If the machine is equipped with QT, the ZP4 turns ON when the No 4-axis machine coordinate is 0.000. ZP4 does not turn ON at the index position.	
BX010 to BX017	X-axis position Y-axis position Z-axis position No 4-axis position No 5-axis position No 6-axis position No 7-axis position No 8-axis position	XPOSSW YPOSSW ZPOSSW 4POSSW 5POSSW 6POSSW 7POSSW 8POSSW	Common	Signal turns ON when the machine coordinates fall within the range (including the boundary value) indicated in the user parameter (switch 2: position signal output range) <*-axis position signal output range (-)/(+)>. Signal turns ON when BY375 is ON and when the machine coordinates fall within the range (including the boundary value) indicated in the user parameter <*-axis position signal output range 3 (-)/(+)>. (Regardless of the ON/OFF status for the servo)When out of range, the signal turns OFF.	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BX018 to BX01F	X-axis position 2 Y-axis position 2 Z-axis position 2 4th-axis position 2 5th-axis position 2 6th-axis position 2 7th-axis position 2 8th-axis position 2	XPOSSW2 YPOSSW2 ZPOSSW2 4POSSW2 5POSSW2 6POSSW2 7POSSW2 8POSSW2	Common	Signal turns ON when BY376 is OFF and when the machine coordinates fall within the range (including the boundary value) indicated in user parameter <*-axis position signal output range 2 (-)/(+)>. Signal turns ON when BY376 is ON and when the machine coordinates fall within the range (including the boundary value) indicated in user parameter <*-axis position signal output range 4 (-)/(+)>. (Regardless of the ON/OFF status for the servo) When out of range, the signal turns OFF.	
BX020 to BX027	X-axis in motion Y-axis in motion Z-axis in motion No 4-axis in motion No 5-axis in motion No 6-axis in motion No 7-axis in motion No 8-axis in motion	MOVEX MOVEY MOVEZ MOVE4 MOVE5 MOVE6 MOVE7 MOVE8	Common	The signal turns ON when the axis is in motion. (NOTE 1) This signal is not related to the operation mode. (NOTE 2) The signal does not turn ON during machine lock.	
BX030 to BX037	X-axis in-position Y-axis in-position Z-axis in-position No 4-axis in-position No 5-axis in-position No 6-axis in-position No 7-axis in-position No 8-axis in-position	INPXON INPYON INPZON INP4ON INP5ON INP6ON INP7ON INP8ON	Common	After the in-position operation has been completed, the signal turns ON. Use this signal for notifying when the positioning is complete or to determine if the positioning is within the acceptable range.	
BX03F	All axes in-position	ALINPOSI	Common	When all in-position operations have been completed for the X-axis, Y-axis, Z-axis, No 4-axis, No 5-axis, No 6-axis, No 7-axis and No 8-axis, the signal turns ON. If the in-position operation has not been completed for one of the above axes, the signal turns OFF.	
BX040	Machine zero return finished	ORGFIN	Common	Turns the power OFF, and then back ON.	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BX041 to BX046	No. 1 reference position return finished No. 2 reference position return finished No. 3 reference position return finished No. 4 reference position return finished No. 5 reference position return finished No. 6 reference position return finished	REFIN REFIN2 REFIN3 REFIN4 REFIN5 REFIN6	NC	Turns ON when the G30 command is executed in MDI and memory operation mode, or when all axes (X, Y, Z, 4, 5, 6, 7 and 8) have returned to the reference positions No.1 through No.6 using an external signal (EXREF and EXREF2 through EXREF6). OFF conditions 1. When at least 1 axis for the axes X, Y, Z, 4, 5, 6, 7 and 8 is not in a reference position (No.1 through No.6) 2. When a stop level 5 alarm is triggered (However, when the user parameter (reference point/conversation language zero point) <Reference point return complete signal output at servo off> is set to <1: Yes> and the distance from the reference point is within the range that is set in the parameter <Reference point allowable range at servo off>, then the signal stays ON even when the servo is OFF.) (NOTE 1) Even when all axes are moved to the reference positions in manual operation mode, the signal does not turn ON. (NOTE 2) When the lathe spindle is selected and all axes except the lathe spindle have returned to the reference positions No.1 through No.6, the signal turns ON.	
BX047 to BX048	Machine zero return finish 2 Machine zero return finish 3	ORGFN2 ORGFN3	Conversation	Outputs when the No.2 or No.3 machine zero return operation is completed using a program command and external signal. OFF conditions 1. When the at least 1 axis is not at the No.2 or No.3 zero point 2. When the servo is OFF due to an emergency stop, etc. (NOTE) The signal does not output when positioning to the No.2 and No.3 zero points manually. The signal also does not output when positioning to or passing through the No.2 and No.3 zero points using a positioning command. In addition, when the user parameter (reference point/conversation language zero point) <Zero point return complete signal output at servo off> is set to <1: Yes> while the servo is OFF and when it is within the range that is set in the parameter <Zero point allowable range at servo off>, then the signal stays ON.	
BX050 to BX054	Clamping (No 4-axis) Clamping (No 5-axis) Clamping (No 6-axis) Clamping (No 7-axis) Clamping (No 8-axis)	CLPON4 CLPON5 CLPON6 CLPON7 CLPON8	Common	The signal turns ON when the axis equipped with a clamp setup is in a clamped status.	
BX058 to BX05B	No 5-axis unclamp No 6-axis unclamp No 7-axis unclamp No 8-axis unclamp	UNCLP5 UNCLP6 UNCLP7 UNCLP8	Common	Indicates the unclamp status for the No 5 through No 8-axes. ON is output when unclamping and OFF is output when clamping.	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BX080	P1-axis command processing	BUSY_P1	Common	Command processing. The command is not accepted while the signal is ON. An alarm is triggered. The signal turns ON during a command operation on the built-in PLC. The signal turns ON during a command operation in manual operation mode. If the machine is equipped with a handle, the signal turns ON when the handle axis is set as an additional axis and the additional axis is set as the P-axis. This signal turns ON when the <Door interlock mode> is set to <Maintenance>. This signal turns ON during the brake load test, or when there is a brake load test error. The signal turns ON when the PLC-axis does not operate due to an NC alarm, such as an alarm that is triggered due to an emergency stop or when a PLC related parameter is changed.	
BX081	P1-axis alarm triggered	ALARM_P1	Common	The signal turns ON during an alarm. If the PLC-axis control signal is OFF, the alarm can be reset with the **[RST]** key. The servo may automatically turn OFF depending on the alarm type. The alarm can be reset or cleared with a reset command. The alarm information is set by the alarm/warning or servo alarm data register.	
BX082	P1-axis servo ON ready	SRDY_P1	Common	The signal turns ON when the servo can be turned ON. If the signal is OFF and a servo ON command is received, an alarm is triggered.	
BX083	P1-axis servo turned ON	SVON_P1	Common	The signal turns ON when the servo ON command is completed.	
BX084	P1-axis command completed	CMDOK_P1	Common	Command completed. The signal turns ON during the initializing status when the power is turned ON, and when a positioning or reference position return command is executed. (The signal turns ON when in-position at the end point. If equipped with a clamp mechanism, the signal turns ON when clamped.) The signal turns OFF if a new command is received. The signal also turns ON after a reset operation is completed and when the PLC-axis control signal turns OFF.	
BX085	P1-axis reference position return finished	RETOK_P1	Common	Reference position return operation completed. The signal turns ON when a command is completed for the machine coordinate at the reference position.	
BX086	P1-axis moving	MOVE_P1	Common	Axis in motion. The signal turns ON while the axis is in motion. The signal turns OFF if the speed command is 0 during a positioning command or a reference position return command.	
BX087	P1-axis in-position finished	INPOK_P1	Common	In-position OK. The signal turns ON when in position. The signal turns ON also when the STOP command is turned ON and operation stops temporarily. The signal turns OFF when a new command is received or when a command is re-issued (re-started).	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BX088	P1-axis rapid feed (close proximity) speed controlled	NEAR_V_P1	Common	The speed is controlled when the proximity speed is engaged.	
BX089	P1-axis clamped	CLP_P1	Common	The signal turns ON when clamped. This signal is only valid when the machine is equipped with a clamp mechanism. The signal is always OFF if there is no clamp mechanism.	
BX08A	P1-axis warning issued	WRN_P1	Common	The signal turns ON when a warning (an error that does not stop travel) is issued. If the PLC-axis control signal is OFF, the alarm can be reset with the **[RST]** key. The warning can be reset or cleared with the RESET command. The warning information is set by the alarm/warning or servo alarm data register.	
BX090	P2-axis command processing	BUSY_P2	Common	This is the status for the 2nd PLC-axis. The content is the same as the 1st axis.	
BX091	P2-axis alarm triggered	ALARM_P2	Common		
BX092	P2-axis servo ON ready	SRDY_P2	Common		
BX093	P2-axis servo turned ON	SVON_P2	Common		
BX094	P2-axis command completed	CMDOK_P2	Common		
BX095	P2-axis reference position return finished	RETOK_P2	Common		
BX096	P2-axis moving	MOVE_P2	Common		
BX097	P2-axis in-position finished	INPOK_P2	Common		
BX098	P2-axis rapid feed (close proximity) speed controlled	NEAR_V_P2	Common		
BX099	P2-axis clamped	CLP_P2	Common		
BX09A	P2-axis warning issued	WRN_P2	Common		
BX0A0	P3-axis command processing	BUSY_P3	Common	This is the status for the 3rd PLC-axis. The content is the same as the 1st axis.	
BX0A1	P3-axis alarm triggered	ALARM_P3	Common		
BX0A2	P3-axis servo ON ready	SRDY_P3	Common		
BX0A3	P3-axis servo turned ON	SVON_P3	Common		
BX0A4	P3-axis command completed	CMDOK_P3	Common		
BX0A5	P3-axis reference position return finished	RETOK_P3	Common		
BX0A6	P3-axis moving	MOVE_P3	Common		
BX0A7	P3-axis in-position finished	INPOK_P3	Common		
BX0A8	P3-axis rapid feed (close proximity) speed controlled	NEAR_V_P3	Common		
BX0A9	P3-axis clamped	CLP_P3	Common		
BX0AA	P3-axis warning issued	WRN_P3	Common		

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BX0B0	P4-axis command processing	BUSY_P4	Common	This is the status for the 4th PLC-axis. The content is the same as the 1st axis.	
BX0B1	P4-axis alarm triggered	ALARM_P4	Common		
BX0B2	P4-axis servo ON ready	SRDY_P4	Common		
BX0B3	P4-axis servo turned ON	SVON_P4	Common		
BX0B4	P4-axis command completed	CMDOK_P4	Common		
BX0B5	P4-axis reference position return finished	RETOK_P4	Common		
BX0B6	P4-axis moving	MOVE_P4	Common		
BX0B7	P4-axis in-position finished	INPOK_P4	Common		
BX0B8	P4-axis rapid feed (close proximity) speed controlled	NEAR_V_P4	Common		
BX0B9	P4-axis clamped	CLP_P4	Common		
BX0BA	P4-axis warning issued	WRN_P4	Common		
BX0C0 to BX0C3	P1-axis unclamp P2-axis unclamp P3-axis unclamp P4-axis unclamp	UCLPP1 UCLPP2 UCLPP3 UCLPP4	Common	Indicates the unclamp status for the P1 through P4-axes. ON is output when unclamping and OFF is output when clamping.	
BX0E0	Spindle rotating	SPTURN	Common	Turns ON while the spindle is rotating. It turns OFF when the spindle is stopped. Note, it does not turn ON during the orientation operation.	
BX0E1	Normal spindle rotating	M03OUT	Common	Turns ON while the spindle is performing normal rotation. It turns ON while the spindle is performing normal rotation even during a tap operation. It turns OFF when the spindle is stopped, and when the spindle is performing reverse rotation. Note, it does not turn ON during the orientation operation.	
BX0E2	Reverse spindle rotating	M04OUT	Common	Turns ON while the spindle is performing reverse rotation. It turns ON while the spindle is performing reverse rotation even during a tap operation. It turns OFF when the spindle is stopped, and when the spindle is performing normal rotation. Note, it does not turn ON during the orientation operation.	
BX0E3	Spindle speed reached	SPREVOK	Common	The conditions to turn ON the signal may vary depending on the type of command for the spindle. • When normal / reverse spindle rotation command is issued: The signal is turned ON when the command spindle speed is reached. • When a stop / orient / tap command is issued: The signal is normally OFF.	
BX0E4	Spindle in-position	SPIPOK	Common	The signal turns ON when the spindle is stopped.	
BX0E5	Orientation finished	SPORFN	Common	Turns ON when the orientation operation has been completed. It turns OFF during spindle rotation, with the reset signal and with the **[SP.STOP]** key.	
BX0E6	Tapping	TAPMOV	Common	The signal turns ON when the tap operation is performed. (NOTE) The signal does not turn ON during machine lock.	
BX0E7	Thread cutting in progress	THRDMOV	NC	The signal turns ON when thread cutting is in progress. (NOTE) The signal does not turn ON during machine lock.	
BX0E8	Turning spindle is rotating	LSTURN	NC	Turns ON when there is a lathe spindle and when the lathe spindle is rotating. It turns OFF when the lathe spindle stops.	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BX0E9	Turning spindle is rotating normally	M303OUT	NC	There is a turning spindle and it turns ON when turning spindle is rotating normally. It turns OFF when the turning spindle is stopped, and when the turning spindle is performing reverse rotation.	
BX0EA	Turning spindle is rotating in reverse	M304OUT	NC	There is a turning spindle and it turns ON when turning spindle is rotating in reverse. It turns OFF when the turning spindle is stopped, and when the turning spindle is performing normal rotation.	
BX0EB	Turning spindle speed reached	LSPREVOK	NC	The conditions to turn ON the signal may vary depending on the type of command for the turning spindle. • When a normal / reverse turning spindle rotation command is issued: The signal is turned ON when the command spindle speed is reached. • When a stop command is issued: The signal is normally OFF.	
BX0EF	Turning spindle selection	LSSLCT	NC	Turns ON when there is a lathe spindle and the lathe spindle is selected.	
BX0F0 to BX0F6	Magazine index 1 Magazine index 2 Magazine index 4 Magazine index 8 Magazine index 16 Magazine index 32 Magazine index 64	MGZ1 MGZ2 MGZ4 MGZ8 MGZ16 MGZ32 MGZ64	Common	The index magazine No. (port No.) is output. Ex 1) When the index magazine No. is 1. MGZ1 = ON MGZ2 = OFF MGZ4 = OFF MGZ8 = OFF MGZ16 = OFF MGZ32 = OFF MGZ64 = OFF Ex 2) When the index magazine No. is 21. MGZ1 = ON MGZ2 = OFF MGZ4 = ON MGZ8 = OFF MGZ16 = ON MGZ32 = OFF MGZ64 = OFF Ex 3) When the magazine is not indexed (= 0). MGZ1 = OFF MGZ2 = OFF MGZ4 = OFF MGZ8 = OFF MGZ16 = OFF MGZ32 = OFF MGZ64 = OFF	
BX0F7	Magazine rotating	MGZROT	Common	The signal turns ON when the spindle is turning or rotating.	
BX0F8 to BX0F9	Pallet index finished_1 Pallet index finished_2	PFIN1 PFIN2	Common	PFIN1: Turns ON when pallet 1 is indexed on the outer side. PFIN2: Turns ON when pallet 2 is indexed on the outer side. The signal turns OFF when traveling from the index position, and it turns ON when the machine returns back to the index position again. (NOTE) The signal turns OFF when the servo is OFF due to an emergency stop, etc. OFF is normally output for machines without QT.	S10
BX0FA	External additional axis 1 indexing	OUT1ON	Common	Signal turns ON while the additional axis (specified in the external pallet user parameter (switch 2: outside index) <Outer rotary table 1 switch address>) is indexing from a panel switch operation or input signal.	
BX0FB	External additional axis 2 indexing	OUT2ON	Common	Signal turns ON while the additional axis (specified in the external pallet user parameter (switch 2: outside index) <Outer rotary table 2 switch address>) is indexing from a panel switch operation or input signal.	
BX100	NC preparation completed	NCOK	Common	Turns ON when the NC preparations are complete and all operations such as axis travel are OK and ready to be received. It turns OFF when an alarm is triggered that turns OFF the servo.	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BX101	Resetting	PRRST	Common	Outputs with one shot (while reset lamp is turned on + 300 msec.) when resetting. The signal does not turn ON when the **[M.LCK]** key is pressed.	S1 S10
BX102	Resetting output	RSTOUT	Common	Outputs as one shot (300 msec. or while reset lamp is lit, whichever is longer) when the reset LED is lit.	S10
BX103	Servo relay 1 Contact ON	KFS_ONCHK	Common	Relay contact ON: 1 Relay contact OFF: 0	
BX106	Automatic data backup	BACKUP	Common	Signal shows whether the data is automatically backing up or not. Automatic data backup not in progress: 0 Automatic data backup in progress: 1	
BX108	Cycle time in progress	CYTMHOLD	Common	Signal turns ON when the cycle time is in progress.	
BX10C	Power failure hold area change notice	HLDMMNT	Common	The input all data operation is performed to maintain and transfer the data, and the power failure hold area is switched. Note, this signal remains ON from the moment the power is turned ON until the power turns OFF, when the input all data operation is performed to maintain data. When using the power failure hold area, creating a recovery circuit is recommended while this signal is turned ON.	
BX10D	Power failure hold area check notice	HLDMINF	Common	Turns ON if the NC unit fails to back up in the power failure hold area on the built-in PLC, when the power supply was turned OFF the previous time. Note, this signal remains ON from moment the power is turned ON until the power turns OFF, when it fails to back up. When using the power failure hold area function, creating a recovery circuit is recommended while this signal is turned ON.	
BX10E	Data protection key	UNLOCK	Common	Signal turns ON when the **<**DATA PROTECTION**>** key is set to <Disable>.	
BX10F	Power supply OFF processing	PWROFF	Common	Turns ON when the power supply is turned OFF.	S2
BX110 to BX114	Alarm stop level 1 Alarm stop level 2 Alarm stop level 3 Alarm stop level 4 Alarm stop level 5	ASTLV1 ASTLV2 ASTLV3 ASTLV4 ASTLV5	Common	When an alarm is triggered for a particular alarm level, the signal for the corresponding level turns ON. When alarms are triggered for multiple levels, each corresponding signal turns ON.	S1 S10
BX115	Error output	ALM	Common	Outputs when a stop level 5 (servo OFF) or a cancel level 3 (power OFF required) alarm is triggered. However, the following alarms are excluded. • <<Door open>> • <<Standby mode>> • <<Enter your password. >> • <<**[Enable]** switch was turned OFF.>> • <<Operation preparation was turned OFF.>> Remove the cause and reset to restore operation. (Reference) Refer to “Chapter 2 – 2.12 Alarm code list” in the Data Bank & Alarm Manual for further details about the alarms. Note that the ALM signal does not output when the power is OFF because the transistor outputs.	S1

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BX116	Error output 2	ALM2	Common	Turns ON when an alarm is triggered that is higher than a stop level alarm as per the user parameter setting <ALM2 signal stop level>. However, the following alarms are excluded. • <<Door open>> • <<Standby mode>> • <<Enter your password.>> • <<**[Enable]** switch was turned OFF.>>	
BX117	Error output 3	ALM3	Common	The signal turns ON when a stop level 5 alarm is triggered. However, the following alarms are excluded. • <<Door open>> • <<Standby mode>> • <<Enter your password.>> • <<**[Enable]** switch was turned OFF.>> • <<Operation preparation was turned OFF.>>	S1
BX118	Emergency switch	EMGSW	Common	Turns ON when the **[EMERGENCY]** switch is pressed. Turns OFF when the **[EMERGENCY]** switch is turned and the depressed status is released.	
BX119 to BX11A	Error output LV1 Error output LV2	ALMLV1 ALMLV2	Common	**ALMLV1** / **ALMLV2** / **Alarm status** OFF / OFF / No error OFF / ON / 1 cycle stop level ON / OFF / Block stop level ON / ON / Higher than a temporary stop level	
BX11B	Air pressure drop	AIRDWN	Common	Turns ON when the air pressure drops. It turns OFF when the air pressure recovers from a pressure drop. It is normally OFF for machines that do not use air.	
BX11C	Coolant M overheat	COOLOH	Common	Turns ON when the coolant motor overheats.	S10
BX11D	Chip shower M overheat	CHPOH	Common	Turns ON when the chip shower motor overheats.	
BX11E	Battery alarm	BATALM	Common	Signal turns ON when the battery for maintaining the backup data is low.	
BX11F	Battery alarm on relocation detection device	RDBTAL	Common	The signal turns ON when the battery for the relocation detection device is low.	
BX120	Manual operation mode	MANUMOD	Common	Turns ON when the current mode is manual operation mode. Turns OFF for all other modes.	
BX121	MDI operation mode	MDIMOD	Common	Turns ON when the current mode is MDI operation mode. Turns OFF for all other modes.	
BX122	Memory operation mode	MEMMOD	Common	Turns ON when the current mode is memory operation mode. Turns OFF for all other modes.	S10
BX123	Program edit mode	EDITMOD	Common	Turns ON when the current mode is program edit mode. Turns OFF for all other modes.	
BX129	Release key depressed	RELESE	Common	Indicates when the **[RELSE]** key is pressed.	S10
BX12A	Start switch	STA	Common	This signal is for the **[START]** switch on the operation panel.	S10
BX12B	Stop	STPOUT	Common	This is the same as the **[FEED HOLD]** switch input.	
BX12C	Stopped	STPLED	Common	Indicates that operation has stopped. This is the same as the stop LED.	S10
BX12D	Pallet start output	QTSTART	Common	This signal is for the **[PALLET START]** switch on the operation panel.	S10
BX12E	Pallet start LED	RDYLED	Common	Indicates that startup is ready.	
BX130	Single operation	SINGL	Common	Indicates that single operation is in progress.	
BX131	Dry run	DRYRUN	Common	Indicates that dry run mode is enabled.	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BX132	Optional block skip check	BSKIP	NC	Indicates when the optional block skip is valid.	
BX133	Machine lock check	MCNLK	Common	Indicates when the machine lock is valid.	S10
BX134	Optional stop check	OPTSTP	NC	Indicates when the OPT stop is enabled.	
BX135	Coolant key	COOLSW	Common	Indicates that the **[CLT.P]** key is ON.	
BX136	Coolant turned ON	COLON	Common	Indicates the status of the **[CLT.P]** key. After turning ON the **[CLT.P]** key, it changes into standby status and turns OFF when the "Automatic coolant off time" continues even though the machine is not being operated, or when the stop level 5 alarm is triggered by the ALM turning ON. Symbol / OFF / ON / Standby status COOLSW / OFF / ON / ON COLON / OFF / ON / OFF	S10
BX137	Chip shower key	CHIPSW	Common	Indicates that the **[CHP.F]** key is ON.	S10
BX138 to BX139	Pallet select key	QTSEL1 QTSEL2	Common	Indicates the status of the **<**PALLET**>** key. **<PALLET> key** / **QTSEL1** / **QTSEL2** OFF / OFF / OFF 1 / ON / OFF 2 / OFF / ON 1↔2 / ON / ON	S10
BX13A	Automatic door open switch	ADOPSW	Common	Turns ON while the automatic door option **[AUTOMATIC DOOR OPEN]** switch is depressed. Turns OFF when not pressed.	S10
BX13B	Automatic door close switch	ADCLSW	Common	Turns ON while the automatic door option **[AUTOMATIC DOOR CLOSE]** switch is depressed. Turns OFF when not pressed.	S10
BX13C	Rapid traverse override 100%	RPD100	Common	Indicates the status of the rapid traverse override. ON: 100% OFF: Not 100%	
BX13D	Cutting feedrate override 100%	FED100	Common	Indicates the status of the cutting override. ON: 100% OFF: Not 100%	
BX13E	Spindle override 100%	SPN100	Common	Indicates the status of the spindle override. ON: 100% OFF: Not 100%	
BX13F	Automatic door MANU-AUTO select switch	ATDRSL	Common	Turns OFF when the automatic door option "MANU" is selected from the **[MANU-AUTO]** select switch. It turns ON when "AUTO" is selected. (NOTE) The logic and signals with the same name for models prior to CNCB00 are reversed.	S10
BX140	Manual operation mode	MANSTL	Common	Turns ON when operating (also including restoration operation) in manual operation mode. Turns OFF after the operation ends.	
BX141	Handle OFF	PULOFF	Common	Turns ON when the axis operation is not available on the handle.	
BX142	High speed feed check	HIGHSPD	Common	Turns ON when the **[RPD]** key is selected. Turns OFF when the **[JOG]** key and the **[STEP]** key are selected.	
BX143	Low speed feed check	LOWSPD	Common	Turns ON when the **[JOG]** key is selected. Turns OFF when the **[RPD]** key and the **[STEP]** key are selected.	
BX144	Step feed check	STEPSPD	Common	Turns ON when the **[STEP]** key is selected. Turns OFF when the **[RPD]** key and the **[JOG]** key are selected.	
BX145	Pendant enable SW	TPVLDSW	Common	Signal indicates the status of the **[ENABLE PENDANT]** switch for the teaching pendant. ON: Teaching pendant enabled OFF: Teaching pendant disabled	

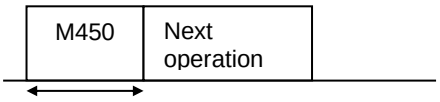
OM name	Name	Symbol	Conversation / NC	Signal description	Use
BX146	Chip auger **[AUTO /MANUAL]** switch	COILSW	Common	The signal turns ON when the **[AUTO /MANUAL]** switch for the coil conveyor is on “AUTO”. It turns OFF when the switch is on “MANUAL”.	S10
BX147	Chip auger **[FORWARD/REVERSE]** switch – forward	COILFWSW	Common	When the **[AUTO /MANUAL]** switch for the coil conveyor is on “MANUAL”, and “FORWARD” is selected on the **[FORWARD/REVERSE]** switch, the signal turns ON while the switch is pressed. The signal turns OFF when the switch is not pressed.	S10
BX148	Chip auger **[FORWARD/REVERSE]** switch - reverse	COILRVSW	Common	When the **[AUTO /MANUAL]** switch for the coil conveyor is on “MANUAL”, and “REVERSE” is selected on the **[FORWARD/REVERSE]** switch, the signal turns ON while the switch is pressed. The signal turns OFF when the switch is not pressed.	S10
BX149	Reserved	RESERVED	Common	This signal is reserved.	
BX14A	Touch sensor 1	TOUCH1_ON	Common	The signal turns ON when one of the bits for the touch sensor is ON. (NOTE) There is no high speed response for the signal.	
BX14B	Touch sensor 2	TOUCH2_ON	Common	The signal turns ON when one of the bits for the touch sensor is ON. (NOTE) There is no high speed response for the signal.	
BX14C	Touch sensor 3	TOUCH3_ON	Common	The signal turns ON when one of the bits for the touch sensor is ON. (NOTE) There is no high speed response for the signal.	
BX14E	Table light	HLIGHT_ON	Common	Signal indicates the status of the **<**TABLE LIGHT**>** key. The ON and OFF status switches back and forth when the **<**TABLE LIGHT**>** key is pressed.	
BX14F	Machine lights	LIGHT_ON	Common	Indicates the status of the **<**LIGHT**>** key. The ON status and OFF status switch back and forth when the **<**LIGHT**>** key is pressed. In addition, after the light is turned ON, the light turns OFF when the user parameter 1 (switch 1: common) <Automatic machine light off time> has elapsed.	S10

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BX150	Memory operation ready	MEMOK	Common	This signal turns ON when memory operation mode is enabled and is ready to start memory operation. However, it turns OFF in the following situations. • When **[SINGL]**, **[DRY]** and **[M.LCK]** keys are ON • While reset is processing • When a door is open • When the user parameter (switch 1: door) <Door lock method> is set to <0: Automatic lock when door closes> and the door is unlocked • When one of the following alarms or operator messages is triggered Alarms with a stop level 2 or higher <<Relocation was detected>> <<Reset the relocation detection status>> <<Execute the automatic oiling/greasing cycle ON command>> <<Brake load test error on *-axis>> • Travel screen is displayed for start position at program restart • <Door interlock mode> is not set to <Auto. operation> • User parameter (switch 1: operation) <Check before chip shower start> is set to <1: Enable> and the **[CHP.F]** key is OFF • User parameter (switch 1: operation) <Check before coolant pump start> is set to <1: Enable> and the **[CLT.P]** key is OFF	
BX152	Load completed (Selection end)	LOADOK	Common	It turns OFF when program is selected, and it turns ON once the program has been read.	
BX153	Automatically operating	AUTO	Common	Turns ON / OFF during the following conditions when operating in MDI or memory operation mode. ON condition → Initial startup input OFF conditions 1. Reset 2. Stop level 5 (servo OFF) alarm (NOTE) It does not turn ON during an external machine zero return operation.	
BX154	Automatic operation starting up	STL	Common	Turns ON / OFF during the following conditions when operating in MDI or memory operation mode. ON condition → Operation executing OFF conditions 1. End of operation 2. Single operation stopped 3. Stop 4. Reset 5. Alarm status (NOTE) It does not turn ON during an external machine zero return operation.	S1 S10
BX155	Extended memory operation	EXPRUN	Common	Turns ON when the program that is displayed in memory operation mode is for an extended operation. If the program size exceeds 32 MB, it becomes an extended memory operation.	
BX156	Operation stop error	RUNSTP	Common	Turns ON when an alarm is triggered that stops operation. It turns OFF when the alarm is cleared and reset.	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BX157	Automatically operating (Including stop status)	STLHOLD	Common	Turns ON / OFF during the following conditions when operating in MDI or memory operation mode. ON condition – Operation executing OFF condition – 1. Operation finished 2. Reset (NOTE) It does not turn ON during an external machine zero return operation.	S10
BX158	Rapid feeding	FEDMOV	Common	The signal turns ON when the axis travels using rapid feed. For rapid feed during automatic operation, it does not just include the travel command issued by the G00 command, but it also includes the positioning operation during a canned cycle and a reference return operation (G28 command), etc. (NOTE 1) It turns ON and OFF even when the machine lock is engaged. (NOTE 2) It does not output during manual operation.	
BX159	Cutting feed performing	CUTMOV	Common	The signal turns ON when the axis travels using the cutting feed operation. (NOTE 1) It turns ON and OFF even when the machine lock is engaged. (NOTE 2) It does not output during manual operation.	
BX15A	Skipping	SKPMOV	Common	The signal turns ON when the skip command is executed. (NOTE) The signal does not turn ON during machine lock.	
BX15B	Startup ready	STRTOK	Common	Turns ON when the start signal can be received.	S10
BX15C	Pallet start ready	OTSTOK	Common	Turns ON when the pallet start signal can be received.	S10
BX15D	Restart	RESTAT	Common	Indicates that it is restarting.	S10
BX15E	Automatic operation start check	STCON	Common	Turns ON when operation is started (including restart also) in MDI or memory operation mode. After turning ON, it turns OFF after 100 msec. has elapsed.	S10
BX15F	Automatically operating (not including operation reset)	STLHOLD2		MDI operation or memory operation mode: ON condition – Operation executing OFF condition – 1. Motion end 2. Reset 3. Operation reset (NOTE) The signal does not turn ON during an external machine zero return operation.	S10
BX160	Deep hole machining cycle executing	DPHLON	Common	For the following deep hole drilling cycles, the signal turns ON for R-point, and the signal turns OFF for the return point. • High-speed deep hole drilling cycle (G73) • Deep hole drilling cycle (G83) • High-speed deep hole drilling cycle (G173) • Deep hole drilling cycle (G183) The signal turns OFF for the Z-point when a command without a return point is issued. In addition, the signal turns OFF when the cycle is cancelled using the reset key. The signal stays ON during a temporary stop in a deep hole cycle, a relief operation and dwell operation.	
BX161	High accuracy mode A operating	HIACU_A	Common	Turns ON when high accuracy mode A is valid. Turns OFF when the mode is invalid.	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BX162	High accuracy mode B operating	HIACU_B	Common	Turns ON when high accuracy mode B is valid. Turns OFF when the mode is invalid.	
BX163	G95 modal	G95MDL	Common	Turns ON / OFF in the program (G95 / G94/ G93) when operating in MDI or memory operation mode. G95: ON G94,G93: OFF	
BX170 to BX172	Mirror imaging X-axis Y-axis Z-axis	 MIRORX MIRORY MIRORZ	NC	Turns ON when the control axis is performing mirror imaging. It turns OFF when the mirror image operation is cancelled.	
BX173 to BX175	Scaling X-axis Y-axis Z-axis	 SCLINGX SCLINGY SCLINGZ	NC	Turns ON when the control axis is scaling. It turns OFF when the scaling operation is cancelled.	
BX180	Program stop	M00	Common	Turns ON when the program stop is executed during memory operation. It turns OFF with the start signal and reset. (NOTE) It does not change when an alarm is triggered.	S10
BX181	Optional stop	M01	NC	Turns ON when the optional stop is executed during memory operation. It turns OFF using the start signal and reset. (NOTE) It does not change during an alarm.	
BX182 to BX183	Program end Program end 2	M30/1 M30/2	Common	Turns ON when the program end is executed during memory operation. • M30/1: Turns ON after the program ends for pallet 1. • M30/2: Turns ON after the program ends for pallet 2. It turns OFF with the start signal and reset. (Reference) Refer to “Program end M30/1 and M30/2” in “Chapter 1 – 1.6.5.2 Detailed description of external output signals” in the Data Bank & Alarm Manual for further details about the program end. (NOTE) Only M30/1 is output for machines without QT. M30/1 is output regardless of the pallet for machines equipped with QT when the **<**PALLET**>** key is turned <OFF>.	S1 S10

<table>
<tr><th>OM name</th><th>Name</th><th>Symbol</th><th>Conversation / NC</th><th>Signal description</th><th>Use</th></tr>
<tr><td>BX188
to
BX18C
BX1A4
to
BX1A7</td><td>External signal output</td><td>M400
M402
M404
M406
M408
M480
M482
M484
M486</td><td>Common</td><td>Turns ON / OFF in the program when operating in MDI or memory operation mode.
Ex) In the program, it turns ON with output signal 400, and it turns OFF with signal output 401.
It is different from M11 - M28 and does not check the M function end signal (MFIN).
<table>
<tr><th>Signal name</th><th>ON</th><th>OFF</th></tr>
<tr><td>M400</td><td>Issues command M400 in program</td><td>Issues command M401 in program</td></tr>
<tr><td>M402</td><td>Issues command M402 in program</td><td>Issues command M403 in program</td></tr>
<tr><td>M404</td><td>Issues command M404 in program</td><td>Issues command M405 in program</td></tr>
<tr><td>M406</td><td>Issues command M406 in program</td><td>Issues command M407 in program</td></tr>
<tr><td>M408</td><td>Issues command M408 in program</td><td>Issues command M409 in program</td></tr>
<tr><td>M480</td><td>Issues command M480 in program</td><td>Issues command M481 in program</td></tr>
<tr><td>M482</td><td>Issues command M482 in program</td><td>Issues command M483 in program</td></tr>
<tr><td>M484</td><td>Issues command M484 in program</td><td>Issues command M485 in program</td></tr>
<tr><td>M486</td><td>Issues command M486 in program</td><td>Issues command M487 in program</td></tr>
</table>
(NOTE 1) It turns OFF when the reset signal is input.
(NOTE 2) It does not turn OFF during an alarm.</td><td>S10</td></tr>
<tr><td>BX18D</td><td>Jig shower</td><td>M418</td><td>Common</td><td>Turns ON / OFF in the program (M418 / M419) when operating in MDI or memory operation mode.
M418: ON
M419: OFF
(NOTE 1) It turns OFF when the reset signal is input.
(NOTE 2) It does not turn OFF during an alarm.</td><td>S10</td></tr>
<tr><td>BX18E</td><td>Spindle air blow</td><td>AIRBLW_ON</td><td>Common</td><td>Signal turns ON when equipped with an arm ATC and it rotates if the machine parameter (system 1: common) <Spindle air blow /tool wash> is set to <0: Spindle air blow type 1> or to <2: Spindle air blow type 2>.
Signal turns OFF when the arm rotation is complete.
When using the turret ATC:
Turns ON when the Z-axis is in the ATC range.
Turns OFF when the Z-axis is outside the ATC range.
In addition, it turns ON / OFF in the program (M498 / M499) when operating in MDI operation.
M498: ON
M499: OFF</td><td>S10</td></tr>
<tr><td>BX18F</td><td>Tool wash filter clogging is predicted.</td><td>TWFLT_PRE</td><td>Common</td><td>Signal turns ON when an error status continues after a given number of checks, set in the user parameter (switch 1: installation) <No. of tool wash filter clogging warning checks (Type 4)>, for the filter clogging check during tool washing.</td><td></td></tr>
</table>

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BX190 to BX197	M function signal output	M11 M12 M14 M18 M21 M22 M24 M28	Common	Outputs the numerical values for the last two digits in BCD code to the PLC, by issuing one of the commands in the signal output range between 800 and 899 using MDI operation mode or memory operation mode. It has the same properties as M00 through M99, MF and MFIN, which are noted in “M function signal output” from the “1.6.5 External output signal” found in the Data Bank & Alarm Manual. However, it is part of another separate signal group.	
BX19F	M output latch	MF	Common	Signals 800 to 899 are output, and the signal MF is output after the <M function signal output delay time> set in the user parameter (switch 1: programming) has elapsed. Read the M function signal output indicated by this signal MF.	
BX1A0 to BX1A3	One shot output	450 451 455 456	Common	The one shot signal is output. The output time is set in the user parameter (switch 1: programming) <One shot signal M450 (M451) output time>.  One shot signal M450 output time 1. It turns OFF with the reset signal input. 2. It does not turn OFF during an alarm.	
BX1A8	M08 modal	M08MDL	Common	Turns ON / OFF in the program (M08 / M09) when operating in MDI or memory operation mode. M08: ON M09: OFF	S10
BX1A9	Interlock during M08ATC	M08ATCIL	Common	Turns ON in the following situations when M08MDL is ON. • MDI operation when the <Coolant during tool change> is set to <0: Stop> or tool changing using the memory operation • Performing tool breakage detection (Type 3 only) when the <Tool breakage detection coolant OFF> is set to <1: YES> in the Conversation language	S10
BX1AA	Interlock during M08 alarm	M08ALIL	Common	Turns ON in the following situation when M08MDL is ON. • When the servo OFF alarm occurs Turns OFF in the following situations: • When the operation starts up in MDI operation or memory operation mode • When it resets • When the operation ends	S10
BX1AB	Drain command	M479	Common	Turns ON for 100 ms after drain cycle ON (M479) command is issued.	S10
BX1AC	M494 modal	M494MDL	Common	Turns ON / OFF in the program (M494 / M495) when operating in MDI or memory operation mode. M494: ON M495: OFF	S10

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BX1AD	M494 interlock	M494IL	Common	Turns ON in the following situations when M494MDL is ON. • When the servo OFF alarm occurs • When the GOFF No. 3 System A/B signal is OFF • When the coolant overheat alarm occurs • When an ATC operation is performed (including moving to the tool installation position on the turret ATC) Turns OFF in the following situations: • When the operation starts up in MDI operation or memory operation mode • When it resets • When the operation ends	S10
BX1AE	Back wash command	CTSBW_ON	Common	Turns ON for 100 ms after back wash cycle ON (M496/ M489) command is issued.	S10
BX1AF	Tool cleaning ON	TW_ON	Common	Turns ON when the tool cleaning is performed.	S10
BX1B0	Workpiece coordinate zero writing error	WPWRT_ER	NC	Signal turns ON when writing is not possible for the workpiece coordinate zero. The signal changes together with BDX182. Signal turns OFF when the writing operation starts from the BY1FF signal.	
BX1B8	Character string writing is restricted	STRCPY_LCK	Common	While this signal is ON, the character string copy command is not possible.	
BX1B9	Character string copied	STRCPY_FIN	Common	This signal turns ON when the BY0DA: character string copy command is complete. When the BY0DA: character string copy command turns OFF, this signal also turns OFF.	
BX1B1	Writing is restricted for workpiece coordinate zero	WPWRTLOCK	NC	Signal turns ON when writing for the workpiece coordinate zero (from signal BY1FF) is restricted.	
BX1C0	Tool life warning	TOLPRE	Common	Turns ON when the tool life being used for operation reaches the life warning. The timing for turning ON is the same as the TOOL signal.	
BX1C1	Tool breakage error	TLBRKE	Common	Turns ON when the alarm <<Tool breakage error>> is triggered and turns OFF when the alarm is reset.	
BX1C2	Managing tool life	TLLFCK	Common	Turns ON if the tool life is set on a spindle tool that is mounted. It turns OFF when one of the following situations applies. • When the tool life is not set • When another language (NC/conversation) is used • When no tools are mounted	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BX1C3	Tool error output	TOOL	Common	Turns ON when the alarm <<Tool breakage error>> is triggered and turns OFF when the alarm is reset. Turns ON when the alarm <<Tool breakage error>> is triggered after the tool breakage is detected, or when the program end is executed after the tool life ends for the tool being used in the executing program during memory operation. (However, it does not turn ON when the spare tool is registered.) The following 2 ways can be used to cancel this signal and restart. (1) Register the spare tool and press the **[RST]** key. (2) Reset the tool life and press the **[RST]** key. (NOTE 1) Press the **[RST]** key without performing the above operations to turn OFF the signal temporarily. In addition, when M120 (M121) is executed and the measurement signal (TOUCH) is OFF (ON), the alarm <<Tool breakage error>> is triggered and the TOOL signal turns ON. (NOTE 2) The operations noted above apply after the alarm <<Tool breakage error>> is triggered, regardless of the user parameter (tool breakage detection) setting for <Tool breakage detection option>.	
BX1C4	During ATC	TLCHG	Common	Turns ON during an ATC operation. Turns OFF once the ATC operation ends. In addition, since the tool change is performed during machine operation when using a turret ATC configuration, this signal turns ON / OFF even during machine operation. This signal turns ON/OFF even when the ATC monitoring parameter is automatically set. Furthermore, it turns ON during the tool change operation (when the LED for the **[ATC]** key is lit, or when the LED for the **[MAG.R]** and the **[MAG.F]** keys are lit) when using manual operation mode. It turns OFF after the operation ends (LED turns off). The automatic setting operation for the ATC monitoring parameter is turned ON after **[RELSE]** + **[M]** are pressed, and it turns OFF after the operation is finished (stopped). When using the tool change command G100/M6, it turns ON at the start of the operation, and it turns OFF after the tool change operation ends. This signal turns OFF if a stop level 5 alarm occurs. It stops at the R-point and the signal stays ON when a stop level 4 alarm occurs during a tool change operation. It turns ON even for the same tool change operation. However, it does not turn ON when the machine lock is engaged. This signal turns ON when the tool change operation is performed using the index command that is issued to installation or removal position in the Conversation language. (Only for turret configuration)	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BX1C5	New tool selection	NEWTL	Common	Turns ON when the tool life value for the spindle tool is the default value (tool life count has not started) during the tool change operation. Turns OFF after 100 msec. has elapsed.	
BX1C6	Machining load maximum	TLDHI	Common	Turns ON when the machining load monitor function detects the alarm <<Machining load maximum>>. It turns OFF when the alarm is not detected or when the alarm is reset.	
BX1C7	Machining load minimum	TLDLO	Common	Turns ON when the machining load monitor function detects the alarm <<Machining load minimum>>. It turns OFF when the alarm is not detected or when the alarm is reset.	
BX1C8	Machining load monitor ON	TLDON	Common	Turns ON when the machining load monitor function is currently monitoring the machining load. It turns OFF when the function is not monitoring or when the monitoring has ended.	
BX1CC	No tool detection (when ATC rises)	TLNONE_U	Common	The signal turns ON when there is no tool at tool unclamping during the ATC operation (when ATC rises). The signal turns OFF when operation is reset or for the next ATC operation. Signal is always OFF when the user parameter (switch 1) <No tool detection function> is set to <0:Invalid>.	
BX1CD	No tool detection (when ATC falls)	TLNONE_D	Common	The signal turns ON when there is no tool at tool unclamping during the ATC operation (when ATC falls). The signal turns OFF when operation is reset or for the next ATC operation. Signal is always OFF when the user parameter (switch 1) <No tool detection function> is set to <0:Invalid>.	
BX1CE	Tool over key detection	TLKYRIDE	Common	The signal turns ON when a tool is mounted over the spindle drive key at tool clamping during the ATC operation. The signal turns OFF when operation is reset or for the next ATC operation. The signal is always OFF when the user parameter (switch 1) <Tool over key detection function> is set to <0: Disable>.	
BX1CF	Tool missing detection	TLFALLOUT	Common	The signal turns ON if a tool is missing or is about to come off when the Z-axis rises. One second after the signal turns ON, it turns OFF. The signal is always OFF when the user parameter (switch 1) <Tool missing detection function> is set to <0: Disable>.	
BX1D0	Workpiece counter end	CNTUP	Common	Turns ON when the workpiece counter has ended. It turns OFF when the end status is reset.	S1
BX1D1	Workpiece counter end approaching setting	CNTPRE	Common	Turns ON when the end warning is issued on workpiece counter. It turns OFF when the end warning status is reset.	
BX1D2	Wave form display notice	WAVDSP	Common	Turns OFF when the waveform display function has started. It turns ON when the trigger conditions are met for the waveform display notification signal.	
BX1D3	Maintenance notice	MNTALM	Common	Turns ON when the maintenance notice is in an alarm form. It turns OFF in all other situations.	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BX1D4	External input signal assignment error	IOASGN_ER	Common	Turns ON when the same signal name for the external input signal and the field bus input signal is repeated and assigned. It turns OFF in all other situations.	
BX1D5	Spindle test run execute request	SPIALM	Common	Turns ON when the spindle test run needs to be executed. It turns OFF in all other situations.	
BX1DD	Teaching communication connection	TPCNNCT	Common	Signal turns ON when the communication with the teaching pendant is established (connected). It turns OFF in all other situations.	
BX1DE	External panel 1 Communication established	GP1CNNCT	Common	The signal turns ON when the communication with external panel 1 is established. It turns OFF in all other situations.	
BX1DF	External panel 2 Communication established	GP2CNNCT	Common	The signal turns ON when the communication with external panel 2 is established. It turns OFF in all other situations.	
BX1E0	Current language mode	LANGNC	Common	Turns ON when the <Conversation/NC language change> item in the user parameter (switch 1) is set to <1: NC>. It turns OFF when the parameter is set to <0: Conversation>.	
BX1E1	Automatic door Area sensor pattern	ATDRSP	Common	Signal turns ON when the user parameter (switch 1: door) <Area sensor pattern> is set to <1: Pattern 2 (Valid when door is operating)>. It turns OFF when the parameter is set to <0: Pattern 1 (Valid during door close operation)>.	
BX1E2	Automatic door valid	ATDRON	Common	Signal turns ON when the machine parameter (system 1: automatic door) <Option> is set to <1: Yes>. It turns OFF when the parameter is set to <0: OFF>.	S10
BX1E3	CT option M	CTSMDM	Common	Signal turns ON when the machine parameter (system 1: common) <Center-through-coolant option> is set to <1:Type1> Turns OFF for all other settings.	S10
BX1E4	CT option H	CTSMDH	Common	Signal turns ON when the machine parameter (system 1: common) <Center-through-coolant option> is set to <2:Type2> Turns OFF for all other settings.	S10
BX1E5	Indicator selection yellow when count ends	CNTUPYEL	Common	Signal turns ON when the user parameter (switch 1: installation) <Lamp status when count ends> is set to <1:Yellow>. It turns OFF for all other settings.	S1
BX1E6	Indicator selection red when count ends	CNTUPRED	Common	Signal turns ON when the user parameter (switch 1: installation) <Lamp status when count ends> is set to <2:Red>. It turns OFF for all other settings.	S1
BX1E8	Startup lamp turns ON during automatic door manual operation	DR_MANLED	Common	Signal turns ON when the user parameter (switch 1: door) <START lamp status when operating automatic door manually> is set to <1:On>. It turns OFF for all other settings.	S10
BX1E9	Machine unit system setting	MCN_UNIT	Common	The signal is the set value in the user parameter (switch 1: system) <Machine unit system>. When set to <0: Metric>, the signal is OFF. When set to <1: Inch>, the signal is ON.	
BX1EA	Minimum setting unit	MIN_UNIT	Common	Signal turns ON/OFF based on the minimum unit setting that is set on the machine. When set to Type 1, the signal is OFF. When set to Type 2, the signal is ON.	
BX1EB	Chip conveyor relay adhering check	SCRWRL_CHK	Common	Signal turns ON when the user parameter (switch 1: installation) <Chip conveyor relay adhering check> is set to <1: Yes>. It turns OFF when the parameter is set to <0: OFF>.	S10

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BX1EC	Time extension when coolant level has dropped	CLLOW_TIM	Common	Signal turns ON when the user parameter (switch 1: installation) <Time extension when coolant level has dropped> is set to <1: Enable>. It turns OFF when the parameter is set to <0: Disable>.	S10
BX1EE	Cyclone filter	CYCLN_FLT	Common	Signal turns ON when the machine parameter (system 1: common) <Cyclone filter> is set to <1: Yes>. The signal turns OFF when the parameter is set to <0: Not equipped>.	S10
BX1EF	Coil conveyor	COILCON	Common	Signal turns ON when the machine parameter (system 1: common) <Chip auger> is set to <1: Yes>. The signal turns OFF when the parameter is set to <0: Not equipped>.	S10
BX1F0	Auto notification sending	SND_TRAN	Common	Turns ON when the auto notification is sending. It is normally OFF when the communication parameter (auto notification function) <Control method> is set to <1: Simple>.	
BX1F1	Receive auto notification response	SND_RET	Common	Turns ON when the auto notification is sent and a response is received. This signal is turned off when the SND-REQ is turned OFF. It is normally OFF when the communication parameter (auto notification function) <Control method> is set to <1: Simple>.	
BX200 to BX27F	Macro variables	#1100 : #1131 #1300 : #1395	Common	Outputs the value for the macro variables 1100~#1131 and #1300~#1395. It output OFF when <0> and ON when <1>.	
BX280	<u>CTS</u> CTS pump overheat	CTSPMPOH	Common	<u>CTS</u> CTS pump normal: ON CTS pump error: OFF	S10
BX281	<u>CTS</u> CTS pressure SW	CTSPSON	Common	<u>CTS</u> CTS pressure drop: ON CTS pressure rise: OFF	S10
BX282	Front low	FL_LOW	Common	This signal is reserved. Therefore, do not use it.	S10
BX283	Front empty	FL_EMP	Common	This signal is reserved. Therefore, do not use it.	
BX284	Float full	FL_FLL	Common	This signal is reserved. Therefore, do not use it.	
BX285	<u>High pressure</u> High-pressure, supply breaker ON OK <u>Cyclone</u> Suction pump overheat	QHON_OK	Common	<u>High pressure</u> This signal is reserved. Therefore, do not use it. <u>Cyclone</u> Suction pump normal: 0 Suction pump error: 1	S10
BX287	Back wash filter clogged	BWFER	Common	No clog: OFF Clogged: ON	S10
BX289	Line filter clogged	LFER	Common	No clog: OFF Clogged: ON	S10
BX28B	Tool washing liquid surface sensor OK	TCLS_OK	Common	Fluid level undetected: OFF Fluid level detected: ON	S10
BX28C	Tool wash filter clogged	TCF_ER	Common	No clog: OFF Clogged: ON	S10
BX290	Automatic door READY	DRRDY	Common	Automatic door not ready: OFF Automatic door ready: ON	
BX291	Automatic door open operation complete	DROFIN	Common	Automatic door open operation not complete: OFF Automatic door open operation complete: ON	S10
BX292	Automatic door close operation complete	DRCFIN	Common	Automatic door close operation not complete: OFF Automatic door close operation complete: ON	S10

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BX293	Automatic door alarm	DRALM	Common	Automatic door alarm not triggered: OFF Automatic door alarm triggered: ON	
BX294	Automatic door stop detection	DRHOLD	Common	No detection of automatic door stop: 0 Detection of automatic door stop: 1	
BX295	*Area sensor input	ARESN_N	Common	Area sensor obstructed: OFF Area sensor not obstructed: ON	
BX297	Door open	DROPEN	Common	No detection of automatic door open position: 0 Detection of automatic door open position: 1	S10
BX298	Door open timeout	DRO_TOV_ERR	Common	Signal turns ON during the alarm <<Door open timeout>>. The stop level changes for the alarm <<Door open timeout>> depending on the user parameter (switch 1: door) setting <Automatic door error level>.	
BX299	Automatic door operating automatically	ATDRMV	Common	Signal turns ON during the open / close operation while the automatic door is in automatic mode.	S10
BX29A	Door operation timeout	DR_TOV_ERR	Common	Signal turns ON during the alarm <<Door operation timeout>>. The stop level changes for the alarm <<Door operation timeout>> depending on the user parameter (switch 1: door) setting <Automatic door error level>.	
BX29B	Close the automatic door	DR_CLODR_ERR	Common	Signal turns ON during the alarm <<Close the automatic door.>>.	
BX29C	Automatic door stopped	ATDRSTP	Common	Signal turns ON when the automatic door is stopped. Signal turns OFF when the machine parameter (system: door) <Option> is set to <0: No>.	S10
BX2A0	Pallet start	EXSTART	Common	Start SW OFF: 0 Start SW ON: 1	
BX2A1	External pallet start	EXOUTST	Common	Pallet start SW OFF: 0 Pallet start SW ON: 1	
BX2B0	*Automatic lubrication_empty	OILEMP_N	Common	Oil empty: 0 Oil full: 1	
BX2B1	Automatic lubrication_pressure SW	PRSNSR	Common	Displays the status of the pressure sensor when lubricating with the automatic oiling device. Pressure stop: 0 Pressure rise: 1	
BX2B8	Chip auger thermal error	IO_CO_CH	Common	No thermal error: 0 Thermal error: 1	S10
BX2BC	Spare input 5	SPIN5	Common	This signal is reserved. Therefore, do not use it.	
BX2BD	Spare input 6	SPIN6	Common	This signal is reserved. Therefore, do not use it.	
BX2BE	Spare input 7	SPIN7	Common	This signal is reserved. Therefore, do not use it.	
BX2BF	Reserved input 8	SPIN8	Common	This signal is reserved. Therefore, do not use it.	
BX2C0 to BX2C3	Reserved	RESERVED	Common	This signal is reserved. Therefore, do not use it.	

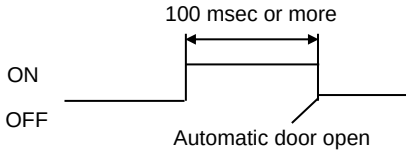
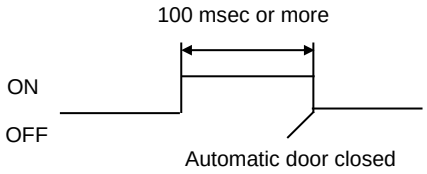
<table>
<tr><th>OM name</th><th>Name</th><th>Symbol</th><th>Conversation / NC</th><th>Signal description</th><th>Use</th></tr>
<tr><td>BX2D0
to
BX2DA</td><td>GOFF No.1 System A
GOFF No.2 System A
GOFF No.3 System A
GOFF No.4 System A
GOFF No.5 System A
GOFF No.6 System A
GOFF No.7 System A
GOFF No.8 System A
GOFF No.9 System A
GOFF No.10 System A
GOFF No.11 System A</td><td>IL_GOFA1
IL_GOFA2
IL_GOFA3
IL_GOFA4
IL_GOFA5
IL_GOFA6
IL_GOFA7
IL_GOFA8
IL_GOFA9
IL_GOFA10
IL_GOFA11</td><td>Common</td><td>Signal displays the status of the hardware gate.
Gate OFF: 0
<table>
<tr><th>System</th><th>Non-R system model</th><th>R system model</th></tr>
<tr><td>No.1</td><td>X-, Y-, Z-, ATC- and additional axes and magazine shutter</td><td>X-, Y-, Z-, ATC- and QT-axes</td></tr>
<tr><td>No.2</td><td colspan="2">M-axis</td></tr>
<tr><td>No.3</td><td colspan="2">Spindle and CTS</td></tr>
<tr><td>No.4</td><td>Lathe spindle</td><td>Coil conveyor (linked with machining chamber)</td></tr>
<tr><td>No.5</td><td>Same as No.1 system</td><td>Additional axis (pallet 1)</td></tr>
<tr><td>No.6</td><td>None</td><td>Additional axis (pallet 2)</td></tr>
<tr><td>No.7</td><td colspan="2">Coil conveyor (linked with setup chamber)</td></tr>
<tr><td>No.8</td><td colspan="2">Chip conveyor</td></tr>
<tr><td>No.9</td><td colspan="2">PLC-axis (outside 1)</td></tr>
<tr><td>No.10</td><td colspan="2">PLC-axis (outside 2)</td></tr>
<tr><td>No.11</td><td colspan="2">Reserved</td></tr>
</table>
Gate ON: 1</td><td>S10</td></tr>
<tr><td>BX2E0
to
BX2EA</td><td>GOFF No.1 System B
GOFF No.2 System B
GOFF No.3 System B
GOFF No.4 System B
GOFF No.5 System B
GOFF No.6 System B
GOFF No.7 System B
GOFF No.8 System B
GOFF No.9 System B
GOFF No.10 System B
GOFF No.11 System B</td><td>IL_GOFB1
IL_GOFB2
IL_GOFB3
IL_GOFB4
IL_GOFB5
IL_GOFB6
IL_GOFB7
IL_GOFB8
IL_GOFB9
IL_GOFB10
IL_GOFB11</td><td>Common</td><td>Signal displays the status of the hardware gate.
Gate OFF: 0
Gate ON: 1
Refer to “BX2D0 to BX2DA” for further details on each system.</td><td>S10</td></tr>
<tr><td>BX2F0
to
BX2FA</td><td>EDM No.1 System
EDM No.2 System
EDM No.3 System
EDM No.4 System
EDM No.5 System
EDM No.6 System
EDM No.7 System
EDM No.8 System
EDM No.9 System
EDM No.10 System
EDM No.11 System</td><td>IL_EDM1
IL_EDM2
IL_EDM3
IL_EDM4
IL_EDM5
IL_EDM6
IL_EDM7
IL_EDM8
IL_EDM9
IL_EDM10
IL_EDM11</td><td>Common</td><td>This is the gate OFF back-check signal.
This signal turns ON when the GOFF signal A and B for each system are both 0. This signal turns OFF for all other situations.
Refer to “BX2D0 to BX2DA” for further details on each system.</td><td>S10</td></tr>
<tr><td>BX2FD</td><td>Chip conveyor error</td><td>IO_CHPC_ER</td><td>Common</td><td>Indicates the error status on the chip conveyor.
Chip conveyor error not triggered: 0
Chip conveyor error triggered: 1</td><td>S10</td></tr>
<tr><td>BX2FE</td><td>Chip conveyor coolant empty</td><td>IO_CHPC_EMP</td><td>Common</td><td>Indicates whether chip conveyor coolant is empty or not. Normal: 0
Coolant empty: 1</td><td>S10</td></tr>
<tr><td>BX2FF</td><td>Chip conveyor manual mode</td><td>IO_CHPC_MNL</td><td>Common</td><td>Signal indicates the status of the [MANUAL/STOP/AUTO] switch on the chip conveyor.
Chip conveyor automatic mode: 0
Another chip conveyor mode: 1</td><td>S10</td></tr>
<tr><td>BX300
to
BX304</td><td>Reserved</td><td>RESERVED</td><td>Common</td><td>This signal is reserved. Therefore, do not use it.</td><td></td></tr>
</table>

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BX305	Door lock power overcurrent 1	SR_PWDLKCK1	Common	Short-circuits are monitored below for the door lock signal system and the operation preparation alarm signal system. Front door, left side door, right side door and magazine chamber door No short-circuit: 0 Short-circuit: 1	
BX306	Door lock power overcurrent 2	SR_PWDLKCK2	Common	Short-circuits are monitored below for the door lock signal system. Outer door, setup chamber left door, setup chamber right door, outside door 1 and outside door 2 No short-circuit: 0 Short-circuit: 1	
BX307	SR brake overcurrent 1	SR_BKSHORT1	Common	Short-circuits are monitored for the Z-axis brake release. No short-circuit: 0 Short-circuit: 1	
BX308	SR brake overcurrent 2	SR_BKSHORT2	Common	Short-circuits are monitored for the magazine axis brake release. No short-circuit: 0 Short-circuit: 1	
BX309	SR brake overcurrent 3	SR_BKSHORT3	Common	Short-circuits are monitored for the additional axis and PLC-axis brake release. No short-circuit: 0 Short-circuit: 1	
BX30A	SR brake overcurrent 4	SR_BKSHORT4	Common	Short-circuits are monitored for the additional axis and PLC-axis brake release. No short-circuit: 0 Short-circuit: 1	
BX30B	SR brake overcurrent 5	SR_BKSHORT5	Common	hort-circuits are monitored for the additional axis and PLC-axis brake release.	
BX30C	Reserved	RESERVED	Common	This signal is reserved. Therefore, do not use it.	
BX30D	IL 24 V overcurrent 1	IL24VERR_OC1	Common	This signal is reserved. Therefore, do not use it.	
BX30E	IL 24 V overcurrent 2	IL24VERR_OC2	Common	This signal is reserved. Therefore, do not use it.	
BX30F	Reserved	RESERVED	Common	This signal is reserved. Therefore, do not use it.	
BX310	Brake release - System 1	SR_BKIN1	Common	Indicates the brake status for the Z-axis motor. Brake ON: 0 Brake release: 1	
BX311	Brake release - System 2	SR_BKIN2	Common	<u>For arm type ATC machine model</u> Indicates the brake status for the Y-axis motor. Brake ON: 0 Brake release: 1 <u>For all other machine models</u> Indicates the brake status for the magazine axis motor. Brake ON: 0 Brake release: 1	
BX312	Brake release - System 3	SR_BKIN3	Common	Signal indicates the brake status for the additional axis/PLC-axis motor. Brake ON: 0 Brake release: 1	
BX313	Brake release - System 4	SR_BKIN4	Common	<u>For machine models with a lathe function</u> Indicates the brake status for the 5th-axis motor. Brake ON: 0 Brake release: 1 <u>For all other machine models</u> Signal indicates the brake status for the additional axis/PLC-axis motor. Brake ON: 0 Brake release: 1	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BX314	Brake release - System 5	SR_BKIN5	Common	Signal indicates the brake status for the additional axis/PLC-axis motor. Brake ON: 0 Brake release: 1	
BX320	Reserved	RESERVED	Common	This signal is reserved.	
BX32E	Chip conveyor relay error status	CHPCON_ERR_STS	Common	Signal indicates the relay error status on the chip conveyor and chip auger. Relay error normal: 0 Relay error abnormal: 1	S10
BX32F	Alarm stop level 4 (for automatic door)	ASTLV4_DR	Common	The signal turns ON when a stop level 4 alarm is triggered. However, the following alarms are excluded. <<Counter * end >> (*: 1 to 4) <<Tool life end (*)>> (*: tool number)	
BX340 BX341	Door interlock mode status	DIMST1 DIMST2	Common	Signal indicates the status of the door interlock mode. DIMST2 / DIMST1 / Door interlock mode status OFF / ON / Machine setup ON / OFF / Maintenance ON / ON / Automatic operation	S10
BX343	Enable switch	ENSW	Common	Signal turns ON when an enable switch is turned ON.	S10
BX350	Brake load test extension 1	BTWRN1	Common	Signal turns ON when the timing falls within the range set in the user parameter (switch 1: brake load test) <Time extension for brake load test execution> for the axis where the input signal SR_BK*RLS (*: 1 to 5) is assigned. ON conditions 1. When the execution time set in the user parameter (switch 1: brake load test) <Brake load test start time 1 to 4> applies. OFF conditions 1. Brake load test is executed. 2. <Door lock release> key on the <Brake load test> screen is pressed.	
BX351	Brake load test extension 2	BTWRN2	Common		
BX352	Brake load test extension 3	BTWRN3	Common		
BX353	Brake load test extension 4	BTWRN4	Common		
BX354	Brake load test extension 5	BTWRN5	Common		
BX355	Brake load test pending 1	BTALM1	Common	Signal indicates that the brake load test has not been executed on the axis assigned to the input signal SR_BK*RLS (*: 1 to 5). ON conditions (one of the following conditions) 1. When the user parameter (switch 1: brake load test) <Time extension for brake load test execution> elapses after <Brake load test start time 1 to 4> 2. When the brake load test executed and results in an error OFF conditions 1. Brake load test executed and results in a pass 2. <Door lock release> key on the <Brake load test> screen is pressed.	
BX356	Brake load test pending 2	BTALM2	Common		
BX357	Brake load test pending 3	BTALM3	Common		
BX358	Brake load test pending 4	BTALM4	Common		
BX359	Brake load test pending 5	BTALM5	Common		
BX35A	Brake load test in progress	BTRUN	Common	Signal turns ON during the brake load test for the axis where the input signal SR_BK*RLS (*: 1 to 5) is assigned. When the brake load test starts, the signal turns ON, and when the test for all axes ends, the signal turns OFF.	
BX35B	Brake load test - Under load 1	BTLD1	Common	Signal turns ON while a load is applied to the brake in the brake load test, on an axis where	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BX35C	Brake load test - Under load 2	BTLD2	Common	the input signal SR_BK*RLS (*: 1 to 5) is assigned.	
BX35D	Brake load test - Under load 3	BTLD3	Common		
BX35E	Load 4 in brake load test	BTLD4	Common		
BX35F	Load 5 in brake load test	BTLD5	Common		
BX360 to BX36F	Specified main input signal 0 to 15	SEL_IN_SIGNAL0~ SEL_IN_SIGNALF	Common	BDX112: Outputs 16 bit part of the main input signal in accordance with the signal No. Refer to “6.2.1.3. Signal assignment” in the Installation Manual for the input values. (NOTE) Some signals cannot be output.	
BX370	Specified alarm 1	ALMON1	Common	The signal turns ON when an alarm No. for BDY122 under the category No. for BDY121 is triggered.	
BX371	Specified alarm 2	ALMON2	Common	Signal turns ON when an alarm/operator message is triggered that is specified in the user parameter (switch 3: specified alarm signal setting) <Specified alarm signal 2>.	
BX372	Specified alarm 3	ALMON3	Common	Signal turns ON when an alarm/operator message is triggered that is specified in the user parameter (switch 3: specified alarm signal setting) <Specified alarm signal 3>.	
BX373	Specified alarm 4	ALMON4	Common	Signal turns ON when an alarm/operator message is triggered that is specified in the user parameter (switch 3: specified alarm signal setting) <Specified alarm signal 4>.	
BX374	Specified alarm 5	ALMON5	Common	Signal turns ON when an alarm/operator message is triggered that is specified in the user parameter (switch 3: specified alarm signal setting) <Specified alarm signal 5>.	
BX375	Axis position signal output selection	POSSEL1	Common	BX010 to BX017 signals turn ON when this signal is OFF and when the machine coordinates fall within the range indicated in user parameter (switch 2: position signal output range) <*-axis position signal output range (-)/(+)>. BX010 to BX017 signals turn ON when this signal is ON and when the machine coordinates fall within the range indicated in user parameter (switch 2: position signal output range) <*-axis position signal output range 3 (-)/(+)>.	
BX376	Axis position signal output selection 2	POSSEL2	Common	BX018 to BX01F signals turn ON when this signal is OFF and when the machine coordinates fall within the range indicated in user parameter (switch 2: position signal output range) <*-axis position signal output range 2 (-)/(+)>. BX018 to BX01F signals turn ON when this signal is ON and when the machine coordinates fall within the range indicated in user parameter (switch 2: position signal output range) <*-axis position signal output range 3 (-)/(+)>	

<table>
<tr><th>OM name</th><th>Name</th><th>Symbol</th><th>Conversation / NC</th><th>Signal description</th><th>Use</th></tr>
<tr><td>BX378
BX379
BX37A</td><td>Rapid traverse override</td><td>RPDOR0
RPDOR1
RPDOR2</td><td>Common</td><td>This signal resets the [RAPID TRAVERSE OVERRIDE] switch value.
<table>
<tr><th>RPDOR2</th><th>RPDOR1</th><th>RPDOR0</th><th>Override value</th></tr>
<tr><td>OFF</td><td>OFF</td><td>OFF</td><td>100%</td></tr>
<tr><td>OFF</td><td>OFF</td><td>ON</td><td>Speed 4</td></tr>
<tr><td>OFF</td><td>ON</td><td>OFF</td><td>Speed 3</td></tr>
<tr><td>OFF</td><td>ON</td><td>ON</td><td>Speed 2</td></tr>
<tr><td>ON</td><td>OFF</td><td>OFF</td><td>Speed 1</td></tr>
</table>
(NOTE) When the [RAPID TRAVERSE OVERRIDE] switch is an intermediate value, the value before turning into an intermediate value is maintained.</td><td>S1</td></tr>
<tr><td>BX37B</td><td>Rapid traverse override IN</td><td>RPDORIN</td><td>Common</td><td>When this signal turns ON, the [RAPID TRAVERSE OVERRIDE] switch becomes an intermediate value.</td><td></td></tr>
<tr><td>BX380</td><td>Transient communication response</td><td>TRSDRES</td><td>Common</td><td>Signal turns ON when there is a response to the transient communication request.</td><td></td></tr>
<tr><td>BX390</td><td>LB internal start output</td><td>LB_STRT_OUT</td><td>Common</td><td>This signal is the start signal from the loading system.</td><td>S10</td></tr>
<tr><td>BX3A0</td><td>Reset when operation preparation is ON</td><td>MO_RESET</td><td>Common</td><td>Signal turns OFF when the user parameter (switch 1: common) <Reset when operation preparation is ON> is set to <0: No>. Signal turns ON when set to <1: Yes>.</td><td>S1</td></tr>
<tr><td>BX3A1</td><td>Operation preparation hardware conditions</td><td>HW_MO</td><td>Common</td><td>Operation preparation ON: 1
Operation preparation OFF: 0</td><td>S1</td></tr>
<tr><td>BX3A2</td><td>Operation preparation function</td><td>MO</td><td>Common</td><td>Signal turns OFF when the user parameter (switch 1: installation) <Operation preparation function> is set to <0: Disable>.
Signal turns OFF when set to <1: Enable>.</td><td>S1</td></tr>
<tr><td>BX3A3</td><td>Operation preparation circuit selection</td><td>MO_CSEL</td><td>Common</td><td>Signal turns OFF when the user parameter (switch 1: installation) <Operation preparation circuit selection> is set to <0: NC circuit>.
Signal turns OFF when set to <1: Safety relay>.</td><td>S1</td></tr>
<tr><td>BX3A4</td><td>Operation preparation safety relay</td><td>MOSRU</td><td>Common</td><td>Signal turns OFF when the external input signal <Operation preparation safety relay> is set to ON.</td><td>S1</td></tr>
<tr><td>BX3C0</td><td>Door locked
(Outer door / front door)</td><td>DLKON_F</td><td>Common</td><td>Turns ON if the outer door / front door is locked. Turns OFF if it is not locked.</td><td></td></tr>
<tr><td>BX3C1</td><td>Reserved</td><td>RESERVED</td><td>Common</td><td>This signal is reserved.</td><td></td></tr>
<tr><td>BX3C2</td><td>Door locked
(Left side door)</td><td>DLKON_L</td><td>Common</td><td>Turns ON if the left side door is locked. Turns OFF if it is not locked.</td><td></td></tr>
<tr><td>BX3C3</td><td>Door locked
(Right side door)</td><td>DLKON_R</td><td>Common</td><td>Turns ON if the right side door is locked.
Turns OFF if it is not locked.</td><td></td></tr>
<tr><td>BX3C4</td><td>Door locked (Setup chamber left door)</td><td>DLKON_SL</td><td>Common</td><td>Signal turns ON if the setup chamber left door is locked. Signal turns OFF if it is not locked.</td><td></td></tr>
<tr><td>BX3C5</td><td>Door locked (Setup chamber right door)</td><td>DLKON_SR</td><td>Common</td><td>Signal turns ON if the setup chamber right door is locked. Signal turns OFF if it is not locked.</td><td></td></tr>
<tr><td>BX3C6</td><td>Door locked
(External-machine door 1)</td><td>DLKON_X1</td><td>Common</td><td>Turns ON if external-machine door 1 is locked.
Turns OFF if it is not locked.</td><td></td></tr>
<tr><td>BX3C7</td><td>Door locked
(External-machine door 2)</td><td>DLKON_X2</td><td>Common</td><td>Turns ON if external-machine door 2 is locked.
Turns OFF if it is not locked.</td><td></td></tr>
<tr><td>BX3C8</td><td>Reserved</td><td>RESERVED</td><td>Common</td><td>This signal is reserved.</td><td></td></tr>
<tr><td>BX3C9</td><td>Reserved</td><td>RESERVED</td><td>Common</td><td>This signal is reserved.</td><td></td></tr>
<tr><td>BX3CF</td><td>Door open ready</td><td>DROPNOK</td><td>Common</td><td>Turns ON if the outer door / front door is ready to open, and turns OFF if the door cannot be opened.</td><td></td></tr>
</table>

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BX3D0	Door (Outer door / front door) closes	DRCLS	Common	Indicates that the outer door (front door) is closed.	S10
BX3D1	Inner door closes	INDRCL	Common	Indicates that the inner door is closed.	S10
BX3D2	Side door closes	SDDRCL	Common	Indicates that the side door is closed.	
BX3D3	Left side door closed	LDRCL	Common	Signal indicates the status of the left side door.	S10
BX3D4	Right side door closed	RDRCL	Common	Signal indicates the status of the right side door.	S10
BX3D5	Setup chamber left door closed	SLDRCL	Common	Signal indicates the status of the setup chamber left door.	S10
BX3D6	Setup chamber right door closed	SRDRCL	Common	Signal indicates the status of the setup chamber right door.	S10
BX3D7	External-machine door 1 closes	XDR1CL	Common	Indicates that the outside door 1 is closed.	
BX3D8	External-machine door 2 closes	XDR2CL	Common	Indicates that the outside door 2 is closed.	
BX3D9	Reserved	RESERVED	Common	This signal is reserved.	
BX3DA	Reserved	RESERVED	Common	This signal is reserved.	
BX3DB	Automatic door opens	ATDROP	Common	It is the same as opening with **[AUTOMATIC DOOR OPEN/CLOSE]** switch. It opens the door. Operation is possible regardless of the status of the **[MANU-AUTO]** select switch • Signal block 	S10
BX3DC	Automatic door closes	ATDRCL	Common	It is the same as closing with **[AUTOMATIC DOOR OPEN/CLOSE]** switch. It closes the door when the automatic door is open. Operation is possible regardless of the status of the **[MANU-AUTO]** select switch • Signal block 	S10
BX410 BX411 BX412 BX413	J1-axis in motion J2-axis in motion J3-axis in motion TR-axis in motion	LB_MOVEJ1 LB_MOVEJ2 LB_MOVEJ3 LB_MOVETR	Common	This signal turns ON while each axis is in motion on the loading system. (NOTE) Signal turns ON in manual and automatic operation modes.	
BX420 BX421 BX422 BX423	J1-axis in-position J2-axis in-position J3-axis in-position TR-axis in-position	LB_INPJ1ON LB_INPJ2ON LB_INPJ3ON LB_INPTRON	Common	This signal turns ON when the in-position operation has been completed for each axis on the loading system.	
BX42F	All LB axes in-position	LB_INPALLON	Common	This signal turns ON when the in-position operation has been completed for all axes on the loading system.	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BX430 BX431 BX432 BX433 BX434	LB alarm stop level 1 to 5	LB_ASTLV1 LB_ASTLV2 LB_ASTLV3 LB_ASTLV4 LB_ASTLV5	Common	When an alarm is triggered for a particular alarm level on the loading system, the signal for the corresponding level turns ON. When multiple alarms are triggered, each corresponding signal turns ON.	
BX440	LB paused	LB_STOP_LD	Common	This signal turns ON when there is no program running with <Normal> start conditions and there is at least one program paused on the loading system. It turns OFF in all other situations.	
BX441	LB resetting	LB_RSTST	Common	One shot (300 ms + until reset processing is complete) signal turns ON when the loading system is resetting.	
BX442	LB operation mode status	LB_MODEST	Common	This signal turns ON when the loading system operation mode is in “Automatic operation mode”. It turns OFF when in “Manual operation mode”.	
BX443	LB servo ON	LB_SVONST	Common	This signal turns ON when the servo is ON for all axes on the loading system.	
BX444	LB single operation in progress	LB_SINGL	Common	This signal turns ON when single operation is set on the loading system.	
BX445	LB automatic operation starting up	LB_STL	Common	This signal turns ON when even just one program with <Normal> start conditions is running while the loading system is in automatic operation mode. It turns OFF when the loading system is in one of the following situations. • When all programs with <Normal> start conditions are running or paused • Reset • When an alarm and operator message with a stop level of 3 or higher are triggered	
BX446	LB automatic operation ready	LB_AUTOOK	Common	This signal turns ON when the loading system is ready to start in automatic operation mode. It turns OFF when the loading system is in one of the following situations. • When at least one program with <Normal> start conditions is running • When there is a display device (teaching pendant and LB Manager) with operation control • Resetting • When either the safety fence or front door is open • Single operation is set • When an alarm and operator message with a stop level of 2 or higher are triggered However, this excludes the operator message <<Standby mode>>. • When a program with <Alarm> start conditions is running	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BX450 BX451	LB specified coordinate system 1 LB specified coordinate system 2	LB_CORD1 LB_CORD2	Common	This signal indicates the current coordinate system that is output to BDX256 to BDX263 by the loading system. Output coordinate system is the “Tool 0 coordinate”: LB_CORD2:0、LB_CORD1:0 Output coordinate system is the “TCP coordinate in the selected tool coordinate”: LB_CORD2:0、LB_CORD1:1 Output coordinate system is the “Joint coordinate”: LB_CORD2:1、LB_CORD1:0	
BX458	LB coordinate 1 to 4 unit system	LB_CDPOS_UNIT	Common	This signal indicates the unit for the coordinate that is output to BDX256 to 263 by the loading system. Metric: 0 Inches: 1	
BX460	LB brake load test extension 1	LBBTW1	Common	Refer to “1.6.5.1 List of output signals” in the Data Bank & Alarm Manual for further details.	
BX461	LB brake load test extension 2	LBBTW2	Common		
BX462	LB brake load test extension 3	LBBTW3	Common		
BX463	LB brake load test pending 1	LBBTA1	Common		
BX464	LB brake load test pending 2	LBBTA2	Common		
BX465	LB brake load test pending 3	LBBTA3	Common		
BX466	LB brake load test - Under load 1	LBBTL1	Common		
BX467	LB brake load test - Under load 2	LBBTL2	Common		
BX468	LB brake load test - Under load 3	LBBTL3	Common		
BX621	Index check for QT No.2 pallet	IL_SETPLT2	Common	Not indexed: 0 Indexed: 1	
BX622	Index check for QT No.1 pallet	IL_SETPLT1	Common	Not indexed: 0 Indexed: 1	
BX623	QT clamp check	IL_QT_CLP	Common	QT unclamp: 0 QT clamp: 1	
BX624	Setup chamber right door	IL_DR8_LK	Common	Door unlocked: 0 Door locked: 1	
BX625	LS close check for setup chamber right door	IL_DR8_CL	Common	Door open: 0 Door close: 1	
BX626	LS lock check for setup chamber right door	IL_DR7_LK	Common	Door unlocked: 0 Door locked: 1	
BX627	LS close check for setup chamber left door	IL_DR7_CL	Common	Door open: 0 Door close: 1	
BX628	LS lock check for setup chamber front door	IL_DR2_LK	Common	Door unlocked: 0 Door locked: 1	
BX629	LS close check for setup chamber right door	IL_DR2_CL	Common	Door open: 0 Door close: 1	
BX62A	LS lock check for outside door 2	IL_DR6_LK	Common	Door unlocked: 0 Door locked: 1	
BX62B	LS close check 3 for outside door 2	IL_DR6_3_CL	Common	Door open: 0 Door close: 1	
BX62C	LS close check 2 for outside door 2	IL_DR6_2_CL	Common	Door open: 0 Door close: 1	
BX62D	LS close check 1 for outside door 2	IL_DR6_1_CL	Common	Door open: 0 Door close: 1	
BX62E	LS lock check for outside door 1	IL_DR5_LK	Common	Door unlocked: 0 Door locked: 1	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BX62F	LS close check 3 for outside door 1	IL_DR5_3_CL	Common	Door open: 0 Door close: 1	
BX630	LS close check 2 for outside door 1	IL_DR5_2_CL	Common	Door open: 0 Door close: 1	
BX631	LS close check 1 for outside door 1	IL_DR5_1_CL	Common	Door open: 0 Door close: 1	
BX632	Magazine shutter close check	IL_MGS_CL	Common	Door open: 0 Door close: 1	
BX633	Magazine door lock check	IL_DRMG_LK	Common	Door unlocked: 0 Door locked: 1	
BX634	Magazine door close check	IL_DRMG_CL	Common	Door open: 0 Door close: 1	
BX635	LS lock check for machining chamber right door	IL_DR4_LK	Common	Door unlocked: 0 Door locked: 1	
BX636	LS close check for machining chamber right door	IL_DR4_CL	Common	Door open: 0 Door close: 1	
BX637	LS lock check for machining chamber left door	IL_DR3_LK	Common	Door unlocked: 0 Door locked: 1	
BX638	LS close check for machining chamber left door	IL_DR3_CL	Common	Door open: 0 Door close: 1	
BX639	LS lock check for machining chamber front door	IL_DR1_LK	Common	Door unlocked: 0 Door locked: 1	
BX63A	LS close check for machining chamber front door	IL_DR1_CL	Common	Door open: 0 Door close: 1	
BX7FE	Delete input signal (Always-ON)	BXRVON	Common	Signal is used for conversion of an OM that was deleted in D00. The signal is always ON.	
BX7FF	Delete input signal (Always-OFF)	BXRVOFF	Common	Signal is used for conversion of an OM that was deleted in D00. The signal is always OFF.	

4.2.4 Internal Output (PLC→NC, BY000 to BY7FF)

4.2.4.1 Overview

This signal is from the PLC→NC.
It indicates the internal status of the PLC in bits.

OM name	BY000 to BY7FF (hexadecimal shown)
Total number of signals	2048
Type	Bit
Output specification	Available / None
Double coil check	ON
Record when power is OFF	Volatile
Index addressing	Available

4.2.4.2 Signal List

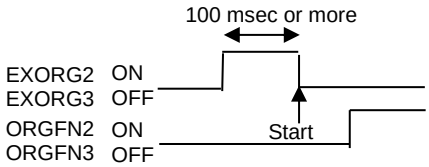
The list of signals from the PLC→NC is shown below.

OM name : This column is used for specifying the signal in the PLC program description.
Symbol : This symbol is for supporting the OM. When the OM comment displayed is enabled, this symbol is displayed on the screen.
Use : This column shows whether PLC→NC is the factory default setting being used.
The following symbols indicate the control program being used.
Note that the symbol already being used cannot be used because it would become a double coil.
High speed control program 10 : H10
Standard control program 1 : S1
Standard control program 10 : S10

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BY000	External machine zero return	EXORG	Common	Once the contact for the signal EXORG has been closed, when it is opened, the machine moves to the zero point. This is valid in memory operation mode. (NOTE 1) Travel occurs at the positioning speed. (NOTE 2) The travel order is Z → XY (simultaneous). (NOTE 3) Travel to the middle point is not carried out. (NOTE 4) The signal input must be faster than 100 msec. (NOTE 5) Automatic operation starting (STL) does not turn ON during an external machine zero return. (NOTE 6) When the machine is equipped with a lathe spindle and the lathe spindle is selected, the additional axis that corresponds to the lathe spindle does not perform a machine zero return operation. <u>Enable conditions</u> (1) Automatic operation starting (STL) is OFF. (2) The **[FEED HOLD]** switch is not pressed. (3) The external stop (EXSTOP) is OFF. (4) An alarm higher than a stop level 2 alarm has not been triggered. (5) The door is closed. (6) When one of the following applies: • <DOOR INTERLOCK MODE> is set to <SERVICE> and the spindle is stopped. • <DOOR INTERLOCK MODE> is set to <AUTOMATIC> or <SETTING>. (7) Brake load test executing (BTRUN) signal is OFF (8) Screen <Move start position> is not currently displaying	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BY001	External Z-axis machine zero return	EXZORG	Common	Once the contact for the signal EXZORG has been closed, when it is opened, it moves to the zero point. This is valid in memory operation mode. (NOTE 1) Travel occurs at the positioning speed. (NOTE 2) Travel to the middle point is not carried out. (NOTE 3) The signal input must be faster than 100 msec. (NOTE 4) Automatic operation starting (STL) does not turn ON during an external Z-axis machine zero return. <u>Enable conditions</u> (1) Automatic operation starting (STL) is OFF. (2) The **[FEED HOLD]** switch is not pressed. (3) The external stop (EXSTOP) is OFF. (4) An alarm higher than a stop level 2 alarm has not been triggered. (5) The door is closed. (6) When one of the following applies: • <DOOR INTERLOCK MODE> is set to <SERVICE> and the spindle is stopped. • <DOOR INTERLOCK MODE> is set to <AUTOMATIC> or <SETTING>. (7) Brake load test executing (BTRUN) signal is OFF (8) Screen <Move start position> is not currently displaying	

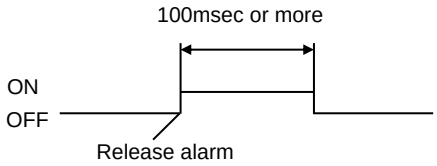
OM name	Name	Symbol	Conversation / NC	Signal description	Use
BY002 to BY007	External No 1 reference position return External No 2 reference position return External No 3 reference position return External No 4 reference position return External No 5 reference position return External No 6 reference position return	EXREF EXREF2 EXREF3 EXREF4 EXREF5 EXREF6	NC	Once the contact for the signals EXREF/EXREF2 through EXREF6 closes, when it opens, the machine moves to the reference positions No.1 through No.6, which are set in the parameter. This is valid in memory operation mode. (NOTE 1) Travel occurs at the positioning speed. (NOTE 2) The travel order is Z → XY (simultaneous). (NOTE 3) Travel to the middle point is not carried out. (NOTE 4) When travel to the reference positions No.1 through No.6 is completed normally, the reference position complete signal (REFIN/REFIN2 through REFIN6) is output. (NOTE 5) The signal input must be faster than 100 msec. (NOTE 6) Automatic operation starting (STL) does not turn ON during an external return operation to the reference positions No.1 through No.6. (NOTE 7) When the machine is equipped with a lathe spindle and the lathe spindle is selected, the additional axis that corresponds to the lathe spindle does not perform a reference position return operation. <u>Enable conditions</u> (1) Automatic operation starting (STL) is OFF (2) The **[FEED HOLD]** switch is not pressed. (3) The external stop (EXSTOP) is OFF. (4) An alarm higher than a stop level 2 alarm has not been triggered. (5) The door is closed. (6) When one of the following applies: • <DOOR INTERLOCK MODE> is set to <SERVICE> and the spindle is stopped. • <DOOR INTERLOCK MODE> is set to <AUTOMATIC> or <SETTING>. (7) Brake load test executing (BTRUN) signal is OFF (8) Screen <Move start position> is not currently displaying	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BY008 to BY009	External machine zero return 2 External machine zero return 3	EXORG2 EXORG3	Conversation	Once the contact for the signal EXORG2 or EXORG3 closes, when it opens, a return operation to reference position No.2 or No.3, which is set in the parameter, is possible. This is valid in memory operation mode.  The signal ORGFN2 or ORGFN3 can be used to confirm the completion of the return operation. (Reference) Refer to section “Machine zero return finish 2” for further details. (NOTE 1) The signals EXORG2 and EXORG3 are enabled when the signal STL is OFF. (NOTE 2) The signal STL does not output during a return. (NOTE 3) If in edit mode while operating, wait longer than 500 msec before inputting. (NOTE 4) The signals are enabled when the doors are closed. (NOTE 5) The signals are enabled when one of the following applies. • <DOOR INTERLOCK MODE> is set to <SERVICE> and the spindle is stopped. • <DOOR INTERLOCK MODE> is set to <AUTOMATIC> or <SETTING>. (NOTE 6) These signals are enabled when the brake load test executing (BTRUN) signal is OFF.	
BY00A	External pallet index	CTURN	Common	Once the contact for the signal CTURN has been closed, when it is opened, the pallet is indexed. The operation is the same as pressing the <P.INDX> key. This is valid in memory operation mode. <u>Enable conditions</u> (1) Automatic operation starting (STL) is OFF (2) The **[FEED HOLD]** switch is not pressed. (3) The external stop (EXSTOP) is OFF. (4) An alarm higher than a stop level 2 alarm has not been triggered. (5) The door is closed. (6) Brake load test executing (BTRUN) signal is OFF (7) Screen <Move start position> is not currently displaying	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BY00B to BY00C	External additional axis 1 (+) index External additional axis 1 (-) index	OUT1P OUT1M	Common	BDY048/BDY049: Rotates the additional axis on the outer pallet the amount (value) that is specified in the angle command for the external additional axis 1. The additional axis address is specified in the user parameter (switch 2: outside index) <Outer rotary table 1 switch address>. When the command value is 0 (not set), the outer pallet additional axis rotates the amount (value) that is set in OUT1P <Outer rotary table 1 indexing (+)> and in OUT1M <Outer rotary table 1 indexing (-)> in the user parameter. Operation is started during the fall of the signal. It does not operate when the outer door (side door) is open. This is valid in memory operation mode.	
BY00D to BY00E	External additional axis 2 (+) index External additional axis 2 (-) index	OUT2P OUT2M	Common	BDY050/BDY051: Rotates the additional axis on the outer pallet the amount (value) that is specified in the angle command for the external additional axis 2. The additional axis address is specified in the user parameter (switch 2: outside index) <Outer rotary table 2 switch address>. When the command value is 0 (not set), the outer pallet additional axis rotates the amount (value) that is set in OUT2P <Outer rotary table 2 indexing (+)> and in OUT2M <Outer rotary table 2 indexing (-)> in the user parameter. Operation is started during the fall of the signal. It does not operate when the outer door (side door) is open. This is valid in memory operation mode.	
BY00F	PLC skip	PLCSKIP	Common	The user can create skip conditions on the built-in PLC and execute skip operations. (G31 skip and tool length measurement, etc.)	
BY010	Spindle operation lock	SPLOCK	Common	Prohibits spindle rotation operation. Stop and orientation operations are possible. The spindle does not stop during spindle rotation even if this signal is ON.	
BY011	ATC operation lock	ATCLCK	Common	Prohibits ATC operation. The magazine turn, the **[ATC]** key, tool change in MDI and memory operation mode and the ready operation for the next tool are prohibited.	
BY012 to BY018	X-,Y-axes operation lock Z-axis operation lock No 4-axis operation lock No 5-axis operation lock No 6-axis operation lock No 7-axis operation lock No 8-axis operation lock	XYLOCK ZLOCK 4LOCK 5LOCK 6LOCK 7LOCK 8LOCK	Common	Prohibits operation on the following axes: X, Y, Z, 4, 5, 6, 7 and 8. When another axis is operating at the same time, the other axis also does not operate. The clamp / unclamp operations are also prohibited on the additional axis. (NOTE) The lathe spindle does not stop while the axis that corresponds to the lathe spindle is rotating, even if this signal is ON.	
BY01F	External additional axis index lock	OUTLCK	Common	Prohibits the index operation for the external A-, B- and C-axes when this signal is ON. The panel switch and external input signal are both prohibited.	
BY028	No 5-axis overtravel (+)	OVTR5P	Common	Inputs the overtravel (+) status for the No 5-axis. The alarm <<Overtravel (+5)>> is triggered on the machine when ON.	

