HITACHI PROGRAMMABLE CONTROLLER

MICRO-EHV

PROGRAMMING MANUAL

DRAFT

NJI-590F(X)

○ Warranty period and coverage

The warranty period is the shorter period either 18 months from the data of manufacture or 12 months from the date of installation.

However within the warranty period, the warranty will be void if the fault is due to;

(1) Incorrect use as directed in this manual and the application manual.

(2) Malfunction or failure of external other devices than this unit.

(3) Attempted repair by unauthorized personnel.

(4) Natural disasters.

The warranty is for the PLC only, any damage caused to third party equipment by malfunction of the PLC is not covered by the warranty.

○ Repair

Any examination or repair after the warranty period is not covered. And within the warranty period any repair and examination which results in information showing the fault was caused by any of the items mentioned above, the repair and examination cost are not covered. If you have any questions regarding the warranty please contact wither your supplier or the local Hitachi Distributor. (Depending on failure part, examination might be impossible.)

○ Ordering parts or asking questions

When contacting us for repair, ordering parts or inquiring about other items, please have the following details ready before contacting the place of purchase.

(1) Model

(2) Manufacturing number (MFG.NO.)

(3) Details of the malfunction

○ Reader of this manual

This manual is described for the following person.

・Person considering the introduction of PLC
・PLC system engineer
・Person handling PLC
・Manager after installing PLC

Warning

(1) This manual may not be reproduced in its entirety or any portion thereof without prior consent.
(2) The content of this document may be changed without notice.
(3) This document has been created with utmost care. However, if errors or questionable areas are found, please contact us.

Microsoft® and Windows® are registered trademarks of America and other registered countries of Microsoft Corp. of the United States.

Safety Precautions

Read this manual and related documents thoroughly before installing, operating, performing preventive maintenance or performing inspection, and be sure to use the unit correctly. Use this product after acquiring adequate knowledge of the unit, all safety information, and all cautionary information. Also, make sure this manual enters the possession of the chief person in charge of safety maintenance.
Safety caution items are classifies as “Danger” and “Caution” in this document.

⚠ DANGER : Cases where if handled incorrectly a dangerous circumstance may be created, resulting in possible death or severe injury.

⚠ CAUTION : Cases where if handled incorrectly a dangerous circumstance may be created, resulting in possible minor to medium injury to the body, or only mechanical damage

However, depending on the circumstances, items marked with ⚠ CAUTION may result in major accidents.

In any case, they both contain important information, so please follow them closely.

Icons for prohibited items and required items are shown blow:

🚫 : Indicates prohibited items (items that may not be performed). For example, when open flames are prohibited, 🚫 is shown.

❗ : Indicates required items (items that must be performed). For example, when grounding must be performed, ⏚ is shown.

1. About installation

⚠ CAUTION

- Use this product in an environment as described in the catalog and this document.
  If this product is used in an environment subject to high temperature, high humidity, excessive dust, corrosive gases, vibration or shock, it may result in electric shock, fire or malfunction.
- Perform installation according to this manual.
  If installation is not performed adequately, it may result in dropping, malfunction or an operational error in the unit.
- Do not allow foreign objects such as wire chips to enter the unit.
  They may become the cause of fire, malfunction or failure.

2. About wiring

REQUIRED

- Always perform grounding (FE terminal).
  If grounding is not performed, there is a risk of electric shocks and malfunctions.

⚠ CAUTION

- Connect power supply that meets rating.
  If a power supply that does not meet rating is connected, fire may be caused.
- The wiring operation should be performed by a qualified personnel.
  If wiring is performed incorrectly, it may result in fire, damage, or electric shock.

3. Precautions when using the unit

DANGER

- Do not touch the terminals while the power is on.
  There is a risk of electric shock.
- Structure the emergency stop circuit, interlock circuit, etc. outside the programmable controller (hereinafter referred to as PLC).
  Damage to the equipment or accidents may occur due to failure of the PLC.
  However, do not interlock the unit to external load via relay drive power supply of the relay output module.

⚠ CAUTION

- When performing program change, forced output, RUN, STOP, etc., while the unit is running, be sure to verify safety.
  Damage to the equipment or accidents may occur due to operation error.
- Supply power according to the power–up order.
  Damage to the equipment or accidents may occur due to malfunctions.

4. About preventive maintenance

DANGER

- Do not connect the (+) and (-) of the battery in reverse polarity. Do not recharge, disassemble, heat, place in fire, or short circuit the battery. There is a risk of explosion or fire.

PROHIBITED

- Do not attempt to disassemble, repair or modify any part of the PLC.
  Electric shock, malfunction or failure may result.

CAUTION

- Turn off the power supply before removing or attaching module/unit.
  Electric shock, malfunction or failure may result.

Revision History

No.	Description of Revision	Date of Revision	Manual number
1	The first edition	2020.00	NJI-590F(X)

MEMO

PROGRAMMING MANUAL Table of Contents

Chapter 1 Introduction 1-1 to 1-8

Chapter 2 Basic operations of MICRO-EHV 2-1 to 2-10

Chapter 3 User program 3-1 to 3-22

Chapter 4 Procedure to Create User Program 4-1 to 4-10

Chapter 5 Command Specifications 5-1 to 5-496

Chapter 1 Introduction

Thank you for choosing Hitachi Programming Logic Controller (hereinafter referred to as PLC) MICRO-EHV Series. This manual describes the information mostly about creating a user program for the MICRO-EHV series. Please read this manual thoroughly before using the PLC for creating a program.

Also, refer to the related materials listed in Table 1.1.

Table 1.1 List of description materials

Items	Title of material	Manual number*
MICRO-EHV hardware	MICRO-EHV HARDWARE MANUAL	NJI-589*(X)
MICRO-EHV function	MICRO-EHV USER'S MANUAL	NJI-591*(X)
Programming software	Control Editor INSTRUCTION MANUAL	NJI-537*(X)

* The last digit of the manual number may change according to an updated version.

1.1 Confirmation of purchase

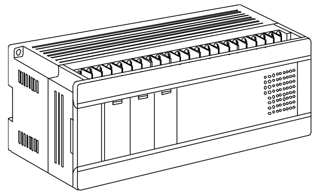
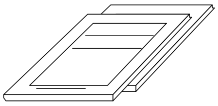
This product has been manufactured carefully; however, please check the following immediately after your purchasing. If there is anything wrong with your purchased product package, please contact your dealer.

(1) Whether the model is as your order.

(2) Whether the product has no damage.

(3) Whether there are all of bundled items in Table 1.2.

Table 1.2 List of content of MICRO-EHV package

No.	Item name	Model	Appearance	No. of items	Remarks
1	MICRO-EHV	MVH-*64** MVL-*64** MVH-*40** MVL-*40** MVL-*20**		1	
2	Instruction manual*	NJI-595* NJI-595*(X)		1	

* The last digit of the manual number may change according to an updated version.

1.2 Doing after unpacking

(1) Installing Battery

A battery is optional for the MICRO-EHV.

If you want to use a clock function and hold internal data when the power is off, please purchase a battery and connect it to the MICRO-EHV main body.

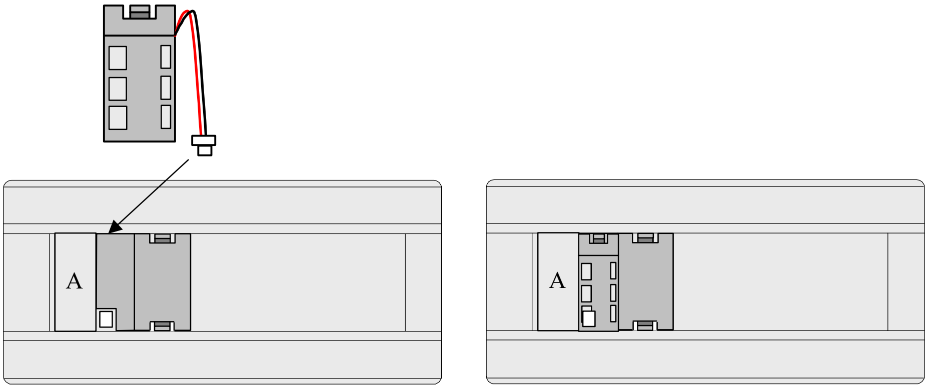


Figure 1.1 Battery connection

DANGER: Precautions for handling the battery

Be sure to use the dedicated battery. Please note that incorrect connection may cause the battery to explode.

Do not charge, disassemble, heat, or short-circuit the battery, place it in fire, or insert it with the wrong polarity (+, -).

(2) Initializing User Program

A blinking pattern may be displayed on the OK LED, which indicates a memory error, because the memory is unstable within the MICRO-EHV right after unpacking. After connecting a battery, initialize the memory of the MICRO-EHV (initialize the CPU) before using it.

Reference

☞ To initialize the memory of the MICRO-EHV, select [Online] - [Operate CPU] - [CPU Initialize] on the programming tool Control Editor menu.

Reference

The CPU initialization initializes a user program, data memory (internal output), and a part of parameters. Communication parameters are not initialized.

(3) Communication parameter settings

The communication parameters are set to the factory default, so if they need to be changed, connect the programming tool (*), set necessary parameters, and turn off and then on the PLC.

(The set parameters are memorized in a backup memory. Once they are set, no more setting is required.)

* When connecting a programming tool to the serial communication port or Ethernet port, set the communication of the programming tool to the default values described in the next page. For the USB port, setting parameters is not required.

Table 1.3 Communication parameters (factory default)

No.	Parameter			Factory default
1	IP address	IP address		192. 168. 0. 1
		Subnet mask		255. 255. 255. 0
		Default gateway		0. 0. 0. 0
		Link Speed / Duplex		Auto Negotiation
2	NTP	Enable/Disable		Disable
		Timezone		GMT + 09:00
3	Serial communication setting	Programming/General purpose		Programming
		Port type		RS-232C
		Baudrate		38.4 kbps
		Protocol		Procedure 1 (1:1)
4	Ethernet communication setting (Task code)	Port 1	Enable/Disable	Enable
			Port No.	3004
			Protocol	TCP/IP
		Port 2	Enable/Disable	Enable
			Port No.	3005
			Protocol	TCP/IP
		Port 3	Enable/Disable	Enable
			Port No.	3006
			Protocol	TCP/IP
		Port 4	Enable/Disable	Enable
			Port No.	3007
			Protocol	TCP/IP
		Timeout (sec.)		30
5	Ethernet communication setting (ASR)	Port 1 Enable/Disable		Disable
		Port 2 Enable/Disable		Disable
		Port 3 Enable/Disable		Disable
		Port 4 Enable/Disable		Disable
		Port 5 Enable/Disable		Disable
		Port 6 Enable/Disable		Disable
6	Modbus-TCP/RTU setting	Port No.		502
		Gateway Enable/Disable		Disable
		Ethernet timeout (×10 ms)		3000
		Serial communication baudrate		38.4 kbps
		Serial communication format		8-E-1
		Serial communication timeout (×10 ms)		100

☞ To change each setting, select it from [CPU Settings] on the programming tool menu.

(4) Clock data settings

When the power is turned on after unpacking (or after the unit is left for a long time with a battery unconnected), the time of the clock data is updated from the initial value. To use the clock function, set the clock data with the programming tool after a battery is attached.

☞ To set a specific time for the PLC or the time of the connected PC, select [Tool] - [CPU Settings] – [Calendar Clock] on the programming tool menu.

Reference

The initial value of the clock is 00:00:00, Saturday, January 1, 2000.

1.3 About manuals

Dedicated instruction manuals for MICRO-EH series expansion units have been issued.

For further details, refer to the expansion units instruction manuals described in Table 1.4.

Table 1.4 MICRO-EH expansion unit-related instruction manual (1/2)

Item name	Model	Type	Instruction manual No.*1	
			Japanese	English
64-point expansion unit	EH-A64EDR	AC power supply, DC input 40 points, Transistor output 24 points	NJI-522□	NJI-522□ (X)
	EH-D64EDR	DC power supply, DC input 40 points, Transistor output 24 points		
	EH-D64EDT	DC power supply, DC input 40 points, Transistor output 24 points (sink)		
	EH-D64EDTPS	DC power supply, DC input 40 points, Transistor with short circuit protection output 20 points (source), Transistor output 4 points (source)		
28-point expansion unit	EH-A28EDR	AC power supply, DC input 16 points, Relay output 12 points	NJI-419□	NJI-419□ (X)
	EH-D28EDR	DC power supply, DC input 16 points, Relay output 12 points		
	EH-D28EDT	DC power supply, DC input 16 points, Transistor output 12 points (sink)		
	EH-D28EDTP	DC power supply, DC input 16 points, Transistor output 12 points (source)		
	EH-D28EDTPS	DC power supply, DC input 16 points, Transistor with short circuit protection output 12 points (source)		
14-point expansion unit *1	EH-A14EDR	AC power supply, DC input 8 points, Relay output 6 points	MICRO-EH Application Manual NJI-349□	MICRO-EH Application Manual NJI-350□ (X)
	EH-D14EDR	DC power supply, DC input 8 points, Relay output 6 points		
	EH-D14EDT	DC power supply, DC input 8 points, Transistor output 6 points (sink)		
	EH-D14EDTP	DC power supply, DC input 8 points, Transistor output 6 points (source)		
	EH-D14EDTPS	DC power supply, DC input 8 points, Short circuit protection output 6 points (source)		
16-point expansion unit	EH-D16ED	DC power supply, DC input 16 points	NJI-467□	NJI-467□ (X)
	EH-D16ER	DC power supply, Relay output 16 points		
	EH-D16ET	DC power supply, Transistor output 16 points (sink)		
	EH-D16ETPS	DC power supply, Transistor with short circuit protection output 16 points (source)		
8-point expansion unit	EH-D8ED	DC power supply, DC input 8 points		
	EH-D8ER	DC power supply, Relay output 8 points		
	EH-D8ET	DC power supply, Transistor output 8 points (sink)		
	EH-D8ETPS	DC power supply, DC input 4 points, Transistor with short circuit protection output 4 points (source)		
	EH-D8EDR	DC power supply, DC input 4 points, Relay output 4 points		
	EH-D8EDT	DC power supply, DC input 4 points, Transistor output 4 points (sink)		
	EH-D8EDTPS	DC power supply, DC input 4 points, Transistor with short circuit protection output 4 points (source)		

□ The end alphabet (one character) of the manual No. indicates the version. The first version is indicated with a space.

*1 For a 14-point expansion unit, refer to the MICRO-EH Application Manual.

Table 1.4 MICRO-EH expansion unit-related instruction manual (2/2)

Item name	Model	Type	Instruction manual No.*1	
			Japanese	English
Analog expansion unit	EH-A6EAN	AC power supply, 4ch input, 2ch output	NJI-424☐	NJI-424☐ (X)
	EH-D6EAN	DC power supply, 4ch input, 2ch output		
RTD expansion unit	EH-A6ERTD	AC power supply, 4ch input, 2ch output	NJI-453☐	NJI-453☐ (X)
	EH-D6ERTD	DC power supply, 4ch input, 2ch output		
	EH-A4ERTD	AC power supply, 4ch input		
	EH-D4ERTD	DC power supply, 4ch input		
Thermocouple expansion unit	EH-D6ETC	DC power supply, 4ch input, 2ch output	NJI-515☐	NJI-515☐ (X)
	EH-D4ETC	DC power supply, 4ch input		

☐ The end alphabet (one character) of the manual No. indicates the version. The first version is indicated with a space.

1.4 Control Editor compatibility

Control Editor which is the programming software for MICRO-EHV is updating according to the modification of MICRO-EHV. If you use the old version of Control Editor, you may not be able to make the program for MICRO-EHV or you cannot use new additional function.

We recommend always using the latest version of Control Editor.

Table 1.5 Function correspondence table

MICRO-EHV Firmware version	Control Editor Recommended version	Main additional function
~ Ver.x102	Ver.4.01 or newer	
Ver.x104	Ver.4.02 or newer	Analog expansion unit, Special I/O (Pulse, PWM output), Modbus communication
Ver.x105	Ver.4.13 or newer	Analog option board
Ver.x106		Added commands. (OMST1, OCTP1, etc.)
Ver.x107		UDP/IP can be used in Ethernet communication (ASR).
Ver.x108		UDP/IP can be used in Ethernet communication (Hitachi dedicated protocol). Backup values of specific data memory without battery.
Ver.x109	Ver.4.20 or newer	Data logging, Analog output option board. Modified Ethernet communication.
Ver.x110		
Ver.x120	Ver.5.00 or newer	Simple positioning function
Ver.x121		
Ver.x122		
Ver.x123		
Ver.x124		
Ver.x125		
Ver.x126	Ver.7.10 or newer	HSDL communication, Insulated analog option board, Insulated RTD option board.

Caution

- Control Editor Ver.3.xx or earlier does not support MICRO-EHV series. If you want to make the program for MICRO-EHV, please prepare the Control Editor Ver.4.01 or newer.

MEMO

Chapter 2 Basic operations of MICRO-EHV

The MICRO-EHV runs on two programs: System program, which controls the MICRO-EHV, and user program, which is created by the user.

The system program is always running while the MICRO-EHV is powered on and monitors MICRO-EHV errors and user program execution/stop. The user program is created by the user in the programming tool and executed based on its operation conditions.

2.1 Structure of MICRO-EHV

The MICRO-EHV consists of the main processor, user memory, backup memory, data memory, and system memory. The internal structure of the MICRO-EHV is shown in Figure 2.1.

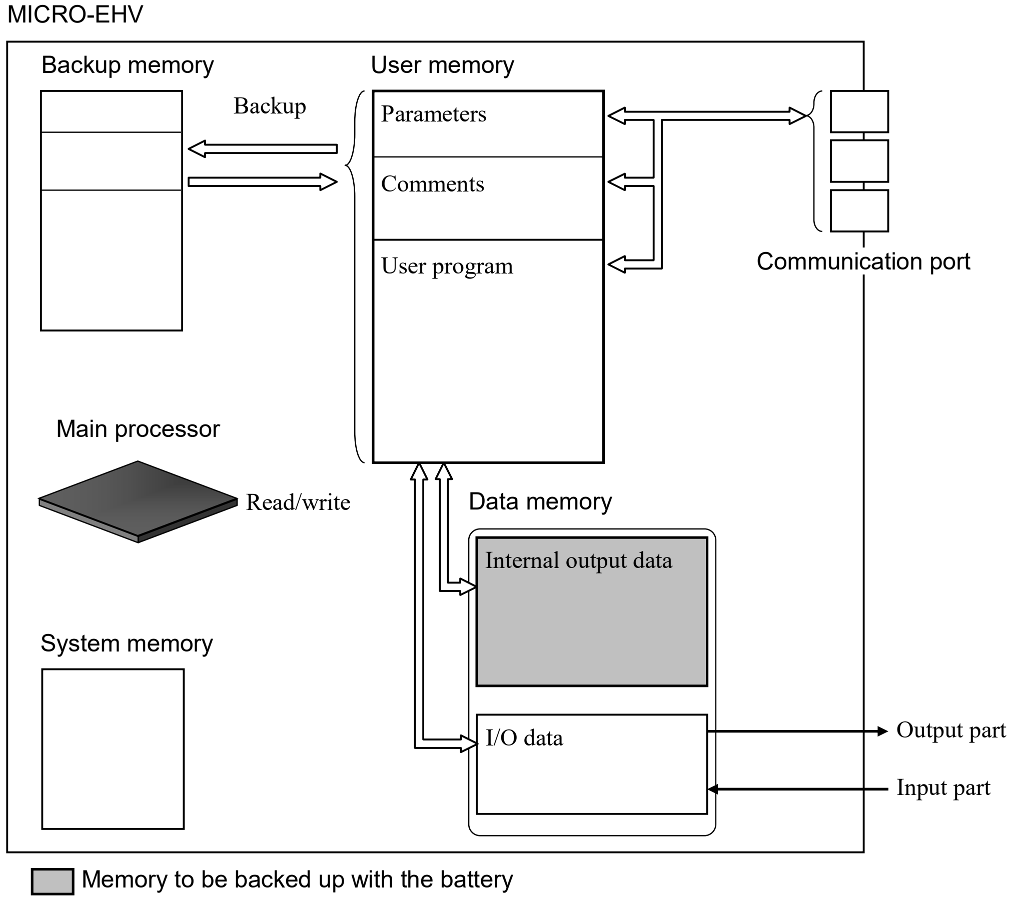


Figure 2.1 MICRO-EHV internal structure

(1) Main processor

This processor is used to execute the system program and user program.

(2) User memory

This memory stores the user program, parameters, and comments.

A user program transferred from the programming tool is written into the user memory.

■ User program

The user program instructs the PLC to perform operations specified by the user. It contains parameters related to I/O installation information and PLC operations, as well as descriptions about combinations of instructions.

See "Chapter 4 Procedure to Create User Program".

■ Parameters

There are various types of parameters, such as communication settings, error display, and program execution.

You can set only specific parameters or add parameters to the user program when transferring it. (The setting method is determined according to the parameter type.)

For details on the operation parameters, see “Chapter 4 Procedure to Create User Program“. For other parameters, see the Hardware Manual and User's Manual.

■ Comments

A comment is a note to be added to the user program to make it easy to read. There are the following comment types: I/O comment, circuit comment, and box comment.

The MICRO-EHV can internally store comments together with the program.

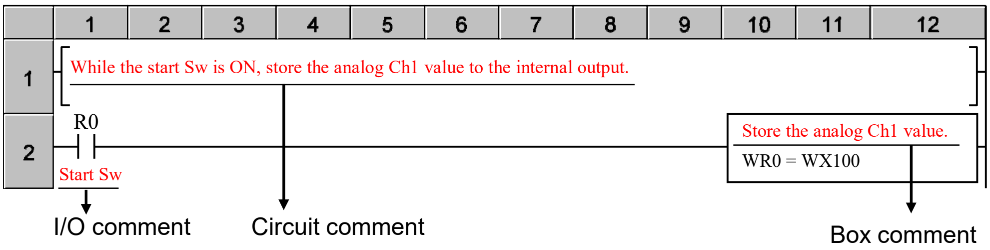


Figure 2.2 Comment examples

(3) Backup memory

This memory is used to retain user memory data. Data is automatically moved to the backup memory during program write (including change in RUN). If the user memory is undefined status at power-on, the data is restored from the backup memory.

Reference

Data memory values are not stored in the backup memory. To retain data memory values at power-off, install the battery and set the retentive area.

(4) Data memory

This memory stores user program operation results and I/O data.

Since internal output data, in particular, can be backed up using the battery, the data is retained even at power-off if it is set as retentive area.

■ Internal output area

A register used as calculating or storing in the user program is called as "internal output". An internal output has an area made up of only bit data and an area made up of word data.

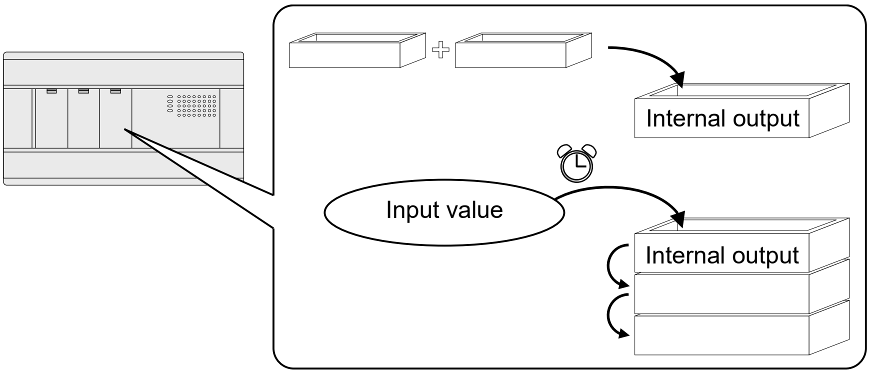


Figure 2.3 Internal output image

■ I/O data area

The I/O data area stores input information obtained from input circuits or output information to be specified in the user program. The MICRO-EHV automatically updates obtained input information and output information to external outputs. This processing is called I/O refresh.

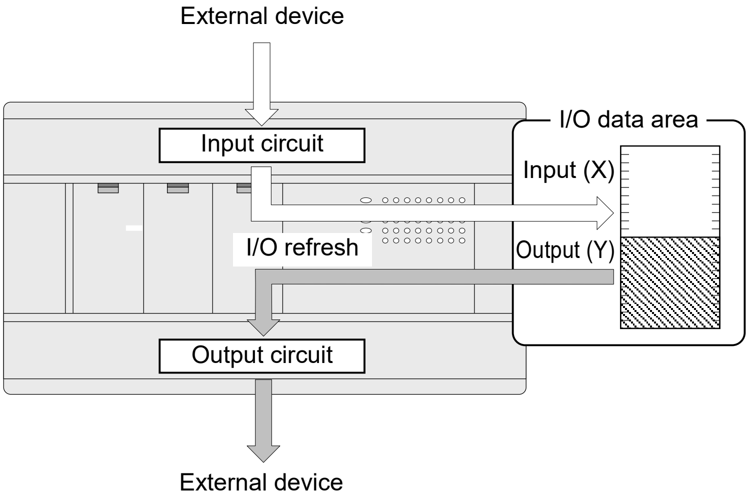


Figure 2.4 Overview of I/O data area

(5) System memory

This memory stores the system program, which controls the MICRO-EHV. The system program is non-rewritable.

2.2 RUN and STOP

The state where the MICRO-EHV is executing the user program is called "RUN" while the state where the MICRO-EHV is not executing the user program is called "STOP".

(1) STOP ➔ RUN

When the MICRO-EHV has no error, if you start the MICRO-EHV containing a correct user program, the program is executed.

Reference

The internal information not specified as retentive area is cleared when the PLC starts running.

(2) RUN ➔ STOP

If you stop the running MICRO-EHV, the user program execution is stopped.

It is also stopped when an error is found during the RUN state.

If a serious error is found, not only the user program but also the system program is stopped.