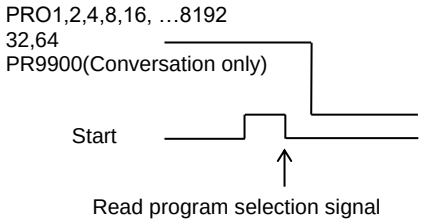
OM name	Name	Symbol	Conversation / NC	Signal description	Use
BY029	No 5-axis overtravel (-)	OVTR5M	Common	Inputs the overtravel (-) status for the No 5-axis. The alarm <<Overtravel (-5)>> is triggered on the machine when ON.	
BY02A	No 6-axis overtravel (+)	OVTR6P	Common	Inputs the overtravel (+) status for the No 6-axis. The alarm <<Overtravel (+6)>> is triggered on the machine when ON.	
BY02B	No 6-axis overtravel (-)	OVTR6M	Common	Inputs the overtravel (-) status for the No 6-axis. The alarm <<Overtravel (-6)>> is triggered on the machine when ON.	
BY02C	No 7-axis overtravel (+)	OVTR7P	Common	Inputs the overtravel (+) status for the No 7-axis. The alarm <<Overtravel (+7)>> is triggered on the machine when ON.	
BY02D	No 7-axis overtravel (-)	OVTR7M	Common	Inputs the overtravel (-) status for the No 7-axis. The alarm <<Overtravel (-7)>> is triggered on the machine when ON.	
BY02E	No 8-axis overtravel (+)	OVTR8P	Common	Inputs the overtravel (+) status for the No 8-axis. The alarm <<Overtravel (+8)>> is triggered on the machine when ON.	
BY02F	No 8-axis overtravel (-)	OVTR8M	Common	Inputs the overtravel (-) status for the No 8-axis. The alarm <<Overtravel (-8)>> is triggered on the machine when ON.	
BY030	PLC-axis machine lock	MCNLKPL CON	Common	Enables the machine lock on the PLC-axis. The machine lock is only enabled on an axis where the PLC-axis control signal is valid. If this signal status changes during a travel operation, the machine operation stops temporarily and the machine lock status changes. In addition, the command issued during the travel operation is cleared.	
BY031 to BY033	PLC-axis override 1 PLC-axis override 2 PLC-axis override 3	OVRPLC1 OVRPLC2 OVRPLC3	Common	Enables override during rapid feed and reference position return operations. The change is reflected even during a travel operation. (The change is reflected on the axis where the PLC-axis control signal is valid.) 000: Invalid (100%) 001: Override speed 1 010: Override speed 2 011: Override speed 3 100: Override speed 4 101: Invalid (100%) 110: Invalid (100%) 111: Invalid (100%)	
BY034	P1-axis control	CTRL_P1	Common	The PLC-axis control processing is performed on the built-in PLC while the PLC-axis control signal is ON. When this signal is OFF, operations such as JOG travel can be performed in manual operation mode. Note, when this signal is changed during a travel operation, the machine decelerates and stops. After decelerating and stopping, if this signal turns ON, the operation follows the command status in the PLC-axis control data register. If this signal turns OFF, operation in manual operation mode is enabled.	
BY035	P2-axis control	CTRL_P2	Common		
BY036	P3-axis control	CTRL_P3	Common		
BY037	P4-axis control	CTRL_P4	Common		

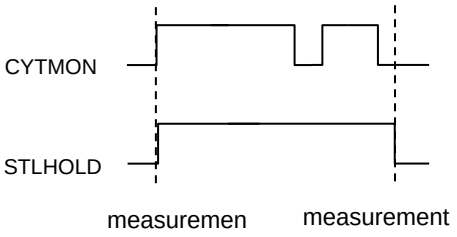
OM name	Name	Symbol	Conversation / NC	Signal description	Use
BY040	P1-axis command start	START_P1	Common	Command start. Travel starts during the rise time when switching from OFF to ON based on the values for the command type / command data 1 through 3. The commands, for positioning during servo OFF and for the reference position return, are issued (start) after the servo turns ON. The command is not accepted when the PLC-axis control signal is OFF. The command is not accepted when the status is BUSY. The command is not accepted when an ALARM status is triggered.	
BY041	P1-axis reset	RESET_P1	Common	The alarm is cleared or reset. The current command is cancelled. When the signal turns ON during a travel operation, travel decelerates and stops before the reset processing is performed. The command is not accepted when the PLC-axis control signal is OFF. However, the alarm that appears on the NC screen is also reset (Only for the PLC-axis that is turned ON).	
BY042	P1-axis servo OFF	SVOFF_P1	Common	Changes the status to ON when the servo is OFF. The command is not accepted when the PLC-axis control signal is OFF. When the signal turns ON during a travel operation (including temporary stoppage during a travel operation such as positioning or a reference position return), the alarm “P*-axis servo off command detected” is triggered, and the servo turns OFF after travel decelerates and stops. The alarm “No comm receivable as P*-axis servo OFF is active” is triggered when the signal turns ON at the start of a travel operation.	
BY043	P1-axis temporary stop	STOP_P1	Common	Temporary stop. The command is not accepted when the PLC-axis control signal is OFF. If the signal turns ON during a travel operation, travel decelerates and stops. If the signal turns OFF, the command is re-issued (re-started). However, only positioning and reference position return commands are re-issued (re-started). The alarm “No command receivable as P*-axis stop is active” is triggered when the signal turns ON at the start of a travel operation.	
BY044	P1-axis abso./incre.	ABSINC_P1	Common	This signal refers to the Abso./Incre. operation for a positioning command. If the signal turns ON, it becomes a Abso. command. When the command START changes from OFF to ON, the setting that corresponds to the ON status is valid.	
BY045	P1-axis near override	NEAR_P1	Common	Near override enabled. When the command START changes from OFF to ON, the setting that corresponds to the ON status is valid.	
BY050	P2-axis command start	START_P2	Common	Controls the 2nd axis on the PLC. The content is the same as the 1st axis.	
BY051	P2-axis reset	RESET_P2	Common		
BY052	P2-axis servo OFF	SVOFF_P2	Common		
BY053	P2-axis temporary stop	STOP_P2	Common		
BY054	P2-axis abso./incre.	ABSINC_P2	Common		
BY055	P2-axis near override	NEAR_P2	Common		
BY060	P3-axis command start	START_P3	Common	Controls the 3rd axis on the PLC. The content is the same as the 1st axis.	
BY061	P3-axis reset	RESET_P3	Common		
BY062	P3-axis servo OFF	SVOFF_P3	Common		

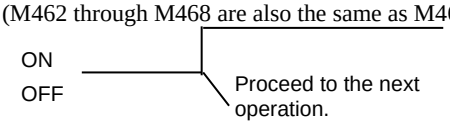
OM name	Name	Symbol	Conversation / NC	Signal description	Use
BY063	P3-axis temporary stop	STOP_P3	Common		
BY064	P3-axis abso./incre.	ABSINC_P3	Common		
BY065	P3-axis near override	NEAR_P3	Common		
BY070	P4-axis command start	START_P4	Common	Controls the 4th axis on the PLC. The content is the same as the 1st axis.	
BY071	P4-axis reset	RESET_P4	Common		
BY072	P4-axis servo OFF	SVOFF_P4	Common		
BY073	P4-axis temporary stop	STOP_P4	Common		
BY074	P4-axis abso./incre.	ABSINC_P4	Common		
BY075	P4-axis near override	NEAR_P4	Common		
BY080	External reset	EXRST	Common	Same as the **[RST]** key on the operation panel. It is used with standard ladder 1.	S1
BY081	External alarm release	ALMRLS	Common	If the machine stops due to the following alarm, stop level 1 and 2 alarms can be released by inputting the ALMRLS signal as an external input signal. • When the alarm is an <<External error**>> (**:10 to 99) and the <Cancel level> for the external I/O signal setting is <4: ALMRLS signal>.  	
BY082	Change memory operation mode	CHGMDMEM	Common	The fall of the signal is the same as when the **[MEM]** key is pressed. However, the prohibit operation does not function with the key operation prohibit signal, because this is not a key operation.	
BY083	Change MDI operation mode	CHGMDMDI	Common	The fall of the signal is the same as when the **[MDI]** key is pressed. However, the prohibit operation does not function with the key operation prohibit signal, because this is not a key operation.	
BY084	Change manual operation mode	CHGMDMNL	Common	The fall of the signal is the same as when the **[MANU]** key is pressed. However, the prohibit operation does not function with the key operation prohibit signal, because this is not a key operation.	
BY085	Change program edit mode	CHGMDEDT	Common	The fall of the signal is the same as when the **[EDIT]** key is pressed. However, the prohibit operation does not function with the key operation prohibit signal, because this is not a key operation.	
BY086	Servo relay ON command	SVRON	Common	Issues the servo relay ON command to the NC at the falling edge of the signal. It is used with standard ladder 1.	S1
BY087	Power OFF	PLC_PWOFF	Common	During the falling edge of the signal, a power OFF command is issued to the NC. In addition, the power turns OFF when the following conditions are met during operation. - Program end (Program is considered ended even for a scheduled restart and pallet start.) - Operation pause or block stop (NOTE) Input the signal after at least 100 ms.	
BY088	Pallet start	EXSTART	Common	Issues the start command to the NC for the falling edge of the signal. If used in another program apart from standard control program 10, the circuit cannot be saved because it becomes a duplicate circuit. As a basic rule, use the startup output (for conversion) (BY3F0).	S10
BY089	External stop	EXSTOP	Common	Same as the **[FEED HOLD]** switch on the operation panel.	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BY08A	Pallet start	OUTSTA	Common	Issues the pallet start command to the NC for the falling edge of the signal. If used in another program apart from standard control program 10, the circuit cannot be saved because it becomes a duplicate circuit. As a basic rule, use the pallet startup output (for conversion) (BY3F1).	S10
BY08B	External start (LB)	EXSTART_LB	Common	Issues the start command to the NC for the falling edge of the signal.	S10
BY08D	LB internal reset input	LB_RST_IN	Common	This signal is a reset signal to the loading system. Turn ON this signal for 100 ms or longer. If used in another program apart from standard control program 10, the circuit cannot be saved because it becomes a duplicate circuit. As a basic rule, use the internal reset input (for conversion) (BY7F7).	S10
BY090	Single block	SINGLON	Common	The fall of the signal is the same as when the **[SINGL]** key is pressed. However, the prohibit operation does not function with the key operation prohibit signal, because this is not a key operation.	
BY091	Dry run	DRYRUNON	Common	The fall of the signal is the same as when the **[DRY]** key is pressed. However, the prohibit operation does not function with the key operation prohibit signal, because this is not a key operation.	
BY092	External optional skip	OPSKIP	NC	When this signal is ON, the block with a slash (/) is ignored and is not executed.	
BY093	Machine lock	MCNLKON	Common	The fall of the signal is the same as when the **[M.LCK]** key is pressed.	
BY094	External optional stop	OPSTOP	NC	When this signal is ON, memory operation stops temporarily when M01 is executed.	
BY095	Coolant OFF key	CLSWOF	Common	Turns OFF the **[CLT.P]** key during the fall of the signal. It also turns OFF when the coolant pump automatically turns OFF.	
BY096	Coolant ON key	CLSWON	Common	Turns ON the **[CLT.P]** key during the fall of the signal. It does not turn ON when the coolant pump automatically turns OFF.	
BY097	Chip shower OFF key	CHIPSWROFF	Common	Turns OFF the **[CHP.F]** key during the fall of the signal.	
BY098	Chip shower ON key	CHIPSWR	Common	Turns ON the **[CHP.F]** key during the fall of the signal.	
BY0A0	Data protection	PROTECT	Common	This signal is the same as the <Data protection> key (**[DATA PROTECTION]** switch).	
BY0A1 to BY0A4	Operation level 1 Operation level 2 Operation level 3 Operation level 4	LEVEL1 LEVEL2 LEVEL3 LEVEL4	Common	The operation level can be set. When signals for multiple operation levels are ON, it is set to the highest operation level among those that are ON. When all signals are OFF, it is assigned level 0.	
BY0A5	Mode change lock	MDLOCK	Common	Prohibits mode changeover.	
BY0A6	Program edit lock	PRLOCK	Common	Prohibits data change operations (insert, delete and set) in edit mode (program screen). Deleting and copying files in the memory display is also prohibited. In addition, operations that change the data during preparation are prohibited. However, the prohibit signal is canceled when the **[EMERGENCY]** switch is ON.	
BY0A7	Edit lock	EDLOCK	Common	Prohibits program editing. Data bank editing is also prohibited. Workpiece counter, ATC tool and maintenance notice editing is also prohibited. However, the prohibit signal is canceled when the **[EMERGENCY STOP]** switch is ON.	
BY0A8	Key operation lock 1	KYLOCK1	Common	Prohibits or locks the edit operation key, the mode change key and the screen change key. In addition, the tap operation is also prohibited. Note that operation is available only for the subtool keys (except <Accessories>key).	

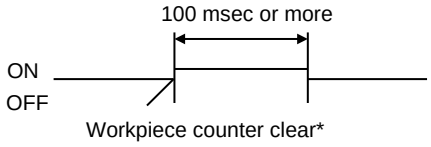
OM name	Name	Symbol	Conversation / NC	Signal description	Use
BY0A9	Key operation lock 2	KYLOCK2	Common	Prohibits or locks the operation function key.	
BY0AA	Key operation lock 3	KYLOCK3	Common	Prohibits or locks the **[CLT.P]** key and the **[CHP.F]** key.	
BY0AB	Key operation lock 4	KYLOCK4	Common	Prohibits or locks the <LIGHT> and <TABLE LIGHT> keys.	
BY0AC	Key operation lock 5	KYLOCK5	Common	Prohibits or locks the manual operation key.	
BY0AD	Key operation lock 6	KYLOCK6	Common	Prohibits or locks the **[RST]** key.	
BY0AE	Mode change available	MODEOK	Common	Even if the mode lock (MDLOCK) is ON, the mode can be changed if this signal is ON.	
BY0AF	JOG key lock	JOGLCK	Common	This signal locks the keys: **[+X]**, **[-X]**, **[+Y]**, **[-Y]**, **[+Z]**, **[-Z]**, **[+4]** and **[-4]**. (NOTE) During step travel, motion does not stop until step travel is complete, even when the signal is turned ON.	
BY0B0	Tool data editing possible	TLEDOK	Common	Even if the edit lock (EDLOCK) is ON, the tool data (NC) and the tool list (conversation) can be edited if this signal is ON.	
BY0B1	User parameter editing possible	UPEDOK	Common	Even if the edit lock (EDLOCK) is ON, the user parameter can be edited if this signal is ON.	
BY0B2	External I/O signal editing possible	IOEDOK	Common	Even if the edit lock (EDLOCK) is ON, editing is possible on the external I/O signal screen if this signal is ON.	
BY0B3	Communication parameter editing possible	CPEDOK	Common	Even if the edit lock (EDLOCK) is ON, editing is possible on the communication parameter screen if this signal is ON.	
BY0B4	Machine parameter editing possible	MPEDOK	Common	Even if the edit lock (EDLOCK) is ON, the machine parameter can be edited if this signal is ON.	
BY0B5	Workpiece counter editing possible	WCEDOK	Common	Even if the edit lock (EDLOCK) is ON, the workpiece counter can be edited if this signal is ON.	
BY0B6	ATC tool editing possible	MGEDOK	Common	Even if the edit lock (EDLOCK) is ON, the ATC tool can be edited if this signal is ON.	
BY0B7	Program editing possible	PREDOK	Common	Even if the edit lock (EDLOCK) is ON, or even if the program edit lock (PRLOCK) is ON, the program can be edited if this signal is ON.	
BY0B8	Maintenance notice editing possible	MTEDOK	Common	Even if the edit lock (EDLOCK) is ON, the maintenance notice can be edited if this signal is ON.	
BY0B9	Directory display available	DIROK	Common	Even if the edit lock (EDLOCK) is ON, or even if the program edit lock (PRLOCK) is ON, the directory display operation can be edited if this signal is ON.	
BY0BB	Workpiece coordinate zero editing possible	WPEDOK	NC	Even if the edit lock (EDLOCK) is ON, the workpiece coordinate zero can be edited if this signal is ON.	
BY0BC	Macro data editing possible	MCEDOK	NC	Even if the edit lock (EDLOCK) is ON, the macro data can be edited if this signal is ON.	
BY0BD	Cutting condition editing possible	CTEDOK	Conversation	Even if the edit lock (EDLOCK) is ON, the cutting conditions in the data bank can be edited when this signal is turned ON.	
BY0BE	Tapping drill diameter editing possible	THEDOK	Conversation	Even if the edit lock (EDLOCK) is ON, the tapping drill diameter in the data bank can be edited when this signal is turned ON.	
BY0BF	Tool pattern editing possible	TPEDOK	Conversation	Even if the edit lock (EDLOCK) is ON, the tool pattern in the data bank can be edited when this signal is turned ON.	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BY0C0 to BY0CD	External program selection 1 External program selection 2 External program selection 4 External program selection 8 External program selection 16 External program selection 32 External program selection 64 External program selection 128 External program selection 256 External program selection 512 External program selection 1024 External program selection 2048 External program selection 4096 External program selection 8192	PRO1 PRO2 PRO4 PRO8 PRO16 PRO32 PRO64 PRO128 PRO256 PRO512 PRO1024 PRO2048 PRO4096 PRO8192	Common	Used when the program created beforehand is specified externally.  In the program number range 1 to 8999, use the external program selection 1 to 8192 to specify the program. (NOTE) To specify a schedule number after 9900 in conversation language, turn ON the external program selection 9900 and use external program selection 1 to 64 to specify a program.	
BY0CF	External program selection 9900	PR9900	Conversation	Used when a conversation schedule program is used. 9900 is added to the signal number of the external program selection.	
BY0D0	Change tool change mode	ARMSPEED	Common	The fall of the signal is the same as when the M432 (signal output 432: speed change on arm swivel (low speed)) command is issued.	
BY0D2	Start LED operation	STALED	Common	The startup LED light turns on when this signal is turned ON. The startup LED light turns off when this signal is turned OFF. However, even if the signal turns OFF, the startup LED light does not turn off during automatic operation, during external machine zero return and during external reference position return. The startup LED light turns off when the operation ends.	S10
BY0D3	External program load request	PLC_PRLOAD	Common	External program selection (BY0C0 to 0CF) or the external program selection number (BDY064) is loaded and displayed on the <Program> screen for the falling edge of the signal. This signal is ignored during operation or while the program is being loaded. (NOTE) Input the signal for 100 ms or longer. The error <<External program number error>> is triggered if a program number is selected that is outside of the program range.	
BY0D4	Interrupt type macro	UINT	NC	Cancels and interrupts the program that is currently executing and executes the specified program operation when an interrupt macro (UINT) signal is ON during program execution.	
BY0D5	Restart command	RESTRTON	Common	When this signal turns ON, it is the same as when the user parameter (switch 1: operation) <Program repeat> is set to <1:Repeat>.	

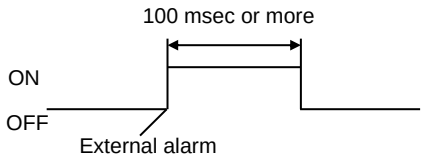
OM name	Name	Symbol	Conversation / NC	Signal description	Use
BY0D6	Shifting external workpiece coordinates	EXWORK	NC	If the user parameter (switch 1: operation) <Shift method in external workpiece coordinate system> is set to <0: Method 1>, the set value for the workpiece coordinate zero (external) becomes valid when this signal is ON. If set to <1: Method 2>, the set value for the workpiece coordinate zero (external) is valid even when this signal is not ON.	
BY0D7	Machining load monitor function OFF	TLDSWOFF	Common	Machining load monitor function is disabled.	
BY0D8	Buzzer mute	BUZZ_LOCK	Common	Signal turns ON to mute the buzzer on the machine. Even if BY0E8 is ON, when this signal is turned ON, the buzzer stays muted.	
BY0D9	Cycle time measurement command	CYTMON	Common	While this signal is ON, the cycle time measurement is carried out.  (NOTE) The cycle time is measured while the program is running regardless if this signal is ON or OFF.	S1
BY0DA	Character string copy command	STRCPY_CMD	Common	When this signal rises, it copies the character string. The character string until before “0x00” is input from BDY554 is the character string to be copied. (If “0x00” is not input, then the character string up until BDY607 is used.) The character string cannot be copied when one of the following signals is ON. BX1B8: Character string writing is restricted BX1B9: Character string copied The copied destination will be the MDI operation screen and program edit screen.	
BY0E0	M function finished	MFIN	Common	Signal MFIN is input as an end signal for the M function output M800 to M899 in the internal input. It has the same properties as M00 through M99, MF and MFIN, which are noted in “M function signal output” from the “1.6.5 External output signal” found in the Data Bank & Alarm Manual. However, it is part of another separate signal group.	

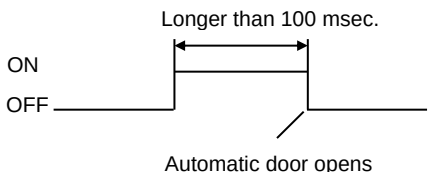
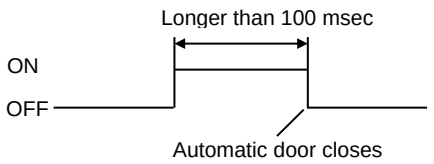
OM name	Name	Symbol	Conversation / NC	Signal description	Use
BY0E1 to BY0E5	Wait	M460 M462 M464 M466 M468	Common	Used as an input signal with M460 (signal output function 460) and for making the next operation ready during machine operation. The signal output 460 waits for the signal M460 contact to close before it proceeds to the next operation. The signal output 461 waits for the signal M460 contact to open before it proceeds to the next operation. (M462 through M468 are also the same as M460.)  (NOTE) The maximum wait time is set in the user parameter <External signal reading time limit>. When the set value is 0, there is no wait time limit.	
BY0E6	Supply water finished	TW_SUPOK	Common	The coolant chamber indicates if the level is full when this signal is turned ON and OFF. OFF → Level is not full ON → Level is full	S10
BY0E7	Discharge finished	TW_FIN	Common	Indicates the status of the tool cleaning discharge when the signal TW_FIN is turned ON and OFF. ON → Discharge finished OFF → Discharge start	S10
BY0E8	Buzzer	BUZZ_ON	Common	When this signal turns ON, the buzzer on the machine sounds. When BY0D8 is ON, even if this signal is turned ON, the buzzer stays muted.	
BY0E9	All LEDs on	LED_ON	Common	When this signal is ON, all of the LEDs below turn on. Use LEDs that can be checked for degradation. Start LED / Stop LED / Pallet start LED / Coolant pump LED / Chip shower LED / Reset LED /Manual operation LED / MDI operation LED / Memory operation LED / Program edit LED / Block skip LED / OPT stop LED / Single operation LED / Machine lock LED / Dry run LED / Zero return LED / Magazine rotation LED / Pallet index LED / Magazine reverse LED / ATC LED / Spindle forward rotation LED / Lathe spindle selection LED / High-speed travel LED / Low-speed travel LED / Step travel LED / Shift LED / Insert LED / Door unlock 1 LED / Door unlock 2 LED (NOTE) When another signal apart from this one is turned on, the LEDs will stay on even if this signal is turned OFF.	
BY0F0	Front door lock	FDRLCK	Common	If the user parameter (switch 1: door) <Door lock method> is set to <1: Lock only when needed>, when the signal FDRLCK is turned ON and the front door is closed, then the front door locks. This signal is different from the “Door lock (DRLOCK)” signal for the external input signal, and the left and right side doors do not lock.	
BY0F1	Reserved	RESERVED	Common	This signal is reserved. Therefore, do not use it.	
BY0F2	Left side door lock	LDRLCK	Common	If the user parameter (switch 1: door) <Door lock method> is set to <1: Lock only when needed>, when the signal LDRLCK is turned ON and the left side door is closed, then the left side door locks.	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BY0F3	Right side door lock	RDRLCK	Common	If the user parameter (switch 1: door) <Door lock method> is set to <1: Lock only when needed>, when the signal RDRLCK is turned ON and the right side door is closed, then the right side door locks.	
BY0F4	Setup chamber left door lock	SLDRLCK	Common	If the user parameter (switch 1: door) <Door lock method> is set to <1: Lock only when needed>, when the signal SLDRLCK is turned ON and the setup chamber left door is closed, then the left side door locks.	
BY0F5	Setup chamber right door lock	SRDRLCK	Common	If the user parameter (switch 1: door) <Door lock method> is set to <1: Lock only when needed>, when the signal SRDRLCK is turned ON and the setup chamber right door is closed, then the setup chamber right door locks.	
BY0F6	External-machine door 1 lock	X1DRLCK	Common	If the user parameter (switch 1: door) <Door lock method> is set to <1: Lock only when needed>, when the signal X1DRLCK is turned ON and outside door 1 is closed, then outside door 1 locks.	
BY0F7	External-machine door 2 lock	X2DRLCK	Common	If the user parameter (switch 1: door) <Door lock method> is set to <1: Lock only when needed>, when the signal X2DRLCK is turned ON and outside door 2 is closed, then outside door 2 locks.	
BY0F8	Reserved	RESERVED	Common	This signal is reserved. Therefore, do not use it.	
BY100	Door unlock (outer door / front door)	FDRULK	Common	If the user parameter (switch 1: door) <Door lock method> is set to <0: Automatic lock when door closes>, when the signal FDRULK is turned ON and the outer door (front door) is locked, then the door unlocks. When the outer door (front door) is closed but unlocked, then the door locks.	
BY101	Reserved	RESERVED	Common	This signal is reserved. Therefore, do not use it.	
BY102	Door unlock (left side door)	LDRULK	Common	If the user parameter (switch 1: door) <Door lock method> is set to <0: Automatic lock when door closes>, when the signal LDRULK is turned ON and the left side door is locked, then the door unlocks. When the left side door is closed but unlocked, then the door locks.	
BY103	Door unlock (left side door)	RDRULK	Common	If the user parameter (switch 1: door) <Door lock method> is set to <0: Automatic lock when door closes>, when the signal RDRULK is turned ON and the right side door is locked, then the door unlocks. When the right side door is closed but unlocked, then the door locks.	
BY104	Door unlock (setup chamber left door)	SLDRULK	Common	If the user parameter (switch 1: door) <Door lock method> is set to <0: Automatic lock when door closes>, when the signal SLDRULK is turned ON and the setup chamber left door is locked, then the door unlocks. When the setup chamber left door is closed but unlocked, then the door locks.	
BY105	Door unlock (setup chamber right door)	SRDRULK	Common	If the user parameter (switch 1: door) <Door lock method> is set to <0: Automatic lock when door closes>, when the signal SRDRULK is turned ON and the setup chamber right door is locked, then the door unlocks. When the setup chamber right door is closed but unlocked, then the door locks.	
BY106	Door unlock (outside door 1)	X1DRULK	Common	If the user parameter (switch 1: door) <Door lock method> is set to <0: Automatic lock when door closes>, when the signal X1DRULK is turned ON and outside door 1 is locked, then the door unlocks. When outside door 1 is closed but unlocked, then the door locks.	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BY107	Door unlock (outside door 2)	X2DRULK	Common	If the user parameter (switch 1: door) <Door lock method> is set to <0: Automatic lock when door closes>, when the signal X2DRULK is turned ON and outside door 2 is locked, then the door unlocks. When outside door 2 is closed but unlocked, then the door locks.	
BY108	Reserved	RESERVED	Common	This signal is reserved. Therefore, do not use it.	
BY10A	Door unlock 1	DRUL1	Common	If the user parameter (switch 1: door) <Door lock method> is set to <0: Automatic lock when door closes>, this signal locks or unlocks the door where the user parameter (switch 1: door) <Door unlock signal setting - *> is set to <1: External input 1> or <3: Both>. When this signal is turned ON, it unlocks all the set doors that are locked. However, it locks all the set doors if just one of the doors is unlocked. This signal has no effect on open doors.	
BY10B	Door unlock 2	DRUL2	Common	If the user parameter (switch 1: door) <Door lock method> is set to <0: Automatic lock when door closes>, this signal locks or unlocks the door where the user parameter (switch 1: door) <Door unlock signal setting - *> is set to <2: External input 2> or <3: Both>. When this signal is turned ON, it unlocks all the set doors that are locked. However, it locks all the set doors if just one of the doors is unlocked. This signal has no effect on open doors.	
BY110 to BY113	Workpiece counter clear 1 Workpiece counter clear 2 Workpiece counter clear 3 Workpiece counter clear 4	WKCLR1 WKCLR2 WKCLR3 WKCLR4	Common	When this signal is ON, the current value on each workpiece counter is initialized. • Signal block  	
BY118	Green indicator	EXGRN	Common	Issues a command to turn the green indicator light ON/OFF. Light up condition: Automatic operation starting (STL) is ON Light off conditions: apart from above Sampling is used for this signal in standard control program 1. In order to use the green indicator signal from another control program, delete the signal that corresponds to standard control program 1 to avoid creating a duplicate circuit, or use the signal (GRNCNV) for BY7F4: green indicator (for converting).	S1
BY119	Yellow indicator	EXYEL	Common	Issues a command to turn the yellow indicator light ON/OFF. Light up conditions (one of the following) • M30/1 or M30/2 is ON. • The user parameter <Count end lamp status> is <Yellow> and the workpiece counter has ended. Light off conditions: apart from above For example, this signal is used in standard control program 1. In order to use the yellow indicator signal from another control program, delete the signal that corresponds to standard control program 1 to avoid creating a duplicate circuit, or use the signal (YELCNV) for BY7F5: yellow indicator (for converting).	S1

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BY11A	Red indicator	EXRED	Common	Issues a command to turn the red indicator light ON/OFF. Light up conditions (one of the following) • An error output (ALM) signal is ON. • The user parameter (switch 1: installation) <Lamp status when count ends> is set to <Red> and the workpiece counter has ended. Light off conditions: apart from above For example, this signal is used in standard control program 1. In order to use the red indicator signal from another control program, delete the signal that corresponds to standard control program 1 to avoid creating a duplicate circuit, or use the signal (REDCNV) for BY7F6: red indicator (for converting).	S1
BY11B	M400	EXM400	Common	Signal turns ON when the M400 command is executed in a program during MDI or memory operation. It turns OFF after the M401 command is issued once the user parameter (switch 1: installation) <Chip shower retention time> has elapsed. It turns OFF even with the reset operation.	S10
BY143	Thermal error (coolant)	CTOHERR	Common	The alarm <<Thermal error (coolant)>> is triggered.	S10
BY144	CTS pressure increase error	CTSUPPER	Common	The alarm <<CTS pressure has not increased>> is triggered.	S10
BY145	CTS pressure decrease error	CTSUNPER	Common	The alarm <<CTS pressure has not lowered>> is triggered.	S10
BY146	Line filter error	CTSLFER	Common	The alarm <<Line filter error>> is triggered.	S10
BY147	Back wash filter error	CTSRVFER	Common	The alarm <<Back wash filter error>> is triggered.	S10
BY14F	Thermal error (Cyclone)	CFOHERR	Common	The alarm <<Thermal error (cyclone coolant drawn up)>> is triggered.	S10
BY150	Tool washing liquid surface is abnormal	TW_LQLV ERR	Common	The alarm <<The tool washing liquid surface is abnormal>> is triggered. After the alarm is triggered, it continues for 100 ms. The stop level changes for the alarm <<The tool washing liquid surface is abnormal>> depending on the setting for the user parameter (switch 1: installation) <Faulty tool wash coolant level alarm stop (Type4)>.	S10
BY152	Chip conveyor error	SCRWSME R	Common	The alarm <<Chip conveyor error>> is triggered.	S10
BY153	Low coolant level	COOLLOW ER	Common	The alarm <<The coolant level has dropped>> is triggered. When the user parameter (switch 1: installation) <Time extension when coolant level has dropped> is set to <1: Enable>, then the alarm <<The coolant level has dropped.>> is triggered after the time extension elapses or when operation stopped during the time extension.	S10
BY154	Chip conveyor manual mode	SCRWMNL ER	Common	The alarm <<It is in the manual mode of chip conveyor>> is triggered.	S10
BY155	Chip conveyor relay error	SCRWRLE R	Common	The alarm <<Chip conveyor relay error>> is triggered.	S10
BY156	Back wash cycle not completed	CTSBW_NF _ERR	Common	The alarm <<Back wash cycle not complete>> is triggered.	S10
BY157	Back wash cycle completed	CTSBW_FI N	Common	Signal turns ON when the back wash cycle is completed	S10
BY15C	Start signal is ON	DR_NOTST A_ERR	Common	The alarm <<Start signal is ON>> is triggered.	
BY15D	Pallet start signal is ON	DR_NOTOS T_ERR	Common	The alarm <<Pallet start signal is ON>> is triggered.	
BY15E	Coil conveyor relay error	COILCONR LER	Comum	The alarm <<Coil conveyor relay error>> is triggered.	S10
BY15F	Coil conveyor thermal error	COILCONT MLER	Comum	The alarm <<Thermal error (coil conveyor motor)>> is triggered.	S10

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BY180 to BY1D9	External error 10 to External error 99	EXER10 to EXER99	Common	When the contact for one of the signals EXER10 through EXER99 is closed, the corresponding alarm between <<External error 10>> and <<External error 99>> is displayed on the screen. The alarm level can be set under the <Stop level> setting for alarms <<External error 10>> through <<External error 99>> as an <External I/O signal>. The following options can be selected: Stop level 1: Display only Step level 2: 1 cycle stop Step level 3: Operation block stop Stop level 4: Temporary operation stop Stop level 5: Servo OFF The cancel level can be set under the <Cancel level> setting for alarms <<External error 10>> through <<External error 99>> as an <External I/O signal>. The following options can be selected: Cancel level 1: Any key operation Cancel level 2: **[RST]** key Cancel level 3: Power OFF required Cancel ALMRLS signal: ALMRLS signal (NOTE) When setting stop levels 3 through 5, it is processed like cancel level 2 even if setting cancel level 1 or 4. • Signal block  	
BY1F0	Auto notification send request	SND_REQ	Common	When ON, a request is made to send the auto notification. When the communication parameter (auto notification function) <Control method> is set to <0: Standard>, turn this signal OFF after the SND_RET turns ON. If the communication parameter (auto notification function) <Control method> is set to <1: Simple>, the NC turns this signal turns OFF after the auto notification is sent and a reply is received.	
BY1FF	Writing request for workpiece coordinate zero	WPWRTWT	NC	When this signal rises, it requests the writing of the workpiece coordinate zero that is set for BDY140 to 151. Refer to the chapter “4.4.1 Workpiece coordinate zero writing” for further details.	
BY200 to BY27F	Macro variable #	#1000 to #1031 #1200 to #1295	NC	Loads the values internally for the macro variables #1000 to #1031 and #1200 to #1295. The ON status is processed as “1” and the OFF status is processed as “0”.	
BY280	CTS discharge valve ON	CTSVON	Common	Controls the ON/OFF for the CTS discharge valve. CTS discharge valve OFF: 0 CTS discharge valve ON: 1	S10
BY281	CTS pump ON	CTSPMPON	Common	Controls the ON/OFF for the CTS pump. CTS pump OFF: 0 CTS pump ON: 1	S10

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BY282	CTS CTS back wash valve ON Cyclone Sludge discharge valve ON	CTSBWVON	Common	CTS Controls the ON/OFF for the CTS back wash valve. CTS back wash valve OFF: 0 CTS back wash valve ON: 1 Cyclone Controls the ON/OFF for the sludge discharge valve. Sludge discharge valve OFF: 0 Sludge discharge valve ON: 1	S10
BY283	Reserved	RESERVED	Common	This signal is reserved. Therefore, do not use it.	
BY284	CTS drain valve ON	CTSDVON	Common	Controls the ON/OFF for the CTS drain valve. CTS drain valve OFF: 0 CTS drain valve ON: 1	S10
BY285	Tool wash / Air blow valve ON	TCVON	Common	Controls the ON/OFF for the tool cleaning valve. Tool cleaning valve OFF: 0 Tool cleaning valve ON: 1	S10
BY286	Jig shower	JIGSHW	Common	Controls the ON/OFF for the jig shower. Jig shower OFF: 0 Jig shower ON: 1	S10
BY287	Tool cleaning blowoff valve B	TBVON	Common	Controls the ON/OFF for the tool cleaning blowoff valve. Tool cleaning blowoff valve B OFF: 0 Tool cleaning blowoff valve B ON: 1	S10
BY288	Reserved	RESERVED	Common	This signal is reserved. Therefore, do not use it.	
BY289	Coolant pump ON	COLPMP	Common	Controls the ON/OFF for the coolant pump. Coolant pump OFF: 0 Coolant pump ON: 1	S10
BY28A	Chip shower pump ON	CHIPPMP	Common	Controls the ON/OFF for the chip shower pump. Chip shower pump OFF: 0 Chip shower pump ON: 1	S10
BY28B	Coolant valve ON M08	M08	Common	Controls the ON/OFF for the coolant valve. Coolant pump OFF: 0 Coolant pump ON: 1	S10
BY28C	Cyclone sludge discharge valve ON	CYCVON	Common	Signal controls the ON/OFF for the cyclone sludge discharge valve.	S10
BY28D	Cyclone suction pump ON	CTSCYCON	Common	Signal controls the ON/OFF for the cyclone suction pump.	S10
BY295	Automatic door open command	DROCMD	Common	Signal issues a command to open the automatic door. 	S10
BY296	Automatic door close command	DRCCMD	Common	Signal issues a command to close the automatic door. 	S10
BY2A0	Indicator, Red	RED	Common	Switches the red indicator light ON/OFF. Light OFF: 0 Light ON: 1	S1
BY2A1	Indicator, Yellow	YEL	Common	Switches the yellow indicator light ON/OFF. Light OFF: 0 Light ON: 1	S1

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BY2A2	Indicator, Green	GRN	Common	Switches the green indicator light ON/OFF. Light OFF: 0 Light ON: 1	S1
BY2A3	Machine lights	LIGHT	Common	Switches the machine light ON/OFF. Light OFF: 0 Light ON: 1 In addition, it operates together with the LED on the operation panel.	S10
BY2A4	Table light	HLIGHT	Common	Signal switches the table light ON/OFF. Light OFF: 0 Light ON: 1 In addition, it operates together with the LED on the operation panel.	
BY2C0	Chip conveyor ON	CHPCON	Common	Issues a command to stop or drive the chip conveyor. Chip conveyor stop: 0 Chip conveyor drive operation: 1	S10
BY2D0	PLC-axis teaching permission	TEACHOK	Common	Signal gives permission to perform PLC-axis teaching. When this signal is ON, the <Grab all> key and <←grab> key on the teaching pendant is enabled and the PLC-axis machine coordinates can be imported (grabbed) to register D.	
BY32E	Coil conveyor forward command	COILCONFW	Common	When this signal turns ON, the coil conveyor operates in a forward direction.	S10
BY32F	Coil conveyor reverse command	COILCONRU	Common	When this signal turns ON, the coil conveyor operates in reverse.	S10
BY340	Transient communication request	TRSDREQ	Common	Signal turns ON when there is a transient communication request.	
BY341	Transient communication type	TRKIND	Common	Signal sets the transient communication type. Reading: 0 Writing: 1	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BY350	Brake load test (all applicable axes)	BTSTAL	Common	After the contact closes for the signal BTSTAL, when it opens, the brake load test is executed for all axes where the input signal SR_BR*RLS (*:1 to 5) is assigned. This is valid in memory operation mode. (NOTE 1) Travel occurs at the positioning speed. (NOTE 2) Refer to “Brake load test” for the test execution order. (NOTE 3) The signal input must be faster than 100 msec. (NOTE 4) Automatic operation starting (STL) does not turn ON during the brake load test. (NOTE 5) The test is not executed on an axis where the machine parameter (system 2: additional axis) <Brake load test> (5th to 8th-axis) and the <Brake load test> (P1 to P4-axis) is set to <0: No test>. (NOTE 6) When the lathe spindle is selected and the test is executed for the lathe spindle, the alarm <<The lathe spindle is being selected.>> is triggered and the test cannot be executed. (NOTE 7) If the PLC-axis test is executed when the PLC-axis control signal is ON, the alarm <<The PLC axis control signal is ON.>> is triggered. <u>Enable conditions</u> (1) Automatic operation starting (STL) is OFF (2) Axis travel is not carried out (3) **[FEED HOLD]** switch is not pressed (4) External stop (EXSTOP) is OFF (5) An alarm higher than a stop level 2 has not been triggered (6) Door is closed (7) Screen <Move start position> is not currently displaying (8) Brake load test executing (BTRUN) signal is OFF	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BY351	Brake load test 1	BTST1	Common	After the contact closes for the signals BTST1 to 4, when it opens, the brake load test is executed for the axis where the input signal SR_BK*RLS (*:1 to 5) is assigned. (NOTE 1) Travel occurs at the positioning speed. (NOTE 2) Refer to “Brake load test” for the test execution order. (NOTE 3) The signal input must be faster than 100 msec. (NOTE 4) Automatic operation starting (STL) does not turn ON during the brake load test. (NOTE 5) The test is not executed on an axis where the machine parameter (system 2: additional axis) <Brake load test> (5th to 8th-axis) and the <Brake load test> (P1 to P4-axis) is set to <0: No test>. (NOTE 6) When the lathe spindle is selected and the test is executed for the lathe spindle, the alarm <<The lathe spindle is being selected.>> is triggered and the test cannot be executed. (NOTE 7) If the PLC-axis test is executed when the PLC-axis control signal is ON, the alarm <<The PLC axis control signal is ON.>> is triggered. Enable conditions (1) Automatic operation starting (STL) is OFF (2) Axis travel is not carried out (3) **[FEED HOLD]** switch is not pressed (4) External stop (EXSTOP) is OFF (5) An alarm higher than a stop level 2 has not been triggered (6) Door is closed (7) Screen <Move start position> is not currently displaying (8) Brake load test executing (BTRUN) signal is OFF This is valid in memory operation mode.	
BY352	Brake load test 2	BTST2	Common		
BY353	Brake load test 3	BTST3	Common		
BY354	Brake load test 4	BTST4	Common		
BY355	Brake load test 5	BTST5	Common		
BY375	Specified axis position 1	SEL_POSSW1	Common	BX010 to BX017 signals turn ON when this signal is OFF and when the machine coordinates fall within the range indicated in user parameter (switch 2: position signal output range) <*-axis position signal output range (-)/(+)>. BX010 to BX017 signals turn ON when this signal is ON and when the machine coordinates fall within the range indicated in user parameter (switch 2: position signal output range) <*-axis position signal output range 3 (-)/(+)>.	
BY376	Specified axis position 2	SEL_POSSW2	Common	BX018 to BX01F signals turn ON when this signal is OFF and when the machine coordinates fall within the range indicated in user parameter (switch 2: position signal output range) <*-axis position signal output range 2 (-)/(+)>. BX018 to BX01F signals turn ON when this signal is ON and when the machine coordinates fall within the range indicated in user parameter (switch 2: position signal output range) <*-axis position signal output range 3 (-)/(+)>.	

<table>
<tr><th>OM name</th><th>Name</th><th>Symbol</th><th>Conversation / NC</th><th>Signal description</th><th>Use</th></tr>
<tr><td>BY378
BY379
BY37A</td><td>PLC rapid traverse override</td><td>PLC_RPDOR0
PLC_RPDOR1
PLC_RPDOR2</td><td>Common</td><td>This signal sets the rapid traverse override value from the PLC.

<table><tr><th>PLC_RPDOR2</th><th>PLC_RPDOR1</th><th>PLC_RPDOR0</th><th>Override value</th></tr><tr><td>OFF</td><td>OFF</td><td>OFF</td><td>100%</td></tr><tr><td>OFF</td><td>OFF</td><td>ON</td><td>Speed 4</td></tr><tr><td>OFF</td><td>ON</td><td>OFF</td><td>Speed 3</td></tr><tr><td>OFF</td><td>ON</td><td>ON</td><td>Speed 2</td></tr><tr><td>ON</td><td>OFF</td><td>OFF</td><td>Speed 1</td></tr></table>Override operates at 100% when a value that is out of range is set.

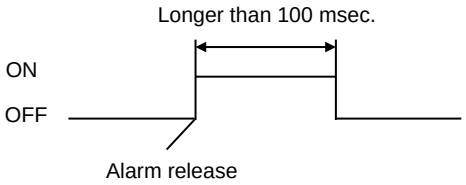
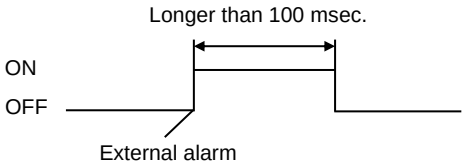
Signal can be used when the user parameter (switch 1: installation) <PLC rapid traverse override function> is set to <1: Enable>.</td><td>S1</td></tr>
<tr><td>BY37B</td><td>PLC rapid traverse override IN</td><td>PLC_RPDORIN</td><td>Common</td><td>When this signal is turned ON, the PLC rapid traverse override value becomes fixed.
When this signal is turned ON before changing the PLC rapid traverse override value, turn this signal OFF after changing the override value.</td><td></td></tr>
<tr><td>BY380
to
BY3DF</td><td>Reserved</td><td>RESERVED</td><td>Common</td><td>This signal is reserved.
Therefore, do not use it.</td><td></td></tr>
<tr><td>BY400</td><td>LB internal start input</td><td>LB_STRT_IN</td><td>Common</td><td>This signal is the start signal to the loading system. After the signal is ON for at least 100 ms, the loading system starts at the falling edge of the signal.
(NOTE 1) The loading system does not start when the teaching pendant or the LB Manager has operation control.
(NOTE 2) A program can be started by using the LB internal program selection* input (BY430 to BY435). If the LB internal program selection* input does not input, the program currently selected in the loading system starts up.</td><td></td></tr>
<tr><td>BY401</td><td>LB internal stop input</td><td>LB_STOP_IN</td><td>Common</td><td>This signal is a pause signal to the loading system. Turn ON this signal for 100 ms or longer.</td><td></td></tr>
<tr><td>BY407
BY408</td><td>LB coordinate output selection 1
LB coordinate output selection 2</td><td>LB_CORD1SEL
LB_CORD2SEL</td><td>Common</td><td>This signal selects the loading system coordinate (tool 0 position, TCP position in the tool coordinate that is selected and joint coordinate) to be referenced between BDX256 and BDX263.

<u>When referencing the coordinate as the “Tool 0 position”</u>
LB_CORD1SEL2: OFF,
LB_CORD1SEL1: OFF

<u>When referencing the coordinate as the “TCP position in the tool coordinate that is selected ”</u>
LB_CORD1SEL2: OFF,
LB_CORD1SEL1: ON

<u>When referencing the coordinate as the “Joint coordinate”</u>
LB_CORD1SEL2: ON,
LB_CORD1SEL1: OFF

(NOTE) When one of the above combinations does not apply, BDX256 to BDX263 become an undefined value.</td><td></td></tr>
</table>

<table>
<tr><th>OM name</th><th>Name</th><th>Symbol</th><th>Conversation / NC</th><th>Signal description</th><th>Use</th></tr>
<tr><td>BY409</td><td>LB external alarm release</td><td>LB_ALMRLS</td><td>Common</td><td>When LB_ALMRLS turns ON, the <LS_ALMRLS signal> external alarm can be released for a loading system stop level 1 and 2 and a cancel level 4.
 
</td><td></td></tr>
<tr><td>BY410
~BY423</td><td>LB external alarm 10 to 29</td><td>LB_EXER10
~LB_EXER29</td><td>Common</td><td>When a signal between LB_EXER10 and LB_EXER29 turns ON, the corresponding alarm between <<External alarm 10>> and <<External alarm 29>> is triggered on the loading system. The stop and cancel levels of each external alarm can be set in the user parameter for the loading system.
(NOTE) When setting stop levels 3 through 5, it is processed like cancel level 2 even if setting cancel level 1 or 4.
 
</td><td></td></tr>
<tr><td>BY430
BY431
BY432
BY433
BY434
BY435
BY436
BY437</td><td>LB internal program selection 1 input
LB internal program selection 2 input
LB internal program selection 4 input
LB internal program selection 8 input
LB internal program selection 16 input
LB internal program selection 32 input
LB internal program selection 64 input
LB internal program selection 128 input</td><td>LB_PRO1_IN
LB_PRO2_IN
LB_PRO4_IN
LB_PRO8_IN
LB_PRO16_IN
LB_PRO32_IN
LB_PRO64_IN
LB_PRO128_IN</td><td>Common</td><td>This signal is used to specify a program number for the loading system to start in automatic operation mode. LB_PRO1_IN, LB_PRO2_IN, LB_PRO4_IN, LB_PRO8_IN, LB_PRO16_IN, LB_PRO32_IN, LB_PRO64_IN and LB_PRO128_IN load into the loading system at the falling edge of the LB internal start input (BY400) or the LB external start. However, maintain that status until the LB automatic operation starting (LB_STL) status turns ON in order to ensure a program is specified.
<table>
<tr><th rowspan="2">Specified program No.</th><th colspan="8">LB_PRO*_IN</th></tr>
<tr><th>128</th><th>64</th><th>32</th><th>16</th><th>8</th><th>4</th><th>2</th><th>1</th></tr>
<tr><td>Currently selected program</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>1</td><td>OFF</td><td>OFF</td><td>OFF</td><td>OFF</td><td>OFF</td><td>OFF</td><td>OFF</td><td>ON</td></tr>
<tr><td>2</td><td>OFF</td><td>OFF</td><td>OFF</td><td>OFF</td><td>OFF</td><td>OFF</td><td>ON</td><td>OFF</td></tr>
<tr><td>3</td><td>OFF</td><td>OFF</td><td>OFF</td><td>OFF</td><td>OFF</td><td>OFF</td><td>ON</td><td>ON</td></tr>
<tr><td>・</td><td>・</td><td>・</td><td>・</td><td>・</td><td>・</td><td>・</td><td>・</td><td>・</td></tr>
<tr><td>127</td><td>OFF</td><td>ON</td><td>ON</td><td>ON</td><td>ON</td><td>ON</td><td>ON</td><td>ON</td></tr>
<tr><td>128</td><td>ON</td><td>OFF</td><td>OFF</td><td>OFF</td><td>OFF</td><td>OFF</td><td>OFF</td><td>OFF</td></tr>
</table>
(NOTE 1) The alarm <<LB internal program selection input error>> is triggered if any of the following applies.
• When the combinations in the above table for the specified program number do not apply
• When the start conditions for the specified program number is not set to <Normal>
(NOTE 2) When a program with <Normal> start conditions is paused, even if the specified program number is the program currently selected or No. 1 to 32128, the program is invalid.</td><td></td></tr>
</table>

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BY440	LB brake load test 1	LBBTS1	Common	Execute the brake load test on the J*-axis (*: 1, 2 or 3) in the loading system. Refer to the loading system manual for further details. Conditions to enable loading system (1) LB automatic operation starting (LB_STL) is OFF (2) Axis travel is not carried out (3) **[Stop]** key is not pressed (not paused) (4) LB external stop (BIO_EXSTOP) is OFF (5) An alarm higher than a stop level 2 has not been triggered (6) Door guard is closed Ex: Side door and safety fence that separate the loading system and SPEEDIO are closed (7) No display device or NC operation panel with operation control (8) Brake load test executing (LB_BTRUN) signal is OFF In addition, keep the LB brake load test 1 to 3 signals in a 1: ON state for at least 100 ms. This is valid in memory operation mode.	
BY441	LB brake load test 2	LBBTS2	Common		
BY442	LB brake load test 3	LBBTS3	Common		
BY480 ~BY4FF	LB internal general input 0 to 127	LB_BYGNL 000 ~LB_BYGN L127	Common	This signal is a general input signal to the loading system.	
BY700 to BY71F	Delete output signal 1 to 32	BYRV1~32	Common	Signal is used for conversion of an OM that was deleted in D00.	
BY7E0	Change chip shower control	CHIPCTL	Common	Changes the chip shower control. When ON, if the **[CHP.F]** key is turned ON even when the M400 signal command is not issued, the chip shower discharges when starting in memory operation and MDI operation mode.	S1 S10
BY7E1	Change chip shower control_Door	CHIPCTL_ DR	Common	This signal is reserved.	S10
BY7E2	Disable coolant level drop alarm	CLLWERC NCL	Common	This signal controls the alarm <<The coolant level has dropped>>. When the signal is ON, the alarm <<The coolant level has dropped>> is disabled. Set up a circuit to trigger a substitute alarm in one of the standard control programs (1 to 9). The default setting for this signal is OFF. (NOTE) When using this signal, set the user parameter (switch 1: installation) <Time extension when coolant level has dropped> to <0: Disable>.	S10
BY7E3	Disable chip conveyor manual mode alarm	SCRWMNL ERCNCL	Common	This signal controls the alarm <<It is in the manual mode of chip conveyor>>. When the signal is ON, the alarm <<It is in the manual mode of chip conveyor>> is disabled. Set up a circuit to trigger a substitute alarm in one of the standard control programs (1 to 9). The default setting for this signal is OFF.	S10
BY7E4	Disable startup output	STACNCL	Common	Enables or disables the startup from BX12A<STA>. When this signal is ON, startup from <STA> is disabled. The default setting for this signal is OFF.	S10
BY7E5	Disable pallet start output	QTSTACN CL	Common	Enables or disables the pallet startup from BX12D<QTSTART>. When this signal is ON, pallet startup from <QTSTART> is disabled. The default setting for this signal is OFF.	S10

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BY7E6	Automatic door - automatic mode open disabled	ADROPCNCL	Common	Enables or disables the automatic door to open when the **[MANU - AUTO]** switch is set to <AUTO>. When this signal is ON, the automatic door open operation is disabled. The default setting for this signal is always OFF.	S10
BY7E7	Coolant valve external control	EXM08	Common	Controls the coolant valve externally. When the signal is ON, the coolant valve turns ON. When the signal is OFF, the coolant valve turns OFF. The factory default setting for this signal is always OFF. (NOTE 1) Modal M08 does not enable with this signal. In addition, the coolant valve does not turn OFF even if this signal turns OFF while modal M08 is enabled. (NOTE 2) Even when this signal turns ON, the following restrictions apply. • The coolant valve does not turn OFF during a tool change when the machine parameter (system 1: common) <Coolant during tool change> is set to <0: Stop> and the M9 modal is being used. • The coolant valve does not turn ON if the tool breakage detection (Type 3 only) is activated when the <Tool breakage detection coolant OFF> is set to <1: Yes> while in conversation language mode. • The coolant valve does not turn ON when BX1AA <M08ALIL> is ON. • The coolant valve turns OFF even if this signal stays ON when the door is opened.	S10
BY7E8	Coil conveyor external control	EXCOILCON	Common	The coil conveyor is controlled externally. When this signal turns ON, the coil conveyor operates in a forward direction. When this signal turns OFF, coil conveyor operation is stopped. When using the coil conveyor option, set up this signal's control and the processing to turn ON the BY3E9 <COILCONCNCL> in one of the standard control programs (1 to 9). The factory default setting for this signal is always OFF.	S10
BY7E9	Coil conveyor automatic mode: Disable	COILCONCNCL	Common	This signal enables and disables the forward command in automatic mode for the coil conveyor. When this signal turns ON, automatic mode for coil conveyor is disabled and the conveyor does not operate. When using the coil conveyor option, set up the processing to turn ON this signal and the control for BY3E8 <EXCOILCON> in one of the standard control programs (1 to 9). The factory default setting for this signal is always OFF.	S10
BY7EB	LB start output OFF	LB_STACNC	Common	Start input from BX390 is enabled and disabled. When this signal is ON, startup from BY08B is disabled. The factory default setting for this signal is always OFF.	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BY7ED	LB reset output OFF	LB_RSTCNCL	Common	Reset notification to the loading system is enabled and disabled. When this signal is ON, BY08D reset notification is disabled. The factory default setting for this signal is always OFF.	
BY7EE	Coolant pump disable	COLPMPCNCL	Common	Signal enables/disables the coolant pump. When this signal turns ON, BY289 (coolant pump ON) is disabled and the coolant pump does not operate. The factory default setting for this signal is always OFF.	S10
BY7EF	Cyclone filter suction pump disable	CTSCYCONCNCL	Common	Signal enables/disables the cyclone filter suction pump. When this signal turns ON, BY28D (cyclone filter suction pump ON) is disabled and the cyclone filter suction pump does not operate. The factory default setting for this signal is always OFF.	S10
BY7F0	Start output (for conversion)	STACNV	Common	Outputs the start signal. The signal is integrated so that it can be used as a contact for the startup (BY088) in standard control program 10. Use it to start up from a program that is not standard control program 10.	S10
BY7F1	Pallet start output (for conversion)	QTSTACNV	Common	Outputs the pallet start signal. The signal is integrated so that it can be used as a contact for the external startup (BY08A) in standard control program 10. Use it for an external start up from a program that is not standard control program 10.	S10
BY7F2	Automatic door open (for conversion)	ADROPCNV	Common	Outputs the automatic door open signal. Signal is used for converting the ladder program (previously used in C00) to use with D00.	S10
BY7F3	Automatic door close (for conversion)	ADRCLCNV	Common	Outputs the automatic door close signal. Signal is used for converting the ladder program (previously used in C00) to use with D00.	S10
BY7F4	Green indicator (for conversion)	GRNCNV	Common	Outputs the light ON signal for the green indicator. The signal is integrated so that it can be used as a contact for the green indicator (BY118) in standard control program 1. Use it to issue a command from a program that is not standard control program 1.	S1
BY7F5	Yellow indicator (for conversion)	YELCNV	Common	Outputs the light ON signal for the yellow indicator. The signal is integrated so that it can be used as a contact for the yellow indicator (BY119) in standard control program 1. Use it to issue a command from a program that is not standard control program 1.	S1
BY7F6	Red indicator (for conversion)	REDCNV	Common	Outputs the light ON signal for the red indicator. The signal is integrated so that it can be used as a contact for the red indicator (BY11A) in standard control program 1. Use it to issue a command from a program that is not standard control program 1.	S1
BY7F7	LB internal reset input (for conversion)	LB_RSTCNV	Common	Reset signal to the loading system is output.	
BY7F8	External reset (alternative)	EXRSTSUB	Common	External reset signal outputs. The signal is integrated so that it can be used as a contact for the external reset (BY080) in standard control program 1. Use it to issue a command from a program that is not standard control program 1.	S1

4.2.5 Internal Input Data (NC→PLC, BDX0 to BDX1023)

4.2.5.1 Overview

This signal is from the NC→PLC.
It indicates the internal status of the NC using data.
The internal input data is comprised of 16 bits per 1 point, and it is read in 16-bit units.
When handling 32-bit data, 2 points are used. The data register number that is specified in a 32-bit command is the low-order 16-bit, and the data register + 1 that is specified is the high-order 16-bit.

OM name	BDX0 to BDX1023 (decimal shown)
Total number of signals	1024
Type	Word (16-bit)
Output specification	Available / None
Double coil check	-
Record when power is OFF	-
Index addressing	Available

(NOTE) The output specification is available, however overwriting is performed randomly by the NC.

4.2.5.2 Signal List

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BDX000 to BDX015	Machine coordinate X-axis, low-order: 0-15 bit X-axis, high-order: 16-31 bit Y-axis, low-order Y-axis, high-order Z-axis, low-order Z-axis, high-order No 4-axis, low-order No 4-axis, high-order No 5-axis, low-order No 5-axis, high-order No 6-axis, low-order No 6-axis, high-order No 7-axis, low-order No 7-axis, high-order No 8-axis, low-order No 8-axis, high-order	 MCNCDXL MCNCDXH MCNCDYL MCNCDYH MCNCDZL MCNCDZH MCNCD4L MCNCD4H MCNCD5L MCNCD5H MCNCD6L MCNCD6H MCNCD7L MCNCD7H MCNCD8L MCNCD8H	Common	The current machine coordinates on each axis are set. The unit is saved to the smallest unit.	
BDX016 to BDX031	Feedback machine coord. X-axis, low-order: 0-15 bit X-axis, high-order: 16-31 bit Y-axis, low-order Y-axis, high-order Z-axis, low-order Z-axis, high-order No 4-axis, low-order No 4-axis, high-order No 5-axis, low-order No 5-axis, high-order No 6-axis, low-order No 6-axis, high-order No 7-axis, low-order No 7-axis, high-order No 8-axis, low-order No 8-axis, high-order	 FBCDXL FBCDXH FBCDYL FBCDYH FBCDZL FBCDZH FBCD4L FBCD4H FBCD5L FBCD5H FBCD6L FBCD6H FBCD7L FBCD7H FBCD8L FBCD8H	Common	The current feedback machine coordinates on each axis are set. The unit is saved to the smallest unit.	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BDX032 to BDX047	Skip coordinate position X-axis, low-order X-axis, high-order Y-axis, low-order Y-axis, high-order Z-axis, low-order Z-axis, high-order No 4-axis, low-order No 4-axis, high-order No 5-axis, low-order No 5-axis, high-order No 6-axis, low-order No 6-axis, high-order No 7-axis, low-order No 7-axis, high-order No 8-axis, low-order No 8-axis, high-order	 SKIPXL SKIPXH SKIPYL SKIPYH SKIPZL SKIPZH SKIP4L SKIP4H SKIP5L SKIP5H SKIP6L SKIP6H SKIP7L SKIP7H SKIP8L SKIP8H	Common	The skip coordinates on each axis are set. The unit is saved to the smallest unit.	
BDX048 to BDX055	Torque command X-axis Y-axis Z-axis No 4-axis No 5-axis No 6-axis No 7-axis No 8-axis	 OVLDX OVLDY OVLDZ OVLD4 OVLD5 OVLD6 OVLD7 OVLD8	Common	The torque command for each axis is set. The unit is 0.1%.	
BDX056 to BDX063	Torque monitor X-axis Y-axis Z-axis No 4-axis No 5-axis No 6-axis No 7-axis No 8-axis	 TRQMNTX TRQMNTY TRQMNTZ TRQMNT4 TRQMNT5 TRQMNT6 TRQMNT7 TRQMNT8	Common	The torque monitor on each axis is set. The unit is 0.1%.	
BDX064 to BDX071	Machine coordinate P1-axis, low-order P1-axis, high-order P2-axis, low-order P2-axis, high-order P3-axis, low-order P3-axis, high-order P4-axis, low-order P4-axis, high-order	 MCNCDP1L MCNCDP1H MCNCDP2L MCNCDP2H MCNCDP3L MCNCDP3H MCNCDP4L MCNCDP4H	Common	The current machine coordinates on each axis are set. The unit is saved to the smallest unit. Refer to the “Chapter 10 PLC-axis” for further details.	
BDX072 to BDX079	Travel remaining P1-axis, low-order P1-axis, high-order P2-axis, low-order P2-axis, high-order P3-axis, low-order P3-axis, high-order P4-axis, low-order P4-axis, high-order	 RMDTP1L RMDTP1H RMDTP2L RMDTP2H RMDTP3L RMDTP3H RMDTP4L RMDTP4H	Common	The current travel remaining on each axis is set. The unit is saved to the smallest unit. Refer to the “Chapter 10 PLC-axis” for further details.	
BDX080 to BDX083	Alarm / Warning P1-axis P2-axis P3-axis P4-axis	 ALMP1 ALMP2 ALMP3 ALMP4	Common	The alarm / warning number on each axis is set. Refer to the “Chapter 10 PLC-axis” for further details.	
BDX084 to BDX087	Servo error P1-axis P2-axis P3-axis P4-axis	 SVALP1 SVALP2 SVALP3 SVALP4	Common	The servo error number on each axis is set. Refer to “Chapter 2 – 2.13 Servo error list” in the Data Bank & Alarm Manual for further details about the servo error numbers. If a servo error is triggered, all axes are stopped.	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BDX088 to BDX091	Torque command P1-axis P2-axis P3-axis P4-axis	 OVLDP1 OVLDP2 OVLDP3 OVLDP4	Common	The torque command for each axis is set. The unit is 0.1%.	
BDX092 to BDX095	Torque monitor value P1-axis P2-axis P3-axis P4-axis	 TRQMNTP1 TRQMNTP2 TRQMNTP3 TRQMNTP4	Common	The torque monitor value on each axis is set. The unit is 0.1%.	
BDX096 to BDX097	Spindle feedback position Low-order High-order	 FBCDSL FBCDSH	Common	The spindle feedback machine coordinates is set. The unit is 0.001 degrees. (NOTE) After the power is turned ON, the value is undefined until the spindle rotation is carried out.	
BDX098 to BDX099	Spindle rotation speed Low-order High-order	 SPRTRGL SPRTRGH	Common	The actual rotation speed of the spindle is set. The unit is min^{-1}.	
BDX100	Spindle torque command	OVLDS	Common	The spindle torque command is set. The unit is 0.1%.	
BDX101	Spindle torque monitor value	TRQMNTS	Common	The torque monitor value on the spindle is set. The unit is 0.1%.	
BDX106 to BDX107	Turning spindle rotation speed Low-order High-order	 LSRTRGL LSRTRGH	NC	The actual rotation speed for the turning spindle is set when there is a turning spindle. The unit is min^{-1}.	
BDX108 to BDX109	Actual feedrate Low-order High-order	 FBFDL FBFDH	Common	Command value is set for the actual feedrate. The unit depends on the machine unit system and the minimum unit setting. However, the value for the feedrate per minute is set regardless of the modal status. (NOTE) 0 (zero) is set during rapid feed.	
BDX110	Machine coordinate update counter	MCNCDCNT	Common	Before updating all axes in the machine coordinates for BDX000 to BDX015, the counter increases to an odd number. In addition, after being updated, the counter increases to an even number. Odd number: Machine coordinates being updated Even number: Machine coordinates are updated When the counter value is 32767, the next value after the increase is -32768.	
BDX111	Feedback machine coordinate update counter	FBCDCNT	Common	Before updating all axes in the feedback machine coordinates for BDX016 to BDX031, the counter increases to an odd number. After being updated, the counter increases to an even number. Odd number: Feedback machine coordinates being updated Even number: Feedback machine coordinates are updated When the counter value is 32767, the next value after the increase is -32768.	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BDX112	Main input signal No.	IN_SIGNAL_NO	Common	The main input signal No., which outputs between BDX360 and BX36F, is set. Refer to “6.2.1.3. Signal assignment” in the Installation Manual for details on the main input signal No.	
BDX116	Feedrate override	FEDOR	Common	The **[RAPID TRAVERSE OVERRIDE]** switch value is set. (NOTE) When the **[FEEDRATE OVERRIDE]** switch is an intermediate value, the value before turning into an intermediate value is maintained.	S1
BDX117	Spindle override	SPNOR	Common	The **[SPINDLE OVERRIDE]** switch value is set. (NOTE) When the **[SPINDLE OVERRIDE]** switch is an intermediate value, the value before turning into an intermediate value is maintained.	S1
BDX118 to BDX122	Reserved	RESERVED	Common	This signal is reserved. Therefore, do not use it.	
BDX124 to BDX125	Control unit internal temperature Low-order High-order	 THRML THRMH	Common	The control unit’s internal temperature is set. The set value is compensated by the machine parameter (system 1) <CPU temperature compensation>.	
BDX126	Teaching pendant point number	TPPNT	Common	Use the up and down keys to change the grab point. Or, use the <Point No.> key to specify the point number directly. The point number range is between 0 and 99. The point number is set by BDX126. To grab multiple points, refer to the point number (BDX126) in the ladder program, and create a PLC program that can change between the OMs (BDY028 to BDY031).	
BDX128	Operation program number	RNPGNO	Common	The program number that is currently selected is set.	
BDX130 to BDX131	S code data Low-order High-order	 SDATAL SDATAH	Common	When using the NC language, the S modal value is set. When using the Conversation language, the command value for the spindle rotation command that is issued last is set.	
BDX132 to BDX133	T code data Low-order High-order	 TDATAL TDATAH	Common	When using the NC language, the T modal value is set. When using the Conversation language, the command value for the tool change number that is issued last is set.	
BDX134 to BDX135	F code data Low-order High-order	 FDATAL FDATAH	Common	When using the NC language, the F modal value is set. When using the Conversation language, the command value for the feedrate command that is issued last is set. The unit depends on the smallest unit and on the G94 modal.	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BDX139	Current operation level	OPLEVEL	Common	Signal outputs the current operation level. 0 to 5	
BDX140	Current screen selection	DSPSEL	Common	Signal outputs the current screen selection. 0: Display OFF 1: Alarm screen 2: Data bank screen 3: Magazine tool screen 4: Program screen 5: Manual conditions screen 6: Position screen 7: Input/Output screen 8: Production monitor screen 9: Graph screen	
BDX141	Running program number	STPGNO	Common	Program number is set for the program currently running. When the timing is before operation, the program number that is currently selected is set.	
BDX142 to BDX143	Running block number/job number Low-order High-order	 STBKNOL STBKNOH	Common	Block number/job number is set for the program currently running. When the timing is before operation, <0> is set in the NC language and <1> is set in the conversation language.	
BDX144	Specified tool Tool number	TLNO	Common	The number is set for the tool specified by BDY120 or for the tool that is currently mounted on the magazine. 0 is set when a tool is not mounted, or when the tool that is mounted uses a different language (NC / Conversation).	
BDX145	Specified tool Data flag / Status	TLDAT	Common	The extension information is set for the tool set by BDX144 or for the tool that is currently mounted on the magazine. The following bits are supported. Bit0: Tool cleaning 0: Available, 1: Not available Bit1: CTS 0: Available, 1: Not available 0 is set when a tool is not mounted, or when the tool that is mounted uses a different language (NC / Conversation).	
BDX146 to BDX147	Specified tool Group number Low-order High-order	 TLGRPL TLGRPH	Common	The group number is set for the tool set by BDX144 or for the magazine tool. When using Conversation, the “Main tool” is set. 0 is set when the spindle tool is not set, not mounted, or when the language (NC / Conversation) is different.	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BDX148 to BDX149	Specified tool Current tool life data Low-order High-order	 TLADTL TLADTH	Common	The current tool life is set for the tool set by BDX144 or for the tool that is mounted on the magazine. If the tool is not set, not mounted, or if the language (NC / conversation language) is different and the user parameter (switch 1: data) <Tool life counting method> is set to <0: Method 1>: When the language is NC, 999999 is set. When the language is conversation, 999999 is set. <1: Method 2>: 0 is set.	
BDX150 to BDX151	Specified tool Tool life data (Initial tool life / End of tool life) Low-order High-order	 TLLFTL TLLFTH	Common	The tool life data (initial tool life / end of tool life values) is set for the tool set by BDX144 or for the tool that is mounted on the magazine. If the tool is not set, not mounted, or if the mounted tool has a different language (NC / conversation language), and the user parameter (switch 1: data) <Tool life counting method> is set to <0: Method 1>: 0 is set. <1: Method 2>: When the language is NC, 999999 is set. When the language is conversation, 999999 is set.	
BDX152 to BDX153	Specified tool Tool length offset data (Z) Low-order High-order	 TLLGTL TLLGTH	Common	The tool length offset (Z-axis) is set for the tool set by BDX144 or for the magazine tool. 0 is set when the tool is not set, not mounted, or when the language (NC / Conversation) is different.	
BDX154 to BDX155	Specified tool Tool diameter / nose R compensation data Low-order High-order	 TLDIAL TLDIAH	Common	The cutter/nose R compensation data is set for the tool set by BDX144 or for the magazine tool. 0 is set when the tool is not set, not mounted, or when the language (NC / Conversation) is different.	
BDX156 to BDX157	Mounted tool Tool length wear data (Z) Low-order High-order	 TLLGWL TLLGWH	Common	The tool length wear compensation (Z-axis) is set in the NC language for the tool set by BDX144 or for the magazine tool. The cutting position compensation data is set in the conversation language. 0 is set when the tool is not mounted, or when the language (NC / Conversation) is different.	
BDX158 to BDX159	Specified tool Tool diameter / nose R wear data Low-order High-order	 TLDIWL TLDIWH	Common	The cutter compensation/nose R wear offset is set in the NC language for the tool set by BDX144 or for the magazine tool. The cutting position compensation data is set in the conversation language. 0 is set when the tool is not mounted, or when the language (NC / Conversation) is different.	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BDX160 to BDX161	Specified tool Maximum spindle speed Low-order High-order	 TLMXRL TLMXRH	NC	<Max. speed> in the tool list (extension information) is set for the tool set by BDX144 or for the magazine tool. 0 is set when the tool is not mounted, or when the language (NC / Conversation) is different.	
BDX162 to BDX163	Specified tool Rotation feed Low-order High-order	 TLRTFDL TLRTFDH	NC	<Feed per rotation> value in the tool list (extension information) is set for the tool set by BDX144 or for the magazine tool. 0 is set when the tool is not mounted, or when the language (NC / Conversation) is different.	
BDX164 to BDX165	Specified tool Peripheral speed Low-order High-order	 TLPSRTL TLPSRTH	NC	<Peripheral speed> value in the tool list (extension information) is set for the tool set by BDX144 or for the magazine tool. 0 is set when the tool is not mounted, or when the language (NC / Conversation) is different.	
BDX166 to BDX167	Specified tool Feedrate Low-order High-order	 TLFDL TLFDH	NC	<F command value(G94)> in the tool list (extension information) is set for the tool set by BDX144 or for the magazine tool. 0 is set when the tool is not mounted, or when the language (NC / Conversation) is different.	
BDX168 to BDX169	Specified tool Spindle speed Low-order High-order	 TLRTSPL TLRTSPH	NC	<S command value(G97)> in the tool list (extension information) is set for the tool set by BDX144 or for the magazine tool. 0 is set when the tool is not mounted, or when the language (NC / Conversation) is different.	
BDX170 to BDX171	Specified tool Tool position compensation data (X) Low-order High-order	 TLOFSXL TLOFSXH	NC	Tool position compensation (X-axis) is set for the tool set by BDX144 or for the magazine tool. 0 is set when the tool is not set, not mounted, or when the language (NC / Conversation) is different.	
BDX172 to BDX173	Specified tool Tool position compensation data (Y) Low-order High-order	 TLOFSYL TLOFSYH	NC	Tool position compensation (Y-axis) is set for the tool set by BDX144 or for the magazine tool. 0 is set when the tool is not set, not mounted, or when the language (NC / Conversation) is different.	
BDX174 to BDX175	Specified tool Tool position wear offset data (X) Low-order High-order	 TLOFSWXL TLOFSWXH	NC	Tool position wear offset (X-axis) is set for the tool specified by BDX144 or for the magazine tool. 0 is set when the tool is not mounted, or when the language (NC / conversation language) is different.	
BDX176 to BDX177	Specified tool Tool position wear offset data (Y) Low-order High-order	 TLOFSWYL TLOFSWYH	NC	Tool position wear offset (Y-axis) is set for the tool set by BDX144 or for the magazine tool. 0 is set when the tool is not mounted, or when the language (NC / Conversation) is different.	
BDX178	Specified tool Virtual teeth direction data	TLNOSE	NC	Virtual teeth direction is set for the tool set by BDX144 or for the magazine tool. 0 is set when the tool is not mounted, or when the language (NC / Conversation) is different.	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BDX180 to BDX181	Magazine remaining distance Low-order High-order	 RMDTMZL RMDTMZH	Common	Current travel distance remaining on the magazine axis is set. This depends on the machine unit system and the minimum unit setting.	
BDX182	Writing counter for workpiece coordinate zero	WPWRTCNT	NC	Counter counts up regardless of the writing result when the workpiece coordinate zero writing (requested by BY1FF) has ended. BX1B0 turns ON when writing is not possible. When the counter value is 32767, the next value after the increase is -32768.	
BDX184	Tool change magazine No.	TLNEXT	Common	The magazine number is set for the tool change when a G100/M06 command is issued during memory operation. When using a turret type ATC mechanism, the magazine number is set during the following conditions. • When rising to between the Z-axis zero point and ATC zero point • During magazine rotation 0 is set when the machine is in standby or when a tool change is completed.	
BDX185	Parameter number for machining load monitor	TLOADNO	Common	Parameter number currently being measured is set in the machining load monitor function.	
BDX186 to BDX187	Current machining load Low-order High-order	 TLOADL TLOADH	Common	Current machining load measurement value that is being measured is set in the machining load monitor function. The set value is the machining load (peak value) calculated using a moving average with the <Time constant> for the machining load monitor parameter being used. The unit is 0.001 Nm.	
BDX188 to BDX189	Current machining load 2 Low-order High-order	 TLOAD2L TLOAD2H	Common	The measurement value for the machining load is set before the moving average is applied. The set value is the machining load before being applying a moving average with the <Time constant> for the machining load monitor parameter. The value is always set regardless if the machining load monitor function is ON or OFF. The unit is 0.001 Nm.	
BDX191	Transient communication result	TRRESULT	Common	This data is taken from the transient communication results. When normal: 0 When there is an error: Error code	
BDX192 to BDX193	#1132 Low-order High-order	 1132L 1132H	NC	The value for collective signal writing (#1100 to #1131) (32-bit) is set.	
BDX194 to BDX195	#1133 Low-order High-order	 1133L 1133H	NC	The value for collective signal writing (#1300 to #1331) (32-bit) is set.	
BDX196 to BDX197	#1134 Low-order High-order	 1134L 1134H	NC	The value for collective signal writing (#1332 to #1363) (32-bit) is set.	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BDX198 to BDX199	#1135 Low-order High-order	 1135L 1135H	NC	The value for collective signal writing (#1364 to #1395) (32-bit) is set.	
BDX201	Automatic door opening operation check time	DROPCK_T	Common	Signal sets the value from the machine parameter (system 1: automatic door) <Door opening operation check time>.	
BDX202	Tool cleaning control	TW_CONTROL	Common	The discharge conditions for tool cleaning are set.	S10
BDX203	CTS pressurize check delay time	CTSM_UPCK_T	Common	Signal sets the value from user parameter (switch 1: installation) <CTS pressurize check delay time>.	S10
BDX204	Delay time for CTS pressure decrease check	CTSM_DWCK_T	Common	Signal sets the value from user parameter (switch 1: installation) <CTS depressurize check delay time>.	S10
BDX205	Blowoff time	CTSM_PRSREMV	Common	Signal sets the value from user parameter (switch 1: installation) <CTS pressure bleed time>.	S10
BDX206	Maximum automatic door operation time	DRMOVE_T	Common	Signal sets the value from the machine parameter (system 1: automatic door) <Operation time>.	
BDX207	Chip shower retention time	M400_OFFDL	Common	Signal sets the value from user parameter (switch 1: installation) <Chip shower retention time>.	S10
BDX256 BDX257 BDX258 BDX259 BDX260 BDX261 BDX262 BDX263	LB coordinate 1 Low-order: 0-15 bit LB coordinate 1 High-order: 16-31 bit LB coordinate 2 Low-order: 0-15 bit LB coordinate 2 High -order: 16-31 bit LB coordinate 3 Low-order: 0-15 bit LB coordinate 3 High-order: 16-31 bit LB coordinate 4 Low-order: 0-15 bit LB coordinate 4 High-order: 16-31 bit	LB_CDPOS1L LB_CDPOS1H LB_CDPOS2L LB_CDPOS2H LB_CDPOS3L LB_CDPOS3H LB_CDPOS4L LB_CDPOS4H	Common	Current coordinates in the loading system are set in the coordinate system that outputs to LB specified coordinate system 1 (LB_CORD1) and system 2 (LB_CORD2). When the output coordinate system is the “Tool 0 coordinate” or the “TCP coordinate in the selected tool coordinate”, then LB coordinate 1, LB coordinate 2, LB coordinate 3 and LB coordinate 4 are used for the X-axis, Y-axis, Z-axis and B-axis coordinates, respectively. When the output coordinate system is the “Joint coordinate”, then LB coordinate 1, LB coordinate 2, LB coordinate 3 and LB coordinate 4 are used for the J1-axis, J2-axis, J3-axis and TR-axis coordinates, respectively. The unit for the X-, Y-, Z- and TR-axes is 0.0001 mm when the LB coordinate 1 to 4 unit system (LB_CDPOS_UNIT) is set to <0: Metric>. The unit is 0.00001 inches when set to <1: Inch>. The unit for B-, J1-, J2- and J3-axes is 0.0001 degrees.	
BDX272 ~BDX303	LB internal general output data 0 to 31	LB_BDXGNL00 ~LB_BDXGNL31	Common	This data is a general output data from the loading system.	
BDX350	Total data frame loss 1	RIOERR	Common	Signal sets the total data frame loss 1.	
BDX351	Total data frame loss 2	RIOERR2	Common	Signal sets the total data frame loss 2.	
BDY354 to BDY355	Automatic door operation time Low-order High-order	 DR_TIMEL DR_TIMEH	Common	When using an automatic door specification, the operation time for the automatic door is set.	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BDX400	PLC output No.1	PLCOUTN1	Common	The PLC output No., which outputs between BDX408 and BDX423, is set. Refer to “4.5 PLC Data Output” for details about the PLC output No.	
BDX401	PLC output No.2	PLCOUTN2	Common	The PLC output No., which outputs between BDX424 and BDX439, is set. Refer to “4.5 PLC Data Output” for details about the PLC output No.	
BDX402	PLC output No.3	PLCOUTN3	Common	The PLC output No., which outputs between BDX440 and BDX455, is set. Refer to “4.5 PLC Data Output” for details about the PLC output No.	
BDX403 to BDX407	Reserved		Common	This signal is reserved. Therefore, do not use it.	
BDX408 to BDX423	PLC output 1-0 to PLC output 1-15	PLCOUT1_0 to PLCOUT1_F	Common	Data specified in BDX400 is set. The updating cycle is 10 ms. Refer to “4.5 PLC Data Output” for details about the data that was set.	
BDX424 to BDX439	PLC output 2-0 to PLC output 2-15	PLCOUT2_0 to PLCOUT2_F	Common	Data specified in BDX401 is set. The updating cycle is 10 ms. Refer to “4.5 PLC Data Output” for details about the data that was set.	
BDX440 to BDX455	PLC output 3-0 to PLC output 3-15	PLCOUT3_0 to PLCOUT3_F	Common	Data specified in BDX402 is set. The updating cycle is 10 ms. Refer to “4.5 PLC Data Output” for details about the data that was set.	
BDX456 to BDX535	Reserved	RESERVED	Common	This signal is reserved. Therefore, do not use it.	
BDX543	Reserved	RESERVED	Common	This signal is reserved. Therefore, do not use it.	

4.2.6 Internal Output Data (PLC→NC, BDY0 to BDY1023)

4.2.6.1 Overview

This signal is from the PLC→NC.
It indicates the internal status of the PLC using data.
The internal output data is comprised of 16 bits per 1 point, and it is written in 16-bit units.
When handling 32-bit data, 2 points are used. The data register number that is specified in a 32-bit command is the low-order 16-bit, and the data register + 1 that is specified is the high-order 16-bit.

OM name	BDY0 to BDXY1023 (decimal shown)
Total number of signals	1024
Type	Word (16-bit)
Output specification / Output range specification	Available / None
Double coil check	None
Record when power is OFF	Volatile
Index addressing	Available

4.2.6.2 Data List

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BDY000 to BDY007	Command data 1 P1-axis, low-order P1-axis, high-order P2-axis, low-order P2-axis, high-order P3-axis, low-order P3-axis, high-order P4-axis, low-order P4-axis, high-order	 ORD1P1L ORD1P1H ORD1P2L ORD1P2H ORD1P3L ORD1P3H ORD1P4L ORD1P4H	Common	The content changes depending on the command type. Refer to “Chapter 10 PLC-axis” for further details.	
BDY008 to BDY015	Command data 2 P1-axis, low-order P1-axis, high-order P2-axis, low-order P2-axis, high-order P3-axis, low-order P3-axis, high-order P4-axis, low-order P4-axis, high-order	 ORD2P1L ORD2P1H ORD2P2L ORD2P2H ORD2P3L ORD2P3H ORD2P4L ORD2P4H	Common	The content changes depending on the command type. Refer to “Chapter 10 PLC-axis” for further details.	
BDY016 to BDY023	Command data 3 P1-axis, low-order P1-axis, high-order P2-axis, low-order P2-axis, high-order P3-axis, low-order P3-axis, high-order P4-axis, low-order P4-axis, high-order	 ORD3P1L ORD3P1H ORD3P2L ORD3P2H ORD3P3L ORD3P3H ORD3P4L ORD3P4H	Common	The content changes depending on the command type. Refer to “Chapter 10 PLC-axis” for further details.	
BDY024 to BDY027	Command type P1-axis P2-axis P3-axis P4-axis	 ORDKNDP1 ORDKNDP2 ORDKNDP3 ORDKNDP4	Common	When the command START changes from OFF to ON, the setting that corresponds to the ON status is valid. Refer to “Chapter 10 PLC-axis” for further details.	
BDY028 to BDY031	Teaching pendant register P1-axis P2-axis P3-axis P4-axis	 TPREGP1 TPREGP2 TPREGP3 TPREGP4	Common	When the <Grab all> key and <← grab> key are pressed, the current position (machine coordinate) of the PLC-axis is imported to register D that is specified on the screen. (If the PLC-axis does not exist, it is not displayed.) The following can be specified for register D. Create and specify in a PLC program. P1-axis: BDY028, P2-axis: BDY029, P3-axis: BDY030 and P4-axis: BDY031	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BDY048 to BDY049	External additional axis 1 command angle Low-order High-order	 OTAX1ORDL OTAX1ORDH	Common	The travel amount on the **[OUTER ROTARY TABLE 1]** switch (+/-) can be set. (Travels in the same direction even if the switch is (+) or (-).) The travel amount used on the **[OUTER ROTARY TABLE 1]** switch (+/-) checks following the priority below and uses the corresponding value when zero does not apply. 1. BDY048 to 049 - External additional axis 1 command angle 2. BDY052 to 055 - External additional axis 1 command angle +/- 3. User parameter (switch 2) <Outer rotary table 1 indexing (+)> or <Outer rotary table 1 indexing (-)>	
BDY050 to BDY051	External additional axis 2 command angle Low-order High-order	 OTAX2ORDL OTAX2ORDH	Common	The travel amount on the **[OUTER ROTARY TABLE 2]** switch (+/-) can be set. (Travels in the same direction even if the switch is (+) or (-).) The travel amount used on the **[OUTER ROTARY TABLE 2]** switch (+/-) checks following the priority below and uses the corresponding value when zero does not apply. 1. BDY050 to 051 - External additional axis 2 command angle 2. BDY056 to 059 - External additional axis 2 command angle +/- 3. User parameter (switch 2) <Outer rotary table 2 indexing (+)> or <Outer rotary table 2 indexing (-)>	
BDY052 to BDY053	External additional axis 1 command angle e + Low-order High-order	 OTAX1RPL OTAX1RPH	Common	The travel amount on the **[OUTER ROTARY TABLE 1]** switch (+) can be set. The travel amount used on the **[OUTER ROTARY TABLE 1]** switch (+) checks following the priority below and uses the corresponding value when zero does not apply. 1. BDY048 to 049 - External additional axis 1 command angle 2. BDY052 to 055 - External additional axis 1 command angle + 3. User parameter (switch 2) <Outer rotary table 1 indexing (+)>	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BDY054 to BDY055	External additional axis 1 command angle - Low-order High-order	 OTAX1RML OTAX1RMH	Common	The travel amount on the **[OUTER ROTARY TABLE 1]** switch (-) can be set. The travel amount used on the **[OUTER ROTARY TABLE 1]** switch (-) checks following the priority below and uses the corresponding value when zero does not apply. 1. BDY048 to 049 - External additional axis 1 command angle 2. BDY054 to 055 - External additional axis 1 command angle + 3. User parameter (switch 2) <Outer rotary table 1 indexing (-)>	
BDY056 to BDY057	External additional axis 2 command angle + Low-order High-order	 OTAX2RPL OTAX2RPH	Common	The travel amount on the **[OUTER ROTARY TABLE 2]** switch (+) can be set. The travel amount used on the **[OUTER ROTARY TABLE 2]** switch (+) checks following the priority below and uses the corresponding value when zero does not apply. 1. BDY050 to 051 - External additional axis 2 command angle 2. BDY056 to 057 - External additional axis 2 command angle + 3. User parameter (switch 2) <Outer rotary table 2 indexing (+)>	
BDY058 to BDY059	External additional axis 2 command angle - Low-order High-order	 OTAX2RML OTAX2RMH	Common	The travel amount on the **[OUTER ROTARY TABLE 2]** switch (-) can be set. The travel amount used on the **[OUTER ROTARY TABLE 2]** switch (-) checks following the priority below and uses the corresponding value when zero does not apply. 1. BDY050 to 051 - External additional axis 2 command angle 2. BDY058 to 059 - External additional axis 2 command angle + 3. User parameter (switch 2) <Outer rotary table 2 indexing (-)>	
BDY064	External program selection No.	PLC_PRO	Common	Set the program number to start or for the external program read request. The setting range is 1 to 9999 and is the same for both the NC and conversation languages. If a value outside of the range is set, the alarm <<External program number error>> is triggered at the start or at the time of the program load request.	
BDY080	Tool cleaning status	TW_STATE	Common	The discharge status for tool cleaning is set.	S10
BDY096	Auto notification data type	SND_TYPE	Common	Set the data type for the auto notification when the communication parameter (auto notification function) <Method to specify sending data> is set to <1: Indirect>. 0: No data 1: PLC internal relay (M) 2: PLC data register (D) 16 bit 3: PLC data register (D) 32 bit	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BDY097	Leading address for auto notification data	SND_TOP	Common	Set the leading address for the data sent in the auto notification when the communication parameter (auto notification function) <Method to specify sending data> is set to <1: Indirect>. The settings that are available change depending on the value for SND_TYPE (BDY096). If SND_TYPE is 1 → 0 to 10239 If SND_TYPE is 2 → 0 to 8191 If SND_TYPE is 3 → even numbers from 0 to 8190	
BDY098	Auto notification data quantity	SND_LEN	Common	Set the data quantity sent in the auto notification when the communication parameter (auto notification function) <Method to specify sending data> is set to <1: Indirect>. The settings that are available change depending on the value for SND_TYPE (BDY096). If SND_TYPE is 1 → 0 to 600 If SND_TYPE is 2 → 1 to 100 If SND_TYPE is 3 → 1 to 50	
BDY099	Control program with auto notification	SND_PLC	Common	This signal sets the task to assign a control program that uses the auto notification function. Set before the SND_REQ is turned ON. 1: High speed task 1 2: Standard task 1 3: High speed task 2 4: Standard task 2 5: Standard task 3	
BDY112	Specified main input signal No.	SEL_IN_SIGNAL_NO	Common	Set the main input signal No. where you wish to output between BDX360 and BX36F. Refer to “6.2.1.3. Signal assignment” in the Installation Manual for details on the main input signal No.	
BDY116	PLC feedrate override	PLC_FEDOR	Common	Set the feedrate override value from the PLC. 0% to 200.00% (0~20000) Override operates at 100% when a value that is out of range is set. The signal can be used when the user parameter (switch 1: installation) <PLC feedrate override function> is set to <1: Enable>.	S1
BDY117	PLC spindle override	PLC_SPNOR	Common	Set the spindle override value from the PLC. 50.00% to 200.00% (5000~20000) Override operates at 100% when a value that is out of range is set. The signal can be used when the user parameter (switch 1: installation) <PLC spindle override function> is set to <1: Enable>.	S1

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BDY120	Specified tool No.	SEL_TLNO	Common	After setting the tool number or magazine number, the tool information is output to BD144 to BDX178. When the magazine number is set, the tool information for the corresponding magazine is output. 0: Mounted tool 1 to 99: Tool No. 1001 to 10xx: Magazine No.	
BDY121	Specified alarm type	SEL_ALMTP1	Common	After the alarm category number is set, BX370 turns ON when the corresponding alarm is triggered. Category number 0: OM 1: EX 2: EC 3: SV 4: NC 5: IO 6: SP 7: SM 8: SL 9: CM 10: PN 11: FC	
BDY122	Specified alarm number	SEL_ALMNO1	Common	After the alarm number is set, BX370 turns ON when the corresponding alarm is triggered.	
BDY128	Transient request data address	TRADRS	Common	Signal sets the initial address (D0 to D8191) for the storage location of the send data during transient communication. However, refer to the CC-Link standards specification for more details about the formatting when storing data.	
BDY129	Transient response data address	TRADRR	Common	Signal sets the initial address (D0 to D8191) for the storage location of the receive data during transient communication. However, refer to the CC-Link standards specification for more details about the formatting when storing data.	
BDY130	Transient request data length	TRLENS	Common	Signal sets the data length for the transient request.	
BDY131	Transient response data length	TRLENR	Common	Signal sets the data length for the transient response.	
BDY139	Workpiece coordinate system selection	WPSLCT	NC	Signal sets the coordinate system for writing the workpiece coordinate zero specified in BDY140 to 151. 6: External (NOTE) If another value (apart from the aforementioned values) is used, when BY1FF (writing request for workpiece coordinate zero) is turned ON, the alarm <<Workpiece coordinate zero setting error>> is triggered and the writing operation is not carried out.	

OM name	Name	Symbol	Conversation / NC	Signal description	Use
BDY140 to BDY151	Workpiece coordinate zero X-axis low-order: 0 to 15 bit X-axis high-order: 16 to 31 bit Y-axis low-order Y-axis high-order Z-axis low-order Z-axis high-order A-axis low-order A-axis high-order B-axis low-order B-axis high-order C-axis low-order C-axis high-order	 WPXL WPXH WPYL WPYH WPZL WPZH WPAL WPAH WPBL WPBH WPCL WPCH	NC	The coordinate set to this signal is written to the workpiece coordinate zero set in BDY139. The unit is based on the smallest unit.	
BDY272 ~BDY303	LB internal general input data 0 to 31	LB_BDYGNL00 ~ LB_BDYGNL31	Common	This data is general input data to the loading system.	
BDY400	Specified PLC output No.1	SEL_PLCOUTN1	Common	Set the PLC output No. to be output between BDX208 and BDX223. Refer to “4.5 PLC Data Output” for details about the PLC output No.	
BDY401	Specified PLC output No.2	SEL_PLCOUTN2	Common	Set the PLC output No. to be output between BDX224 and BDX239. Refer to “4.5 PLC Data Output” for details about the PLC output No.	
BDY402	Specified PLC output No.3	SEL_PLCOUTN3	Common	Set the PLC output No. to be output between BDX240 and BDX255. Refer to “4.5 PLC Data Output” for details about the PLC output No.	
BDY403 to BDY407	Reserved	RESERVED	Common	This signal is reserved. Therefore, do not use it.	
BDY543	Reserved	RESERVED	Common	This signal is reserved. Therefore, do not use it.	
BDY544 ~BDY607	Character string	STRCPY1 ~STRCPY64	Common	A maximum of 128 characters can be entered. (Refer to “3.5.1 Character code table (NC language mode)” “3.5.2 Character code table (conversation mode)” in the Operation Manual II for details about character codes.) The display languages support the following character codes. Japanese: Shift-JIS English, German, French, Portuguese, Italian and Spanish: ISO8859-1 Chinese: GB18030 Polish and Czech: ISO8859-2 Korean: EUC-KR	

4.2.7 Internal Relay (M0 to M10239)

This internal auxiliary relay can be used for general purposes.
When this relay is turned ON, signal output is available to the NC.
Even if the power is turned OFF, there are areas that can maintain data and areas that cannot maintain it.
It is necessary to specify the range used in each control program when using it for output (WRITE) to the internal relay. When using it as an input (READ), it is also possible to use in other control programs and as a means to transfer data between control programs.
In addition, the external panel is used for external input and output data.

OM name	M0 to M10239 (decimal shown)
Total number of signals	10240
Type	Bit
Output specification / Output range specification	Available / Present
Double coil check	Present
Record when power is OFF	Non-volatile area can be specified
Index addressing	Available

(NOTE) When using non-volatile areas (power failure hold areas), creating a recovery circuit is recommended while the BX10C (Power failure hold area change notice) and the BX10D (Power failure hold area check notice) signals are turned ON.

4

4.2.7.1 Reserved Internal Relay

Reserve M8192 to M9215 as the high speed task 2 range.
Reserve M9216 to M10239 as the standard task 3 range.
In addition, M8704 to M9215 and M9728 to M10239 are reserved. Therefore, do not use those ranges.

4.2.8 Data Register (D0 to D8191)

This data register can be used for general purposes.
Even if the power is turned OFF, there are areas that can maintain data and areas that cannot maintain it.
The data register is comprised of 16 bits per 1 point, and it is read and written in 16-bit units.
When handling 32-bit data, 2 points are used. The data register number that is specified in a 32-bit command is the low-order 16-bit, and the data register + 1 that is specified is the high-order 16-bit.
It is necessary to specify the range used in each control program when using it for output (WRITE) to the data register. When using it as an input (READ), it is also possible to use in other control programs and as a means to transfer data between control programs.
In addition, the external panel is used for external input and output data.

OM name	D0 to D8191 (decimal shown)
Total number of signals	8192
Type	Word
Output specification / Output range specification	Available / Present
Double coil check	None
Record when power is OFF	Non-volatile area can be specified
Index addressing	Available

(NOTE) When using non-volatile areas (power failure hold areas), creating a recovery circuit is recommended while the BX10C (Power failure hold area change notice) and the BX10D (Power failure hold area check notice) signals are turned ON.

4.2.8.1 Reserved Data Register

Reserve D7680 to D7935 as the high speed task 2 range.
Reserve D7936 to D8191 as the standard task 3 range.
D7808 to D7935 and D8064 to D8191 are reserved. Therefore, do not use them.

4.3 Local OM Description

The local OM is a memory area that can be used for each control program.

4.3.1 Local Relay (LM0 to LM4095)

The local relay is an auxiliary relay that can be used internally for each control program.

OM name	LM0 to M4095 (decimal shown)
Total number of signals	4096
Type	Bit
Output specification / Output range specification	Available / None
Double coil check	-
Record when power is OFF	Volatile
Index addressing	Available

4.3.1.1 Reserved Local Relay

LM2048 to LM4095 are reserved for both standard task 3 and high speed task 2. Therefore, do not use them.

4.3.2 Local Data Register (LD0 to LD8191)

The local data register is a memory that can store numerical value data (-32768 to 32767, or H0000 to HFFFF) for each control program. When handling 32-bit data, two registers are used. The data already stored in the program is maintained until other data is stored.

OM name	LD0 to LD8191 (decimal shown)
Total number of signals	8192
Type	Word
Output specification / Output range specification	Available / None
Double coil check	-
Record when power is OFF	Volatile
Index addressing	Available

4.3.2.1 Reserved Local Data Register

LD4096 to LD8191 are reserved for both standard task 3 and high speed task 2. Therefore, do not use them.

4.3.3 Local Timer (LT0 to LT511)

The local timer uses an addition formula and starts timing when the timer coil turns ON. When the current value = set value, the time runs out or expires, and the contact turns ON. When the local timer coil turns OFF, the current value becomes 0 and the contact turns OFF. The timing period for the local timer is fixed. The high speed control program is 1 ms, and a standard control program is 10 ms.

OM name		LT0 to LT511 (decimal shown)
Total number of signals		512
Type		Word
Record when power is OFF		Volatile
Index addressing		Available
Unit	High speed control program	1 ms
	Standard control program	10 ms

(NOTE) Make sure that the LT is set below 15000. If set to a value greater than that, the timer may overflow.

4.3.3.1 Reserved Local Timer

LT256 to LT511 are reserved for both standard task 3 and high speed task 2.Therefore, do not use them.

4.3.4 Local Counter (LC0 to LC511)

The local counter uses an addition formula, and when the count value = set value, the count is up (ends), and the contact turns ON.
The local counter is an OM that counts the startup times per the input conditions in the control program.When the OUT C command is executed, it performs processing for turning the count coil ON / OFF, for updating the current value and for turning the contact ON / OFF. The current value is updated at the rising edge of the OUT C command (OFF → ON). The current value is not updated when the OUT C command is OFF, when it changes from ON → ON, or when it changes from ON → OFF. The current value for the local counter is cleared even when the OUT C command is OFF. The RST command clears (resets) the current value for the local counter and turns the contact OFF. The counter value, at the time when the RST C command is executed, is cleared and the contact is turned OFF.

OM name	LC0 to LC511 (decimal shown)
Total number of signals	512
Type	Word
Record when power is OFF	Volatile
Index addressing	Available

4.3.4.1 Reserved Local Counter

LC256 to LC511 are reserved for both standard task 3 and high speed task 2.Therefore, do not use them.

4.3.5 Special Relay (SM0 to SM255)

This relay is specific to each control program and can be used as an internal auxiliary relay. Signals which indicate the sequence command status or results and which perform special operations are assigned.

OM name	SM0 to SM255 (decimal shown)
Total number of signals	256
Type	Bit
Index addressing	Available

4.3.5.1 List

OM No.	Function
SM0	Always ON
1	Scan 1 ON after RUN
2	Error flag (except operation errors and command execution errors)
3	Operation error flag
4	Command execution error flag
5	Compile error flag (High speed task 1)
6	Compile error flag (High speed task 2)
7	Carry flag
8	RUN/STOP status for RUN (ON: RUNNING, OFF: STOPPED)
9	Error reset flag
10	0.1 second clock (ON/OFF every 0.1 sec.)
11	0.2 second clock (ON/OFF every 0.2 sec.)
12	1 second clock (ON/OFF every 1 sec.)
13	Cycle specified clock (ON/OFF intervals set in SD13, ×10 msec.)
14	Assigned status of control program
15	(Not used)
16	Reserved (DFC1 being executed)
17	Reserved (DFC1 execution completed)
18	Reserved (DFC2 being executed)
19	Reserved (DFC2 execution completed)
20	Reserved (DFC3 being executed)
21	Reserved (DFC3 execution completed)
22	Reserved (DFC4 being executed)
23	Reserved (DFC4 execution completed)
;	;
;	;

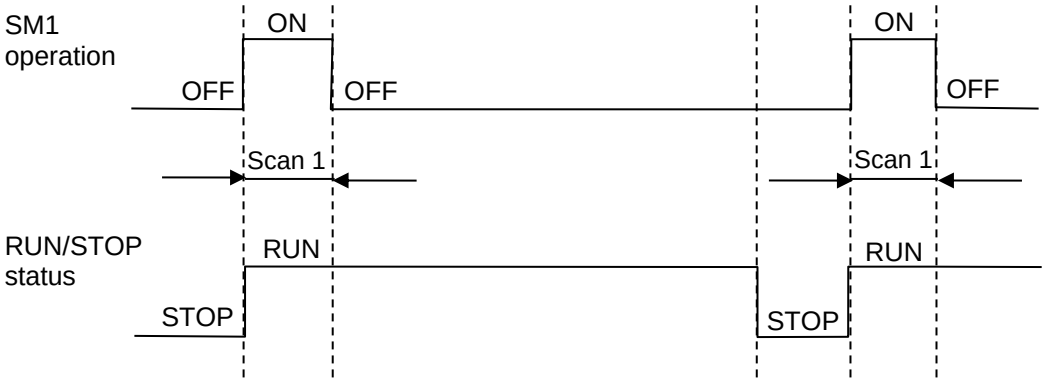
OM No.	Function
;	;
;	;
136	Reserved (DFC61 being executed)
137	Reserved (DFC61 execution completed)
138	Reserved (DFC62 being executed)
139	Reserved (DFC62 execution completed)
140	Reserved (DFC63 being executed)
141	Reserved (DFC63 execution completed)
142	Reserved (DFC64 being executed)
143	Reserved (DFC64 execution completed)
144	(Not used)
;	;
255	(Not used)

4.3.5.2 Special Relay Description

(NOTICE) Do not perform output and write operations with the special relay because there is a risk of malfunction.

1. Always ON (SM0)
   This relay is normally ON from the beginning of the scan for the control program operation. The ON status is always maintained even when the control program is in a STOP or DEBUG-RUN state.

2. Scan 1 ON after RUN (SM1)
   This relay turns ON only for scan 1 when the control program changes from a STOP state to a RUN state. It turns OFF thereafter from scan 2. The operation state is shown below in the timing chart. When using SM1 as “Scan 1 OFF after RUN”, SM1 is used as a contact *b* (normally closed) in the control program.



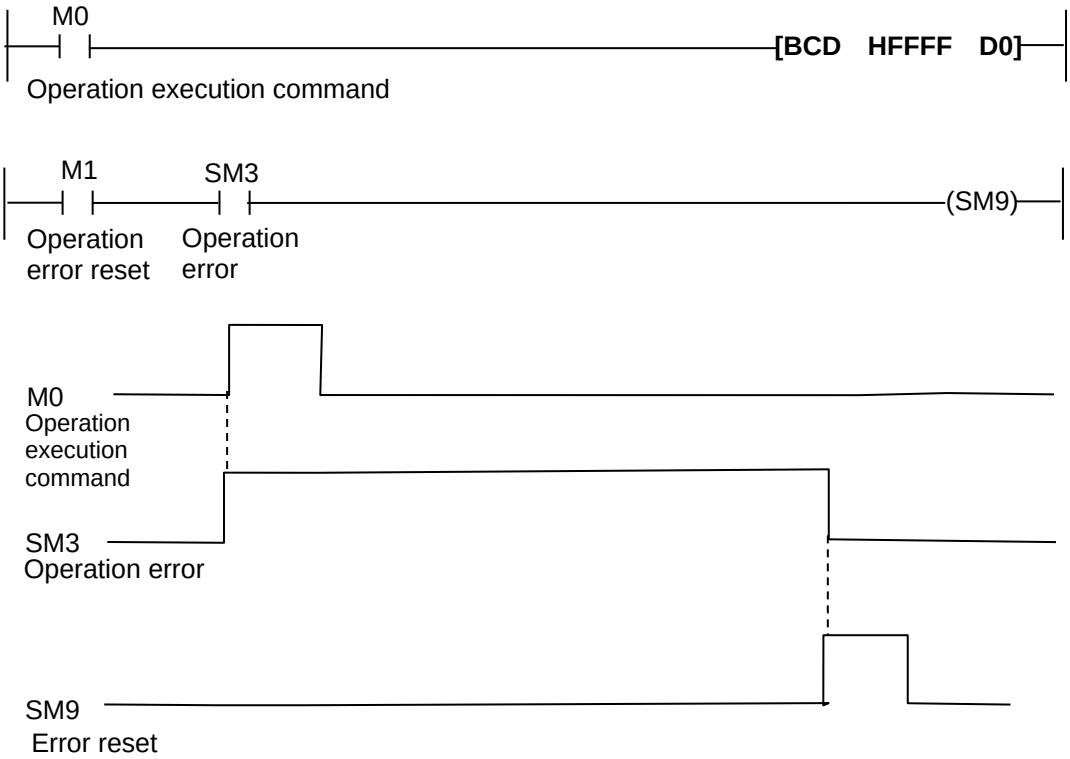
3. Error flag: Excluding operation errors and command execution errors (SM2)
   SM2 turns ON when an error occurs, excluding operation errors and command execution errors, during control program execution. SM2 turns ON the following errors:

Error name	Description	Control program operation
<<PLC watch dog timeout>>	The watch dog timeout is triggered.	Depends on the parameter setting
<<Abnormal memory>>	Failed to allocate necessary memory.	Stop
<<Function execution error>>	Failed to execute system function.	Stop

   SM2 (error flag) turns ON when the control program command is executed.

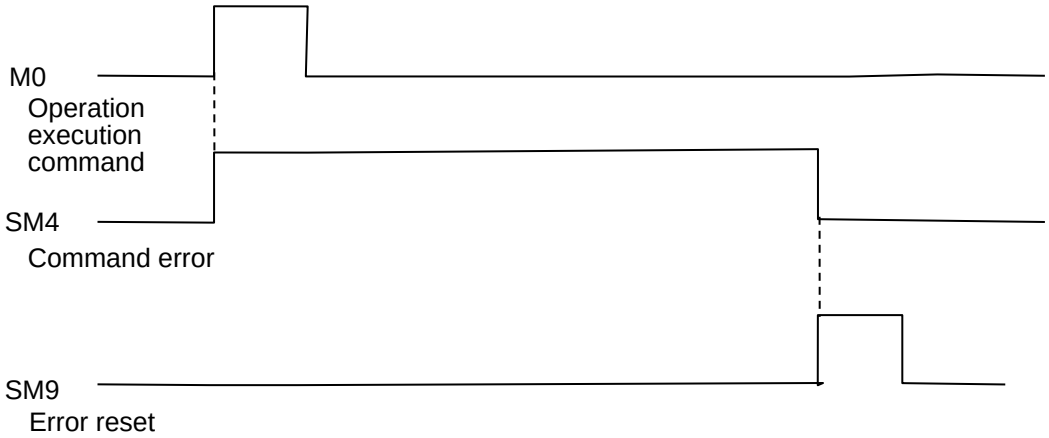
4. Operation error flag (SM3)
   SM3 turns ON when an operation error occurs during control program execution. SM3 does not turn OFF even if the operation error execution status turns OFF. Accordingly, the user needs to turn ON the error reset flag (SM9) in the program in order to reset the error status.
   SM3 turns ON the following errors:

Error name	Description	Control program operation
<<BCD or BIN conversion error>>	A conversion error occurred during the BCD/BIN conversion.	Depends on the parameter setting
<<Divided by zero>>	Attempted to execute a division operation with the numerical value 0.	Depends on the parameter setting



SM3 (operation error flag) turns ON when the control program command is executed.

5. Command execution error flag (SM4)
   SM4 turns ON when a command execution error occurs during control program execution. SM4 does not turn OFF even if the operation error execution status turns OFF. Accordingly, the user needs to turn ON the error reset flag (SM9) in the program in order to reset the error status. Refer to “3. Command execution error code (SD5) and command execution error step (SD6)” in “4.3.6.2 Special register description” for further details on errors turned ON by SM4.



SM4 (command execution error flag) turns ON when the control program command is executed.

6. Compile error flag (High speed task 1/2) (SM5/6)
   The flag turns ON when an error occurs during compiling in the control program that is assigned to high speed task 1 and 2. If a compile error occurs, the control program changes into the same status as when the program does not write because it is not saved in an executable area. The compile error flag either turns OFF when the error reset flag (SM9) turns ON, or when the control program successfully completes the compile operation.
   This flag is normally OFF. Refer to “4. Compile error code (High speed 1/2) (SD7) and compile error step (High speed 1/2) (SD8)” in “4.3.6.2 Special register description” for further details on errors turned ON by SM5/6.

7. Carry flag (SM7)
   - Right rotation command
   - Left rotation command
   - Right shift command
   - Left shift command

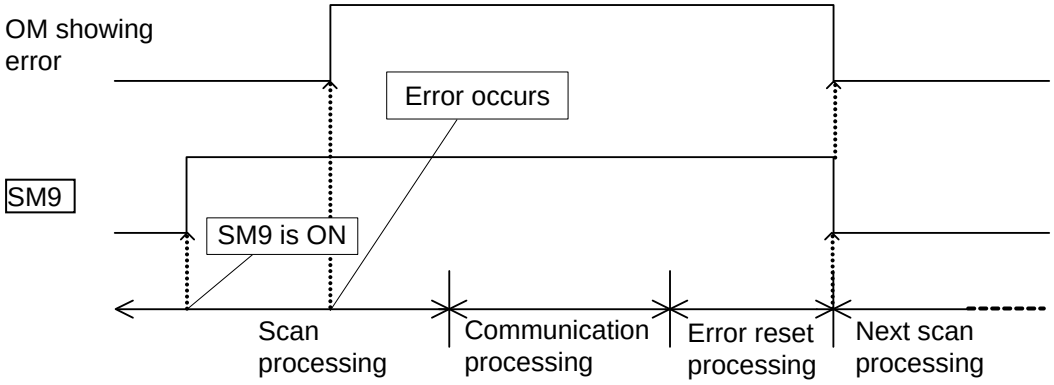
   Carry flag turns ON when the control program command is executed.

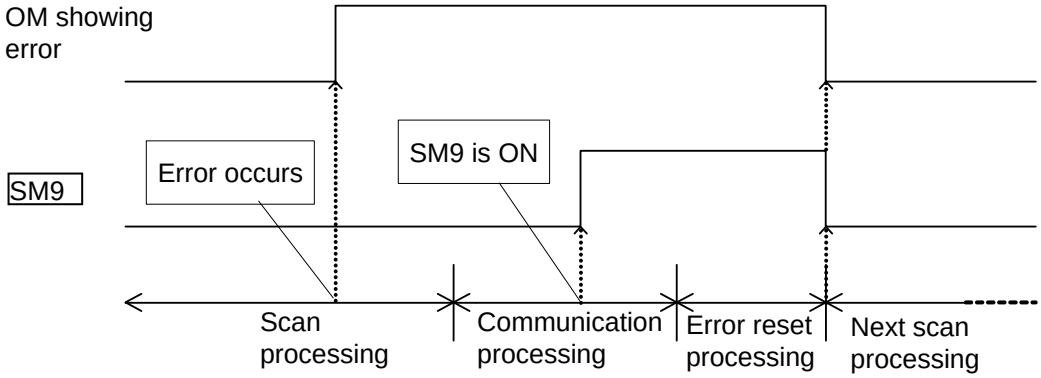
8. RUN/STOP status (SM8)
   SM8 only turns ON when the control program is in RUN state. It turns OFF in all other situations.

9. Error reset flag (SM9)
   If SM9 turns ON when one of the following is ON: SM2 (error flag), SM3 (operation error flag) and SM4 (command execution error flag), then the flag for the error that occurs turns OFF, and the corresponding SD value becomes 0. In addition, SM9 itself turns OFF at the same time.
   The relationship between the error flag and the error code, or the error triggered step is shown below.

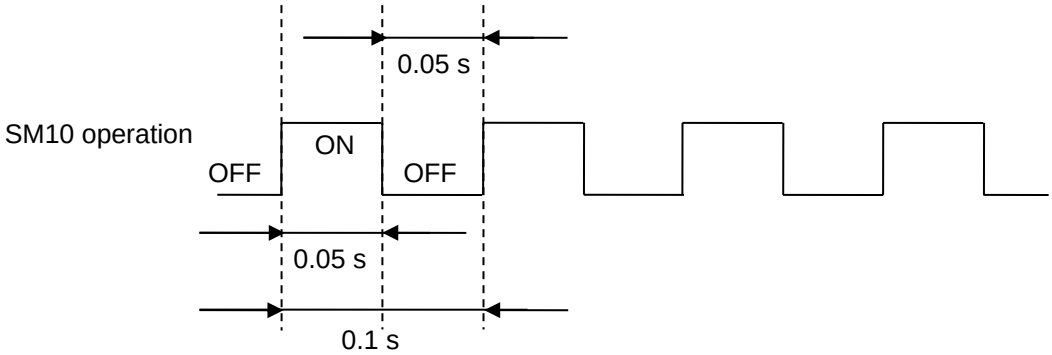
Error type	Corresponding SM and SD	When error occurs	When error reset flag executes
Error flag	SM2 (error flag)	ON	OFF
	SD1 (error code)	Error code	0
	SD2 (error step)	Error triggered step	0
Operation error	SM3 (operation error flag)	ON	OFF
	SD3 (operation error code)	Error code	0
	SD4 (operation error step)	Error triggered step	0
Command error	SM4 (command execution error flag)	ON	OFF
	SD5 (command execution error code)	Error code	0
	SD6 (command execution error step)	Error triggered step	0
Compile error	SM5 (compile error flag (High speed task 1))	ON	OFF
	SM6 (compile error flag (High speed task 2))	ON	OFF
	SD7 (error code (high-speed 1/2) during a compile error)	Error code	0
	SD8 (error step (high-speed 1/2) during a compile error)	Error triggered step	0

The error reset processing is performed after the scan and communication processing. As a result, it does not reset immediately even if the SM9 turns ON.The following charts illustrate when SM9 turns ON inside the ladder program, and when SM9 turns ON with the SET/RESET operation from the editor.

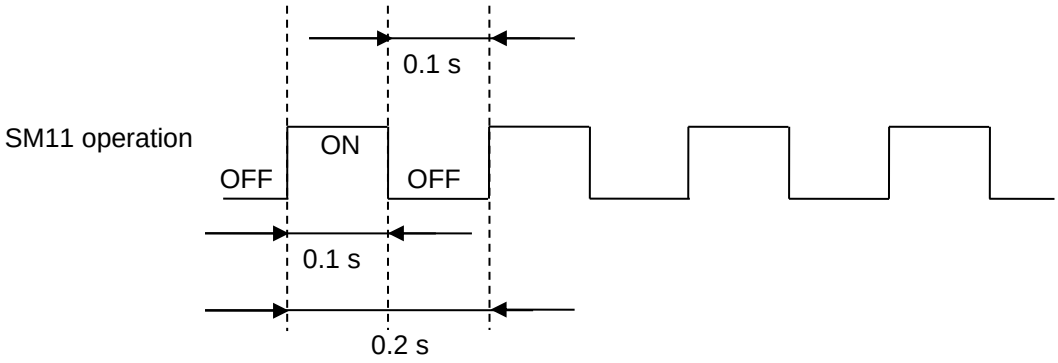




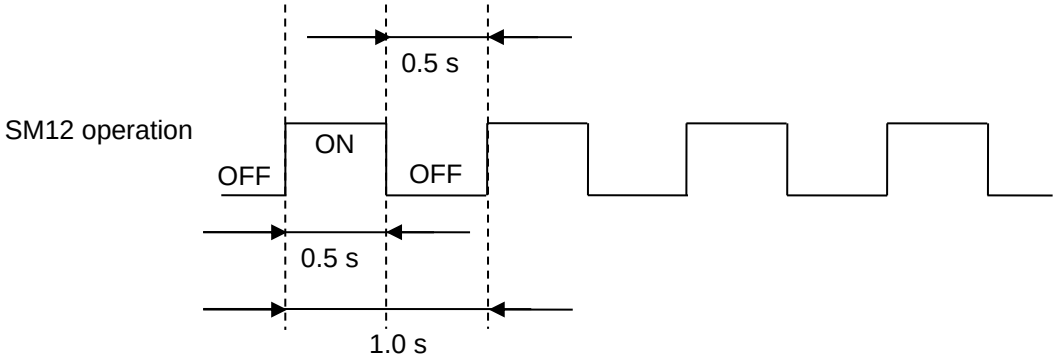
10. 0.1 second clock (SM10)
    This contact repeatedly turns ON/OFF in 0.1 second cycles regardless of the control program execution.
    The ON/OFF cycle is updated when the end processing is executed for a control program.



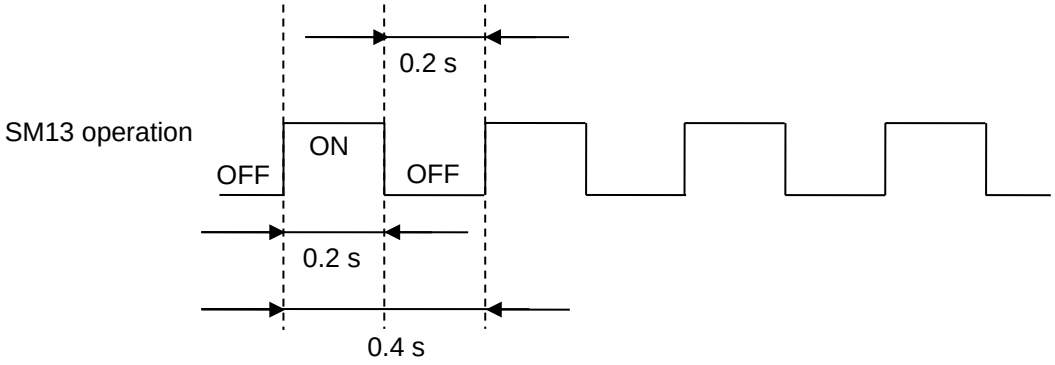
11. 0.2 second clock (SM11)
    This contact repeatedly turns ON/OFF in 0.2 second cycles regardless of the control program execution.
    The ON/OFF cycle is updated when the end processing is executed for a control program.



12. 1.0 second clock (SM12)
    This contact repeatedly turns ON/OFF in 1 second cycles regardless of the control program execution.
    The ON/OFF cycle is updated when the end processing is executed for a control program.



13. Cycle specified clock (SM13)
    This contact repeatedly turns ON/OFF in cycles set by SD13 regardless of the control program execution.
    The SM13 operation is shown below when 40 is set for SD13 (changes to 400 ms cycles).
    SM13 stays OFF when SD13 is less than 0.
    When the SD13 value is changed, the change is reflected in the end processing of each control program.
    The ON/OFF cycle is updated when the end processing is executed for a control program.



14. Assigned status of control program (SM14)
    This contact shows whether or not the control program has been assigned to a task.
    This contact turns ON when the control program has been assigned to a task. It turns OFF in all other situations.

4.3.6 Special Register (SD0 to SD255)

(NOTICE) Do not perform output and write operations to the special register because there is a risk of malfunction.

This data register can be used specifically for a control program.
The sequence command operation status and results are assigned.

OM name	SD0 to SD255 (decimal shown)
Total number of signals	256
Type	Word
Index addressing	Available

4.3.6.1 List

OM No.	Function
SD0	(Not used)
1	Error code (except operation errors and command execution errors)
2	Error step (except operation errors and command execution errors)
3	Operation error code
4	Operation error step
5	Command execution error code
6	Command execution error step
7	Error code (high-speed 1/2) during a compile error
8	Error step (high-speed 1/2) during a compile error
9	(Not used)
10	Minimum scan time
11	Current scan time
12	Maximum scan time
13	Interval (×10 msec) that turns cycle specified clock (SM13) ON/OFF
14	(Not used)
15	(Not used)
16	Reserved (FC1 execution result)
17	Reserved (DFC1 error code)
18	Reserved (DFC2 execution result)
19	Reserved (DFC2 error code)
20	Reserved (DFC3 execution result)
21	Reserved (DFC3 error code)
22	Reserved (DFC3 execution result)
23	Reserved (DFC3 error code)
;	;
;	;
;	;
;	;
56	Reserved (DFC13 execution result)
57	Reserved (DFC13 error code)
58	Reserved (DFC14 execution result)
59	Reserved (DFC14 error code)
60	Reserved (DFC15 execution result)
61	Reserved (DFC15 error code)
62	Reserved (DFC16 execution result)
63	Reserved (DFC16 error code)
64	(Not used)
;	;
254	(Not used)
255	RUN counter

4.3.6.2 Special Register Description

1. Error code (SD1) and error step (SD2)
   The error code when SM2 (error flag) turns ON is stored in SD1, and the step position for the control program where the error occurred is stored in SD2.
   When the error is cancelled and SM2 turns OFF, 0 is stored in SD1 and SD2.
   The following shows the stored error code and error description.

Error name	Description	Cause	Ladder operation	Error code
<<PLC watch dog timeout>>	Scan 1 in the sequence program did not end within the time set in the watch dog timeout setting.	• Error in creating control program • Setting error for parameter (setting time is too short)	Parameter	1
<<Abnormal memory>>	Failed to allocate necessary memory.	-	Stop	2
<<Function execution error>>	Failed to execute system function.	-	Stop	3

   - SD1 (error code) and SD2 (error step) are stored if an error occurs at the time of command execution. (not end refresh)
   - SD1 and SD2 do not change to 0 even if the operation error execution status turns OFF. Accordingly, the user needs to turn ON the error reset flag (SM9) in the program in order to reset the error status and change the value to 0.

2. Operation error code (SD3) and operation error step (SD4)
   The error code when SM3 (operation error flag) turns ON is stored in SD3, and the step position for the control program where the error occurred is stored in SD4. However, the high speed control programs 1 and 2 are fixed at 0 for SD4.
   When the error is cancelled and SM3 turns OFF, 0 is stored in SD3 and SD4.
   The following shows the stored operation error code and operation error description.

Error name	Description	Cause	Description	Error code
<<BCD/BIN conversion error>>	A conversion error occurred during the BCD/BIN conversion.	Detected during sequence program execution	Parameter	10
<<Divided by zero>>	Attempted to execute a division operation with a denominator of 0.	Detected during sequence program execution	Parameter	11

   - SD3 (operation error code) and SD4 (operation error step) are stored if an error occurs at the time of command execution. (not end refresh)
   - SD3 and SD4 do not change to 0 even if the operation error execution status turns OFF. Accordingly, the user needs to turn ON the error reset flag (SM9) in the program in order to reset the error status and change the value to 0.

3. Command execution error code (SD5) and command execution error step (SD6)
The error code when SM4 (command execution error flag) turns ON is stored in SD5, and the step position for the control program where the error occurred is stored in SD6.
When the error is cancelled and SM4 turns OFF, 0 is stored in SD5 and SD6.
The following shows the stored command execution error code and command execution error description.

Error name	Description	Cause	Operation	Error code
<<Command code is abnormal.>> (NOTE)	Control program is corrupt	• Control program data is corrupt • Memory error	Stop	20
<<No specif. label f. JMP, JE or CALL.>> (NOTE)	The corresponding labels for the JMP command, the JE command and the CALL command are not set.	• Creation error in sequence program • Detected during sequence program execution	Depends on the parameter setting.	21
<<CALL-RET, etc. not in a set>> (NOTE)	The CALL-RET, the MC-MCR and the FOR-NEXT commands are not in a set. Or, it was omitted in the JMP command and the JE command, after the CALL command, the MC command and the FOR command are executed but before the RET command, the MCR command and the NEXT command are executed. Apart from the FOR-NEXT command, the BREAK command is executed.	• Creation error in sequence program • Detected during sequence program execution	Depends on the parameter setting.	22
<<Access to OM outside the area>>	Attempted to access but exceeded the maximum area of the OM.	• Creation error in sequence program • Detected during sequence program execution	Depends on the parameter setting.	23
<<MC, CALL, FOR nesting too deep>> (NOTE)	Nesting for the MC command, the CALL command and the FOR command exceeds the maximum value.	• Creation error in sequence program • Detected during sequence program execution	Depends on the parameter setting.	24
Reserved <<DFC command execution failed>>	Failed to execute the DFC command.	This error does not occur on the current version because it is a future function.	Depends on the parameter setting.	25

(NOTE) The error occurs when a standard task command is executed. The error occurs during compiling in a high speed task.

- SD5 (command execution error code) and SD6 (command execution error step) are stored if an error occurs at the time of command execution. (not end refresh)
- SD5 and SD6 do not change to 0 even if the command error execution status turns OFF. Accordingly, the user needs to turn ON the error reset flag (SM9) in the program in order to reset the error status and change the value to 0.

4. Error code (high-speed 1/2) during compile error (SD7) and error step (high-speed 1/2) during compile error (SD8)
   The error code for the compile error, triggered in the control program assigned to high speed tasks 1 and 2, is stored in SD7, and the step position for the control program where the compile error occurred is stored in SD8.
   When the error is cancelled and SM5 or SM6 turns OFF, 0 is stored in SD7 and SD8.
   The following shows the stored compile error code and the error description during the compile error.

Error name	Description	Error code
<<Command code is abnormal.>>	Control program is corrupt	20
<<No specif. label f. JMP, JE or CALL.>>	The corresponding labels for the JMP command, the JE command and the CALL command are not set.	21
<<CALL-RET, etc. not in a set>>	The CALL-RET command, the MC-MCR command and the FOR-NEXT command are not in a set.	22
<<MC, CALL, FOR nesting too deep>>	Nesting for the MC command and the FOR command exceeds the maximum value.	24
<<The size is exceeded.>>	Code after the compile result exceeds the specified size (high speed 1/2 compile code storage area)	26

   The compile error stops the compile operation when the error is first triggered, and the error code is stored at that time.
   SD7 and 8 turn ON the error reset flag (SM9), or they clear it to 0 when the ladder program finishes compiling normally.

5. Minimum scan time (SD10), current scan time (SD11) and maximum scan time (SD12)
   The scan times for each ladder program are stored.

SD	Description	Unit for value stored	
		High speed	Standard
SD10 (minimum scan time)	Minimum scan time from when the control program is RUN until the current time.	×1 ms	×1 ms
SD11 (current scan time)	Most recent scan time for the control program	×1 ms	×1 ms
SD12 (maximum scan time)	Maximum scan time from when the control program is RUN until the current time.	×1 ms	×1 ms

6. Cycle set value (SD13) turns cycle specified clock (SM13) ON/OFF
   The set values are stored for the cycle specified clock that turns SM13 ON/OFF.
   The stored value is ×10 ms for both the high speed and standard control programs.

7. RUN counter (SD255)
   When the control program is in a RUN state, SD255 increases the count by increments of 1 for each scan.
   The value is maintained and does not increase while in STOP status. When the SD255 value is 32767, the next value after the increase is -32768.

4.3.7 Index Register (IX0 to IX7)

The index is an OM that is used to indirectly set (index addressing) the OM used in a control program.
There eight indexes from IX0 to IX7.
The index is comprised of 16 bits per 1 point, and read and write operations can be performed in 16-bit units.

OM name	IX0 to IX7 (decimal shown)
Total number of signals	8
Type	Word
Index addressing	Not available

Usage example) M1IX1 refers to M11 when IX1 = 10.
After M1IX1 is input while editing, M1I1 is displayed on the screen. In addition, M1I1 can also be input.

When in monitor mode, the display for the index modified OM shows the OM status/value that ignores the index.

Ex: The example assumes the following conditions: M1IX1(= M11) in IX1 = 10 is input for contact *a*, and M1 is OFF.
When M11 changes from OFF to ON, the M1IX1 display in the ladder program ignores the index. Therefore, M1IX1 for contact *a* does not change to ON.

(NOTE) The reference address range is not checked for the OM where the index addressing is being performed while the control program is executed. When the addressing exceeds the range for each OM due to the index register values, do not perform the check because it references and updates the OMs, etc., for types that have a different purpose.

4.3.8 Label (L0 to L1023)

The label is an OM that is used in structured commands. 8 points can be used in program 1 that is being executed. When the **[L]** key, a **[NUMBER]** and the **[ENT]** key are input at a line where the step number is displayed (a line where a number is entered on the leftmost position of the LD program), the user can set a label for that line. (On a line where the step number is not displayed, the message <Invalid execution point.> is displayed on the bottom-left corner of the screen. The **[ENT]** key is disabled and setting is not possible.)

OM name	L0 to L1023 (decimal shown)
Total number of signals	1024
Index addressing	Not available

Specification of the call destination (subroutine program header) for the label application subroutine call command (CALL and CALLP)

(NOTE 1) The subroutine assigned to L0 is executed when the built-in PLC standard control program execution changes from a STOP state to a RUN state.
(NOTE 2) L1023 indicates the program END.

4.4 PLC Data Input

4.4.1 Workpiece Coordinate Zero Writing

Execution procedure

1. Set the coordinate system for writing to the signal BDY139 (workpiece coordinate system selection).
2. Set the coordinate for each axis to the signals BDY140 to 151 (workpiece coordinate zero).
3. Make sure that the signal BX1B1 (writing is restricted for workpiece coordinate zero) is not ON.
4. Turn the signal BY1FF (writing request for workpiece coordinate zero) from OFF to ON to perform the writing operation.

(NOTE 1) The writing counter for workpiece coordinate zero (BDX182) counts up regardless if the writing operation was successful or not. If the writing operation is not carried out, the signal BX1B0 (workpiece coordinate zero writing error) turns ON.

(NOTE 2) In the following situations, a block stop is carried out without performing the writing operation.

- When an out of range value is set to a signal between BDY139 and BDY151
  Current alarm: <<Workpiece coordinate zero setting error>>
- When the selected workpiece coordinate zero is being used and one of the following is being executed
  - Scaling
  - Mirror image mode
  - Feature coordinate manufacturing mode engaged

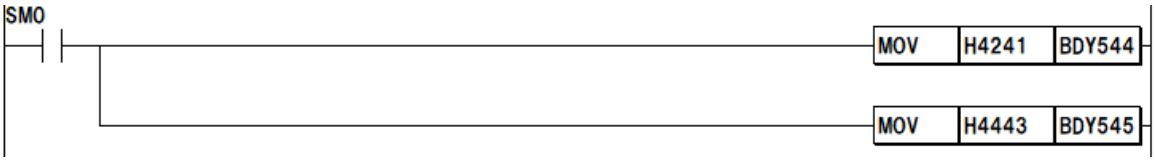
  Current alarm: <<Writing not possible for workpiece coordinate zero>>

4.4.2 Character String

Execution procedure

1. Character string is input into BDY544 to BDY607.
   Since BDY is 16-bit data, a maximum of 2 characters can be input.

   Ex: Input character string “ABCD”

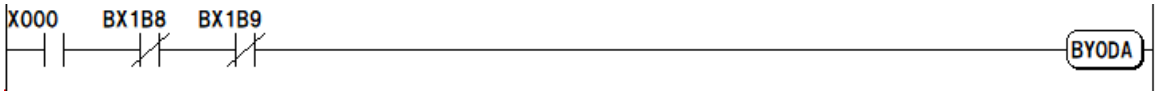


The letter is converted to a hexadecimal following the ASCII code and then input.
“A” 0x41 “B” 0x42 “C” 0x43 “D” 0x44
For multibytes, the corresponding character codes vary depending on the display language.

The data is input directly to BDY in the PLC program, or the data is input to the LD/D register and then forwarded to BDY via communication.

D0 to D63 is forwarded to BDY544 to BDY607.

2. The character string copy command is executed.
   When BY0DA is turned ON, the character string copy command is executed.
   However, the character string copy command is not executed when one of the following signals is ON.
   - BX1B8: Character string writing is restricted
   - BX1B9: Character string copied



   When the character string is copied (copy completed), BX1B9 turns ON, therefore set BY0DA to turn OFF.

3. Paste
   The character string is saved in the copy function on the MDI operation/program edit screen.
   Therefore, use and press <Paste> on the NC screen.

4.5 PLC Data Output

4.5.1 Data list

Particular data can be output by entering the PLC output No. below for BDY400/401/402.

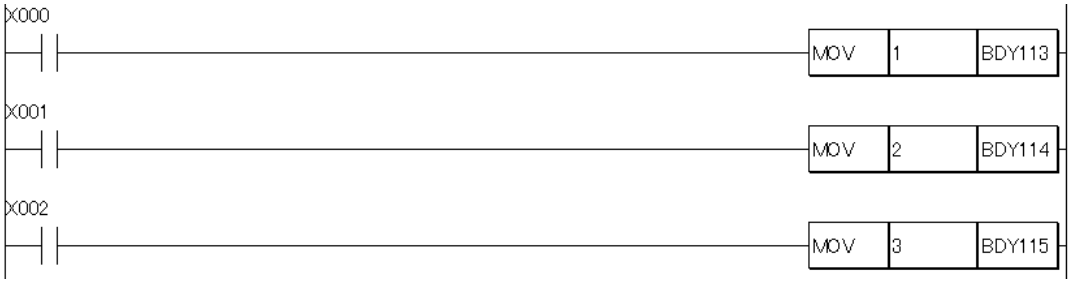
PLC output No.	Output data name
0	None
1	Current alarm/operator message screen 1
2	Current alarm/operator message screen 2
3	Current alarm/operator message screen 3
2001	Production monitor – Time display screen 1
201x (NOTE)	Production monitor - Workpiece counter screen
202x (NOTE)	Production monitor - Automatic thermal distortion compensation screen
4xx1 (NOTE)	Input/Output servo (*-axis) screen 1
4xx2 (NOTE)	Input/Output servo (*-axis) screen 2
4xx3 (NOTE)	Input/Output servo (*-axis) screen 3
500x (NOTE)	Workpiece coordinate zero screen
51xx (NOTE)	Workpiece coordinate zero (extended) screen

(NOTE) "xx" varies depending on the destination being specified. Refer to the details of each output data.

Data is output to the BDX below depending on the aforementioned PLC output No.

	PLC output 1	PLC output 2	PLC output 3
0	BDX408	BDX424	BDX440
1	BDX4BDX409	BDX425	BDX441
2	BDX410	BDX426	BDX442
3	BDX411	BDX427	BDX443
4	BDX412	BDX428	BDX444
5	BDX413	BDX429	BDX445
6	BDX414	BDX430	BDX446
7	BDX415	BDX431	BDX447
8	BDX416	BDX432	BDX448
9	BDX417	BDX433	BDX449
10	BDX418	BDX434	BDX450
11	BDX419	BDX435	BDX451
12	BDX420	BDX436	BDX452
13	BDX421	BDX437	BDX453
14	BDX422	BDX438	BDX454
15	BDX423	BDX439	BDX455

Ex: When BDY113=1, BDY114=2 and BDY115=3



When X000 signal is ON, the data on current alarm/operator message screen 1 is output between BDX208 and BDX223.
When X001 signal is ON, the data on current alarm/operator message screen 2 is output between BDX224 and BDX239.
When X002 signal is ON, the data on current alarm/operator message screen 3 is output between BDX240 and BDX255.

4.5.2 PLC output No.0 (None)

Output data	Name	Description
0	-	Outputs 0.
1	-	Outputs 0.
2	-	Outputs 0.
3	-	Outputs 0.
4	-	Outputs 0.
5	-	Outputs 0.
6	-	Outputs 0.
7	-	Outputs 0.
8	-	Outputs 0.
9	-	Outputs 0.
10	-	Outputs 0.
11	-	Outputs 0.
12	-	Outputs 0.
13	-	Outputs 0.
14	-	Outputs 0.
15	-	Outputs 0.

4.5.3 PLC output No.1 (Current alarm/operator message screen 1)

Output data	Name	Description
0	Category No.1	Outputs the first category number on the current alarm/operator message screen.
1	Alarm No.1	Outputs the first alarm number on the current alarm/operator message screen.
2	Category No.2	Outputs the second category number on the current alarm/operator message screen.
3	Alarm No.2	Outputs the second alarm number on the current alarm/operator message screen.
4	Category No.3	Outputs the third category number on the current alarm/operator message screen.
5	Alarm No.3	Outputs the third alarm number on the current alarm/operator message screen.
6	Category No.4	Outputs the fourth category number on the current alarm/operator message screen.
7	Alarm No.4	Outputs the fourth alarm number on the current alarm/operator message screen.
8	Category No.5	Outputs the fifth category number on the current alarm/operator message screen.
9	Alarm No.5	Outputs the fifth alarm number on the current alarm/operator message screen.
10	Category No.6	Outputs the sixth category number on the current alarm/operator message screen.
11	Alarm No.6	Outputs the sixth alarm number on the current alarm/operator message screen.
12	Category No.7	Outputs the seventh category number on the current alarm/operator message screen.
13	Alarm No.7	Outputs the seventh alarm number on the current alarm/operator message screen.
14	Category No.8	Outputs the eighth category number on the current alarm/operator message screen.
15	Alarm No.8	Outputs the eighth alarm number on the current alarm/operator message screen.

4.5.4 PLC output No.2 (Current alarm/operator message screen 2)

Output data	Name	Description
0	Category No.9	Outputs the ninth category number on the current alarm/operator message screen.
1	Alarm No.9	Outputs the ninth alarm number on the current alarm/operator message screen.
2	Category No.10	Outputs the tenth category number on the current alarm/operator message screen.
3	Alarm No.10	Outputs the tenth alarm number on the current alarm/operator message screen.
4	Category No.11	Outputs the eleventh category number on the current alarm/operator message screen.
5	Alarm No.11	Outputs the eleventh alarm number on the current alarm/operator message screen.
6	Category No.12	Outputs the twelfth category number on the current alarm/operator message screen.
7	Alarm No.12	Outputs the twelfth alarm number on the current alarm/operator message screen.
8	Category No.13	Outputs the thirteenth category number on the current alarm/operator message screen.
9	Alarm No.13	Outputs the thirteenth alarm number on the current alarm/operator message screen.
10	Category No.14	Outputs the fourteenth category number on the current alarm/operator message screen.
11	Alarm No.14	Outputs the fourteenth alarm number on the current alarm/operator message screen.
12	Category No.15	Outputs the fifteenth category number on the current alarm/operator message screen.
13	Alarm No.15	Outputs the fifteenth alarm number on the current alarm/operator message screen.
14	Category No.16	Outputs the sixteenth category number on the current alarm/operator message screen.
15	Alarm No.16	Outputs the sixteenth alarm number on the current alarm/operator message screen.

4.5.5 PLC output No.3 (Current alarm/operator message screen 3)

Output data	Name	Description
0	Category No.17	Outputs the seventeenth category number on the current alarm/operator message screen.
1	Alarm No.17	Outputs the seventeenth alarm number on the current alarm/operator message screen.
2	Category No.18	Outputs the eighteenth category number on the current alarm/operator message screen.
3	Alarm No.18	Outputs the eighteenth alarm number on the current alarm/operator message screen.
4	-	Outputs 0.
5	-	Outputs 0.
6	-	Outputs 0.
7	-	Outputs 0.
8	-	Outputs 0.
9	-	Outputs 0.
10	-	Outputs 0.
11	-	Outputs 0.
12	-	Outputs 0.
13	-	Outputs 0.
14	-	Outputs 0.
15	-	Outputs 0.

4.5.6 PLC Output No.2001 (Production monitor – Time display screen 1)

Output data	Name	Description
0	Cycle time P1 (low-order)	When the machine model is equipped with a QT (quick table), the pallet 1 cycle time is output on the time display for the production monitor. When the machine model is not equipped with a QT (quick table), the cycle time is output on the time display for the production monitor.
1	Cycle time P1 (high-order)	
2	Cycle time P2 (low-order)	When the machine model is equipped with a QT (quick table), the pallet 2 cycle time is output on the time display for the production monitor. When the machine model is not equipped with a QT (quick table), 0 is output.
3	Cycle time P2 (high-order)	
4	-	Outputs 0.
5	-	Outputs 0.
6	-	Outputs 0.
7	-	Outputs 0.
8	-	Outputs 0.
9	-	Outputs 0.
10	-	Outputs 0.
11	-	Outputs 0.
12	-	Outputs 0.
13	-	Outputs 0.
14	-	Outputs 0.
15	-	Outputs 0.

4.5.7 Output Data No.201x (Production Monitor - Workpiece Counter Screen)

x = 1: Counter 1 2: Counter 2 3: Counter 3 4: Counter 4

Output data	Name	Description
0	Count (low-order)	Outputs the count for counter x on the workpiece counter screen for the production monitor.
1	Count (high-order)	
2	Current value (low-order)	Outputs the current value for counter x on the workpiece counter screen for the production monitor.
3	Current value (high-order)	
4	Target value (low-order)	Outputs the target value for counter x on the workpiece counter screen for the production monitor.
5	Target value (high-order)	
6	End signal value (low-order)	Outputs the end signal value for counter x on the workpiece counter screen for the production monitor.
7	End signal value (high-order)	
8	-	Outputs 0.
9	-	Outputs 0.
10	-	Outputs 0.
11	-	Outputs 0.
12	-	Outputs 0.
13	-	Outputs 0.
14	-	Outputs 0.
15	-	Outputs 0.

4.5.8 Output Data No.202x (Production Monitor - Automatic Thermal Distortion Compensation Screen)

x = 1: X-axis compensation 2: Y-axis compensation 3: Z-axis compensation

Output data	Name	Description
0	Position 1 (low-order)	Outputs the position setting for <Position 1> on the automatic thermal distortion compensation screen for the production monitor.
1	Position 1 (high-order)	
2	Compensation 1 (low-order)	Outputs the current time compensation for <Compensation 1> on the automatic thermal distortion compensation screen for the production monitor.
3	Compensation 1 (high-order)	
4	Position 2 (low-order)	Outputs the position setting for <Position 2> on the automatic thermal distortion compensation screen for the production monitor.
5	Position 2 (high-order)	
6	Compensation 2 (low-order)	Outputs the current time compensation for <Compensation 2> on the automatic thermal distortion compensation screen for the production monitor.
7	Compensation 2 (high-order)	
8	-	Outputs 0.
9	-	Outputs 0.
10	-	Outputs 0.
11	-	Outputs 0.
12	-	Outputs 0.
13	-	Outputs 0.
14	-	Outputs 0.
15	Operation status	Outputs the <Operation status> on the automatic thermal distortion compensation screen for the production monitor. 0: Stopped 1: Operating

4.5.9 PLC Output No.4xx1 (Input/Output Servo (*-axis) Screen 1)

xx = 00: Spindle 01: X-axis 02: Y-axis 03: Z-axis 04: 4th-axis 05: 5th-axis 06: 6th-axis 07: 7th-axis 08: 8th-axis 09: M-axis 10: AT-axis 11: P1-axis 12: P2-axis 13: P3-axis 14: P4-axis

Output data	Name	Description
0	Servo power (low-order)	Servo power is output on the input/output – servo (spindle) screen. The updating cycle for this data is approximately 100 ms. The unit is in W. (Resolution is 10W) (NOTE) When an axis that is not the spindle is specified, the output data outputs a zero.
1	Servo power (high-order)	
2	Speed monitor (low-order)	Speed monitor on the input/output – servo (spindle) screen is output. (NOTE) When an axis that is not the spindle is specified, the output data outputs a zero.
3	Speed monitor (high-order)	
4	Pulse displacement amount (low-order)	Pulse displacement amount on the input/output – servo (spindle) screen is output. (NOTE) When an axis that is not the spindle is specified, the output data outputs a zero.
5	Pulse displacement amount(high-order)	
6	RST voltage	RST voltage on the input/output – servo (spindle) screen is output. (NOTE) When an axis that is not the spindle is specified, the output data outputs a zero.
7	PN voltage	PN voltage on the input/output – servo (spindle) screen is output. (NOTE) When an axis that is not the spindle is specified, the output data outputs a zero.
8	Encoder position interpolation count (low-order)	Outputs the encoder position interpolation count on the I/O servo (*-axis) screen. The unit is the number of times.
9	Encoder position interpolation count (high-order)	
10	-	Outputs 0.
11	-	Outputs 0.
12	-	Outputs 0.
13	-	Outputs 0.
14	-	Outputs 0.
15	-	Outputs 0.

4.5.10 PLC Output No.4xx2 (Input/Output Servo (*-axis) Screen 2)

xx = 00: Spindle 01: X-axis 02: Y-axis 03: Z-axis 04: 4th-axis 05: 5th-axis 06: 6th-axis 07: 7th-axis 08: 8th-axis 09: M-axis 10: AT-axis 11: P1-axis 12: P2-axis 13: P3-axis 14: P4-axis

Output data	Name	Description
0	Status word	Status word on the input/output – servo (*-axis) screen is output. Status word 1 on the input/output – servo (*-axis) screen is output.
1	Status word 1	
2	Position monitor (low-order)	Position monitor on the input/output – servo (*-axis) screen is output. The unit is in pulses.
3	Position monitor (high-order)	
4	Position deviation (low-order)	Position deviation on the input/output – servo (*-axis) screen is output. The unit is in pulses.
5	Position deviation (high-order)	
6	Torque command	Torque command on the input/output – servo (*-axis) screen is output. The unit is a percentage (%).
7	Torque monitor	Torque monitor on the input/output – servo (*-axis) screen is output. The unit is a percentage (%).
8	Monitor effective load	Effective load monitoring on the input/output – servo (*-axis) screen is output. The unit is a percentage (%).
9	Overload prediction	Overload prediction on the input/output – servo (*-axis) screen is output. The unit is a percentage (%).
10	Amplifier temperature	Amplifier temperature on the input/output – servo (*-axis) screen is output. The unit is in Celsius (°C).
11	Encoder temperature	Encoder temperature on the input/output – servo (*-axis) screen is output. The unit is in Celsius (°C).
12	Encoder communication error counter (low-order)	Encoder communication error counter on the input/output – servo (*-axis) screen is output. The unit is the number of times.
13	Encoder communication error counter (high-order)	
14	Encoder warning counter (low-order)	Encoder warning counter on the input/output – servo (*-axis) screen is output. The unit is the number of times.
15	Encoder warning counter (high-order)	

4.5.11 PLC Output No.4xx3 (Input/Output Servo (*-axis) Screen 3)

xx = 00: Spindle 01: X-axis 02: Y-axis 03: Z-axis 04: 4th-axis 05: 5th-axis 06: 6th-axis 07: 7th-axis 08: 8th-axis 09: M-axis 10: AT-axis 11: P1-axis 12: P2-axis 13: P3-axis 14: P4-axis

Output data	Name	Description
0	Device (0-15bit)	Device on the input/output – servo (*-axis) screen is output.
1	Device (16-31bit)	
2	Device (32-47bit)	
3	Blank	
4	Hardware version (low-order)	Hardware version on the input/output – servo (*-axis) screen is output.
5	Hardware version (high-order)	
6	Software version (low-order)	Software version on the input/output – servo (*-axis) screen is output.
7	Software version (high-order)	
8	Converter software version (low-order)	Converter software version on the input/output – servo (*-axis) screen is output.
9	Converter software version (high-order)	
10	Alarm code (low-order)	Alarm code on the input/output – servo (*-axis) screen is output.
11	Alarm code (high-order)	
12	Warning	Warning on the input/output – servo (*-axis) screen is output.
13	Motor cord	Motor cord on the input/output – servo (*-axis) screen is output.
14	Sensor divisions code	Sensor divisions code on the input/output – servo (*-axis) screen is output.
15	Sensor type code	Sensor type code on the input/output – servo (*-axis) screen is output.

4.5.12 Output Data No.500x (Workpiece Coordinate Zero Screen)

x = 0: G54 1: G55 2: G56 3: G57 4: G58 5: G59 6: External

Output data	Name	Description
0	X-axis workpiece coordinate (low-order)	Outputs the workpiece coordinate X-axis on the workpiece coordinate zero screen.
1	X-axis workpiece coordinate (high-order)	
2	Y-axis workpiece coordinate (low-order)	Outputs the workpiece coordinate Y-axis on the workpiece coordinate zero screen.
3	Y-axis workpiece coordinate (high-order)	
4	Z-axis workpiece coordinate (low-order)	Outputs the workpiece coordinate Z-axis on the workpiece coordinate zero screen.
5	Z-axis workpiece coordinate (high-order)	
6	A-axis workpiece coordinate (low-order)	Outputs the workpiece coordinate A-axis on the workpiece coordinate zero screen.
7	A-axis workpiece coordinate (high-order)	
8	B-axis workpiece coordinate (low-order)	Outputs the workpiece coordinate B-axis on the workpiece coordinate zero screen.
9	B-axis workpiece coordinate (high-order)	
10	C-axis workpiece coordinate (low-order)	Outputs the workpiece coordinate C-axis on the workpiece coordinate zero screen.
11	C-axis workpiece coordinate (high-order)	
12	-	Outputs 0.
13	-	Outputs 0.
14	-	Outputs 0.
15	-	Outputs 0.

All zeros are output when using conversation language.

4.5.13 Output Data No.51xx (Workpiece Coordinate Zero (Extended) Screen)

x = 01 to 48

Output data	Name	Description
0	X-axis workpiece coordinate (low-order)	Outputs the workpiece coordinate X-axis on the workpiece coordinate zero screen.
1	X-axis workpiece coordinate (high-order)	
2	Y-axis workpiece coordinate (low-order)	Outputs the workpiece coordinate Y-axis on the workpiece coordinate zero screen.
3	Y-axis workpiece coordinate (high-order)	
4	Z-axis workpiece coordinate (low-order)	Outputs the workpiece coordinate Z-axis on the workpiece coordinate zero screen.
5	Z-axis workpiece coordinate (high-order)	
6	A-axis workpiece coordinate (low-order)	Outputs the workpiece coordinate A-axis on the workpiece coordinate zero screen.
7	A-axis workpiece coordinate (high-order)	
8	B-axis workpiece coordinate (low-order)	Outputs the workpiece coordinate B-axis on the workpiece coordinate zero screen.
9	B-axis workpiece coordinate (high-order)	
10	C-axis workpiece coordinate (low-order)	Outputs the workpiece coordinate C-axis on the workpiece coordinate zero screen.
11	C-axis workpiece coordinate (high-order)	
12	-	Outputs 0.
13	-	Outputs 0.
14	-	Outputs 0.
15	-	Outputs 0.

All zeros are output when using conversation language.

CHAPTER 5

I/O INFORMATION

5.1 <PLC (I/O Control)> Screen
5.2 Layout and Operations on OM Monitor Screen
5.3 Global OM
5.4 Local OM
5.5 Registration OM Monitor/Test
5.6 Status Screen

5.1 <PLC (I/O Control)> Screen

When <1. I/O Control> is selected on **<**PLC (Menu)> screen, the <PLC (I/O control)> screen is displayed. On the <PLC (I/O control)> screen, the user can monitor the main OMs that make up the PLC system. The menu and supporting functions are as follows.

<1. Global OM monitor/test> : The user can reference and edit global OMs.
<2. High speed task 1 local OM monitor/test>to <6. Standard task 3 local OM monitor/test>
: The user can reference and edit local OMs for each task.
<7. OM registration monitor/test> : The user can register a given OM and reference and edit it.
<8. Status> : The user can reference the status of the PLC system.

Position	Name	Description
1	I/O control menu	Displays the menu for PLC setting and edit items.

Function key display area

Column	Position	Label	Description
1	(1)	PLC menu	Changes to the <PLC (Menu)> screen.

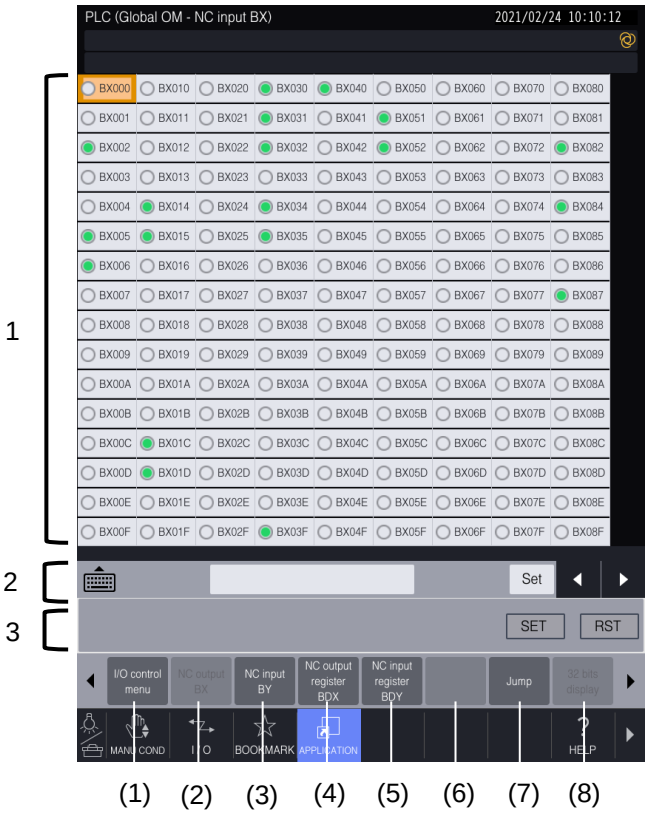
5.2 Layout and Operations on OM Monitor Screen

The basic layout and operations are the same on the monitor screen for both the global OMs and the local OMs for each task.

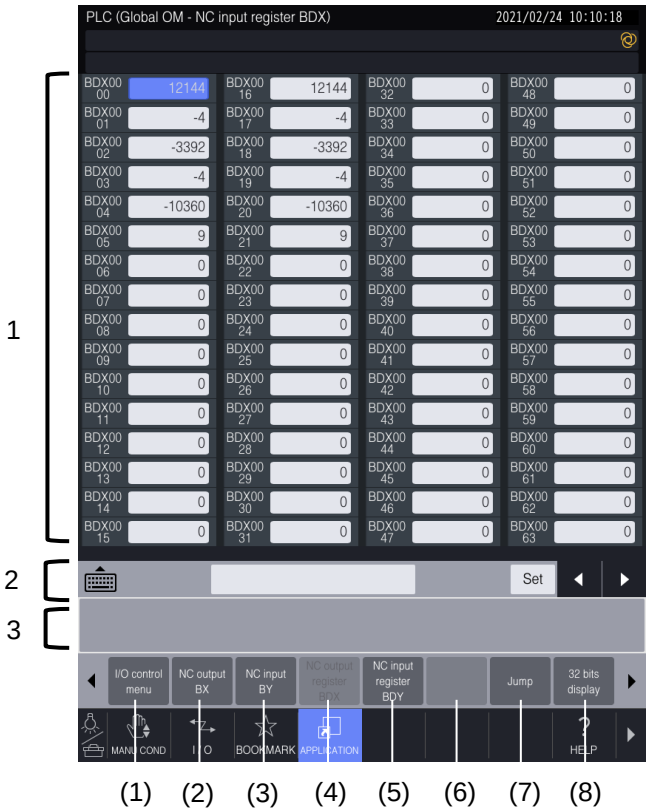
5.2.1 Description of Screen Display

There are two types of display formats on the monitor screen: bit display and word display.
On the bit display screen, each signal lights up when the signal is ON and the light turns off when the signal is OFF.
On the word display screen, the values are displayed in decimals, and the user can switch between a 16 bit and 32 bit display using the function keys. When using a 32 bit display, two register values are displayed for the 32 bit data. Therefore, only the odd number columns display the values, and the even number columns are blank.

Example of bit display screen



Example of word display screen



Position	Name	Description
1	OM data display	Displays the OM numbers and values.
2	Data input field	This field can be used to enter an OM address for the jump function and to change the OM value in the word display.
3	Instructions message area	Displays the system messages for the PLC system When the bit display is being used, the user can press the <SET>/ <RESET> key to turn it ON or OFF.

5.2.2 Description of Screen Operations

The user can change the pages by pressing the left and right arrow buttons. The user can turn the page to display an OM for referencing or editing.

5.2.3 Special Notes

The following covers special notes about the OM monitor screen.

1. When an OM has a set value in a control program stored in the built-in PLC, any value that is entered using a key operation is overwritten by the control program and is not reflected.
2. An <<Input data error>> occurs if a character or a numerical value out of range is entered into the data input field and the **[ENT]** key is pressed.

5.3 Global OM

The global OMs share values between the tasks. Among the global OMs, the user can use X/Y/M/D for general operations, and Brother has also provided BX/BY/BDX/BDY to use for special applications.
On the global OM screen, the user can reference and edit these OMs. However, only Y/M/D can be edited. Refer to “Chapter 4 OM” for a more detailed description about each OM.

Description of function operation

Column	Position	Label	Description
1	(1)	I/O control menu	Changes to the <PLC (I/O control)> screen.
	(2)	External input X	Changes to the <PLC (Global OM - External input X)> screen.
	(3)	External output Y	Changes to the <PLC (Global OM - External output Y)> screen.
	(4)	Internal relay M	Changed to the <PLC (Global OM - Internal relay M)> screen.
	(5)	Data register D	Changed to the <PLC (Global OM - Data register D)> screen.
	(6)		
	(7)	Jump	Jumps to the specified OM number (included in the address on the left side of the column) and displays the monitor results.
	(8)	16-bit / 32-bit display	Only enabled for D/BDX/BDY. Switches between the 16-bit display (-32768 to 32767) and the 32-bit display (-2147483648 to 2147483647). The key name is shown for the screen not currently displayed.

Column	Position	Label	Description
2	(1)	I/O control menu	Changes to the <PLC (I/O control)> screen.
	(2)	NC input BX	Changes to the <PLC (Global OM - NC input BX)> screen.
	(3)	NC output BY	Changes to the <PLC (Global OM - NC output BY)> screen.
	(4)	NC input register BDX	Changes to the <PLC (Global OM - NC input register BDX)> screen.
	(5)	NC output register BDY	Changes to the <PLC (Global OM - NC output register BDY)> screen.
	(6)		
	(7)	Jump	The function is the same as the key from column 1.
	(8)	16-bit / 32-bit display	The function is the same as the key from column 1.

5.4 Local OM

When an item between <2. High speed task 1 local OM monitor/test> and <6. Standard task 1 local OM monitor/test> is selected on the /<PLC (I/O control)> screen, the user can perform local OM operations for each task.

Among the local OMs, the user can use LM/LD/LT/LC for general operations, and SM/SD is provided to use for special OMs that indicate the PLC system status. There are values that are independent for each task. On the local OM screen, the user can reference and edit these OMs. Refer to “Chapter 4 OM” for a more detailed description about each OM.

Description of function operation

Column	Position	Label	Description
1	(1)	I/O control menu	Changes to the <PLC (I/O control)> screen.
	(2)	Local relay LM	Changes to the <PLC (**** - Local relay LM)> screen.
	(3)	Local register LD	Changes to the <PLC (**** - Local register LD)> screen.
	(4)	Local timer LT	Changes to the <PLC (**** - Local timer LT)> screen.
	(5)	Local counter LC	Changes to the <PLC (**** - Local counter LC)> screen.
	(6)		
	(7)	Jump	Jumps to the specified OM number (included in the address on the left side of the column) and displays the monitor results.
	(8)	16-bit / 32-bit display	Only enabled for LD/SD. Switches between the 16-bit display (-32768 to 32767) and the 32-bit display (-2147483648 to 2147483647). The key name is shown for the screen not currently displayed.

Column	Position	Label	Description
1	(1)	I/O control menu	Changes to the <PLC (I/O control)> screen.
	(2)	Special relay SM	Changes to the <PLC (**** - Special relay SM)> screen.
	(3)	Special register SD	Changes to the <PLC (**** - Special register SD)> screen.
	(4)		
	(5)		
	(6)		
	(7)	Jump	Jumps to the specified OM number (included in the address on the left side of the column) and displays the monitor results.
	(8)	16-bit / 32-bit display	Only enabled for D/BDX/BDY. Switches between the 16-bit display (-32768 to 32767) and the 32-bit display (-2147483648 to 2147483647). The key name is shown for the screen not currently displayed.

(NOTE) The asterisks (****) in the table refer to the task name (high speed task 1 to standard task 3).

5.4.1 Local Timer Screen

The display shows the local timer from LT0 to LT511 for the selected task in real time. The numerical values on the left side for the timer are the current values, and those on the right side are the set values.

A timer value can be edited, but the following message appears in the instructions area: <Changing the timer setting affects machine operation. OK to change?>. Press the <Yes> key to execute the edit operation.

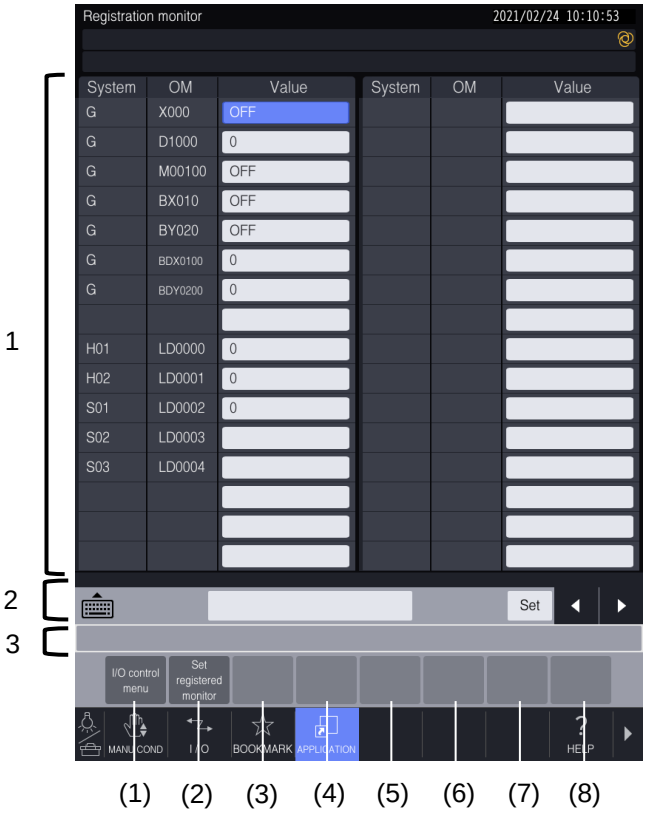
5.4.2 Local Counter Screen

The word display shows the local counter from LC0 to LC511 in real time. The counter can be edited.

The numerical values on the left side for the counter are the current values, and those on the right side are the set values. 0 is displayed for the counters that are not used. A counter value can be edited, but the following message appears in the instructions area: <Changing the counter setting affects machine operation. OK to change?>. Press the <Yes> key to execute the edit operation.

5.5 Registration OM Monitor/Test

On the <PLC (registered monitor)> screen, the user can select a given OM from the global or local OMs and reference or edit the OM list. However, the values cannot be edited for X, BX/BY, and BDX/BDY.
The user can register a maximum of 32 OMs.



5.5.1 Description of Screen Display

Position	Name	Description
1	Registration monitor	Displays the system, name and value for OMs that are registered for monitoring.
2	Data input field	This field is used for setting OM names and values.
3	Instructions area	Displays the system messages for the PLC system

5.5.2 Registered Monitor Setting Screen

Column	Position	Label	Description
1	(1)	I/O control menu	Changes to the <PLC (I/O control)> screen.
	(2)	Set registered monitor	Changes to the <Registered monitor setting> screen.
	(3)		
	(4)		
	(5)		
	(6)		
	(7)		
	(8)		

5.5.3 Registered Monitor Setting Screen

Column	Position	Label	Description
1	(1)	Back	Changes to the monitoring screen in the monitoring registration.
	(2)		
	(3)		
	(4)		
	(5)		
	(6)		
	(7)		
	(8)	Clear all registration settings	Deletes all monitoring settings that are registered.

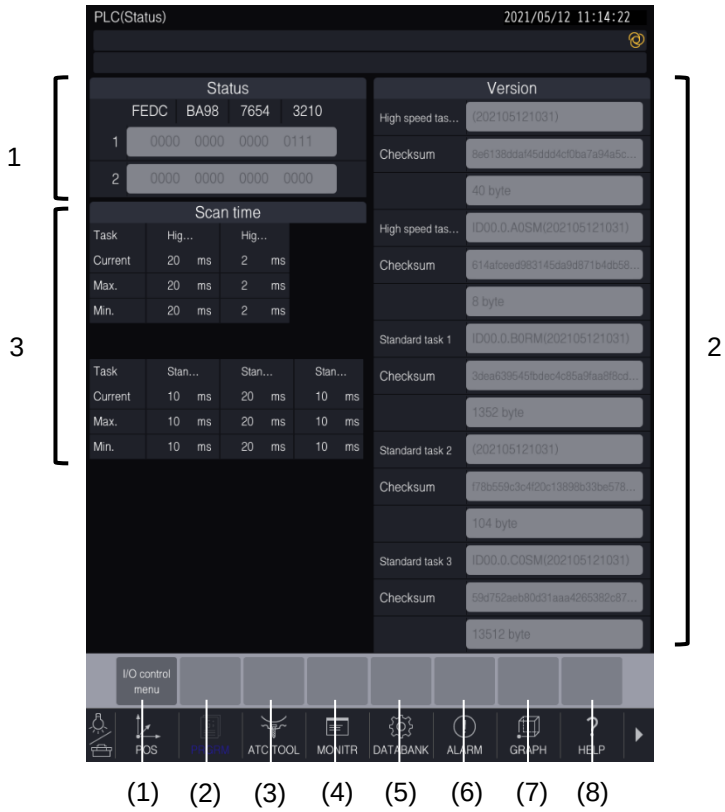
Follow the procedure below to register.

1. Tap to select an item for the OM you wish to register.
2. Select the desired task for monitoring from the combo box <System>.
   The selections and corresponding systems are as follows.
   G: 0: Global OM
   H01: High speed task 1
   H02: High speed task 2
   S01: Standard task 1
   S02: Standard task 2
   S03: Standard task 3
3. Select <OM>, enter the desired OM for monitoring into the data input field and press the **[ENT]** key.
   Only the following OM addresses can be set.
   When entering an address that cannot be set, the alarm <<Input data error>> is triggered.
   - When <G: global OM> is selected in the <System>:
     X/Y, M, D, BX/BY and BDX/BDY
   - When another selection besides <G: Global OM> is made for <System>:
     LM, LD, LT (Current value) and LC (Current value)

   In addition, when L is entered at the end of the set value for D and LD, it is displayed as a 32-bit value.
   Ex: D1000 → D1000 is shown in a 16-bit display
   D1000L → D1000 and D1001 are shown a 32-bit display
4. To delete each item, tap on the desired item for deletion and press the **[DEL]** key.
5. To delete all registration settings, press the <Clear all registration settings> key.
   The message <Clear all registered monitors?> appears in the instructions area.
   Select the <Yes> key to execute the clear operation, and select the <No> key to cancel the clear operation.

5.6 Status Screen

On the <PLC (Status)> screen, the user can reference the status of the PLC system.



Position	Name	Description
1	Status	Displays the names and values for the status data items. • Status 1 Bit 0 to 7: Shows the PLC startup status Bit 0: Initial status Bit 1: Local-main communication possible Bit 2: PLC execution possible Bit 3 to 6: Spare Bit 7: Error status Bit 8 to 15: Spare • Status 2 Bit 0 to 15: Spare
2	Version	Displays the version, checksum and size of the control program assigned for each task. This field is blank when the control program has not been assigned.
3	Scan time	Displays the scan time for each control program. The unit is in msec.

Function key display area

Column	Position	Label	Description
1	(1)	I/O control menu	Changes to the <PLC (I/O control)> screen.
	(2)		
	(3)		
	(4)		
	(5)		
	(6)		
	(7)		
	(8)		

(This page was intentionally left blank.)

CHAPTER 6

PROGRAM ORGANIZATION UNIT

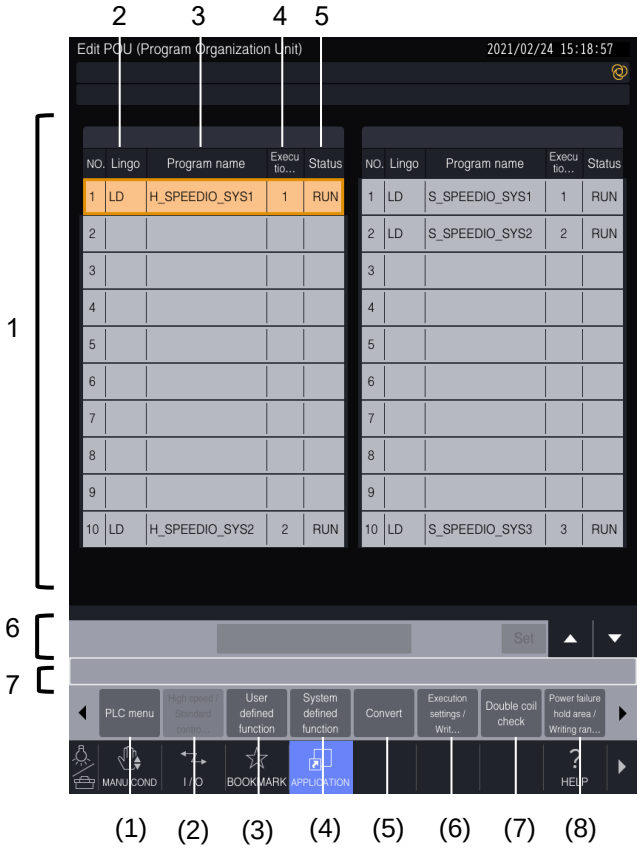
6.1 Runtime System Settings
6.2 Conversion
6.3 Global Variable Settings
6.4 Edit POU (Functions)
6.5 Check POU
6.6 Parameter Settings

6.1 Runtime System Settings

In the PLC system, the user can register up to 20 control programs as POUs: 10 high speed and 10 standard. Therefore, the user can assign two high speed and three standard control programs as executions programs to the tasks.

6.1.1 Edit POU (High Speed / Standard Programs)

On this screen, a user can create a new control program and carry out operations related to the control program files.



Description of screen display

Position	Name	Description
1	Registered program list	Displays a list of programs registered to each task. After tapping on a program, the edit screen is displayed for the registered program.
2	Language	The user can select a control program to create in a specific language. LD: LD program ST: ST program FBD: FBD program
3	Program name	Displays the name of the registered program. When a program has not been converted, <*> is displayed at the tail end of the program name.
4	Execution setting	Displays an execution program.
5	Status	Displays the status of an execution program. RUN: Running STOP: Stopped
6	Data input field	The user can enter a number for selecting a setting item.
7	Message area	Displays the system messages for the PLC system

Function key display area

Column	Position	Label	Description
1	(1)	PLC menu	Opens the <PLC (Menu)> screen.
	(2)	High speed runtime / Standard control program	This cannot be selected.
	(3)	User defined function	Opens the <PLC (Edit user defined function)> screen.
	(4)	System defined function	Opens the <PLC (Edit system defined function)> screen.
	(5)	Convert	Converts all the control programs.
	(6)	Execution settings / Write settings	Opens the <PLC (Program execution settings)> screen.
	(7)	Double coil check	Opens the <PLC (Double coil check result)> screen.
	(8)	Power failure hold area / Writing range check	Opens the <PLC (Power failure hold area / Writing range check)> screen.
2	(1)	PLC menu	Opens the <PLC (Menu)> screen.
	(2)	Check for writing to OMs outside writing range	Opens the screen <Result from writing check of OMs outside writing range>.
	(3)	(Blank field)	-
	(4)	Conversion mode setting	Opens the <PLC (Conversion mode setting)> screen.
	(5)	Global variable settings	Opens the <PLC (Global variable settings)> screen.
	(6)		
	(7)		
	(8)	File operations	The user can create a new control program as well as rename, copy/paste and delete an existing program.

(NOTE 1) If a control program or the function data structure is corrupt, the alarm <<Checksum error (PLC:***)>> (***: high speed/standard control program or user/system defined function) may trigger. At this time, the error <Checksum error (###)> appears in the instructions area on the <PLC (Edit POU (Program Organization Unit))> screen. Note that if "###" refers to a control program, the corresponding program number also appears.

(NOTE 2) If a checksum error is triggered in Note 1, then delete the corresponding control program/function.

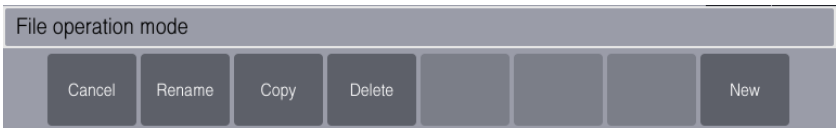
6.1.2 File Operation

After selecting the <File operations> key, the function key layout changes.
Here, the user can create a new control program as well as rename, copy/paste and delete an existing control program.

(NOTE) A name change and an overwrite operation or a deletion during pasting is not possible for control programs assigned to tasks in the program execution settings and for user/system defined functions used in control programs. The following is displayed in the instructions area.

- When a name change is carried out: <Change is not possible because the POU is being used in a task.>
- When an overwrite operation or a deletion is carried out during pasting: <Deletion is not possible because the function is being used in a task.>

Naming rules
The user can save a control program using a file name that is less than 16 characters.
Alphanumeric (does not distinguish between uppercase and lowercase) characters and the symbol "_" can be used in the file name. The user cannot specify a program name with a number as its first character.



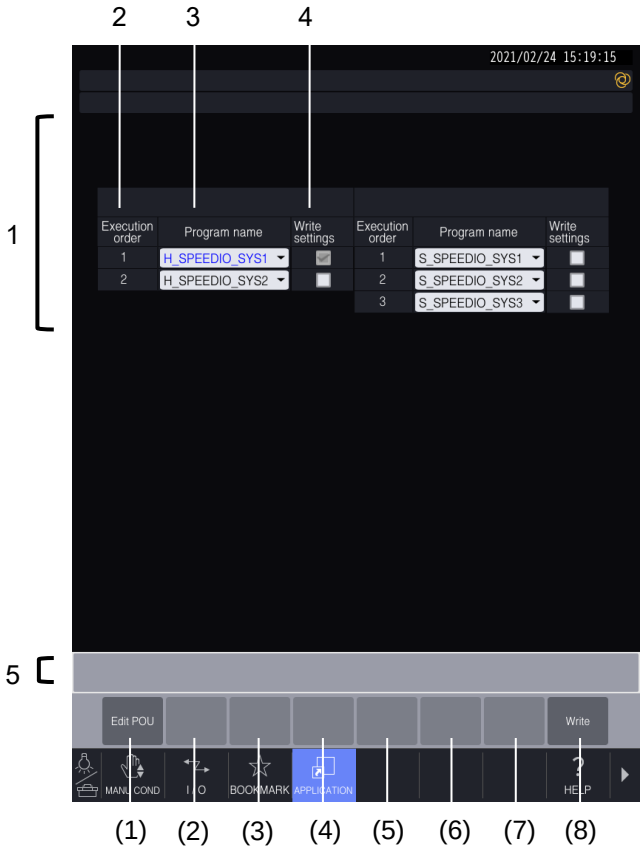
1. Renaming
   In the registered program list, move the cursor to the control program you wish to rename. Press the **<**Change name**>** key and enter a program name. Thereafter, the program name is changed.
2. Copying
   Select a line for copying and then select the **<**Copy**>** key. Thereafter, the message <Copy source program ** -Control program No. xx> appears in the message area (** refers to high speed or standard, and xx refers to the program number). Then, tap to select a copy destination and press the **<**Copy**>** key again. Thereafter, the copy source program will be copied to the destination. Note that if a line is selected (as the destination) where a program already exists and the **<**Copy**>** key is pressed, the popup message <Overwrite program?> is displayed. Press **<**Yes**>** to overwrite the data.
3. Delete
   Select the desired control program and then press the **[Delete]** key. Thereafter, only the function keys **<**Cancel**>** and **[Delete]** become valid. When the **[Delete]** key is pressed, the popup message <Delete ** control program – Program No. xx> is displayed (** refers to high speed or standard, and xx refers to the program number). Press **<**Yes**>** to delete the data.
4. Creating a new program
   Select a line to create a new program from the registered program list. After selecting the **<**New program**>** key, the function keys change. Then, press a language key to use for creating a new program. The <Language> that is selected is displayed.
   The system name is displayed for the program name at the selected line. Once a given name is entered into the data input field, it can be renamed. After renaming it, press the **[ENT]** key.