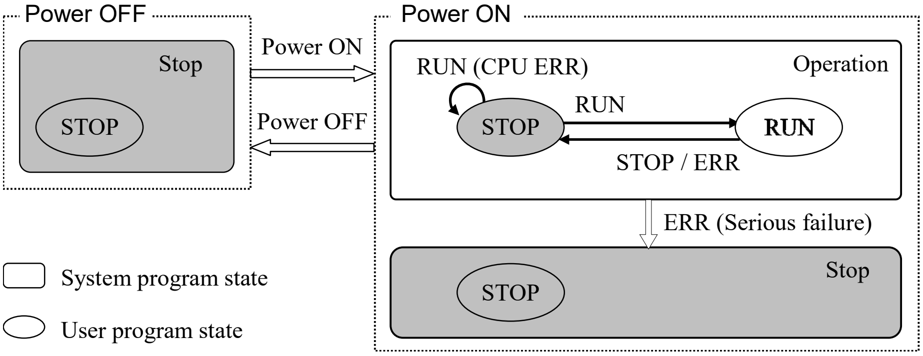


Figure 2.5 State transition diagram of RUN and STOP

Reference

In the STOP state, external outputs are shut off with internal information remaining.

(3) Errors to be detected by MICRO-EHV

The MICRO-EHV detects the following level errors: Serious failure, medium failure, minor failure, and warning. The table below shows the operation state when each error category occurs.

Table 2.1 Errors to be detected by MICRO-EHV

Category	Description	Operation
Serious failure	Power failure, microcomputer error, system ROM error, system RAM error, system bus error, etc. It indicates an unrecoverable serious error.	Stop
Medium failure	Data memory failure, system program error, user memory error, etc. It indicates an error that causes a malfunction if the operation continues.	Stop
Minor failure	I/O information verification error, scan time error, exceeded number of assigned I/Os, etc. It indicates an error that allows the operation to continue by setting the operation parameters.	Stop (Set to continue)
Warning	Transmission error, etc. This is such a minor error that the operation can continue.	Continue

☞ For details on the error codes, see "Chapter 7 Troubleshooting".

(4) RUN and STOP operations

The figure below shows the RUN and STOP operations and the MICRO-EHV states.

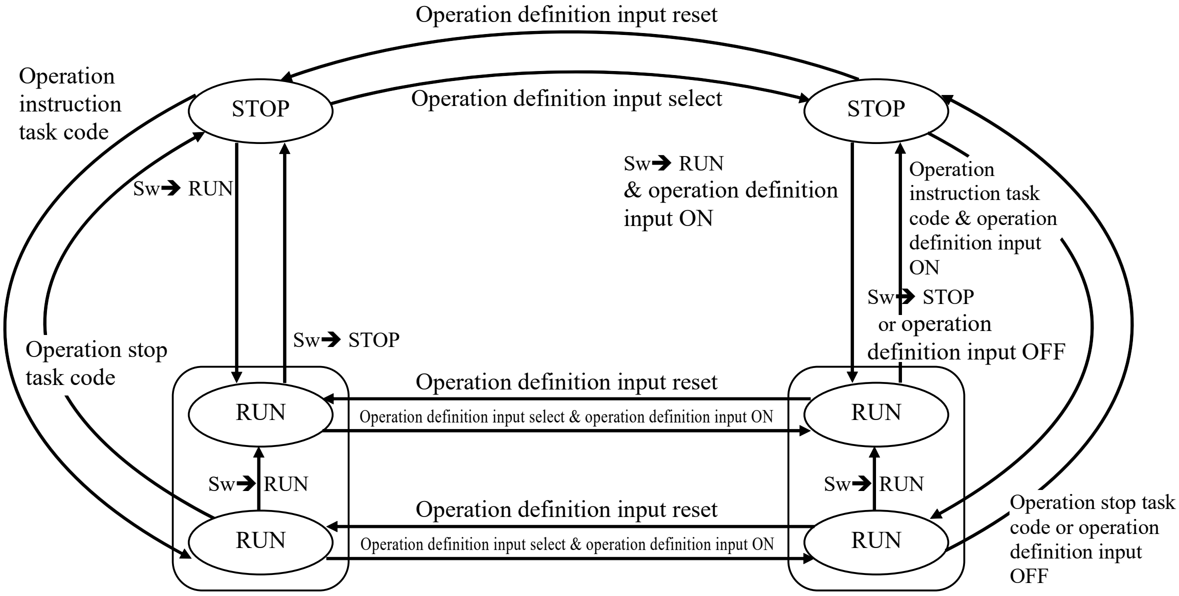


Figure 2.6 State transition diagram of RUN and STOP

To run or stop the PLC, use the hardware switch (RUN/STOP switch), operation definition input, or task code communication. However, if there are multiple conditions, all must be met to start the PLC (Example: The RUN/STOP switch is set to RUN and the operation definition input is set to ON). Even if one of them is not met, the PLC stops.

Term description Task code communication

The MICRO-EHV uses the dedicated communication protocol for communications. This dedicated protocol is called "Hi-Protocol". Since a command in the communication format defined as Hi-Protocol is called a task code, communication using the dedicated protocol is also called "task code communication".

The Control Editor has the RUN/STOP instruction functions, so even if you have no knowledge about the task code communication format, you can perform the RUN and STOP operations in the Control Editor.

2.2.1 Operation during RUN

The figure below shows the operation overview when the MICRO-EHV is in the RUN state. When the MICRO-EHV starts running, it executes the user program sequentially from the beginning.

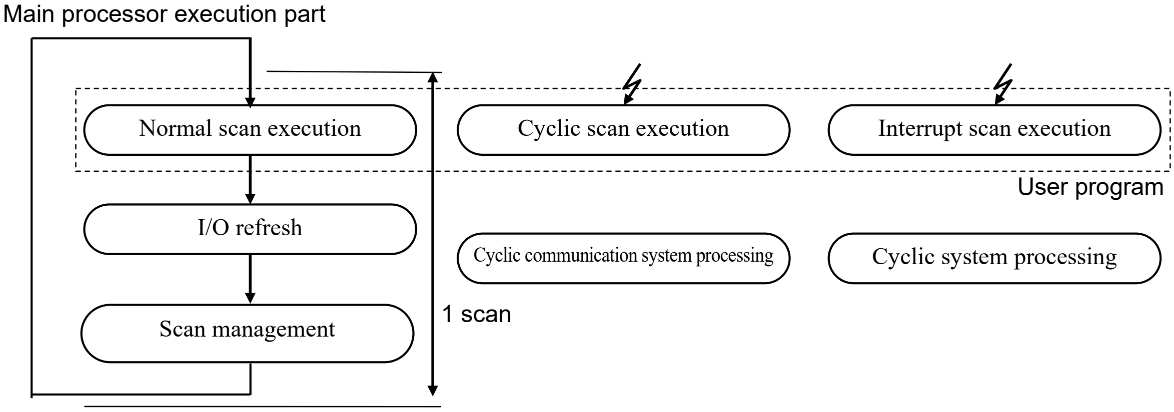


Figure 2.7 Overview of RUN operation

Reference

The user program is largely divided into three types: Normal scan, Cyclic scan, and Interrupt scan. They are executed in order of priority - cyclic scan, interrupt scan, and then normal scan.

(1) Normal scan

This scan program has two operation modes: Normal scan, which cyclically runs while the MICRO-EHV is in the RUN mode, and constant scan, which runs at a predetermined cycle. "Scan time" refers to a time from the beginning of a normal scan to the completion of system processing.

(2) Cyclic scan

This scan program is executed only once at each predetermined cycle. With the highest priority, a cyclic scan is executed after interrupting the operation even during I/O refresh in a normal scan.

(3) Interrupt scan

This scan program is executed only once when there is input to the input terminal assigned as interrupt input or there is input to the input terminal assigned as counter input and the current counter value is greater than the comparison value while the CPU is running.

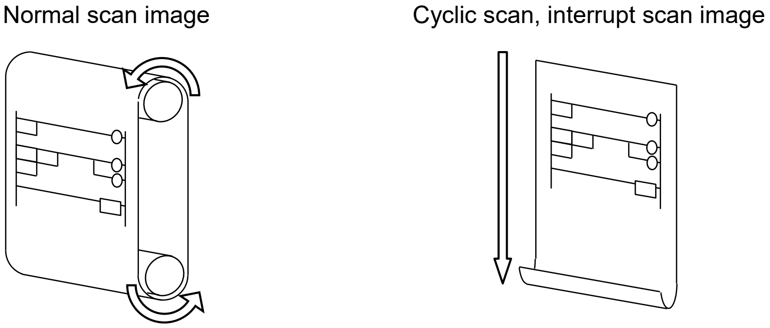


Figure 2.8 Normal scan and cyclic scan images

2.2.2 Operation during STOP

The figure below shows the operation overview when the MICRO-EHV is in the STOP state.

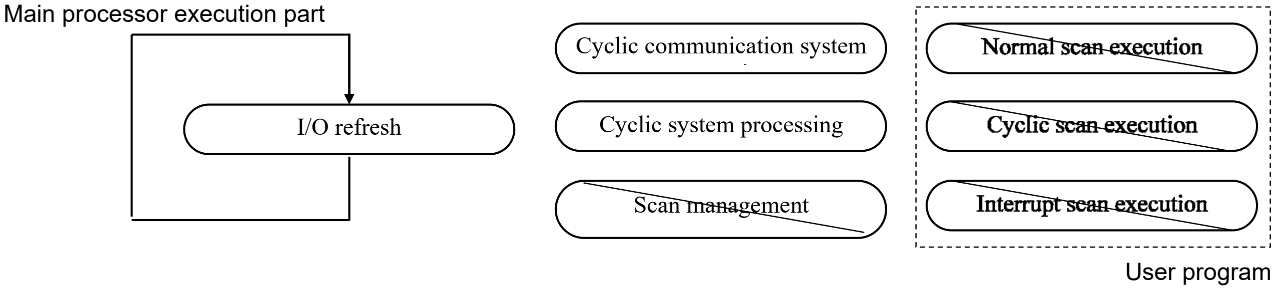


Figure 2.9 Overview of STOP operation

During STOP, the user program is not executed, but I/O refresh is executed.

Reference

Since the refresh process still runs during STOP, you can monitor the states of inputs or turn on outputs in the programming tool.

The MICRO-EHV clears the input/output data and the internal output information not specified as a power failure memory area when it starts running. Therefore, an output turned on during STOP is turned off when the MICRO-EHV starts running and then turned on and off according to the user program.

2.2.3 Data update

The PLC handles external inputs/outputs and data (internal outputs) used by the user program.

(1) External inputs/outputs

The MICRO-EHV updates input data in the data memory according to actual input signals and updates actual output signals according to output data in the data memory.

The PLC updates input/output data regardless of whether it is in the RUN or STOP state. Data is updated at a fixed cycle during STOP or at the end of the user program (scan END) during RUN. (Updating all input/output data at once during RUN is called the "refresh method".)

A ladder program accesses data on the data memory. For example, if you change an output value while a scan is running, the new value is used for subsequent scans. (The value at the scan END is actually output.)

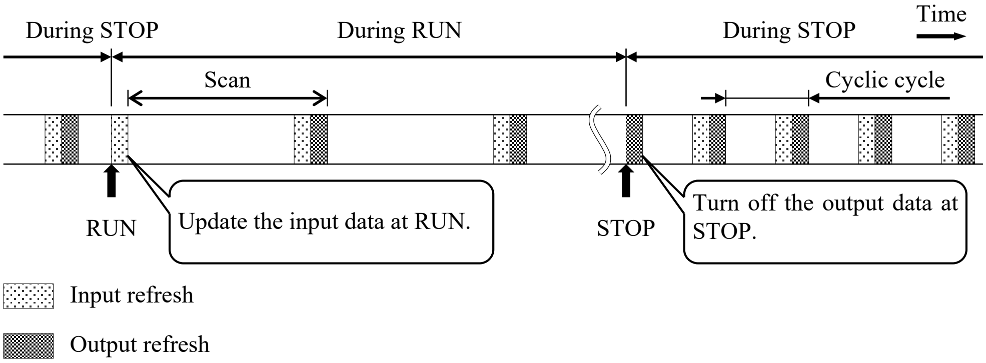


Figure 2.10 Conceptual drawing of external input/output refresh

Reference

Reading the state of an external input or updating it to an external output when an instruction is executed is called the "direct method". The MICRO-EHV adopts the refresh method, but you can refresh external input/output data during a scan by using an instruction to refresh I/Os.

The PLC also has the refresh prohibiting function, which allows you to temporarily prohibit both input refresh and output refresh using the Control Editor.

☞ To prohibit input refresh, select [Online] - [Operate CPU] - [Input refresh disabled] on the programming tool menu. To prohibit output refresh, select [Online] - [Operate CPU] - [Output refresh disabled] on the Control Editor menu.

(2) Internal output data

The values of internal output data are applied when they are set regardless of whether the MICRO-EHV is in the RUN or STOP state.

Internal output values are cleared, except for retentive areas, when the PLC starts running. When the PLC is stopped, the values before the stop are retained.

Reference

You can set to retain internal output values even when the PLC is turned off (retentive area setting). The battery is required for the retentive area function.