6.1.3 Program Edit

After selecting a line from the registered program list, the edit screen for the program appears and the user can edit the program. Refer to the edit screen chapter for the corresponding language for further details on the edit screen layout.

6.1.4 Program Execution Settings

In order to run a control program in the PLC system, the user must set the control program as an execution program in a task, and then the user must write the program into the PLC system.



Description of screen display

Position	Name	Description
1	Execution program settings	Sets an execution program to a task.
2	Execution order	This is the execution order when a task is run.
3	Program name	Selects and sets a control program as an execution program. When a program has not been converted, <*> is displayed at the tail end of the program name.
4	Write settings	Use **<**Write**>** to select a program to write to the PLC system. After the checkbox is ticked (ON), the program is written. However, the checkbox does not display when the set program is not converted.
5	Message area	Displays the system messages for the PLC system

Function key display area

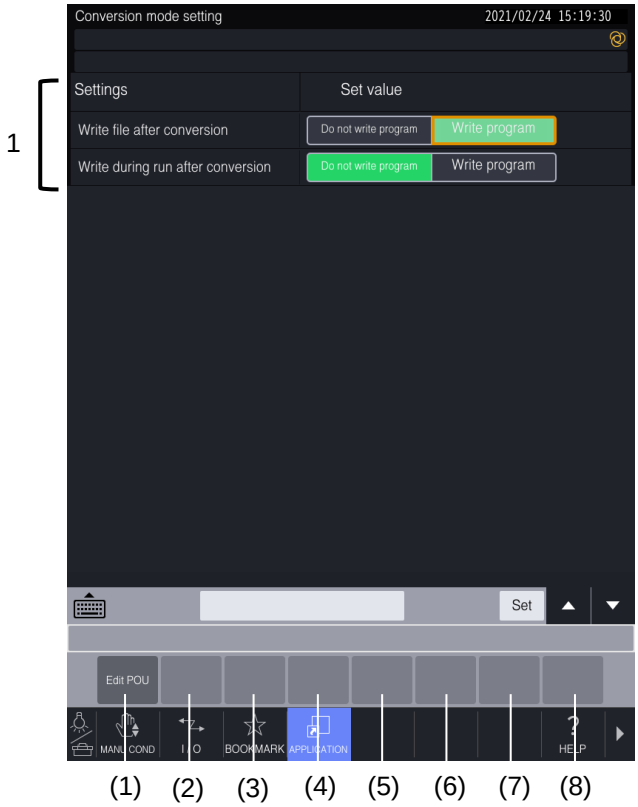
Column	Position	Label	Description
1	(1)	Edit POU	Opens the <PLC (Edit POU (Program Organization Unit))> screen.
	(2)		
	(3)		
	(4)		
	(5)		
	(6)		
	(7)		
	(8)	Write	Writes a control program to the PLC system when the checkbox is ticked (ON) for the <Write settings>.

6.2 Conversion

The conversion function converts a control program to an intermediate code for executing it in the PLC system. The program must be completely converted before it can be written as an execution program in the PLC system. There are two ways to convert a program.

(1) The user can convert a program/function on the edit screen for each language.
(2) All the programs can be converted at once with the **<**Convert**>** key

6.2.1 Conversion Mode Settings



Description of screen display

Position	Name	Description
1	Conversion parameter list	Sets the behavior or conditions when converted. The details will be discussed later.

Function key display area

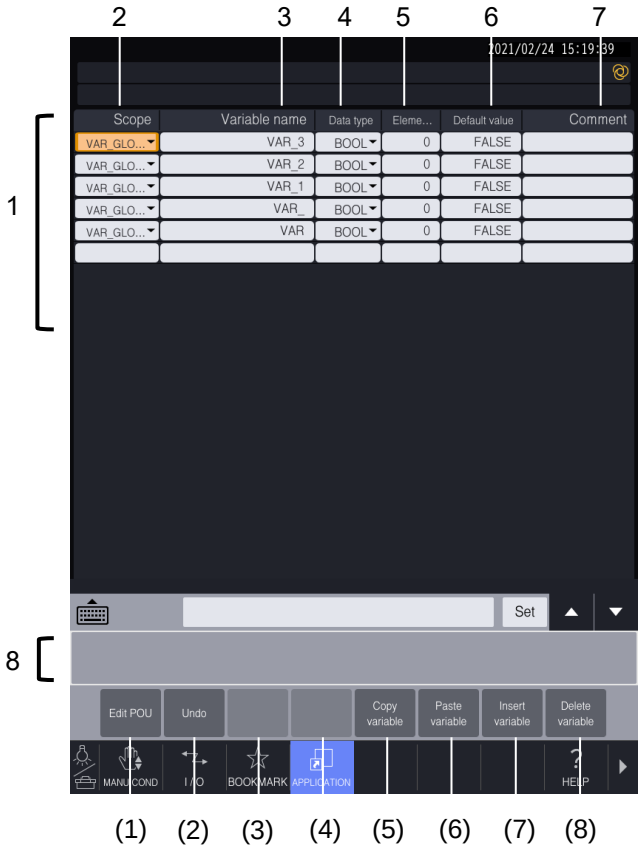
Column	Position	Label	Description
1	(1)	Edit POU	Opens the <PLC (Edit POU (Program Organization Unit))> screen.
	(2)		
	(3)		
	(4)		
	(5)		
	(6)		
	(7)		
	(8)		

Description of conversion parameters

No	Parameter name	Description
1	Write file after conversion	Sets whether to write or not write the file after conversion. 0: No 1: Yes
2	Write during run after conversion	Sets whether to write or not write the control program during run after conversion. 0: No 1: Yes

6.3 Global Variable Settings

Under the <PLC (Global variable settings)>, the user can define and edit global variables on the <PLC (Program edit - ST)> screen and the <PLC (Program edit - FBD)> screen.However, before editing, adding or deleting global variables on this screen, the user must change the status of all programs (programs that are defined as < VAR_EXTERNAL> in the variable editing area) that define and use the global variables to an unconverted status.



Description of screen display

Position	Name	Description
1	Global variable setting area	The user can set the global variables.
2	Scope	The user can set the valid scope of the variable. Only <VAR_GLOBAL> variables can be set.
3	Variable name	The user can enter the variable name.
4	Data type	The user can select and set the data type.
5	Element number	The user can set the element number of the variable. When the element number is 0, it is a single variable. When the element number is not 0, it is an array variable.
6	Default value	The user can set the default value of the variable.
7	Comment	The user can enter a comment for a variable.
8	Message area	Displays the system messages for the PLC system

Function key display area

Column	Position	Label	Description
1	(1)	Edit POU	Opens the <PLC (Edit POU (Program Organization Unit))> screen.
	(2)		
	(3)		
	(4)		
	(5)		
	(6)		
	(7)		
	(8)		

6.3.1 Available Data Capacity

The maximum data capacity available to create global variables is shown below per data type.
Maximum total capacity : 2048 variables
BOOL type : 1024 variables
Non-BOOL data type : 1024 variables

A maximum of 1024 global variables can be created for the BOOL type. In addition, a maximum of 1024 global variables can be created for all other data types besides the BOOL type. When the user attempts to create a variable, exceeding the total of 1024 variables, the message <The number of variables (non-BOOL type) has exceeded the upper limit. A variable cannot be created.> appears in the message display area.
In addition, when the user attempts to create a BOOL type variable, exceeding the total of 1024 variables, the message <The number of variables BOOL type variables has exceeded the upper limit. A variable cannot be created.> appears in the message display area and a variable cannot be created.

Note that the number of non-BOOL type variables that can be used varies according to the data 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
TIME	4	TOD	4	DT	4
STRING	16				

For example, when only creating WORD type variables, the user can create a maximum of 1024 variables. However, when only creating DWORD type variables, the user can create a maximum of 512 variables.
When creating arrays, the total element number for each array that can be created cannot exceed the maximum number of BOOL type or non-BOOL type variables.

6.4 Edit POU (Functions)

The user can manipulate units that can be integrated into the ST and FBD programs as functions. There are two types of functions: system defined function and user defined function. Only the user defined function can be created, edited and deleted.

(NOTE 1) Function cannot be selected as an execution program.
(NOTE 2) When variables are created in functions, they are reset each time the function is called.

6.4.1 Edit user defined function

On this screen, the user can create new user defined functions and edit them. A maximum of 924 user defined functions can be stored in the PLC system.

Description of screen display

Position	Name	Description
1	Current folder	Displays a folder where the displayed function is saved.
2	Function information	• Function name Displays the function name and folder name. • Language Displays the function language (ST/FBD). • Size Displays the amount of memory used by the function. • Date Displays the date and time the last time the function was saved in the NC.
3	Function list	Displays a list of user defined functions in the current folder. When a function is selected, the user can edit it.

Function key display area

Column	Position	Label	Description
1	(1)	PLC menu	Opens the <PLC (Menu)> screen.
	(2)	High speed / Standard control program	Changes to the <PLC (Edit POU (Program Organization Unit))> screen.
	(3)	User defined function	This cannot be selected.
	(4)	System defined function	Opens the <PLC (Edit system defined function)> screen.
	(5)		
	(6)		
	(7)		
	(8)	File operations	The user can create a new function as well as rename and copy/paste an existing file.

6.4.2 Edit System Defined Functions

On this screen, the user can browse through the system defined functions. A maximum of 100 system defined functions can be stored in the PLC system.

Description of screen display

Position	Name	Description
1	Current folder	Displays a folder where the displayed function is saved.
2	Function information	• Function name Displays the function name and folder name. • Language Displays the function language (ST/FBD). • Size Displays the amount of memory used by the function. • Date Displays the date and time the last time the function was saved in the NC.
3	Function list	Displays a list of system defined functions in the current folder. When a function is selected, the user can edit it.

Function key display area

Column	Position	Label	Description
1	(1)	PLC menu	Opens the <PLC (Menu)> screen.
	(2)	High speed / Standard control program	Opens the <PLC (Edit POU (Program Organization Unit))> screen.
	(3)	User defined function	Changes to the <PLC (Edit user defined function)> screen.
	(4)	System defined function	This cannot be selected.
	(5)		
	(6)		
	(7)		
	(8)	File operation	The user can create a new function as well as rename and copy/paste an existing file.

6.4.3 Number of Registered Variables

The maximum number of user or system defined functions (which call tasks that execute) that a user can create are shown below.

Maximum for BOOL type :4096
Maximum for non-BOOL type :4096

(NOTE 1) The user or system defined function itself (return value) is also included in the number of variables that are registered. For example, if the following has already been set, the number of BOOL type variables for the user defined function is counted as follows: 31 = 1 + 10 + 20.

- User defined function 1
  : Return value = BOOL type (count equal to 1 BOOL type variable)
  Number of registered BOOL type variables = 10
- User defined function 2
  : Return value = INT type (count equal to 1 INT type variable)
  Number of registered BOOL type variables = 20

(NOTE 2) Refer to “6.3.1 Available data capacity” for the quantity that is equal to one non-BOOL type variable and the counting method for the number of registered arrays.

6.4.4 File Operations

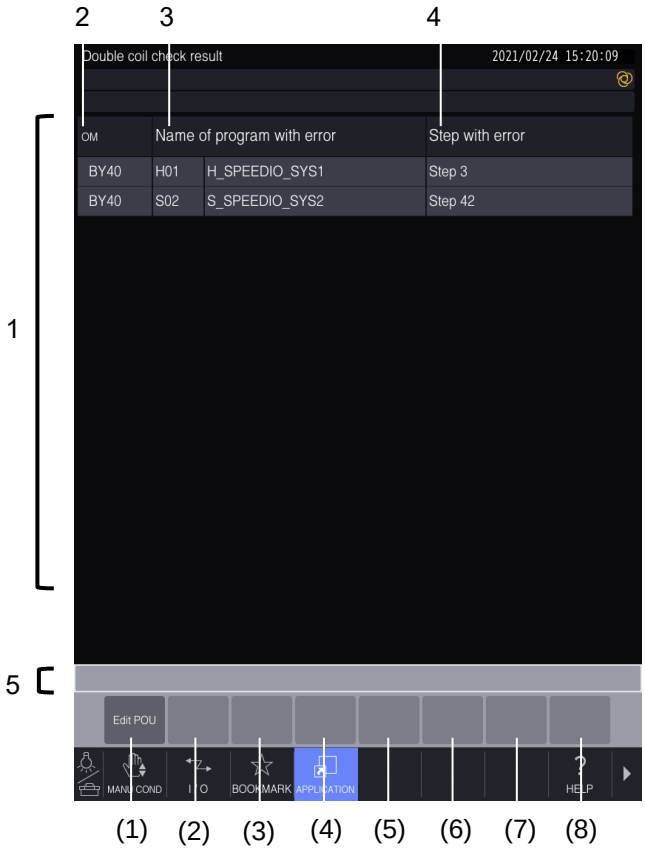
The file operations have the same functions as the keys (with same name) that appear on the <PLC (Edit POU (Program Organization Unit))> screen.

6.5 Check POU

6.5.1 Double Coil Check Result

A double coil check is carried out for the execution program when a control program is written and when the <PLC (Double coil check result)> screen appears.
The double coil check operation checks whether there is an output or not to the same OM in different execution programs.
If a check error is found, the program writing operation is not executed.
In addition, the OM and step number with the error are listed on the <PLC (Double coil check result)> screen. The edit screen can be displayed for the execution program with the error by selecting a line from the list.
When converting while engaged in write during RUN mode, the double coil check is performed. If a check error is detected in write during RUN mode, then the write during RUN mode is disengaged after program conversion. At this time, the program writing operation is not executed.

(NOTE) Double coils inside a single control program are not checked.



Description of screen display

Position	Name	Description
1	Double coil error list	Displays the location where a double coil error occurred. When a line is selected, the corresponding program edit screen is opened and the error location is displayed.
2	OM	Displays the OM where the error occurred.
3	Name of program with error	Displays the number and name of the control program. H: High speed control program S: Standard control program
4	Step with error	Displays the step number where the error occurred.
5	Message area	Displays the system messages for the PLC system

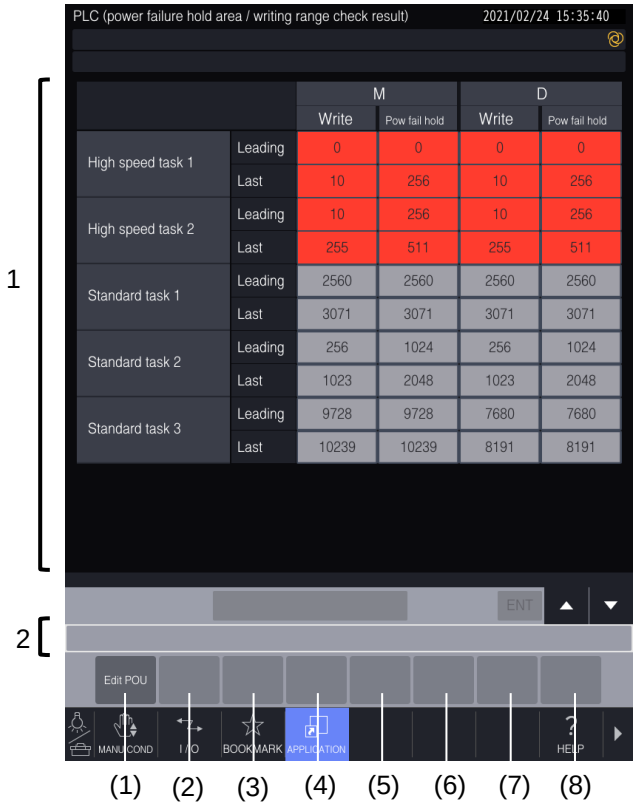
Function key display area

Column	Position	Label	Description
1	(1)	Edit POU	Changes to the <PLC (Edit POU (Program Organization Unit))> screen.
	(2)		
	(3)		
	(4)		
	(5)		
	(6)		
	(7)		
	(8)		

6.5.2 Results from Power Failure Hold Area / Writing Range Check

When a control program is written and when the <PLC (Power failure hold area / Writing range check result> screen is opened, the power failure hold area and writing range are checked for the execution program.

On the <PLC (Power failure hold area / Writing range check result)> screen, the display highlights the error locations in red where the power failure hold area and/or the writing range are the same or overlapping. Set the power failure hold area and writing range on the <PLC (Parameter setting)> screen for each control program. The user can display the <PLC (Parameter setting)> screen by pressing the <Leading> or <Last> number of the control program with an error. The user cannot change to the <PLC (Parameter setting)> screen even when <Leading> or <Last> is pressed for a control program (white highlight) without an error.



Description of screen display

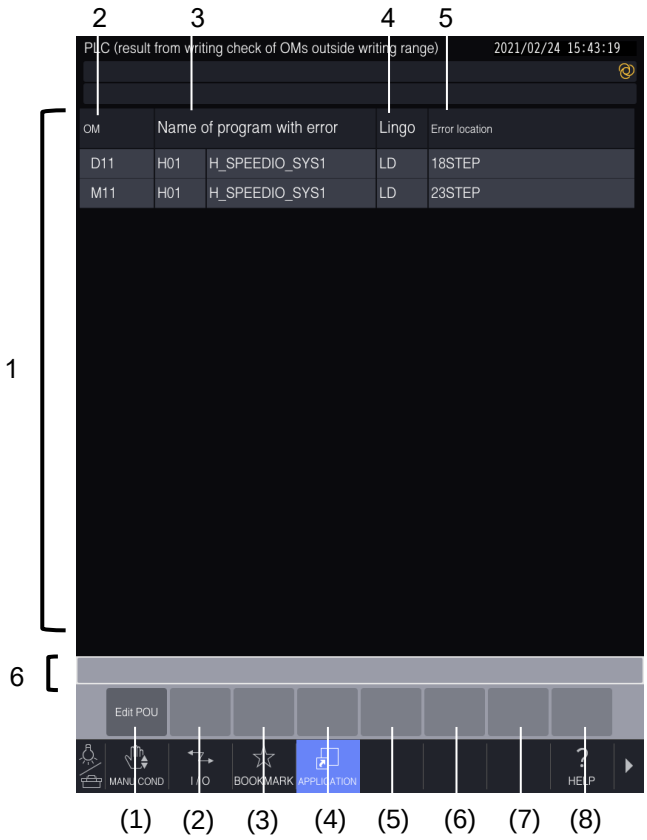
Position	Name	Description
1	List of errors for power failure hold area / writing range	Displays the <Write> and <Pow fail hold> ranges for each OM (M/D) that are set for the execution programs. The display highlights the error locations in red where the power failure hold area and/or the writing range are the same or overlapping. When a task does not have a set execution program, the field is blank.
2	Message area	Displays error messages specific to the PLC.

Function key display area

Column	Position	Label	Description
1	(1)	Edit POU	Changes to the <PLC (Edit POU (Program Organization Unit))> screen.
	(2)		
	(3)		
	(4)		
	(5)		
	(6)		
	(7)		
	(8)		

6.5.3 Result from Writing Check of OMs Outside Writing Range

When a program is written and when the screen changes to this screen, the writing check operation checks execution programs and whether or not they are written to OMs (M/D) outside the writing range. If an OM is written to outside of the writing range, the display shows the location of the error.
When there is an error, the program will not write.



Description of screen display

Position	Name	Description
1	Check results list	Displays the results of the writing check to OMs outside the writing range. When a line in the list is selected, a screen appears showing the program and location that correspond to that line.
2	OM	Displays OMs outside the writing range where a program is written.
3	Name of program with error	Displays the control program type and a two digit number on the left side. H: High speed control program S: Standard control program Therefore, “H07” refers to program No.7 for the high speed control program. The right side cell shows the program name.
4	Language	Displays the program language.
5	Error location	The display is different for each <Language>. LD: Displays the steps with errors. ST: Displays the line numbers with errors. FBD: Displays the network numbers with errors.
6	Message area	Displays the system messages for the PLC system

Function key display area

Column	Position	Label	Description
1	(1)	Edit POU	Changes to the <PLC (Edit POU (Program Organization Unit))> screen.
	(2)		
	(3)		
	(4)		
	(5)		
	(6)		
	(7)		
	(8)		

6.5.4 Check Lines with Errors

After selecting a line from the check results (on the <PLC (Double coil check result)> screen, on the <PLC (Power failure hold area / Writing range check result)> screen and on the screen <PLC (Result from writing check of OMs outside writing range)>), the program and/or step that caused the error is displayed.

6.6 Parameter Settings

6.6.1 <PLC (Parameter setting)> Screen

The user can set the parameters for the built-in PLC per control program. The following shows how to display the screens.

- Select **<**Parameter setting**>** from the control program edit screen
- Press the <Leading> or <Last> number from the list of power failure hold area / writing range errors that is found on the <PLC (Power failure hold area / Writing range check result)> screen
  Refer to “6.5.2 Results from power failure hold area / writing range check” for further details.

  The user can save data when changing from this screen to another screen. However, any edits will not be reflected in the built-in PLC motions until after the program is written.

  When creating a control program that uses the power failure hold area and/or writing range, first, check the power failure hold area and writing range settings for all of the control programs assigned to tasks.

  (To customers who transferred their ladder programs from CNC-C00 to CNC-D00)
  In the CNC-D00 specification, strictly speaking, an error is triggered in the PLC program when there is a coil or output for an OM outside of the writing range. When transferring ladder program data from CNC-C00, if the alarm <<There is a control program outputting outside of the writing range>> is triggered prohibiting operation, then we recommend re-setting the power failure hold area and writing range in the circuit.

Description of screen display

Position	Name	Description
1	Parameter list	Each parameter is for the built-in PLC.

Function key display area

Column	Position	Label	Description
1	(1)	<Program edit>	Opens the control program edit screen for the source. When the parameter is changed and this key is pressed, the <Save the setting data?> dialog appears. Select <Yes> to save the setting change. Select <No> to discard the setting change.
	(2)	-	-
	(3)	-	-
	(4)	-	-
	(5)	-	-
	(6)	-	-
	(7)	-	-
	(8)	-	-

6.6.2 Setting Items

The setting items are shown below. Enter the scan time for each ladder.

No.	Item	Setting range	Description
1	<Scan time>	High speed task: 1~100(ms) Standard task: 10~1000(ms)	Enter the scan time for each task.
2	<Monitor WDT>	<0: No> <1: Yes>	Enable or disable the watch dog timer (WDT) monitor setting.
3	<WDT monitor time>	High speed task: 10~1000(ms) Standard task: 100~10000(ms)	Set the watch dog timer (WDT) monitor time. Set a value that is larger than the scan time.
4	<Operation method in case of WDT error>	<0: Stop> <1: Continue>	Enable or disable the built-in PLC stop setting in the event that a WDT error occurs.
5	<Operation method for calculation error>	<0: Stop> <1: Continue>	Enable or disable the built-in PLC stop setting in the event that a calculation error occurs.
6	<Operation method for command exec error>	<0: Stop> <1: Continue>	Enable or disable the built-in PLC stop setting in the event that a command execution error occurs.
7	<Output for STOP->RUN>	<0: Output state before STOP> <1: Output OFF>	Set the output method for the external output relay (Y) when STOP changes to RUN in the programming software or when in monitor mode.
8	<Specify global OM (M) power failure hold (latch) area>	0 to 10239	Set the leading address and last address for the global OM (M) power failure hold area.
9	<Specify global OM (D) power failure hold (latch) area>	0 to 8191	Set the leading address and last address for the global OM (D) power failure hold area.
10	<Specify global OM (M) writing range>	0 to 10239	Specify and set the leading address and last address for the global OM (M) writing range.
11	<Specify global OM (D) writing range>	0 to 8191	Specify and set the leading address and last address for the global OM (D) writing range.

(NOTE 1) Do not change the set values for high speed task 2 and standard task 3.

(NOTE 2) The user can select whether to stop or continue operation of the built-in PLC control program when one of the following errors occurs: <WDT error>, <Computation error> or <Command execution error>. However, an alarm is triggered and stops machine operation regardless of the parameter settings.

(NOTE 3) When the <WDT monitor time> value is smaller than the <Scan time>, the error <<Input data error>> is triggered.

(NOTE 4) After saving the set values for the power failure hold area, if the power is turned OFF before writing the data, the power failure hold area data (after the change) will be restored but any data outside of the range will be cleared. In addition, when the power is turned OFF, if the power failure hold area is the same or overlapping between control programs (that are assigned to tasks), then all the data in the power failure hold area will be cleared in the control program where the area is the same or overlapping.

(NOTE 5) If a control program is set to output an OM that is outside of the writing range and then an attempt is made to write the program, the writing operation will not be executed and the following message will appear in the instructions area: <OM that has not been set in the writing area is being used in the program. Check the screen: Result from writing check of OMs outside writing range.>.

(This page was intentionally left blank.)

CHAPTER 7

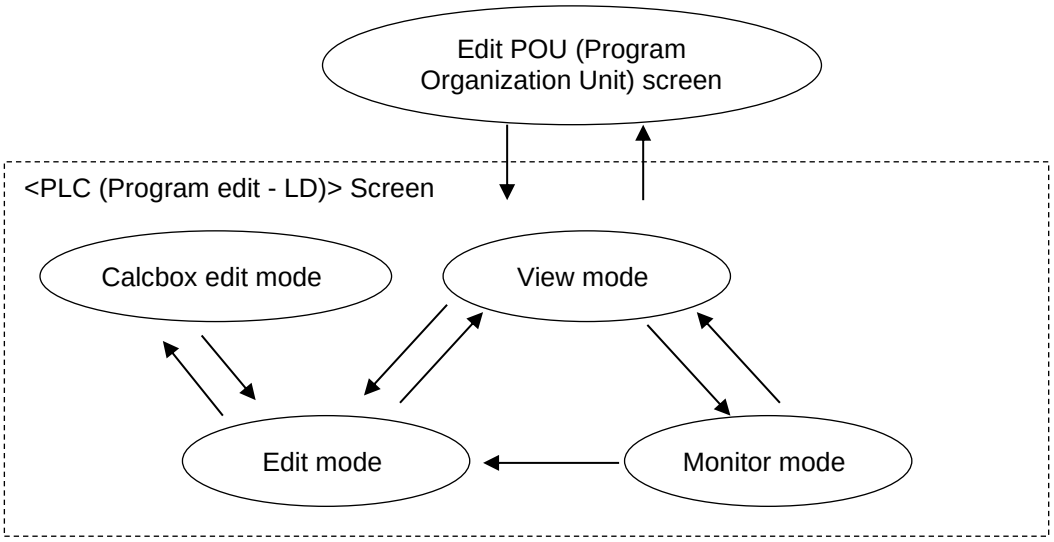
PROGRAMMING FUNCTION LD

7.1 <PLC (Program edit - LD)> Screen

7.1 <PLC (Program edit - LD)> Screen

7.1.1 Overview

On this screen, the user can edit and monitor programs where the <Language> is set to <LD> for high speed and standard control programs. The <LD> refers to an LD (ladder) program.
The user can open the <PLC (Program edit - LD)> screen by selecting the aforementioned program on the <PLC (Edit POU (Program Organization Unit)> screen.
The <PLC (Program edit - LD)> screen has four modes available: view, edit, monitor and calcbox edit modes. The mode changes to view mode immediately after changing screens. Each mode can be switched using the function keys.

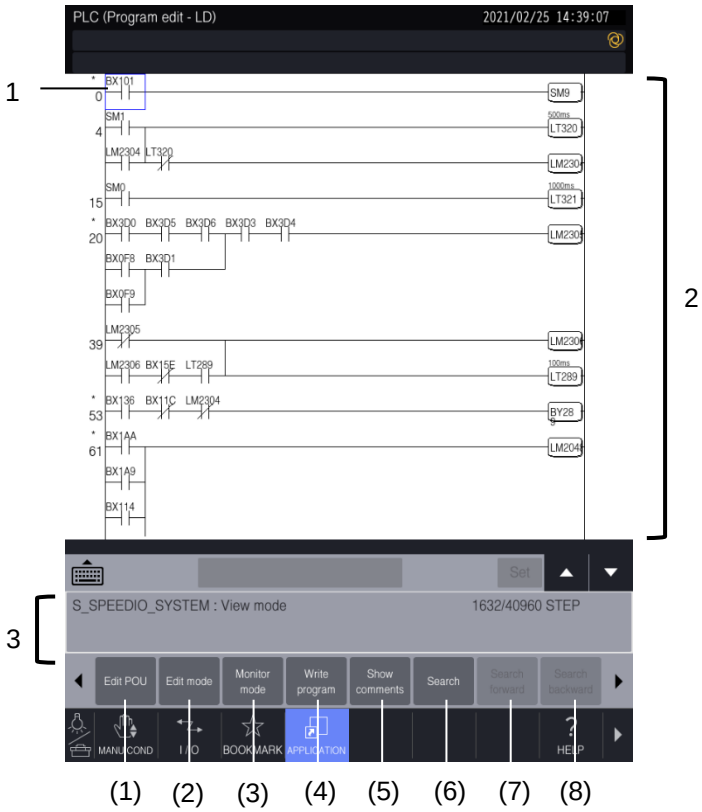


(NOTE) When editing and writing a LD program, setting parameters, setting the power failure hold / writing range or setting the conversion mode, be sure to set the user parameter <PLC program protection>> to <0: Invalid>.
However, when editing and writing for high speed and standard control program 10 (assigned to standard task 3 and high speed task 2), make sure <Parameter change> on the <I/O (extension I/O maintenance)> screen is set to <1: Writable>. Then, set <PLC program protection> to <0: Invalid>.

7.1.2 View Mode

7.1.2.1 Screen layout

The following shows the screen layout for view mode.



Description of screen display

Position	Name	Description
1	Selection cursor	Displays the range for the current selection.
2	Program display area	Displays the program. The user can change the display by moving the cursor or by using the <Previous page> key or the <Next page> key.
3	Instructions area	Displays items such as operation instructions or a program syntax error. In addition, the step number displays for the program in the <Used steps/Maximum steps>.

Function key display area

Column	Position	Label	Description
1	(1)	<Edit POU>	Opens the <PLC (Edit POU (Program Organization Unit))> screen.
	(2)	<Edit mode>	Changes to the program edit mode.
	(3)	<Monitor mode>	Changes to the program monitor mode. This grays out when the displayed program is not an execution program.
	(4)	<Write program>	Writes the program into the built-in PLC. However, this key grays out when the execution program is not registered. Refer to “7.1.2.2 Write program” for further details.
	(5)	<Show comments>	Shows or hides the comments. Refer to “7.1.2.8 Comment” for further details.
	(6)	<Search>	Searches a program. There are three types of searches: an OM, a step and a command search. Refer to “7.1.2.7 Search” for further details.

Column	Position	Label	Description
	(7)	<Search forward>	Searches the next ladder block after executing a search for an OM or command. Refer to “7.1.2.7 Search” for further details.
	(8)	<Search backward>	Searches the previous ladder block after executing a search for an OM or command. Refer to “7.1.2.7 Search” for further details.
2	(1)	<Edit POU>	Opens the <PLC (Edit POU (Program Organization Unit))> screen.
	(2)	-	-
	(3)	-	-
	(4)	<New program>	Discards the edited program and changes the program to a default or new program status. Refer to “7.1.2.5 New program” for further details.
	(5)	<Save>	Saves the program that is being edited. Refer to “7.1.2.4 Save” for further details.
	(6)	<Read program>	Discards the edited program and overwrites the content of the program (prior to editing). Refer to “7.1.2.3 Read program” for further details.
	(7)	<Cancel new program>	Cancels the **<**New program**>** operation and returns the program content back to previous state. After the new program is created, this key is valid until the program is saved with the <Save> key. This grays out in all other situations. Refer to “7.1.2.6 Cancel new program” for further details.
	(8)	<Parameter setting>	Displays the <PLC (Parameter setting)> screen.

7

7.1.2.2 Write program

A program registered as an execution program is written to the built-in PLC. This cannot be selected for any other program. The procedure is shown below.

1. Select <Write program>.
2. If the program is stopped, the popup message <Program will be written. Proceed?> is displayed. If the program is executing, the popup message <Stop? Program resumes automatically after writing is completed.> is displayed.
   Select <Yes> to proceed with the operation. Select <No> to cancel the operation.
3. This executes and writes the program. During the writing operation, the message <Writing.> appears in the instructions area. After the writing operation is completed, the message <Completed.> appears.
4. When the above operation is carried out while the program is executing, the PLC automatically starts up.

(NOTE 1) The alarm <<Control program stopped (****)>> is triggered immediately after the program write operation is executed. “****” refers to high speed tasks 1, 2 or standard tasks 1, 2 or 3.

(NOTE 2) Do not execute the program write operation while the machine is operating.

(NOTE 3) If the program write operation is executed while the control program is stopped, then the control program stays stopped after the writing operation.

(NOTE 4) When a program is written, a few checks are carried out, for example, a double coil check, a writing range check of the power failure hold area and a writing check of the OMs outside the writing range.

7.1.2.3 Read program

The user can discard the edited program, read the content of the saved program and then re-edit it. The procedure is shown below.

1. Select <Read program>.
2. When a program is being edited, if there is any difference with the program being read, the popup message <<Program being edited will be removed. OK?>> is displayed.
   Select <Yes> to proceed with the operation. Select <No> to cancel the operation.
3. The program read operation is executed. When the read operation is complete, the message <Completed.> appears in the instructions area. To edit the program, switch to edit mode.

(NOTE) It may take time for the program reading operation to be completed. The processing time takes longer when there are many steps.

7.1.2.4 Save

The <Save> operation saves the program being edited as a file. The user can call a saved program with the <Read program> operation.

7.1.2.5 New program

The user can discard the edited program, and re-edit the program again starting from a default or new program status. The procedure is shown below.

1. Select <New program>.
2. The popup message <Program being edited will be removed. OK?> is displayed. Select <Yes> to proceed with the operation.
   Select <No> to cancel the operation.
3. A new program is created. To edit the program, switch to edit mode.

(NOTE 1) After a new program is created, the <Undo> key will not change the new program content back to the previous status (program content prior to using the new program key).

(NOTE 2) After a new program is created, any data that was copied or edited (before creating a new program) is cleared in the program.

7.1.2.6 Cancel new program

After the <New program> key is used, the user can cancel the new program operation and restore the program content back to the content prior to using the new program key. The procedure is shown below.

1. Press <Cancel new program>.
2. The popup message <New program content will be discarded, and the program will be restored to the content prior to editing. Proceed?> is displayed. Select **<**Yes**>** to proceed with the operation. Select <No> to cancel the operation.
3. The program content is restored to its previous status (program content prior to using the new program key). To edit the program, switch to edit mode.

(NOTE) Once **<**Save**>** is used to save the program file after creating a new program, the user will not be able to restore the program content back to the previous status (program content prior to using the new program key).

7.1.2.7 Search

In view mode and monitor mode (discussed later), the user can conduct a search in a program. After pressing <Search>, enter a “step number”, “OM”, “command” or a combination of these into the data input field in order to specify the search. Once the search is executed, it starts searching from the beginning of the program and the cursor moves to the corresponding block. If there is no corresponding OM, then the message <Corresponding program is missing.> appears in the instructions area.

(NOTE 1) In the search function, a numerical value cannot be specified when it is greater the step number at the program end. The message <Corresponding STEP is missing.> will appear in the instructions area.

(NOTE 2) In the search function, a negative value, character string or multiple bits cannot be specified. The message <Invalid setting.> will appear in the instructions area. For OMs that are specified by multiple bit access, search for the smallest OM number specified.

There are three search functions, which are show below.

1. Step search
2. OM search
3. Command search

1. Step search

   After entering a step number into the search input field and pressing **[ENT]**, the cursor will move to the beginning of the circuit block (that includes the entered step).

   Ex: The cursor moves to the following blocks when steps 0, 10 and 20 are entered for the beginning circuit blocks.
   Enter 0 to 9 → 0 step block
   Enter 10 to 19 → 10 step block
   Enter 20 or higher→ 20 step block

   (NOTE 1) Steps after the END command cannot be specified. The message <Corresponding STEP is missing.> will appear in the instructions area.

   (NOTE 2) Values that exceed the maximum step number cannot be specified. The message <Invalid setting.> will appear in the instructions area.

2. OM search

   After entering an OM into the search input field and pressing **[ENT]**, the cursor will move to the specified OM (when the specified OM exists). When <Search forward> is pressed at this time, the search toggles to the next OM search result. When <Search backward> is pressed, the search toggles to the previous OM search result.
   In addition, when an “OM name” and “step number” are entered, the user can search in the corresponding circuit block after the specified step number.
   However, if the specified OM does not exist, then the message <Corresponding program is missing.> appears in the instructions area.

3. Command search

   There are two procedures for command input:
   (1) Specify only the command (mnemonic)
   Ex: When “PLS” is specified, a search is carried out looking for all PLS commands.
   (2) Specify the command (mnemonic) + OM
   Ex: When “PLS M0” is specified, a search is carried out looking for a PLS command that specifies M0.

   When one of the above procedures is used to enter a command in the search input field and **[ENT]** is pressed, the cursor will move to the corresponding command (when that command exists). When <Search forward> is pressed at this time, the search toggles to the next command search result. When <Search backward> is pressed, the search toggles to the previous command search result.
   In addition, when an “PLS step number” is entered, the search is conducted in the circuit block after the specified step number.
   However, if the command does not exist, then the message <Corresponding program is missing.> appears in the instructions area.

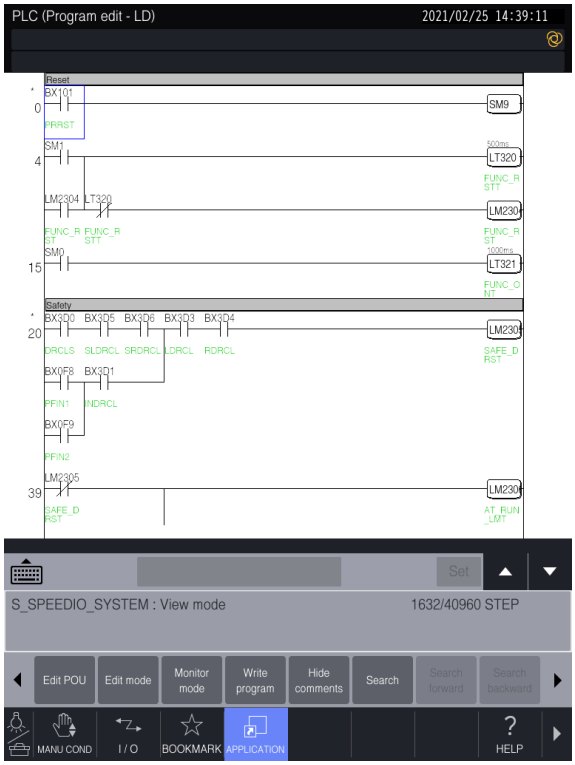
7.1.2.8 Comment

In view mode and monitor mode, the user can show or hide the comments, and in edit mode, the user can edit an OM or calcbox comment.
There are four types of comments, which are shown below.

(NOTE) There is no coil remark function.

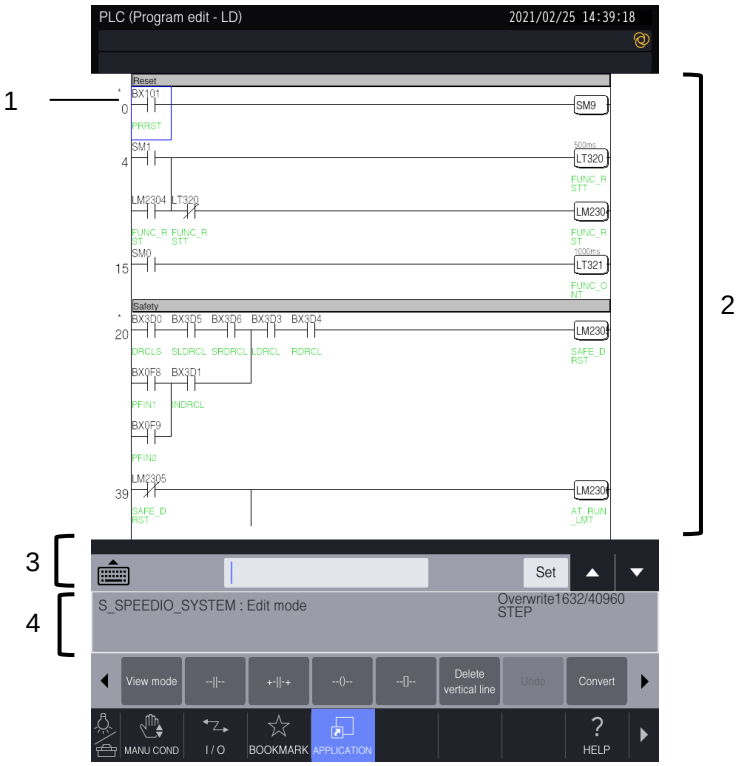
Name	Description
OM comment	A comment can be provided for each OM. A maximum of 16 characters can be used in the comment. In this document, the term “comment” refer to an OM comment.
Inline comment	A comment can be provided for each circuit block. A 64 character comment can be used at each circuit block, on a maximum of 64 circuit blocks. The comment cannot be edited in the built-in PLC.
Label comment	A comment can be provided for labels (L0 to 255). A 64 character comment can be used for one label. The comment cannot be edited in the built-in PLC.
Calcbox comment	A comment can be provided for a calcbox (0 to 255). A 20 character comment can be used for one calcbox. The comment always displays regardless of the setting to show or hide the comment.

Comment display image



7.1.3 Edit Mode

7.1.3.1 Screen layout



Description of screen display

Position	Name	Description
1	Selection cursor	Displays the range for the current selection.
2	Program display area	Displays the program. The user can change the display by moving the cursor or by using the <Previous page> key or the <Next page> key.
3	Data input field	The user can enter data in order to edit a program.
4	Message display area	Displays items such as operation instructions or a program syntax error. In addition, the step number displays for the program in the <Used steps/Maximum steps>.

Function key display area

Column	Position	Label	Description
1	(1)	<View mode >	Changes to view mode.
	(2)	<--||-->	Inserts contact *a* at the position of the cursor.
	(3)	<+-||-+>	Inserts contact *a* (OR) at the position of the cursor.
	(4)	<--()-->	Inserts a coil at the position of the cursor.
	(5)	<--**[]**-->	Inserts a data command or comparison command at the position of the cursor.
	(6)	<Delete vertical line>	Deletes a vertical line where the cursor is.
	(7)	<Undo>	Cancels the program edit operation and returns it back to the previous state. Refer to “7.1.3.9 Undo edit operation” for further details.
	(8)	<Convert>	Converts a program being edited to binary. Refer to “7.1.3.10 Convert” for further details.

Column	Position	Label	Description
2	(1)	<View mode>	Changes to view mode.
	(2)	<--|/|-->	Inserts contact *b* at the position of the cursor.
	(3)	<+-|/|-+>	Inserts contact *b* (OR) at the position of the cursor.
	(4)	<--|P|-->	Inserts a rising edge contact at the position of the cursor.
	(5)	<--|N|-->	Inserts a falling edge contact at the position of the cursor.
	(6)	<Edit comment >	The user can edit an OM comment in the selection area. Refer to “7.1.3.12 Edit comment” for further details.
	(7)	<Copy a rung>	Records or copies the rung in the selection area.
	(8)	<Paste a rung>	Pastes the copied rung in the program.
3	(1)	<View mode>	Changes to view mode.
	(2)	<Edit CalcBox>	Changes to the calcbox edit mode. Refer to “7.1.5 Calcbox edit mode” and “7.1.5.2 Edit calcbox” for further details.
	(3)	<--|PI|-->	Inserts a rising edge NOT contact at the position of the cursor.
	(4)	<--|NPI|-->	Inserts a falling edge NOT contact at the position of the cursor.
	(5)	<Show comments>	Shows or hides the comments. Refer to “7.1.3.12 Edit comment” for further details.
	(6)	<Insert >	Changes the input format to insert/overwrite mode. Refer to “7.1.3.6 Insert/Overwrite mode” for further details.
	(7)	<Delete>	Deletes the selected OM.
	(8)	-	-

7.1.3.2 Key assignment

In edit mode, the keys and functions are assigned as shown in the table below.

Key	Setting	Insert	Delete	Cancel	A to Z	Numbers	Cursor	Page
Shift OFF	Enter	Change to insert/ overwrite	Delete symbol	Esc	Enter for command / OM / comment		Move cursor	Change page
Shift ON	-	Insert rung	Delete rung	-	Enter for command / OM / comment		Write ruled line	-

7.1.3.3 Symbol input

To input a symbol, select the desired function, enter a command (OM) and then press **[ENT]**. In addition, to input a ruled line (horizontal or vertical), first move the cursor to the start position. Then, while pressing **[SHIFT]**, use the cursor keys to move the cursor in the desired direction to draw the ruled line.

After entering the command (OM) into the data input field and pressing **[ENT]**, the symbol, command and OM at the cursor position will change. The background of the block that has been changed will turn gray.

- When inputting a coil and data command, the horizontal line will automatically fill in.
- If the user tries input an OM and command that cannot be used, an input error will occur.

7.1.3.4 Multiple bit access for bit OM

Not only can a singular bit memory be accessed but a series of multiple bit memories can also be accessed.

A “W” and “L” are respectively inserted after the bit OM number for 16-/32-bit access.

Input example	Description
M16W	Accesses a 16-bit segment from M16 to M31.
M64L	Accesses a 32-bit segment from M64 to M95.

“:n” (n is the bit number) is inserted after the bit OM in the bit access. It is possible to specify a maximum of 32 in 4-bit units for n.

Input example	Description
M4:4	Accesses a 4-bit segment from M4 to M7.
M32:20	Accesses a 20-bit segment from M32 to M51.

- Set the OM number when using multiple bit access in multiples of 4.
- When the OM is outside the range due to multiple bit access, an input error occurs.
- When specifying a 32-bit value for the constant, L is inserted at the tail end.
  Ex: When a constant value of 40000 is entered, 40000L is shown.

7.1.3.5 32-bit access for word OM

The memory range for a word OM is not 16-bit but 32-bit.

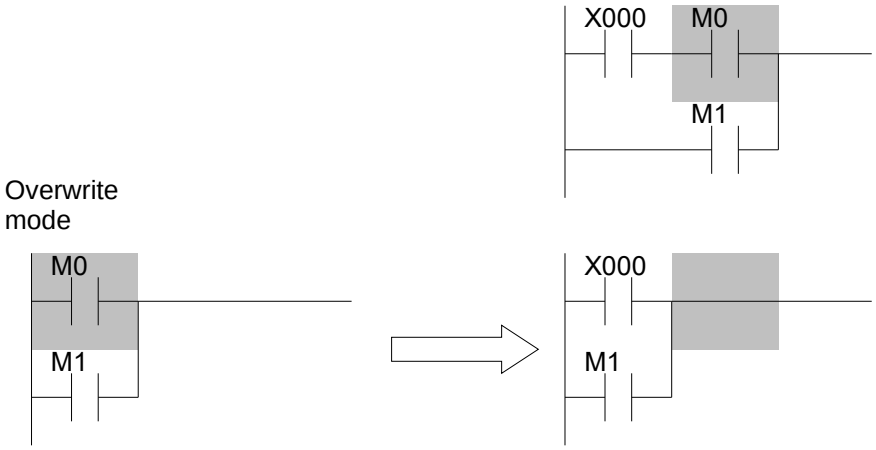
An “L” is inserted after the word OM number.

Input example	Description
D16L	Accesses a 32-bit segment for D16 and D17.

- When the OM is outside the range due to the 32-bit access to the word OM, an input error occurs.
- There is no start number restriction when the OM number is within range for 32-bit access of a word OM.
- Insert an “L” at the tail end of the numerical value when performing 32-bit changes to the current value for the 32-bit access of a word OM.
  Ex: “32768L”

7.1.3.6 Insert / Overwrite mode

Pressing **[INS]** while in edit mode switches the setting back and forth between insert mode and overwrite mode. In insert mode, when a symbol is input, the existing symbol shifts to the right.

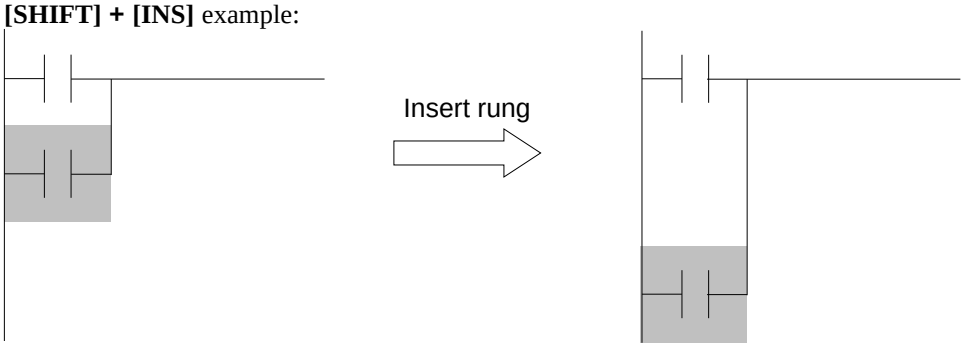


7.1.3.7 Rung operations

The user can perform operations on each rung by pressing one of the following keys.

Key	Description
[SHIFT] + [INS]	Adds an empty rung at the cursor position. When an operation is carried out on a rung with a vertical line, it adds a rung while automatically filling in the vertical line.
[SHIFT] + [DEL]	Deletes a rung.
<Copy a rung**>**	Records or copies the program rung at the cursor position.
<Paste a rung**>**	Pastes the recorded or copied program rung. The paste operation can be performed any number of times. This operation behaves differently in insert and overwrite mode. It inserts the copied rung directly before the rung with the cursor in insert mode. When pasting a rung in overwrite mode, it replaces the ladder where the rung where the cursor is, with the copied rung.

[SHIFT] + [INS] example:



(NOTE 1) If a rung is pasted at the END rung in overwrite mode, an error occurs and the message <Invalid execution point.> appears in the instructions area.

(NOTE 2) If a rung is deleted at the END rung, an error occurs and the message <Not possible to edit END instruction.> appears in the instructions area.

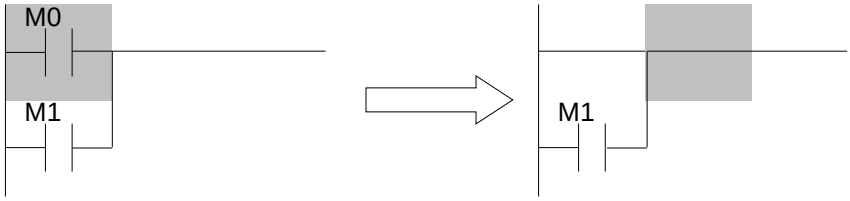
7.1.3.8 Delete symbol

To delete a symbol, move the cursor to the desired symbol and vertical line, and press **[DEL]** and <Delete vertical line>**.**

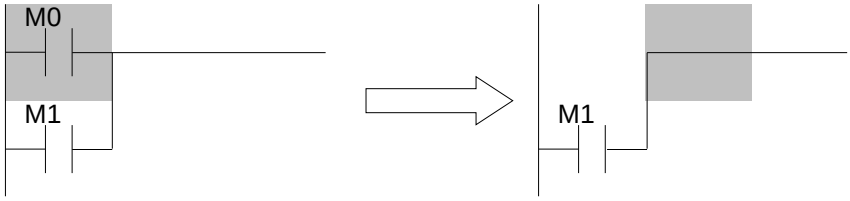
The operation behaves differently in insert and overwrite modes. When in insert mode, the ruled line remains (is not deleted). When in overwrite mode, everything is deleted.

Delete example in insert and overwrite modes:

Insert mode



Overwrite mode



(NOTE 1) The END rung and END command cannot be deleted. If a delete operation is executed, an error occurs and the message <Not possible to edit END instruction.> appears in the instructions area.
(NOTE 2) If a symbol is accidentally deleted while a contact/coil in the circuit is ON, that status will be maintained.

7.1.3.9 Undo edit operation

Press <Undo> to undo the edit operation.
Each time <Undo> is pressed, it undoes an edit operation. The undo operation can be performed a maximum of 5 times. However, after a conversion execution, the undo operation cannot restore to the status or condition prior to conversion.
The undo operation can be performed for the following:

- Symbol input
- Symbol deletion
- Rung operation
- Edit comment

7.1.3.10 Convert

Before writing a program, an unconverted circuit must always be converted into execution binary code.
The message <Converting.> appears in the instructions area during conversion. Once the conversion is completed, the message <Completed.> appears. At this time, the highlighted display changes color from gray to white.
The following items are checked in the conversion process.

- Disconnections
- Short-circuits
- Comparison of program size
- Double coil check

7

(NOTE 1) A ladder circuit that has a conversion error turns yellow.
(NOTE 2) After a conversion, the data cannot be restored back to its original state prior to conversion.

If <Write file after conversion> is set to <1:Yes>, the file is saved after conversion. The file is saved even when the alarm <<Double coil error>> and <<There is a control program outputting outside of the writing range>> are triggered.

7.1.3.11 Write during run

In an LD program, when the user presses <Convert>, the unconverted circuit is converted into binary code for execution and the file is saved. In addition, the user can also write the changed portion without stopping the control program during operation (during RUN).

<u>Conditions</u>

- <Write during run after conversion> is set to <1:Yes>
- When changes are made to an LD program that is registered as an execution program

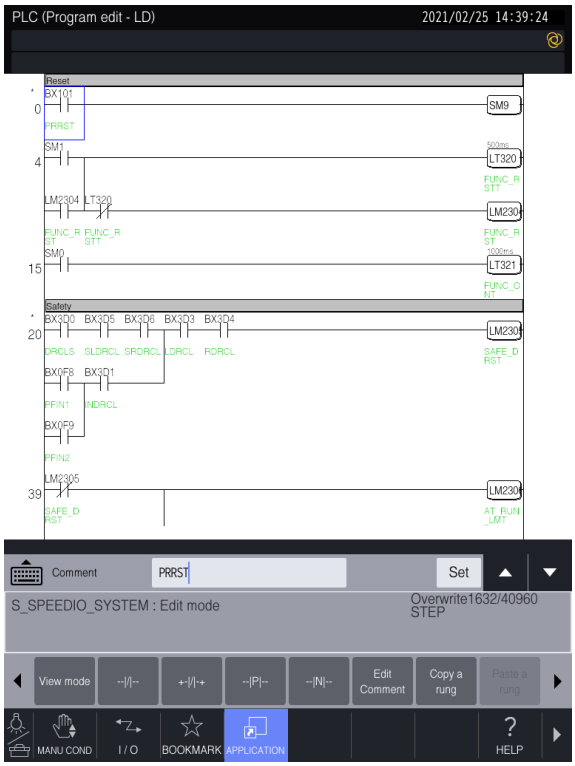
<u>Procedures</u>

1. Press **<**Convert**>** to execute the conversion.
2. When the conversion is executed, the message <Write during run?> appears.
   - Pressing <Yes> verifies or cross-checks the LD program that is running. If the program matches, it executes the write operation during RUN and returns to the edit screen. If the program does not match, the message <Write during RUN is not possible due to program mismatch.> appears, and it returns to the edit screen without executing the write operation during RUN.
   - Pressing <No> returns it to the edit screen without executing the write operation during RUN.

In addition, the file is saved even when the alarm <<Double coil error>> and <<There is a control program outputting outside of the writing range>> are triggered.

7.1.3.12 Edit comment

In edit mode, the user can edit an OM comment at the cursor position.
After the user enters a comment and presses **[ENT]**, the edits are reflected in the program.

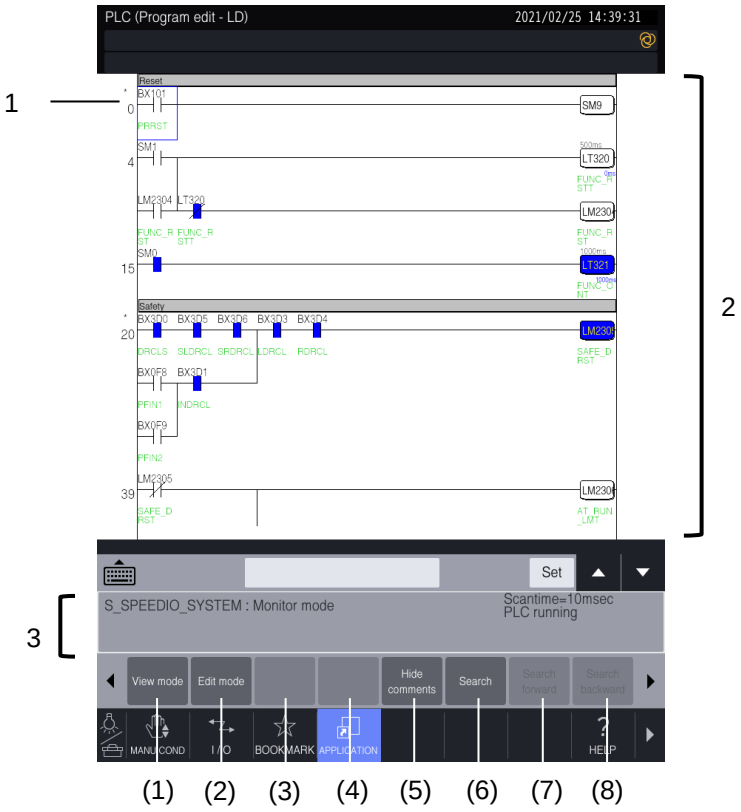


(NOTE 1) Only alphanumeric characters can be used and entered in this system.
(NOTE 2) Comments cannot be edited for an index register (IX).
(NOTE 3) If the comment size is exceeded when entering a comment, the message <Maximum program size will be exceeded.> appears in the instructions area and the input comment is discarded. To set a new comment again in this situation, re-enter the comment after deleting the comment at more than one location.

7.1.4 Monitor Mode

7.1.4.1 Screen layout

The following shows the screen layout for monitor mode.



Description of screen display

Position	Name	Description
1	Selection cursor	Displays the range for the current selection.
2	Program display area	Displays the operation status of the program. Refer to “7.1.4.2 Symbol display” for further details about the display. The user can change the display by moving the cursor or by using the <Previous page> key or the <Next page> key.
3	Instructions area	Displays items such as operation instructions or a program syntax error. In addition, the step number displays for the program in the <Used steps/Maximum steps>.

Function key display area

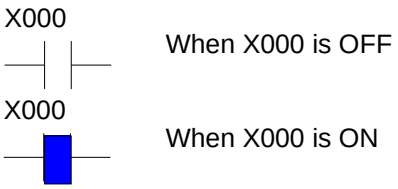
Column	Position	Label	Description
1	(1)	<View mode>	Changes to the program view mode.
	(2)	<Edit mode>	Changes to the program edit mode.
	(3)	-	-
	(4)	-	-
	(5)	<Show comments>	Shows or hides the comments. Refer to “7.1.2.8 Comment” for further details.
	(6)	<Search>	Searches a program. There are three types of searches: an OM, a step and a command search. Refer to “7.1.2.7 Search” for further details.
	(7)	<Search forward>	Searches the next ladder block after executing a search for an OM or command. Refer to “7.1.2.7 Search” for further details.
	(8)	<Search backward>	Searches the previous ladder block after executing a search for an OM or command. Refer to “7.1.2.7 Search” for further details.

Column	Position	Label	Description
2	(1)	<View mode>	Changes to the program view mode.
	(2)	-	-
	(3)	<SET/RESET>	Changes the bit OM value in the selection area. Refer to “7.1.4.3 Set / Reset” for further details.
	(4)	<Change current value>	Changes the word OM value in the selection area. Refer to “7.1.4.4 Change current value” for further details.
	(5)	<Clear local OM>	Clears all local OMs displayed except OMs for SM and SD. Refer to “7.1.4.6 Clear all local OMs” for further details.
	(6)	<Execute>	Executes built-in PLC operation. Refer to “7.1.4.5 Execute / Stop” for further details.
	(7)	<Stop>	Stops built-in PLC operation. Refer to “7.1.4.5 Execute / Stop” for further details.
	(8)	<Clear global OM>	Clears all global OMs displayed except OMs for X and Y. Refer to “7.1.4.7 Clear all global OMs” for further details.

7.1.4.2 Symbol display

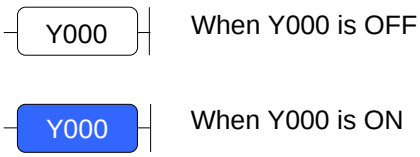
When in monitor mode, the built-in PLC reads the OM status in fixed cycles and displays that information on the screen. This section describes the display and what it means.

1. Contact
   The conductive status is shown by a space being colored in (blue).

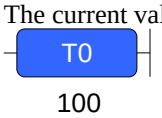


   It is conductive at contact *b* when X000 is OFF.

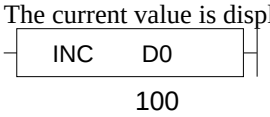
2. Coil
   The conductive status is shown by the space being colored.



   The current value is displayed at the same time under the symbol for the timer and counter.



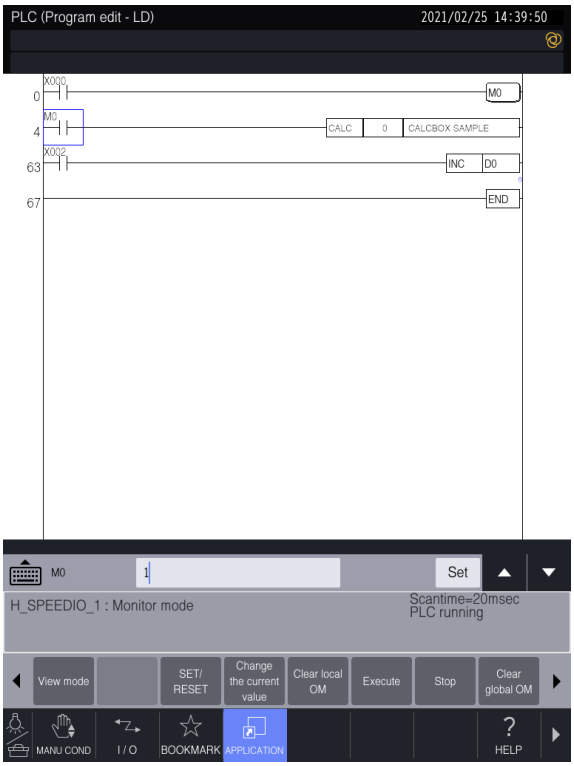
3. Data command / comparison command
   The current value is displayed under each OM.



4. Status display
   The scan time is displayed in the instructions area. In addition, the error code is displayed in the instructions area when an error occurs.

7.1.4.3 Set / Reset

This operation changes the bit OM signal. Move the cursor to the desired OM, press <SET/RESET> and enter “0” or “1”. “0” resets the OM, and “1” sets the value “1” to the OM. When the cursor is moved to the desired position, pressing **[ENT]** also performs the same operation as pressing <SET/RESET>.



(NOTICE) The SET/RESET operation turns ON the output for certain devices or parts which may suddenly start moving.
Take the necessary precautions when operating to prevent getting caught or hit by said devices or parts.

(NOTE 1) The computation results in the program are given priority during the run operation in the built-in PLC. Therefore, stop the built-in PLC when setting or resetting the output command (OUT command, etc.).

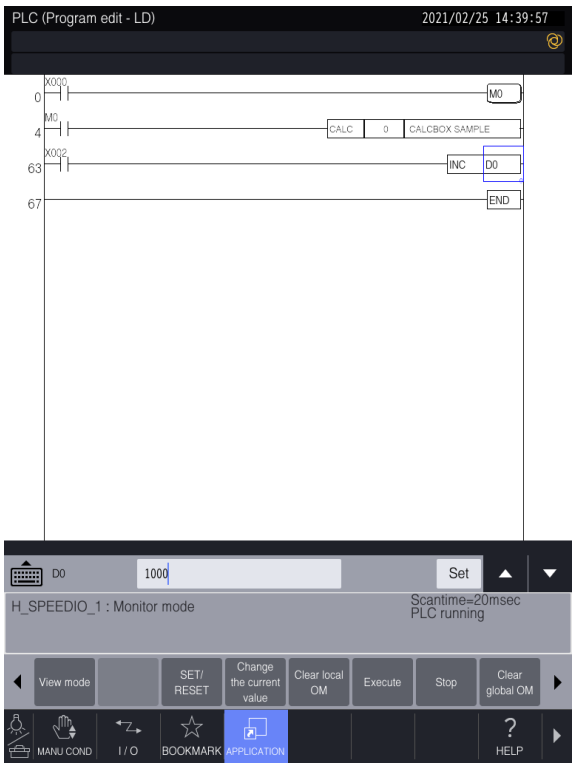
(NOTE 2) For the global OMs, the user can use SET and RESET to change all the points for Y/BY and change the value of the writing range for M.

(NOTE 3) The SET/RESET operation for global OMs does not check whether it falls outside of the power failure hold area or not.

(NOTE 4) An error message appears when the operation is performed for timers/counters, word OMs and OMs specified by multiple bit access.

7.1.4.4 Change current value

This operation changes the current value for a word OM. Move the cursor to the desired OM, press <Change current value> and enter the new value. When the cursor is moved to the desired position, pressing **[ENT]** also performs the same operation as pressing <Change current value>.



(NOTE 1) This operation applies to the current value for the timer / counter. The set value cannot be changed.

(NOTE 2) Bit OMs and bit OMs with multiple bit access cannot be changed. <Invalid OM. Select Word OM.> appears in the instructions area.

(NOTE 3) When changing the current value of a global OM, the user can change all the points for BDY and the values in the writing range for D.

(NOTE 4) When changing the current value for global OMs, it does not check whether it falls outside of the power failure hold area or not.

7.1.4.5 Execute / Stop

The user can execute or stop operation on the built-in PLC.
When <Execute> is pressed, the popup message <Start PLC operation?> is displayed.
<Yes>: Starts executing operation in the built-in PLC.
<No>: Cancels processing.

(NOTE) Operation cannot be started for the built-in PLC during an alarm that is related to the built-in PLC system, such as a PLC version mismatch or using an invalid control program.

When <Stop> is pressed, the popup message <Stop PLC operation?> is displayed.
<Yes>: Stops built-in PLC operation.
<No>: Cancels processing.

7.1.4.6 Clear all local OMs

The user can clear all local OMs in the program being displayed except OMs for SD and SM.
When <Clear all local OMs> is pressed, the popup message <Clear all local OMs?> is displayed.
<Yes>: Clears all the local OMs.
<No>: Cancels processing.

(NOTE) The clear operation for all local OMs is not available when a program or a debug process is running. The message <Stop program execution, and then re-execute.> appears in the instructions area.

7.1.4.7 Clear all global OMs

The user can clear all global OMs in the program being displayed except OMs for X and Y.
When <Clear all global OMs> is pressed, the popup message <Clear all global OMs?> is displayed.
<Yes>: Displays the following popup message.
<No>: Cancels processing.

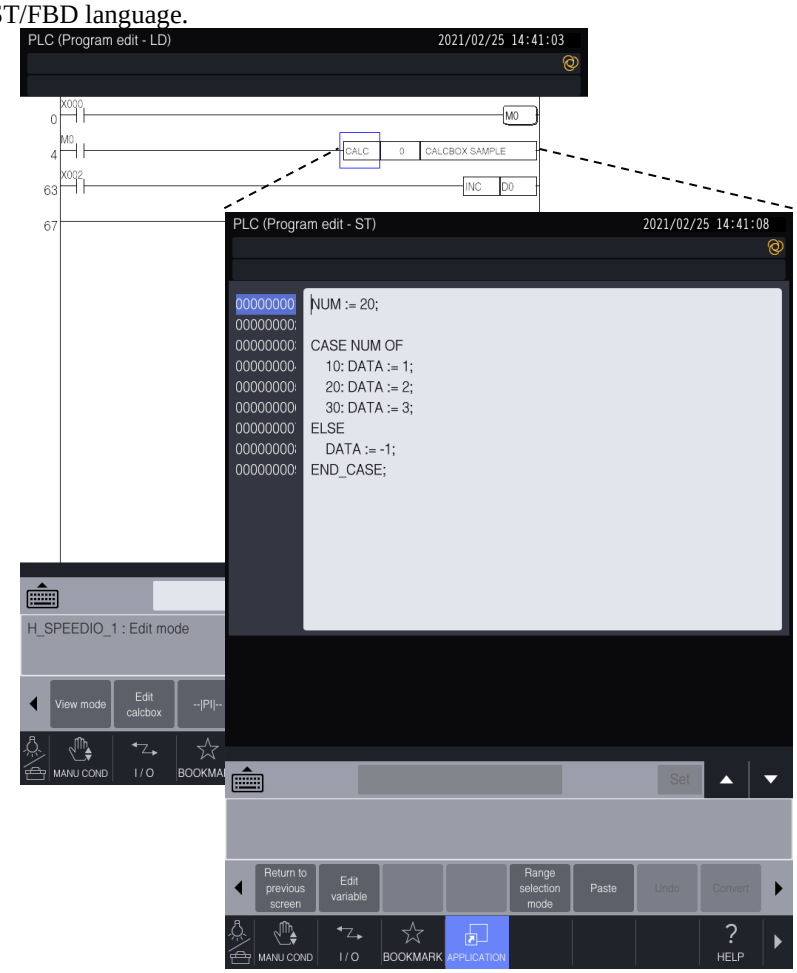
If <Yes> is selected above, the popup message <Clear the power failure hold area?>.

<Yes>: Clears all global OMs including the power failure hold area.
<No>: Clears all global OMs excluding the power failure hold area.

(NOTE) The clear operation for all global OMs is not available when a program or a debug process is running. The message <Stop program execution, and then re-execute.> appears in the instructions area.

7.1.5 Calcbox Edit Mode

The user can use calcbox to call a program (created in ST language) as an LD program function.
The aforementioned user defined function / system defined function can be called from calcbox in the ST/FBD language.



7.1.5.1 Create new calcbox

When <Calcbox edit> is selected without moving the cursor to a calcbox, the screen changes to the <PLC (Program edit - ST)> screen and the user can create a new calcbox.
When the screen is changed to the <PLC (Program edit - LD)> screen after editing, the edited calcbox is created at the cursor position before creating a new one.
An available number is automatically assigned by the system to the created a calcbox.

7.1.5.2 Edit calcbox

The user can edit an ST program for the selected calcbox by following the procedure below.

1. Move the cursor to the desired calcbox.
2. Select **<**Edit Calcbox**>**.
3. Change the screen to the <PLC (Program edit - ST)> screen. The ST program can be edited.
4. When the <PLC (Program edit - LD)> screen is opened after editing, the popup message <Apply the edits?> is displayed.
   <Yes>: Carries out conversion. After the conversion is complete, the mode changes to edit mode.
   <No>: Discards the edited content and changes to edit mode.

7.1.5.3 Delete calcbox

The user can delete the selected calcbox by following the procedure below.

1. Move the cursor to the desired calcbox.
2. When **[DEL]** is selected, the popup message <Delete selected calcbox?> is displayed.
   <Yes>: Deletes the selected calcbox.
   <No>: Cancels the deletion.

(This page was intentionally left blank.)

CHAPTER 8

PROGRAMMING FUNCTION ST

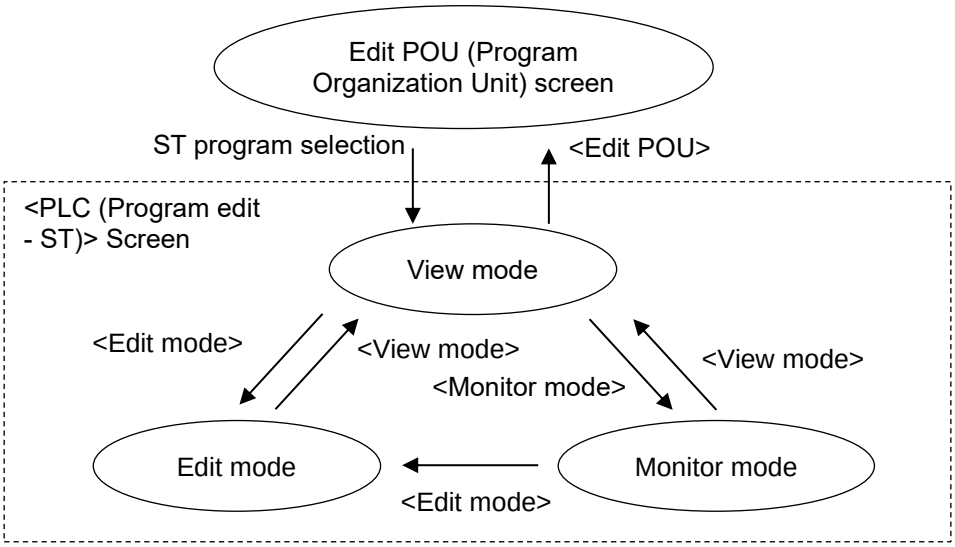
8.1 <PLC (Program edit - ST)> Screen
8.2 Edit Variables

8

8.1 <PLC (Program edit - ST)> Screen

8.1.1 Overview

On this screen, the user can edit and monitor programs where the <Language> is set to <ST> for high speed and standard control programs. The <ST> refers to an ST program.
The user can open the <PLC (Program edit - ST)> screen by selecting the aforementioned program on the <PLC (Edit POU (Program Organization Unit)> screen. The <PLC (Program edit - ST)> screen has three modes available: view, edit and monitor modes. The mode changes to view mode immediately after changing screens. Each mode can be switched using the function key.



(NOTE) When editing and writing an ST program, setting parameters, setting the power failure retention / write range or setting the conversion mode, be sure to set the user parameter <PLC program protection> to <0: Invalid>.
However, when editing and writing for high speed and standard control program 10 (assigned to standard task 3 and high speed task 2), make sure <Parameter change> on the I/O (extension I/O maintenance) screen is set to <1: Writable>. Then, set <PLC program protection> to <0: Invalid>.

8

8.1.2 View Mode

8.1.2.1 Screen layout

The following shows the screen layout for view mode.

Description of screen display

Position	Name	Description
1	Program display area	Displays the program. The user can change the display by moving the cursor or by using the <Previous page> key or the <Next page> key.
2	Instructions area	Displays the step number for the program in the <Used steps/Maximum steps>.

Function key display area

Column	Position	Label	Description
1	(1)	<Edit POU>	Opens the <PLC (Edit POU (Program Organization Unit))> screen.
	(2)	<Edit mode>	Changes to the program edit mode.
	(3)	<Monitor mode>	Changes to the program monitor mode. This grays out when the displayed program is not an execution program.
	(4)	<Write program>	Writes the program into the built-in PLC. However, this key grays out when the execution program is not registered. Refer to “8.1.2.2 Write program” for further details.
	(5)	<Edit variable>	Opens the <PLC (Edit variable)> screen. Refer to “8.2 Edit variable” for further details.
	(6)	<Find character string>	Searches for a character string in the program. Refer to “8.1.2.7 Find character string” for further details.
	(7)	<Search forward>	Searches downward and goes to the next character string found in the search. Refer to “8.1.2.7 Find character string” for further details.

Column	Position	Label	Description
	(8)	<Search backward>	Searches upward and goes to the next character string found in the search. Refer to “8.1.2.7 Find character string” for further details.
2	(1)	<Edit POU>	Changes to the <PLC (Edit POU (Program Organization Unit))> screen.
	(2)	-	-
	(3)	-	-
	(4)	<New program>	Discards the edited program and resets the content to create a new program. Refer to “8.1.2.5 New program” for further details.
	(5)	<Save>	Saves the program that is being edited. Refer to “8.1.2.4 Save” for further details.
	(6)	<Read program>	The user can read and re-edit the program (before editing). Refer to “8.1.2.3 Read program” for further details.
	(7)	<Cancel new program>	Cancels the **<**New program **>** operation and returns the program content back to previous state. After the new program is created, this key is valid until the program is saved with the <Save> key. This grays out in all other situations. Refer to “8.1.2.6 Cancel new program” for further details.
	(8)	<Parameter setting>	Displays the <PLC (Parameter setting)> screen.

8.1.2.2 Write program

A program registered as an execution program is written to the built-in PLC. The writing operation cannot be executed for other programs that are not execution programs. The procedure is shown below.

8

1. Press **<**Write program**>**.
2. The popup message <Write during run operation?> is displayed. Select **<**Yes**>** to proceed with the operation. Press **<**No**>** to cancel the operation.
3. This executes and writes the program. During the writing operation, the message <Currently writing> appears in the instructions area. After the writing operation is completed, the message <Completed.> appears.
4. When the above operation is carried out while the program is executing, the PLC automatically starts up after the writing operation.

(NOTE 1) The alarm <<Control program stopped (****)>> is triggered immediately after the program write operation is executed. “****” refers to high speed tasks 1, 2 or standard tasks 1, 2 or 3.

(NOTE 2) Do not execute the program write operation while the machine is operating.

(NOTE 3) If the program write operation is executed while the control program is stopped, then the control program stays stopped after the writing operation.

(NOTE 4) If an OM (Y, BY) with the same output exists between programs, the alarm <<Double coil error>> is triggered.

(NOTE 5) When writing a control program with an output command to a global OM (M, D) that is not in the writing range, the writing operation itself is executed. However, the alarm <<There is a control program outputting outside of the writing range>> is triggered.

8.1.2.3 Read program

The user can discard the edited program, read the content of the program (prior to editing) and then re-edit it.
The operation procedure is shown below.

1. Press <Read program>.
2. Only when a program is being edited and there is a difference with the program being read, the popup message <<Program being edited will be removed. OK?>> is displayed.
   Select <Yes> to proceed with the operation. Select <No> to cancel it.
3. The program read operation is executed. When the read operation is complete, the message <Completed> appears. To edit the program after the reading operation, switch to edit mode.

(NOTE) It may take time for the program reading operation to be completed. The processing time takes longer when there are many steps.

8.1.2.4 Save

A program being edited is saved as a file. The user can call a saved program by reading the program.

8.1.2.5 New program

The user can discard the edited program, and re-edit the program again starting from a blank program platform.
The procedure is shown below.

1. Press the <New program> key.
2. The popup message <Program being edited will be removed. OK?> is displayed. Select <Yes> to proceed with the operation. Select <No> to cancel the operation.
3. A new program is created. To edit the program, switch to edit mode.

(NOTE 1) After a new program is created, the <Undo> key will not change the new program content back to the previous status (program content prior to using the new program key).

(NOTE 2) After a new program is created, any data that was copied or edited (before creating a new program) is cleared in the program.

8.1.2.6 Cancel new program

When the **<**New program**>** key is used, the user can cancel the operation and restore the program content back to the previous status (program content prior to using the new program key). The procedure is shown below.

1. Press <Cancel new program>.
2. The popup message <Discard new program?> is displayed. Select <Yes> to proceed with the operation. Select <No> to cancel the operation.
3. The program content is restored to its previous status (program content prior to using the new program key). To edit the program, switch to edit mode.

(NOTE) Once **<**Save**>** is used to save the program content after creating a new program, the user will not be able to restore the program content back to the previous status (program content prior to using the new program key).

8.1.2.7 Find character string

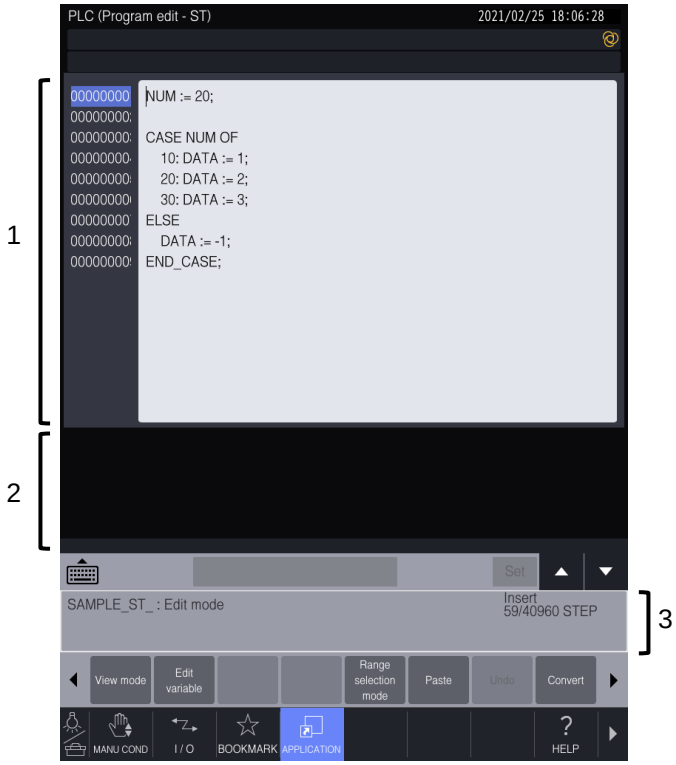
When using the ST language, the user can search for a character string inside the program in all modes.
After pressing **<**Find character string**>**, enter the character string into the data input field in order to specify the search. Once the search is executed, it starts searching from the beginning of the program and the cursor moves to the corresponding program line.
After pressing **<**Search backward**>** or **<**Search forward**>**, the search is carried out going forward or going backward in the program using the cursor position as a reference point.

(NOTE) If the corresponding character string cannot be found, then the message <Corresponding program is missing.> appears.

8.1.3 Edit Mode

8.1.3.1 Screen layout

The following shows the screen layout for edit mode.



Description of screen display

Position	Name	Description
1	Program editing area	The program can be edited. The user can change the display by moving the cursor or by using the <Previous page> key or the <Next page> key.
2	Conversion error display area	Displays any errors triggered during the program conversion due to a syntax error in the program. The user can swipe vertically to change the display.
3	Message display area	Displays the step number for the program in the <Used steps/Maximum steps>. In addition, the edit mode is displayed either in overwrite mode or insert mode.

Function key display area

Column	Position	Label	Description
1	(1)	<View mode>	Changes to the program view mode.
	(2)	<Edit variable>	Opens the <PLC (Edit variable)> screen. Refer to “8.2 Edit variable” for further details.
	(3)	-	-
	(4)	-	-
	(5)	<Range selection mode>	Changes to the range selection mode.
	(6)	<Paste>	Pastes the text copied in range select mode.
	(7)	<Undo>	Cancels the program edit operation and returns it back to the previous state. The undo operation can be performed a maximum of 5 times. After a conversion, the undo operation cannot be used.
	(8)	<Convert>	Converts a program being edited to binary. Refer to “8.1.3.7 Conversion” for further details.

Column	Position	Label	Description
2	(1)	<View mode>	Changes to the program view mode.
	(2)	<Replace>	Replaces the specified character string in the program with a new character string.
	(3)	<Find character string>	Searches for a character string in the program. Refer to “8.1.2.7 Find character string” for further details.
	(4)	<Search forward>	Searches downward and goes to the next character string found in the search. Refer to “8.1.2.7 Find character string” for further details.
	(5)	<Search backward>	Searches upward and goes to the next character string found in the search. Refer to “8.1.2.7 Find character string” for further details.
	(6)	-	-
	(7)	-	-
	(8)	-	-

The user can use the virtual keyboard on the edit screen in the ST language to enter and insert characters.

QWERTY keyboard layout

The user can scroll through the symbols at the top of the virtual keyboard, and any available symbols can be inserted.

(Scrolling through symbols)

NC keyboard layout

In the symbol display, any available symbols can be inserted. The alphanumeric key display is the same as other screens.

8.1.3.2 Multiple bit access for bit OM

Not only can a singular bit memory be accessed but a series of multiple bit memories can also be accessed.

A “W” and “L” are respectively inserted after the bit OM number for 16-/32-bit access.

Input example	Description
M16W	Accesses a 16-bit segment from M16 to M31.
M64L	Accesses a 32-bit segment from M64 to M95.

“:n” (n is the bit number) is inserted after the bit OM in the bit access. It is possible to specify a maximum of 32 in 4-bit units for N.

Input example	Description
M4:4	Accesses a 4-bit segment from M4 to M7.
M32:20	Accesses a 20-bit segment from M32 to M51.

- Set the OM number when using multiple bit access in multiples of 4.
- When the OM is outside the range due to multiple bit access, an input error occurs.
- When specifying a 32-bit value for the constant, L is inserted at the tail end.
  Ex: When a constant value of 40000 is entered, 40000L is shown.

8.1.3.3 32-bit access for word OM

The memory range for a word OM is not 16-bit but can use or handle 32-bit.

An “L” is inserted after the word OM number.

Input example	Description
D16L	Accesses a 32-bit segment for D16 and D17.

- When the OM is outside the range due to the 32-bit access to the word OM, an input error occurs.
- There is no start number restriction when the OM number is within range for 32-bit access in a word OM.
- Insert an “L” at the tail end of the numerical value when performing 32-bit changes to the current value for the 32-bit access of a word OM.
  Ex: “32768L”

8

8.1.3.4 Specify OM indirectly

By using the index register (IX) to indirectly set an OM (modifying index), the user can specify an OM that includes a value stored in IX for OM numbers that can be modified.

Ex: When 10 is stored to IX0
M1IX0 can be used to express M11.

(NOTICE) Set the IX value so that the OM number does not exceed the maximum value based on index addressing. If the OM number exceeds the maximum value, an out of range OM is specified and another type of OM may be read and written. Be careful when specifying the number, otherwise this will cause the program to malfunction.

8.1.3.5 Undo edit operation

Press <Undo> to undo the edit operation.
Each time <Undo> is pressed, it undoes the previous edit operation. The undo operation can be performed a maximum of 5 times. However, after a conversion execution, the undo operation cannot restore to the status or condition prior to conversion.
The undo operation can be performed for the following:

- Write character string
- Replace operation

8.1.3.6 Write during RUN

In a control program, when the user presses <Convert>, the unconverted circuit is converted into binary code for execution and the file is saved. In addition, the user can also write the changed portion without stopping the program during operation (during RUN).

Conditions

- <Write during run after conversion> is set to <1: Yes> in the conversion mode setting.
- When making changes to the same control program that is operating.

Procedure

1. Press the <Convert> key to execute the conversion.
2. The popup message <Write during run?> is displayed.

    - Pressing <Yes> verifies or cross-checks the control program that is running. If the program matches, it executes the write operation during RUN and returns to the edit screen. If the program does not match, the message <Write during RUN is not possible due to program mismatch.> appears, and it returns to the edit screen without executing the write operation during RUN.
    - Pressing <No> returns it to the edit screen without executing the write operation during RUN.

In addition, the file is saved even when the alarm <<Double coil error>> and <<There is a control program outputting outside of the writing range>> are triggered.

8.1.3.7 Conversion

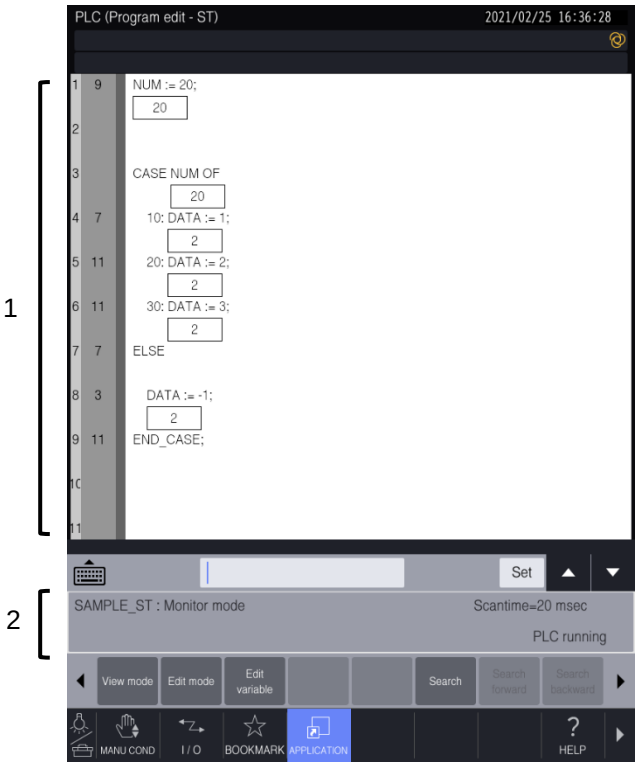
The user needs to carry out a conversion operation in order to run an edited ST program as an execution program in the built-in PLC. The checks in the table below are carried out during the conversion operation. If the check results are OK, then the conversion operation is completed. In addition, the message <Converting> appears during conversion. Once the conversion is completed, the message <Conversion is complete.> appears.

Item	Description
Statement check	Checks whether the statement complies with the language specification.
Variable type check	Checks whether the variable types match in the computation for the program processing.
Program size check	Checks whether the program size exceeds the maximum number of steps after the conversion operation.

8.1.4 Monitor Mode

8.1.4.1 Screen layout

The following shows the screen layout for monitor mode.



Description of screen display

Position	Name	Description
1	Program editing area	Displays the program. The user can change the display by moving the cursor or by using the <Previous page> key or the <Next page> key.
2	Instructions area	Displays the scan time and the status of the PLC.

Function key display area

Column	Position	Label	Description
1	(1)	<View mode>	Changes to the program view mode.
	(2)	<Edit mode>	Changes to the program edit mode.
	(3)	<Edit variable>	Opens the <PLC (Edit variable)> screen. Refer to “8.2 Edit variable” for further details.
	(4)	-	-
	(5)	-	-
	(6)	<Find character string>	Searches a program. There are three types of searches: an OM, a step and a command search. Refer to “8.1.2.7 Find character string” for further details.
	(7)	<Search forward>	Searches downward and goes to the next character string found in the search. Refer to “8.1.2.7 Find character string” for further details.
	(8)	<Search backward>	Searches upward and goes to the next character string found in the search. Refer to “8.1.2.7 Find character string” for further details.
2	(1)	<View mode>	Changes to the program view mode.
	(2)	-	-
	(3)	-	-
	(4)	<Change the current value>	Changes the current value for the variable in the program display area. Refer to “8.1.4.2 Change current value” for further details.

Column	Position	Label	Description
	(5)	<Clear all local Oms>	Clears all local OMs displayed except OMs for SM and SD. Refer to “8.1.4.4 Clear all local OMs” for further details.
	(6)	<Execute>	Starts operation for the built-in PLC. Refer to “8.1.4.3 Execute / Stop” for further details.
	(7)	<Stop>	Stops operation for the built-in PLC. Refer to “8.1.4.3 Execute / Stop” for further details.
	(8)	<Clear all global Oms>	Clears all global OMs displayed except OMs for X and Y. Refer to “8.1.4.5 Clear all global OMs” for further details.

8.1.4.2 Change Current Value

The user can change the current value for the variable.

1. Press the <Change current value> key.
2. Enter the desired OM and value.
   Ex: D10 100
3. Press the **[ENT]** key.

8.1.4.3 Execute / Stop

The user can start and stop operation for the built-in PLC. The user can only use these operations for a program registered as an execution program.
When <Execute> is pressed, the popup message <Start PLC operation?> is displayed.
<Yes> : Starts operation for the built-in PLC.
<No> : Cancels the process.

(NOTE) Operation cannot be started for the built-in PLC during an alarm that is related to the built-in PLC system, such as a PLC version mismatch or using an invalid control program.

When <Stop> is pressed, the popup message <Stop PLC operation?> is displayed.
<Yes> : Stops the built-in PLC.
<No> : Cancels the process.

8.1.4.4 Clear all local OMs

The user can clear all local OMs in the program being monitored except OMs for SM and SD.
When <Clear all local OMs> is pressed, the popup message <Clear all local OMs?> is displayed.
<Yes> : Clears all the local OMs.
<No> : Cancels the process.

(NOTE) The clear operation for all local OMs is not available when a program or a debug process is running. The message <Stop the built-in PLC, and then re-execute.> appears in the instructions area.

8.1.4.5 Clear all global OMs

The user can clear all global OMs in the program being monitored except OMs for X and Y.
When <Clear all global OMs> is pressed, the popup message <Clear all global OMs?> is displayed.
<Yes> : Displays the popup message <Clear the power failure hold area?>.
<No> : Cancels the process.

<Yes> : Clears all global OMs including the power failure hold area.
<No> : Clears all global OMs excluding the power failure hold area.

(NOTE) The clear operation for all global OMs is not available when all programs or a debug process is running. The message <Stop the built-in PLC, and then re-execute.> appears in the instructions area.

8.2 Edit Variables

When in edit mode on the <PLC (Edit variable)> screen, the user can define available variables in an ST program. When in display / monitor mode, the user can reference the variables.
Refer to “Chapter 12 Commands (ST)” for further information on the data types and their default values.

Description of screen display

Position	Name	Description
1	Scope	The user can set the valid scope of the variable.
2	Variable name	The user can enter the variable name. The user can use alphanumeric characters (does not distinguish between uppercase and lowercase) and an “_” (underscore) symbol. However, a number cannot be used as the beginning of the variable name. When the scope is <VAR_EXTERNAL>, the user can select only from variables that are defined on the <PLC (Global variable settings)> screen.
3	Data type	The user can select and set the data type. The type cannot be changed when the variable is being used in the program and when the scope is <VAR_EXTERNAL>.
4	Element number	The user can set the element number of the variable. When the element number is 0, it is a single variable. When the element number is not 0, it is an array variable. The number cannot be changed when the variable is being used in the program and when the scope is <VAR_EXTERNAL>.
5	Default value	The user can set the initial value right after the power is turned ON and the program is written. The value cannot be changed when the scope is <VAR_EXTERNAL>.
6	Comment	The user can enter a comment for a variable. The comment cannot be changed when the scope is <VAR_EXTERNAL>.

Function key display area (view mode)

Column	Position	Label	Description
1	(1)	Return to previous screen	Changes to the program view mode.
	(2)		
	(3)		
	(4)		
	(5)		
	(6)		
	(7)		
	(8)		

Function key display area (in edit mode)

Column	Position	Label	Description
1	(1)	Return to previous screen	Changes to the program view mode.
	(2)	Copy variable	Records the variable selected in the variable edit area. Refer to “8.2.2 Copy / Paste” for further details.
	(3)	Paste variable	Pastes the variable that is recorded with <Copy variable>. Refer to “8.2.2 Copy / Paste” for further details.
	(4)	Insert variable	Creates and inserts a new variable at the selected line inside the variable edit area. Refer to “8.2.1 New variable” for further details.
	(5)	Insert external variable	Creates and inserts a new external variable at the selected line inside the variable edit area. Refer to “8.2.1 New variable” for further details.
	(6)	Delete variable	Deletes the selected variable inside the variable edit area. Refer to “8.2.3 Delete variable” for further details.
	(7)	Undo	Cancels the program edit operation and returns it back to the previous state. The undo operation can be performed a maximum of 5 times. After a conversion, the undo operation cannot be used.

Function key display area (monitor mode)

Column	Position	Label	Description
1	(1)	Return to previous screen	Changes to the program monitor mode.
	(2)		
	(3)		
	(4)		
	(5)		
	(6)		
	(7)		
	(8)	Change current value	Changes the current value for the variable in the program display area. Refer to “8.1.4.2 Change current value” for further details.

8.2.1 New variable

The following shows the procedure for creating a variable.

1. Press **<**Insert variable**>** to create and insert a new variable at the selected line.
2. Set the scope for the variable that was created.
3. After the variable name is automatically assigned, the user can edit it to any given name.
4. Edit the data type, and set the default value and/or comment when necessary.
   If the user selects <VAR_EXTERNAL> for the scope in step 2, this operation is not required.

The maximum number of variables in one POU is as follows.
BOOL type : 1024
Non-BOOL type : 1024

(NOTE 1) When non-BOOL type variables are created, the number of variables that can be used varies according to the data type.
Refer to “6.3.1. Available data capacity” for further details.

(NOTE 2) When creating arrays, the total element number for each array that can be created cannot exceed the maximum number of BOOL type or non-BOOL type variables.

(NOTE 3) When a variable is executed and the maximum number of BOOL type variables is exceeded, the message <Data size used in the BOOL type parameter variables exceeds the limit: 1024> appears in the conversion error display area. When the maximum number of non-BOOL type variables is exceeded as well, the message <Data size used in the non-BOOL type parameter variables exceeds the limit: 1024> appears in the conversion error display area.

8.2.2 Copy & Paste

The user can copy and paste a variable to duplicate it. The procedure is noted below.

1. Press <Copy variable> to copy the selected variable.
2. Move the cursor to a given position, and then press <Paste variable>. The variable copied in step 1 is inserted.

8.2.3 Delete Variable

The user can press <Delete variable> to delete the selected variable by line.

CHAPTER 9

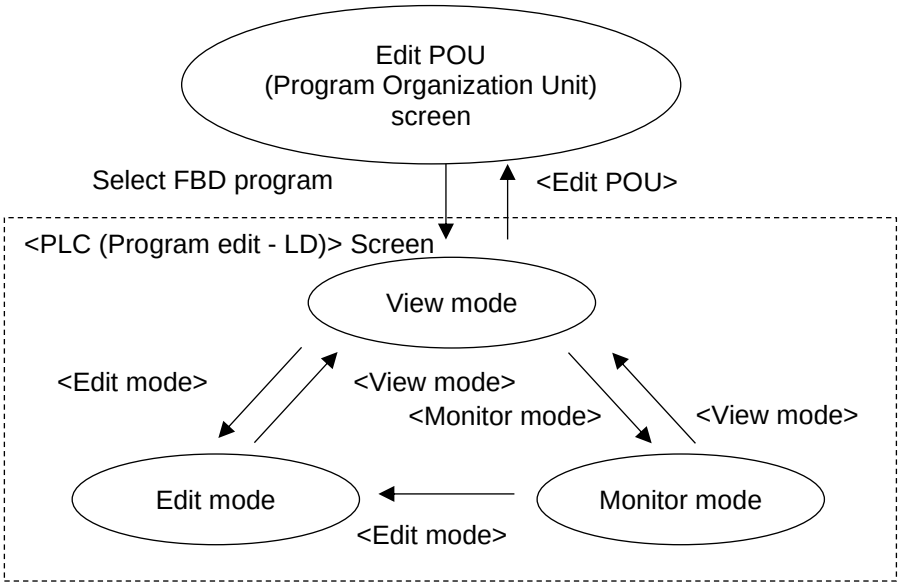
PROGRAMMING FUNCTION FBD

9.1 <PLC (Program edit - FBD)> Screen
9.2 Edit Variables

9.1 <PLC (Program edit - FBD)> Screen

9.1.1 Overview

On this screen, the user can edit and monitor programs where the <Language> is set to <FBD> for high speed and standard control programs. The <FBD> refers to an FBD program. The user can open the <PLC (Program edit -FBD)> screen by selecting the aforementioned program on the <PLC (Edit POU (Program Organization Unit)> screen.
The <PLC (Program edit - FBD)> screen has three modes available: view, edit and monitor modes. The mode changes to view mode immediately after changing screens. Each mode can be switched using the function key.

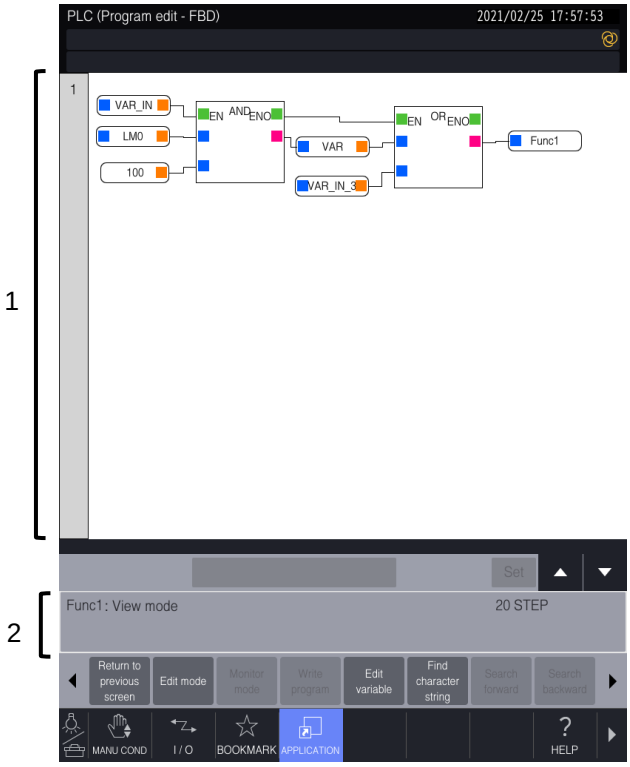


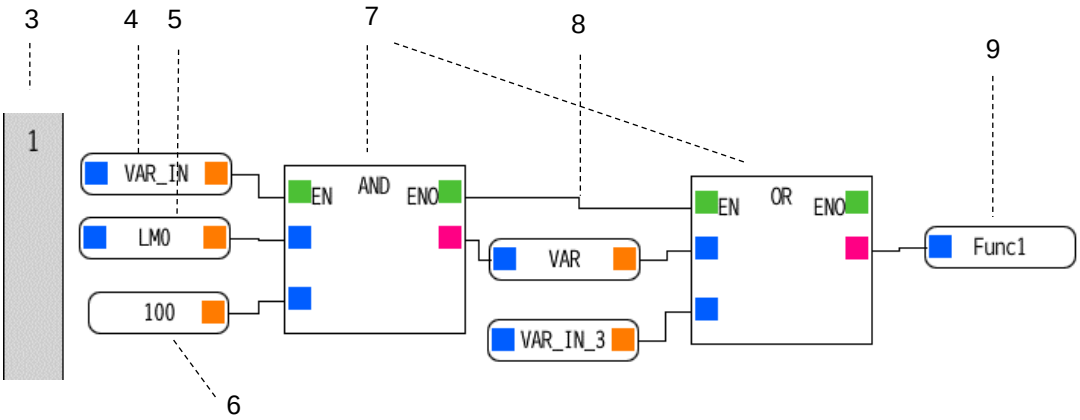
(NOTE) When editing and writing an FBD program, setting parameters, setting the power failure hold / writing range or setting the conversion mode, be sure to set the user parameter <PLC program protection> to <0: Invalid>.
However, when editing and writing for high speed and standard control program 10 (assigned to standard task 3 and high speed task 2), make sure <Parameter change> on the I/O (extension I/O maintenance) screen is set to <1: Writable>. Then, set <PLC program protection> to <0: Invalid>.

9.1.2 View mode

9.1.2.1 Screen layout

The following shows the screen layout for view mode.





Description of screen display

Position	Name	Description
1	Program display area	Displays the program. The user can change the display by moving the cursor or by changing the page.
2	Instructions area	Displays the step number for the program in the <Used steps/Maximum steps>.
3	Network	This provides an overview of 1 process that connects a given number of blocks with connectors. Execution is carried out in order starting from the smallest network number. Two independent processes cannot be created that are not connected to a connector in a network.
4	Variable block	This component assigns variables defined on the <PLC (Edit variable)> screen.
5	OM block	This component assigns OMs.
6	Constant block	This component assigns constants.
7	Function block	This component assigns functions. When the user taps on a user defined function, the function edit screen is displayed.
8	Connector	This component connected blocks.
9	Return value block	This component assigns the return value for a function.

Function key display area

Column	Position	Label	Description
1	(1)	<Edit POU>	Opens the <PLC (Edit POU (Program Organization Unit))> screen.
	(2)	<Edit mode>	Changes to the program edit mode.
	(3)	<Monitor mode>	Changes to the program monitor mode. This grays out when the displayed program is not an execution program.
	(4)	<Write program>	Writes the program into the built-in PLC. However, this key grays out when the execution program is not registered. Refer to “9.1.2.2 Write program” for further details.
	(5)	<Edit variable>	Opens the <PLC (Edit variable)> screen. Refer to “9.2 Edit variables” for further details.
	(6)	<Find character string>	Searches for a character string in the program. Refer to “9.1.2.5 Search” for further details.
	(7)	<Search forward>	Searches downward and goes to the next character string found in the search. Refer to “9.1.2.5 Search” for further details.
	(8)	<Search backward>	Searches upward and goes to the next character string found in the search. Refer to “9.1.2.5 Search” for further details.
2	(1)	<Edit POU >	Changes to the <PLC (Edit POU (Program Organization Unit))> screen.
	(2)	<Change network>	Displays the network for the next network number. If the displayed network is the last one, then the first network is displayed here.
	(3)	<Network jump>	Press this key, and enter a network number in the data input field. After pressing the **[ENT]** key, it jumps to the specified network.
	(4)	-	-
	(5)	<Save>	Saves the program that is being edited. Refer to “9.1.2.4 Save” for further details.
	(6)	<Read program>	The user can read and re-edit the program (before editing). Refer to “9.1.2.3 Read program” for further details.
	(7)	-	-
	(8)	<Parameter setting>	Displays the <PLC (Parameter setting)> screen.

9.1.2.2 Write program

A program registered as an execution program is written to the built-in PLC. The writing operation cannot be executed for other programs that are not execution programs. The procedure is shown below.

1. Press **<**Write program**>**.
2. The popup message <Program will be written during RUN. Proceed?> is displayed.
   Select **<**Yes**>** to proceed with the operation. Select **<**No**>** to cancel the operation.
3. This executes and writes the program. During the writing operation, the message <Currently writing> appears in the instructions area. After the writing operation is completed, the message <Completed.> appears.
4. When the above operation is carried out while the program is executing, the PLC automatically starts up after the writing operation.

(NOTE 1) The alarm <<Control program stopped (****)>> is triggered immediately after the program write operation is executed. "****" refers to high speed tasks 1, 2 or standard tasks 1, 2 or 3

(NOTE 2) Do not execute the program write operation while the machine is operating.

(NOTE 3) If the program write operation is executed while the control program is stopped, then the control program stays stopped after the writing operation.

(NOTE 4) If an OM (Y, BY) with the same output exists between programs, the alarm <<Double coil error>> is triggered.

(NOTE 5) When writing a control program with an output command to a global OM (M, D) that is not in the writing range, the writing operation itself is executed. However, the alarm <<There is a control program outputting outside of the writing range>> is triggered.

9.1.2.3 Read program

The user can discard the edited program, read the content of the program (prior to editing) and then re-edit it.
The procedure is shown below.

1. Press **<**Read program**>**.
2. Only when a program is being edited and there is a difference with the program being read, the popup message <<Program being edited will be removed. OK?>> is displayed.
   Select **<**Yes**>** to proceed with the operation. Select **<**No**>** to cancel it.
3. The program read operation is executed. When the read operation is complete, the message <Completed> appears. To edit the program after the reading operation, switch to edit mode.

(NOTE) It may take time for the program reading operation to be completed. The processing time takes longer when there are many steps.

9.1.2.4 Save

A program being edited is saved as a file. The user can call a saved program by reading the program.

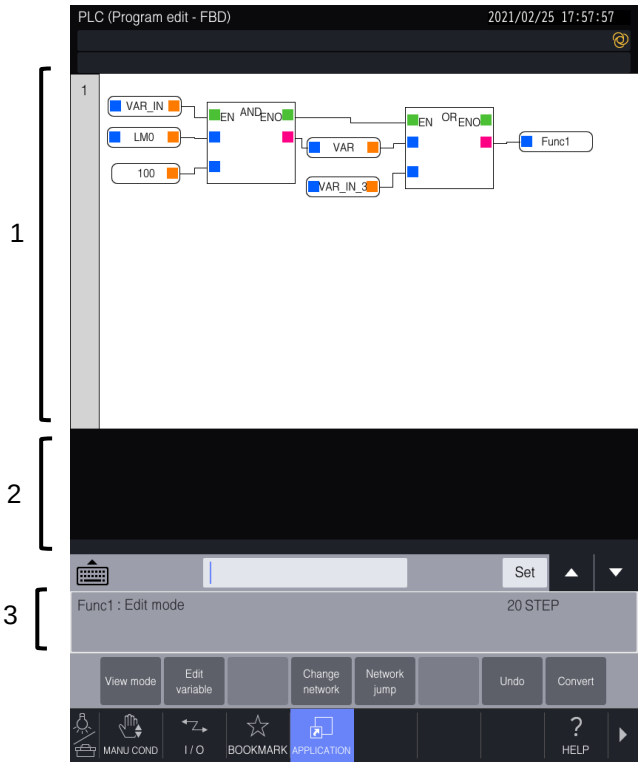
9.1.2.5 Search

When using the FBD language, the user can search for a character string inside the program in the view or monitor modes.
After pressing **<**Find character string**>**, enter the character string into the data input field in order to specify the search. Once the search is executed, it starts searching from the beginning of the program and the cursor moves to the corresponding OM / variable box.
After pressing **<**Search backward**>** or <Search forward**>**, the search is carried out going forward or going backward in the program using the cursor position as a reference point.

9.1.3 Edit Mode

9.1.3.1 Screen layout

The following shows the screen layout for edit mode.



Description of screen display

Position	Name	Description
1	Program editing area	The program can be edited. The user can change the display by moving the cursor or by swiping horizontally.
2	Conversion error display area	Displays any errors triggered during the program conversion due to a syntax error in the program. The user can swipe vertically to change the display.
3	Message display area	Displays the step number for the program in the <Used steps/Maximum steps>. In addition, the edit mode is displayed either in overwrite mode or insert mode.

Function key display area

Column	Position	Label	Description
1	(1)	<View mode>	Changes to the program view mode.
	(2)	<Edit variable>	Opens the <PLC (Edit variable)> screen. Refer to “9.2 Edit variables” for further details.
	(3)		
	(4)	<Change network>	Displays the network for the next network number. If the displayed network is the last one, then the first network is displayed here.
	(5)	<Network jump>	Press this key, and enter a network number in the data input field. After pressing the **[ENT]** key, it jumps to the specified network.
	(6)	-	-
	(7)	<Undo>	Cancels the program edit operation and returns it back to the previous state. The undo operation can be performed a maximum of 5 times. After a conversion, the undo operation cannot be used.
	(8)	<Convert>	Converts a program being edited to binary. Refer to “9.1.3.8 Convert” for further details.

9.1.3.2 Program edit

The user can tap on a block in the program and edit the OM, constant or variable block, or the user defined function.

When editing an OM, constant or variable block

1. Tap on the desired block.
2. Enter the OM name, constant or variable name, and then press the **[ENT]** key.

When editing a user defined function

1. When the user taps on the desired function, the function edit screen is displayed.
2. After editing, press <Return to previous screen**>** to go back to the display screen for the source program.

(NOTE) Blocks and connectors can only be added or deleted from the PLC programming software.

9.1.3.3 Multiple bit access for bit OM

Not only can a singular bit memory be accessed but a series of multiple bit memories can also be accessed.

A “W” and “L” are respectively inserted after the bit OM number for 16-/32-bit access.

Input example	Description
M16W	Accesses a 16-bit segment from M16 to M31.
M64L	Accesses a 32-bit segment from M64 to M95.

“:n” (n is the bit number) is inserted after the bit OM in the bit access. It is possible to specify a maximum of 32 in 4-bit units for n.

Input example	Description
M4:4	Accesses a 4-bit segment from M4 to M7.
M32:20	Accesses a 20-bit segment from M32 to M51.

- Set the OM number when using multiple bit access in multiples of 4.
- When the OM is outside the range due to multiple bit access, an input error occurs.
- When specifying a 32-bit value for the constant, L is inserted at the tail end.
  Ex: When a constant value of 40000 is entered, 40000L is shown.

9.1.3.4 32-bit access for word OM

The memory range for a word OM is not 16-bit but 32-bit.

An “L” is inserted after the word OM number.

Input example	Description
D16L	Accesses a 32-bit segment for D16 and D17.

- When the OM is outside the range due to the 32-bit access to the word OM, an input error occurs.
- There is no start number restriction when the OM number is within range for 32-bit access of a word OM.
- Insert an “L” at the tail end of the numerical value when performing 32-bit changes to the current value for the 32-bit access of a word OM.
  Ex: “32768L”

9.1.3.5 Specify OM indirectly

By using the index register (IX) to indirectly set an OM (modifying index), the user can specify an OM that includes a value stored in IX for OM numbers that can be modified.

Ex: When 10 is stored to IX0
M1IX0 can be used to express M11.

(NOTICE) Set the IX value so that the OM number does not exceed the maximum value based on index addressing. If the OM number exceeds the maximum value, an out of range OM is specified and another type of OM may be read and written. Be careful when specifying the number, otherwise this will cause the program to malfunction.

9.1.3.6 Undo edit operation

Press **<**Undo> to undo the edit operation.
Each time <Undo> is pressed, it undoes the previous edit operation. The undo operation can be performed a maximum of 5 times. However, after a conversion execution, the undo operation cannot restore to the status or condition prior to conversion.
The undo operation can be performed for the following:

- Write character string
- Replace

9.1.3.7 Write during RUN

In a control program, when the user presses **<**Convert**>**, the unconverted circuit is converted into binary code for execution and the file is saved. In addition, the user can also write the changed portion without stopping the program during operation (during RUN).

Conditions

- <Write during run after conversion> is set to <1: Yes> in the conversion mode setting.
- When making changes to the same control program that is operating.

Procedures

1. Press the **<**Convert**>** key to execute the conversion.
2. The popup message <Write during run?> is displayed.
   - Pressing **<**Yes**>** verifies or cross-checks the control program that is running. If the program matches, it executes the write operation during RUN and returns to the edit screen. If the program does not match, the message <Write during RUN is not possible due to program mismatch.> appears, and it returns to the edit screen without executing the write operation during RUN.
   - Pressing <No**>** returns it to the edit screen without executing the write operation during RUN.

In addition, the file is saved even when the alarm <<Double coil error>> and <<There is a control program outputting outside of the writing range>> are triggered.

9.1.3.8 Convert

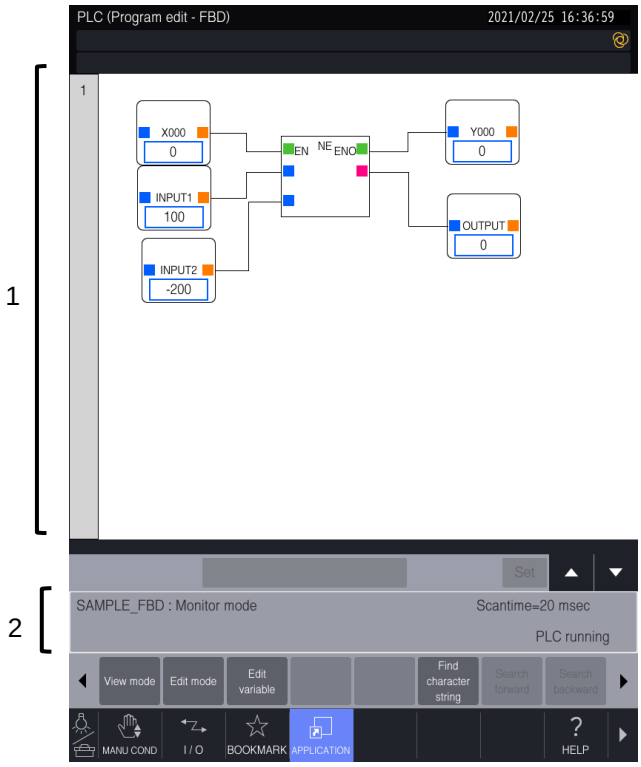
The user needs to carry out a conversion operation in order to run an edited FBD program as an execution program in the built-in PLC. The checks in the table below are carried out during the conversion operation. If the check results are OK, then the conversion operation is completed. In addition, the message <Converting> appears during conversion. Once the conversion is completed, the message <Conversion is complete.> appears.

Item	Description
Statement check	Checks whether the statement complies with the language specification.
Variable type check	Checks whether the variable types match in the computation for the program processing.
Program size check	Checks whether the program size exceeds the maximum number of steps after the conversion operation.

9.1.4 Monitor mode

9.1.4.1 Screen layout

The following shows the screen layout for monitor mode.



Description of screen display

Position	Name	Description
1	Program display area	Displays the program. The user can change the display by moving the cursor or by swiping vertically.
2	Instructions area	Displays the scan time and the status of the PLC.

Function key display area

Column	Position	Label	Description
1	(1)	<View mode>	Changes to the program view mode.
	(2)	<Edit mode>	Changes to the program edit mode.
	(3)	<Edit variable>	Opens the <PLC (Edit variable)> screen. Refer to “9.2 Edit variables” for further details.
	(4)	-	-
	(5)	-	-
	(6)	<Find character string>	Searches a program. There are three types of searches: an OM, a step and a command search. Refer to “9.1.2.5 Search” for further details.
	(7)	<Search forward>	Searches downward and goes to the next character string found in the search. Refer to “9.1.2.5 Search” for further details.
	(8)	<Search backward>	Searches upward and goes to the next character string found in the search. Refer to “9.1.2.5 Search” for further details.

Column	Position	Label	Description
2	(1)	<View mode>	Changes to the program view mode.
	(2)	<Change network>	Displays the network for the next network number. If the displayed network is the last one, then the first network is displayed here.
	(3)	<Network jump>	Press this key, and enter a network number in the data input field. After pressing the **[ENT]** key, it jumps to the specified network.
	(4)	<Change current value>	Changes the current value for the variable in the program display area. Refer to “9.1.4.2 Change current value” for further details.
	(5)	<Clear all local Oms>	Clears all local OMs displayed except OMs for SM and SD. Refer to “9.1.4.4 Clear all local OMs” for further details.
	(6)	<Execute>	Starts operation for the built-in PLC. Refer to “9.1.4.3 Execute / Stop” for further details.
	(7)	<Stop>	Stops operation for the built-in PLC. Refer to “9.1.4.3 Execute / Stop” for further details.
	(8)	<Clear all global Oms>	Clears all global OMs displayed except OMs for X and Y. Refer to “9.1.4.5 Clear all global OMs” for further details.

9.1.4.2 Change current value

The user can change the current value for the variable / OM.

1. Tap on the desired variable or OM block, and press the **<**Change current value> key.
2. Enter the set value, and press the **[ENT]** key.

9.1.4.3 Execute / Stop

The user can start and stop operation for the built-in PLC. The user can only use these operations for a program registered as an execution program.
When **<**Execute**>** is pressed, the popup message <Start PLC operation?> is displayed.

<Yes**>**: Starts operation for the built-in PLC.
<No**>** : Cancels the process.

(NOTE) Operation cannot be started for the built-in PLC during an alarm that is related to the built-in PLC system, such as a PLC version mismatch or using an invalid control program.

When **<**Stop**>** is pressed, the popup message <Stop PLC operation?> is displayed.

<Yes**>**: Stops built-in PLC operation.
<No**>** : Cancels the process.

9.1.4.4 Clear all local OMs

The user can clear all local OMs in the program being monitored except OMs for SM and SD.
When **<**Clear all local OM**s>** is pressed, the popup message <Clear all local OMs?> is displayed.

<Yes**>**: Clears all the local OMs.
<No**>** : Cancels the process.

(NOTE) The clear operation for all local OMs is not available when a program or a debug process is running. The message <Stop the program execution and re-execute it.> appears in the instructions area.

9.1.4.5 Clear all global OMs

The user can clear all global OMs in the program being monitored except OMs for X and Y.
When **<**Clear all global OMs**>** is pressed, the popup message <Clear all global OMs?> is displayed.

<Yes**>**: Displays the popup message <Clear the power failure hold area?>.
<No> : Cancels the process.

<Yes**>**: Clears all global OMs including the power failure hold area.
<No**>** : Clears all global OMs excluding the power failure hold area.

(NOTE) The clear operation for all global OMs is not available when all programs or a debug process is running. The message <Stop the program execution and re-execute it.> appears in the instructions area.

9.2 Edit Variables

When in edit mode on the <PLC (Edit variable)> screen, the user can define available variables in an FBD program. When in display / monitor mode, the user can reference the variables.
Refer to “Chapter 12 Commands (FBD)” in the PLC System Manual (Option) for further information on the scope, data types and their default values.

Scope	Variable name	Data type	Eleme...	Default value	Comment
VAR_INPUT	VAR_IN	BOOL	0	FALSE	
VAR	VAR	BOOL	0	FALSE	
VAR_INPUT	VAR_IN_3	BOOL	0	FALSE	

Description of screen display

Position	Name	Description
1	Scope	The user can set the valid scope of the variable.
2	Variable name	The user can enter the variable name. The user can use alphanumeric characters (does not distinguish between uppercase and lowercase) and an “_” (underscore) symbol. However, a number cannot be used as the beginning of the variable name. When the scope is <VAR_EXTERNAL>, the user can select only from variables that are defined on the <PLC (Global variable settings)> screen.
3	Data type	The user can select and set the data type. The type cannot be changed when the variable is being used in the program and when the scope is <VAR_EXTERNAL>.
4	Element number	The user can set the element number of the variable. When the element number is 0, it is a single variable. When the element number is not 0, it is an array variable. The element number cannot be changed when the variable is being used in the program and when the scope is <VAR_EXTERNAL>.
5	Default value	The user can set the initial value right after the power is turned ON and the program is written. The value cannot be changed when the scope is <VAR_EXTERNAL>.
6	Comment	The user can enter a comment for a variable. The comment cannot be changed when the scope is <VAR_EXTERNAL>.

Function key display area (view mode)

Column	Position	Label	Description
1	(1)	Back to previous edit screen	Changes to the program view mode.
	(2)	-	-
	(3)	-	-
	(4)	-	-
	(5)	-	-
	(6)	-	-
	(7)	-	-
	(8)	-	-

Function key display area (in edit mode)

Column	Position	Label	Description
1	(1)	Back to previous edit screen	Changes to the program edit mode.
	(2)	Copy variable	Records the variable selected in the variable edit area. Refer to “9.2.2 Copy / Paste” for further details.
	(3)	Paste variable	Pastes the variable that is recorded with **<**Copy variable>. Refer to “9.2.2 Copy / Paste” for further details.
	(4)	Insert variable	Creates and inserts a new variable at the selected line inside the variable edit area. Refer to “9.2.1 New variable” for further details.
	(5)	Insert external variable	Creates and inserts a new external variable at the selected line inside the variable edit area. Refer to “9.2.1 New variable” for further details.
	(6)	Delete variable	Deletes the selected variable inside the variable edit area. Refer to “9.2.3 Delete” for further details.
	(7)	Undo	Cancels the program edit operation and returns it back to the previous state. The undo operation can be performed a maximum of 5 times. After a conversion, the undo operation cannot be used.
	(8)		

Function key display area (in monitor mode)

Column	Position	Label	Description
1	(1)	Back to previous edit screen	Changes to the program monitor mode.
	(2)	-	-
	(3)	-	-
	(4)	-	-
	(5)	-	-
	(6)	-	-
	(7)	-	-
	(8)	Change current value	Changes the current value for the variable in the program display area. Refer to “9.1.4.2 Change current value” for further details.

9.2.1 New variable

The following shows the procedure for creating a variable.

1. Press **<**Insert variable**>** to create and insert a new variable at the selected line.
2. Set the scope for the variable that was created.
3. After the variable name is automatically assigned, the user can edit it to any given name.
4. Edit the data type, and set the default value and/or comment when necessary.
   If the user selects <VAR_EXTERNAL> for the scope in step 2, this operation is not required.

The maximum number of variables in one POU is as follows.
BOOL type :1024
Non-BOOL type :1024

(NOTE 1) When non-BOOL type variables are created, the number of variables that can be used varies according to the data type.
Refer to “6.3.1. Available data capacity” for further details.

(NOTE 2) When creating arrays, the total element number for each array that can be created cannot exceed the maximum number of BOOL type or non-BOOL type variables.

(NOTE 3) When a variable is executed and the maximum number of BOOL type variables is exceeded, the message <Data size used in the BOOL type parameter variables exceeds the limit: 1024> appears in the conversion error display area. When the maximum number of non-BOOL type variables is exceeded as well, the message <Data size used in the non-BOOL type parameter variables exceeds the limit: 1024> appears in the conversion error display area.

9.2.2 Copy / Paste

The user can copy and paste a variable to duplicate it. The procedure is noted below.

1. Press **<**Copy variable**>** to copy the selected variable.
2. Move the cursor to a given position, and then press **<**Paste variable**>**. The variable copied in step 1 is inserted.

9.2.3 Delete Variable

The user can press **<**Delete variable**>** to delete the selected variable by line.

CHAPTER 10

PLC AXIS

10.1 Overview
10.2 Parameters
10.3 Door Interlock Specifications
10.4 Screen Display
10.5 Manual Operation Mode
10.6 PLC-Axis Control Using Built-in PLC
10.7 Clamp Mechanism
10.8 Special Notes

10.1 Overview

The NC control axis (SXYZ45678M axis) is independent and can control up to four PLC axes. The PLC-axis is controlled based on the PLC-axis control information in the control program that is executed in the PLC with a built-in NC. However, some operations are possible in manual operation mode.

Main specifications

Item	Description
Control axes	Up to 4 axes
Type	Linear axis: mm (inch) Rotation axis: deg
Minimum command unit	Linear axis: 0.001 mm (0.0001 inch) Rotation axis: 0.001 deg
Feedrate	Linear axis: 0 – 9999999 mm/min (0 – 999999 inch/min) Rotation axis: 0 – 9999999 deg/min
Travel command	• Increase value command (increment) from current position • Absolute value command (absolute) for machine coordinate system 0 to ±9999999 Linear axis: 0.001 mm (0.0001 inch) Rotation axis: 0.001 deg
Operation mode	Rapid feed, JOG feed and reference position return
Backlash compensation	Yes
Pitch error compensation	None
Stroke	Linear axis: Yes Rotation axis: Yes / None
Coordinate system	Machine coordinates and relative coordinates (Absolute coordinates display the same values as the machine coordinates)
Override	Yes

(NOTE) However, the PLC axis unit depends on the minimum setting value, except for the feedrate.

10.2 Parameters

10.2.1 Machine Parameter

No.	Item name	Setting range	Description
18**0001 **: P1-axis → 11 P2-axis → 12 P3-axis → 13 P4-axis → 14	Optional axis	0~1	
18**0002	Type	0~1	<0: Linear axis, 1: Rotation axis> Linear axis: Displays and controls the axis using the mm/inch system. Rotation axis: Displays and controls the axis using the degree system.
18**0003	Installation position	0~4	<0: Individual, 1: Table 1, 2: Table 2, 3: Outside 1, 4: Outside 2> Sets the installation position for each axis. Depending on the installation position, the conditions vary for servo OFF and stoppage during an open door operation. The <0: Individual> selection is valid for machines without QT and is set when installing inside the machine. <1: Table 1> and <2: Table 2> selections are valid for machines with QT and are set when installing on pallet 1 or 2. <3: Outside 1> and <4: Outside 2> selections are set when installing outside of the machine. The servo turns OFF when the input signals <IL_DR5A1> and <IL_DR5B1> turn ON for the selection <3: Outside 1>, or when the input signals <IL_DR6A1> and <IL_DR6B1> turn ON for the selection <4: Outside 2>. The parameter settings are used only for speed control and during a PLC-axis stop as specified by the safety specifications (i.e. when the door is opened). The address in the coordinate display is not configured by this parameter setting.
18**0004	Backlash compensation	0~99999 pulses	The backlash offset is measured regularly according to the machine usage conditions. Each axis is offset or compensated by the number of pulses when necessary.
18**0005	Lead/gear ratio	1~999 mm 1~999	Linear axis: Sets the lead on each axis (distance traveled for 1 motor rotation). Rotation axis: Sets the gear ratio on each axis (motor speed required for the axis to rotate 360 degrees).
18**0006	Absolute encoder rotation direction	0~1	<0: Forw. with +command, 1: Forw. with –command> Sets the rotational direction of the absolute encoder on each axis.
18**0007	Brake	0~5	<0: None, 1 to 5: Brake 1 to 5> Sets whether the machine is equipped or not equipped with a brake on each axis. The <Brake 1 to 5> selection sets which input signal for SR_BKIN*_N (*:1 to 5) to check as the brake input signal. If another setting other than <0: Type 1> is selected for the <Clamp mechanism> (P1- to P4-axes), and this parameter is not set to <0: None>, the alarm <<Machine param. setting error (PLC)>> (Stop level 5, recovery level 3) is triggered. When setting the brake that is used on another axis, <<Machine param. setting error (PLC)>> (Stop level 5, recovery level 3) is triggered.

No.	Item name	Setting range	Description
18**0008	Clamp mechanism	0~2	<0: Type 1, 1: Type 2, 2: Type 3> Sets the type for the clamp mechanism on each axis. The <0: Type 1> setting is used when there is no clamp mechanism. The <1: Type 2> setting is used when there is a clamp mechanism and the servo is turned OFF during the clamp operation. The <2: Type 3> setting is used when there is a clamp mechanism and the servo is turned ON during the clamp operation. If another setting other than <0: None> is selected for the <P1 to P4 axes brake>, and this parameter is not set to <0: Type 1>, the alarm <<Machine parameter setting error (PLC)>> (Stop level 5, recovery level 3) is triggered.
18**0009	Unclamp check input signal	0~1	<0:Confirm with 0, 1: Confirm with 1> Sets the polarity of the unclamp confirmation input signal on each axis.
18**0010	Clamp check input signal	0~2	<0: Invalid, 1: Confirm with 0, 2: Confirm with 1> Sets the polarity of the clamp confirmation input signal on each axis. If the <0: Invalid> setting is selected and the clamp operation is complete, the operation is confirmed using the OFF status for the unclamp confirmation input signal (not the clamp confirmation input signal).
18**0011	Unclamp output signal	0~1	<0: 0=Unclamp, 1: 1=Unclamp> Sets the polarity of the unclamp output signal on each axis.
18**0012	Unclamp check time	0~9999 msec	Sets the confirmation time from when the unclamp confirmation input signal on each axis turns ON until the rotation operation is executed.
18**0013	Clamp check time	0~9999 msec	Sets the confirmation time from when the clamp confirmation input signal on each axis turns ON until the next operation is executed.
18**0014	Position check pulse	0~999999 pulses	Sets the tolerance for misalignment when clamped on each axis.
18**0015	NC stop level for servo error	1~5	<1: Stop level 1, 2: Stop level 2, 3: Stop level 3, 4: Stop level 4, 5: Stop level 5> Sets the stop level on the machine side (excluding the PLC-axis, other axes) when a servo error is triggered on the PLC-axis. When not set, the default level is <5: Stop level 5>. A servo communication error and servo status error are a stop level 5 regardless of this parameter setting. A servo warning is a stop level 1 regardless of this parameter setting. (NOTE) The operation on the machine side does not immediately stop when a PLC-axis servo error is triggered if this parameter is set to <3: Stop level 3>, <2: Stop level 2> or <1: Stop level 1>. Be careful of interference between moving parts controlled by the PLC-axis and the machine.
18**0016	In-position timeout period	0~9999 msec	Sets the time allowed for the in-position check on each axis. The check operation does not execute when the time is “0”.
18**0017	Positioning end check distance/ angle	0.001~9.999 mm 0.001~9.999 degrees	Sets the allowable range for the final position when the position operation is finished.
18**0018	Positioning check time	0~9999 msec	Sets the confirmation time starting after the rotation is complete on each axis until the unclamp output signal turns OFF.
18**0019	Rapid feedrate	1~9999999 mm/min 1~9999999 deg/min	Sets the maximum feedrate on each axis.
18**0020	Manual speed	1~9999999 mm/min 1~9999999 deg/min	Sets the maximum feed rate on each axis during manual operations (JOG, STEP and HANDLE).

No.	Item name	Setting range	Description
18**0021 to 18**0023	Manual time constant 1 Manual time constant 2 Manual time constant 3	0~400 msec	Sets the acceleration and deceleration time constant on each axis during manual operations (JOG, STEP and HANDLE). It accelerates and decelerates using the rapid feed time constant 1, 2 and 3 when the manual time constants 1, 2 and 3 settings are all 0 or are not set. Even if values greater than 100 msec. are set, they are restricted to 100 msec. when the machine is equipped with a clamp mechanism. It accelerates and decelerates using the rapid feed time constant 1, 2 and 3 when a JOG command is issued from the internal PLC.
18**0024	Rapid feedrate (door open)	1~9999999 mm/min 1~9999999 deg/min	Sets the maximum feed rate on each axis when a door is open. When the <Type> (P1- to P4-axes) is set to <0: Linear axis>, even if the value that is entered exceeds 2000, the actual feed rate is limited 2000 mm/min. When the <Type> (P1- to P4-axes) is set to <1: Rotation axis>, even if the value that is entered exceeds 18000, the actual feed rate is limited to 18000 deg/min.
18**0025	Rapid feedrate (Outer pallet)	1~9999999 mm/min 1~9999999 deg/min	Sets the maximum feedrate when P1 through P4 axes are at the outer pallet. The axes are also controlled with this speed when the pallet is not indexed.
18**0026 to 18**0028	Rapid feed time constant 1 Rapid feed time constant 2 Rapid feed time constant 3	0~9999 msec	Sets the time constant for acceleration and deceleration during the rapid feed operation on each axis.
18**0029	Brake load test	0~1	<0: No test, 1: Test> Set whether or not to execute the brake load test on the each axis. Set to <1: Test> on axes that fall due to a dead load. When the brake load test is set to <1: Test> on axes where <Brake> (P1 to P4-axes) is set to <0: None>, the alarm <<Machine param. setting error (PLC)>> (Stop level 5, recovery level 3) is triggered.
18**0030	Load torque for brake load test	1~999%	Set the torque for the load applied to each axis brake in the brake load test. The set value is the ratio when the rated torque for the motor is 100%.
18**0031	Travel amount under load for brake load test	0.001~9.999 mm 0.001~9.999 degrees	Set the travel amount that applies a load on each axis brake.
18**0032	Allowable travel amount for brake load test	0.001~9.999 mm 0.001~9.999 degrees	Set the tolerance value for the shift in position when the load is applied to each axis brake. After applying the load to the brake, when the travel amount exceeds the set value, the alarm <<Brake load test error on P*-axis>> is triggered.
18**0033	Delay time for brake operation	0.0~9999.9 msec	Set the time it takes from when the brake release signal for each axis turns OFF until the brake is applied.
18**0034	Delay time for brake release	0.0~9999.9 msec	Set the time it takes from when the brake release signal for each axis turns ON until the brake is released.
18**0035	Load time for brake load test	1.0~9999.9 msec	Set the time for the load being applied on each axis brake.
18**0036	Timeout time for load during brake load test	0.0~9999.9 msec	Set the time allowed for the load that normally applies to each axis brake during the brake load test. The check operation does not execute when the time is “0.0”.
18**0037	Wait time before applying brake in brake load test	0.0~9999.9 msec	Set the wait time before applying the brake on each axis in the brake load test.
18**0038	Wait time before releasing brake in brake load test	0.0~9999.9 msec	Set the wait time before releasing the brake on each axis in the brake load test.
18**0039	Door open time constant 1	0.0~99.9 msec	Set the acceleration and deceleration time constant for each axis when <Door interlock mode> is set to <Machine setup> and a door is open. <Door open time constant 1> is the time it takes to reach the
18**0040	Door open time constant 2	0.0~99.9 msec	

No.	Item name	Setting range	Description
18**0041	Door open time constant 3	0.0~99.9 msec	command speed. <Door open time constant 2> and <Door open time constant 3> are the times for ideal acceleration and deceleration.
	Adjustment mark	0~1	<0: No, 1: Yes> Set a mark that is used for the position adjustment on each axis.
	Marked position for adjustment	-9999.999~9999.999 mm -9999.999~9999.999 degrees	Set the marked position that is used for the position adjustment on each axis with the machine coordinates.

10.2.2 User Parameter

No.	Item name	Setting range	Description
110001 to 110004	P1-axis stroke control P2-axis stroke control P3-axis stroke control P4-axis stroke control	0: No 1: Yes	When a PLC-axis type is set as the rotation axis, specify <1: Yes> for stroke control and specify <0: No> to cancel stroke control. When <1: Yes> is selected, the values become valid for <P1 to P4-axes stroke (-) and (+)>. (NOTICE) If this parameter is set to <1:Yes> and the machine parameter (PLC: PLC-axis) <Lead/gear ratio> (P1- to P4-axes) is set to "1", when the power is turned OFF after P1- to P4-axes have been moved to somewhere outside the stroke or when the P1- to P4-axes are moved to somewhere outside of the stroke while the power is OFF, then the coordinates for the P1- to P4-axes may be out of position at power ON.
110005 to 110012	P1-axis stroke (-) P2-axis stroke (-) P3-axis stroke (-) P4-axis stroke (-) P1-axis stroke (+) P2-axis stroke (+) P3-axis stroke (+) P4-axis stroke (+)	-9999999 to 9999999	The stroke value is input in the machine coordinates, if the linear axis or the rotation axis is set for the PLC-axis type and the <1: Yes> setting is selected. The unit varies depending on the minimum unit setting. When the minimum unit setting is <Type 1 (Micron)> Linear axis (meters): 0.001 mm Linear axis (inches): 0.0001 inch Rotation axis: 0.001 degrees When the minimum unit setting is <Type 2 (Submicron)> Linear axis (meters): 0.0001 mm Linear axis (inches): 0.00001 inch Rotation axis: 0.0001 degrees (NOTE) In the following situations, the error <<User param. setting error (PLC)>> is triggered. - When the machine parameter (PLC: PLC-axis) <Optional axis> (P1- to P4-axes) is configured to another setting besides <No>, when the machine parameter (PLC: PLC-axis) <Type> (P1- to P4-axes) is set to <1: Rotation axis>, when the machine parameter (PLC: PLC-axis) <Lead/gear ratio> (P1- to P4-axes) is set to "1", when the user parameter (PLC) <P1- to P4-axis stroke control> is set to <1:Yes> and when the <P1- to P4-axis stroke(+)> and <P1- to P4-axis stroke(-)> are set to a value larger than 350 degrees
110013 to 110016	Reference point P1 axis Reference point P2 axis Reference point P3 axis Reference point P4 axis	-9999999 to 9999999	Sets the travel position for the reference position return command by using the machine coordinate position. The unit varies depending on the minimum unit setting. When the minimum unit setting is <Type 1 (Micron)> Linear axis (meters): 0.001 mm Linear axis (inches): 0.0001 inch Rotation axis: 0.001 degrees When the minimum unit setting is <Type 2 (Submicron)> Linear axis (meters): 0.0001 mm Linear axis (inches): 0.00001 inch Rotation axis: 0.0001 degrees

No.	Item name	Setting range	Description
110017 to 110032	P1-axis override speed 1 P1-axis override speed 2 P1-axis override speed 3 P1-axis override speed 4 P2-axis override speed 1 P2-axis override speed 2 P2-axis override speed 3 P2-axis override speed 4 P3-axis override speed 1 P3-axis override speed 2 P3-axis override speed 3 P3-axis override speed 4 P4-axis override speed 1 P4-axis override speed 2 P4-axis override speed 3 P4-axis override speed 4	0 to 9999999	Specifies the override speed. The following units are used regardless of the minimum unit setting. Linear axis (meters): mm/min Linear axis (inches): inch/min Rotation axis: degrees/min
110033 to 110036	Collis. detct.-P1ax stop meth. Collis. detct.-P2ax stop meth.` Collis. detct.-P3ax stop meth. Collis. detct.-P4ax stop meth.	0: Method 1 1: Method 2	Select the stop method for the PLC axis when a collision detection alarm is triggered. If <0: Type 1> is selected, it stops at the same time as the rapid feed when a collision detection alarm is triggered. It takes a longer time to stop, but no vibrations occur caused by sudden stopping. If <1: Type 2> is selected, it stops in the shortest time when a collision detection alarm is triggered. Vibrations can occur due to a sudden stop.

10.2.3 Special Notes When Changing Parameters

If the data bank is changed and an alarm is triggered, all axes including the PLC axis are temporarily stopped. (Depending on the parameter changes, the servo can also be turned OFF.) Cancel levels 2 and 3 are the levels that reflect the parameter changes.

- Cancel level 2
  The value is reflected with the **[RST]** key. The PLC axis cannot be operated until the **[RST]** key is pressed.
- Cancel level 3
  The value is reflected when the power is turned OFF. The PLC axis cannot be operated until the power is turned ON again.

10.3 Door Interlock Specifications

In the PLC-axis door interlock specification, when the machine parameter (PLC: PLC-axis) <Installation position> (P1- to P4-axes) is set to <0: No pallet>, <1: Table 1> or <2: Table 2>, the operation is the same as the additional axes: 5th- to 8th-axes.
When set to <3: Outside 1> or <4: Outside 2>, the operation follows the specifications noted below.

10.3.1 Special Notes When Changing Door Interlock Mode

When the door interlock mode is changed, all axis servos are temporarily turned OFF and then turned ON again.
Be careful because an axis may shift or move.

10.3.2 Operation Status in Each Door Interlock Mode

(Description provided in table)
○: Full operation is possible ×: Full operation is not possible Δ: Operation is restricted

Door interlock mode / Outside door status / Enable switch / Operation type	Auto. operation		Machine setup			Service	
Outside door status	Closed	Open	Closed	Open		Closed	Open
Enable switch	-	-	-	OFF	ON	-	-
Manual operation	○	×	○	×	Δ	Δ	Δ
PLC-axis control using built-in PLC	○	×	○	×	×	×	×

10.3.3 Restricted Operation in Each Door Interlock Mode

- Service mode
  Axis travel is not possible during spindle and lathe spindle rotation.

10.3.4 Limits on Operation When Doors are Open

If a door is open and the door interlock mode is set to machine setup or service mode, operation is restricted.
The axis travel speed and axis rotation speed is restricted to a low speed. In addition, the operation restrictions in the table below also apply.

(Description provided in table)
○: Operation is possible ×: Operation is not possible

Operation / Door interlock mode		Machine setup	Service
Manual operation	Jog feed	○	○
	Step feed	×	○ (NOTE)
	Handle operation	○	○

(NOTE) Operation is possible on the rotation axis if the maximum step travel amount is 1 degree.

10.3.5 Door Lock

The user can select the user parameter (switch 1: door) <Door lock method> to either automatically lock the door when the door is closed, or to automatically lock the door when operation starts. If the <1: Lock only when needed> is selected, the table below shows the corresponding operations when the machine locks the door automatically. If <0: Automatic lock when door closes> is selected and an attempt is made to start one of these operations while the door is unlocked, the alarm <<Door was not locked>> or <<Command not receivable with door unlocked>> is triggered. Lock the door using a manual operation. Refer to “3.4.3.2 Sub tool keys” in Operation Manual I for further details on operations.

(Description provided in table)
○: Door locks ×: Door does not lock -: Execution is not possible for corresponding operation

Door interlock mode / Enable switch / Operation	Auto. operation	Machine setup		Service
	-	OFF	ON	-
Manual operation	○	○	○(NOTE)	-
PLC-axis control using built-in PLC	○	○	○	-
Brake load test	○	○	○	○

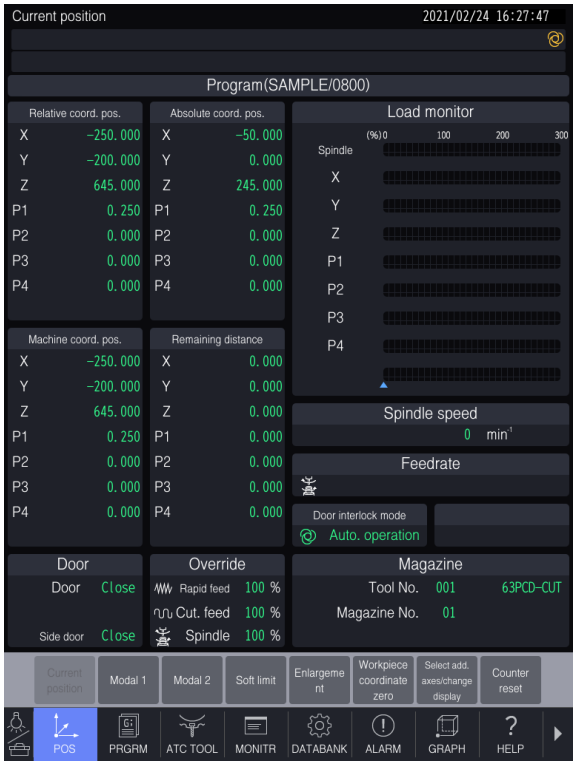
(NOTE) The door does not lock when the door is open.

10.4 Screen Display

The PLC-axis machine coordinates can be displayed on the screen using the following methods.

- Assign the PLC-axis in the user parameter <4th to 12th row display axis>, and press the <Change additional axes display> function key on the screen where the additional coordinates are displayed.

  Ex: Current position screen



- Enlarged display of position screen
  Press the <Enlarge> function key at the current position screen to enlarge the display of the coordinates for the PLC-axes.

10.5 Manual Operation Mode

Jog, step and handle operations are possible.
Travel operations at high speed, low speed or in steps can be performed using the **[+4]** and **[-4]** keys.
To operate the PLC axis using the **[+4]** and **[-4]** keys, display the manual conditions screen and select the axis with the **<**Change additional axes display> key.

Use the **[+4]** and **[-4]** keys to start operation after adjusting the corresponding manual conditions.

(NOTE) If operating the PLC axis in manual operation mode, the conditions for operation vary depending on the installation position settings.
If the installation position is set to <Individual>, <Table 1> or <Table 2>, operations are executed under the same conditions as an additional axis.
If the installation position is set to <Outside 1> or <Outside 2>, an NC alarm is triggered on the NC and operation is not possible.

10.5.1 High-speed Travel

The machine parameter (PLC: PLC-axis) <Manual speed> (P1- to P4-axes) can be used to set the travel speed on the P1- to P4-axes to 5 different levels: 10%, 25%, 50%, 75% and 100%.

The PLC axis can be moved in the desired direction using the **[+4]** and **[-4]** keys. When changing the travel speed, move the cursor on the manual conditions screen to the setting <High travel speed (XYZ/P)> or <High rotation speed (4/P)>, and use the **[+]** or **[-]** key to change the high travel speed.

(NOTE) When the <Type> (P1- to P4-axes) is set to the linear axis, the operation speed follows <High travel spd(XYZ/P)>, and when it is set to the rotation axis, the operation speed follows <High rotation spd(4/P)>.
When changing the high travel speed during axis travel, the change is immediately reflected in the travel speed.

10.5.2 Low-speed Travel

The travel speed of the PLC axis, which is configured to the linear axis type, can be set and operated at 22 different levels from 50 to 4000 mm/min (2.0 to 157.4 inch/min). In addition, the rotation speed of the PLC axis, which is configured to the rotation axis type, can be set and operated at 19 different levels from 0.1 to 7.5 min^{-1}.

The PLC axis can be moved in the desired direction using the **[+4]** and **[-4]** keys. When changing the travel speed, move the cursor on the manual conditions screen to the setting <Low travel speed (XYZ/P)> or <Low rotation speed (4/P)>, and use the **[+]** or **[-]** key to change the low travel speed.

(NOTE) When changing the low travel speed during axis travel, the change is immediately reflected in the travel speed.

10.5.3 Step Travel

The PLC axis can be moved only in steps. Step travel can be set and operated at various step levels or amounts: 0.001, 0.01, 0.1 and 1.0 mm/step (0.0001, 0.001, 0.01 and 0.1 inch/step, and 0.001 to 180.0 deg./step).

The PLC axis can be moved in the desired direction using the axis travel keys. When changing the travel amount of the step, move the cursor on the manual conditions screen to the setting <Step travel (XYZ/P)> or <Step rotation amount (4/P)>, and use the **[+]** or **[-]** key to change the travel amount for the step.

10.5.4 Handle Travel Operation

Use the handle to move the PLC axis. The travel amount and scale are the same as the X-, Y- and Z-axes when using a linear axis type, and they are the same as an additional axis when using a rotation axis type.

10.5.5 Other Operations

The PLC axis operation is not carried out even if the **[Z.RTN]** key is pressed.

10.5.6 Special Notes

The PLC axis operation in manual operation mode is only possible when the PLC axis control signal is OFF.
If the above operation is performed when the PLC axis control signal is ON, the alarm <<The PLC axis control signal is ON>> is triggered.
When the machine parameter (system 1: automatic door) <Option> is set to <1:Yes>, operation is not possible on the P4-axis in manual operation mode.

10.6 PLC Axis Control Using Built-in PLC

The flow of the PLC axis control on the built-in PLC is shown below.

1. Create a control program to control the PLC-axis.
2. The PLC axis control signal turns ON.
3. The NC controls the PLC axis based on the PLC control information in the ladder program that is executed on the built-in PLC.

10.6.1 Command Method Between PLC-NC

1. PLC axis control signal
   The PLC axis control processing is performed on the built-in PLC while the PLC axis signal is ON.
   When this signal is OFF, the following operations can be performed.
   - JOG travel in manual mode, etc.
   - Brake load test using an external input signal

Signal name	P1-axis	P2-axis	P3-axis	P4-axis
PLC axis control signal	BY034	BY035	BY036	BY037

   Note, when this signal is changed during a travel operation, the machine decelerates and stops. After decelerating and stopping, if the signal above is turned ON, the operation follows the command status in the PLC axis control data register. If the signal turns OFF, operation in manual operation mode is enabled.

2. PLC axis control data register
   The information used to control the PLC axis is set in the data register.

Input

Signal name	P1-axis	P2-axis	P3-axis	P4-axis
Status	BX080 to 08A	BX090 to 09A	BX0A0 to 0AA	BX0B0 to 0BA
Alarm/Warning	BDX080	BDX081	BDX082	BDX083
Servo error	BDX084	BDX085	BDX086	BDX087
Machine coordinate	BDX064 BDX065	BDX066 BDX067	BDX068 BDX069	BDX070 BDX071
Remaining distance	BDX072 BDX073	BDX074 BDX075	BDX076 BDX077	BDX078 BDX079
Torque command	BDX088	BDX089	BDX090	BDX091
Torque monitor	BDX092	BDX093	BDX094	BDX095

Output

Signal name	P1-axis	P2-axis	P3-axis	P4-axis
Command	BY040 to 045	BY050 to 055	BY060 to 065	BY070 to 075
Command type	BDY024	BDY025	BDY026	BDY027
Command data 1	BDY000 BDY001	BDY002 BDY003	BDY004 BDY005	BDY006 BDY007
Command data 2	BDY008 BDY009	BDY010 BDY011	BDY012 BDY013	BDY014 BDY015
Command data 3	BDY016 BDY017	BDY018 BDY019	BDY020 BDY021	BDY022 BDY023

3. Rapid traverse override

Signal name	Register	Description
PLC-axis override	BY031 BY032 BY033	Enables override during rapid feed and reference position return operations. The change is reflected even during a travel operation. (The change is reflected on the axis where the PLC-axis control signal is valid.) BY033 BY032 BY031 0 0 0 : Disabled (100%) 0 0 1 : Override speed 1 0 1 0 : Override speed 2 0 1 1 : Override speed 3 1 0 0 : Override speed 4 1 0 1 : Disabled (100%) 1 1 0 : Disabled (100%) 1 1 1 : Disabled (100%)

4. Machine lock

Signal name	Register	Description
PLC axis machine lock	BY030	Enables the machine lock on the PLC axis. The machine lock is only enabled on an axis where the PLC axis signal is valid. If this signal status changes during a travel operation, the machine operation stops temporarily and the machine lock status changes. In addition, the command issued during the travel operation is cleared.

10.6.2 PLC Axis Control Data Register Details

1. Command

BY04*:07*	Code	Description
0	START	Command start. Travel starts during the rise time when switching from OFF to ON based on the values for the command type / command data 1 through 3. The commands, for positioning during servo OFF and for the reference position return, are issued (start) after the servo turns ON. The command is not accepted when the PLC axis control signal is OFF. The command is not accepted when the status is BUSY. The command is not accepted when an ALARM status is triggered. Do not turn ON during the same cycle as the PLC-axis control signal.
1	RESET	Resets the alarm for the PLC-axis, excluding servo errors. A servo error for the PLC-axis cannot be reset. To reset, turn the PLC-axis control signal OFF and press the **[RST]** key. The current command is cancelled. When the signal turns ON during a travel operation, travel decelerates and stops before the reset processing is performed. The command is not accepted when the PLC-axis control signal is OFF. However, the alarm that appears on the NC screen is also reset (Only for the PLC-axis that is turned ON).
2	SVOFF	Changes the status to ON when the servo is OFF. The command is not accepted when the PLC axis control signal is OFF. When the signal turns ON during a travel operation (including temporary stoppage during a travel operation such as positioning or a reference position return), the alarm <<The P*-axis servo OFF command is detected>> is triggered, and the servo turns OFF after travel decelerates and stops. The alarm <<No comm receivable as P*-axis servo OFF is active>> is triggered when the signal turns ON at the start of a travel operation.

BY04*:07*	Code	Description
3	STOP	Temporary stop. The command is not accepted when the PLC-axis control signal is OFF. If the signal turns ON during a travel operation, travel decelerates and stops, and INPOK turns ON. If the signal is turned OFF before RESET or an alarm is triggered, the command is re-issued (re-started). However, only positioning and reference position return commands are re-issued (re-started). Turn RESET ON to avoid re-issuing (re-starting) the command. The alarm <<No command receivable as P*-axis stop is active>> is triggered when the signal is ON at the start of a travel operation.
4	ABSINC	This signal refers to the Abso./Incre. operation for a positioning command. If the signal turns ON, it becomes an Abso. command. When the command START changes from OFF to ON, the setting that corresponds to the ON status is valid.
5	NEAR	Near override enabled. When the command START changes from OFF to ON, the setting that corresponds to the ON status is valid.
6		
7		
8		
9		
A		
B		
C		
D		
E		
F		

2. Command type
   When the command START changes from OFF to ON, the setting that corresponds to the ON status is valid.
   JOG feed travels at the command speed while START is ON.

No.	Description
0	Positioning
1	
2	JOG feed (+)
3	JOG feed (-)
4	Reference position return
5	
6	
255	

(NOTE) Operation stops at the stroke end when moving from the inside the stroke to out of the stroke range while using JOG feed.

3. Command data 1 to 3
   The content for command data 1 through 3 varies depending on the command type.
   When the command START changes from OFF to ON, the setting that corresponds to the ON status is valid, except for the JOG feed speed command.

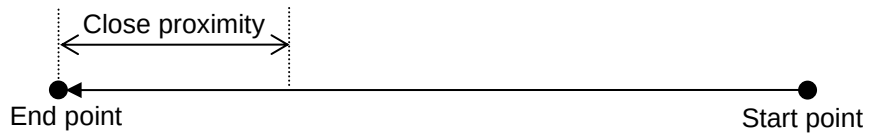
Command type	Command data 1	Command data 2	Command data 3
Positioning	Position command	Close proximity	Proximity speed
JOG feed	Invalid	Invalid	Speed command
Reference position return	Invalid	Close proximity	Proximity speed

Position command
Sets the travel command for the PLC-axis.
The meaning of the set value changes according to the flag value of the abso./incre. command.
0 (Incre.): Increase from the current position
1 (Abso.): Position of the machine coordinate

Refer to the table below for the commnd unit or the range. The alarm <<P*-axis position command error>> is triggered when a command is issued for data that falls outside of the range.

P*-axis type	Unit system	Command range	
		Micron	Submicron
Linear axis	Meter	-9999999 to 9999999 (0.001 mm)	-99999999 to 99999999 (0.0001 mm)
	Inch	-9999999 to 9999999 (0.0001 inch)	-99999999 to 99999999 (0.00001 inch)
Rotation axis	-	-9999999 to 9999999 (0.001 degrees)	-99999999 to 99999999 (0.0001 degrees)

Close proximity / Proximity speed
If the command 'near override' is enabled, the speed is controlled with the 'proximity speed' when the remaining distance to the end point is less than the 'close proximity' value that is set.



The alarm <<P*-axis position command error>> is triggered when a command is issued for data that falls outside of the range for close proximity shown below.

P*-axis type	Unit system	Command range	
		Micron	Submicron
Linear axis	Meter	0 to 9999999 (0.001 mm)	0 to 99999999 (0.0001 mm)
	Inch	0 to 9999999 (0.0001 inch)	0 to 99999999 (0.00001 inch)
Rotation axis	-	0 to 9999999 (0.001 degrees)	0 to 99999999 (0.0001 degrees)

The alarm <<P*-axis speed command error>> is triggered when the set value for the proximity speed is greater than the machine parameter (PLC: PLC-axis) <Rapid feedrate> (P1- to P4-axes) setting. In addition, the alarm <<P*-axis speed command error>> is also triggered when the value is set below 0.

Speed command
A speed command is issued during a JOG feed operation on the PLC-axis. The set value is immediately reflected in the travel speed.
The alarm <<P*-axis speed command error>> is triggered when the set value for the speed command is greater than the machine parameter (PLC: PLC-axis) <Rapid feedrate> (P1- to P4-axes) setting. In addition, the alarm <<P*-axis speed command error>> is also triggered when the value is set to 0.

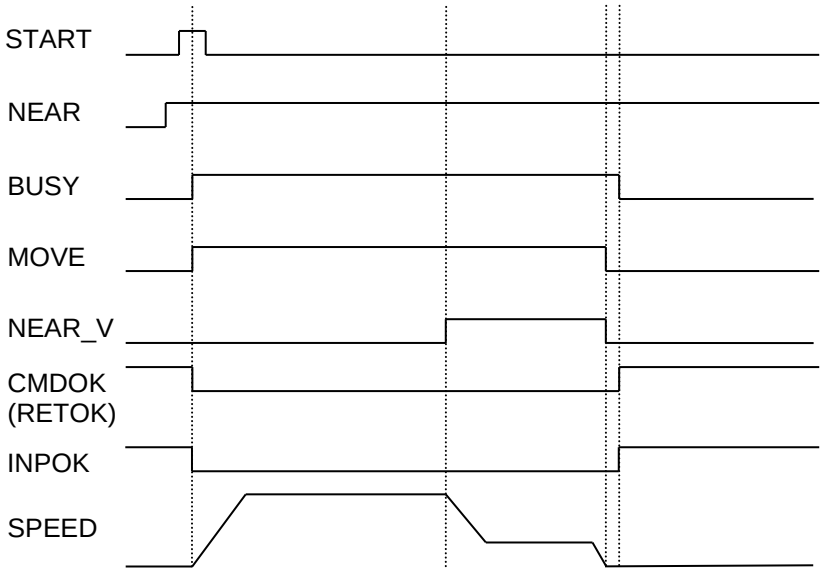
P*-axis type	Unit system	Command range
Linear axis	Meter	1 to <Rapid feedrate> (P1- to P4-axis) (mm/min)
	Inch	1 to <Rapid feedrate> (P1- to P4-axis) (inch/min)
Rotation axis	-	1 to <Rapid feedrate> (P1- to P4-axis) (degrees/min)

4. Status
   Regardless of the status of the PLC axis control signal, the signals below change when using either built-in PLC control or manual operation mode control.

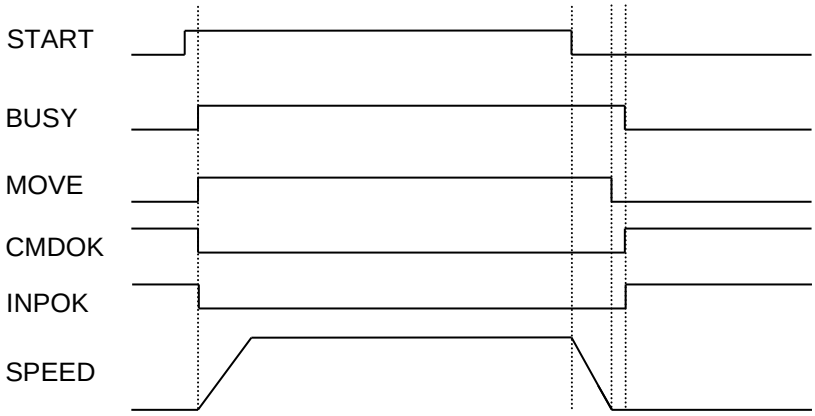
BX08* to 0B*	Code	Description
0	BUSY	Command processing. The command is not accepted (alarm is triggered) while the signal is ON. The signal turns ON during a command operation on the built-in PLC. The signal turns ON during a command operation in manual operation mode. If the machine is equipped with a handle, the signal turns ON when the handle axis is set as an additional axis and the additional axis is set as the P-axis. This signal turns ON when the <Door interlock mode> is set to <Maintenance>. This signal turns ON during the brake load test, or when there is a brake load test error. The signal turns ON when the PLC axis does not operate due to an NC alarm, such as an alarm that is triggered when an emergency stop or PLC related parameter is changed.
1	ALARM	The signal turns ON during an alarm. If the PLC-axis control signal is OFF, the alarm can be reset with the **[RST]** key. The servo may automatically turn OFF depending on the alarm type. The alarm can be reset with a reset command. The alarm information is set by the alarm/warning or servo error data register.
2	SRDY	The signal turns ON when the servo can be turned ON. If the signal is OFF and a servo ON command is received, an alarm is triggered.
3	SVON	The signal turns ON when the servo ON command is completed.
4	CMDOK	Command completed. The signal turns ON during the initializing status when the power is turned ON, and when a positioning or reference position return command is executed. (The signal turns ON when in position status applies at the end point. If equipped with a clamp mechanism, the signal turns ON when clamped.) The signal turns OFF if a new command is received. The signal also turns ON after a reset operation is completed and when the PLC-axis control signal turns OFF.
5	RETOK	Reference position return operation completed. The signal turns ON when a command is completed for the machine coordinate at the reference position.
6	MOVE	Axis in motion. The signal turns ON while the axis is in motion. The signal turns OFF if the speed command is 0 during a positioning command or a reference position return command.
7	INPOK	In-position OK. The signal turns ON when in position. The signal turns ON also when the STOP command is turned ON and operation stops temporarily. The signal turns OFF when a new command is received or when a command is re-issued (re-started).
8	NEAR_V	The speed is controlled when the proximity speed is engaged.
9	CLP	The signal turns ON when clamped. (If equipped with a clamp mechanism) The signal is always OFF if there is no clamp mechanism.
A	WRN	The signal turns ON when a warning (an error that does not stop travel) is issued. If the PLC-axis control signal is OFF, the alarm can be reset with the **[RST]** key. The warning can be reset with the RESET command. The warning information is set by the alarm/warning or servo error data register.
B		
C		
D		
E		
F		

5. Timing chart

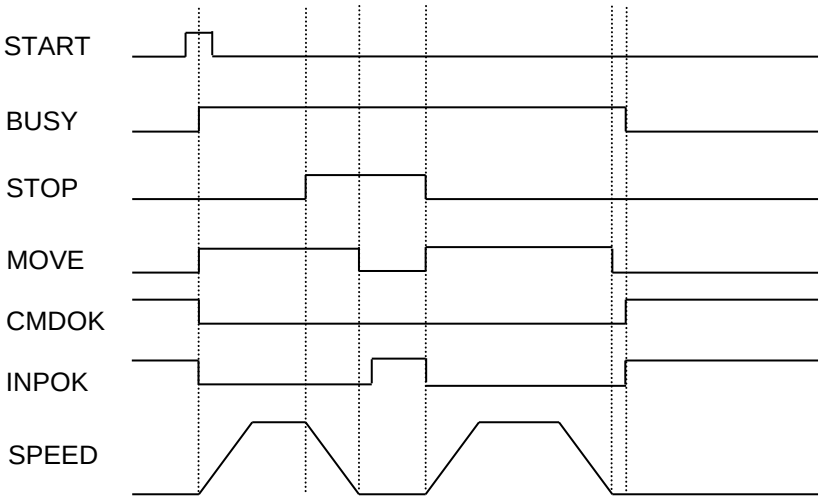
Positioning travel (with proximity override)



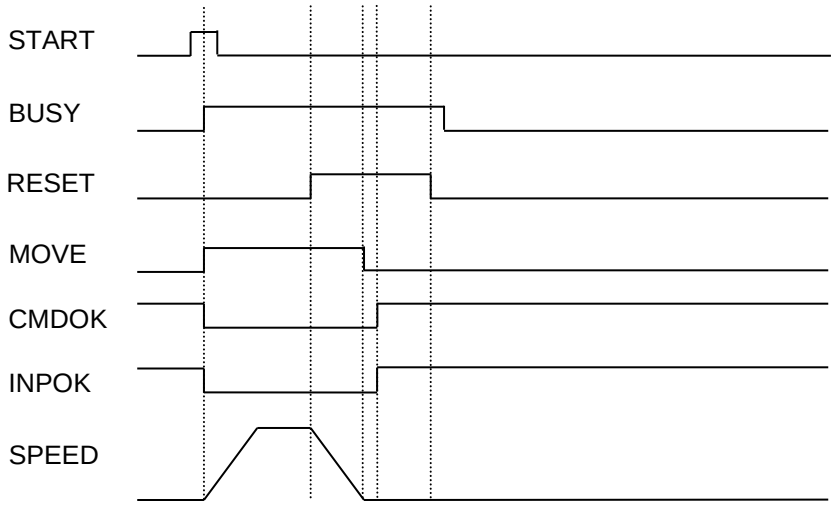
JOG travel



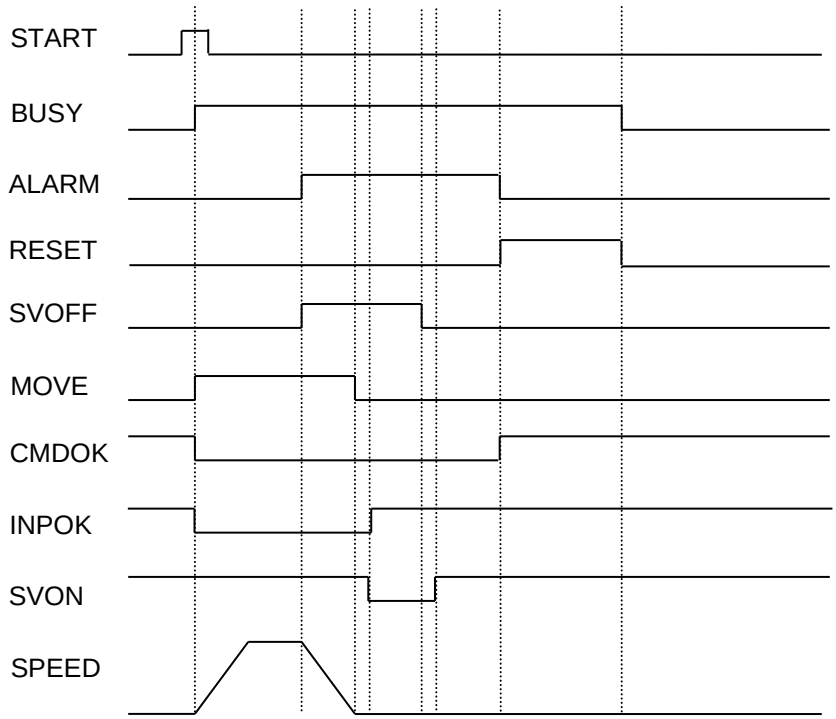
Temporary stop (pause) command



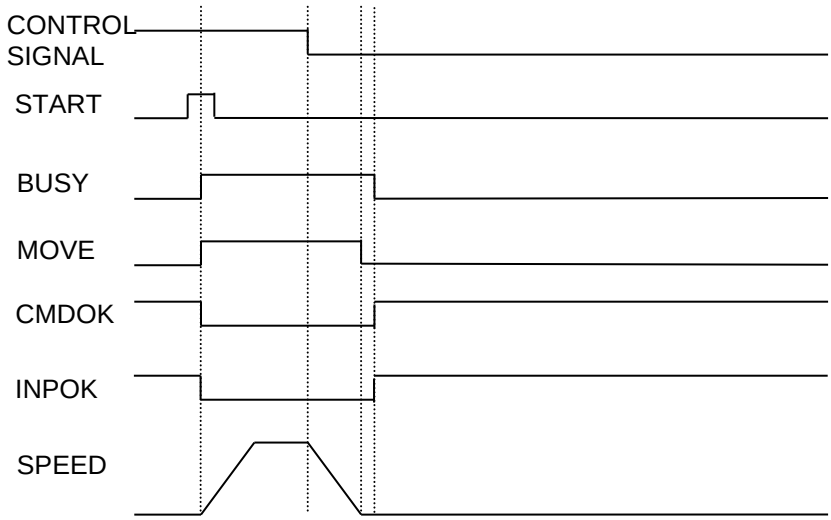
Reset command



Servo OFF command



When PLC-axis control signal is OFF



6. Alarm / Warning
If an alarm or warning is triggered on a PLC-axis, the subsequent operation follows the stop level/cancel level as noted below.
In this situation, the alarm is also displayed on the NC screen.
The stop level here only applies to the PLC-axis where the alarm occurred. Other PLC-axes are unaffected and the memory operation also does not stop.

PLC-axis alarm stop level
- Alarm stop level 3 for PLC-axis
  The servo turns OFF. In addition, the ALARM status turns ON.
- Alarm stop level 2 for PLC-axis
  The operation currently being executed stops immediately, and all operations thereafter are disabled.
  In addition, the ALARM status turns ON.
- Alarm stop level 1 for PLC-axis
  Only the WRN status turns ON, without impacting any operation.

Alarm cancel level for PLC-axis
- Alarm cancel level 2 for PLC-axis
  It does not cancel or reset unless the power is turned OFF.
- Alarm cancel level 1 for PLC-axis
  The alarm is canceled or reset when the RESET command is turned ON.
  If the PLC-axis control signal is OFF, the alarm is canceled or reset with the **[RST]** key.

Alarm No.	Description	PLC axis alarm stop level	PLC axis alarm cancel level
0	No alarm		
501	Servo cannot be ON	3	1
502	Servo power off	3	1
503	Speed error when door is open	3	1
504	In-position check timeout	3	1
505	Position shifted	3	1
1501	There is no option.	2	1
1502	Stroke exceeded (-)	2	1
1503	Stroke exceeded (+)	2	1
1504	Not clamped	2	1
1505	Cannot unclamp	2	1
1506	Clamp	2	1
1507	Unclamp	2	1
1508	Position shifted	2	1

Alarm No.	Description	PLC axis alarm stop level	PLC axis alarm cancel level
1509	Command error	2	1
1510	Position command error	2	1
1511	Servo OFF command is detected.	2	1
2501	No command is receivable as control signal is invalid. (NOTE 1)	1	1
2502	Command is rejected (command in process) (NOTE 1)	1	1
2503	Servo warning	1	1
2504	No command is receivable as servo OFF is active. (NOTE 1)	1	1
2505	No command is receivable as stop command is active. (NOTE 1)	1	1
2506	No command is receivable as door is open. (NOTE 1)	1	1
2507	No command is receivable because the servo is not valid. (NOTE 1)	1	1
2508	Command cannot be received during brake load test (NOTE1) (NOTE 2)	1	1
2509	Command not receivable with door unlocked (NOTE 1)	1	1
2510	Command not receivable during spindle rotation (NOTE 1)	1	1
2511	Command not receivable during lathe spindle rotation (NOTE 1)	1	1
2512	Command not receivable during service mode (NOTE 1)	1	1
2513	Side door lock restricted (NOTE 1)	1	1
	:		
65535			

(NOTE 1) The alarm is only triggered during the rise time of the START command.
(NOTE 2) This alarm is triggered when an attempt is made to move an axis during a brake load test, or an axis with one of the following alarms: <<Signal cannot be confirmed for P*-axis brake release.>>, <<Timeout for load check during brake load test on P*-axis>>, <<Brake load test start position error on P*-axis>> or <<Brake load test error on P*-axis>>.

When more than one alarm is triggered, the following description compares the current alarm that was triggered with a new alarm being triggered, and it shows which alarm is maintained.

- The new alarm being triggered is maintained when the cancel level for the new alarm is higher than the cancel level for the current alarm that was triggered.
- The new alarm being triggered is maintained when the stop level for the new alarm is higher than the stop level for the current alarm.
- The current alarm that was triggered is maintained when the stop level for the new alarm is lower than the stop level for the current alarm.
- The current alarm is maintained for all other situations not mentioned above.

7. Servo error
   The PLC-axis, where the servo error occurred, stops.
   The stop level on the machine side can be set using the machine parameter (PLC: PLC-axis) <NC stop level during axis servo error> (P1- to P4-axes).
   Refer to “2.13 Servo error list” in the Data Bank & Alarm Manual for further details on the servo errors.

8. Machine coordinates
   The machine coordinate for the PLC-axis is displayed (same as the screen display).
   The unit is based on the smallest command unit.

9. Remaining distance
   The remaining distance for the PLC-axis is displayed (same as the screen display).
   The unit is based on the smallest command unit.

10. Torque command
    The torque command for the PLC-axis is displayed (same as the load monitor on the screen display).
    The unit is 0.1%.

11. Torque monitor
    The torque monitor for the PLC-axis is diplayed.
    The unit is 0.1%.

10.6.3 Special Notes

When the machine parameter (system 1: automatic door) <Option> is set to <1:Yes>, P4-axis control is not possible using the built-in PLC.

10.7 Clamp Mechanism

The PLC axis is controlled with the clamp when the clamp mechanism in the machine parameter is set to type 2 or 3.
Normally, the clamp / unclamp operation is automatically controlled on the NC, without using the manual operation mode/built-in PLC command.

- Unclamps at start of a travel operation. (After servo ON when the clamp mechanism is set to type 2)
- Clamps when travel is completed. (After servo OFF when the clamp mechanism is set to type 2)
- If the clamp mechanism is set to type 2, the servo does not turn ON when the clamp is engaged, even if the SVOFF command is in a servo ON status.

The following operations engage the clamp when the PLC axis control signal is ON.

- RESET command is turned ON
- SVOFF command is turned ON
- STOP command is turned ON
- The machine lock on the PLC axis is enabled or disabled

The following operations engage the clamp when the PLC axis control signal is OFF.

- The **[RST]** key is pressed
- The **[FEED HOLD]** switch is pressed
- The **[M.LCK]** key is enabled or disabled

In addition, the clamp engages in the following situation regardless if the PLC axis control signal is ON or OFF.

- When an alarm is triggered that turns the PLC axis servo OFF
- When the PLC axis control signal is switched ON or OFF

10.8 Special Notes

When the PLC axis control signal is enabled:

1. Normally, the PLC axis does not stop except during a PLC axis alarm. (However, the PLC axis will stop for some alarms that occur on the NC or when the door is opened. In this situation, the travel command for the PLC axis is cleared.)
2. The PLC axis does not stop when using the **[RST]** key.
3. The PLC axis does not stop when using the **[FEED HOLD]** switch.
4. The PLC axis does not stop when using the **[EMERGENCY]** switch.
5. The PLC axis is not impacted by the **[DRY]** key.
6. The PLC axis is not impacted by the **[M.LCK]** key.
7. The safety specifications (such as the servo control when the door is opened and closed) are impacted by the installation position settings for the PLC axis.
8. Limits on operation from the <<Brake load test error on *-axis>> affect the PLC axis.

(Supplementary note)
Normally, the PLC axis is not impacted by any keys or switches, etc.
To link the PLC axis with key/switch operations, the PLC axis must be controlled in the ladder program.

(This page was intentionally left blank.)

CHAPTER 11

COMMANDS (LD)

11.1 LD Language Specifications
11.2 LD Basic Commands
11.3 LD Application Commands
11.4 Error Message List During LD Program Conversion

11.1 LD Language Specifications

11.1.1 Commands

The following shows the commands available in the LD language.

Basic command	35 commands available
Application command	71 commands available

- Basic command
  There are 5 types of basic commands: contact command, connection command, output command, program control command and end command. There are a total of 35 commands available that the user can create. Refer to “11.2.2 LD basic command description” for further details about the commands.

- Application command
  There are 15 types of application commands: data comparison, arithmetic operation, BCD-BIN conversion, transfer, branch, logic operation, rotation, shift, data processing, FIFO, loop, carry flag, date/time, palletize and user function. There are a total of 71 commands available that the user can create. Refer to “11.3.2 LD application command description” for further details about the commands.

11.1.2 Summary of Commands

Starting from the next section, short summaries of each command are presented as shown below.

SET

Function

Command	Description
SET	Turns ON the specified OM value and maintains that status. When turned OFF, the status of the OM does not change.

Usable OM

Bit									Word								Constant		Label	Nesting	WL designation	INDEX
X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
	○	○		○	○	○																○

Circuit diagram

Usable OMs are marked with a small circle ○.
The table below shows the OMs for the “Bits” and “Words”.

Type	OM
Bit	X, Y, M, BX, BY, LM, SM, LT(LTS, LTC), LC(LCS, LCC)
Word	D, BDX, BDY, LD, LT (current value), LC (current value), SD and IX

(NOTE 1) Destination specifies an OM for storing data. OM (X) for input cannot be used.
(NOTE 2) When using LT and LC in the commands below, it functions as a bit OM.
LD, LDN, AND, ANDN, OR, ORN, LDP, LDNP, LDPI, LDNPI, ANDP, ANDNP, ANDPI, ANDNPI, ORP, ORNP, ORPI, ORNPI, OUT and RST
Any other command not listed above can only be used as a word OM.

"WL designation" and "INDEX" are described below.
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.

(NOTE 1) 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.

(NOTE 2) Variables cannot be used in an LD program except for the CALC command.

11.2 LD Basic Commands

11.2.1 List of LD Basic Commands

The basic commands in the LD language are noted below. The abbreviations used in the table are defined as follows.

Sn : Source OM (OM used for operation)
Dn : Destination OM (OM that stores operation result)
* n in Sn and Dn: Constant
L : Label
N : Nesting

Basic command

Category	Command		Symbol	Processing	Step
Contact command	LD	S		Contact a	2
	LDN	S		Contact b	2
	AND	S		Contact a series connection	2
	ANDN	S		Contact b series connection	2
	OR	S		Contact a parallel connection	2
	ORN	S		Contact b parallel connection	2
	LDP	S		Rising edge trigger	2
	LDNP	S		Falling edge trigger	2
	ANDP	S		Rising edge trigger	2
	ANDNP	S		Falling edge trigger	2
	ORP	S		Rising edge trigger	2
	ORNP	S		Falling edge trigger	2
	LDPI	S		Rising edge NOT trigger	2
	LDNPI	S		Falling edge NOT trigger	2
	ANDPI	S		Rising edge NOT trigger	2
	ANDNPI	S		Falling edge NOT trigger	2
	ORPI	S		Rising edge NOT trigger	2
	ORNPI	S		Falling edge NOT trigger	2
Connection command	AND-BLK	*1	—	AND used between blocks	1
	OR-BLK	*1	—	OR used between blocks	1
	M-PUSH	*1	—	Record operation result	1
	M-READ	*1	—	Read operation result	1
	M-POP	*1	—	Extract operation result	1
Output command	OUT	D		Coil output	2
	OUT	LT set value		Timer output	3
	OUT	LC set value		Counter output	3
Output command	SET	D	SET D	Bit set	2
	RST	D	RET D	OM reset	2
	PLS	D	PLS D	Pulse output	2
	PLSN	D	PLSN D	Pulse OFF output	2
	SFT	D	SFT D	Bit shift	2

Category	Command		Symbol	Processing	Step
Program control command	MC	N D	MC N D	Start program control	3
	MCR	N	MCR N	Cancel program control	2
End command	END	*2	-	Program end	1
	ENDS		ENDS	Program processing end	1

*1 This command does not appear on the connection in the ladder diagram. User input not possible.
*2 This command is automatically added to the end of the program. User input not possible.