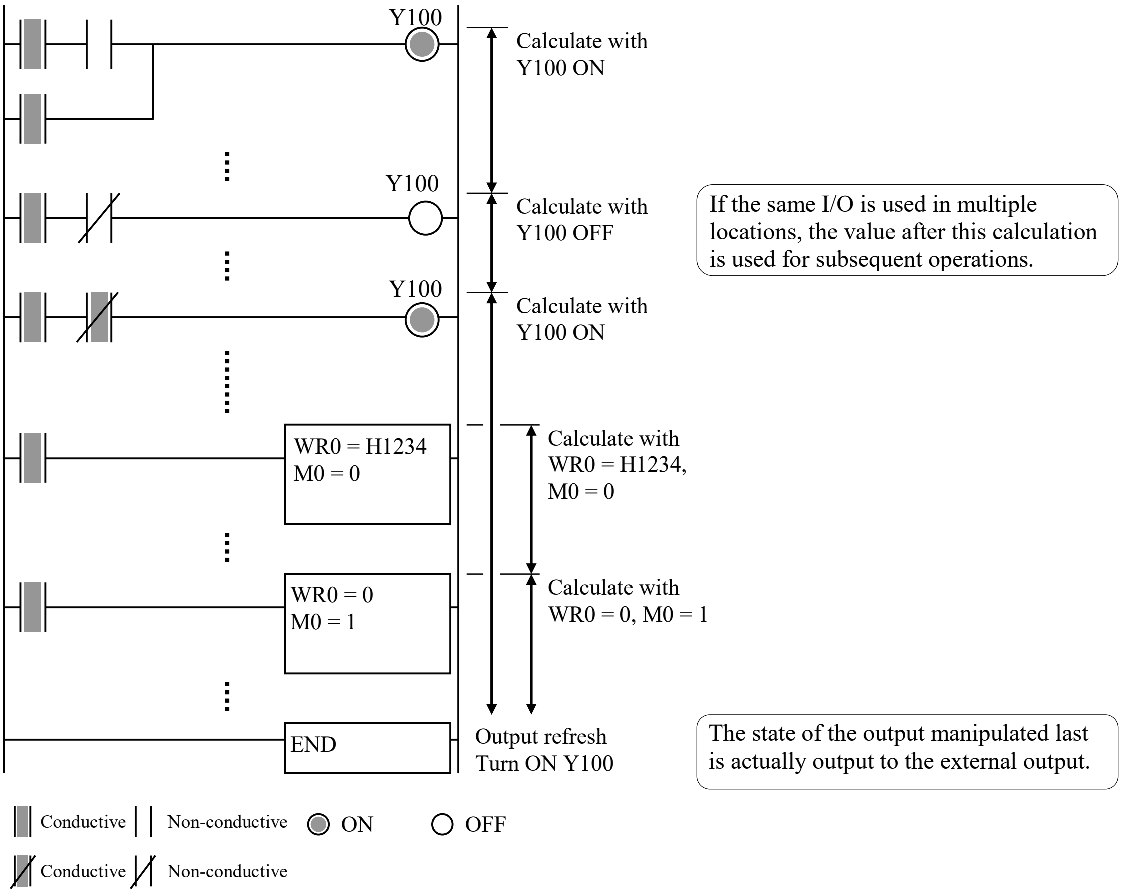


Figure 2.11 Data refresh example

2.2.4 System processing

A process to control MICRO-EHV operation is called system processing, which is executed periodically (at a cycle of 5 ms). There are the following types of system processing:

(1) Communication processing

This process communicates with peripheral equipment connected via communication ports on the MICRO-EHV.

(2) Error monitoring (self-diagnosis)

This process monitors if the MICRO-EHV has an error.

Failure reasons are categorized into four: Serious failure, medium failure, minor failure, and warning, and the operation after error detection varies depending on the target error level. In case of a serious failure, the MICRO-EHV stops the operation. In case of a medium failure, it stops program running. In case of a minor failure or warning, it stops program running or displays an error.

(3) Scan management

The main processor is used to detect errors in the operation part and change the user program. These processes are called scan management processing, which is executed once per scan.

Scan management is given the highest priority of all system processing operations.

Chapter 3 User Program

3.1 Structure of user program

The structure of the user program is shown in Figure 3.1.

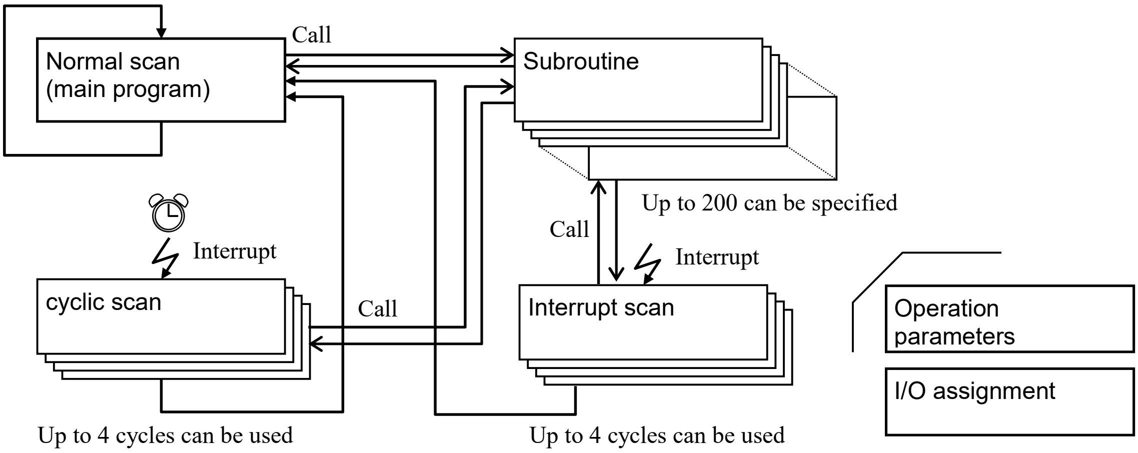


Figure 3.1 User program structure

The user program consists of the programming part, where contains combinations of instructions, operation parameter part, and I/O assignment table. Normally, the user program refers to the programming part, but when a program is transferred from the programming tool, the operation parameters and I/O assignment table are already added to the program. All elements are important to execute the program.

(1) Programming part

The programming part can be divided into four: "normal scan", "cyclic scan", "interrupt scan", and "subroutine".
In particular, a normal scan is required as the main program. For a cyclic scan, interrupt scan, and subroutine, use them as necessary.

(2) Operation parameters

Set the parameters related to MICRO-EHV operation and error display. The operation parameters are required for the user program and already set to the default value when a new program is created. So, even if they are not set, the program can be written into the MICRO-EHV. Change the parameters according to your purpose.

(3) I/O assignment table

I/O assignment refers to module installation information. The MICRO-EHV updates external I/O data and exchanges data with expansion units based on this information.
When inputting the program, if you specify the I/O numbers of external inputs/outputs without I/O assignment, an error occurs. Therefore, you need to set the table before program input.

3.2 Normal Scan

(1) Definition and operation of normal scan

A normal scan refers to operation of the main program and execution of the END instruction (scan END processing). When the MICRO-EHV starts running, it executes the main program (normal scan) from the beginning to the END instruction, which indicates the end of the program. Then, the PLC executes the main program again from scratch.

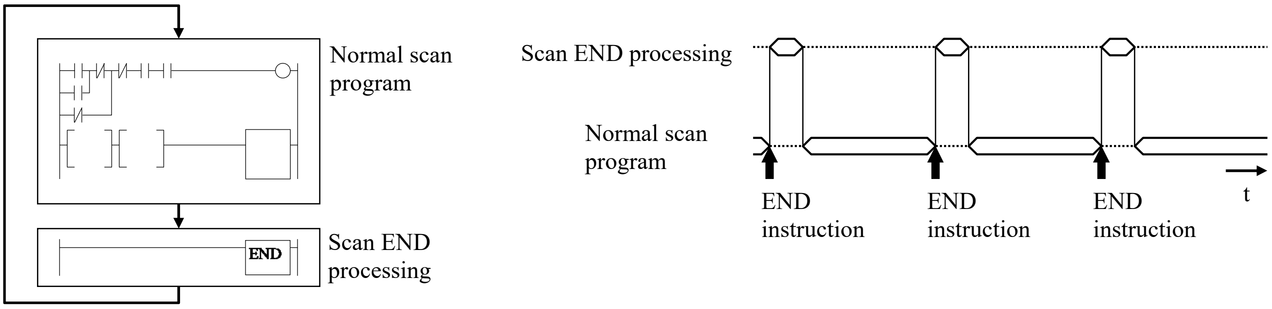


Figure 3.2 Overview of normal scan

Reference

When a normal scan is not combined with a cyclic scan, interrupt scan, or subroutine, the END instruction can be omitted.

(2) Reason for scan time error in normal scan

There are the following two reasons why a normal scan results in a scan time error:

A) The time of one scan is so long that the scan time exceeds the congestion check time.

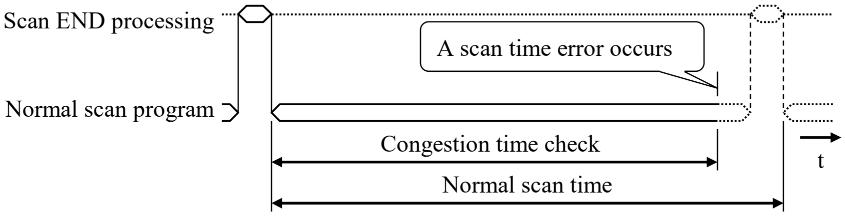


Figure 3.3 Reason for scan time error in normal scan (1)

B) A cyclic scan causes normal scan congestion more often, resulting in exceeding the congestion check time.

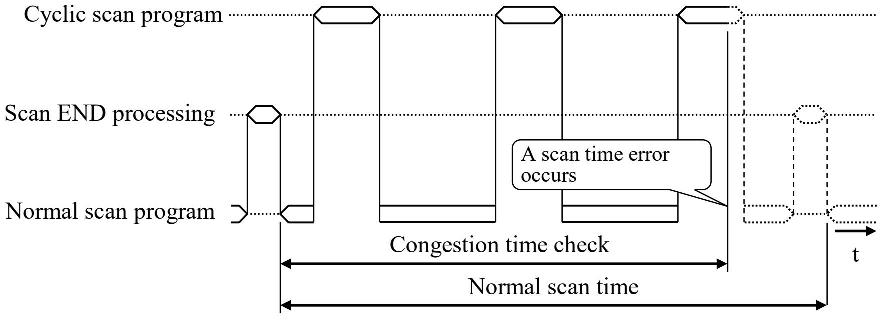


Figure 3.4 Reason for scan time error in normal scan (2)

A normal scan halts the process when a cyclic scan starts running but does not stop monitoring scan time errors. For this reason, an additional cyclic scan may cause a scan time error.

C) An interrupt scan causes normal scan congestion more often, resulting in exceeding the congestion check time.

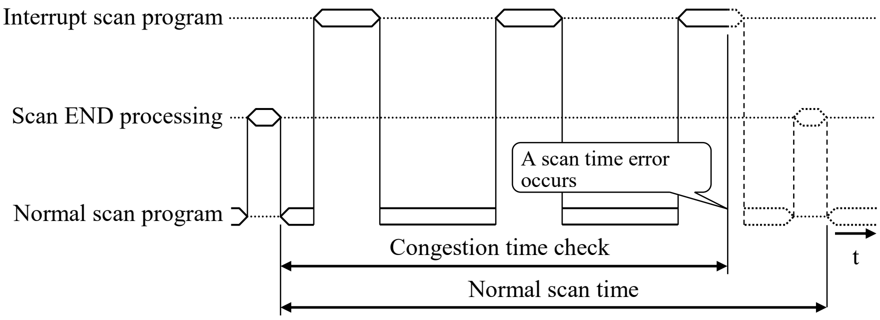


Figure 3.5 Reason for scan time error in normal scan (3)

A normal scan halts the process when an interrupt scan starts running but does not stop monitoring scan time errors. For this reason, an additional interrupt scan may cause a scan time error.

(3) Continuing operation on scan time errors

When you set the operation when normal scan time errors occur to [RUN] in the operation parameters, scan time errors are no longer detected. Therefore, a normal scan is executed regardless of the congestion monitoring time.

Caution

If you create a program that loops infinitely in a normal scan, the scan does not stop.

In such a case, I/Os are not refreshed because the scan END processing is not executed. You also cannot perform change in RUN because it is executed at the scan END.

(4) Useful programming tool function

In the programming tool, you can write a normal scan program in separate sheets.

This function allows you to create an easy-to-read program.

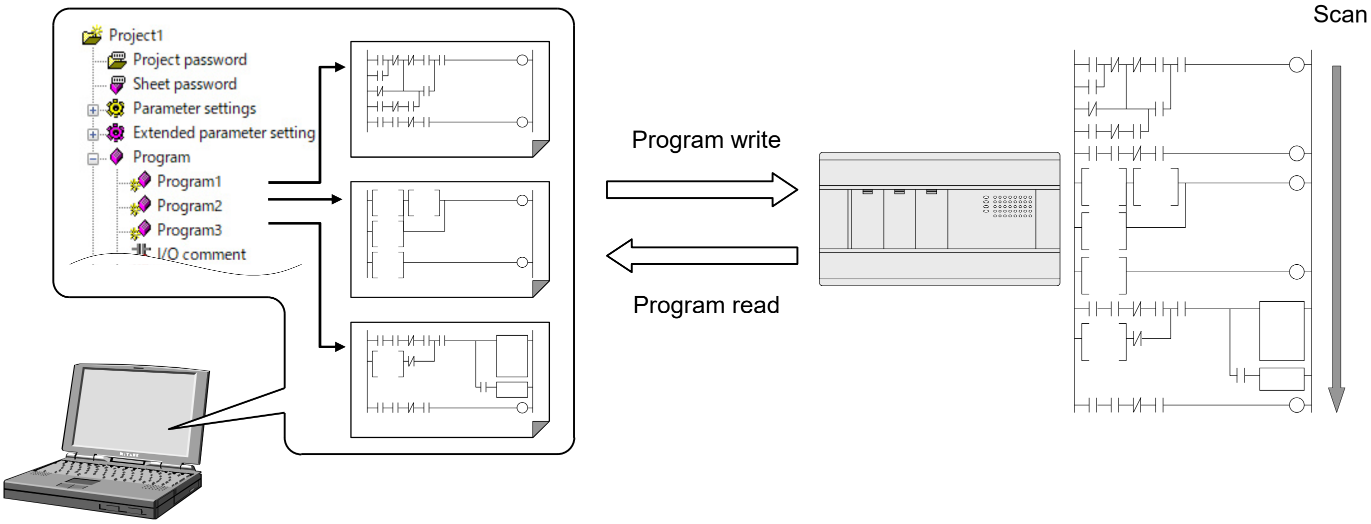


Figure 3.6 Normal scan sheet separation

Caution

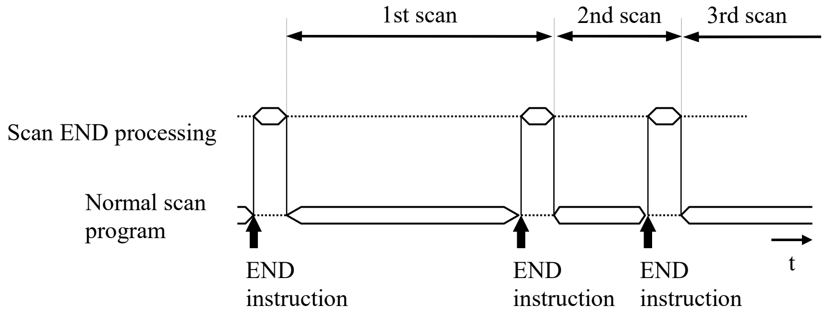
The MICRO-EHV executes sheets registered to the project tree in order from top to bottom. Please pay attention to the sheet order when creating a program to be processed in order.

If you write a subroutine, cyclic scan, or interrupt scan in a separate sheet, make sure that the END instruction is written at the end of the normal scan sheet.

(5) Constant scan function

The MICRO-EHV has the constant scan function, which executes a normal scan at a predetermined cycle like a cyclic scan. In a normal scan, the main program starts from the beginning immediately after being executed, but when the constant scan function is enabled, the main program is idled after being executed until the setting value is reached.

■ When the constant scan function is disabled (normal scan)



■ When the constant scan function is enabled

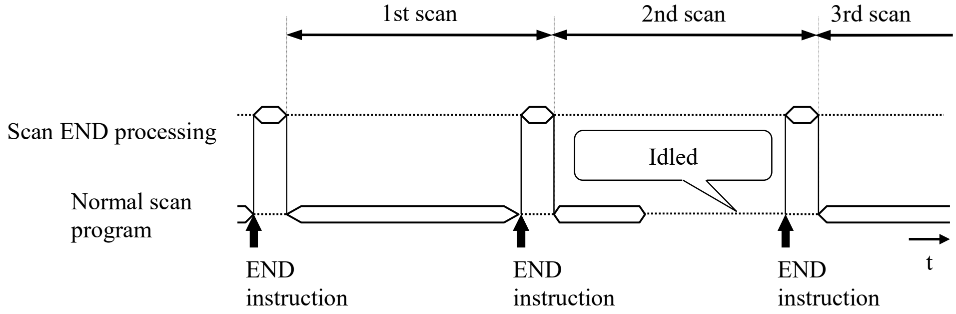
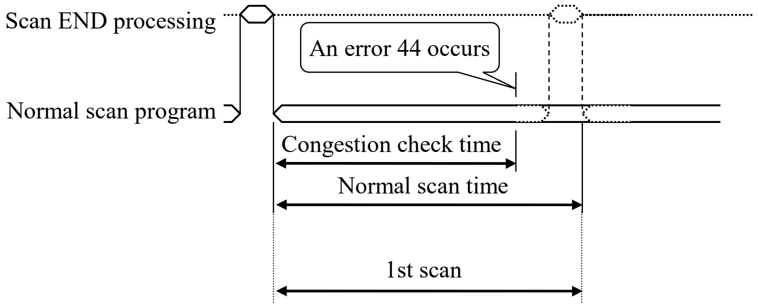


Figure 3.7 Constant scan function

When the constant scan function is enabled, a scan time error is detected if the END instruction is not executed within the constant scan cycle time. When you set the operation when scan time errors occur to [RUN], the scan is resumed from the next constant cycle after the END instruction is executed.

■ When the operation is set to [STOP] on scan time errors



■ When the operation is set to[RUN] on scan time errors

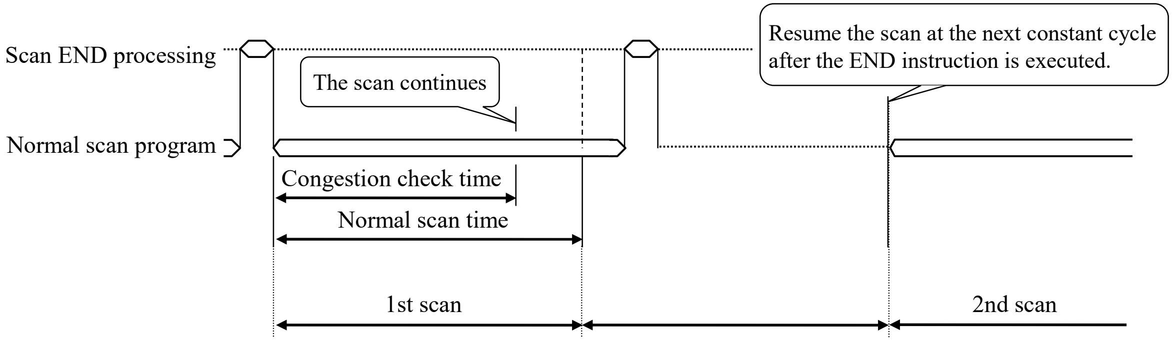


Figure 3.8 Behavior on scan time errors

3.3 Cyclic Scan

(1) Definition and operation of cyclic scan

A cyclic scan executes a program specified between the INT and RTI instructions at a predetermined cycle. When the MICRO-EHV reaches the set cycle after running, it executes sequentially from the INT instruction to the RTI instruction. You can create a cyclic scan program of up to four cycles. (Multiple cyclic scans of the same cycle cannot be created.) With higher priority than a normal scan, a cyclic scan starts after halting the normal scan at a cyclic scan timing even if a normal scan is running. When the cyclic scan is completed, the normal scan is restarted from where it was halted.

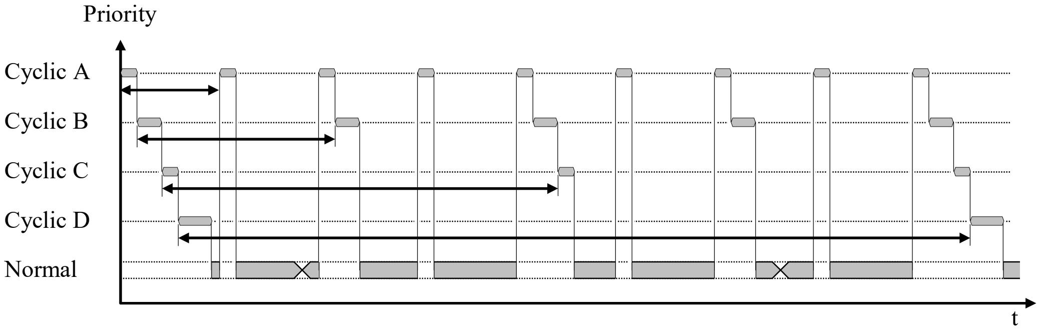


Figure 3.9 Scan execution timing (when multiple cyclic scans run simultaneously)

The following shows the behavior when multiple cyclic scans are executed. Of cyclic scans, the shorter cycle a scan has, the higher priority is given.

- ■ Simultaneous run of cyclic scans

If a cyclic scan with higher priority and a cyclic scan with lower priority start simultaneously, the higher priority one is executed first. When the higher priority one is finished, the lower priority one is executed.

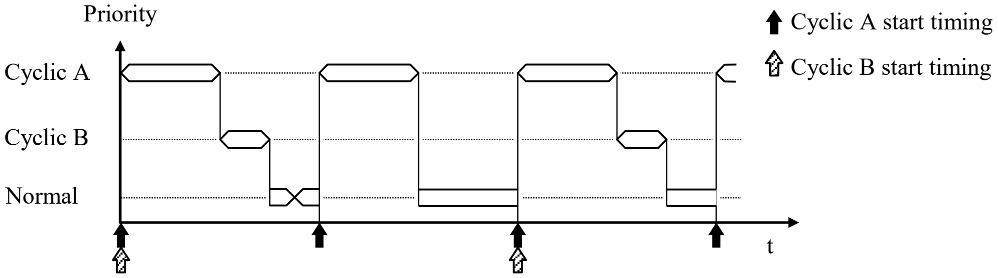


Figure 3.10 Simultaneous run of cyclic scans

- ■ Multiple interrupts of cyclic scans

If a cyclic scan with higher priority starts while a cyclic scan with lower priority is running, the lower priority one is halted, and the higher priority one is executed. When the higher priority one is completed, the lower priority one, which has been halted, is restarted.

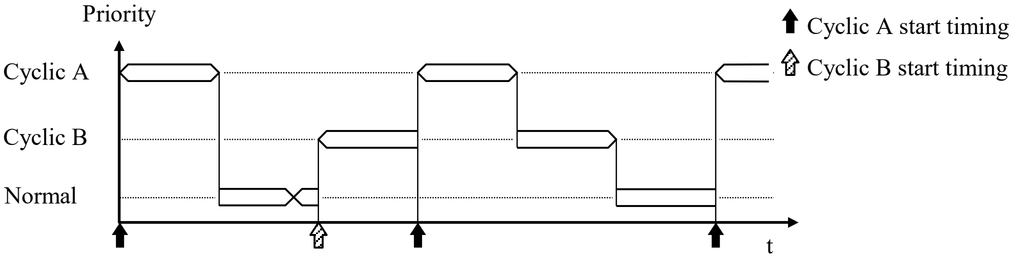


Figure 3.11 Multiple interrupts of cyclic scans

(2) Reason for scan time error in cyclic scan

- A scan time error (Error code: 45) occurs when a cyclic scan of the same cycle starts while a cyclic scan is already running.

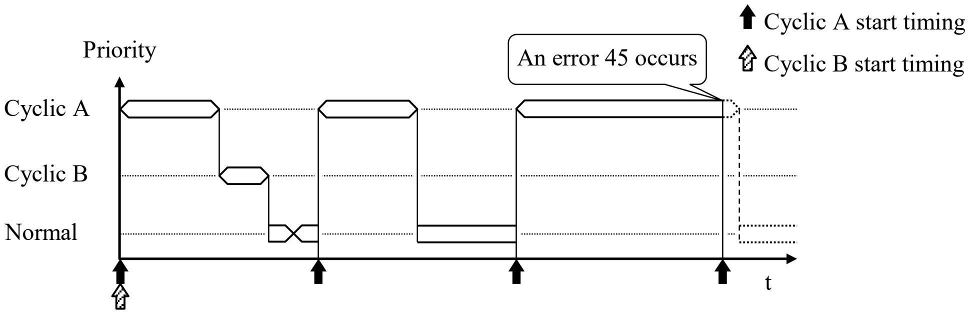


Figure 3.12 Reason for scan time error in cyclic scan [1]

- A scan time error (Error code: 4F) occurs when a cyclic scan with lower priority starts while a cyclic scan with higher priority is running.