11.2.2 LD Basic Command Description

11.2.2.1 Contact commands

LD, LDN, AND, ANDN, OR and ORN

Function

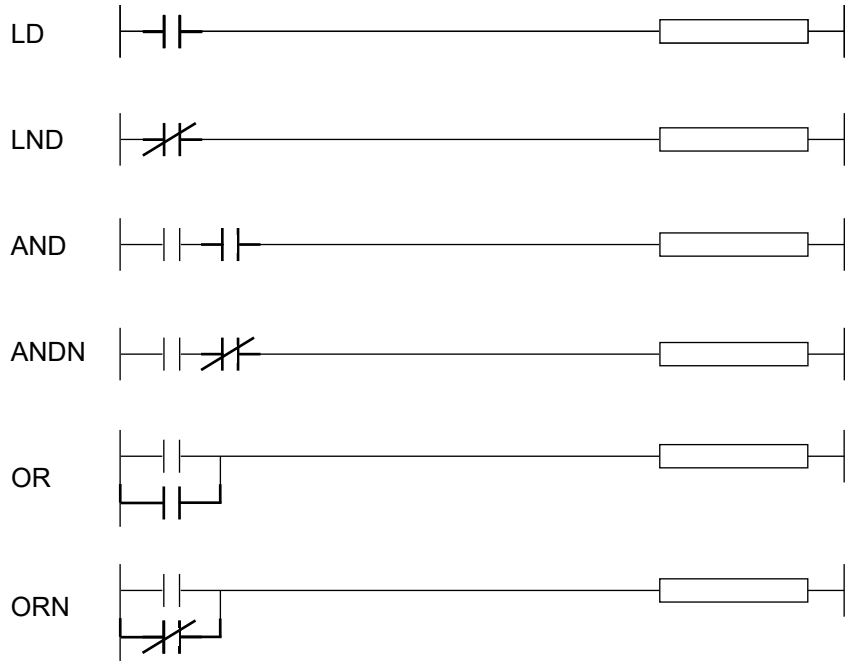
Command	Description
LD, LDN	Starts the operation.
AND, ANDN	This type of command is a series connection command. A logical conjunction (AND) is applied to the previous logic operation results up until then, and the value produced becomes the operation result.
OR, ORN	This type of command is a parallel connection command. A logical disjunction (OR) is applied to the previous operation results up until then, and the value produced becomes the operation result.

For LD, AND and OR, the command is valid when the contact is ON, and for LDN, ANDN and ORN, the command is valid when the contact is OFF.

Usable OM

Bit									Word								Constant		Label	Nesting	WL designation	INDEX
X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
○	○	○	○	○	○	○	○	○														○

Circuit diagram



The commands: LD, LDN, AND, ANDN, OR and ORN are not displayed as commands in the ladder program. The commands are distinguished automatically by the position used for the contact symbol.

LDP, LDNP, LDPI, LDNPI, ANDP, ANDNP, ANDPI, ANDNPI, ORP, ORNP, ORPI, ORNPI

Function

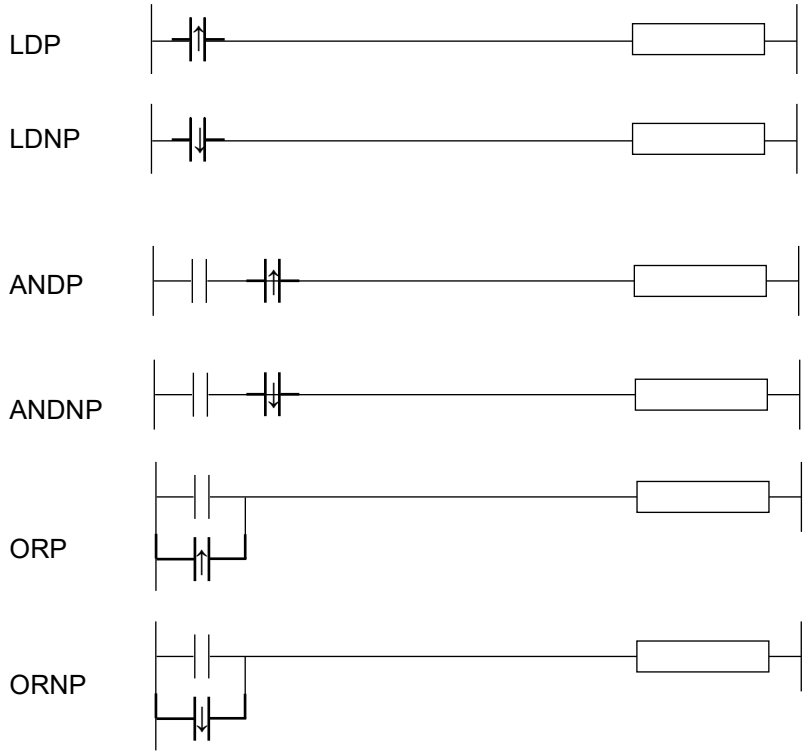
Command	Description
LDP, LDNP, LDPI, LDNPI	Starts the operation.
ANDP, ANDNP, ANDPI, ANDNPI	This type of command is a series connection command. A logical conjunction (AND) is applied to the previous operation results up until then, and the value produced becomes the operation result.
ORP, ORNP, ORPI, ORNPI	This type of command is a parallel connection command. A logical disjunction (OR) is applied to the previous operation results up until then, and the value produced becomes the operation result.

For LDP, ANDP, ORP, LDPI, ANDPI and ORPI, the command is valid only when contact changes from OFF → ON, and for LDNP, ANDNP, ORNP, LDNPI, ANDNPI and ORNPI, the command is valid only when contact changes from ON → OFF.
After executing a command, RUN changes to STOP and the command will not execute even if RUN is performed again.

Usable OM

Bit									Word								Constant		Label	Nesting	WL designation	INDEX
X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
○	○	○	○	○	○	○	○	○														○

Circuit diagram



The commands: LDP, LDNP, LDPI, LDNPI, ANDP, ANDNP, ANDPI, ANDNPI, ORP, ORNP, ORPI and ORNPI are not displayed as commands in the ladder program. The commands are distinguished automatically by the position used for the contact symbol.

11.2.2.2 Connection commands

AND-BLK, OR-BLK

Function

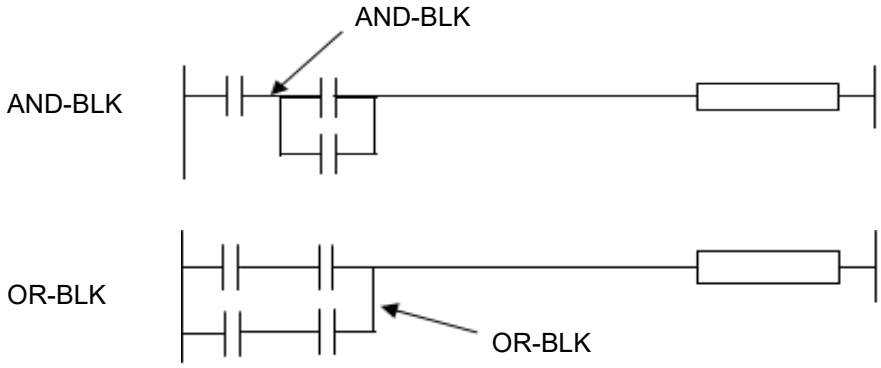
Command	Description
AND-BLK	Performs an operation beforehand for the block. A logical conjunction (AND) is applied to the previous operation results up until then, and the value produced becomes the operation result.
OR-BLK	Performs an operation beforehand for the block. A logical disjunction (OR) is applied to the previous operation results up until then, and the value produced becomes the operation result.

This command does not require any particular input, because it is added automatically through sequence pattern analysis.

Usable OM

Bit									Word								Constant		Label	Nesting	WL designation	INDEX
X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		

Circuit diagram



M-PUSH, M-READ and M-POP

Function

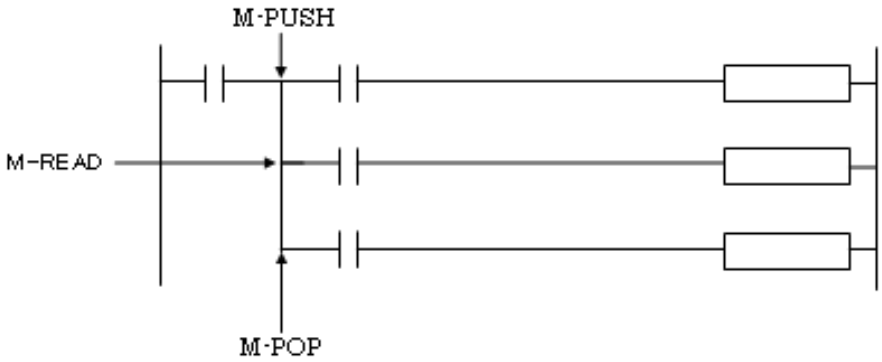
Command	Description
M-PUSH	Records the operation result immediately prior to the command.
M-READ	Reads the operation result recorded by the M-PUSH command. This command can be executed any number of times.
M-POP	Reads the operation result recorded by the M-PUSH command, and clears the operation result recorded with the M-PUSH command.

This command does not require any particular input, because it is added automatically through sequence pattern analysis.

Usable OM

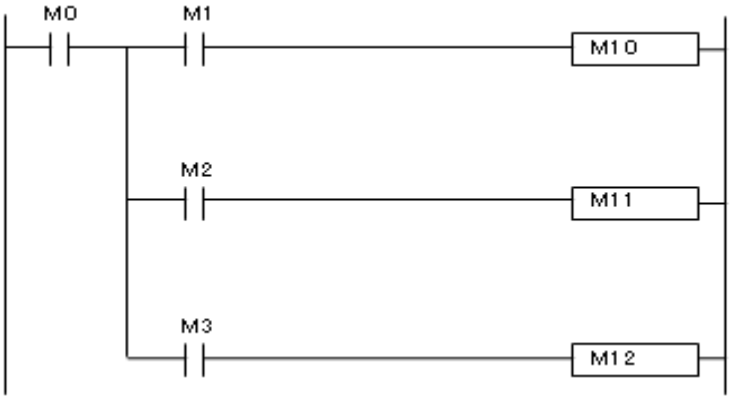
Bit									Word								Constant		Label	Nesting	WL designation	INDEX
X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
○	○	○	○	○	○	○	○	○														○

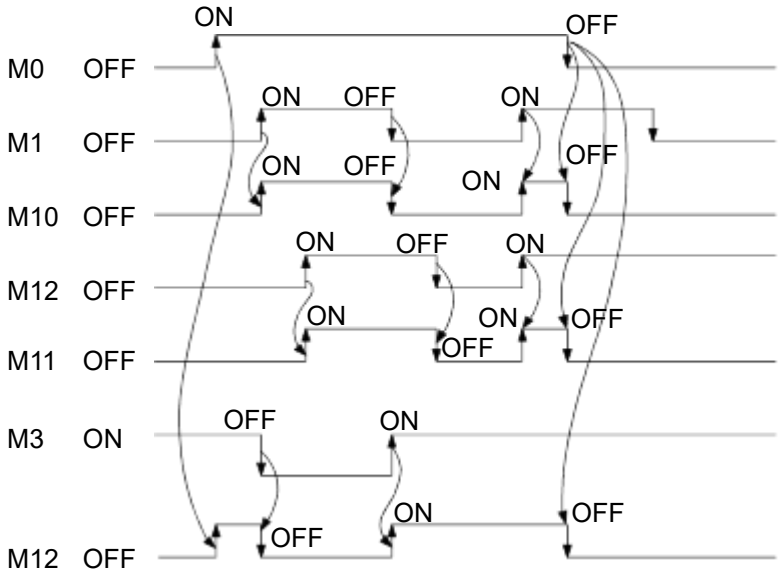
Circuit diagram



11

Time chart
A time chart for the circuit example is shown below.





11.2.2.3 Output commands

OUT

Function

Command	Description
OUT (Bit OM)	Outputs the operation results up to the OUT command to the specified OM.
OUT (LT)	Local timer value counts up when turned ON.
OUT (LC)	Local counter value counts up when turned ON.

(NOTE 1) When LT/LC is specified for the OUT command, the set value needs to be specified next.
(NOTE 2) Only a decimal constant, or D and LD can be specified for the set value.
(NOTE 3) When a decimal constant is specified for the set value, a value between 0 and 32,767 can be specified.
(NOTE 4) Execute the local counter reset using the RST command.
(NOTE 5) When using a standard control program, enter a value in 10 ms units for the LT. (The display is in ms units. When “1” is entered as the set value, the display shows “10 msec” and operates accordingly.)
When using a high speed control program, enter a value in 1 ms units for the LT. (The display is in ms units. When “1” is entered as the set value, the display shows “1 msec” and operates accordingly.)
(NOTE 6) Index addressing will not be carried out for LT/LC.
(NOTE 7) The time will not expire when the set value for the the local timer is 0 or less.
(NOTE 8) If the set value for the local counter is 0 or less, one input turns ON the contact and coil while the count value stays at 0.

Usable OM

		Bit									Word								Constant		Label	Nesting	WL designation	INDEX
		X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
Bit OM			○	○		○	○	○																○
Timer	OM							○																
	Set value									○			○					○						
Counter	OM								○															
	Set value	○	○		○	○	○			○			○					○						

Circuit diagram

Difference between OUT command and SET command

The OM that turns ON with the OUT command, turns OFF when the startup conditions are OFF. On the other hand, the OM that turns ON with the SET command, stays ON even when the startup conditions are OFF. The OM that turns ON with the SET command, turns OFF with the RST command.

SET

Function

Command	Description
SET	Turns ON the specified OM value and maintains that status. When turned OFF, the status of the OM does not change.

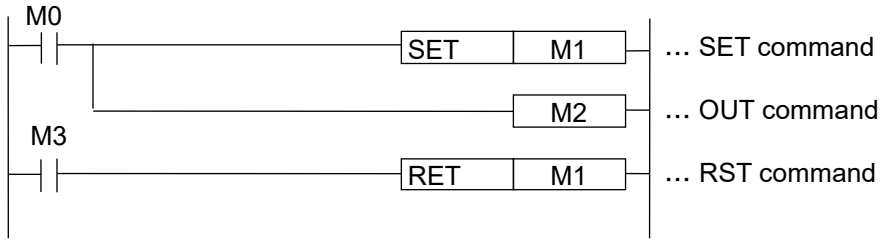
Usable OM

Bit									Word								Constant		Label	Nesting	WL designation	INDEX
X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
	○	○		○	○	○																○

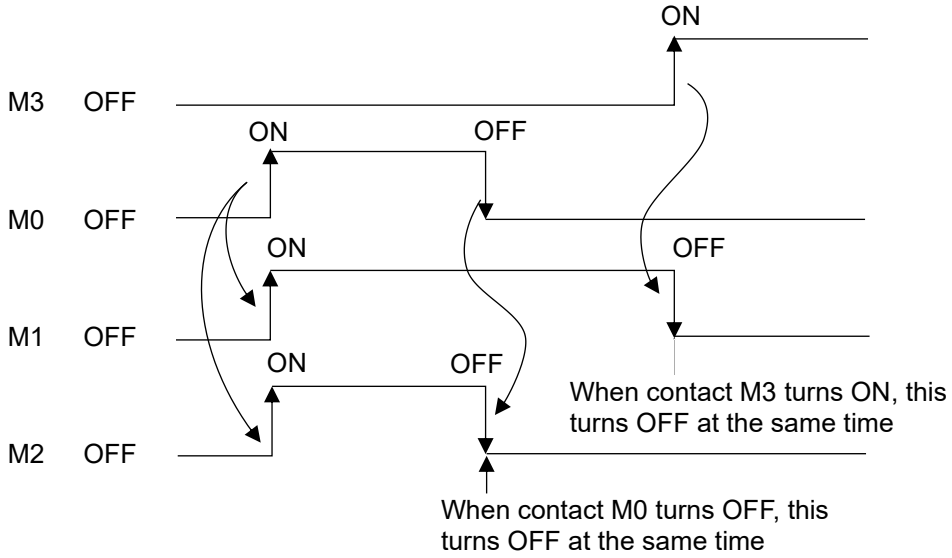
Circuit diagram

The program example for the SET command, OUT command and RST command as well as the ON/OFF timing for the OM are shown below.

Program example



Timing chart



11

RST

Function

Command	Description
RST	Clears or resets the specified OM value. When turned OFF, the status of the OM does not change. Bit OM: Turns OFF coil / contact. Word OM: Changes current value to 0 (zero). LT/LC: Changes current value to 0 (zero) and turns OFF coil / contact.

Usable OM

Bit									Word								Constant		Label	Nesting	WL designation	INDEX
X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
	○	○		○	○	○			○	○	○	○	○	○	○	○						○

Circuit diagram

PLS and PLSN

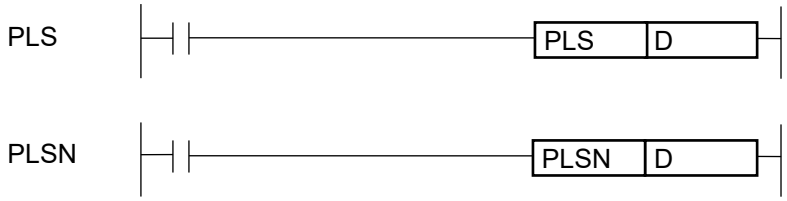
Function

Command	Description
PLS	When the operation result changes from OFF to ON, the specified OM turns ON for only 1 scan.
PLSN	When the operation result changes from ON to OFF, the specified OM turns ON for only 1 scan.

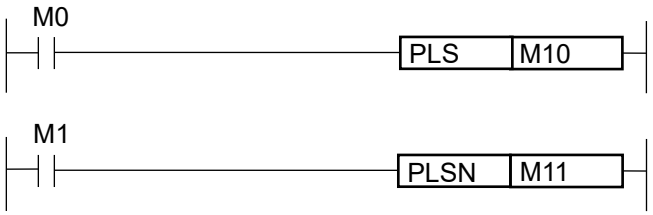
Usable OM

Bit									Word								Constant		Label	Nesting	WL designation	INDEX
X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
	○	○		○	○	○																○

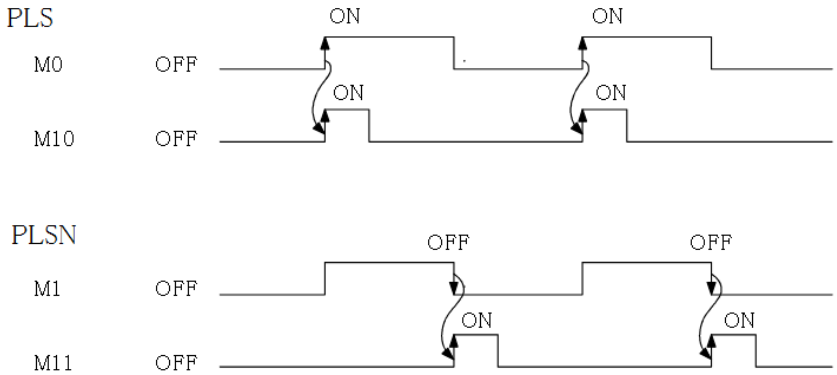
Circuit diagram



Program example



Time chart



11

SFT(P)

Function

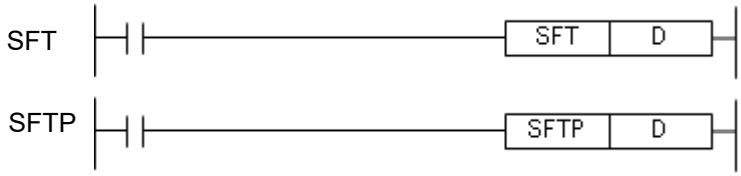
Command	Description
SFT	Shifts the bit OM status to the next OM number bit when the operation result is ON. The ON/OFF status for the OM (that is lower than the OM specified by D) shifts, and changes the lower OM to OFF.
SFTP	Shifts the bit OM status to the next OM number bit when the operation result changes from OFF to ON. The ON/OFF status for the OM (that is lower than the OM specified by D) shifts, and changes the lower OM to OFF.

(NOTE 1) Use the SET command to turn ON the OM at the beginning that shifts.
(NOTE 2) Program from a larger OM number when using SFT and SFTP successively.

Usable OM

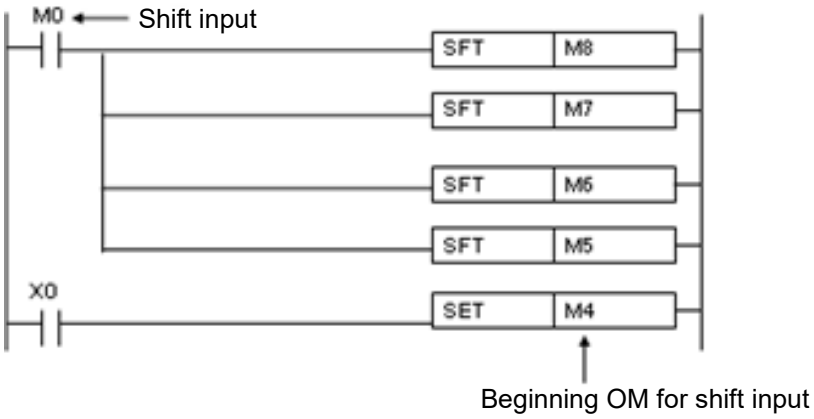
Bit									Word								Constant		Label	Nesting	WL designation	INDEX
X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
	○	○		○	○	○																○

Circuit diagram



Program example

Ex) Values for M5 through M8 shift when the SFT command is issued.



Operation of a program example is shown below.

	M9	M8	M7	M6	M5	M4	M3	M2	
1	0	0	0	0	0	1	1	0	X0 ON
							0		
2	0	0	0	0	1	0	1	0	1st shift input
							0		
3	0	0	0	1	0	0	1	0	2nd shift input
4	0	0	0	1	0	1	1	0	X0 ON
							0		
5	0	0	1	0	1	0	1	0	After 3rd shift input
							0		
6	0	1	0	1	0	0	1	0	After 4th shift input
							0		
7	0	0	1	0	0	0	1	0	After 5th shift input

11.2.2.4 Program control commands

MC

Function

Command	Description
MC	When the contact is ON, the operations are carried out following the commands between the MC and MCR. The scan time does not become shorter because the scan between the MC and MCR commands is executed even if the MC command is OFF. In addition, the operation result between the MC and MCR is as follows when the MC command is OFF. Local timer: Coil and contact both turn OFF when the count value turns 0. OMs during OUT command: All OMs turn OFF. OMs being used by other commands: Status is maintained.

(NOTE 1) Nesting is possible for up to 8 objects. In this situation, specify from the lower number for nesting N.
Note that the proper operations will not be carried out when the numbers are not in order.

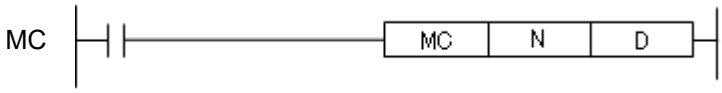
(NOTE 2) An MC command can use the same nesting (N) number any number of times by changing the OM in D.

(NOTE 3) When the MC command is ON, the OM coil specified for D turns ON. In addition, if the same OM is used with the OUT command, etc., then it becomes a double coil. Therefore, do not use the OM specified for D while another command is being carried out.

Usable OM

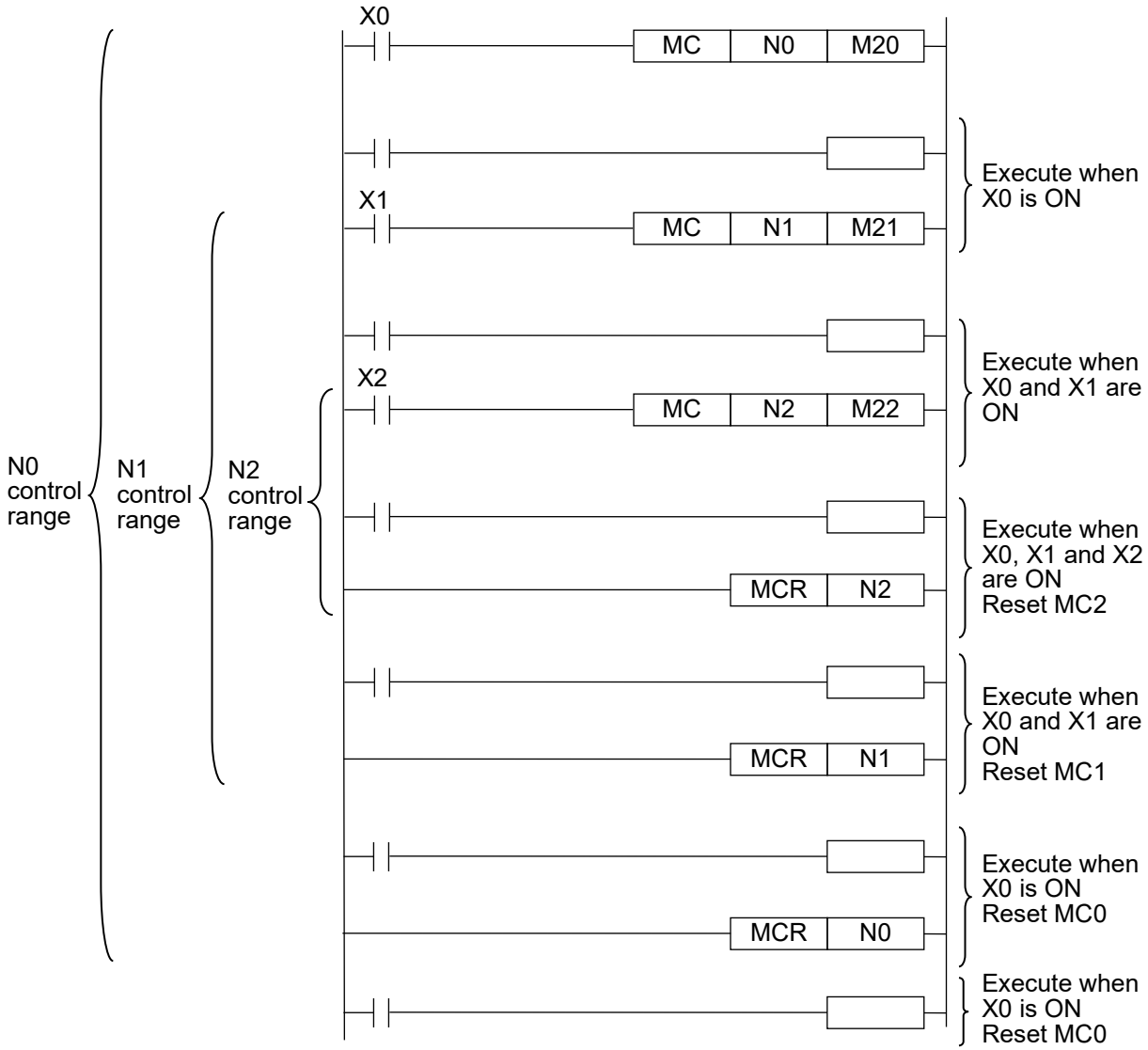
	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
N																					○		
D		○	○		○	○	○																

Circuit diagram



11

Program example
A control example is shown below when using an MC command.



Error code
- When the number of MC and MCR commands do not match. (Error code 22)
- When the MCR command is executed before the MC command is executed. (Error code 22)

MCR

Function

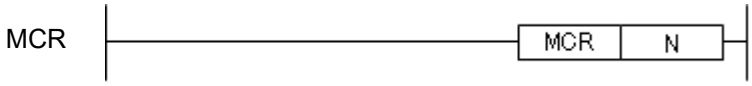
Command	Description
MCR	Indicates the end of the MC control range with the MC release command.

(NOTE 1) Do not put a contact command before the MCR command.
(NOTE 2) Use higher numbers for nesting N when nesting MC and MCR commands.
(NOTE 3) All MC commands can be reset with the lowest number N when the MCR commands are set up together in one location. (Refer to the program example)

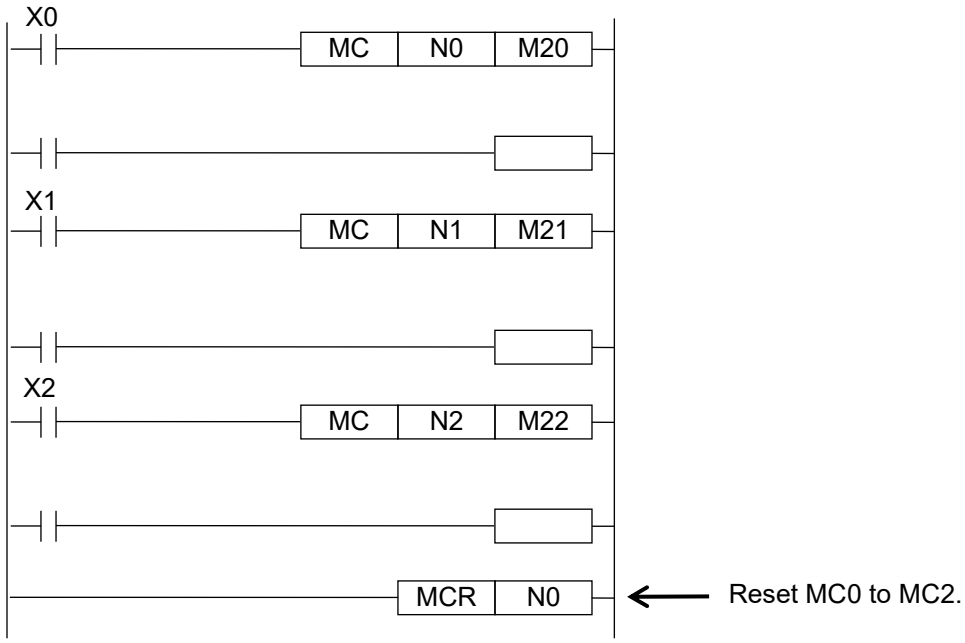
Usable OM

Bit									Word								Constant		Label	Nesting	WL designation	INDEX
X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
																				○		

Circuit diagram



Program example



Error code

- When the number of MC and MCR commands do not match. (Error code 22)
- When the MCR command is executed before the MC command is executed. (Error code 22)

11.2.2.5 End commands

END

Function

Command	Description
END	Indicates the end of the program.

(NOTE) This command does not require any particular input, because it is added automatically.

Usable OM

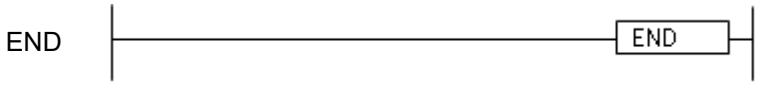
Bit									Word								Constant		Label	Nesting	WL designation	INDEX
X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		

Error code

An error is triggered when a command is executed in the following situations.

- When the END command is executed after the CALL (P) command is executed but before the RET command is executed. (Error code 22)
- When the END command is executed after the FOR command is executed but before the NEXT command is executed. (Error code 22)

Circuit diagram



ENDS

Function

Command	Description
ENDS	Ends the main routine processing. Use this command when defining a subroutine program, or when branching the processing.

(NOTE) The user cannot put conditions onto the ENDS command.

Usable OM

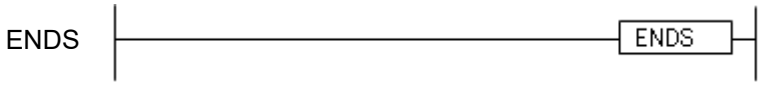
Bit									Word								Constant		Label	Nesting	WL designation	INDEX
X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		

Error code

An error is triggered when a command is executed in the following situations.

- When the ENDS command is executed after the CALL (P) command is executed but before the RET command is executed. (Error code 22)
- When the ENDS command is executed after the FOR command is executed but before the NEXT command is executed. (Error code 22)
- When the ENDS command is executed before the MCR command is executed. (Error code 22)

Circuit diagram



11.3 LD Application Commands

11.3.1 List of LD Application Commands

The following shows the application commands available in the LD language. The abbreviations used in the table are defined as follows.

Sn : Source OM (OM used for operation)
Dn : Destination OM (OM that stores operation result)
n : Constant
L : Label
N : Nesting

Application command

Category	Command		Symbol	Processing	Step
Data comparison	S1 = S2		S1 = S2	Conduction when S1 = S2	3
	S1 >= S2		S1 >= S2	Conduction when S1 ≥ S2	3
	S1 <= S2		S1 <= S2	Conduction when S1≤ S2	3
	S1 > S2		S1 > S2	Conduction when S1 > S2	3
	S1 < S2		S1 < S2	Conduction when S1 < S2	3
	S1 <> S2		S1 <> S2	Conduction when S1< >S2	3
Arithmetic operation	+	S D	+ S D	Store S + D (BIN data) in D	3
	+	S1 S2 D	+ S1 S2 D	Store S1 + S2 (BIN data) in D	4
	-	S D	- S D	Store D – S (BIN data) in D	3
	-	S1 S2 D	- S1 S2 D	Store S1 – S2 (BIN data) in D	4
	*	S1 S2 D	* S1 S2 D	Store S1 × S2 (BIN data) in D	4
	/	S1 S2 D	/ S1 S2 D	Store S1 / S2 (BIN data) in D	4
	B+	S D	B+ S D	Store S + D (BCD data) in D	3
	B+	S1 S2 D	B+ S1 S2 D	Store S1 + S2 (BCD data) in D	4
	B-	S D	B- S D	Store D – S (BCD data) in D	4
	B-	S1 S2 D	B- S1 S2 D	Store S1 – S2 (BCD data) in D	4
	B*	S1 S2 D	B* S1 S2 D	Store S1 × S2 (BCD data) in D	4
	B/	S1 S2 D	B/ S1 S2 D	Store S1 / S2 (BCD data) in D	4
	INC	D	INC D	Increment	2
	BINC	D	BINC D	BCD increment	2
	DEC	D	DEC D	Decrease	2
	BDEC	D	BDEC D	BCD decrease	2
BCD-BIN conversion	BCD	S D	BCD S D	Convert BCD	3
	BIN	S D	BIN S D	Convert BIN	3

Category	Command		Symbol	Processing	Step
Transfer	MOV	S D	MOV S D	Transfer S to D	3
	MOVN	S D	MOVN S D	Invert S at each bit and transfer to D	3
	MCPY	S D n	MCPY S D n	Transfer n points starting from S to n points starting from D	4
	MSET	S D n	MSET S D n	Transfer S to n points starting from D	4
	XCHG	D1 D2	XCHG D1 D2	Exchange bit data for D1 and D2	3
Branch	JE	L	JE L	Jump to L when input conditions are met	2
	JMP	L	JMP L	Jump to L without any conditions	2
	CALL	L	CALL L	Execute subroutine specified in L	2
	RET		RET	Return from subroutine	1
Logic operation	LAND	S D	LAND S D	Store AND for S and D in D	3
	LAND	S1 S2 D	LAND S1 S2 D	Store AND for S1 and S2 in D	4
	LOR	S D	LOR S D	Store OR for S and D in D	3
	LOR	S1 S2 D	LOR S1 S2 D	Store OR for S1 and S2 in D	4
	LXOR	S D	LXOR S D	Store XOR for S and D in D	3
	LXOR	S1 S2 D	LXOR S1 S2 D	Store XOR for S1 and S2 in D	4
	LXNR	S D	LXNR S D	Store XNOR for S and D in D	3
	LXNR	S1 S2 D	LXNR S1 S2 D	Store XNOR for S1 and S2 in D	4
	NEG	D	NEG D	Invert sign	2
Rotation	ROR	D n	ROR D n	Rotate D n bits to the right, not including the carry flag	3
	RCR	D n	RCR D n	Rotate D n bits to the right, including the carry flag	3
	ROL	D n	ROL D n	Rotate D n bits to the left, not including the carry flag	3
	RCL	D n	RCL D n	Rotate D n bits to the left, including the carry flag	3
Shift	SHR	D n	SHR D n	Shift D n bits to the right	3
	SHL	D n	SHL D n	Shift D n bits to the left	3
	BSHR	D n	BSHR D n	Shift n bits starting from D to the right 1 bit	3
	BSHL	D n	BSHL D n	Shift n bits starting from D to the left 1 bit	3
	WSHR	D n	WSHR D n	Shift n points starting from D to the right 1 point	3
	WSHL	D n	WSHL D n	Shift n points starting from D to the left 1 point	3

Category	Command		Symbol	Processing	Step
Data processing	SUM	S D	SUM S D	Store the ON bit sum in D from 16-bit data for S	3
	DECO	S D n	DECO S D n	Decode the low-order n bit for S and store in the 2^n bit from D	4
	ENCO	S D n	ENCO S D n	Encode the 2^n bit from S and store it in D	4
	BSET	D n	BSET D n	Set the n bit for D	3
	BRST	D n	BRST D n	Reset the n bit for D	3
	DDV	S D n	DDV S D n	Store the low-order n digit for S to the low-order 4-bit for n points	4
	DCV	S D n	DCV S D n	Store the low-order 4-bit data for n points from S in D	4
	AVG	S D n1 n2	AVG S D n1 n2	Store the average of the S values (for n1 to maintain the n2 buffer) in D	5
FIFO	FIFW	S D	FIFW S D	Write to FIFO table	3
	FIFR	D1 D2	FIFR D1 D2	Read from FIFO table	3
Loop	FOR	n	FOR n		2
	NEXT		NEXT	Execute from FOR to NEXT n times	1
	BREAK		BREAK	Execute the next step for NEXT	1
Carry flag	STC		STC	Set the carry flag contact	1
	CLC		CLC	Reset the carry flag contact	1
Date/Time	DTRD	D	DTRD D	Store the date and time in 7 words from D	5
Palletize	PALETTE	S D	PALETTE S D	Calculate coordinates for holding workpiece from S, and store in D	6
User function	CALC		CALC	Perform operation noted by user in CALC command	Depends on specified process

11.3.2 LD Application Command Description

11.3.2.1 Data comparison

=, >, >=, <, <= and <>

Function

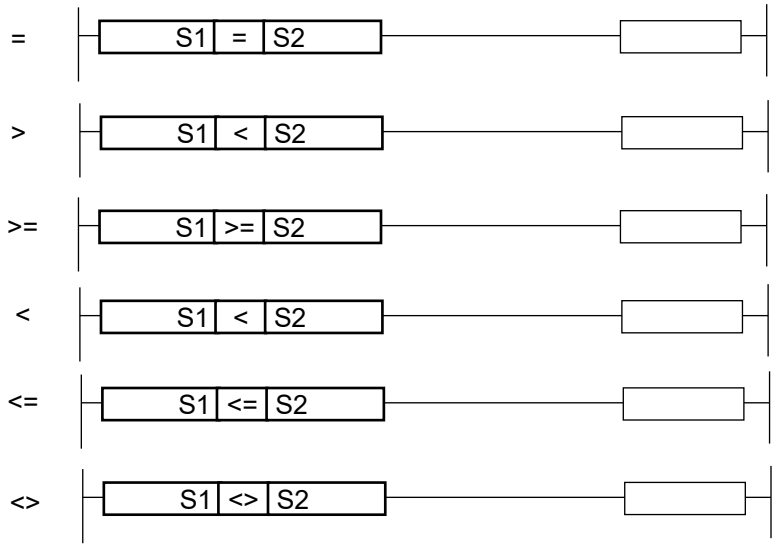
Command	Description
=, >, >=, <, <= and <>	Compares the OM values. = : Conditions are met when S1 = S2 <> : Conditions are met when S1 ≠ S2 > : Conditions are met when S1 > S2 <= : Conditions are met when S1 ≦ S2 < : Conditions are met when S1 < S2 >= : Conditions are met when S1 ≧ S2 The operation result is true when the conditions are met. The items being compared for word OMs can be specified using either 16-bit or 32-bit. When using a bit OM, digit-specified access is required. The type is automatically converted to a large type when the type being compared is different. (16-bit < 32-bit)

Usable OM

	Bit (*)									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S1	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
S2	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○

* When using a bit OM for S1and S2, digit-specified access is required.

Circuit diagram



11.3.2.2 Arithmetic operation

+(P)(2)

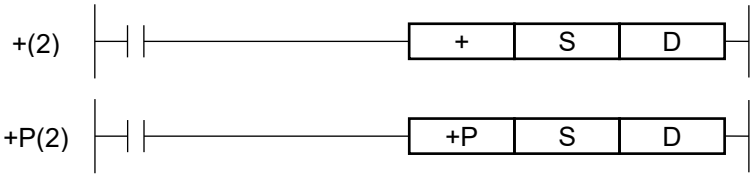
Function

Command	Description
+(2)	Carries out an addition operation for the BIN data specified in D and the BIN data specified in S (executes D+S), and the result is stored in the OM specified in D. The range between -32768 and 32767 (BIN 16-bit) can be specified in S and D. The positive and negative status of the data is assessed with the most significant bit (b15). 0: Positive, 1: Negative The carry flag does not turn ON when an underflow occurs at the 0 bit. The carry flag does not turn ON when an overflow occurs at the 15 bit.
+P(2)	Executes +(2) when the operation result changes from OFF → ON.

Usable OM

	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
D		○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

Circuit diagram



+(P)(3)

Function

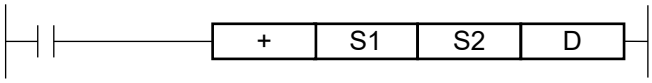
Command	Description
+(3)	Carries out an addition operation for the BIN data specified in S1 and the BIN data specified in S2 (executes S1+S2), and the result is stored in the OM specified in D. The range between -32768 and 32767 (BIN 16-bit) can be specified in S1 and S2. The positive and negative status of the data is assessed with the most significant bit (b15). 0: Positive, 1: Negative The carry flag does not turn ON when an underflow occurs at the 0 bit. The carry flag does not turn ON when an overflow occurs at the 15 bit.
+P(3)	Executes +(3) when the operation result changes from OFF → ON.

Usable OM

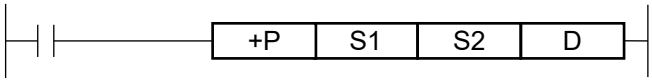
	Bit									Word								Constant		Label	Nesting	WL designation(*)	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S1	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
S2	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
D		○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

Circuit diagram

+(3)



+P(3)



11

–(P)(2)

Function

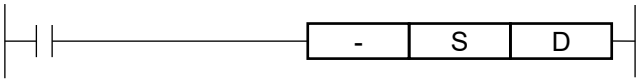
Command	Description
-(2)	Carries out a subtraction operation for the BIN data specified in D and the BIN data specified in S (executes D-S), and the result is stored in the OM specified in D. The range between -32768 and 32767 (BIN 16-bit) can be specified in S and D. The positive and negative status of the data is assessed with the most significant bit (b15). 0: Positive, 1: Negative The carry flag does not turn ON when an underflow occurs at the 0 bit. The carry flag does not turn ON when an overflow occurs at the 15 bit.
-P(2)	Executes -(2) when the operation result changes from OFF → ON.

Usable OM

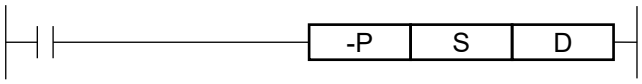
	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
D		○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

Circuit diagram

-(2)



-P(2)



–(P)(3)

Function

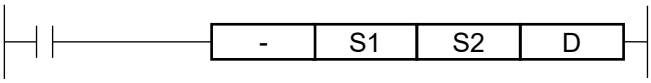
Command	Description
-(3)	Carries out a subtraction operation for the BIN data specified in S1 and the BIN data specified in S2 (executes S1-S2), and the result is stored in the OM specified in D. The range between -32768 and 32767 (BIN 16-bit) can be specified in S1 and S2. The positive and negative status of the data is assessed with the most significant bit (b15). 0: Positive, 1: Negative The carry flag does not turn ON when an underflow occurs at the 0 bit. The carry flag does not turn ON when an overflow occurs at the 15 bit.
-P(3)	Executes -(3) when the operation result changes from OFF → ON.

Usable OM

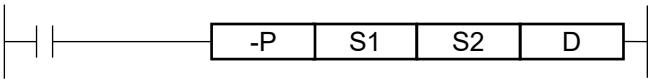
	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S1	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
S2	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
D		○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

Circuit diagram

-(3)



-P(3)



11

*(P)

Function

Command	Description
*	Carries out a multiplication operation for the BIN data specified in S1 and the BIN data specified in S2 (executes S1×S2), and the result is stored in the OM specified in D. When D is a bit OM, it is specified with the least significant bit. The range between -32768 and 32767 (BIN 16-bit) can be specified in S1 and S2. (It is handled as a signed integer.)
*P	Executes * when the operation result changes from OFF → ON.

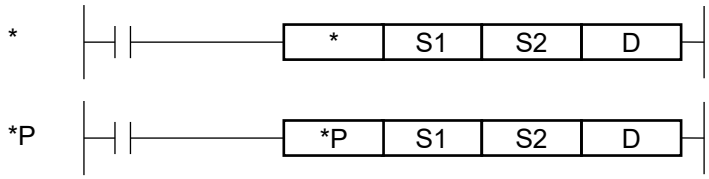
(NOTE) S is specified in 16-bit and D is always specified in 32-bit. (Operations between 32-bit operands are not possible.)

Usable OM

	Bit									Word								Constant		Label	Nesting	WL designation(*)	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S1	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
S2	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
D		○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

* When specifying a bit OM for S1 and S2, multiple bit access is possible (4-bit, 8-bit, 12-bit and 16-bit). When specifying a bit OM for D, multiple bit access is possible (20-bit, 24-bit, 28-bit and 32-bit). When specifying a word OM for S1and S2, only 16-bit access is possible. When specifying a word OM for D, only 32-bit access is possible.

Circuit diagram



/(P)

Function

Command	Description
/	Carries out a division operation for the BIN data specified in S1 and the BIN data specified in S2 (executes S1/S2), and the result is stored in the OM specified in D. The result uses 32-bit for a word OM and stores the quotient and remainder. 16-bit is used for a bit OM and only the quotient is stored. Quotient: Stored in the low-order 16-bit. Remainder: Stored in the high-order 16-bit (only for a word OM). The range between -32768 and 32767 (BIN 16-bit) can be specified in S1 and S2. (It is handled as a signed integer.)
/P	Executes / when the operation result changes from OFF → ON.

The operation result is stored as follows.

S1		S2		D	D+1
7700	/	1234	→	6	296
				Quotient	Remainder

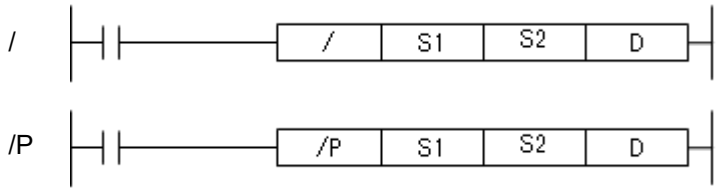
(NOTE) S is specified in 16-bit and D is always specified in 32-bit. (Operations between 32-bit operands are not possible.)

Usable OM

	Bit									Word								Constant		Label	Nesting	WL designation(*)	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S1	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
S2	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
D		○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

* When specifying a bit OM for S1 and S2, multiple bit access is possible (4-bit, 8-bit, 12-bit and 16-bit). When specifying a bit OM for D, multiple bit access is possible (20-bit, 24-bit, 28-bit and 32-bit). When specifying a word OM for S1and S2, only 16-bit access is possible. When specifying a word OM for D, only 32-bit access is possible.

Circuit diagram



Error code

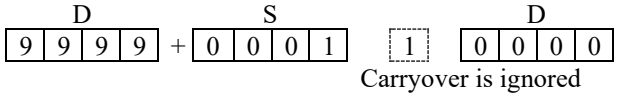
When 0 is stored to S2. (Error code: 11)

B+(P)(2)

Function

Command	Description
B+(2)	Carries out an addition operation for the BCD data specified in D and the BCD data specified in S (executes D+S), and the result is stored in the OM specified in D. The range between 0 and 9999 (BCD 4-digit) can be specified in S and D. Even if the result exceeds 9999, the carry flag does not turn ON, and the carry operation is ignored.
B+P	Executes B+ when the operation result changes from OFF → ON.

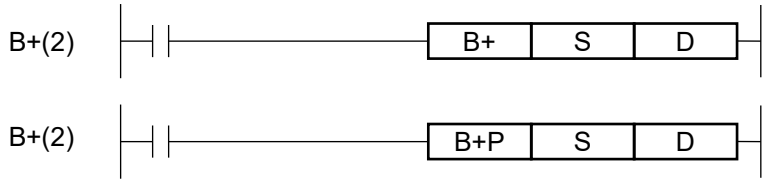
When the operation result exceeds 9999 (16-bit data) or 99999999 (32-bit data), the carryover is ignored.



Usable OM

	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
D		○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

Circuit diagram



Error code

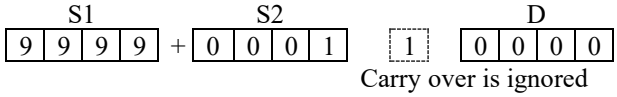
When S and D are not BCD data (when values other than 0 through 9 are stored for each digit). (Error code 10)

B+(P)(3)

Function

Command	Description
B+(3)	Carries out an addition operation for the BCD data specified in S1 and the BCD data specified in S2 (executes S1+S2), and the result is stored in the OM specified in D. The range between 0 and 9999 (BCD 4-digit) can be specified in S and D. Even if the result exceeds 9999, the carry flag does not turn ON, and the carry operation is ignored.
B+P(3)	Executes B+(3) when the operation result changes from OFF → ON.

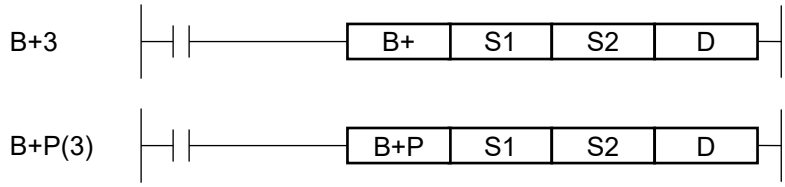
When the operation result exceeds 9999 (16-bit data) or 99999999 (32-bit data), the carryover is ignored.



Usable OM

	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S1	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
S2	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
D		○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

Circuit diagram



Error code

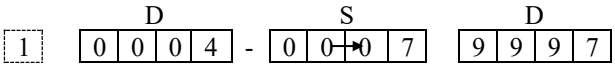
When S1 and S2 are not BCD data (when values other than 0 through 9 are stored for each digit). (Error code 10)

B-(P)(2)

Function

Command	Description
B-(2)	Carries out a subtraction operation for the BCD data specified in S and the BCD data specified in D (executes D-S), and the result is stored in the OM specified in D. The range between 0 and 9999 (BCD 4-digit) can be specified in S and D. The positive and negative status of the operation result needs to be assessed in the program.
B-P	Executes B-(2) when the logic operation result changes from OFF → ON.

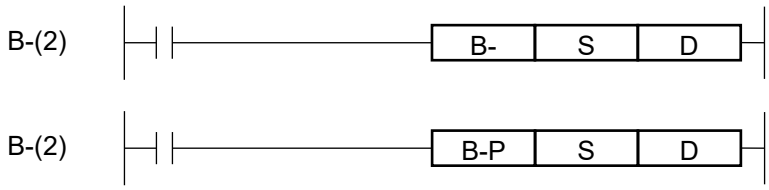
The logic operation result is as follows when D < S.



Usable OM

	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
D		○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

Circuit diagram



Error code

When S and D are not BCD data (when values other than 0 through 9 are stored for each digit).
(Error code 10)

B-(P)(3)

Function

Command	Description
B-(3)	Carries out a subtraction operation for the BCD data specified in S1 and the BCD data specified in S2 (executes S1-S2), and the result is stored in the OM specified in D. The range between 0 and 9999 (BCD 4-digit) can be specified in S and D. The positive and negative status of the operation result needs to be assessed in the program.
B-P(3)	Executes B-(3) when the operation result changes from OFF → ON.

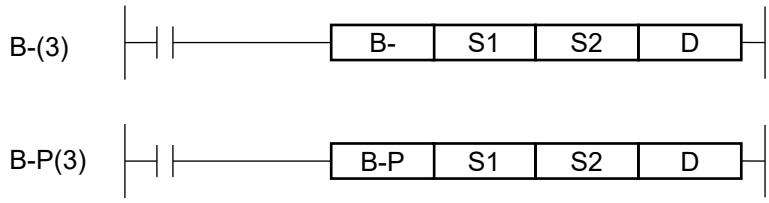
The operation result is as follows when S1 < S2.

	S1		S2		D
1	0 0 0 4	-	0 0 0 7		9 9 9 7

Usable OM

	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S1	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
S2	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
D		○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

Circuit diagram



Error code

When S and D are not BCD data (when values other than 0 through 9 are stored for each digit).
(Error code 10)

B*(P)

Function

Command	Description
B*	Carries out a multiplication operation for the BCD data for the OM specified in S1 and the BCD data for the OM specified in S2 (executes S1×S2), and the result is stored in the OM specified in D. The range between 0 and 9999 (BCD 4-digit) can be specified in S1 and S2.
B*P	Executes B* when the operation result changes from OFF → ON.

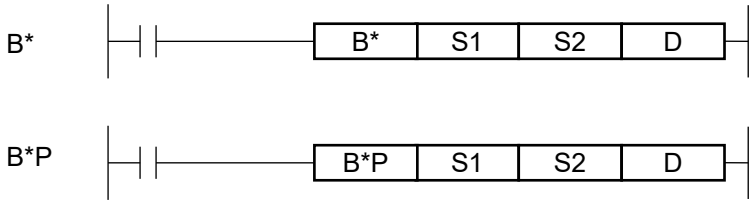
(NOTE) S is specified in 16-bit and D is always specified in 32-bit. (Operations between 32-bit operands are not possible.)

Usable OM

	Bit									Word								Constant		Label	Nesting	WL designation(*)	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S1	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
S2	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
D		○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

* When specifying a bit OM for S1 and S2, multiple bit access is possible (4-bit, 8-bit, 12-bit and 16-bit). When specifying a bit OM for D, multiple bit access is possible (20-bit, 24-bit, 28-bit and 32-bit). When specifying a word OM for S1and S2, only 16-bit access is possible. When specifying a word OM for D, only 32-bit access is possible.

Circuit diagram



Error code

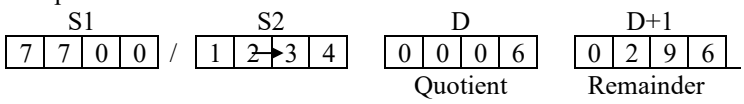
When S1 and S2 are not BCD data (when values other than 0 through 9 are stored for each digit). (Error code 10)

B/(P)

Function

Command	Description
B/	Carries out a division operation using the BCD data for the OM specified in S1 and the BCD data for the OM specified in S2, and the result is stored in the OM specified in D. The result uses 32-bit and stores the quotient and remainder. Quotient (BCD 4 digits): Stored in the low-order 16-bit. Remainder (BCD 4 digits): Stored in the high-order 16-bit. The remainder from the result is not stored when D is specified in a bit OM.
B/P	Executes B/ when the logic operation result changes from OFF → ON.

The operation result is stored as follows.



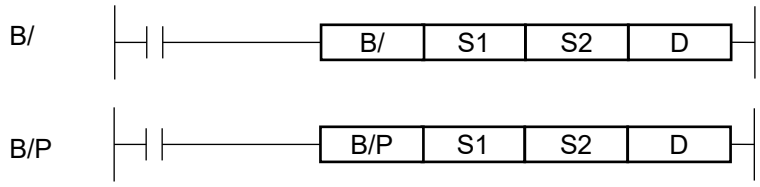
(NOTE) S is specified in 16-bit and D is always specified in 32-bit. (Operations between 32-bit operands are not possible.)

Usable OM

	Bit									Word								Constant		Label	Nesting	WL designation(*)	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S1	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
S2	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
D		○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

* When specifying a bit OM for S1 and S2, multiple bit access is possible (4-bit, 8-bit, 12-bit and 16-bit). When specifying a bit OM for D, multiple bit access is possible (20-bit, 24-bit, 28-bit and 32-bit). When specifying a word OM for S1and S2, only 16-bit access is possible. When specifying a word OM for D, only 32-bit access is possible.

Circuit diagram



Error code

An operation error is triggered when a command is executed in the following situations.

- An error is triggered when an operation is executed if S1 and S2 are not BCD data (when values other than 0 through 9 are stored for each digit).
- An error is triggered when an operation is executed if 0 is stored in S2.

INC(P)

Function

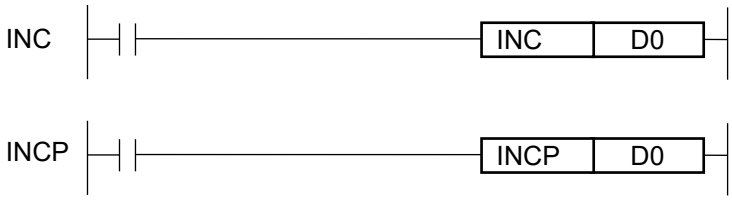
Command	Description
INC	Increases the OM value by an increment (+1) when the operation result is ON.
INCP	Increases the OM value by an increment (+1) when the operation result changes from OFF → ON.

(NOTE) Specify 32-bit access when handling a 32-bit value.

Usable OM

Bit									Word								Constant		Label	Nesting	WL designation	INDEX
X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
	○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

Circuit diagram



BINC(P)

Function

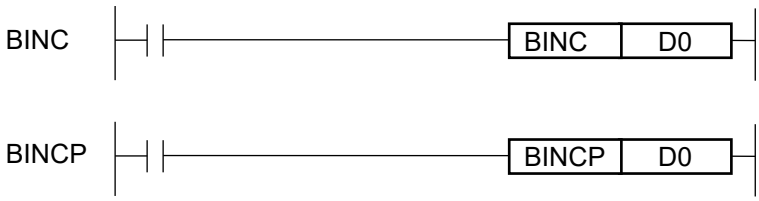
Command	Description
BINC	Increases the OM value by an increment (+1) using the BCD value when the operation result is ON.
BINCP	Increases the OM value by an increment (+1) using the BCD value when the operation result changes from OFF → ON.

(NOTE) Specify 32-bit access when handling a 32-bit value.

Usable OM

Bit									Word								Constant		Label	Nesting	WL designation	INDEX
X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
	○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

Circuit diagram



DEC(P)

Function

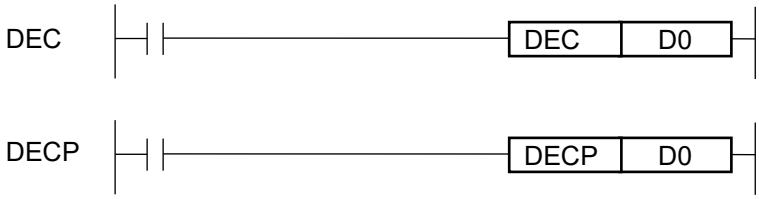
Command	Description
DEC	Decreases the OM value by the decrement (-1) when the operation result is ON.
DECP	Decreases the OM value by the decrement (-1) when the operation result changes from OFF → ON.

(NOTE) Specify 32-bit access when handling a 32-bit value.

Usable OM

Bit									Word								Constant		Label	Nesting	WL designation	INDEX
X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
	○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

Circuit diagram



BDEC(P)

Function

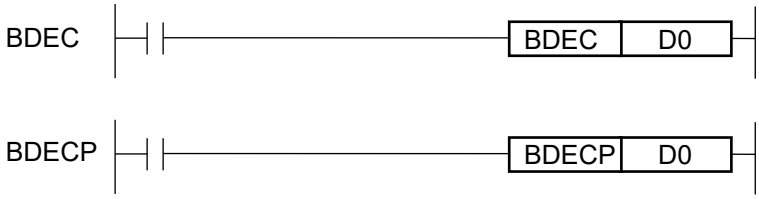
Command	Description
BDEC	Decreases the OM value by the decrement (-1) using the BCD value when the operation result is ON.
BDECP	Decreases the OM value by the decrement (-1) using the BCD value when the operation result changes from OFF → ON.

(NOTE) Specify 32-bit access when handling a 32-bit value.

Usable OM

Bit									Word								Constant		Label	Nesting	WL designation	INDEX
X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
	○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

Circuit diagram



11.3.2.3 BCD-BIN conversion

BCD(P)

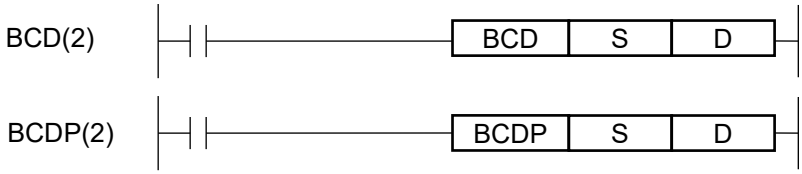
Function

Command	Description
BCD	Converts the BIN data (0 ~ 9999) for the OM specified in S to BCD, and transfers it to the OM specified in D.
BCDP	Converts BIN data (0 ~9999) for the OM specified in S to BCD and transfers it to the OM specified in D when the operation result changes from OFF → ON.

Usable OM

	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○					○	○
D		○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

Circuit diagram



Error code

When the value stored in S does not fall between 0 and 9999 (for 16-bit data) or 0 and 99999999 (for 32-bit data). (Error code 10)

BIN(P)

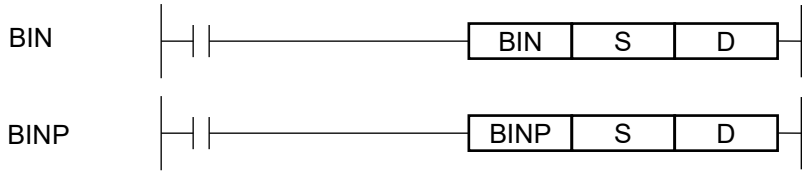
Function

Command	Description
BIN	Converts the BCD data (0 ~ 9999) for the OM specified in S to BIN, and transfers it to the OM specified in D.
BINP	Converts BCD data (0 ~ 9999) for the OM specified in S to BIN and transfers it to the OM specified in D when the operation result changes from OFF → ON.

Usable OM

	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○					○	○
D		○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

Circuit diagram



Error code

When S is not BCD data (when values other than 0 through 9 are stored for each digit). (Error code 10)

11.3.2.4 Transfer

MOV(P)

Function

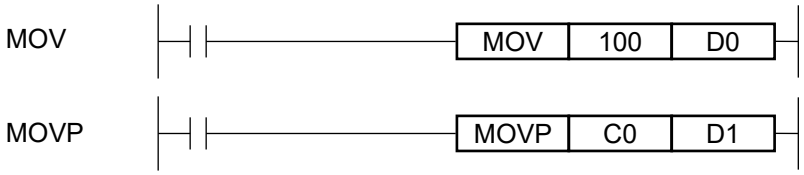
Command	Description
MOV	Transfers the value for the OM specified in S to the OM specified in D when the operation result is ON.
MOVP	Transfers the value for the OM specified in S to the OM specified in D when the operation result changes from OFF → ON.

(NOTE) Specify 32-bit access when handling a 32-bit value.

Usable OM

	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
D		○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

Circuit diagram

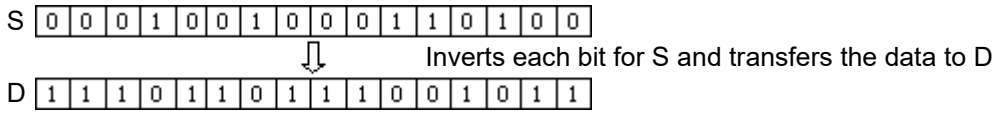


MOVN(P)

Function

Command	Description
MOVN	Inverts the S bit data for each bit, and transfers the result to the D side.
MOVNP	Inverts the bit data on the S side for each bit, and transfers the result to the D side when the operation result changes from OFF → ON.

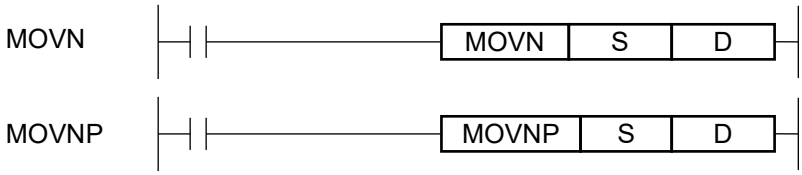
The operation result is as follows when MOVN (P) is executed.



Usable OM

	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
D		○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

Circuit diagram



MCPY(P)

Function

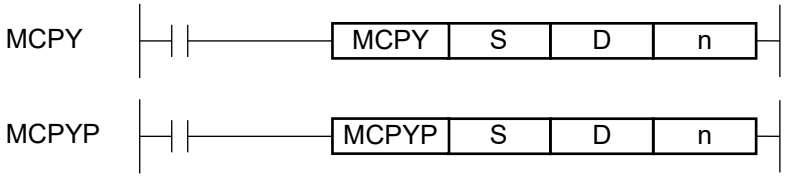
Command	Description
MCPY	Transfers all the n point data from the OM specified in S, to the n point from the OM specified in D. The user can transfer the data even when the source and destination OMs are the same. When transferring data to a low OM number, the data is transferred from S. When transferring to a high OM number, the data is transferred from S+(n-1). When specifying the bit OM digits for S and D, make sure to always use the same number of digits for S and D.
MCPYP	Transfers all the n point data from the OM specified in S to the n point for the OM specified in D when the operation result changes from OFF → ON.

Usable OM

	Bit									Word								Constant		Label	Nesting	WL designation(*)	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S	○	○	○	○	○	○	○			○	○	○	○	○	○	○						○	○
D		○	○		○	○	○			○	○	○	○	○	○	○						○	○
n																		○	○				

* Only multiple bit access (up to 16-bit) for bit OMs is possible.

Circuit diagram



Error code

An error is triggered when a command is executed in the following situations.

- When the range for moving or transferring data exceeds the corresponding OM. (Error code 23)

MSET(P)

Function

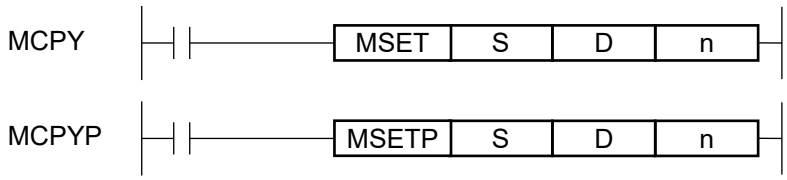
Command	Description
MSET	Transfers the OM data specified in S, to the n point from the OM specified in D.
MSETP	Transfers the OM data specified in S to the n point for the OM specified in D when the operation result changes from OFF → ON.

Usable OM

	Bit									Word								Constant		Label	Nesting	WL designation(*)	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
D		○	○		○	○	○			○	○	○	○	○	○	○						○	○
n																		○	○				

* Only multiple bit access (up to 16-bit) for bit OMs is possible.

Circuit diagram



Error code

- When the range for moving or transferring data exceeds the corresponding device. (Error code 23)

11

XCHG(P)

Function

Command	Description
XCHG	Exchanges the 16-bit data for D1 and D2.
XCHGP	Exchanges the 16-bit data for D1 and D2 when the operation result changes from OFF → ON.

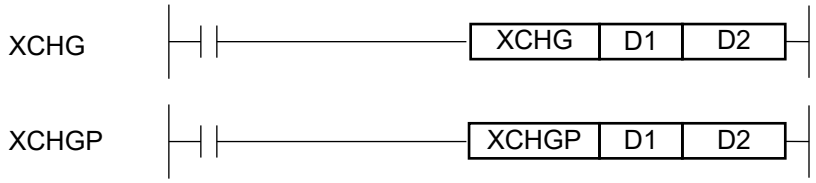
The operation result is as follows when XCHG (P) is executed.

D1 before execution	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
D2 before execution	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
D1 after execution	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
D2 after execution	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0

Usable OM

	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
D1		○	○		○	○	○			○	○	○	○	○	○	○	○					○	○
D2		○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

Circuit diagram



11.3.2.5 Branch

JE

Function

Command	Description
JE	Executes the program with the label number specified in the same POU when the contact is ON. The program for the next step is executed when the contact is OFF.

(NOTE 1) Pay attention to the following points when using the JE command.
- After the local timer coil turns ON, the local timer continues the count even if there is a local timer jump (JE) with the coil that has been turned ON.
- When the OUT command is jumped with a JE command, the coil status is maintained.
- When a jump is carried out backward with a JE command, the scan time becomes shorter.
- The JE command can be used to jump to a step that is lower than the step currently being executed.
  However, a method to break out of this closed loop must be considered.

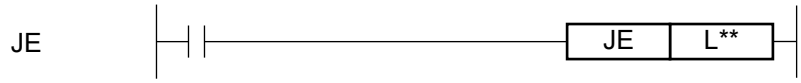
(NOTE 2) When the end of a label (L1023) is specified, it jumps to the END command.

(NOTE 3) The user cannot jump to another POU label.

Usable OM

Bit									Word								Constant		Label	Nesting	WL designation	INDEX
X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
																			○			

Circuit diagram



Error code

When there is no label for the jump location. (Error code: 21)

JMP

Function

Command	Description
JMP	Executes the program with the label number specified in the same POU without any conditions.

(NOTE 1) Pay attention to the following points when using the JMP command.
- After the local timer coil turns ON, the local timer continues the count even if there is a local timer jump (JMP) with the coil that has been turned ON.
- When the OUT command is jumped with a JMP command, the coil status is maintained.
- When a jump is carried out backward with a JMP command, the scan time becomes shorter.
- The JMP command can be used to jump to a step that is lower than the step currently being executed.
  However, a method to break out of this closed loop must be considered.

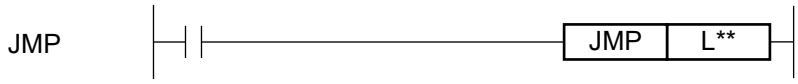
(NOTE 2) When the end of a label (L1023) is specified, it jumps to the END command.

(NOTE 3) The user cannot jump to another POU label.

Usable OM

Bit									Word								Constant		Label	Nesting	WL designation	INDEX
X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
																			○			

Circuit diagram



Error code

When there is no label for the jump location. (Error code: 21)

CALL(P)

Function

Command	Description
CALL	Executes the subroutine with the label number specified in the same POU when the previous operation result is ON. Nesting (nesting structure) is possible for up to 8 levels.
CALLP	Executes the subroutine with the label number specified in the same POU only when the previous operation result changes from OFF → ON. Nesting (nesting structure) is possible for up to 8 levels.

(NOTE 1) The RET command executes the program from the step following the CALL command.

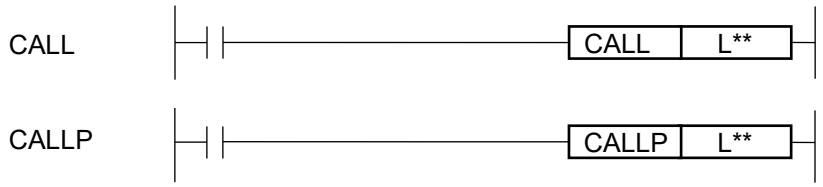
(NOTE 2) Create a subroutine after the ENDS command.

(NOTE 3) The user cannot execute another POU subroutine.

Usable OM

Bit									Word								Constant		Label	Nesting	WL designation	INDEX
X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
																			○			

Circuit diagram



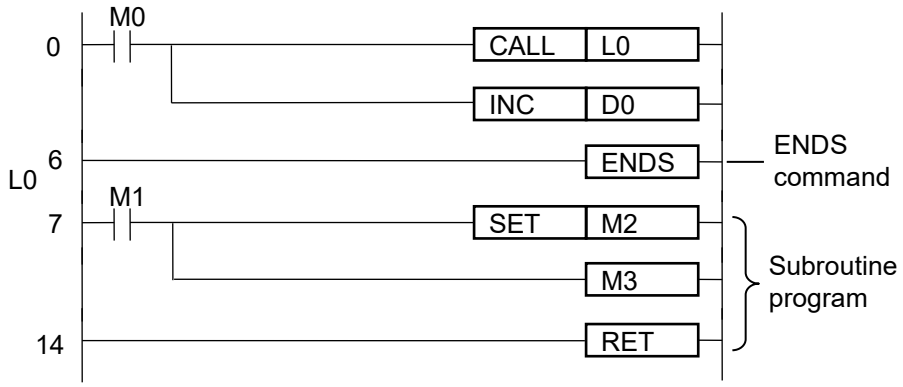
Error code

An error is triggered when a command is executed in the following situations.

- When the END (ENDS) command is executed after the CALL (P) command is executed but before the RET command is executed. (Error code 22)
- When the RET command is executed before the CALL (P) command is executed. (Error code 22)
- When there are 9 or more levels of nesting. (Error code 24)

Program

Ex) The subroutine after step 7 is called.



RET

Function

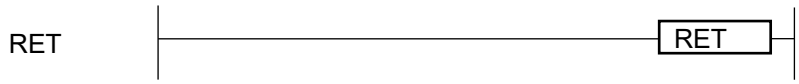
Command	Description
RET	Returns from a subroutine to the main routine.

(NOTE 1) Conditions cannot be put onto the RET command.
(NOTE 2) Use this command so it is paired with the CALL (P) command.

Usable OM

Bit									Word								Constant		Label	Nesting	WL designation	INDEX
X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		

Circuit diagram



Error code

An error is triggered when a command is executed in the following situations.

- When the END (ENDS) command is executed after the CALL (P) command is executed but before the RET command is executed. (Error code 22)
- When the RET command is executed before the CALL (P) command is executed. (Error code 22)
- When execution breaks out from a subroutine using the JMP command before the RET command is executed. (Error code 22)

11.3.2.6 Logic operation

LAND(P)(2)

Function

Command	Description
LAND (2)	Carries out an AND operation for each bit in the 16-bit data for the OM specified in D and in the 16-bit data for the OM specified in S, and the result is stored in the OM specified in D. Anything more than the specified digits for the bit OM is processed as a 0 (zero) in the operation.
LANDP (2)	Executes LAND (2) when the operation result changes from OFF → ON.

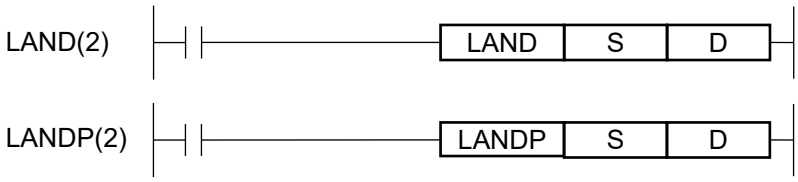
The operation result is as follows when the LAND (P) command is executed.

	b15								…							b0
S before execution	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
D before execution	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
D after execution	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0

Usable OM

	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
D		○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

Circuit diagram



LAND(P)(3)

Function

Command	Description
LAND (3)	Carries out an AND operation for each bit in the 16-bit data for the OM specified in S1 and in the 16-bit data for the OM specified in S2, and the result is stored in the OM specified in D. 0 is used to perform operations for bits that are greater than the digit specified in the bit OM.
LANDP (3)	Executes LAND (3) when the operation result changes from OFF → ON.

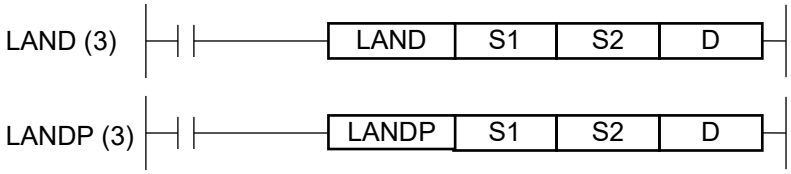
The operation result is as follows when the LAND (P) command is executed.

	b15								…							b0
S1 before execution	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
S2 before execution	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
D after execution	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0

Usable OM

	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S1	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
S2	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
D		○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

Circuit diagram



11

LOR(P)(2)

Function

Command	Description
LOR (2)	Carries out an OR operation for each bit in the 16-bit data for the OM specified in D and in the 16-bit data for the OM specified in S, and the result is stored in the OM specified in D. Anything more than the specified digits for the bit OM is processed as a 0 (zero) in the operation.
LORP (2)	Executes LOR (2) when the operation result changes from OFF → ON.

The operation result is as follows when the LOR (P) command is executed.

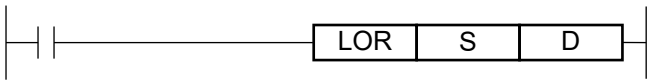
	b15								…							b0
S before execution	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
D before execution	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
D after execution	1	1	1	1	1	0	1	0	1	1	1	1	1	0	1	0

Usable OM

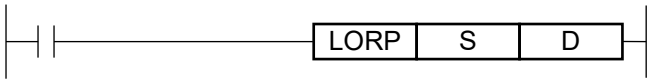
	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
D		○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

Circuit diagram

LOR (2)



LORP (2)



LOR(P)(3)

Function

Command	Description
LOR (3)	Carries out an OR operation for each bit in the 16-bit data for the OM specified in S1 and in the 16-bit data for the OM specified in S2, and the result is stored in the OM specified in D. Anything more than the specified digits for the bit OM is processed as a 0 (zero) in the operation
LORP (3)	Executes LOR (3) when the logic operation result changes from OFF → ON.

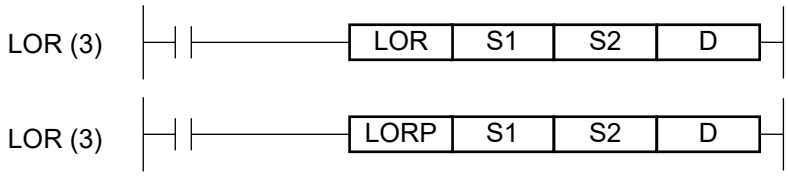
The logic operation result is as follows when the LOR (P) command is executed.

	b15								…							b0
S1 before execution	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
S2 before execution	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
D after execution	1	1	1	1	1	0	1	0	1	1	1	1	1	0	1	0

Usable OM

	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S1	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
S2	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
D		○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

Circuit diagram



11

LXOR(P)(2)

Function

Command	Description
LXOR (2)	Carries out an XOR operation for each bit in the 16-bit data for the OM specified in D and in the 16-bit data for the OM specified in S, and the result is stored in the OM specified in D. Anything more than the specified digits for the bit OM is processed as a 0 (zero) in the operation
LXORP (2)	Executes LXOR (2) when the operation result changes from OFF → ON.

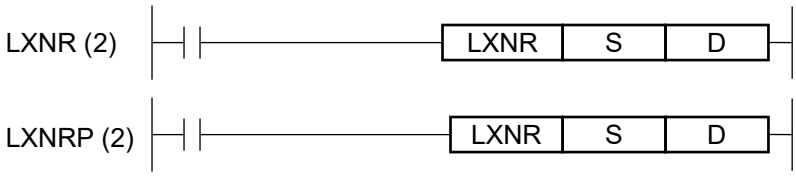
The operation result is as follows when the LXOR (P) command is executed.

	b15								…							b0
S before execution	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
D before execution	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
D after execution	0	1	0	1	1	0	1	0	0	1	0	1	1	0	1	0

Usable OM

	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
D		○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

Circuit diagram



11

LXOR(P)(3)

Function

Command	Description
LXOR (3)	Carries out an XOR operation for each bit in the 16-bit data for the OM specified in S1 and in the 16-bit data for the OM specified in S2, and the result is stored in the OM specified in D. Anything more than the specified digits for the bit OM is processed as a 0 (zero) in the operation
LXORP (3)	Executes LXOR (3) when the operation result changes from OFF → ON.

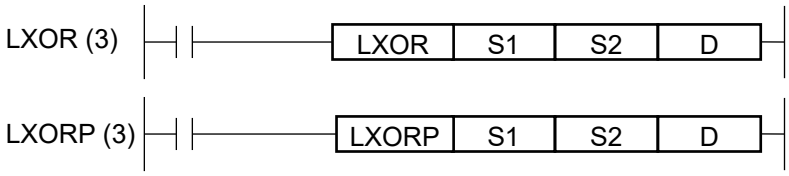
The operation result is as follows when the LXOR (P) is executed.

	b15								…							b0
S1 before execution	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
S2 before execution	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
D after execution	0	1	0	1	1	0	1	0	0	1	0	1	1	0	1	0

Usable OM

	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S1	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
S2	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
D		○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

Circuit diagram



11

LXNR (P)(2)

Function

Command	Description
LXNR (2)	Carries out an XNOR operation for each bit in the 16-bit data for the OM specified in D and in the 16-bit data for the OM specified in S, and the result is stored in the OM specified in D. Anything more than the specified digits for the bit OM is processed as a 0 (zero) in the operation.
LXNRP (2)	Executes LXNR (2) when the operation result changes from OFF → ON.

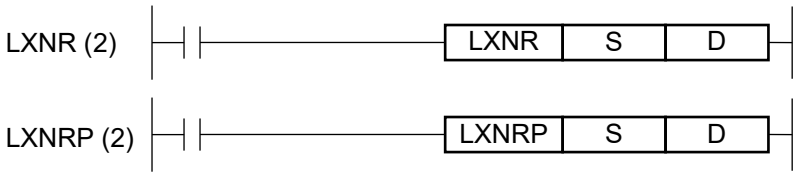
The operation result is as follows when the LXNR (P) command is executed.

	b15								…							b0
S before execution	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
D before execution	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
D after execution	1	0	1	0	0	1	0	1	1	0	1	0	0	1	0	1

Usable OM

	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
D		○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

Circuit diagram



LXNR (P)(3)

Function

Command	Description
LXNR (3)	Carries out an XNOR operation for each bit in the 16-bit data for the OM specified in S1 and in the 16-bit data for the OM specified in S2, and the result is stored in the OM specified in D. Anything more than the specified digits for the bit OM is processed as a 0 (zero) in the operation.
LXNRP (3)	Executes LXNR (3) when the operation result changes from OFF → ON.

The operation result is as follows when the LXNR (P) command is executed.

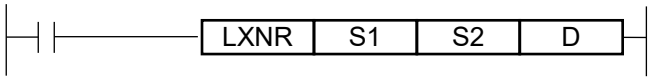
	b15								…							b0
S1 before execution	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
S2 before execution	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
D after execution	1	0	1	0	0	1	0	1	1	0	1	0	0	1	0	1

Usable OM

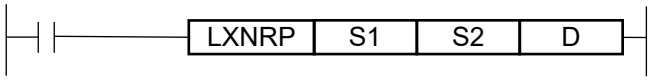
	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S1	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
S2	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
D		○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

Circuit diagram

LXNR (3)



LXNRP (3)



NEG(P)

Function

Command	Description
NEG	Inverts the sign for the 16-bit OM specified in D, and stores it to the OM specified in D. The command is used when inverting positive and negative signs.
NEGP	Inverts the sign for the 16-bit OM specified in D when the operation result changes from OFF → ON, and stores the result to the OM specified in D.

The operation result is as follows when the NEG (P) command is executed.

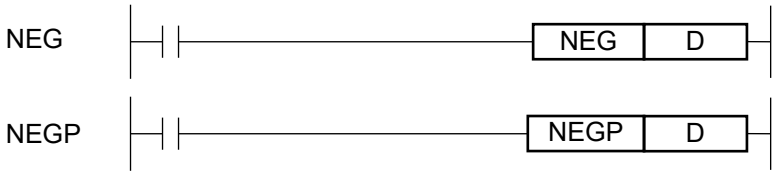
D before execution	1	1	0	1	1	0	0	0	1	1	1	1	0	0	0	0	-10000
D after execution	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	+10000

Usable OM

	Bit									Word								Constant		Label	Nesting	WL designation(*)	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
D		○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

* Only multiple bit access (up to 16-bit) for bit OMs is possible. 32-bit access for word OMs is not possible.

Circuit diagram



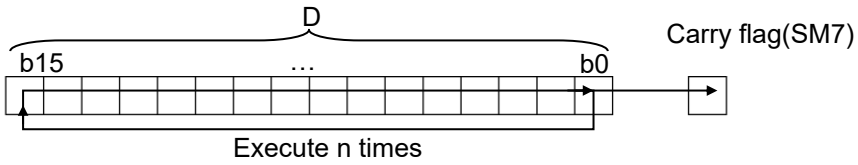
11.3.2.7 Rotation

ROR(P)

Function

Command	Description
ROR	Rotates the D data to the right by n bits. The value for b0 is stored in the carry flag. If the OM specified in D is 16-bit, the n value can be specified between 1 and 15. For 32-bit, the n value can be specified between 1 and 31.
RORP	Rotates the D data to the right by n bits when the operation result changes from OFF → ON. The value for b0 is stored in the carry flag.

Each bit for D, not including the carry flag, is rotated to the right when the ROR (P) command is executed.

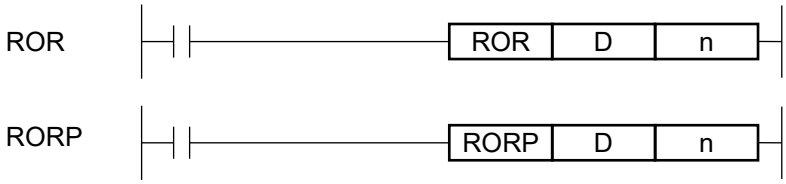


(NOTE) The carry flag (SM7) stores the overflowing bits when the bits are shifted.
The carry flag becomes either 1 or 0 depending on the status prior to the execution of ROR.

Usable OM

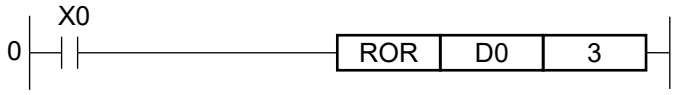
	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
D										○	○	○	○	○	○	○	○					○	○
n																		○	○				

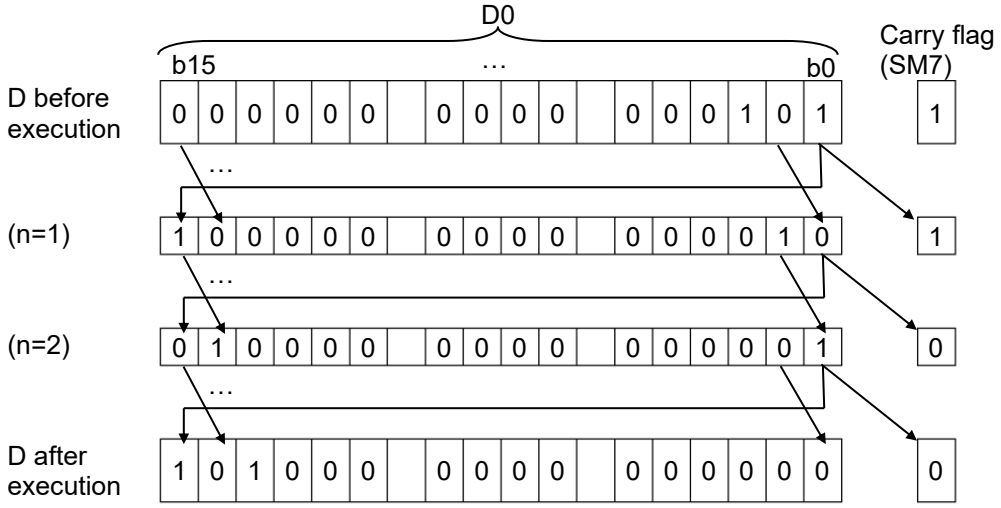
Circuit diagram



11

Program example
Ex) When X0 is ON, D0 rotates to the right by 3 bits.



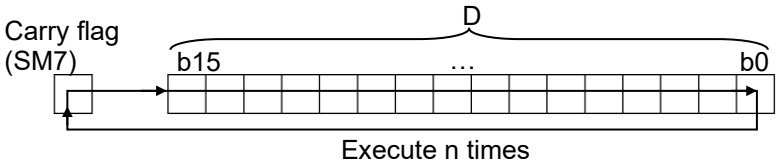


RCR(P)

Function

Command	Description
RCR	Rotates the D data, including the carry flag, to the right by n bits. If the OM specified in D is 16-bit, the n value can be specified between 1 and 16. For 32-bit, the n value can be specified between 1 and 32.
RCRP	Rotates the D data, including the carry flag, to the right by n bits when the operation result changes from OFF → ON.

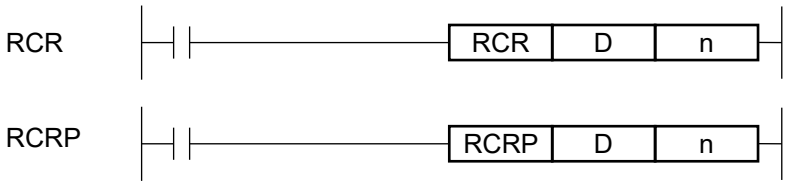
Each bit for D, including the carry flag, is rotated to the right when the RCR (P) command is executed.



Usable OM

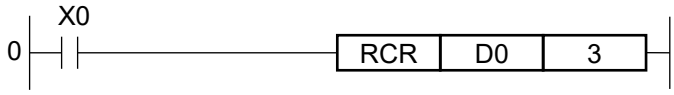
	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
D										○	○	○	○	○	○	○	○					○	○
n																		○	○				

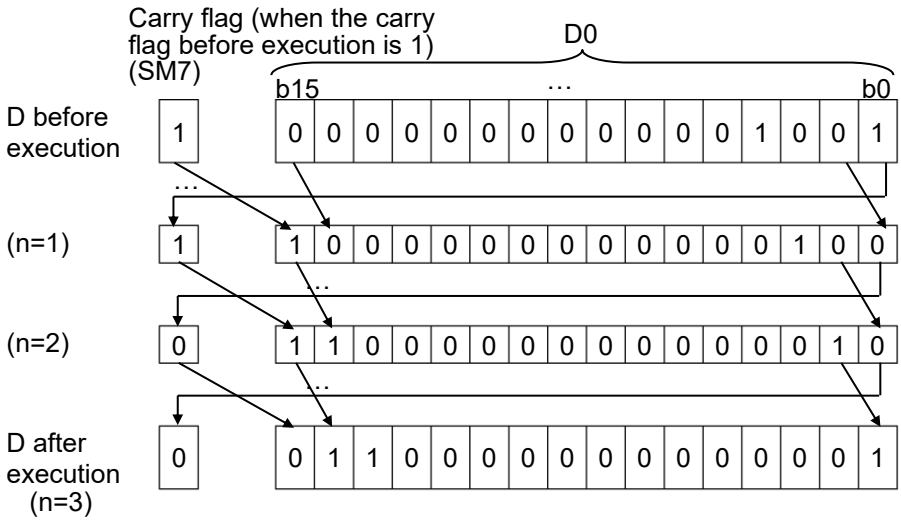
Circuit diagram



Program example

Ex) When X0 is ON, D0 rotates to the right by 3 bits.