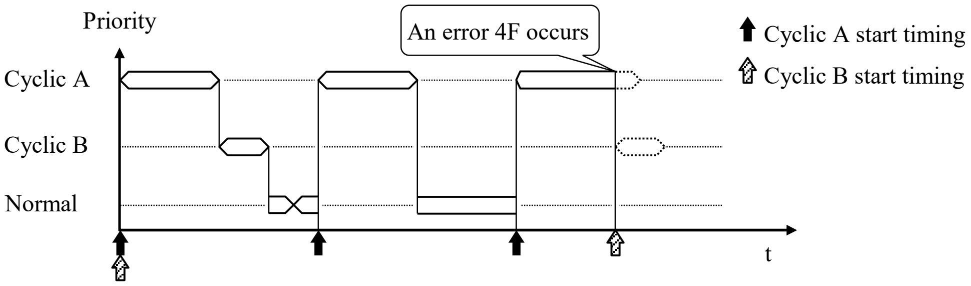


Figure 3.13 Reason for scan time error in cyclic scan [2]

If an error 45 and an error 4F occur at the same time, the error 4F takes precedence.

(3) Continuing operation on scan time errors

When you set the operation when cyclic scan time errors occur to [RUN] in the operation parameters, the operation continues even if a scan time error occurs. The behavior is different between errors 45 and 4F as follows:

■ Continuing operation when a scan time error 45 occurs

When an error 45 occurs, the cyclic scan is halted, and a cyclic scan of the same cycle is executed from the beginning.

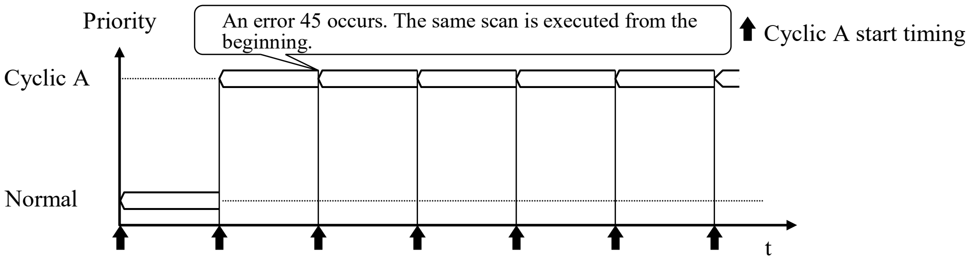


Figure 3.14 Continuing operation on scan time errors in cyclic scan [1]

Caution

When the operation is set to continue on scan time errors only for cyclic scan, if the situation shown in Figure 3.13 occurs, a normal scan time error occurs without executing a normal scan. For this reason, to continue the cyclic scan, you need to set the operation to continue on scan time errors even for normal scan. In this state, the scan END processing in a normal scan is not executed, so the external inputs/outputs are not refreshed. To use external input/outputs, write an instruction to refresh the external inputs/outputs in a cyclic scan. Please note that when the scan END processing is not executed, you cannot perform change in RUN.

■ Continuing operation when a scan time error 4F occurs

When an error 4F occurs, all cyclic scans are stopped, returning to a normal scan. At the next cyclic scan timing, the target cyclic scan starts.

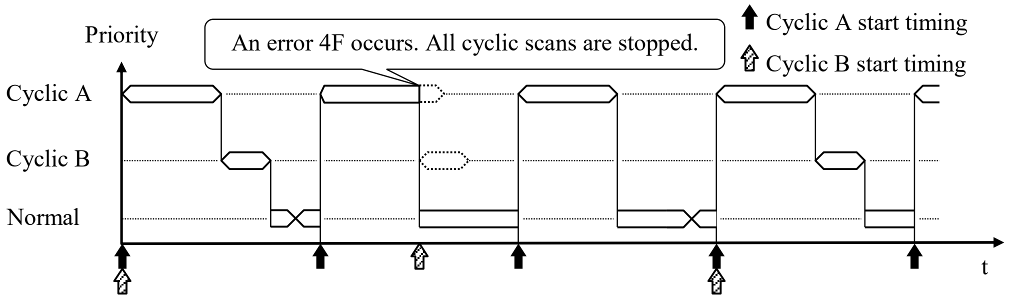


Figure 3.15 Continuing operation on scan time errors in cyclic scan [2]

Example: For cyclic scans INT (2) and INT (3) (Scan time: INT (2) = 1.2 ms, INT (3) = 0.5 ms)

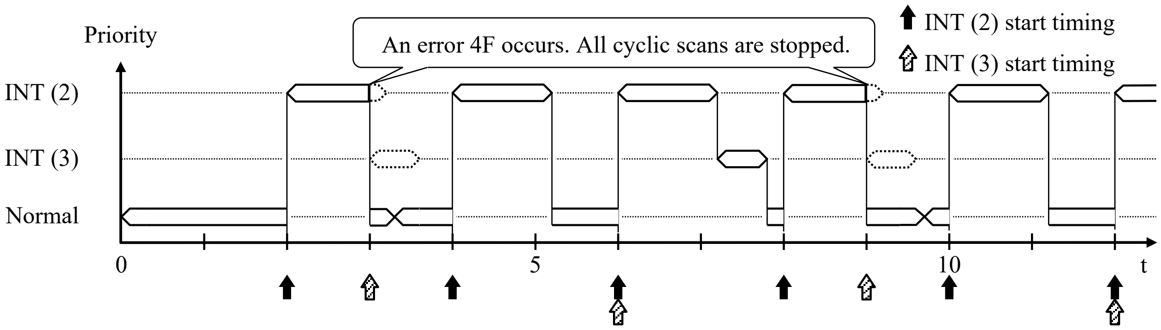


Figure 3.16 Example of continuing operation when scan time error occurs in cyclic scan

(4) Cycle setting for cyclic scans

If two or more cyclic scans are used, you can prevent an error 4F from occurring by setting cycles that meet one of the two conditions shown below.

■ Cycles must be integral multiples in descending order of priority.

When cycles A [ms], B [ms], C [ms], and D [ms] are set in order from shortest to longest, use the following formula:

B = A x m (m: Integer), C = B x n (n: Integer), D = C x p (p: Integer)

Example: When A is 3 ms: B is 6, 9, 12, 15, ..., 3 x m (m: Integer).

When A is 3 ms and B is 9 ms: C is 18, 27, 36, 45, ..., 9 x n (n: Integer).

When A is 3 ms, B is 9 ms, and C is 27 ms: D is 54, 81, 108, 135, ..., 27 x p (p: Integer).

■ The scan time of all cyclic scans but the one with the lowest priority must be less than 1 ms.

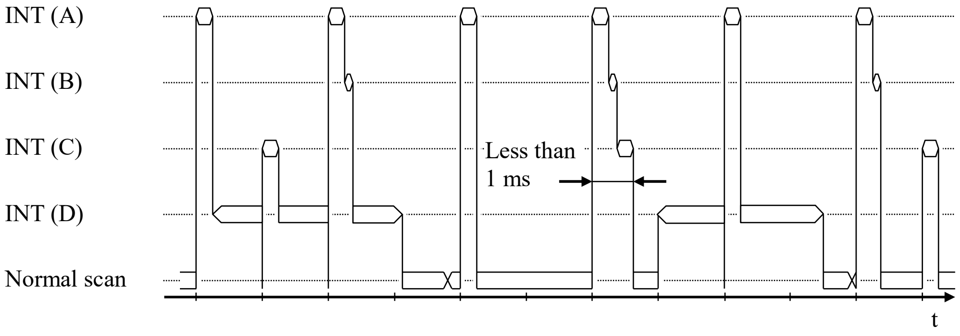


Figure 3.17 Cycle setting for cyclic scans

3.4 Interrupt Scan

(1) Definition and operation of interrupt scan

When there is input to the input terminal assigned as interrupt input or there is input to the input terminal assigned as counter input and the current counter value is greater than the comparison value while the CPU is running, a corresponding interrupt program (interrupt scan) starts. An interrupt scan by interrupt input executes an interrupt program from the XINT instruction to the XRTI instruction. An interrupt scan by current counter value match interrupt executes an interrupt program from CINTP to CRTIP and from CINTN to CRTIN.

If another factor interrupt is input while an interrupt scan is running, it starts when the current interrupt scan is completed. If multiple interrupts are input while an interrupt scan is running, they start in order from lowest to highest INT numbers when the current interrupt scan is completed.

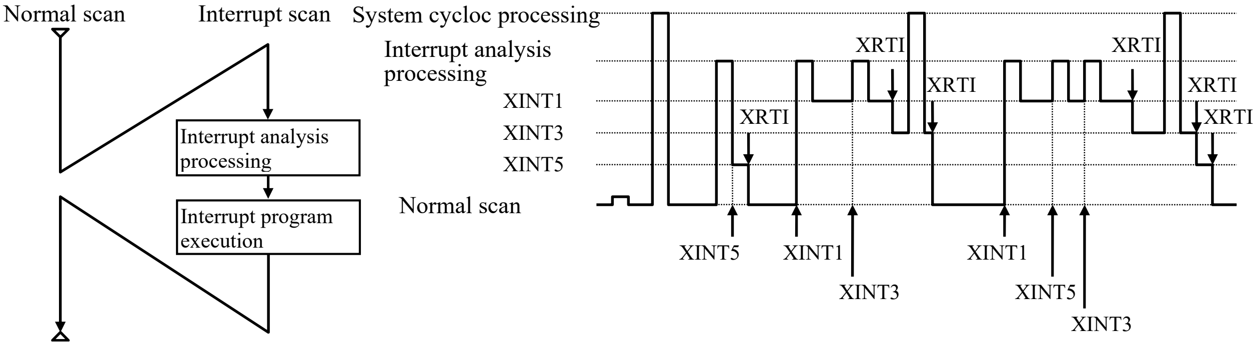


Figure 3.18 Interrupt scan operation 1

(2) Reason for scan time error in interrupt scan

If an interrupt of the same number is input while an interrupt scan is already running, an interrupt scan time error occurs. If an interrupt is often input, a normal scan error occurs because a normal scan is not executed.

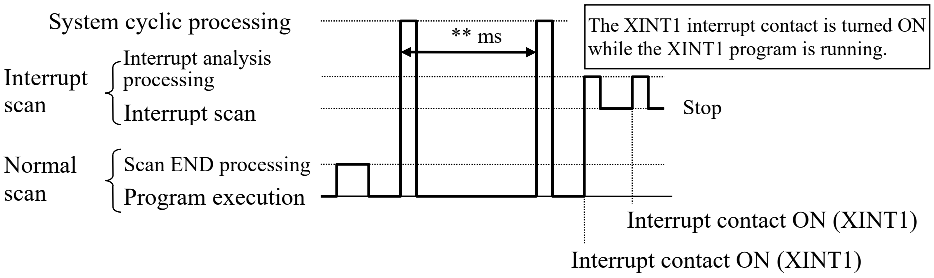


Figure 3.19 Interrupt scan operation 2

(3) Continuing operation on scan time errors

When the bit special internal output R7C2, the instruction to continue the operation on scan time errors, is set to ON, if an interrupt scan time error occurs, the interrupt scan is executed again from the beginning as a new run. Therefore, when an interrupt is often externally input, if the instruction to continue the normal scan operation is set to OFF, this scan is stopped as a normal scan time error. When the instruction to continue the normal scan operation is set to ON, only the interrupt scan continues with an interrupt scan time error. Please note that a normal scan is not executed in this state.

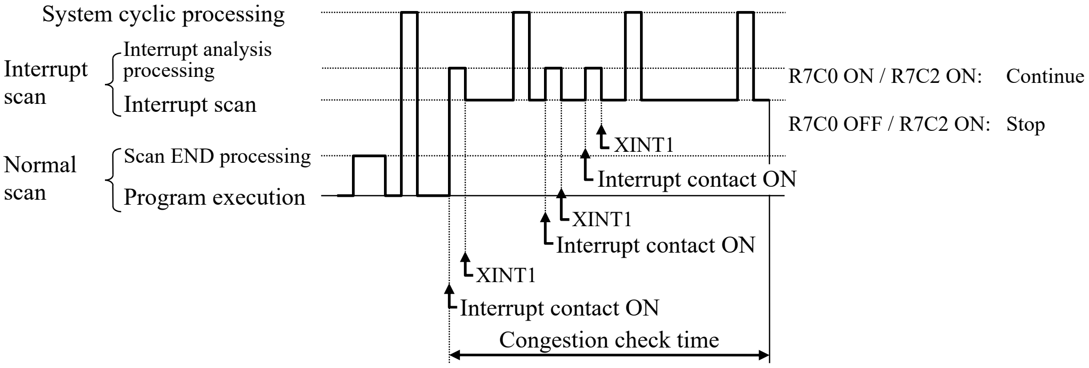


Figure 3.20 Behavior when setting to continue operation on scan time errors

3.5 Subroutine

(1) Definition and operation of subroutine

A subroutine refers to a packaged group of target processes.

A subroutine is a program specified between the SB n instruction (n: Subroutine No.) and the RTS instruction. When called, it executes sequentially from the SB instruction to the RTS instruction and returns to where it was called. You can call a subroutine from a normal scan or cyclic scan. (Even a subroutine of the same No. can be called.)