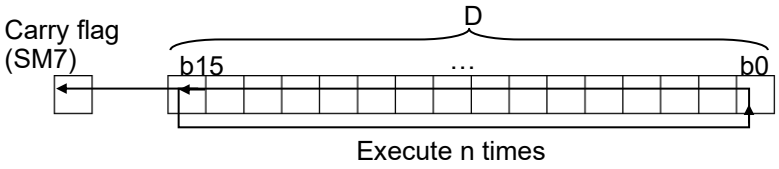


ROL(P)

Function

Command	Description
ROL	Rotates the D data to the left by n bits. The value for b15 is stored in the carry flag. If the OM specified in D is 16-bit, the n value can be specified between 1 and 15. For 32-bit, the n value can be specified between 1 and 31.
ROLP	Rotates the D data to the left by n bits when the operation result changes from OFF → ON. The value for b15 is stored in the carry flag.

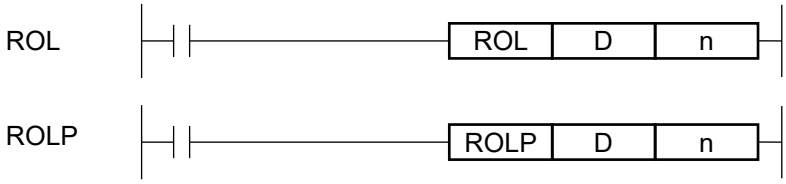
Each bit for D, not including the carry flag, is rotated to the left when the ROL (P) command is executed.



Usable OM

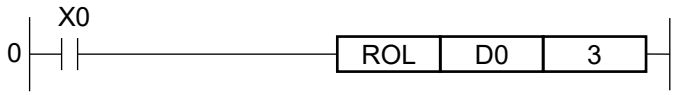
	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
D										○	○	○	○	○	○	○	○					○	○
n																		○	○				

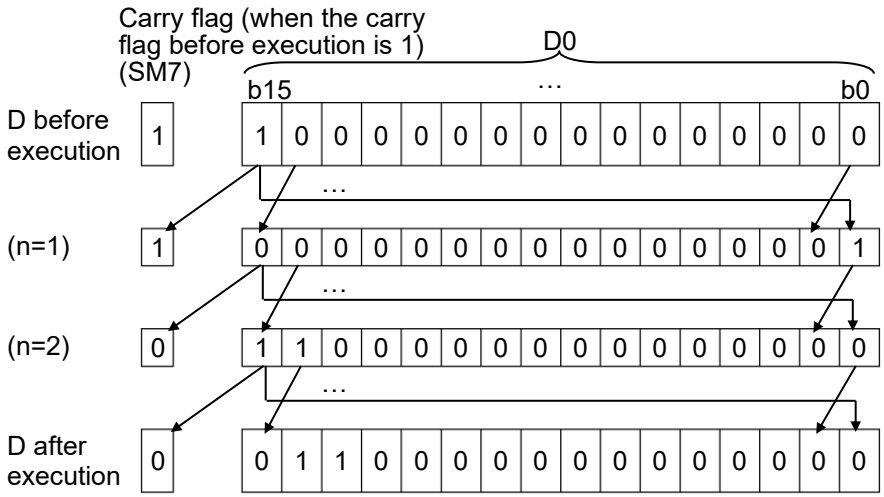
Circuit diagram



Program example

Ex) When X0 is ON, D0 rotates to the left by 3 bits.



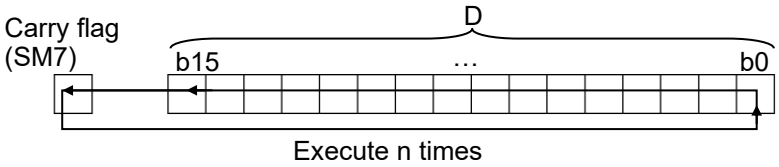


RCL(P)

Function

Command	Description
RCL	Rotates the D data, including the carry flag, to the left by n bits. If the OM specified in D is 16-bit, the n value can be specified between 1 and 16. For 32-bit, the n value can be specified between 1 and 32.
RCLP	Rotates the D data, including the carry flag, to the left by n bits when the operation result changes from OFF → ON.

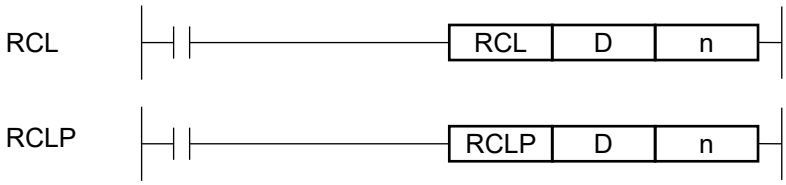
Each bit for D, including the carry flag, is rotated to the left when the RCL (P) command is executed.



Usable OM

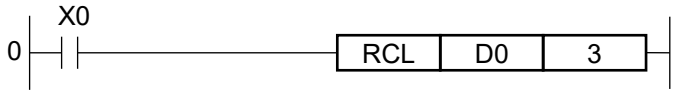
	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
D										○	○	○	○	○	○	○	○					○	○
n																		○	○				

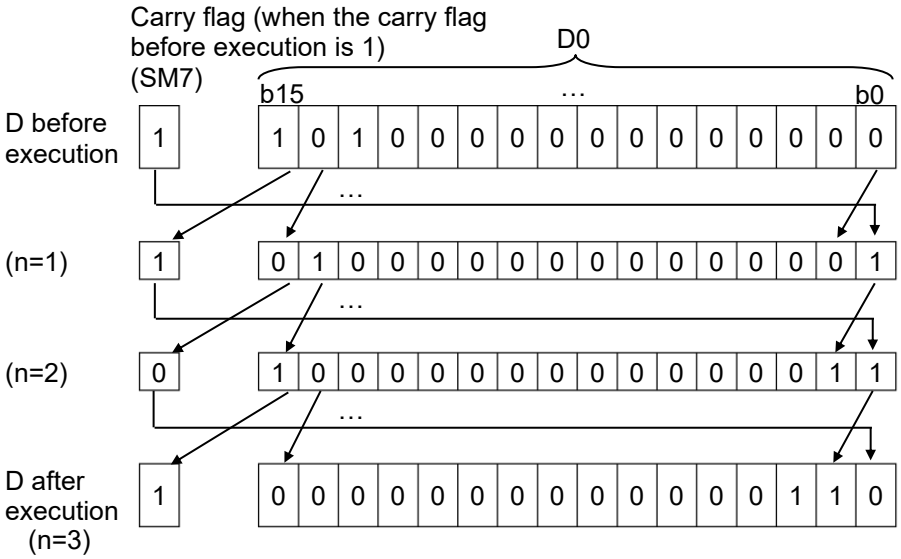
Circuit diagram



Program example

Ex) When X0 is ON, D0 rotates to the left by 3 bits.





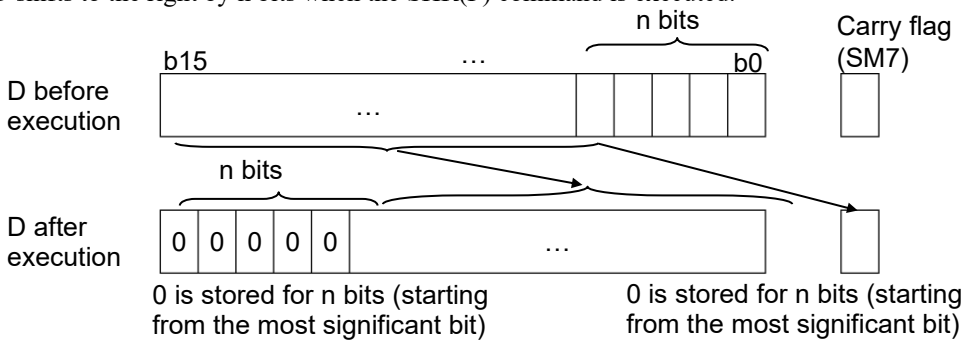
11.3.2.8 Shift

SHR(P)

Function

Command	Description
SHR	Shifts the 16-bit data for the word OM specified in D or shifts the most significant 16 bits (16 points) for the bit OM to the right by n bits. n bits (starting from the most significant bit) become 0. The value for the n bit is stored in the carry flag. The LT / LC shift is a shift of the current value (discrete value, or count value).
SHRP	Shifts the 16-bit data for the OM specified in D to the right by n bits when the operation result changes from OFF → ON. The value for the n bit is stored in the carry flag.

D shifts to the right by n bits when the SHR(P) command is executed.



(NOTE 1) 32-bit data cannot be specified in D. (An error is triggered when the user inputs using the PLC Programming software D00 EDITOR.)

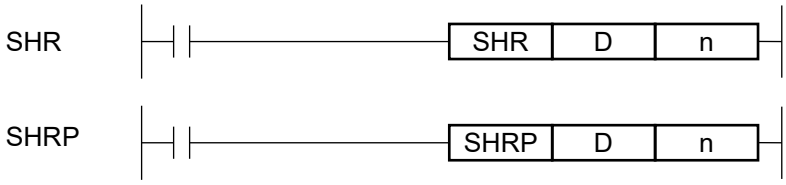
(NOTE 2) An error is triggered if the user inputs a 0 (zero) or a negative number for n when using the PLC Programming software D00 EDITOR.

Usable OM

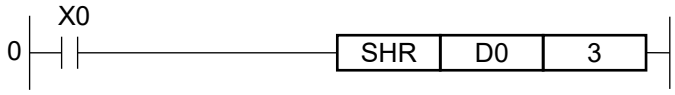
	Bit									Word								Constant		Label	Nesting	WL designation(*)	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
D		○	○	○	○	○	○			○	○	○	○	○	○	○	○					○	○
n																		○	○				

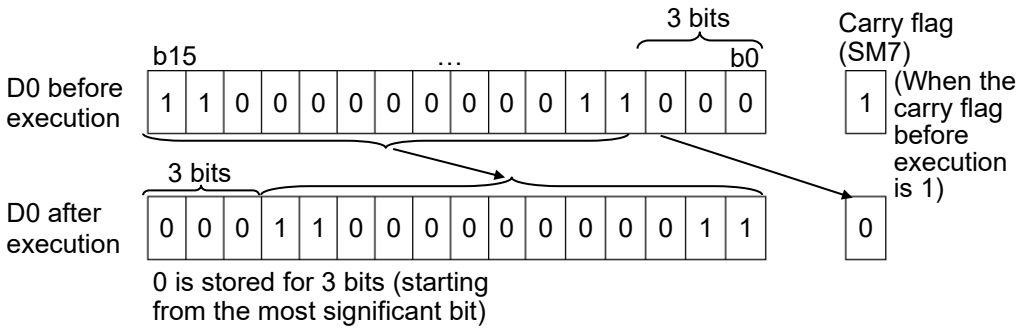
* Only multiple bit access (up to 16-bit) for bit OMs is possible. 32-bit access for word OMs is not possible.

Circuit diagram



Program example
Ex) When X0 is ON, D0 shifts to the right by 3 bits.

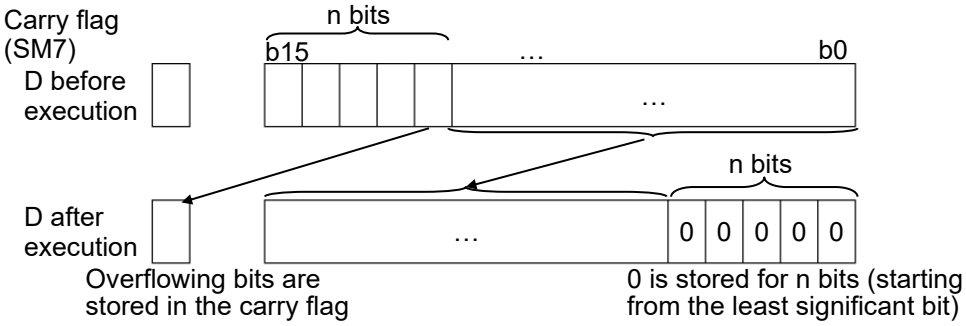




SHL(P)
Function

Command	Description
SHL	Shifts the 16-bit data for the word OM specified in D or shifts the most significant 16 bits (16 points) for the bit OM to the left by n bits. n bits (stating from the least significant bit) become 0. The value for the n bit (starting from the most significant bit) is stored in the carry flag. The LT / LC shift is a shift of the current value (discrete value, or count value). (The shift operation for the set value cannot be performed.)
SHLP	Shifts the 16-bit data for the OM specified in D to the left by n bits when the operation result changes from OFF → ON. The value for the n bit (starting from the most significant bit) is stored in the carry flag.

The D shifts to the right by n bits when the SHL command is executed.



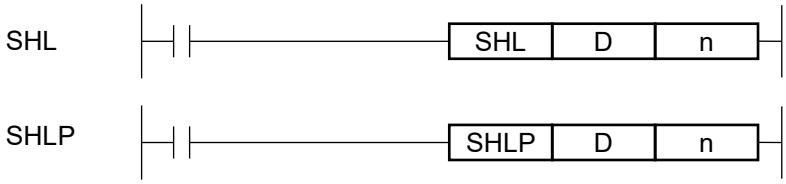
(NOTE 1) 32-bit data cannot be specified in D. (An error is triggered when the user inputs using the PLC Programming software D00 EDITOR.)
(NOTE 2) An error is triggered if the user inputs a 0 (zero) or a negative number for n when using the PLC Programming software D00 EDITOR.

Usable OM

	Bit									Word								Constant		Label	Nesting	WL designation(*)	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
D		○	○	○	○	○	○			○	○	○	○	○	○	○	○					○	○
n																		○	○				

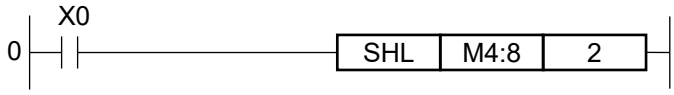
* Only multiple bit access (up to 16-bit) for bit OMs is possible. 32-bit access for word OMs is not possible.

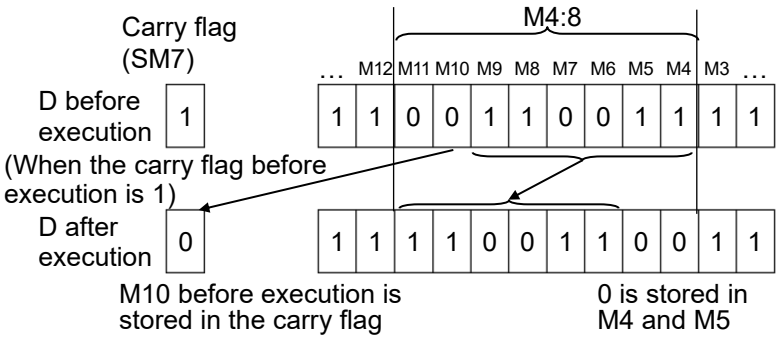
Circuit diagram



Program example

Ex) When X0 is ON, M4:8 shifts to the left by 2 bits.

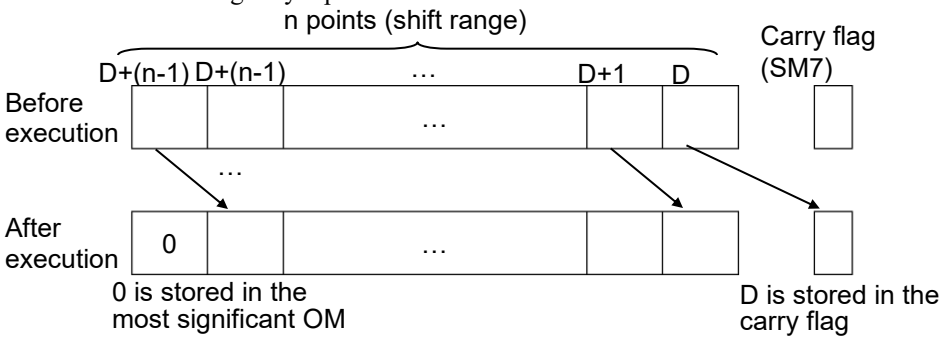




BSHR(P)
Function

Command	Description
BSHR	Shifts n points from the bit OM specified in D to the right by 1 point. The value for D is stored in the carry flag.
BSHRP	Shifts n points from the bit OM specified in D to the right by 1 point when the operation result changes from OFF → ON. The value for D is stored in the carry flag.

n points from D shift to the right by 1 point when the BSHR command is executed.



(NOTE 1) An error is triggered if the user inputs a 0 (zero) or a negative number for n when using the PLC Programming software D00 EDITOR.

(NOTE 2) The user can enter between 1 and 32767 for n in the PLC Programming software D00 EDITOR. However, set a value that does not exceed the range for the OM specified in D.
If a number of points that exceeds the OM range is specified, access is given even for a different type of OM.

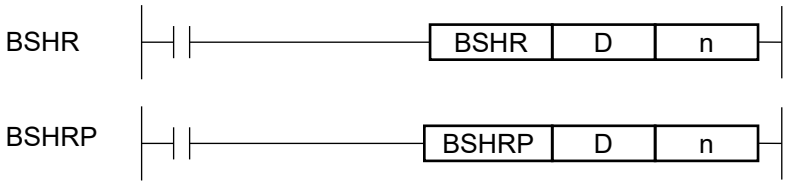
However, if another OM value is read and written, it is read and written following the order below.
Global OM range: X → Y → M → D → BX → BY → BDX → BDY → Variable
Local OM range: LM → LD → LT → LTS → LTC → LTV (current value) → LTP (set value) → LC → LCS → LCC → LCV (current value) → LCP (set value) → SM → SD → IX
Ex) When “BSHR Y7F0 32” is executed, a shift is carried out within the following ranges: between Y7F0 and Y7FF; between M0 and M15.

As a result, an unexpected motion in the program may occur. Therefore, be careful not to set a number of shift points that exceeds the OM range.

Usable OM

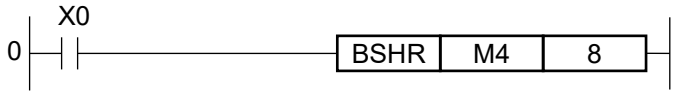
	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
D		○	○		○	○	○																○
n																		○	○				

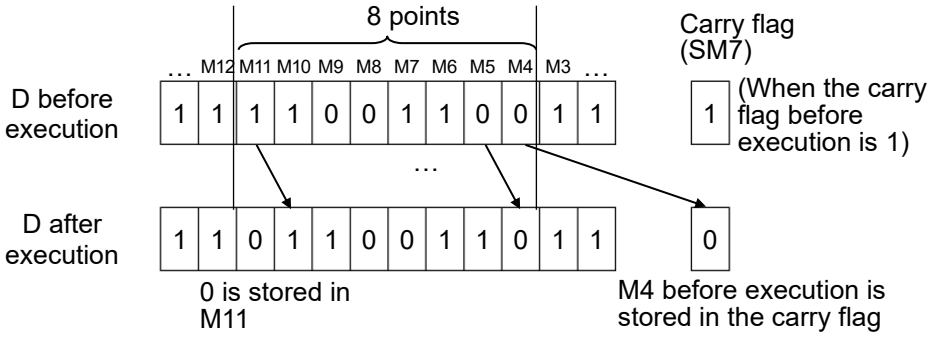
Circuit diagram



Program example

Ex) When X0 is ON, M4 through M11 shift to the right by 1 point.

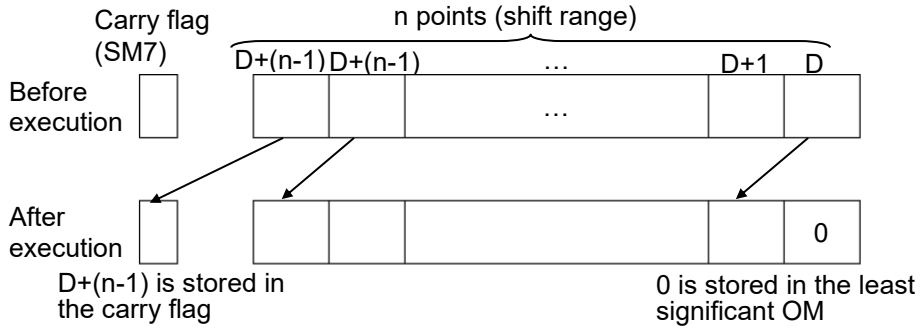




BSHL(P)
Function

Command	Description
BSHL	Shifts the beginning of the bit OM specified in D, n points to the left by 1 point.
BSHLP	Shifts the beginning of the bit OM specified in D, n points to the left by 1 point when the logic operation result changes from OFF → ON.

The n points from D shift to the left by 1 point when the BSHL command is executed.



(NOTE 1) An error is triggered if the user inputs a 0 (zero) or a negative number for n when using the PLC Programming software D00 EDITOR.

(NOTE 2) The user can enter between 1 and 32767 for n in the PLC Programming software D00 EDITOR. However, set a value that does not exceed the range for the OM specified in D.
If a number of points that exceeds the OM range is specified, access is given even for a different type of OM.

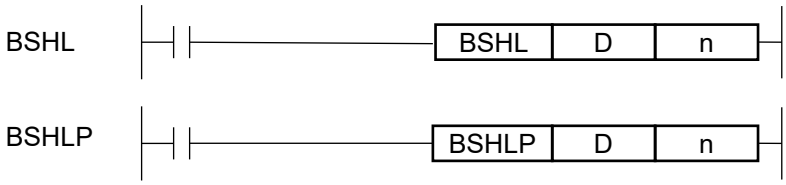
However, if another OM value is read and written, it is read and written following the order below.
Global OM range: X → Y → M → D → BX → BY → BDX → BDY → Variable
Local OM range: LM → LD → LT → LTS → LTC → LTV (current value) → LTP (set value) → LC → LCS → LCC → LCV (current value) → LCP (set value) → SM → SD → IX
Ex) When “BSHL Y7F0 32” is executed, a shift is carried out within the following ranges: between Y7F0 and Y7FF; between M0 and M15.

As a result, an unexpected motion in the program may occur. Therefore, be careful not to set a number of shift points that exceeds the OM range.

Usable OM

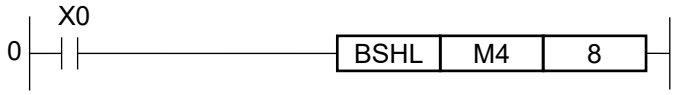
	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
D		○	○		○	○	○																○
n																		○	○				

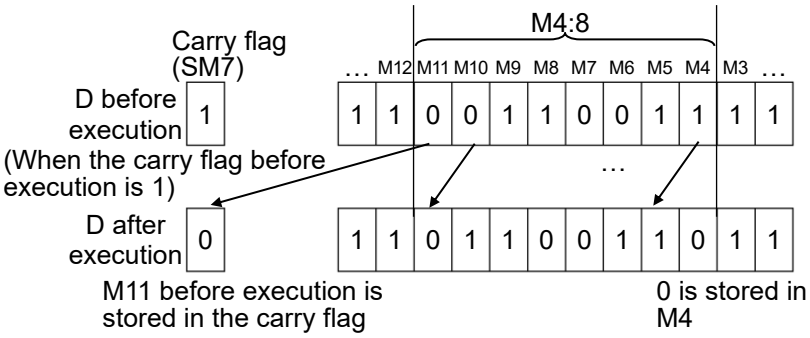
Circuit diagram



Program example

Ex) When X0 is ON, M4 through M11 shift to the left by 1 point.

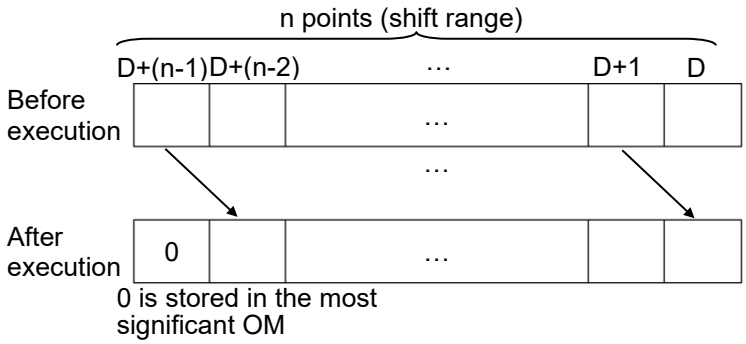




WSHR(P)
Function

Command	Description
WSHR	Shifts the beginning of the OM specified in D, n points to the right by 1 point. The most significant OM becomes 0. The LT / LC shift is a shift of the current value (discrete value, or count value). (The shift operation for the set value cannot be performed.)
WSHRP	Shifts the beginning of the OM specified in D, n points to the right by 1 point when the operation result changes from OFF → ON.

n points from D shift to the right by 1 point when the WSHR command is executed.



(NOTE 1) An error is triggered if the user inputs a 0 (zero) or a negative number for n when using the PLC Programming software D00 EDITOR.

(NOTE 2) The user can enter between 1 and 32767 for n in the PLC Programming software D00 EDITOR. However, set a value that does not exceed the range for the OM specified in D.
If a number of points that exceeds the OM range is specified, access is given even for a different type of OM.

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

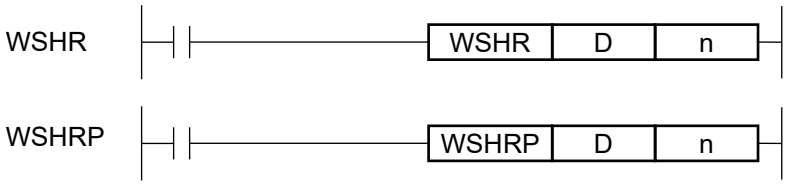
Ex) When “WSHR D8191 2” is executed, a shift is carried out for the following range: D8191, BX0000 to BX00F.

As a result, an unexpected motion in the program may occur. Therefore, be careful not to set a number of shift points that exceeds the OM range.

Usable OM

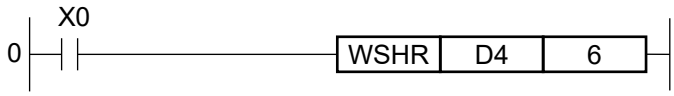
	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
D										○	○	○	○	○	○	○							○
n																		○	○				

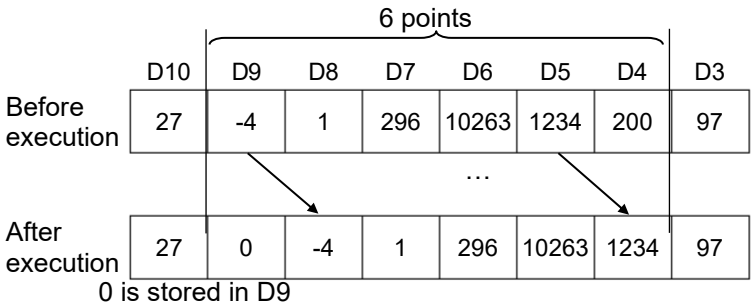
Circuit diagram



Program example

Ex) When X0 is ON, D4 through D9 shift to the right by 1 point.

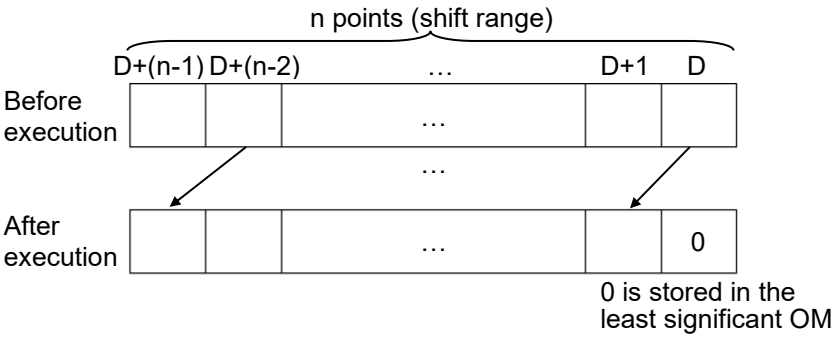




WSHL(P)
Function

Command	Description
WSHL	Shifts the beginning of the OM specified in D, n points to the left by 1 point. The least significant OM becomes 0. The LT / LC shift is a shift of the current value (discrete value, or count value). (The shift operation for the set value cannot be performed.)
WSHLP	Shifts the beginning of the OM specified in D, n points to the left by 1 point when the operation result changes from OFF → ON.

n points from D shift to the left by 1 point when the WSHL command is executed.



(NOTE 1) An error is triggered if the user inputs a 0 (zero) or a negative number for n when using the PLC Programming software D00 EDITOR.

(NOTE 2) The user can enter between 1 and 32767 for n in the PLC Programming software D00 EDITOR. However, set a value that does not exceed the range for the OM specified in D.
If a number of points that exceeds the OM range is specified, access is given even for a different type of OM.

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

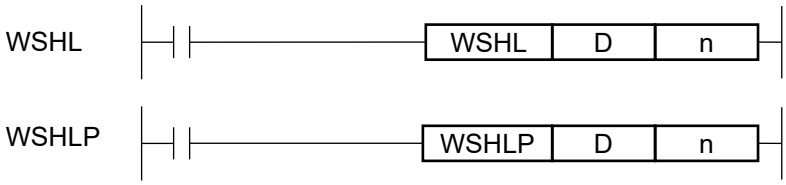
Ex) When “WSHL D8191 2” is executed, a shift is carried out for the following range: D8191, BX000 to BX00F.

As a result, an unexpected motion in the program may occur. Therefore, be careful not to set a number of shift points that exceeds the OM range.

Usable OM

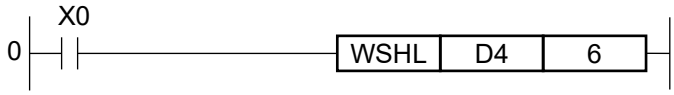
	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
D										○	○	○	○	○	○	○							○
n																		○	○				

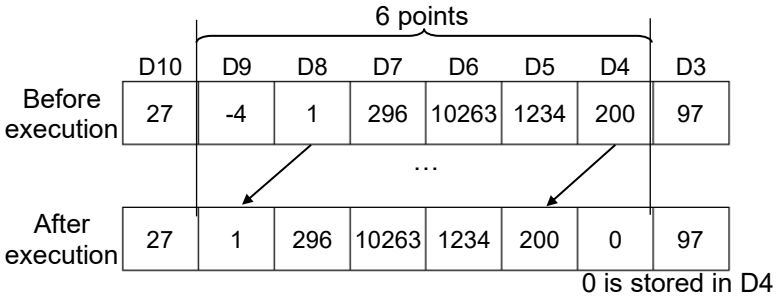
Circuit diagram



Program example

Ex) When X0 is ON, D4 through D9 shift to the right by 1 point.





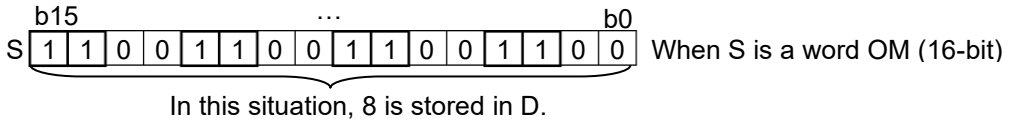
11.3.2.9 Data processing

SUM(P)

Function

Command	Description
SUM	Stores the sum (BIN data) of the 1 bits in the data for the OM specified in S, to the OM specified in D.
SUMP	Stores the sum (BIN data) of the 1 bits in the OM data specified in S, to the OM specified in D when the operation result changes from OFF → ON.

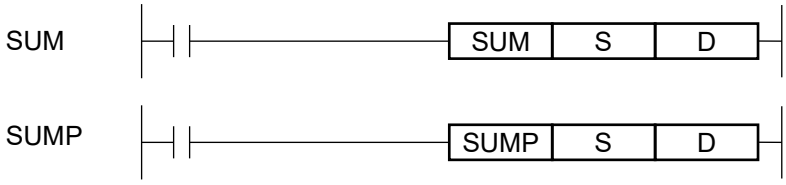
The sum of the 1 bits in the S data is stored in D when the SUM command is executed.



Usable OM

	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○					○	○
D										○	○	○	○	○	○	○	○						○

Circuit diagram



DECO(P)

Function

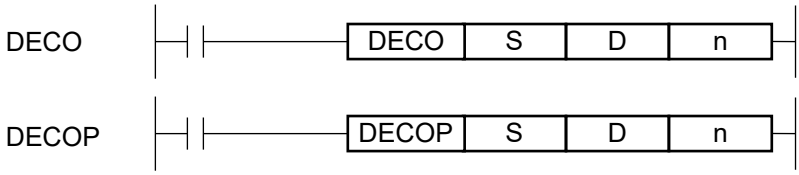
Command	Description
DECO	Decodes the low-order n bits for the OM specified in S, and stores that result as decoded data in the 2^n bits from the OM specified in D. (Decodes 8 → 256-bit) 0 through 8 can be specified for n. When n = 0, there is no processing and the 2^n bit data does not change from the OM specified in D. The bit OM is processed as 1-bit, and the word OM is processed as 16-bit.
DECOP	Executes DECO when the operation result changes from OFF → ON.

(NOTE) An error is triggered when the user inputs an out of range value for n using the PLC Programming software D00 EDITOR.

Usable OM

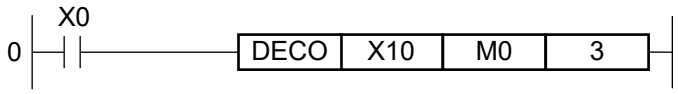
	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○				○
D		○	○		○	○	○			○	○	○	○	○	○	○							○
n																		○	○				

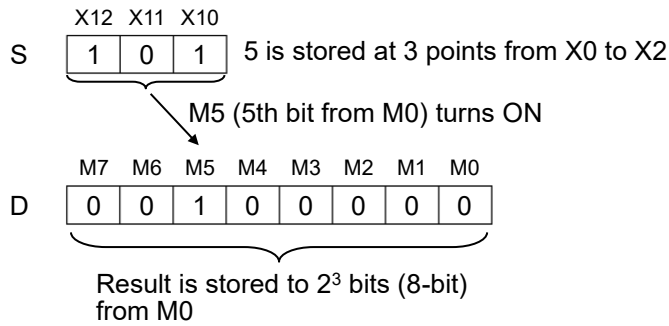
Circuit diagram



Program example

Ex) When X0 is ON, X10 through X12 is decoded and stored in M0 through M7.





11

ENCO(P)

Function

Command	Description
ENCO	Encodes the 2^n bit data from S and stores it in D. (Encodes 256 → 8-bit) 0 through 8 can be specified for n. When n = 0, there is no processing and the D data does not change. The bit OM is processed as 1-bit, and the word OM is processed as 16-bit. When multiple bits are 1, it is processed at the most significant bit position.
ENCOP	Executes ENCO when the operation result changes from OFF → ON.

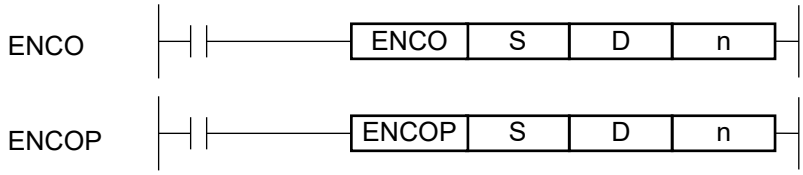
(NOTE) An error is triggered when the user inputs an out of range value for n using the PLC Programming software D00 EDITOR.

Usable OM

	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○						○
D										○	○	○	○	○	○	○	○						○
n																		○	○				

* Only multiple bit access (up to 16-bit) for bit OMs is possible. 32-bit access for word OMs is not possible.

Circuit diagram



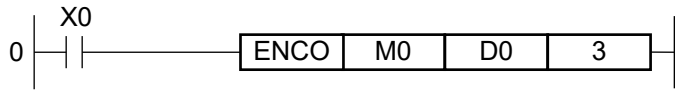
Error code

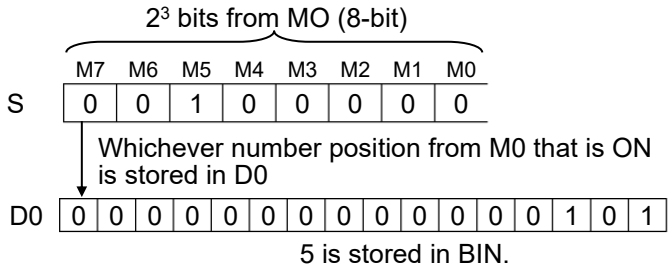
An error is triggered when a command is executed in the following situations.

- When 2^n bits from S are all 0. (Error code 99)

Program example

Ex) When X0 is ON, M0 through M7 is encoded and stored in D0.





BSET(P)

Function

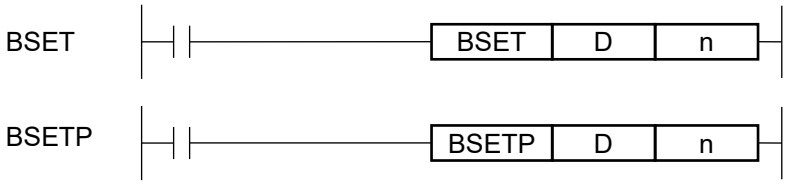
Command	Description
BSET	Sets (1) the nth bit for the word OM that is specified in D. 0 through 15 is valid for n.
BSETP	Sets (1) the nth bit for word OM specified in D when the operation result changes from OFF → ON.

(NOTE) An error is triggered when the user inputs an out of range value for n using the PLC Programming software D00 EDITOR.

Usable OM

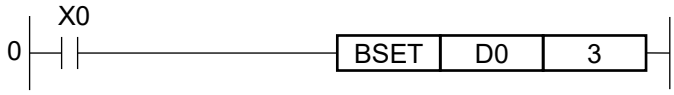
	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
D										○	○	○	○	○	○	○	○						○
n																		○	○				

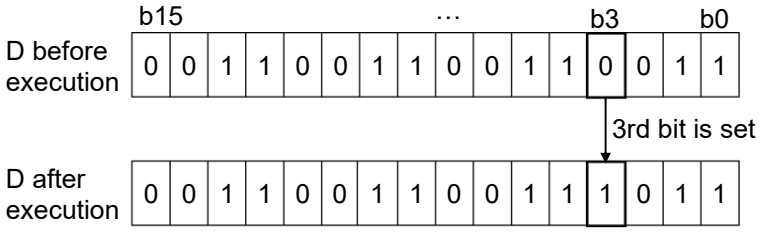
Circuit diagram



Program example

Ex) When X0 is ON, the 3rd bit for D0 is set.





BRST(P)

Function

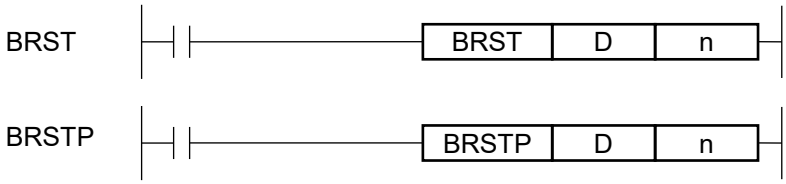
Command	Description
BRST	Resets (0) the nth bit for the word OM that is specified in D. 0 through 15 is valid for n.
BRSTP	Resets (0) the nth bit for word OM specified in D when the operation result changes from OFF → ON.

(NOTE) An error is triggered when the user inputs an out of range value for n using the PLC Programming software D00 EDITOR.

Usable OM

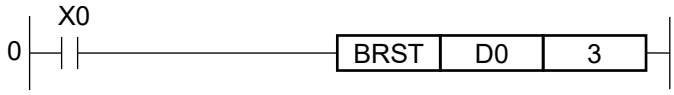
	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
D										○	○	○	○	○	○	○	○						○
n																		○	○				

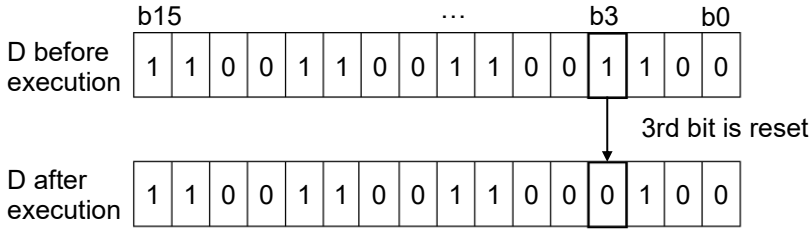
Circuit diagram



Program example

Ex) When X0 is ON, the 3rd bit for D0 is reset.



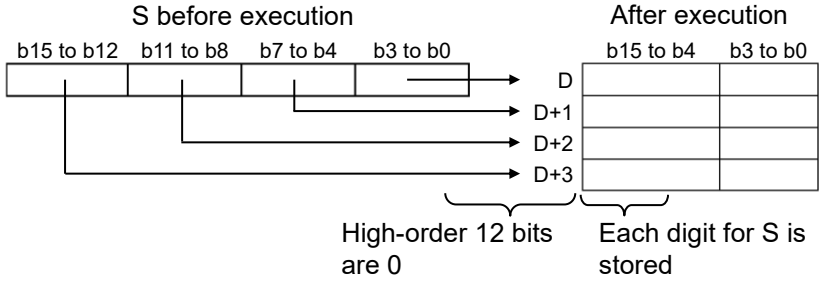


DDV(P)

Function

Command	Description
DDV	Stores the data for the low-order n digits (1 digit 4-bit) of the 16 bits specified in S, to the lower-order 4 bits of the n points from the OM specified in D. The high-order 12 bits for the n points from the OM specified in D become 0. 1 through 4 can be specified for n.
DDVP	Executes DDV when the operation result changes from OFF → ON.

The operation result is as follows when the DDV command is executed. (When n = 4)

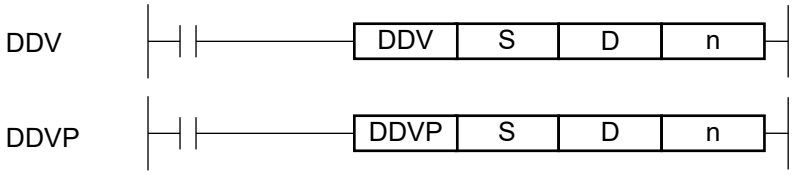


(NOTE) An error is triggered when the user inputs an out of range value for n using the PLC Programming software D00 EDITOR.

Usable OM

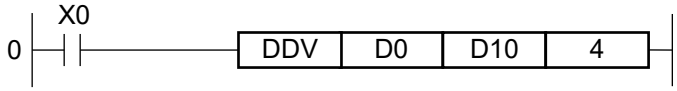
	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S										○	○	○	○	○	○	○	○	○	○				○
D										○	○	○	○	○	○	○							○
n																		○	○				

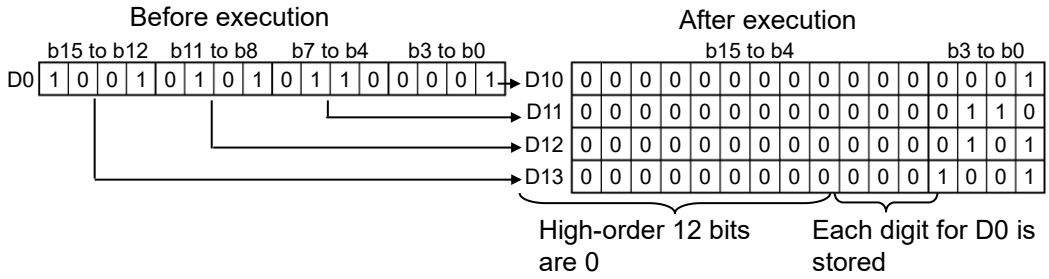
Circuit diagram



Program example

Ex) When X0 is ON, each digit for D0 is stored (for every 4 bits) in D10 through D13.



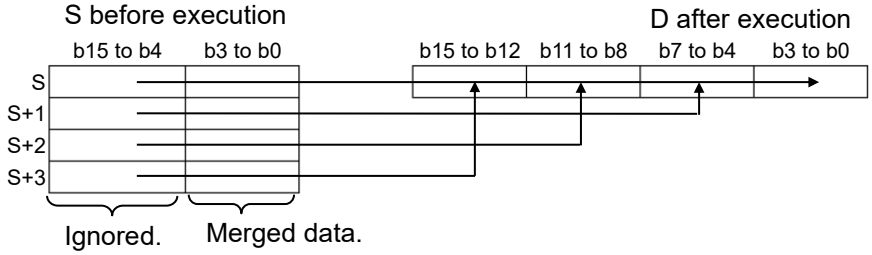


DCV(P)

Function

Command	Description
DCV	Merges the low-order 4 bits in the 16-bit data for the n points from the OM specified in S, with the 16-bit OM specified in D. The high-order (16-n×4) digit bit for the OM specified in D becomes 0. 1 through 4 can be specified for n.
DCVP	Executes DCV when the operation result changes from OFF → ON.

The operation result is as follows when the DDV (P) command is executed. (When n = 4)

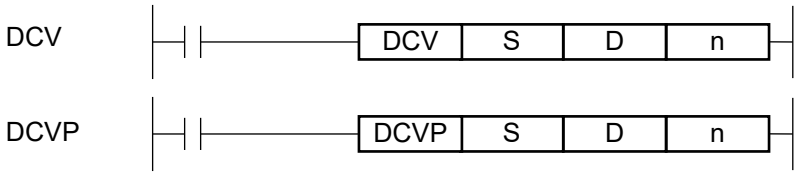


(NOTE) An error is triggered when the user inputs an out of range value for n using the PLC Programming software D00 EDITOR.

Usable OM

	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S										○	○	○	○	○	○	○							○
D										○	○	○	○	○	○	○	○						○
n																		○	○				

Circuit diagram



11

AVG

Function

Command	Description
AVG	Calculates the average value from the values that are stored in S for each scan while the contact is ON, and stores the average in D. The S sample number when calculating the average value for n1 is specified from a range between 1 and 512. The assigned number for the internal buffer that stores the previous S value is specified with n2 from a range between 0 and 9.

(NOTE 1) The S value is stored in the step where the AVG command is executed. Therefore, the value that is stored according to the executing step may change even if the same OM is specified for S.

(NOTE 2) When the contact changes from ON to OFF, the internal buffer being used is cleared. However, the D value is maintained.

(NOTE 3) When calculating the average value, the decimal part is rounded off.

(NOTE 4) The same buffer number may be used in multiple AVG commands.
However, if the contacts for those same AVG commands are turned ON at the same time, the corresponding S values for the internal buffer will get mixed up. And, the correct average number can no longer be calculated.

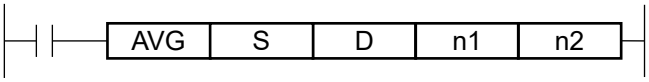
(NOTE 5) If the set value for the expanded function is not greater than 1, this command cannot be used.

Usable OM

	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○					○	○
D		○	○		○	○	○			○	○	○	○	○	○	○	○					○	○
n1										○	○	○	○	○	○	○	○	○	○				○
n2																		○	○				

Circuit diagram

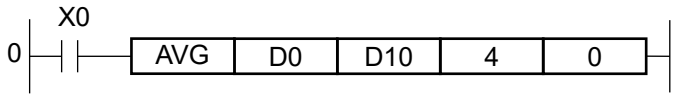
AVG

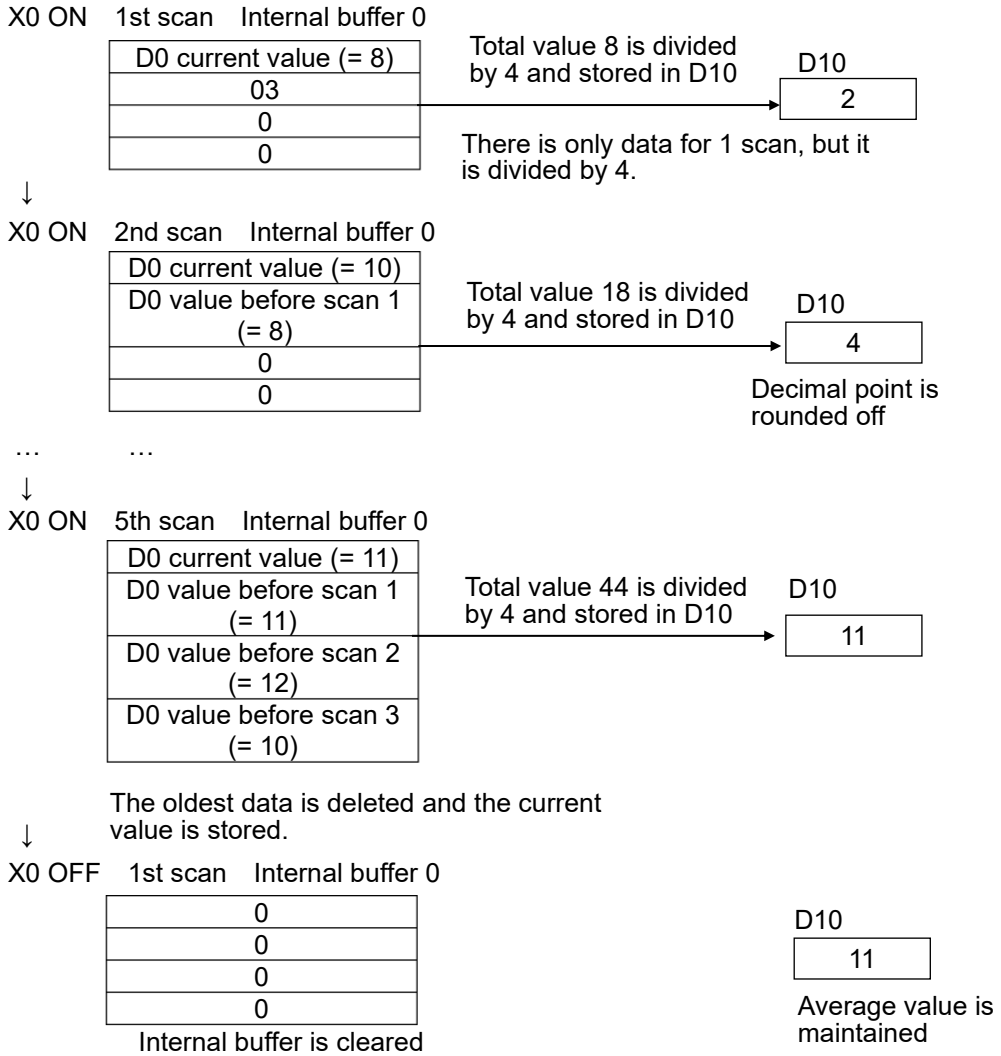


11

Program example

Ex) When X0 is ON, the D0 values for 4 scans are stored in the internal buffer 0 and is averaged out and stored in D10.





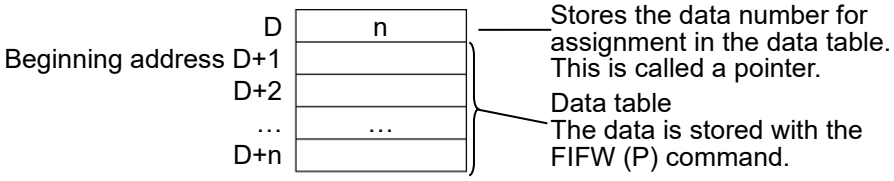
11.3.2.10 FIFO

FIFW(P)

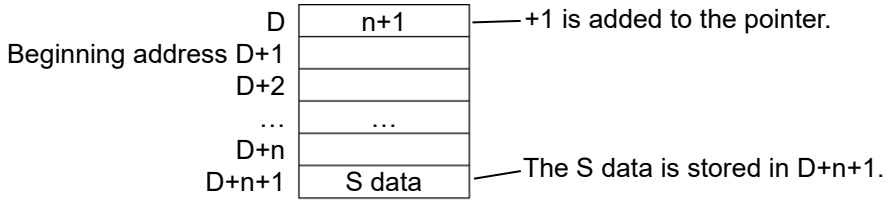
Function

Command	Description
FIFW	Performs the following operations. (1) Stores the data specified in S to the data table in the FIFO table. The following indicates the data storage position. Data storage position = Beginning address in data table + Pointer data (2) Adds +1 to the pointer data. (The pointer uses the OM specified in D.)
FIFWP	Executes FIFW when the operation result changes from OFF → ON.

The FIFO table is structured as follows.



When the FIFW (P) command is executed, the following occurs.



(NOTE 1) When the FIFW command is executed first, be sure to clear the pointer specified in D to 0.

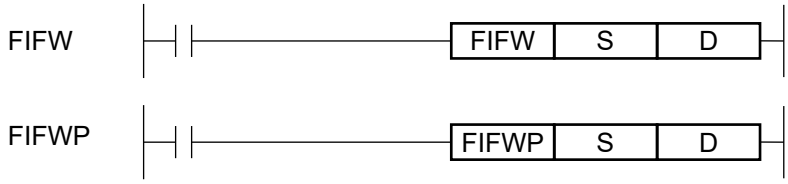
(NOTE 2) The user shall manage the data number that can be written into multiple FIFO tables.

Usable OM

	Bit									Word								Constant		Label	Nesting	WL designation(*)	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○	○	○			○	○
D										○	○	○	○	○	○	○							○

* Only multiple bit access (up to 16-bit) to bit OMs is possible for S.

Circuit diagram



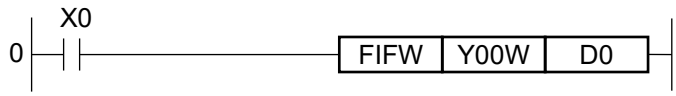
Error code

An error is triggered when a command is executed in the following situations.

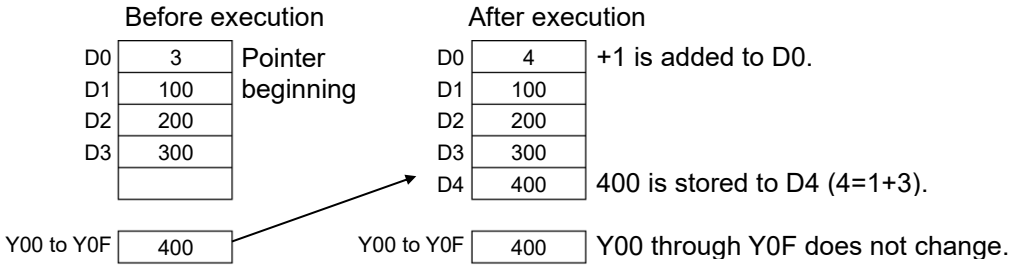
- When the beginning address + pointer data in the FIFO table exceed the range of the corresponding device. (Error code 23)

Program example

Ex) When X0 is ON, the data for Y00 through Y0F is stored in the FIFO table (from D0).



Before execution, the 3 sets of data is stored in the FIFO table.

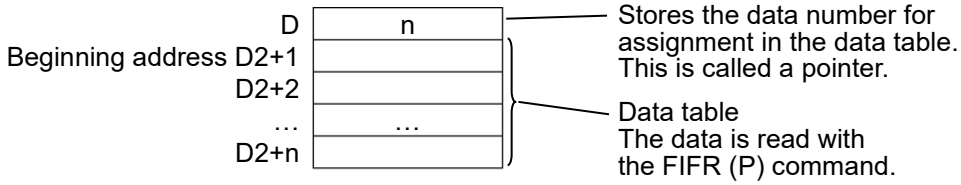


FIFR(P)

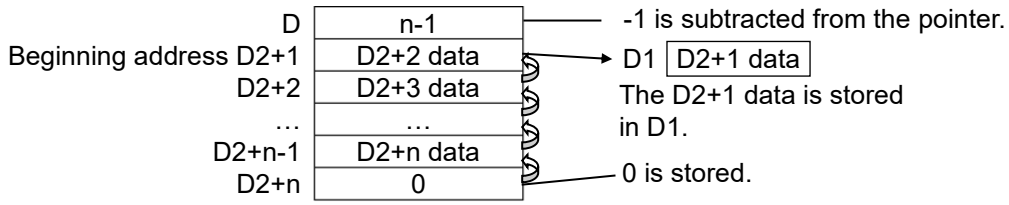
Function

Command	Description
FIFR	Reads the data from the OM that follows the FIFO table pointer and stores that data in the OM on the D1 side. The data (at each address) in the data table shifts up, and the data for the last address becomes 0.
FIFRP	Executes FIFR when the operation result changes from OFF → ON.

The FIFO table is structured as follows.



When the FIFR (P) command is executed, the following occurs.

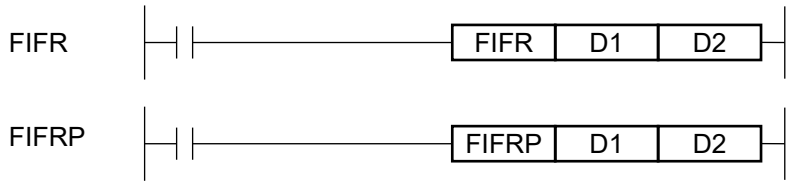


Usable OM

	Bit									Word								Constant		Label	Nesting	WL designation(*)	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S	○	○	○	○	○	○	○			○	○	○	○	○	○	○	○				○	○	○
D										○	○	○	○	○	○	○							○

* Only multiple bit access (up to 16-bit) to bit OMs is possible for S.

Circuit diagram



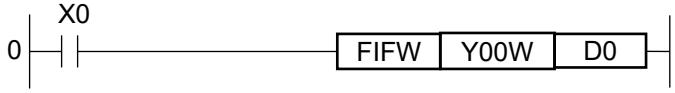
Error code

An error is triggered when a command is executed in the following situations.

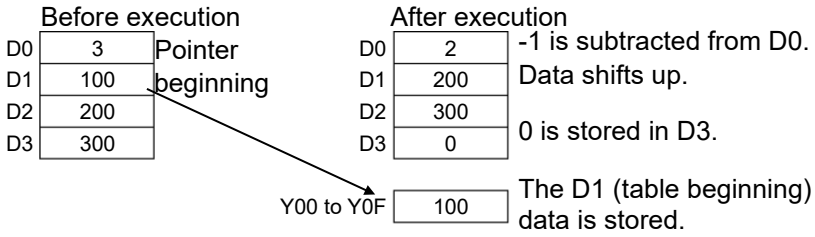
When the pointer data is 0. (Error code 99)

Program example

Ex) When X0 is ON, the data from the FIFO table (from D0) is read and the data for Y00 through Y0F is stored.



Before execution, the 3 data points are stored in the FIFO table.



11.3.2.11 Loop

FOR

Function

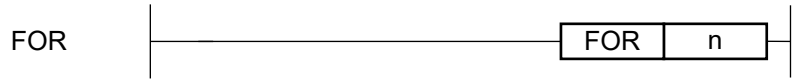
Command	Description
FOR	Performs the processing for the next step after the NEXT command, when executing the processing between the FOR and NEXT commands n times without conditions. 1 through 32767 can be specified for n. When specifying between -32768 and 0, it performs the same processing as when n = 1. Up to 5 levels are possible in FOR nesting.

(NOTE 1) Breaking out from a FOR-NEXT loop using JMP or JE is not possible.
(NOTE 2) Use the BREAK command to break out of the loop.

Usable OM

Bit									Word								Constant		Label	Nesting	WL designation	INDEX
X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
									○	○	○	○	○	○	○	○	○	○				○

Circuit diagram



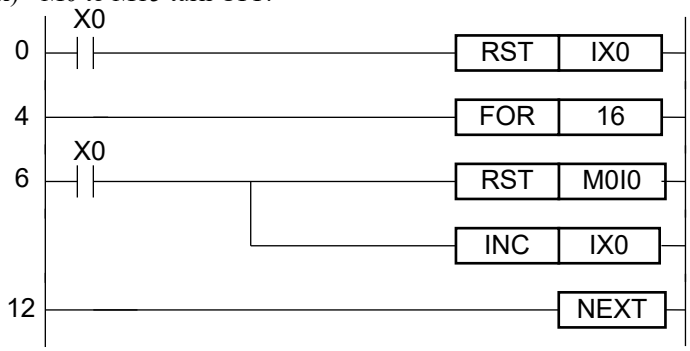
Error code

An error is triggered when a command is executed in the following situations.
In addition, the timing of the error varies depending on the cause of the error.

- When breaking out from a FOR-NEXT loop using JMP or JE.
  (Error code 22, error is triggered when executing END (ENDS))
- When the END (ENDS) command is executed after the FOR command is executed but before the NEXT command is executed.
  (Error code 22, error is triggered when executing END (ENDS))
- When the NEXT command is executed before the FOR command is executed.
  (Error code 22, error is triggered when executing the NEXT command)
- When the number of FOR commands does not match the number of NEXT commands.
  (Error code 22. If there are too many FOR commands, an error is triggered when an END (ENDS) command is executed. If there are too many NEXT commands, an error is triggered when a NEXT command without its pair is executed.)

Program example

Ex) M0 to M15 turn OFF.



NEXT

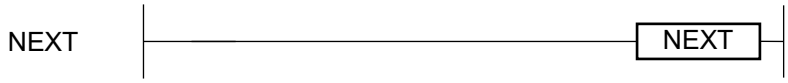
Function

Command	Description
NEXT	Performs the processing for the next step after the NEXT command, when executing the processing between the FOR and NEXT commands n times without conditions.

Usable OM

Bit									Word								Constant		Label	Nesting	WL designation	INDEX
X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		

Circuit diagram



Error code

An error is triggered when a command is executed in the following situations.

- When the NEXT command is executed before the FOR command is executed. (Error code 22)
- When the number of FOR commands does not match the number of NEXT commands and the NEXT command without its pair (FOR command) is executed. (Error code 22)

BREAK

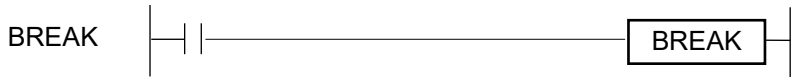
Function

Command	Description
BREAK	Performs the processing for the next step after the NEXT command, when executing from the FOR command to the NEXT command.

Usable OM

Bit									Word								Constant		Label	Nesting	WL designation	INDEX
X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		

Circuit diagram



Error code

An error is triggered when a command is executed in the following situations.

- When using this command in another location that is not between the FOR and NEXT commands. (Error code 22)

11.3.2.12 Carry flag

STC

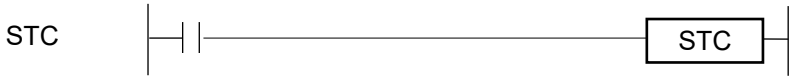
Function

Command	Description
STC	Sets (ON) the carry flag (SM7).

Usable OM

Bit									Word								Constant		Label	Nesting	WL designation	INDEX
X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		

Circuit diagram



CLC

Function

Command	Description
CLC	Resets (OFF) the carry flag (SM7).

Usable OM

Bit									Word								Constant		Label	Nesting	WL designation	INDEX
X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		

Circuit diagram

11

11.3.2.13 Date/Time

DTRD

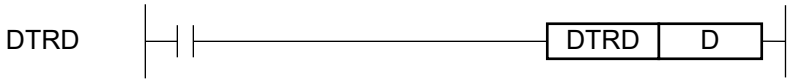
Function

Command	Description
DTRD	Stores the date and time in 7 words from the OM specified in D when the DTRD command is executed.

Usable OM

Bit									Word								Constant		Label	Nesting	WL designation	INDEX
X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
	○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

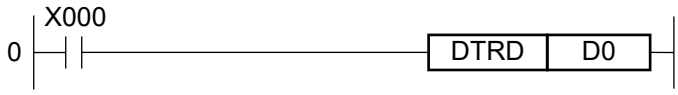
Circuit diagram



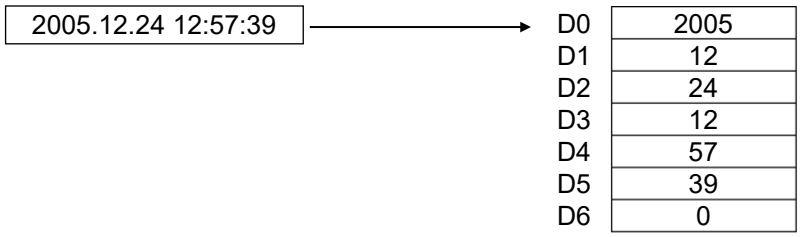
The information stored in the OM is as follows.

Position from OM specified in argument	Stored data	Notes
+0	Year	Year is stored
+1	Month	1 through 12 is used for the month and stored
+2	Day	1 through 31 is used for the day and stored
+3	Hour	0 through 23 is used for the hour and stored
+4	Minute	0 through 59 is used for the minute and stored
+5	Second	0 through 59 is used for the second and stored
+6	Day of week	Data used to store the day of the week is as follows: 0: Sunday 1: Monday 2: Tuesday 3: Wednesday 4: Thursday 5: Friday 6: Saturday

Program example



Time data



11.3.2.14 Palletize

PALETTE

Function

Command	Description
PALETTE	After setting the size of the workpiece to be stacked, the number of stacking levels and the stacking method to the OM specified in S, the palletizing command automatically calculates the coordinates for stacking the workpiece and stores it in the OM specified in D.

Usable OM

	Bit									Word								Constant		Label	Nesting	WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
S		○	○		○	○	○			○	○	○	○	○	○	○	○					○	○
D		○	○		○	○	○			○	○	○	○	○	○	○	○					○	○

Circuit diagram

PALETTE

The information that is set in S is shown below.

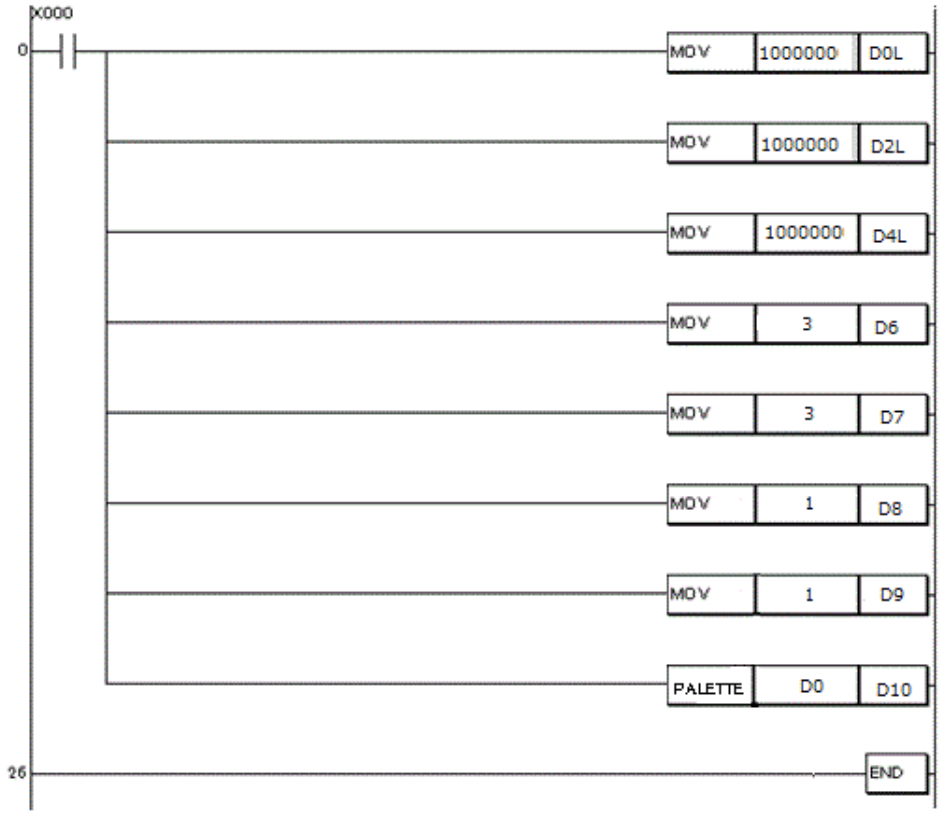
Address	Use	Setting	Notes
Sn, Sn+1	Length of workpiece in X direction	Sets the length in the X direction for the workpieces that are stacked	Setting in µm units
Sn+2, Sn+3	Length of workpiece in Y direction	Sets the length in the Y direction for the workpieces that are stacked	Setting in µm units
Sn+4, Sn+5	Length of workpiece in Z direction	Sets the length in the Z direction for the workpieces that are stacked	Setting in µm units
Sn+6	Number of workpieces stacked in the X direction	Sets the number of workpieces stacked in the X direction	Maximum value: 10
Sn+7	Number of workpieces stacked in the Y direction	Sets the number of workpieces stacked in the Y direction	Maximum value: 10
Sn+8	Number of stacking levels for the workpieces in the Z direction	Sets the number of workpieces stacked in the Z direction	Maximum value: 10
Sn+9	Direction of stacking	Sets the stacking direction of the workpieces: 1: X→Y→Z 2: Y→X→Z	
…	…	…	…

The operation results that are stored in D are shown below.

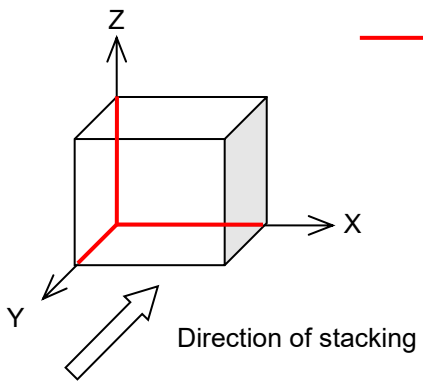
Address	Use	Setting	Notes
Dn, Dn+1	X coordinate for 1st workpiece	X coordinate for 1st workpiece	Setting in µm units
Dn+2, Dn+3	Y coordinate for 1st workpiece	Y coordinate for 1st workpiece	Setting in µm units
Dn+4, Dn+5	Z coordinate for 1st workpiece	Z coordinate for 1st workpiece	Setting in µm units
Dn+6, Dn+7	X coordinate for 2nd workpiece	X coordinate for 2nd workpiece	Setting in µm units
Dn+8, Dn+9	Y coordinate for 2nd workpiece	Y coordinate for 2nd workpiece	Setting in µm units
Dn+10, Dn+11	Z coordinate for 2nd workpiece	Z coordinate for 2nd workpiece	Setting in µm units
…	…	…	…
Dn+m, Dn+m+1	X coordinate for Nth workpiece	X coordinate for Nth workpiece	Setting in µm units
Dn+m+2, Dn+m+3	Y coordinate for Nth workpiece	Y coordinate for Nth workpiece	Setting in µm units
Dn+m+4, Dn+m+5	Z coordinate for Nth workpiece	Z coordinate for Nth workpiece	Setting in µm units

- N refers to the number of stacking workpieces (Number of stacking workpieces in the X direction × Number of stacking workpieces in the Y direction × Number of stacking levels in the Z direction).
- n indicates a given number that is set by the user.

Program example



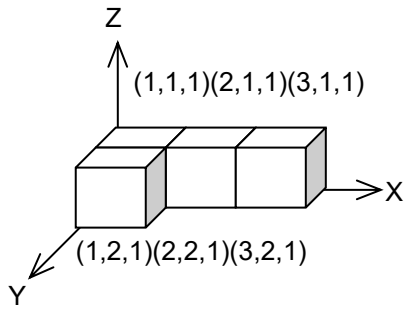
The workpiece dimensions are as follows:



= Length of each direction for the workpiece

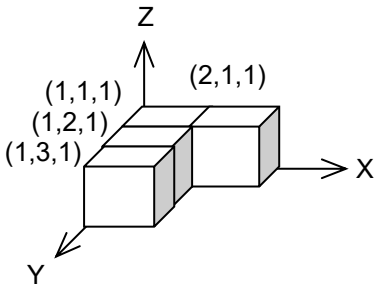
Program example:
Length of X direction is 1 (D0, D1).
Length of Y direction is 1 (D2, D3).
Length of Z direction is 1 (D4, D5).

The workpiece stacking direction is as follows:
1: For X → Y direction



When in the following order (X direction, Y direction and Z direction):
(1, 1, 1) → (2, 1, 1) → (3, 1, 1) →
(1, 2, 1) → (2, 2, 1) → (3, 2, 1) →
(1, 3, 1) → (2, 3, 1) → etc.

2: For Y → X direction

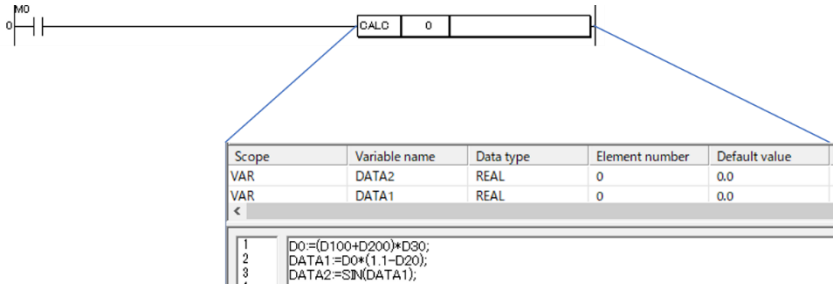


When in the following order (X direction, Y direction and Z direction):
(1, 1, 1) → (1, 2, 1) → (1, 3, 1) →
(2, 1, 1) → (2, 2, 1) → (2, 3, 1) →
(3, 1, 1) → (3, 2, 1) → etc.

11.3.2.15 User function

CALC

The CALC command executes what is written in the ST language in the LD program.



Refer to "Chapter 12 Commands (ST)" for further details about the ST language.
Refer to "7.2.1.4 CALC command input method" in the PLC System Manual (Option) for details on how to input a CALC command.

(NOTE 1) The user can create up to 10 CALC commands for one POU.
(NOTE 2) The user can use up to 1024 characters to input the ST program for one CALC command.
(NOTE 3) The user can only use the parameter variables that are registered in the editing dialog (CALCBOX) for a specific CALC command.
The parameter variables registered in other CALC commands or other POUs cannot be used.
Register the same external variable(s) to each CALC command if common variables are needed in multiple CALC commands.
(NOTE 4) Refer to "12.4 Error message list during ST program conversion" for details about errors displayed in the editing dialog when creating CALC commands.

Function

Command	Description
CALC	Carries out the operation following the ST program (that is set) when the previous operation result is ON.

Usable data type (*1)

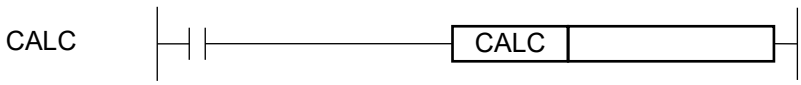
Data type																
BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

Usable OM/constant (*1)

Bit									Word								Constant		Label	Nesting	WL designation	INDEX
X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	Decimal	Hexadecimal	L	N		
○	○	○	○	○	○	○			○	○	○	○			○	○	○	○			○	○

*1: The usable data type/OM varies for each function used in the ST program. Refer to "Chapter 12 Commands (ST)" for further details.

Circuit example



Error code

Refer to "Chapter 12 Commands (ST)".

11.4 Error Message List During LD Program Conversion

The following shows the errors that may trigger during LD program conversion.

Message	Cause	Countermeasure
Invalid label position.	The label (L) is not set at the beginning position on the circuit block.	Set the label at the beginning position of the circuit block.
Ladder is broken.	Ladder (circuit) is broken.	Correct the broken position.
Ladder is shorted.	Ladder (circuit) is shorted.	Correct the shorted position.
Defective point in ladder.	The ladder (circuit) created is invalid due to a reason or cause that is not listed.	Correct the invalid point(s).
Maximum program size will be exceeded.	The POU step number has exceeded the maximum value (High speed task: 5K steps, Standard task: 40K steps).	Correct the program so that the maximum value is not exceeded.
There is an error with the wrapping on the cursor line. Check the number and sequence of wrapping points.	There is a wrapping operation input at the rightmost column, but there is no wrapping connection that is input at the beginning of the next line (rung). Or, the wrapping number is wrong.	Input a wrapping connection on the beginning of the next line (rung), and make sure that the wrapping number is correct.
Data size used in the BOOL parameter variables exceeds the limit: 1024 Data size used in the non-BOOL parameter variables exceeds the limit: 1024 Reduce to the limit or less.	The number of parameter variables used in the CALC command exceeds the number of variables that can be registered in the program overall. • Maximum number of BOOL type variables: 1024 • Maximum number of non-BOOL type variables: 1024	Correct the program so that the maximum number of variables is not exceeded in the program overall.
Data size used in the BOOL external variables exceeds the limit: 1024 Data size used in the non-BOOL external variables exceeds the limit: 1024 Reduce to the limit or less.	The number of external variables used in the CALC command exceeds the number of variables that can be registered in the program overall. • Maximum number of BOOL type variables: 1024 • Maximum number of non-BOOL type variables: 1024	Correct the program so that the maximum number of variables is not exceeded in the program overall.
Data size used for BOOL type parameter variables and return values in functions exceeds the limit: 4096. Data size used for non-BOOL parameter variables and return values in functions exceeds the limit: 4096. Reduce to the limit or less.	For user defined functions/system defined functions called by the CALC commands, the number of parameter variables used in the user defined functions/system defined functions exceeds the number of variables that can be registered in the program overall. • Maximum number of BOOL type variables: 4096 • Maximum number of non-BOOL type variables: 4096	Correct the program so that the maximum number of variables is not exceeded in the program overall.
Number of labels used in the IF statement, CASE statement or RETURN statement exceeds the limit: 1024. Reduce the number of control statements.	Total number of IFs, CASEs and RETURNs in the CALC command exceed 128 in the POU overall.	Correct the program so that the total number of IFs, CASEs and RETURNs used in the CALC command is less than 128 in the POU overall.
There is an error in {1} for the No. {0} CALC command.	There is an invalid program at the corresponding CALC command.	Correct the program for the corresponding CALC command.
There is an error in the No. {0} CALC command.	There is an invalid program at the corresponding CALC command.	Correct the program for the corresponding CALC command.

(This page was intentionally left blank.)

CHAPTER 12

COMMANDS (ST)

12.1 ST Language Specifications
12.2 Control Statements in ST Language
12.3 ST Language Functions
12.4 Error Message List During ST Program Conversion

12.1 ST Language Specifications

The section here describes the requirements when using the PLC Programming software DOO for programming in the ST language.

12.1.1 ST Program Components

The following components are used and combined to create an ST program.

- Operator
- Control statement
- Function
- Variable and OM
- Constant
- Delimiter
- Comment

The user can insert half-width spaces and tabs in between these components.

In the ST language, the user can enter a maximum of 5120 characters in 1 POU when using a high speed control program. A maximum of 40960 characters can be entered when using a standard control program and function.
In addition, the user can enter a maximum of 1024 characters per line. If the user enters more than 1024 characters per line, a conversion error is triggered.

12.1.2 Usable Characters

The user can use half-width alphanumeric characters and symbols in an ST program. The half-width symbols shown from section “12.1.3 Operators” to section “12.1.9 Comments” can be used. Note that Japanese can also be used in the comments.

12.1.2.1 Reserved words and reserved characters

Reserved words and reserved characters are characters and combinations of characters that follow the rules of syntax.
In the ST language, the following words and characters are reserved.

Reserved characters	:=, (,), *, **, /, +, -, <, >, <=, >=, =, <>, | , &, ;, ¥, **[,]**, ., ,, :, #, e, E, $, ', d, D, h, H, m, M, s, S, % and "	
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 and FALSE
	Same character strings as function names and method names	Refer to “12.3.1 List of ST language functions”. Ex) SHR and 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.)

When using reserved words in an ST program, enter them as uppercase letters.
If they are entered as lowercase letters, they may be processed as undefined variables and functions.

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 LD/ST 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.

12.1.3 Operators

The operators that can be used in an ST program are shown. Higher priority is assigned and indicated by a lower number.

Description	Operator	Priority
Parentheses	( )	1
Function	Function (panameter1, parameter 2, …) *1	2
Negation	NOT	3
Exponent	**	4
Multiplication	*	
Division	/	
Remainder	MOD	
Addition	+	5
Subtraction	-	
Comparison	<	6
	>	
	<=	
	>=	
Equality	=	7
Inequality	<>	
Logical conjunction	AND or &	8
Exclusive disjunction	XOR	9
Logical disjunction	OR or |	10
Assignment	:=	11

*1 The format varies depending on the function. Refer to “12.3 ST language functions” for further details.

When operators with the same priority exist together, priority is given to the operator noted on the left side.

Ex) F:=A * B + (C - D) / E;
In the equation noted above, the operations are carried out in the following order.
1: (C-D)
2: A * B
3: 2/E
4: 1 + 3
5: F:= 4

Parentheses cannot be used in the following ways.
- The delimiter ; cannot be included inside the parentheses ( ).
  Ex) (D0:=1;)
- A control statement cannot be included inside the parentheses ( ).
  Ex) (IF M0 THEN)
- When OMs are used for digit-specified access or to specify indexing, only the OMs can be included inside of the parentheses.
  Ex) (M0):4, (D0)IX0

(NOTE) Refer to the “12.1.4 Control statements”, “12.1.6.3 Specifying indexing and digit-specified access for OMs” and “12.1.8 Delimiters” for further details on control statements, digit-specified access, index specification and delimiters.

12.1.4 Control Statements

The following control statements are available. Refer to “12.2 Control statements in ST language” for further details.

- Assignment
- Function call
- IF
- CASE
- FOR DO
- WHILE DO
- REPEAT UNTIL
- RETURN
- EXIT

12.1.5 Functions

Functions are program components that are called from the ST program. There are standard functions available in the PLC Programming software D00.
Refer to “12.3 ST language functions” for details about functions that can be used in an ST program.
In addition, the user can create and use both user defined and system defined functions. Refer to “12.3.3 Create & Use user defined function / system defined function” for details on creating functions.

12.1.6 Variables and OMs

In an ST program, variables and OMs can be used as a “container” to store or hold data.

12.1.6.1 List of data types

The following data types can be used in an ST 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.
When REAL and LREAL variables are used in comparison operations, there may be a difference in the value stored in binary, and therefore, the result may not be as expected.

12.1.6.2 Usable OMs

The following OMs can be used in an ST 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 and SM	BOOL
X, Y, M, BX, BY, LM and SM for specifying 4-, 8-, 12-, 16-bit digits (Ex: M0:8 and M0W)	WORD
D, BDX, BDY, LD, SD and 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

12.1.6.3 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.

12.1.6.4 External variables

When using external variables in an ST program, enter “gv.” in lowercase letters 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)

12.1.6.5 Array variables

Array variables 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 compatible for assigning to array variables and it must be assigned at an array variable with the same element number. Refer to “12.2.2.1 Assignment” 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, the notation would be as follows: Arr**[0]** and Arr**[1]**.
In addition, a variable (excluding arrays) and an OM can also be used to specify an element (Ex: Arr**[VAR]** and Arr**[D0]**). 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 ST program just like other variables (that are not arrays variables).
However, for one conditional expression (such as an IF expression) or calculation formula in a program, the maximum number of array variables that can be used is 256.

Arr1**[0]** := Arr2**[1]** + Arr2**[2]** +・・・+ Arr2**[255]**;

Up to 256 array variables can be used in one calculation formula

(NOTE 1) 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.
(NOTE 2) Even when a line break is inserted in the middle of a formula or expression with an array variable, the number of array variables used is counted up to the delimiter (such as ;) of the formula or expression.

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	Arr1:=0; Arr1:=VAR; Arr1:=D0;
Assigning array variables to data that is not an array	VAR:= Arr1; D0:= Arr1;
Any operations other than assignment	Arr1:=Arr2+Arr3;
Assigning array variables that include data types that cannot be assigned	(Data type of Arr1: INT, Data type of Arr2: STRING) Arr1:=Arr2
Assigning array variables with element numbers that do not match	(Element number of Arr1: 5, Element number of Arr2: 10) Arr1:=Arr2

Item	Notation example (*)
Using array variables in an IF control statement	IF Arr1 THEN (Process) END_IF;
Using array variables in a CASE integer expression or as an integer value	CASE Arr1 OF Arr2:(Process) ELSE END_CASE;
Using array variables as FOR DO variables	FOR Arr1:= 0 TO 10 BY 1 DO (Process) END_FOR;
Using array variables in WHILE DO conditional expressions	WHILE Arr1<100 DO (Control statement) END_WHILE;
Using array variables in REPEAT UNTIL conditional expressions	REPEAT (Process) UNTIL Arr1>100 END_REPEAT;
Using array variables in arguments for functions that do not support arrays	LEN(Arr1);
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.) FUNC1(Arr1,Arr2);
Using another item besides constants, variables or OMs to specify array elements Or, using array variables to specify array elements	Arr1**[1+1]** Arr1**[LEN('123')]** Arr1**[Arr2]**
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	Arr1**[1.23]** Arr1**[VAR_TIME]** (VAR_TIME is a TIME type variable)
Specifying out of range numerical values for elements of variable arrays	Arr1**[-1]** Arr1**[10]** (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.

(NOTE) Refer to “12.2.2 Detailed descriptions of control statements in ST language” for further details on IF, CASE, FOR DO, WHILE DO and REPEAT UNTIL.

12.1.7 Constants

The following constants can be used in an ST program.

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 number	REAL, LREAL	Numerical value is noted To express “10 to ○th 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:Minue:Second]**	DT#2017-10-05-15:51:00
Date	DATE	**[D][#][Year-Month-Day]**	D#2017-10-05
Time of day	TOD	**[TOD][#][Hour:Minue:Second]**	TOD#15:51:00
Time	TIME	**[T][#][Hour:Minute:Second]**	T#12:34:56

* For example, a conversion error is triggered when a letter is entered in lowercase letters like in “16#abcd”.

The following symbols and character code use $ in the notation.

Symbol	Character and character code
$$	$
$'	'
$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 ST 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 ST program, then the value and operation are processed as follows in the 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 (zero) is input and processed for the operation.

Ex 1) When “DATA1” is a constant in ULINT type data
DATA1:=18446744073709551616;
After the operation above is executed, 18446744073709551615 is assigned to DATA1.

Ex 2) When “DATA2” is a constant in LREAL type data
DATA2:=1.79769313486231E+309;
After the operation above is executed, 0 (zero) is assigned to DATA2.
When real constants are used in comparison operations, there may be a difference in the value stored in binary, and therefore, the result may not be as expected.

(NOTE) Refer to “12.2.2.1 Assignment” 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 Control statements in ST language” and “12.3 ST language functions” for details about data types that can be used with each control statement and function when using the ST language.

12.1.8 Delimiters

A semicolon (;) is entered at the end of a control statement or calculation formula.
However, a blank line can also be entered in between calculation formulas.

Ex)
```
IF Data1>100 OR A1<0 THEN
    Data2:=1;
ELSE
    Data2:=0;
END_IF;

Data3:=Data4+Data5;
```

12.1.9 Comments

A comment can be entered in the middle of an ST program.
The part that is enclosed by (* *) is processed as a comment.

Ex)
```
IF A1>100 OR A1<0 THEN
(*If variable A1 is larger than 100 or smaller than 0, then 1 is assigned to the variable B1.*)
    B1:= 1;
ELSE
(*If 0<=A1<=100, then 0 (zero) is assigned to B1.*)
    B1:=0;
END_IF;
```

12.2 Control Statements in ST Language

The chapter here provides a list of control statements and a detailed description.

12.2.1 List of Control Statements in ST Language

The following shows the grammar used with control statements in the PLC Programming software D00.

Item	Control statement format	Description	Step number
Assignment	Variable := Expression;	Expression can be a variable, constant, function call statement or calculation formula using an operator.	3
Function call	Variable := FUNCTION(Parameter 1, Parameter 2,…);	The result of the function execution is input on the left side.	Depends on the function

IF	IF conditional expression 1 THEN
	Process 1;
	ELSIF conditional expression 2 THEN
	Process 2;
	ELSIF conditional expression 3 THEN
	Process 3;
	…
	ELSE
	Process N;
	END_IF;
		The execution processing is selected depending on whether the specified conditional expression (BOOL expression) is true or false.
		If conditional expression 1 is true, process 1 is executed.
		If conditional expression 2 is true, process 2 is executed.
		If conditional expression 3 is true, process 3 is executed.
		…
		If any condition is false, process N is executed.
			IF:1
			ELSIF:4
			THEN:4
			ELSE:7
			END_IF:7

CASE	CASE integer expression OF Integer value 1 : Process 1; Integer value 2 : Process 2; … ELSE Process N; END_CASE;	The executing control statement is selected depending on the specified value for the integer expression. If the result of the integer expression is integer value 1, then process 1 is executed. If the result of the integer expression is integer value 2, then process 2 is executed. … If the result of the integer expression does not match any integer value, then process N is executed.	CASE:0 OF:0 Selected value N: (8*N)-4 ELSE:7 END_CASE:7
FOR DO	FOR variable:= Start value TO end value BY add value DO Process 1; END_FOR;	The variable increases the start value by the add value for the section between FOR and END_FOR, and process 1 execution is repeated until the end value is reached.	FOR:0 TO:0 BY:0 DO:34 END_FOR:10
WHILE DO	WHILE conditional expression DO Process 1; END_WHILE;	Process 1 execution is repeated while the specified conditional expression (BOOL expression) is true.	WHILE:3 DO:8 END_WHILE:5
REPEAT UNTIL	REPEAT Process 1; UNTIL conditional expression END_REPEAT;	Processing is executed and that processing is repeated until the conditional expression (BOOL expression) is true.	REPEAT:5 UNTIL:0 END_REPEAT:10
RETURN	RETURN;	This ends the program. The NC(BUILT IN PLC) executes a repeating program, and the processing restarts from the beginning of the program at the next scan. When RETURN is used in a CALC command in the LD program, it ends the CALC command process.	3
EXIT	EXIT;	EXIT ends the repeat processing for FOR, WHILE and REPEAT.	1

When using control statements in the ST language, the following restrictions apply.
Nesting structure
Nesting in control statements is possible up to 5 levels.

Maximum number of IF, CASE and RETURN statements
IF, CASE and RETURN (*) can be used a total of 1024 times in a program.
When using a user defined function or system defined function in a program, the number of statements used in a user defined function or system defined function is also included in the count.
* RETURN is counted when used in user defined and system defined functions.

12.2.2 Detailed Descriptions of Control Statements in ST Language

12.2.2.1 Assignment

Assignment notation

Variable:= Expression;

The result of the expression is assigned to the variable on the left side. The operator := is used in an assignment statement. However, an error is triggered during conversion if a calculation formula is written on the left side. (Ex: A+B:=C;)

(NOTE) Never assign for input X.

Data type combinations that can be used
All data types can be used on the left and right sides, but the right side needs a data type that can store data on the left side. Refer to the following table for details about motions and data type combinations that can be used on the right and left sides.

Data type on left side	Data type on right side																
	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 0 is assigned if the input value is 0. 1 is assigned if the input value is another value besides 0.
*2 When a real number 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 (zero) 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.

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 (zero), or when the right side is an unsigned value, then the remainder bit (where it is assigned) is filled with a 0 (zero). 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.

(NOTE) An error is triggered at conversion if the combination of data types on the right side and left side are invalid.

In addition, the following OMs can be used.

	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		
Right side	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	○	○	○
Left side	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

Notation example of assignment statement

Ex 1) (*Variable A1 is assigned to E1.*)
E1:= A1;

Ex 2) (*Product of A1*A2 is assigned to E1.*)
E1:= A1*A2;

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 45)

12.2.2.2 Function call

Function call notation
Return value := FUN_NAME(Parameter 1, Parameter 2,…)

A function call statement is executed as follows.
The result (right side) of the FUN_NAME (function name) execution is assigned to return value on the left side.
Variables and OMs can be used for the return value.
Variables, OMs and constants (literals) can be used for parameters.
Commas (,) are used as delimiters between parameters.

Refer to “12.3 ST language functions” for a detailed description related to functions.

Notation example of function call

Ex) OUT1 := ABS(A1); (*Function ABS is called and the operation result is assigned to variable OUT1.*)
D0 := SHR(D0,3); (*Function SHR is called and the operation result is assigned to OM D0.*)

Arguments can be omitted in user defined and system defined functions. When omitting argument input, the default value of the argument specified (when creating the user defined function or system defined function) is used to carry out the operation.

Ex) When all arguments are specified: UserFunc1(In1,In2,In3,Out4);
When the second and third arguments are omitted: UserFunc1(In1,,,Out4);

12.2.2.3 IF

The execution processing is selected depending on whether the specified conditional expression (BOOL expression) is true or false.

IF notation

```
IF conditional expression 1 THEN
        Process 1;
ELSIF conditional expression 2 THEN
        Process 2;
ELSIF conditional expression 3 THEN
        Process 3;
…
ELSE
        Process N;
END_IF;
```

If conditional expression 1 is true, process 1 is executed.
If conditional expression 2 is true, process 2 is executed.
If conditional expression 3 is true, process 3 is executed.
…
If any condition is false, process N is executed.

Omissions are not possible for IF and END_IF.
When there is just one condition, the part between ELSIF and THEN is not needed.
When the condition is false and no-processing is selected, ELSE is not needed.
IF, CASE, FOR, WHILE and REPEAT can be used between IF and END_IF. However, nesting is only possible up to 5 levels.
Use IF and END_IF so they are always a pair.

Refer to “12.2.2.10 Conditional expressions (BOOL expression)” for further details about conditional expressions (BOOL expression).

IF notation examples

<u>Ex 1: When there is one conditional expression</u>

```
IF A1>100 OR A1<0 THEN
        (*If variable A1 is larger than 100 or smaller than 0, then 1 is assigned to the variable B1.*)
        B1:= 1;
ELSE
        (*If 0<=A1<=100, then 0 (zero) is assigned to B1.*)
        B1:=0;
END_IF;
```

Ex 2: When there are multiple conditional expressions

```
IF A1=0 THEN
        (*When A1 is 0 (zero), 0 (zero) is assigned to B1.*)
        B1:= 0;
ELSIF 100>A1 AND A1>0 THEN
        (* When 0<A1<100, 1 is assigned to B1.*)
        B1:=1;
ELSIF 100<=A1 THEN
        (* When A1<=100, 2 is assigned to B1.*)
        B1:=2;
ELSE
        (*In all other cases (when A1<0), -1 is assigned to B1.*)
        B1:=-1;
END_IF;
```

12.2.2.4 CASE

The executing control statement is selected depending on the specified value for the integer expression.

CASE notation

```
CASE integer expression OF
        Integer value 1: Process 1;
        Integer value 2: Process 2;
        …
ELSE
        Process N;
END_CASE;
```

If the result of the integer expression is integer value 1, then process 1 is executed.
If the result of the integer expression is integer value 2, then process 2 is executed.
…
If the result of the integer expression does not match any integer value, then process N is executed.

The following can be used for an “Integer expression”.

- Variable, OM and constant
- Combination of an operation and an operation with a return value (*) (Ex: (D0+1)*LEN('A'))
  (*) Assignment (:=) and MCPY cannot be used because there is no return value.

An “integer expression” must be one of the following data types. However, when using REAL and LREAL, then ELSE processing is carried out.

Data type																
BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

Specify the integer constant for the “integer value”. Constants cannot be specified for REAL, LREAL, TIME, DATE, TOD, DT and STRING types.

Omission is not possible with CASE and END_CASE.
ELSE is not needed to select no-processing when none of the integer values match.
IF, CASE, FOR, WHILE and REPEAT can be used for the processing part. However, nesting is only possible up to 5 levels.
Use CASE and END_CASE so they are always a pair.

Notation example of CASE statement

```
CASE A1 OF
        1:B1:= 100;         (*When A1 is 1, 100 is assigned to B1.*)
        2:B1:= 200;         (*When A1 is 2, 200 is assigned to B1.*)
        3:B1:= 300;         (*When A1 is 3, 300 is assigned to B1.*)
ELSE
        B1:= -1;            (*When A1 is another value, -1 is assigned to B1.*)
END_CASE;
```

12.2.2.5 FOR DO

The repeat processing is executed until the value (increased by the add value for specified variable) reaches the end value.

FOR DO notation
FOR variable:= Start value TO end value BY added value DO
    Process 1;
END_FOR;

Variables and OMs in the following data types can be used for the “Variable” in “Variable:=Start value”.
Constants, variables and OMs in the following data types can be used for the “End value” and “Add value”. Operators and functions cannot be used.
The following data types can be used for variables, end values and add values.

Data type																
BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-

FOR DO is executed as follows. It executes during one scan.

1. The start value is set to the variable.
2. The processing assesses when one of the following conditions is met.

When add value >= 0	Variable value <= End value
When add value < 0	Variable value >= End value

   When a condition is met, the process proceeds to step 3.
   If no condition is met, the FOR DO process ends.
3. “Process 1” is carried out.
4. An add value is added to the variable.
   The process returns to step 2.

(NOTE 1) Set the start value, end value and add value and be careful not to set the processing into an infinite loop.
Ex) WDT error is triggered in the NC(BUILT IN PLC) if the end value is larger than the range that is possible with the variable data type, or if the add value is set to 0 and an infinite loop occurs.

(NOTE 2) When the parameters are set so that the operation continues in NC(BUILT IN PLC) and 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).

Use FOR and END_FOR so they are always a pair.
IF, CASE, FOR, WHILE and REPEAT can be used for “Process 1”. However, nesting is only possible up to 5 levels.

Notation example of FOR DO statement
(*A1 and B1 are INT type variables.*)
(*A1 value is between 0 and 100 with add value of 10, and +1 is added to B1.*)
FOR A1:=0 TO 100 BY 10 DO
    B1:= B1+1;
END_FOR;

12.2.2.6 WHILE DO

Control statement execution is repeated while the specified conditional expression (BOOL expression) is true.

WHILE DO notation

```
WHILE conditional expression DO
        Control statement;
END_WHILE;
```

WHILE DO is executed as follows. It executes during one scan.

1. If the conditional expression is true, then the control statement executes.
   If the conditional expression is not true, then the WHILE processing ends.
2. After the control statement is executed, the processing assesses again whether the conditional expression is true. If the conditional expression is true, then the control statement executes.
   This repeats until the conditional expression is no longer true.

Refer to “12.2.2.10 Conditional expressions (BOOL expression)” for further details about conditional expressions (BOOL expression).

(NOTE 1) Set the conditional expression and be careful not to set the processing into an infinite loop.
WDT error is triggered in the NC (BUILT IN PLC) if an infinite loop occurs.

(NOTE 2) When the parameters are set so that the operation continues in NC(BUILT IN PLC) and 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).

Use WHILE and END_WHILE so they are always a pair.
IF, CASE, FOR, WHILE and REPEAT can be used for the “Control statement”. However, nesting is only possible up to 5 levels.

WHILE DO notation example

```
(*A1 is an INT type variable.*)
WHILE A1<100 DO
        (*A1 increases incrementally until A1 reaches 100 or more.*)
        A1:= A1+1;
END_WHILE;
```

12.2.2.7 REPEAT UNTIL

Processing is executed and that processing is repeated until the conditional expression (BOOL expression) is true.

REPEAT UNTIL notation
REPEAT
    Process 1;
UNTIL conditional expression
END_REPEAT;

REPEAT UNTIL is executed as follows. It executes during one scan.
1. Process 1 is executed.
2. If the conditional expression is false, processing returns to step 1.
   If the conditional expression is true, the REPEAT UNTIL processing ends.

After “Process 1” is executed, the “Conditional expression” is assessed. As a result, “Process 1” is always executed once.
Refer to “12.2.2.10 Conditional expressions (BOOL expression)” for further details about conditional expressions (BOOL expression).

(NOTE 1) Set the conditional expression and be careful not to set the processing into an infinite loop.
WDT error is triggered in the NC(BUILT IN PLC) if an infinite loop occurs.

(NOTE 2) When the parameters are set so that the operation continues in NC(BUILT IN PLC) and 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).

Omission is not possible with REPEAT, UNTIL and END_REPEAT. Always use them as a set.
IF, CASE, FOR, WHILE and REPEAT can be used between REPEAT and UNTIL. However, nesting is only possible up to 5 levels.

A blank line or a comment line cannot be input between the “UNTIL conditional expression” and END_REPEAT. If a comment is needed between the conditional expression and END_REPEAT, then input a comment on the same line as the conditional expression.

REPEAT UNTIL notation example
(*A1 is an INT type variable.*)
REPEAT
    (*A1 value increases incrementally until A1 exceeds 100.*)
    A1:= A1+1;
UNTIL A1 > 100
END_REPEAT;

12.2.2.8 RETURN

RETURN ends the program.
The NC(BUILT IN PLC) executes a repeating program, and the processing restarts from the beginning of the program at the next scan.
When RETURN is used in a CALC command in the LD program, it ends the CALC command process.
When using RETURN in a user defined function or system defined function, that process for the user defined function or system defined function ends.

RETURN notation
RETURN;

RETURN cannot be used between FOR DO and END_FOR, WHILE DO and END_WHILE, as well as REPEAT and UNTIL. A conversion error is triggered at conversion when RETURN is inserted between FOR DO and END_FOR, WHILE DO and END_WHILE, as well as REPEAT and UNTIL.

RETURN notation example

```
A1:=A1+1;
RETURN;
B1:=B1+1; (*Processing written after RETURN is not executed.*)
```

12.2.2.9 EXIT

EXIT ends the repeat processing for FOR, WHILE and REPEAT.

EXIT notation
EXIT;

EXIT notation example

```
(*A1 and B1 are INT type variables.*)
FOR A1:= 0 TO 100 BY 1 DO
        B1:= B1+2;
        IF B1 >50 THEN
            (*FOR process ends when B1 exceeds 50.*)
            EXIT;
        END_IF;
END_FOR;
```

12.2.2.10 Conditional expressions (BOOL expression)

The following can be used in the “Conditional expressions (BOOL expression)” for IF, WHILE DO and REPEAT UNTIL.

- Variable
- OM
- Combination of an operation and an operation with a return value (*) (Ex: (D0+1)*LEN('A'))

(*) Assignment (:=) and MCPY cannot be used because there is no return value.

<u>True conditions for BOOL expression</u>

BOOL expressions are assessed as true or false based on the variable and OM value, or based on the data type of the operation result.

Data type	True	False
BOOL	TRUE(1)	FALSE(0)
BYTE, WORD, DWORD, INT, DINT, LINT, UINT, UDINT, ULINT, REAL, LREAL	Value other than 0 *1	0
TIME	Value other than T#000:00:00	T#000:00:00
DATE	Value besides default value	Default value
TOD	Value other than TOD#00:00:00	TOD#00:00:00
DT	Value besides default value	Default value
STRING	1 character or more	No characters

*1 When REAL or LREAL is used, infinity and non-numbers are also included.

12.3 ST Language Functions

The chapter here provides a list of ST language functions and detailed descriptions.

12.3.1 List of ST Language Functions

The following shows a list of ST language functions that come standard in the PLC Programming software D00.

Category	Command	Expression in ST	Processing	Step (*3)
Convert data type	input_TO_output	Out := input_TO_output(In);	Convert data type	4
	input_TO_BCD	Out := input_TO_BCD(In);	Convert numerical value → BCD	4
	BCD_TO_output	Out := BCD_TO_output (In);	Convert BCD → numerical value	4
Copy binary	input_TO_output	Out := input_TO_output(In);	Copy bit stream	4
Arithmetic operation	ADD	Out := In1 + In2 … + InN;	Addition	4*(N-1)
	SUB	Out := In1 - In2;	Subtraction	4
	MUL	Out := In1 * In2 … * InN;	Multiplication	4*(N-1)
	DIV	Out := In1 / In2;	Division	4
	MOD	Out := In1 MOD In2;	Remainder	12
	EXPT	Out := In1 ** In2;	Exponentiation	4
	MOVE (※1)	Out := In;	Transfer	3
Numerical operation	ABS	Out := ABS(In);	Absolute value	4
	SQRT	Out := SQRT(In);	Square root	4
	LN	Out := LN(In);	Natural logarithm	4
	LOG	Out := LOG(In);	Common logarithm	4
	EXP	Out := EXP(In);	Exponential function	4
	SIN	Out := SIN(In);	Sine value (sin)	4
	COS	Out := COS(In);	Cosine value (cos)	4
	TAN	Out := TAN(In);	Tangent value (tan)	4
	ASIN	Out := ASIN(In);	Arcsine value ($\sin^{-1}$)	4
	ACOS	Out := ACOS(In);	Arccosine value ($\cos^{-1}$)	4
	ATAN	Out := ATAN(In);	Arctangent ($\tan^{-1}$)	4
Bit shift	SHR	Out := SHR(In1, In2);	Bit shift to the right	4
	SHL	Out := SHL(In1, In2);	Bit shift to the left	4
	ROR	Out := ROR(In1, In2);	Bit rotation to the right	4
	ROL	Out := ROL(In1, In2);	Bit rotation to the left	4
Bit computation	AND	Out := In1 AND In2 …AND InN; or Out := In1 & In2 … & InN;	Logical conjunction	4*(N-1)
	OR	Out := In1 OR In2; or Out := In1 | In2;	Logical disjunction	4*(N-1)
	XOR	Out := In1 XOR In2;	Exclusive disjunction	4*(N-1)
	NOT	Out := NOT In1;	Negation	3
Select	SEL	Out:= SEL(G, In0, In1);	Select one from IN0 and IN1	19
	MAX	Out:= MAX(In1, In2, … , InN);	Select maximum value	4+(N-1)*10
	MIN	Out:= MIN(In1, In2, … , InN);	Select minimum value	4+(N-1)*10
	LIMIT	Out:= LIMIT(In, MN, MX);	Limit using upper/lower limit value	5
	MUX	Out:= MUX(K, In0, In1, … ,InN);	Multiplexer	21+(N-1)*7
Comparison	GT	Out:= GT(In1, In2, … , InN);	Size comparison (>)	11+(N-1)*6
	GE	Out:= GE(In1, In2, … , InN);	Size comparison (>=)	11+(N-1)*6
	LT	Out:= LT(In1, In2, … , InN);	Size comparison (<)	11+(N-1)*6
	LE	Out:= LE(In1, In2, … , InN);	Size comparison (<=)	11+(N-1)*6
	EQ	Out:= EQ(In1, In2, … , InN);	Equal (=)	11+(N-1)*3
	NE	Out:= NE(In1, In2);	Not equal (<>)	4

Category	Command	Expression in ST	Processing	Step (*3)
Character string	LEN	Out := LEN(In);	Character string length detection	4
	LEFT	Out := LEFT(In, L);	Extract character string data from the left	5
	RIGHT	Out := RIGHT(In, L);	Extract character string data from the right	5
	MID	Out := MID(In, L, P);	Extract character string data from specified position	6
	CONCAT	Out := CONCAT(In1, In2);	Concatenate character strings	5
	INSERT	Out := INSERT(In1, In2, P);	Insert character string	6
	DELETE	Out := DELETE(In, L, P);	Delete character string	6
	REPLACE	Out := REPLACE(In1, In2, L, P);	Replace character string	7
	FIND	Out:=FIND(In1, In2);	Find character string	4
Time data type	ADD_TIME	Out := In1 + In2;	Add time	4
	ADD_DT_TIME	Out := In1 + In2;	Add date and time	4
	ADD_TOD_TIME	Out := In1 + In2;	Add time of day	4
	SUB_TIME	Out := In1 - In2;	Subtract time	4
	SUB_DATE_DATE	Out := In1 - In2;	Subtract date	4
	SUB_TOD_TIME	Out := In1 - In2;	Subtract time from time of day	4
	SUB_TOD_TOD	Out := In1 - In2;	Subtract time of day from time of day	4
	SUB_DT_TIME	Out := In1 - In2;	Subtract time from date and time	4
	SUB_DT_DT	Out := In1 - In2;	Subtract date and time from date and time	4
	MUL_TIME	Out := In1 * In2;	Multiply time	4
	DIV_TIME	Out := In1 / In2;	Divide time	4
	CONCAT_DATE_TOD	Out := CONCAT_DATE_TOD (In1, In2);	Concatenate date and time of day	5
Extension (such as OM operations)	MCPY	MCPY(S,D,n);	Copy data from OM/variable to OM	4
	TON	TON(IN,PT,N,ET);	Timer OM count	14
User defined	Depends on the function data created by the user. (*2)		Execute function created by the user	7 or greater Varies depending on number of arguments.
System defined	Depends on the function data created by the user. (*2)		Execute function created by the user	7 or greater Varies depending on number of arguments.

*1 Refer to “12.2.2.1 Assignment” for details about MOVE.
*2 Refer to “12.3.3 Create & Use user defined function / system defined function” for details about user defined and system defined functions.
*3 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
- Step numbers needed for assignment to the left side

12.3.2 Detailed Description of ST Language Functions

The following provides detailed descriptions about the ST language 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.

12.3.2.1 Data type conversion function

input_TO_output

Function

Command	Notation	Description
input_TO_output	Out:= input_TO_output(In);	Converts the data type. The data type before conversion is entered for input, and the data type after conversion is entered for output. The data before conversion is “In”, and the data after conversion is “Out”.

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 (zero), then the value is converted to 0 (zero). If the value is not 0 (zero), 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 (zero), 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 “+” or “-”, it is processed as a 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 “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 are invalid.

Description of parameter

Input	Data type before conversion
Output	Data type after conversion
In	Data before converting type
Out	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
In	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	○	○
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	○	○	-	○

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		
In	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○*1	○
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

*1 8-, 16- and 32-bit can only be specified.

Usage example

BOOL type variable A1 is converted to a BYTE type and assigned to variable A2.

A2:=BOOL_TO_BYTE(A1);

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 45)

input_TO_BCD

Function

Command	Notation	Description
input_TO_BCD	Out := input_TO_BCD(In);	Converts an integer (In) to a BCD value. The data type before conversion is specified for input.

(NOTE 1) Store values that can be converted to BCD in the input value (In).
Ex) 0 to 9999 for an INT type

(NOTE 2) An error is triggered at conversion when the data type specified in input does not match the data type for the input value (In).
Ex) "Out := BOOL_TO_BCD(Data1);" is input when Data1 is an INT type variable.

(NOTE 3) Refer to "12.2.2.1 Assignment" for details about how to assign the operation results to "Out".

Description of parameter

input	Input data type
In	Value before conversion
Out	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
In	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○*1	○
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

*1 8-, 16- and 32-bit can only be specified.

Usage example

INT type variable A1 is converted to a BCD type and assigned to variable A2.

Ex) When 1985 is assigned to A1, 16#1985 is assigned to the output value for A2.

A2:= INT_TO_BCD(A1);

BUILT IN PLC execution error

When a numerical value that cannot be converted to BCD is set at the conversion source. (Error code 10)

BCD_TO_output

Function

Command	Notation	Description
BCD_TO_output	Out := BCD_TO_output(In);	Converts a BCD value (In) to an integer value. The data type after conversion is specified for output.

(NOTE 1) Store BCD value in the input value (In).

(NOTE 2) Refer to “12.2.2.1 Assignment” for details about how to assign the operation results to “Out”.

Description of parameter

input	Data type after conversion
In	Data before conversion
Out	Data after conversion

Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

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.

```
Output1:=BCD_TO_INT(Input1);
```

BUILT IN PLC execution error

When a value that is not a BCD value is specified for the conversion source. (Error code 10)

12.3.2.2 Binary copy function

input_TO_output

Function

Command	Notation	Description
input_TO_output	Out:= input_TO_output(In);	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”.

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 at copy source	Data type at copy destination																
	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 are 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.
Out:=REAL_TO_DWORD(In);
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
In	Data at copy source
Out	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
In	○	○	○	-	-	-	-	-	-	-	○	-	-	-	-	-	-
Out	○	○	○	○	-	-	-	-	-	-	○	-	-	-	-	-	-

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		
In	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○*1	○
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

*1 8-, 16- and 32-bit can only be specified.

Usage example

The bit stream for the REAL type variable (A1) is copied to the DWORD type variable (A2).

A2:= REAL_TO_DWORD (A1);

BUILT IN PLC execution error

None

12.3.2.3 Arithmetic operation function

ADD

Function

Command	Notation	Description
ADD	Out := In1 + In2 + … +InN;	Performs an addition operation for the input values (In1, In2,…).

(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 "Out" data type.
For example, when 1 is added to the INT type value 32767 and the operation result is assigned to an INT type, the notation is as follows:
16#7FFF(=32767)+16#0001(1)=16#8000
16#8000 is expressed as -32768 in INT and is assigned to "Out".

(NOTE 2) Refer to "12.2.2.1 Assignment" for details when the data types for the operation result and the "Out" data are different.

Description of parameter

In1 to N	Add value
Out	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
In	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	○	-	○
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	○	-	○

Usage example

Variables VAR1 and VAR2 are added and then assigned to VAR3.

```
VAR3:=VAR2+VAR1;
```

BUILT IN PLC execution error

None

SUB

Function

Command	Notation	Description
SUB	Out := In1 - In2;	Performs input subtraction. The value (In2) is subtracted from the other value (In1), 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 "Out" data type.
For example, when In1 is -32768 in INT, In2 is 1 in INT and the "Out" is an INT type, then 32767 is calculated for "Out". (The INT type is 16-bit, and therefore the result is 32767, which can be expressed in 16-bit.)

(NOTE 2) Refer to "12.2.2.1 Assignment" for details when the data types for the operation result and the "Out" data are different.

Description of parameter

In1	Subtraction value (minuend)
In2	Subtraction value (subtrahend)
Out	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
In	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○

Usage example

When variable A1 is subtracted from variable A2, the result (difference) is assigned to variable A3.

Ex) 3:=4-1;

A3:=A2-A1;

BUILT IN PLC execution error

None

12

MUL

Function

Command	Notation	Description
MUL	Out := In1 * In2* …* InN;	Performs a multiplication operation of the input values (In1, In2,…).

(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 "Out" data type.
For example, when the operation result for 32767 (INT type) × 2 (INT type) is assigned to an INT type, the operation is as follows:
32767 × 2 = 65534
-2 expressed as an INT type is the operation result.

(NOTE 2) Refer to "12.2.2.1 Assignment" for details when the data types for the operation result and the "Out" data are different.

Description of parameter

In1 to InN	Multiplication value (factor)
Out	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
In	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

Usage example

The multiplication of variable A2 and variable A1 is assigned to variable A3.

Ex) When the input values are 3 and 2, then 6 is output.

```
A3:=A2*A1;
```

BUILT IN PLC execution error

None

DIV

Function

Command	Notation	Description
DIV	Out := In1 / In2;	Performs input division. The value (In1) is divided by the value (In2), and that result is output to the output value.

(NOTE 1) It is OK even if different data types are used for In1 and In2. 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 “Out” data type.

(NOTE 4) Refer to “12.2.2.1 Assignment” for details when the data types for the operation result and the “Out” data are different.

Description of parameter

In1	Division value (dividend)
In2	Division value (divisor)
Out	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
In1	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-
In2	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In1	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
In2	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

Usage example

Variable A2 is divided by variable A1, and the result is assigned to variable A3.
For example, when the dividend is 16 and the divisor is 4, then 4 is assigned to the output value.

A3:= A2/A1;

BUILT IN PLC execution error

When the divisor is 0 in the division command. (Error code 11)

MOD

Function

Command	Notation	Description
MOD	Out: = In1 MOD In2;	Calculates the remainder. The value (In1) is divided by the value (In2), and that result (remainder) is output to the output value.

(NOTE) Refer to “12.2.2.1 Assignment” for details when the data types for the operation result and the “Out” data are different.

Description of parameter

In1	Division value (dividend)
In2	Division value (divisor)
Out	Remainder

Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
In2	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In1	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
In2	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

Usage example

The remainder of variable A1 and variable A2 is assigned to variable A3.
For example, when the dividend (A1) is 15 and the divisor (A2) is 4, then 3 is assigned to the output value.

A3:=A1 MOD A2;

BUILT IN PLC execution error

When the divisor is 0 in the division command. (Error code 11)

EXPT

Function

Command	Notation	Description
EXPT	Out := In1 ** In2;	Performs exponentiation with a base value (In1) and an exponent (In2), and outputs 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” for details when the data types for the operation result and the “Out” data are different.

Description of parameter

In1	Base
In2	Exponent
Out	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
In1	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-
In2	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
In2	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

*1 Real numerals can only be specified.

Usage example
A1 to the power of 3 is assigned to B1.

B1:= A1**3;

BUILT IN PLC execution error
None

12.3.2.4 Numerical value operation function

ABS

Function

Command	Notation	Description
ABS	Out := ABS(In);	Calculates the absolute value of the input value (In).

(NOTE 1) Refer to "12.2.2.1 Assignment" for details when the data types for the operation result and the "Out" 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

In	Input value (value to be calculated)
Out	Absolute value

Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In	-	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In	○*1	○*1	○*1	○*1	○*1	○*1	○*1	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

*1 Possible only when digits are specified.

Usage example

Absolute value of variable A1 is assigned to variable A2.

For example, when A1 is -3, then 3 is assigned to A2.

A2:= ABS(A1);

BUILT IN PLC execution error

None

SQRT

Function

Command	Notation	Description
SQRT	Out := SQRT(In);	Calculates the square root of the input value (In).

(NOTE 1) When the input value is negative, -1.#QNAN is output to “Out”.

(NOTE 2) Refer to “12.2.2.1 Assignment” for details when the data types for the operation result and the “Out” data are different.

Description of parameter

In	Input value (value to be calculated)
Out	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
In	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

Usage example

Square root of A1 is assigned to A2.

Ex) When A1 is 4, then 2 is assigned to A2.

A2:= SQRT(A1);

BUILT IN PLC execution error

None

LN

Function

Command	Notation	Description
LN	Out: = LN(In);	Calculates the natural logarithm of the input value (In).

(NOTE 1) When the input value is a negative number, 0 (zero), infinity, or a non-number, the output value is as follows:

- Negative number: Non-number (-1.#QNAN)
- 0 (zero): 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" for details when the data types for the operation result and the "Out" data are different.

Description of parameter

In	Input value (real numeral to be calculated)
Out	Natural logarithm

Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

Usage example

Natural logarithm of A1 is assigned to A2.

Ex) When A1:=3, then 1.098612 is assigned to A2.

A2:=LN(A1);

BUILT IN PLC execution error

None

12

LOG

Function

Command	Notation	Description
LOG	Out: = LOG(In);	Calculates the floating-point natural logarithm of the input value (In).

(NOTE 1) When the input value is a negative number, 0 (zero), infinity, or a non-number, the output value is as follows:

- Negative number: Non-number (-1.#QNAN)
- 0 (zero): 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” for details when the data types for the operation result and the “Out” data are different.

Description of parameter

In	Input value (real numeral to be calculated)
Out	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
In	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

Usage example

Floating-point natural logarithm of A1 is assigned to A2.
For example, when A1 is 10, then 1 is assigned to A2.

```
A2:=LOG(A1);
```

BUILT IN PLC execution error

None

12

EXP

Function

Command	Notation	Description
EXP	Out: = EXP(In);	Calculates the natural exponent (e^{In}).

(NOTE 1) When the input value is infinity or a non-number, the output value is as follows:
- Infinity (+∞): Infinity (+∞) (1.#INF)
- Infinity (-∞): 0 (zero)
- Non-number: Non-number (±1.#QNAN)

(NOTE 2) Refer to "12.2.2.1 Assignment" for details when the data types for the operation result and the "Out" data are different.

Description of parameter

In	Floating-point real numeral
Out	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
In	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

Usage example

Natural exponent of A1 is assigned to A2.

Ex) When A1:= 3, then the result of e^3 is assigned to A2.

A2:=EXP(A1);

BUILT IN PLC execution error

None

SIN

Function

Command	Notation	Description
SIN	Out: = SIN(In);	Calculates the sine (sin) of the input value.

(NOTE 1) Enter the input value in radians.

(NOTE 2) When the input value is infinity, the output value is an undefined value (-1#IND). When the input value is a non-number, the output value is a non-number (±1#QNAN).

(NOTE 3) Refer to "12.2.2.1 Assignment" for details when the data types for the operation result and the "Out" data are different.

Description of parameter

In	Input value
Out	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
In	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

Usage example

Sine (sin) of A1 is assigned to A2.

When A1 is 1.570796(π/2), then 1 is assigned to A2.

A2:=SIN(A1);

BUILT IN PLC execution error

None

COS

Function

Command	Notation	Description
COS	Out: = COS(In);	Calculates the cosine (cos) of the input value.

(NOTE 1) Enter the input value in radians.

(NOTE 2) When the input value is infinity, the output value is an undefined value (-1#IND). When the input value is a non-number, the output value is a non-number (±1#QNAN).

(NOTE 3) Refer to "12.2.2.1 Assignment" for details when the data types for the operation result and the "Out" data are different.

Description of parameter

In	Input value
Out	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
In	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

Usage example

Cosine (cos) of A1 is assigned to A2.

For example, when A1 is 0, then 1 is assigned to A2.

```
A2:=COS(A1);
```

BUILT IN PLC execution error

None

TAN

Function

Command	Notation	Description
TAN	Out: = TAN(In);	Calculates the tangent (tan) of the input value.

(NOTE 1) Enter the input value in radians.

(NOTE 2) When the input value is infinity, the output value is an undefined value (-1#IND). When the input value is a non-number, the output value is a non-number (±1#QNAN).

(NOTE 3) Refer to "12.2.2.1 Assignment" for details when the data types for the operation result and the return value are different.

Description of parameter

In	Input value
Out	Operation result 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
In	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

Usage example

Tangent (tan) of variable A1 is assigned to variable A2.

For example, when A1 is 0, then 0 is assigned to A2.

A2:=TAN(A1);

BUILT IN PLC execution error

None

ASIN

Function

Command	Notation	Description
ASIN	Out := ASIN(In);	Calculates the arcsine ($\sin^{-1}$) of the input value.

(NOTE 1) When the input range is between -1.0 and 1.0, the output result is a non-number (±1.#QNAN).
(NOTE 2) The return value is in radians.
(NOTE 3) Refer to "12.2.2.1 Assignment" for details when the data types for the operation result and the "Out" data are different.

Description of parameter

In	Input value
Out	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
In	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

Usage example

Operation value of variable A1 arcsine is assigned to variable A2.
For example, when A1 is 1, then 1.570796(π/2) is assigned to A2.

A2:=ASIN(A1);

BUILT IN PLC execution error

None

ACOS

Function

Command	Notation	Description
ACOS	Out: = ACOS(In);	Calculates the arccosine ($\cos^{-1}$) of the input value.

(NOTE 1) When the input range is between -1.0 and 1.0, the output result is a non-number (±1.#QNAN).
(NOTE 2) The return value is in radians.
(NOTE 3) Refer to "12.2.2.1 Assignment" for details when the data types for the operation result and the "Out" data are different.

Description of parameter

In	Input value range
Out	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
In	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

Usage example

Arccosine ($\cos^{-1}$) of variable A1 is assigned to variable A2.
For example, when A1 is 0, then 1.570796(π/2) is assigned to A2.

A2:=ACOS(A1);

BUILT IN PLC execution error

None

12

ATAN

Function

Command	Notation	Description
ATAN	Out: = ATAN(In);	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 return value is in radians.
(NOTE 3) Refer to "12.2.2.1 Assignment" for details when the data types for the operation result and the "Out" data are different.

Description of parameter

In	Input value
Out	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
In	-	-	-	-	-	-	-	-	-	-	○	○	-	-	-	-	-
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

Usage example

Arctangent ($\tan^{-1}$) of variable A1 is assigned to variable A2.
For example, when A1 is 0, then 0 is assigned to A2.

```
A2:=ATAN(A1);
```

BUILT IN PLC execution error

None

12.3.2.5 Bit shift function

SHL

Function

Command	Notation	Description
SHL	Out := SHL(In1, In2);	Shifts In1 to the left only for the specified bit number (In2), and that value is output to the output value.

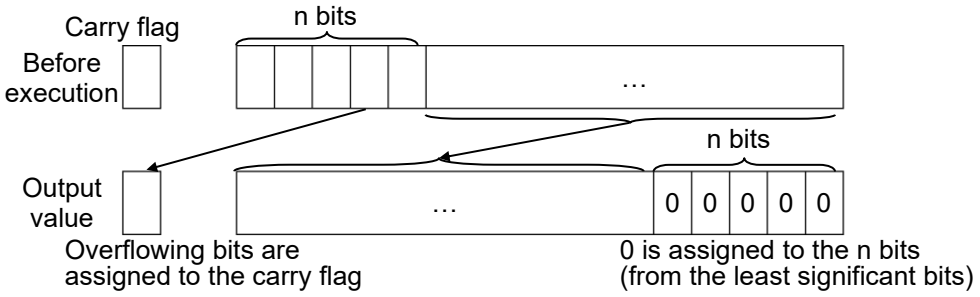
(NOTE 1) Specify a positive integer for the bit number that shifts.
However, when 0 is specified as the bit number that shifts, In1 is output.
When a negative number is specified as the bit number that shifts, 0 is output.

(NOTE 2) In contrast with the SHL command in the LD language, the variable / OM value specified for In1 does not change.

(NOTE 3) Similar to the SHL command in the LD language, in order to shift the variable / OM value specified for In1, the same variable / OM is specified to "Out".
Ex) M4:8:=SHL(M4:8,2);

Operation

When the SHL command is executed, the n bits are shifted from the low-order to the high-order, and that value is output.



Description of parameter

In1	Data before the bit shift
In2	Bit number that shifts to the left
Out	Result after bit shift

Usable data type

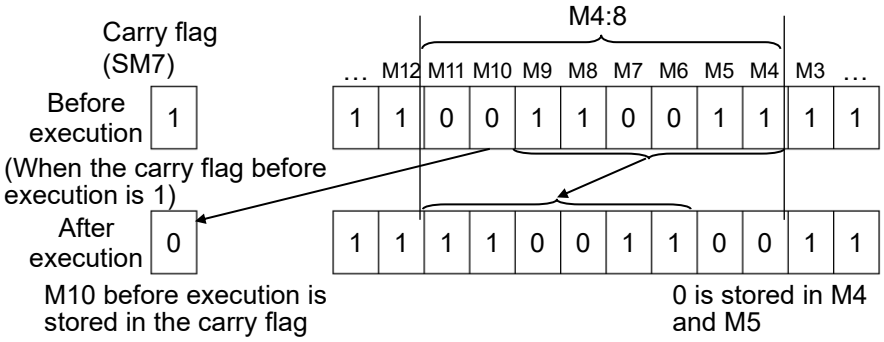
	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
In2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In1	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
In2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

Usage example
Value for OM M4:8 shifts 2 bits to the left.

M4:8:=SHL(M4:8,2);



BUILT IN PLC execution error
None

<u>SHR</u>
Function

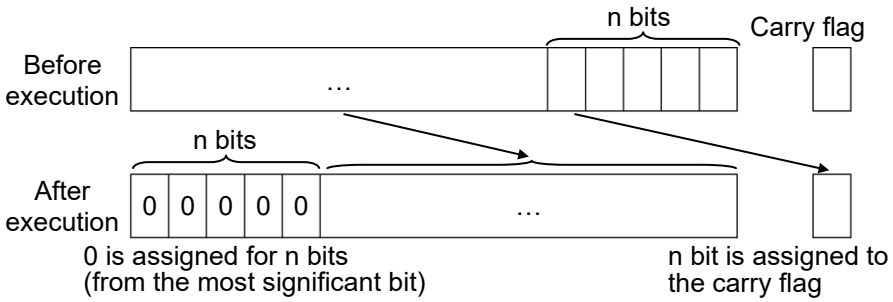
Command	Notation	Description
SHR	Out := SHR(In1, In2) ;	Shifts In1 to the right only for the specified bit number (In2), and that value is output to the output value.

(NOTE 1) Specify a positive integer for the bit number that shifts.
However, when 0 is specified as the bit number that shifts, In1 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 In1 does not change.

(NOTE 3) Similar to the SHR command in the LD language, in order to shift the variable / OM value specified for In1, the same variable / OM is specified to “Out”.
Ex) M4:8:=SHR(M4:8,2);

Operation
When the SHL command is executed, the n bits are shifted from the high-order to the low-order, and that value is output.



Description of parameter

In1	Data before the bit shift
In2	Bit number that shifts to the right
Out	Result after bit shift

Usable data type

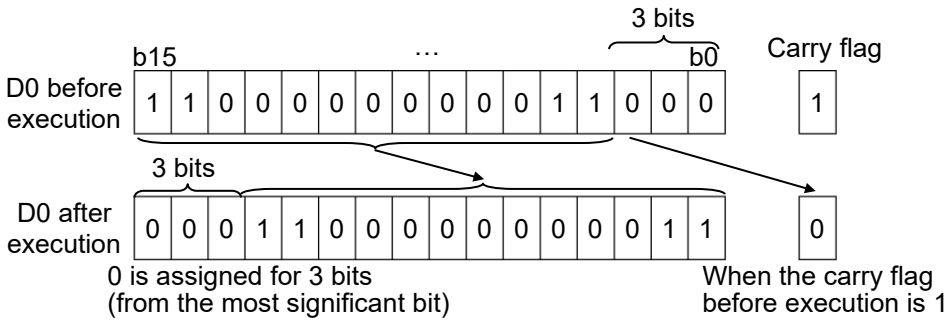
	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
In2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In1	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
In2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

Usage example
Value for OM D0 shifts 3 bits to the right.

D0:=SHR(D0,3);



BUILT IN PLC execution error
None

ROR

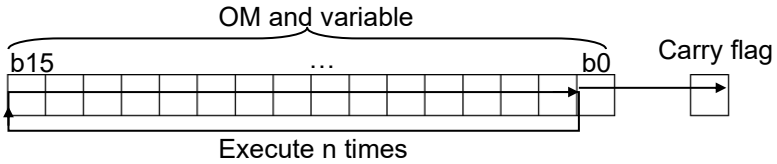
Function

Command	Notation	Description
ROR	Out := ROR(In1,In2);	Rotates the In1 data to the right only for the specified bits (In2), 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 In1 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 ROR command in the LD language, the variable / OM value specified for In1 does not change.
(NOTE 5) Similar to the ROR command in the LD language, in order to shift the variable / OM value specified for In1, the same variable / OM is specified to "Out".
Ex) D0:=ROR(D0,3);

Operation

When the ROR command is executed, each bit is rotated to the left from the high-order bit to the low-order bit, and that value is output.
The value for the least significant bit is assigned to the carry flag.



Description of parameter

In1	Data before the bit shift
In2	Bit number that rotates to the right
Out	Data after rotation

Usable data type

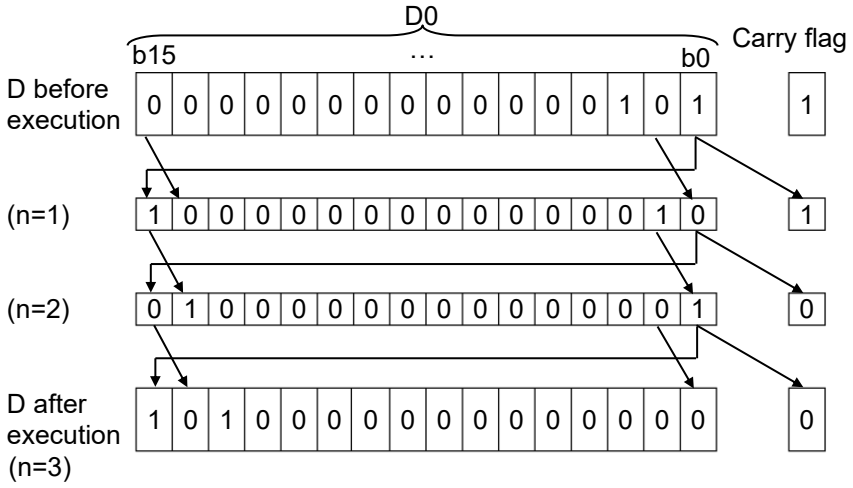
	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1	-	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
In2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In1	-	-	-	-	-	-	-	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○
In2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

Usage example
D0 is rotated 3 bits to the right from the high-order bits to the low-order bits and is assigned to D0.

D0:=ROR(D0,3);



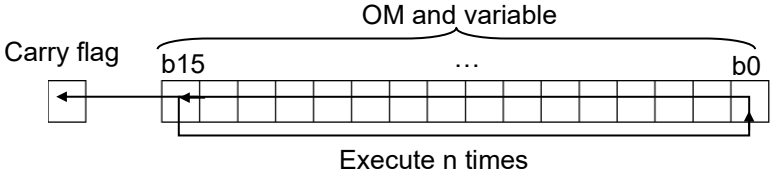
BUILT IN PLC execution error
None

<u>ROL</u>
Function

Command	Notation	Description
ROL	Out := ROL(In, In2);	Rotates the In1 data to the left only for the specified bits (In2), 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 In1 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 In1 does not change.
(NOTE 5) Similar to the ROL command in the LD language, in order to shift the variable / OM value specified for In1, the same variable / OM is specified to “Out”.
Ex) D0:=ROL(D0,3);

Operation
When the ROL command is executed, each bit is rotated to the left from the low-order to the high-order, and that value is output.
The value for the least significant bit is assigned to the carry flag.



Description of parameter

In1	Data before the bit shift
In2	Bit number that rotates to the left
Out	Data after rotation

Usable data type

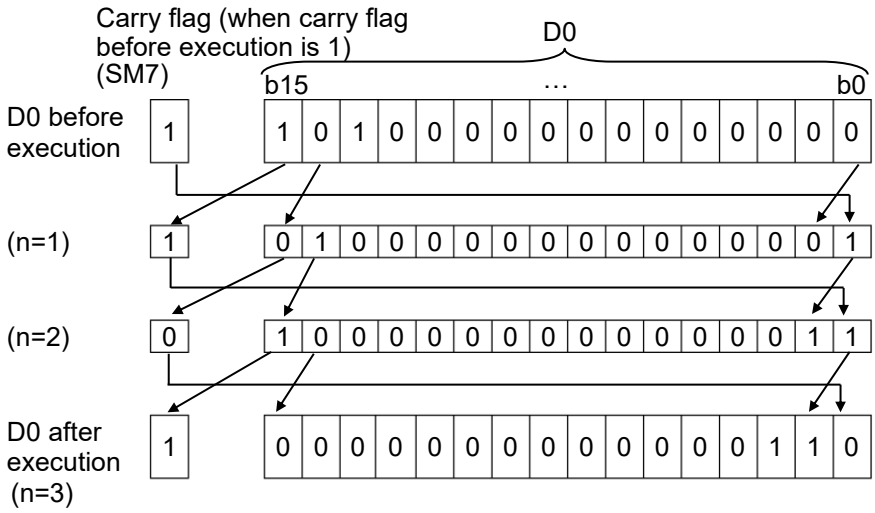
	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1	-	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
In2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In1	-	-	-	-	-	-	-	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○
In2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

Usage example
D0 is rotated 3 bits to the left and is assigned to D0.

D0:=ROL(D0,3);



BUILT IN PLC execution error
None

12.3.2.6 Bit operation function

AND

Function

Command	Notation	Description
AND	Out := In1 AND In2...AND InN; or Out: = In1 & In2 & ...& InN;	Performs a logical conjunction (AND) operation for the input values.

(NOTE) Refer to “12.2.2.1 Assignment” for details when the data types for the operation result and the “Out” data are different.

Description of parameter

In1 to N	Input value
Out	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
In	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

Usage example

The logic 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.

VAR4:= VAR1 AND VAR2 AND VAR3;

BUILT IN PLC execution error

None

OR

Function

Command	Notation	Description
OR	Out := In1 OR In2 OR ...OR InN; or Out: =In 1| In 2 | ...| InN;	Performs a logical disjunction (OR) operation for the input values.

(NOTE) Refer to “12.2.2.1 Assignment” for details when the data types for the operation result and the “Out” data are different.

Description of parameter

In1 to N	Input value
Out	Result of logical disjunction (OR)

Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

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.

```
VAR3:=VAR1 OR VAR2;
```

BUILT IN PLC execution error

None

XOR

Function

Command	Notation	Description
XOR	Out := In1 XOR In2 XOR ... XOR InN;	Performs an exclusive disjunction (XOR) operation for the input values.

(NOTE) Refer to “12.2.2.1 Assignment” for details when the data types for the operation result and the “Out” data are different.

Description of parameter

In1 to N	Input value
Out	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
In	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

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.

```
VAR3:=VAR1 XOR VAR2;
```

BUILT IN PLC execution error

None

NOT

Function

Command	Notation	Description
NOT	Out: = NOT In;	Performs the negation (NOT) operation for the input values.

(NOTE) Refer to “12.2.2.1 Assignment” for details when the data types for the operation result and the “Out” data are different.

Description of parameter

In	Input value
Out	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
In	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

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.

```
VAR:=NOT VAR1;
```

BUILT IN PLC execution error

None

12.3.2.7 Selection function

SEL

Function

Command	Notation	Description
SEL	Out: = SEL(G, In0 ,In1);	Selects and outputs one of two options (In0 and In1) based on the reference value G (0 or another value that is not 0). When the reference value G is 0, then In0 is selected. When the reference value G is 1, then In1 is selected.

(NOTE 1) The selections In0 and In1 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" for details when the data types for the operation result and the "Out" data are different.

Description of parameter

G	Reference value
In0	Selection when G is 0
In1	Selection when G is not 0
Out	Selection result

Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
G	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
In0	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
In1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Out	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

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		
G	○	○	○	○	○	○	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
In0	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	○	○	○
In1	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	○	○	○
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

Usage example

When variable Input1 is false or true, then the value for VAR1 or VAR2 is assigned to Output1.
For example, when VAR1 and VAR2 are 100 and 200, respectively, and Input1 is true, then 200 is assigned to Output1.

```
Output1:=SEL(Input1, VAR1, VAR2);
```

BUILT IN PLC execution error

None

MAX

Function

Command	Notation	Description
MAX	Out: = MAX(In1, In2 …,InN); N=2~15	Outputs the maximum value from the input values.

(NOTE 1) The selections from In1 to InN 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 first 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” for details when the data types for the operation result and the “Out” data are different.

Description of parameter

In1 to InN	Selections
Out	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
In	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Out	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

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		
In	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

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.

Example
- '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 from the variables A1, A2 and A3 is assigned to A4.
For example, when variable A1:=16#0000000F(15), variable A2:=16#FFFFFFFF(-1) and variable A3:=16#00000000(0), then 16#0000000F is assigned to variable A4.

A4:=MAX(A1, A2, A3);

BUILT IN PLC execution error
None

MIN
Function

Command	Notation	Description
MIN	Out: = MIN(In1, In2,…,InN); N=2~15	Assess the minimum value.

(NOTE 1) The selections from In1 to InN 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 first 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” for details when the data types for the operation result and the “Out” 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

In1 to InN	Selections
Out	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
In	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Out	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

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		
In	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

Usage example
The minimum value from the variables A1, A2 and A3 is assigned to A4.
For example, when variable A1:=16#0000000F(15), variable A2:=16#FFFFFFFF(-1) and variable A3:=16#00000000(0), then 16#FFFFFFFF is assigned to variable A4.

A4:= MIN(A1,A2,A3);

BUILT IN PLC execution error
None

LIMIT
Function

Command	Notation	Description
LIMIT	Out: = LIMIT(In, MN, MX);	Assesses the upper and lower limits and controls the input value. MN is the lower limit value and MX is the upper limit value. “In” is the value used in the comparison. If “In” does not exceed the upper or lower limits, then the “In” value is assigned to “Out”. If “In” exceeds the upper limit (MX), or falls below the lower limit (MN), then the MX and MN are assigned to “Out”.

(NOTE 1) In, MN and MX 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. If the comparison value is greater than the upper limit, then the upper limit is output.
When the upper limit is a non-number, if the comparison value is greater than the lower limit, then the comparison value is output. If the comparison value is less than the lower limit, then the lower limit 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” for details when the data types for the operation result and the “Out” 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

In	Comparison value
MN	Lower limit (minimum)
MX	Upper limit (maximum)
Out	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
In	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
MN	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
MX	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Out	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

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		
In	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	○	○	○
MN	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○
MX	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	○	○	○
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

Usage example
The lower limit is VAR_MIN, the upper limit is VAR_MAX and the comparison value is VAR1.
For example, when VAR_MIN is 16#00000000 and VAR_MAX is 16#0000000F, the processing below applies.
When VAR1 is smaller than 16#00000000, then 16#00000000(VAR_MIN) is assigned to VAR_OUT.
When VAR1 is greater than 16#00000000 and less than 16#0000000F, then the VAR1 value is assigned to VAR_OUT.
When VAR1 is larger than 16#0000000F, then 16#0000000F(VAR_MAX) is assigned to VAR_OUT.

VAR_OUT:=LIMIT(VAR1,VAR_MIN,VAR_MAX);

BUILT IN PLC execution error
None

<u>MUX</u>
Function

Command	Notation	Description
MUX	Out: = MUX(K,In0,In1,…,InN); N=2~13	Selects the value to be output using the K value specified for the selection. K=0: Outputs the In0 value K=1: Outputs the In1 value … K=N: Outputs the InN value If the K value is not within this range, then 0 is output.

(NOTE 1) The selections from In0 to InN 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” for details when the data types for the operation result and the “Out” data are different.

Description of parameter

K	Specifies selection
In0 to N	Selections
Out	Selected value

Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
K	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
In0 to N	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Out	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

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		
K	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○
In0 to N	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	○	○	○
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

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 is 0, then the Select1 value is assigned to Out1.

```
Out1:=MUX(VAR,Select1,Select2,Select3 );
```

BUILT IN PLC execution error
None

12.3.2.8 Comparison function

GT

Function

Command	Notation	Description
GT	Out: = GT(In1,In2,…,InN-1,InN); or Out: =(In1> In2) & (In2>In3) &… &(InN-1>InN); N=2~15	Compares the input values and when the following applies: In1>In2>…>InN, 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 the values for comparison are not decimal numerals, then they are converted to signed decimal numerals for comparison.
(NOTE 4) Refer to "MAX" for further details on comparing the data types TIME, DATE, TOD, DT and STRING.

Description of parameter

In1 to N	Input value
Out	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
In	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	○	○	○
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

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.

```
Input1:=10;
Input2:=5;
Input3:=0;
Input4:=-1;
Output1:= GT(Input1,Input2,Input3,Input4);
```

BUILT IN PLC execution error

None

GE

Function

Command	Notation	Description
GE	Out: = GE(In1,In2,…,InN-1,InN); or Out: =(In1>= In2) & (In2>=In3)& …&(InN-1>=InN); N=2~15	Compares the input values and when the following applies: In1≥In2≥…≥InN, 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 the values for comparison are not decimal numerals, then they are converted to signed decimal numerals for comparison.

(NOTE 4) Refer to "MAX" for further details on comparing the data types TIME, DATE, TOD, DT and STRING.

Description of parameter

In1 to N	Input value
Out	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
In	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	○	○	○
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

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.

```
Input1:=10;
Input2:=5;
Input3:=5;
Input4:=-1;
Output1:= GE(Input1,Input2,Input3,Input4);
```

BUILT IN PLC execution error

None

LT

Function

Command	Notation	Description
LT	Out: = LT(In1,In2,… InN-1,InN); or Out: = (In1<In2) & (In2< In3) &… &(InN-1<InN); N=2~15	Compares the input values and when the following applies: In1<In2<…<InN, 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 the values for comparison are not decimal numerals, then they are converted to signed decimal numerals for comparison.

(NOTE 4) Refer to "MAX" for further details on comparing the data types TIME, DATE, TOD, DT and STRING.

Description of parameter

In1 to N	Input value
Out	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
In	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	○	○	○
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

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.

```
Input1:=-10;
Input2:=-5;
Input3:=0;
Input4:=1;
Output1:= LT(Input1,Input2,Input3,Input4);
```

BUILT IN PLC execution error

None

LE

Function

Command	Notation	Description
LE	Out: = LE(In1,In2, …,InN-1,InN); or Out: =(In<= In2) & (In2<=In3) &… &(InN-1<=InN); N=2~15	Compares the input values and when the following applies: In1<=In2<=…<=InN, 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 the values for comparison are not decimal numerals, then they are converted to signed decimal numerals for comparison.

(NOTE 4) Refer to "MAX" for further details on comparing the data types TIME, DATE, TOD, DT and STRING.

Description of parameter

In1 to N	Input value
Out	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
In	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	○	○	○
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

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.

```
Input1:=-10;
Input2:=0;
Input3:=0;
Input4:=1;
Output1:= EQ(Input1,Input2,Input3,Input4);
```

BUILT IN PLC execution error

None

EQ

Function

Command	Notation	Description
EQ	Out: = EQ(In1,In2, …,InN-1,InN); or Out: =(In1=In2) & (In2=In3) &… &(InN-1=InN); N=2~15	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 the values for comparison are not decimal numerals, then they are converted to signed decimal numerals for comparison.

(NOTE 4) Refer to "MAX" for further details on comparing the data types TIME, DATE, TOD, DT and STRING.

Description of parameter

In1 to N	Input value
Out	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
In	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	○	○	○
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

Usage example

The input values from Input1 to Input3 are compared, and the result is assigned to Output1. Input1 to Input3 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.

```
Input1:=10;
Input2:=10;
Input3:=10;
Output1:= EQ(Input1,Input2,Input3);
```

BUILT IN PLC execution error

None

NE

Function

Command	Notation	Description
NE	Out := NE(In1,In2); or Out := In1 <> In2;	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 the values for comparison are not decimal numerals, then they are converted to signed decimal numerals for comparison.
(NOTE 4) Refer to "MAX" for further details on comparing the data types TIME, DATE, TOD, DT and STRING.

Description of parameter

In1 and In2	Input value
Out	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
In1, 2	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	○	○	○
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

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 output to Output1.

```
Input1:=1;
Input2:=-1;
Output1:= NE(Input1,Input2);
```

BUILT IN PLC execution error

None

12

12.3.2.9 Character string function

LEN

Function

Command	Notation	Description
LEN	Out: = LEN (In);	Calculates the number of characters in the character string (In)..

(NOTE 1) When the input value is NULL, the number of characters is processed as 0 (zero).
(NOTE 2) Use ASCII code for the character code of the input value.

Description of parameter

In	Input value (character string)
Out	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
In	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
Out	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

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.

```
Output1:=LEN(Input1);
```

BUILT IN PLC execution error
None

LEFT

Function

Command	Notation	Description
LEFT	Out: = LEFT(In, L);	Acquires the specified number of characters (L) from the left in the character string (In).

(NOTE 1) When the character string (In) is NULL, then the NULL characters are output.

(NOTE 2) When 0 (zero) or a negative number is entered to specify the number of acquired characters, then the NULL characters are output.

(NOTE 3) When the L value is greater than the number of characters, then all characters entered for “In” are acquired. In addition, the difference between the L value and the number of characters is ignored.
For example, when the character string “STRING” is assigned to “In” and the L value is 7, since “STRING” is 6 characters long, all the characters “STRING” are assigned to “Out”.

Description of parameter

In	Character string
L	Specifies number of characters to acquire
Out	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
In	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
L	-	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Out	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○

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		
In	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
L	○*1	○*1	○*1	○*1	○*1	○*1	○*1	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Out	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*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.

Output1:= LEFT(Input1,Length);

BUILT IN PLC execution error

None

RIGHT

Function

Command	Notation	Description
RIGHT	Out: = RIGHT(In, L);	Acquires the specified number of characters (L) from the right in the character string (In).

(NOTE 1) When the character string (In) is NULL, then the NULL characters are output.

(NOTE 2) When 0 (zero) or a negative number is entered for L, then the NULL characters are output.

(NOTE 3) When the L value is greater than the number of characters, then all characters entered for “In” are acquired. In addition, the difference between L and the number of characters is ignored.
For example, when the character string “STRING” is assigned to “In” and the L value is 7, since “STRING” is 6 characters long, all the characters “STRING” are assigned to “Out”.

Description of parameter

In	Character string
L	Specifies number of characters to acquire
Out	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
In	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
L	-	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Out	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○

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		
In	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
L	○*1	○*1	○*1	○*1	○*1	○*1	○*1	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Out	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*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.

Output1:= RIGHT(Input1, Length);

BUILT IN PLC execution error

None

MID

Function

Command	Notation	Description
MID	Out: =MID (In, L, P);	Acquires the specified number of characters (L) in the character string (In), starting from the specified position (P) moving to the right.

(NOTE 1) The position of the beginning character in the character string “In” is processed as 1.
(NOTE 2) When the character string “In” is NULL, then NULL is output.
(NOTE 3) When 0 (zero) or a negative number is specified for the number of characters L, then NULL is output.
(NOTE 4) When the number of characters L 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 P is greater than the number of characters for “In”, or when it is a 0 (zero) or negative number, then NULL is output.

Description of parameter

In	Character string
L	Number of characters to be acquired
P	Start position for acquiring characters
Out	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
In	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
L	-	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
P	-	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Out	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○

Usable OM / Constant

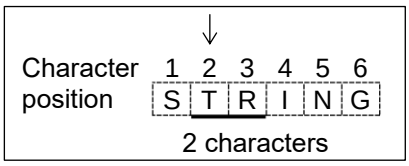
	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
In	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
L	○*1	○*1	○*1	○*1	○*1	○*1	○*1	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
P	○*1	○*1	○*1	○*1	○*1	○*1	○*1	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Out	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*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 Position=2 and Length=2 (number of characters) are specified, then the character string 'TR' is assigned to Output1.

Output1:= MID(Input1, Length, Position);



BUILT IN PLC execution error

None

CONCAT

Function

Command	Notation	Description
CONCAT	Out := CONCAT (In , In2);	Joins or concatenates the character strings In1 and In2.

(NOTE 1) The character string In2 is joined or concatenated at the right of the character string In1.

(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 In1 is “AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA” (30 As) and “In” is “BC”, then the concatenated character string becomes “AAAAAAAAAAAAAAAAAAAAAAAAAAAAAB” (31 characters).

Description of parameter

In1,2	Input character string
Out	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
In1,2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
Out	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○

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		
In1,2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
Out	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

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.

Output1:= CONCAT(Input1, Input2);

BUILT IN PLC execution error

None

INSERT

Function

Command	Notation	Description
INSERT	Out := INSERT(In1, In2, P)	Inserts one character string (In2) at the specified position (P) of another character string (In1).

(NOTE 1) The In2 character string is inserted to the right of the character position specified by P. The position of the beginning character in In1 is processed as 1.

(NOTE 2) When the insert position P is greater than the number of characters for In1, then the In1 character string is output.

(NOTE 3) When the value for the insert position P is a 0 (zero) or a negative number, then the In2 character string is inserted to the left of the character string In1.

(NOTE 4) If the output value is more than 31 characters, any character from the 32nd character is ignored.

Description of parameter

In1	Character string
In2	Inserted character string
P	Insertion position
Out	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
In1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
In2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
P	-	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Out	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○

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		
In1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
In2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
P	○*1	○*1	○*1	○*1	○*1	○*1	○*1	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Out	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

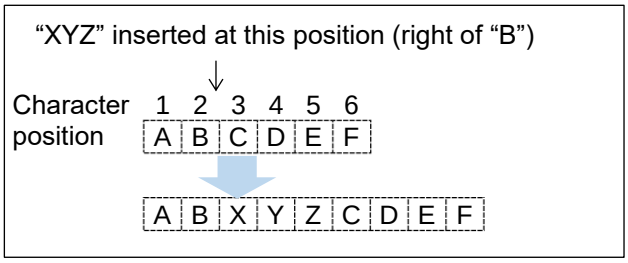
*1 Possible only when digits are specified.

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 the output value.

Output1:= INSERT(Input1,Input2,Position);



BUILT IN PLC execution error

None

DELETE

Function

Command	Notation	Description
DELETE	Out := DELETE(In,L, P);	Deletes only the specified number of characters (L) from the specified position (P) in the character string (In).

(NOTE 1) The position of the beginning character in the character string “In” is processed as 1.
(NOTE 2) When the number of characters L 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 the start position P is greater than the number of characters for input value 1, then the “In” character string is output.
(NOTE 4) When 0 (zero) or a negative number is entered for L and P, then NULL is output.

Description of parameter

In	Character string
L	Number of characters to be deleted
P	Delete start position
Out	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
In	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
L	-	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
P	-	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Out	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○

Usable OM / Constant

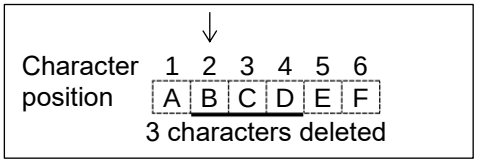
	Bit									Word								Label	Nesting	Constant		WL designation	INDEX
	X	Y	M	BX	BY	LM	SM	LT	LC	D	BDX	BDY	LD	LT	LC	SD	IX	L	N	Value	Character string		
In	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
L	○*1	○*1	○*1	○*1	○*1	○*1	○*1	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
P	○*1	○*1	○*1	○*1	○*1	○*1	○*1	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Out	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*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.

Output1:=DELETE(Input1,Length, Position);



BUILT IN PLC execution error

None

REPLACE

Function

Command	Notation	Description
REPLACE	Out := REPLACE(In1, In2, L, P);	Replaces part of a character string with a specific character string. The specified number of characters (L) from the specified position (P) is replaced in the In1 character string. That section is replaced with the In2 character string.

(NOTE 1) The position of the beginning character in the character string In1 is processed as 1.
(NOTE 2) When the replacement start position P is greater than the number of characters for In1, then the In2 character string is inserted behind the In1 character string.
(NOTE 3) When the number of characters L is greater than the number of characters from the replacement start position P (moving to the right), then the entire character string to the right of the position specified in P is replaced.
(NOTE 4) When 0 (zero) or a negative number is specified for L and P, 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

In1	Character string where replacement is carried out
In2	Character string inserted for replacement
L	Number of characters used in replacement
P	Replacement start position
Out	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
In1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
In2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
L	-	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
P	-	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Out	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○

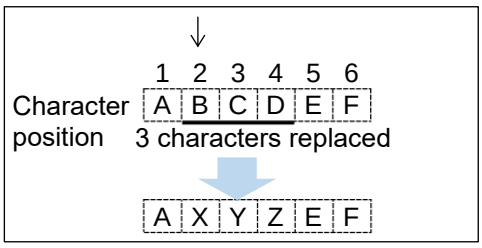
Usable OM / Constant

	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		
In1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
In2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
L	○*1	○*1	○*1	○*1	○*1	○*1	○*1	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
P	○*1	○*1	○*1	○*1	○*1	○*1	○*1	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Out	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*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.

Output1:= REPLACE(Input1,Input2,Length,Position);



BUILT IN PLC execution error
None

FIND

Function

Command	Notation	Description
FIND	Out := FIND(In1, In2);	Assigns the position of the specified character string (In2) from the character string (In1) to “Out”.

(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 the character string In1 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 In1 or In2, 0 (zero) is output.

Description of parameter

In1	Character string where search is carried out
In2	Character string used in search
Out	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
In1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
In2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
Out	○	○	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-

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		
In1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
In2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
Out	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	○	○	○

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.

Output1:=FIND(Input1, Input2);

BUILT IN PLC execution error

None

12.3.2.10 Time data type function

ADD_TIME

Function

Command	Notation	Description
ADD_TIME	Out := In1 + In2;	Adds time (In1) plus time (In2).

(NOTE) If the operation result exceeds the TIME type range, an error is triggered.

Description of parameter

In1 and In2	Time added (TIME type)
Out	Value after addition (TIME type)

Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
In2	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
Out	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-

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		
In1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
In2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
Out	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*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.

Output1:=Input1+ Input2;

BUILT IN PLC execution error

When the operation result exceeds the TIME type range. (Error code 44)

ADD_TOD_TIME

Function

Command	Notation	Description
ADD_TOD_TIME	Out:=In1 +In2;	Adds time (In2) to time of day (In1).

(NOTE) If the operation result exceeds the TOD type range, an error is triggered.

Description of parameter

In1	Time of day that is added (TOD type)
In2	Time that is added (TIME type)
Out	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
In1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
In2	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
Out	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-

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		
In1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
In2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*2	-	-	-
Out	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*1 Possible only with TOD type.
*2 Possible only with TIME type.

Usage example

The TOD type variable Input1 and the TIME type variable Input2 are added, and the result is assigned to Output1.
For example, when T#001:11:11 is added to TOD#12:34:56, then TOD#13:46:07 is assigned to the output value.

Output1:=Input1+ Input2;

BUILT IN PLC execution error

When the operation result exceeds the TOD type range. (Error code 44)

ADD_DT_TIME

Function

Command	Notation	Description
ADD_DT_TIME	Out := In1 + In2;	Adds time (In2) to the date and time (In1).

(NOTE 1) Leap year is also taken into account in the add operation.
For example, the following shows how it applies: TD#2020-02-28-23:59:59+ T#000:00:01=TD#2020-02-29-00:00:00

(NOTE 2) If the operation result exceeds the TOD type range, an error is triggered.

Description of parameter

In1	Date and time that is added to (DT type)
In2	Time that is added (TIME type)
Out	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
In1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-
In2	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
Out	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-

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		
In1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
In2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*2	-	-	-
Out	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*1 Possible only with DT type.
*2 Possible only with TIME type.

Usage example
The DT type variable Input1 and the TIME type variable Input2 are added, and the result is assigned to Output1.
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.

Output1:=Input1+ Input2;

BUILT IN PLC execution error
When the operation result exceeds the DT type range. (Error code 44)

SUB_TIME

Function

Command	Notation	Description
SUB_TIME	Out := In1 - In2;	Subtracts time values. The value (In2) is subtracted from the other value (In1), 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

In1	Subtraction value (minuend) (TIME type)
In2	Subtraction value (subtrahend) (TIME type)
Out	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
In1	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
In2	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
Out	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-

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		
In1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
In2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
Out	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*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.

VAR3:=VAR1-VAR2;

BUILT IN PLC execution error

When the operation result exceeds the TIME type range. (Error code 44)

SUB_DATE_DATE

Function

Command	Notation	Description
SUB_DATE_DATE	Out := In1 - Out2;	Subtracts date values. The value (In2) is subtracted from the other value (In1), 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

In1	Subtraction value (minuend) (DATE type)
In2	Subtraction value (subtrahend) (DATE type)
Out	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
In1	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
In2	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
Out	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-

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		
In1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
In2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
Out	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Usage example

The DATE type VAR2 is subtracted from the DATE type VAR1, and the operation result is assigned to VAR3.

For example, when D#2018-03-02 is subtracted from D#2018-03-03, then T#024:00:00 is assigned to the output value.

VAR3:=VAR1-VAR2;

BUILT IN PLC execution error

When the operation result exceeds the TIME type range. (Error code 44)

SUB_TOD_TIME

Function

Command	Notation	Description
SUB_TOD_TIME	Out := In1 - In2;	Subtracts time from time of day. The value (In2) is subtracted from the other value (In1), 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

In1	Subtraction value (minuend) (TOD type)
In2	Subtraction value (subtrahend) (TIME type)
Out	Subtraction result (TOD type)

Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
In2	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
Out	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-

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		
In1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
In2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*2	-	-	-
Out	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*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.

VAR3:=VAR1-VAR2;

BUILT IN PLC execution error

When the operation result exceeds the TOD type range. (Error code 44)

SUB_TOD_TOD

Function

Command	Notation	Description
SUB_TOD_TOD	Out := In1 - In2;	Subtracts time of day values. The value (In2) is subtracted from the other value (In1), 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

In1	Subtraction value (minuend) (TOD type)
In2	Subtraction value (subtrahend) (TOD type)
Out	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
In1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
In2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
Out	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-

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		
In1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
In2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
Out	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*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 of day TOD#12:34:56, then T#01:01:12 is assigned to the output value.

VAR3:=VAR1-VAR2;

BUILT IN PLC execution error

When the operation result exceeds the TIME type range. (Error code 44)

SUB_DT_TIME

Function

Command	Notation	Description
SUB_DT_TIME	Out := In1 - In2;	Subtracts time from date and time. The value (In2) is subtracted from the other value (In1), 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

In1	Subtraction value (minuend) (DT type)
In2	Subtraction value (subtrahend) (TIME type)
Out	Subtraction result (DT type)

Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-
In2	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
Out	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-

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		
In1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
In2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*2	-	-	-
Out	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*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.

VAR3:=VAR1-VAR2;

BUILT IN PLC execution error

When the operation result exceeds the DT type range. (Error code 44)

SUB_DT_DT

Function

Command	Notation	Description
SUB_DT_DT	Out := In1 - In2;	Subtracts the date and time values. The value (In2) is subtracted from the other value (In1), 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

In1	Subtraction value (minuend) (DT type)
In2	Subtraction value (subtrahend) (DT type)
Out	Value after subtraction (TIME type)

Usable data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-
In2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-
Out	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-

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		
In1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
In2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
Out	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*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#2016-02-13-12:33:55 is subtracted from DT#2016-02-14-12:34:56, then T#024:01:01 is assigned to the output value.

VAR3:=VAR1-VAR2;

BUILT IN PLC execution error

When the operation result exceeds the TIME type range. (Error code 44)

MUL_TIME

Function

Command	Notation	Description
MUL_TIME	Out := In1 * In2;	Multiplies time 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

In1	Time value used in multiplication (TIME type)
In2	Multiplication value
Out	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
In1	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
In2	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Out	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-

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		
In1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
In2	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Out	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*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.

VAR3:=VAR1*VAR2;

BUILT IN PLC execution error

When the operation result exceeds the TIME type range. (Error code 44)

DIV_TIME

Function

Command	Notation	Description
DIV_TIME	Out := In1 / In2;	Divides time values. The value (In1) is divided by the value (In2), 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

In1	Time value for division (TIME type)
In2	Division value (divisor)
Out	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
In1	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-
In2	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Out	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-	-

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		
In1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○*1	-	-	-
In2	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
Out	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*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.

VAR3:=VAR1/VAR2;

BUILT IN PLC execution error

When the divisor is 0 in the division command. (Error code 11)

CONCAT_DATE_TOD

Function

Command	Notation	Description
CONCAT_DATE_TOD	Out := CONCAT_DATE_TOD(In1, In2);	Concatenates or joins the time of day (In2) with the date (In1).

Description of parameter

In1	Date
In2	Time of day
Out	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
In1	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
In2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-
Out	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-

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		
In1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
In2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Out	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

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.

```
Output1:=CONCAT_DATE_TOD(Input1,Input2);
```

BUILT IN PLC execution error

None

12.3.2.11 Expanded function

MCPY

Function

Command	Notation	Description
MCPY	MCPY(In1, In2, In3);	Transfers the number of word points (In3) specified by the user, from the specified variable / OM (In1) to the variable / OM (In2).

(NOTE 1) When In1 (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 In3 (transfer points). If the setting is invalid, an error is triggered at conversion.

(NOTE 4) If the number of points exceeds the range of the OM specified for In1 (source) and In2 (destination) and the number of points is set to In3 (transfer points), then an error is triggered at conversion.

(NOTE 5) When a program such as D10:=MCPY(VAR,D0,1); is created, an error is triggered at conversion, because there is no return value for MCPY.

Description of parameter

In1	Variable / OM for source
In2	Variable / OM for destination
In3	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 data type

	Data type																
	BOOL	BYTE	WORD	DWORD	INT	DINT	LINT	UINT	UDINT	ULINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1	-	-	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
In2	-	-	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
In3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

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		
In1	○*1	○*1	○*1	○*1	○*1	○*1	○*1	-	-	○*2	○*2	○*2	○*2	-	-	○*2	○*2	-	-	-	-	○*1 *2	○
In2	-	-	-	-	-	-	-	-	-	○	○	○	○	-	-	○	○	-	-	-	-	-	○
In3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-

*1 Possible only when 16-bit digits are specified.
*2 Longwords cannot be specified for the word OM.

Usage example
The value for the WORD type variable VAR is 1 word and is transferred to D0.

MCPY(VAR,D0,1);

BUILT IN PLC execution error
- When an attempt is made to access an out of range OM. (Error code 23)

TON
Function

Command	Notation	Description
TON	Out := TON(In ,PT ,N ,ET);	Counts up in the timer OM for the number specified in N when “In” is true. The current timer value is stored in ET. After counting up to the time specified in PT (set value), “Out” becomes true. When “In” is false, the count (time) is reset and “Out” 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 N 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 PT is a 0 (zero), a negative number or greater than 32768, then the count is not performed for the current timer value and “Out” will not become true.

(NOTE 3) When using TON with a CAL command in an LD program, the timer OM for the number specified in the No. 3 argument N counts up and turns ON /OFF. For example, when Out1:=TON(M101,100,0,D100); is executed, the LT0 contact, coil, set value and current value change.
Therefore, be careful not use a timer with the same number for the OUT LT command in the ladder circuit.

Description of parameter

In	Input. Current timer value counts up when it is true. Current timer value resets when it is false.
PT	Timer set value (Only integers are possible. 1 ms units are used for high speed control programs. 10 ms units are used for standard control programs). Set a value between 1 and 32767.
N	Timer number (0 to 511)
ET	Current timer value
Out	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
In	○	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PT	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
N	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ET	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-
Out	○	○	○	○	○	○	○	○	○	○	-	-	-	-	-	-	-

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		
In1	○	○	○	○	○	○	○	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	○
In2	○	○	○	○	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	○	-	○	○
In3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○	-	-	-
In4	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○
In5	-	○	○	-	○	○	○	-	-	○	○	○	○	-	-	○	○	-	-	-	-	○	○

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.

var2:= TON(X000,var1,2,var3);

BUILT IN PLC execution error
None

12.3.3 Create & Use User Defined Function / System Defined Function

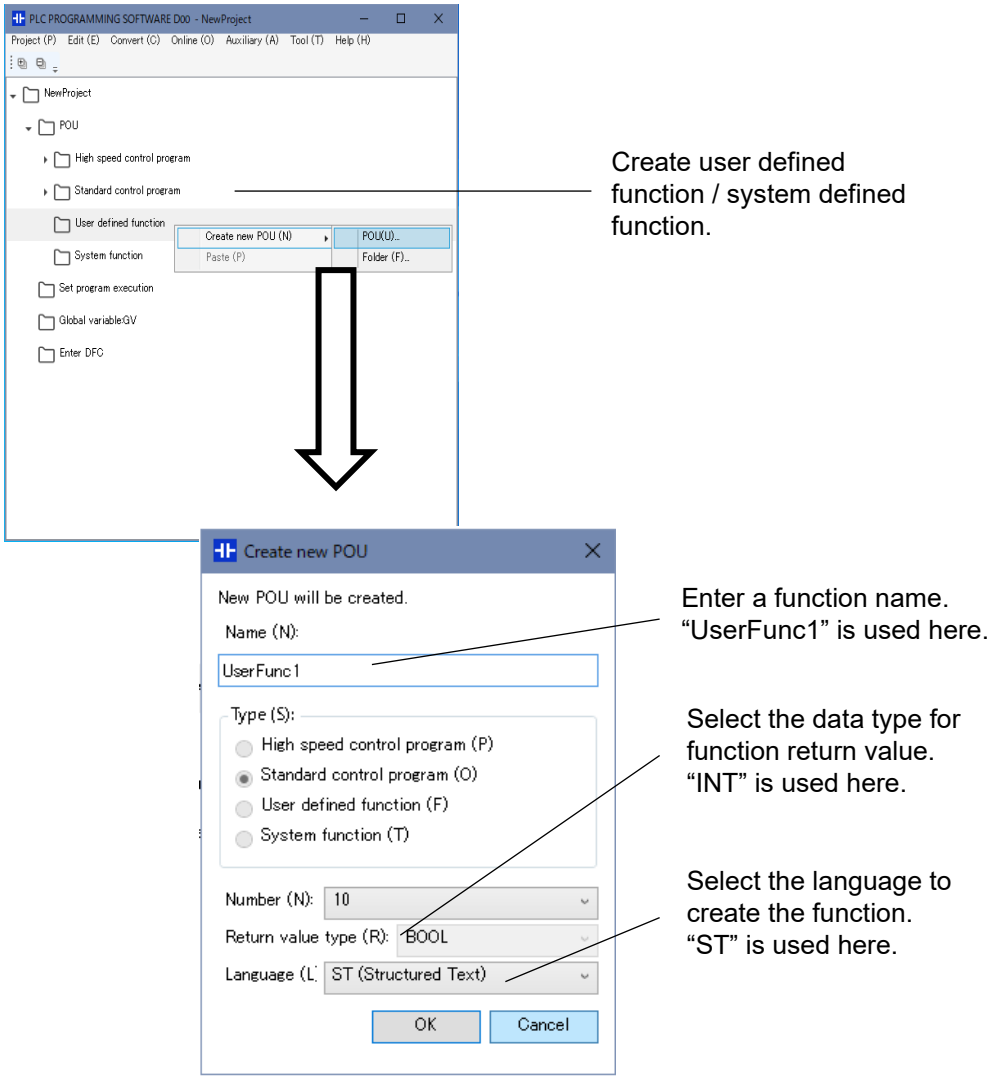
This section provides the following simple example to explain how to create a user defined function and system defined function.

Item		Description
Function name		UserFunc1
Argument	No. 1 argument	Input argument, variable name: In1, data type: INT
	No. 2 argument	Input argument, variable name: In2, data type: INT
	No. 3 argument	Input argument, variable name: In3, data type: INT
	No. 4 argument	Output argument, variable name: Out4, data type: INT
Operation processing		Calculation result of In1 / In2 + In3 is stored in Out4
Return value		-1 when data type is INT and In2 is 0, or 1 in other situations.
Language		ST language

Refer to “Chapter 5 Procedure to create programs” in the PLC System Manual (Option) for details on how to use PLC Programming software D00 EDITOR.

12.3.3.1 Create POU

On the POU control screen, the user can create a new user defined function / system defined function and set the function “name”, “return value type” and “language”.



12.3.3.2 Register variable

The user can open the edit window for the user defined function or system defined function. In this window, the argument for the user defined function or system defined function can be registered as a variable in the variable registration area (**[Parameter]** tab).
The variable registered at the top in the variable registration area is the No. 1 argument. The registration follows the order of the arguments starting from the top.
The type of the variable is specified in the "Scope".

Scope	Description
VAR_OUTPUT	Sets to a variable used as an output variable.
VAR_INPUT	Sets to a variable used as an input variable.
VAR_IN_OUT	Sets to a variable used as an input/output variable.
VAR	Sets to a variable used only in a POU.

The "Variable name", "Data type" and "Default value" are set to the specifications of the user defined function / system defined function to be created.
Note that when the NC(BUILT IN PLC) is executed, a default value is set for the argument each time the user defined function or system defined function is called. However, the argument values specified in the user defined function or system defined function (calling source) are stored for the input variables and input/output variables.
Set the "Comment" when necessary.

In this example, the settings are configured as follows:

Scope	Variable name	Data type	Element number	Default value	Comment
VAR_INPUT	In1	INT	0	0	Argument1
VAR_INPUT	In2	INT	0	0	Argument2
VAR_INPUT	In3	INT	0	0	Argument3
VAR_OUTPUT	Out4	INT	0	0	Argument4

12.3.3.3 Programming and return value

The user defined function and system defined function are created following the same procedure as the programming in the standard ST language. In addition, the user can use another user defined function or system defined function in a user defined or system defined function.
The return value for the user defined or system defined function is specified as follows:

Function name:= Return value

Ex) UserFunc1:=-1;

Specify a value appropriate for the "Return value type" that is set when creating the POU.

The following is a program example.

```
(* Calculate In1÷In2+In3 and output to Out4. When In2 is 0, Out4 is undefined. *)
(* When In2 is 0, -1 is set to return value, and 1 is set when operation is executed. *)

IF In2=0 THEN
        UserFunc1:=-1;
ELSE
        Out4:=In1/In2+In3;
        UserFunc1:=1;
END_IF;
```

When the programming ends, "Design check" is executed and checks whether there is any problem.
If an error is detected, repair or correct the error location.

12.3.3.4 How to use user defined function / system defined function

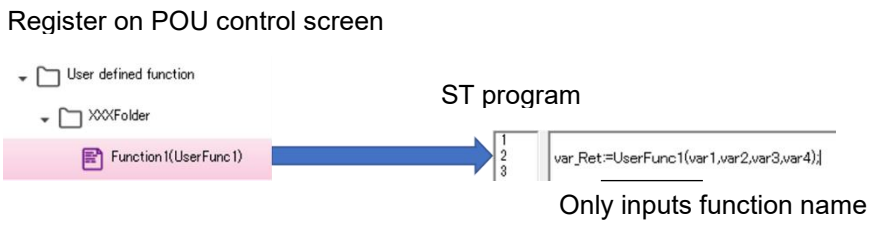
The user defined or system defined function created in the POU is used as follows:

Return value := Function name (No. 1 argument, No. 2 argument,…)

Ex) var_Ret:=UserFunc1(var1,var2,var3,var4);

Make sure to enter the function name correctly because it is case sensitive.

In addition, even if the POU for the user defined function / system defined function is registered to a folder created by the user on the POU control screen, only the function name is input in the ST program. The folder name is not input.



After the user defined function or system defined function is input to the ST program, if only the folder name is changed on the POU control screen, then the ST program does not need to be changed.

12.3.3.5 Restrictions and special notes

This section describes the restrictions and special notes when using a user defined or system defined function.

Recursive Calls with User Defined Function / System Defined Function
Inside a user defined function and system defined function, it cannot call itself.
In addition, the function also cannot indirectly call itself. (For example, when the user defined function Func1 calls the user defined function Func2 and Func2 calls Func1.)

Double coil check
When using user defined or system defined functions at multiple places that are writing for the OM, a double coil is detected. Correct it when necessary.

Nesting with User Defined Function / System Defined Function
It is possible to call a user defined function or system defined function from another user defined function or system defined function. There are no restrictions at the level of the user defined function or system defined function that calls. However, create and use the functions so that the maximum steps are not exceeded in the POU (call source).

Variable registration and step number
In the source POU that calls the user defined function or system defined function, create a program for the user defined function or system defined function that follows the number of steps (below) based on the registered number of variables.

- 3 steps per variable for VAR_INPUT and VAR
- 6 steps per variable for VAR_OUTPUT and VAR_IN_OUT

When calling a user defined or system defined function multiple times in the same POU, the aforementioned step number is needed each time.
Create a POU so that the maximum number of steps is not exceeded in the POU (call source).

Variable initialization (reset)
When variables are created in user or system defined functions, they are reset each time the function is called.

12.4 Error Message List During ST Program Conversion

The following shows a list of error messages during ST program conversion.

ID	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 registered to the POU 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 registered to the POU does not exceed the maximum. (There is no need to decrease the number of OMs used in the program.)
	Data size used for BOOL type parameter variables and return values in functions exceeds the limit: 4096. Data size 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/system defined functions exceed the allowable number that can be registered (BOOL type: 1024, Non-BOOL type: 1024).	Correct the program so that the number of parameter variables registered to the user/system defined function 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> or <System defined function name>.	There is an error in the user defined or system defined function called by the POU.	Check the error description for the user defined or system defined function with the error, and correct the error location.
E0004	“%s is defined by 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.
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 undefined character string is being used for a variable name, function name, etc.	Use a valid name and re-enter it for the variable name or function name. Register any unregistered variable.
		An invalid numerical value is specified for a binary, octal or hexadecimal numeral. (Ex: 2#2, 8#8, 16#G, etc.)	Specify a valid numerical value.
E7003	There is no semicolon (;) before %s.	There is no semicolon (;) at the end of the statement.	Enter a semicolon (;) at the end of the statement.
	There is no semicolon (;) after %s.		
E7005	Number of the parentheses does not match.	The number of opening or closing parentheses does not match to form pairs.	Add or remove a parenthesis so the parentheses form pairs.
E7006	Control statement %s is not valid.	There is a part this is missing for the control statement. Ex: THEN is missing in the IF statement.	Add the missing part to the statement.
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.

ID	Error message	Cause	Countermeasure
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.	
E7009	IF statement is not valid.	There is an END_IF statement but IF is missing.	Use IF and END_IF so they are a pair.
E7010	CASE statement is not valid.	There is an END_CASE statement but CASE is missing. A variable or numerical value with a data type that cannot be used as a “Selection value” was specified in a CASE control statement.	Use CASE and END_CASE so they are a pair. For the “Selection value”, specify a variable or constant that has a data type that can handle integers.
E7011	FOR statement is not valid.	There is an END_FOR statement but FOR is missing.	Use FOR and END_FOR so they are a pair.
E7012	WHILE statement is not valid.	There is an END_WHILE statement but WHILE is missing.	Use WHILE and END_WHILE so they are a pair.
E7013	REPEAT statement is not valid.	There is an END_REPEAT statement but REPEAT is missing.	Use REPEAT and END_REPEAT so they are a pair.
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.
E7018	Character string is too long. Set using no more than 31 characters.	A character string constant with 32 characters or more has been entered.	Enter a character string with 31 characters or less for the corresponding location.
E7021	Subscript for the array variable %s exceeds the element number.	An out of range numerical value is set as the subscript for specifying the element of the array variable.	Set a numerical value within the designated range as the subscript for specifying the element of the array variable.
E7032	“RETURN” cannot be used in an iteration statement.	RETURN is being used in a FOR, WHILE or REPEAT statement.	RETURN cannot be used in a FOR, WHILE or REPEAT statement. Delete RETURN.
E7033	Number of arguments for %s is invalid.	The number of arguments specified for the function is invalid.	Specify a valid number of arguments that can be set for the function.
E7034	EXIT can only be used in an iteration statement.	EXIT is being used outside of the repeating control statement (such as FOR).	Use EXIT so it is inside of the repeating control statement.

ID	Error message	Cause	Countermeasure
E7037	Number of characters per line exceeds the upper limit: %d characters.	The maximum number of characters (1024 characters) per line has been exceeded.	Reduce the number of characters on the line with the error to less than the maximum.
E7038	Statement: '%s cannot be parsed.	The statement does not follow the grammar rules for the ST language.	Rewrite the statement at the corresponding location so it follows the grammar rules.
E7039	Nesting is only possible up to 5 levels.	The nesting for IF, CASE, FOR, WHILE or REPEAT is greater than 5 levels.	Correct the nesting so it is equal or less than 5 levels.
E7040	Invalid characters are used in the character string.	There are characters included that cannot be used in the character string constant.	Rewrite so only valid characters (ASCII) are used.
E7041	Invalid escape sequence exists in the character string.	There are characters in the character string constant that cannot be recognized for the escape sequence.	Check the “12.1.7 Constants” and enter a constant using a valid format.
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.
E7043	Only integer type values can be used for the start value, end value or increment/decrement values in the FOR statement.	Another data type (not BOOL, BYTE, WORD, DWORD, INT, DINT, LINT, UINT, UDINT or ULINT) is being used for the start value, end value or add/subtract value in a FOR statement.	For the start value, end value or add/subtract value in a FOR statement, use one of the following data types: BOOL, BYTE, WORD, DWORD, INT, DINT, LINT, UINT, UDINT or ULINT.
E7044	%s exceeds the range of values that can be input.	There is an out of range value for a constant with a TIME, DATE, TOD or DT data type in the ST program. (*1)	Specify a value within the designated range.
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.
E7046	Number of labels used for control statements has exceeded the limit of 1024. Reduce the number of control statements.	Total number of IFs, CASEs and RETURNs in the CALC command exceed 1024 in the POU overall.	Correct the number of IFs, CASEs and RETURNs used in the program so that the total does not exceed 1024 in the POU overall.
E7047	Function, operator and parenthetical expressions cannot be used for the No. {1} argument for “{0}”.	There are invalid characters set in the function argument.	Check the function specification and re-set the argument correctly.
E7999	Function without a return value cannot be converted into another type.	A function without a return value is assigned to an OM/variable.	Correct the function without a return value so that it is not assigned to an OM/variable.
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.

*1 If the number of digits that can be specified is exceeded, then the conversion error “E7038” is triggered.
For example, when T#000:00:60 is specified, the conversion error “E7044” is triggered. When T#000:00:100 is specified, the conversion error “E7038” is triggered.

(This page was intentionally left blank.)

©2021-2022 BROTHER INDUSTRIES, LTD.
All Rights Reserved.
Printed in Japan.
This is the original instructions.

69A804001
2204 (4)