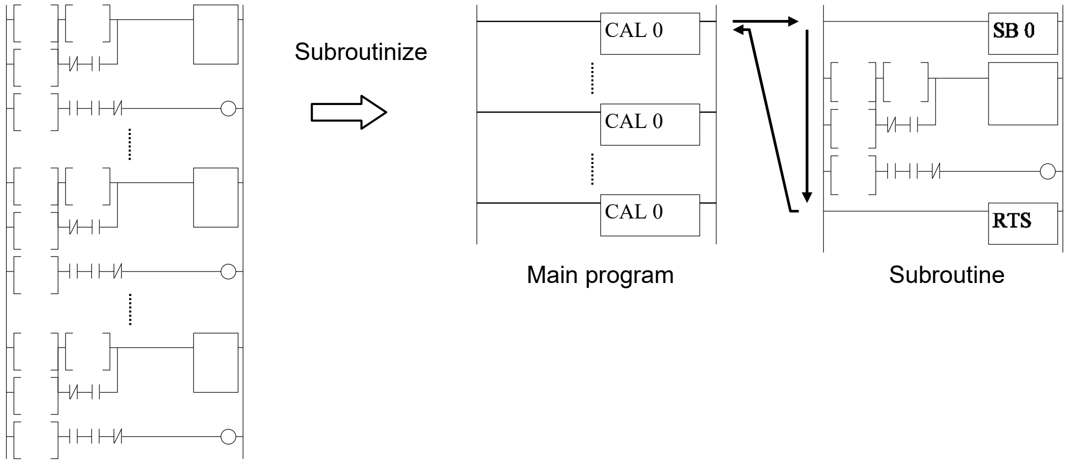


Figure 3.21 Implement optimal subroutine

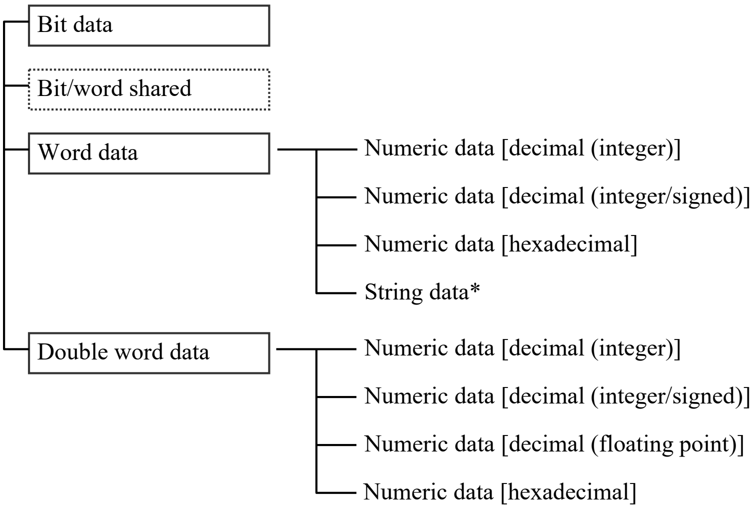
(2) Subroutine benefits

Subroutine has the following advantages:

- The user program can be easy to read.
- One function can be packaged and standardized. (It can be easily used for another system.)
- The program can be easily modified.
- A user program can be created separately.
- The user program size can be reduced.

3.6 Method to specify data

The figure below shows data used in the MICRO-EHV user program.



* Data handing is different from other word data. For details, see the description of string data.

Figure 3.22 Data types

(1) Bit data

In bit data, each bit contains on/off information. There are the following three bit data types:

i) Bit internal output

This data type (internal output) uses the bit-dedicated area.

ii) Bit/word shared internal output (external input/output)

In this data type (internal output, external input/output), bit and word data share the same area. Each word data consists of 16 bits.

iii) Bit specification of word internal output

A word internal output basically accesses the word-dedicated area in units of word, but you can manipulate a specific bit of a word internal output by specifying as follows:

Bit specification of word data: <Word I/O No.>.<Bit No.> (Bit No.: "0" to "F")

Example: To specify the 10th bit of WR100: WR100.A

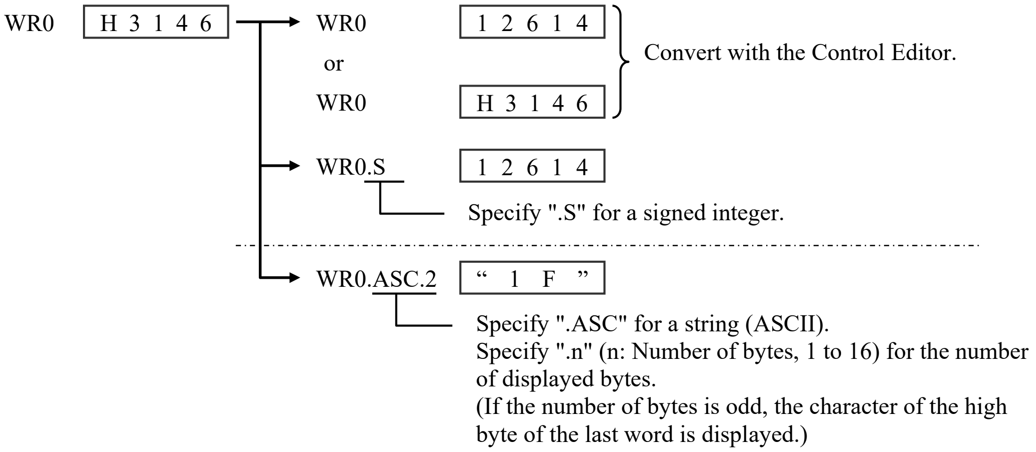
Reference

When a bit of word data is manipulated, the word data is processed in the MICRO-EHV. Therefore, the access speed becomes slower than the bit internal output or bit/word shared internal output.

(2) Word data

Word data consists of 16 bits and is accessed in units of word.

Word data can be stored as a signed integer or string by adding an extension to the word data address.



Convert with the Control Editor.

Specify ".S" for a signed integer.

Specify ".ASC" for a string (ASCII).
Specify ".n" (n: Number of bytes, 1 to 16) for the number of displayed bytes.
(If the number of bytes is odd, the character of the high byte of the last word is displayed.)

Figure 3.23 Word data description

Caution When specifying string data

<Word I/O No.>.ASC.<n> is used when storing string data into several words. If data of three or more bytes is specified, read and write data will be as follows:

WR0.ASC.9 = “Hello□EHV”

⇩

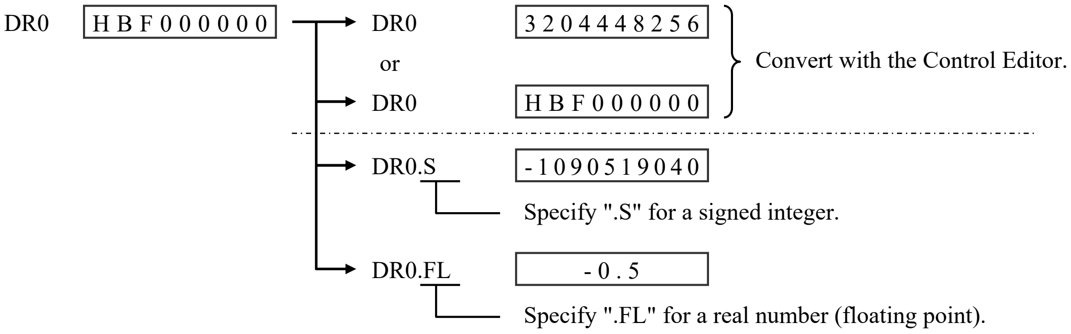
WR0	“H”	“e”		6	8	6	5
WR1	“l”	“l”		6	C	6	C
WR2	“o”	“ ”	=	6	F	2	0
WR3	“E”	“H”		4	5	4	8
WR4	“V”			5	6	0	0

Data is stored in order from high byte to low byte, starting with the specified word I/O No. If the number of bytes is odd, null (H00) is stored in the low byte of the last word I/O.

(3) Double word data

Double word data consists of 2 words/32 bits.

Double word data can be stored as a signed integer or floating point (single precision) by adding an extension to the double word data address.



Convert with the Control Editor.

Specify ".S" for a signed integer.

Specify ".FL" for a real number (floating point).

Figure 3.24 Double word data description

Reference Floating-point format

Floating-point instruction data uses an IEEE754 standard single-precision floating point.

The following describes how the IEEE754 single-precision floating point is represented internally.

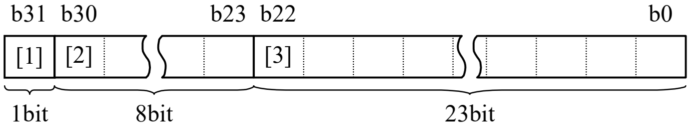


Figure 3.25 Floating-point format

[1] Sign 0: Positive, 1: Negative

[2] Exponent

Exponent (E)	Value (E') that is a power of two
FF	Indicates an overflow value.
FE	127
↓	↓
80	1
7F	0
7E	-1
↓	↓
01	-126
00	Treated as 0.

[3] Mantissa

Mantissa (M)	Value (M') that is a mantissa
7FFFFF	$(1.11...11)_2$
7FFFFE	$(1.11...10)_2$
↓	↓
1	$(1.00...01)_2$
0	$(1.00...00)_2$

■ Formula

The floating point (F) can be calculated from the following formula using the above sign (S), exponent (E), and mantissa (M):

$(F) = (-1)^{S} \times (1 + M \times 2^{-23}) \times 2^{E-7FH} = (-1)^{S} \times M' \times 2^{E'}$

■ Available floating-point range

Hexadecimal representation		Floating-point representation	Note
High word	Low word		
H7F7F	HFFFF	$+3.402823... \times 10^{38}$	Maximum
H0080	H0000	$+1.175494... \times 10^{-38}$	The absolute value of positive numbers is the minimum.
↓		↓	Within this range, the value is treated as 0.
H8080	H0000	$-1.175494... \times 10^{-38}$	The absolute value of negative numbers is the minimum.
HFF7F	HFFFF	$-3.402823... \times 10^{38}$	Minimum

3.6.1 External I/O

As shown in Table 3.1, external inputs and external outputs are represented as symbols X and Y, respectively, and a fixed address is assigned each to them according to the module installation position.

Table 3.1 List of external input/output categories and data types

I/O category	Input/output category	Data type	Note
X	External input	Bit (1 bit)	The address is decimal between 0 and 95.
WX		Word (16 bits)	The address is hexadecimal. 16-bit synchronicity is guaranteed because 16-bit data is processed at once.
DX		Double word (32 bits)	The address is hexadecimal. 32-bit synchronicity is not guaranteed.
Y	External output	Bit (1 bit)	The address is decimal between 0 and 95.
WY		Word (16 bits)	The address is hexadecimal. 16-bit synchronicity is guaranteed because 16-bit data is processed at once.
DY		Double word (32 bits)	The address is hexadecimal. 32-bit synchronicity is not guaranteed.

External input/output I/O numbers are represented based on the following rules.

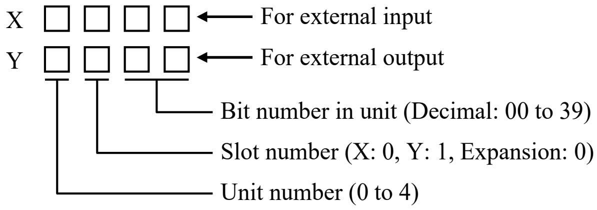
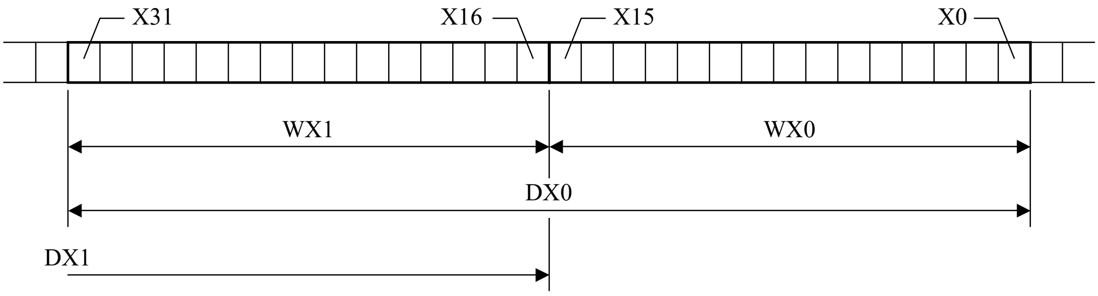


Figure 3.26 External input/output I/O numbering rules

A word of external input/output consists of 16 corresponding bits while a double word consists of 32 bits.

Example: Correspondence between DX0, WX0, and X0 to X15



Reference

I/O assignment varies depending on the MICRO-EHV model. The table below shows the external input / output specifications for each model.

■ Basic unit / Expansion unit

Unit			I/O assignment	20-point unit	40-point unit	64-point unit
Basic unit	Digital	Input	Slot 0: X48	X0 to X11	X0 to X23	X0 to X39
		Output	Slot 1: Y32	Y100 to Y107	Y100 to Y115	Y100 to Y123
			Slot 2: Empty	-	-	-
Expansion unit 1	Digital	Input	Unit 1	X1000 to X1015		
		Output	Slot 0: B1/1	Y1016 to Y1031		
	Analog	Input	Unit 1	WX101 to WX104		
		Output	Slot 0: FUN 0	WY106 to WY107		
Expansion unit 2	Digital	Input	Unit 2	X2000 to X2015		
		Output	Slot 0: B1/1	Y2016 to Y2031		
	Analog	Input	Unit 2	WX201 to WX204		
		Output	Slot 0: FUN 0	WY206 to WY207		
Expansion unit 3	Digital	Input	Unit 3	X3000 to X3015		
		Output	Slot 0: B1/1	Y3016 to Y3031		
	Analog	Input	Unit 3	WX301 to WX304		
		Output	Slot 0: FUN 0	WY306 to WY307		
Expansion unit 4	Digital	Input	Unit 4	X4000 to X4015		
		Output	Slot 0: B1/1	Y4016 to Y4031		
	Analog	Input	Unit 4	WX401 to WX404		
		Output	Slot 0: FUN 0	WY406 to WY407		

■ 64-point expansion unit (EH-A64EDR, EH-D64EDR, EH-D64EDT, EH-D64EDTPS)

Unit			I/O assignment	20-point unit	40-point unit	64-point unit
Expansion unit 1	Digital	Input	Slot 0: X48	X1000 to X1039		
		Output	Slot 1: Y32	Y1100 to Y1123		
			Slot 2: Empty	-		
Expansion unit 2	Digital	Input	Slot 0: X48	X2000 to X2039		
		Output	Slot 1: Y32	Y2100 to Y2123		
			Slot 2: Empty	-		
Expansion unit 3	Digital	Input	Slot 0: X48	X3000 to X3039		
		Output	Slot 1: Y32	Y3100 to Y3123		
			Slot 2: Empty	-		
Expansion unit 4	Digital	Input	Slot 0: X48	X4000 to X4039		
		Output	Slot 1: Y32	Y4100 to Y4123		
			Slot 2: Empty	-		

■ Positioning expansion unit (EH-A2EP, EH-D2EP)

A positioning expansion unit uses I/O assignment of two expansion units.

Unit		I/O assignment	20-point unit	40-point unit	64-point unit
Expansion unit 1	Input	Unit 1 Slot 0: X8W	WX100 to WX107		
	Output	Unit 2 Slot 0: Y8W	WY200 to WY207		
Expansion unit 2	Input	Unit 2 Slot 0: X8W	WX200 to WX207		
	Output	Unit 3 Slot 0: Y8W	WY300 to WY307		
Expansion unit 3	Input	Unit 3 Slot 0: X8W	WX300 to WX307		
	Output	Unit 4 Slot 0: Y8W	WY400 to WY407		

3.6.2 Internal output

An internal output is a register that can be used in the user program.

There are the bit-dedicated area (R), word-dedicated area (WR), and bit/word shared area (M/WM) for internal outputs.

Internal outputs also have an area that can be accessed by the user and an area used for a specific purpose called a "special internal output". A special internal output is used for system setting and state display.

☞ For details on special internal outputs, refer to "Appendix 2 List of Special Internal Output".

Table 3.2 List of internal outputs

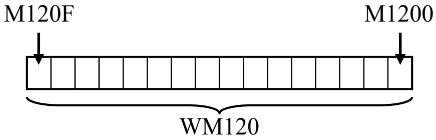
CPU model I/O type		MVL-*20/40/64** (Standard model)	MVH-*40/64** (High Function model)
Bit		1,984 bits (R0 to R7BF)	
Word (WR)		32,768 words (WR0 to WR7FFF)	
Bit/word shared (WM)		32,768 bits, 2,048 words (M0 to M7FFF, WM0 to WM7FF)	
Special internal output	Bit	2,112 bits (R7C0 to RFFF)	
	Word	4,096 words (WRF000 to WRF1FF)	

Internal output I/O numbers are represented based on the following rules.

Table 3.3 List of internal output I/O numbering rules (1/2)

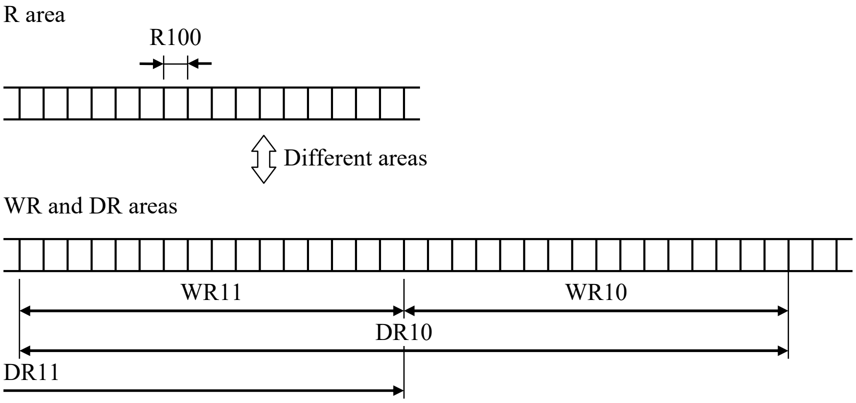
Data type	Numbering rule	
Bit-dedicated type		R □□□ Normal area H000 to H7BF Special area H7C0 to H7FF Both hexadecimal
Word-dedicated type	<For word>	W R □□□□ Normal area H0000 or above Special area HF000 or above Both hexadecimal
	[Bit specification]	W R □□□□.□ Specify ".n". (n: Bit No., 0 to F)
	[Signed integer]	W R □□□□. S Specify ".S".
	[String specification]	W R □□□□. A S C . n Specify ".n". (n: Number of bytes, 1 to 32 [decimal]) Specify ".ASC".
	<For double word>	D R □□□□ Normal area H0000 or above Special area HF000 or above Sequential 2-word WR representation Both hexadecimal
	[Signed integer]	D R □□□□. S Specify ".S".
	[Real number (floating point)]	D R □□□□. FL Specify ".FL".

Table 3.4 List of internal output I/O numbering rules (2/2)

Data type	Numbering rule	
Bit/word shared type	<For bit>	M □□□□□ H00000 or above / H0000 or above Hexadecimal
	<For word>	WM □□□□ H0000 or above Hexadecimal 
	* No bit specification is available for the bit/word shared type.	
	[Signed integer]	WM □□□□.S Specify ".S".
	[String specification]	WM □□□□.ASC.n Specify ".n". (n: Number of bytes, 1 to 32 [decimal]) Specify ".ASC".
	<For double word>	DM □□□□ H0000～ Both hexadecimal Sequential 2-word WR representation
	[Signed integer]	DM □□□□.S Specify ".S".
	[Real number (floating point)]	DM □□□□.FL Specify ".FL".

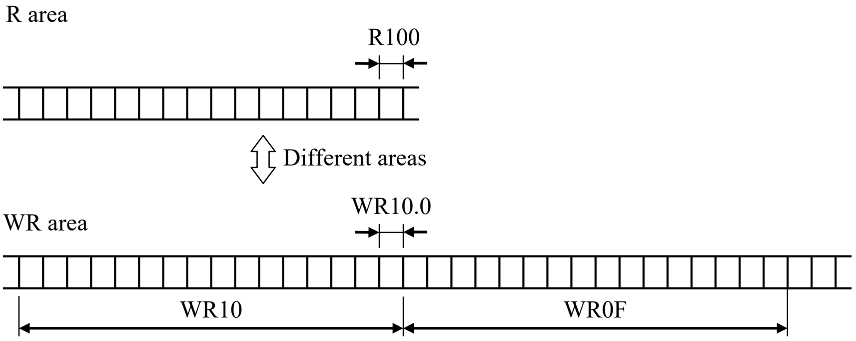
The internal output R area is different from the WR and DR areas.

Example: Correspondence between R100 and WR10/DR10



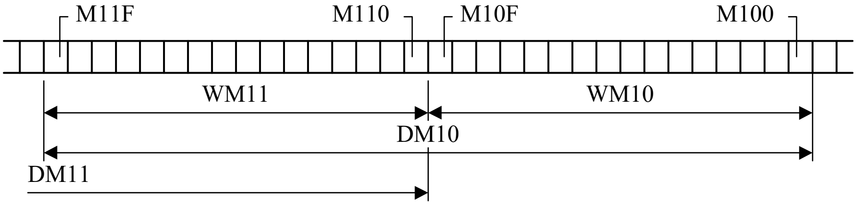
Caution

The MICRO-EHV can select a specific bit from word data to access.



The internal outputs M, WM, and DM use the same area. (Each bit can be manipulated via word I/O.)

Example: Correspondence between M100 and WM10/DM10



3.7 Program capacity

The program size is calculated in units of "step".

A user program is created in combinations of instructions for various purposes, but the number of steps used for each instruction is different. (For details on the number of steps for each instruction, refer to "Chapter 5 Command Specifications ".)

The total number of steps of all programs, such as a normal scan, cyclic scan, interrupt scan, and subroutine, is written into the MICRO-EHV. Create a program so that the number of steps does not exceed the upper limit predetermined for the MICRO-EHV.

Table 3.5 Program size

Item	MVL-*20/40/64** (Standard model)	MVH-*40/64** (High function model)	[Reference] EH-A64DR
Program size	16k step		16k step
Instruction size	48 bits per step		32 bits per step
Comment size	128 kB		(No comment storage function)

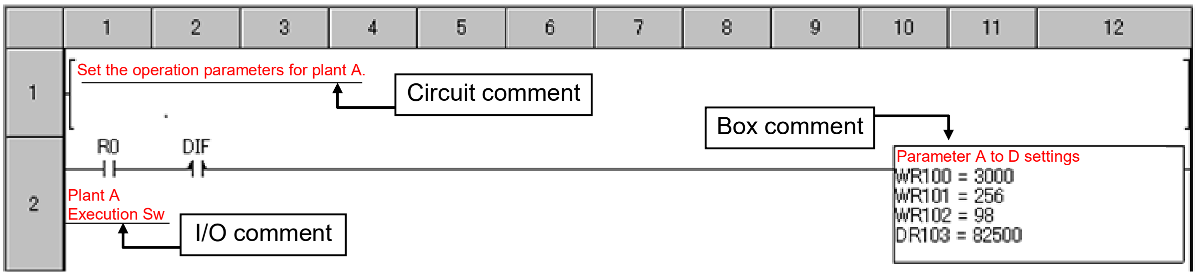
Reference

Comments written in the user program are not counted in the number of steps.

The MICRO-EHV stores not only the user program but also comments into the internal backup memory, and comments are managed as the "comment size". Make sure that the comment size does not exceed the upper limit when creating a user program.

3.8 Comments

To increase the program visibility, you can enter comments into I/Os to be used and internal outputs. No comments are used for PLC operation.



Set the operation parameters for plant A.

Plant A
Execution Sw

Parameter A to D settings
WR100 = 3000
WR101 = 256
WR102 = 98
DR103 = 82500

Figure 3.27 Program with comments written in

The table below shows available comment types.

Table 3.6 Comment types and purposes

No.	Type	Number of characters*	Purpose
1	Circuit comment	128	This comment is dedicated to one circuit. It is used as an index when separating a program by purpose or used to write the descriptions of processes executed in the previous and next circuits.
2	Box comment	64	This comment is written in a processing box. It is used to write the details, notes, and parameter description of a process specified in a processing box.
3	I/O comment	32	This comment is linked to an input/output or internal output. It is used to write an I/O purpose (connected sensor CH number or switch name) for future maintenance.

* The maximum number of allowable characters does not change regardless of whether double-byte or single-byte characters are used.

3.9 Backup of program

The user program, comments, and settings (NTP setting, serial communication setting, IP address, Ethernet communication port, etc.) are automatically stored into the backup memory inside the MICRO-EHV.

Even when the battery is weak or not installed, the user program and comments can be retained.

To retain power failure memory area data, you need the battery because the data is not stored into the backup memory.

Caution

The MICRO-EHV backs up the user program into the backup memory. To reduce the program transfer time, the user program is temporarily moved to the operation execution memory, which is treated as the completion of transfer. Then, the program is backed up to the backup memory, so please wait about three minutes after program transfer before turning off the PLC. Otherwise, a user memory error (31H) may occur. (You can check the completion of transfer to the backup memory with the special internal output R7EF.) The STATUS LED indicates that the MICRO-EHV is writing to the backup memory. The following shows the STATUS LED display during write to the backup memory.

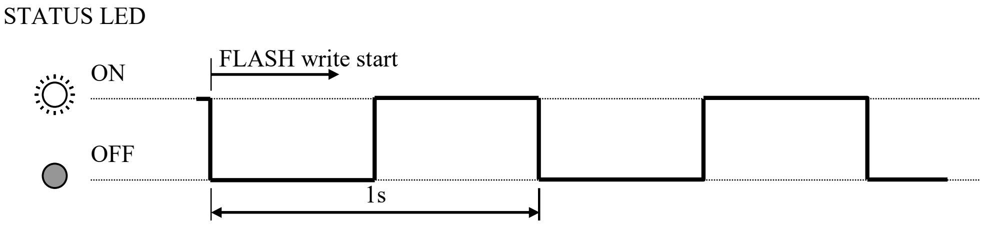


Figure 3.28 STATUS LED display during backup

MEMO

Chapter 4 Procedure to Create User Program

4.1 Flow to create user program

The figure below shows the procedure for creating a user program.

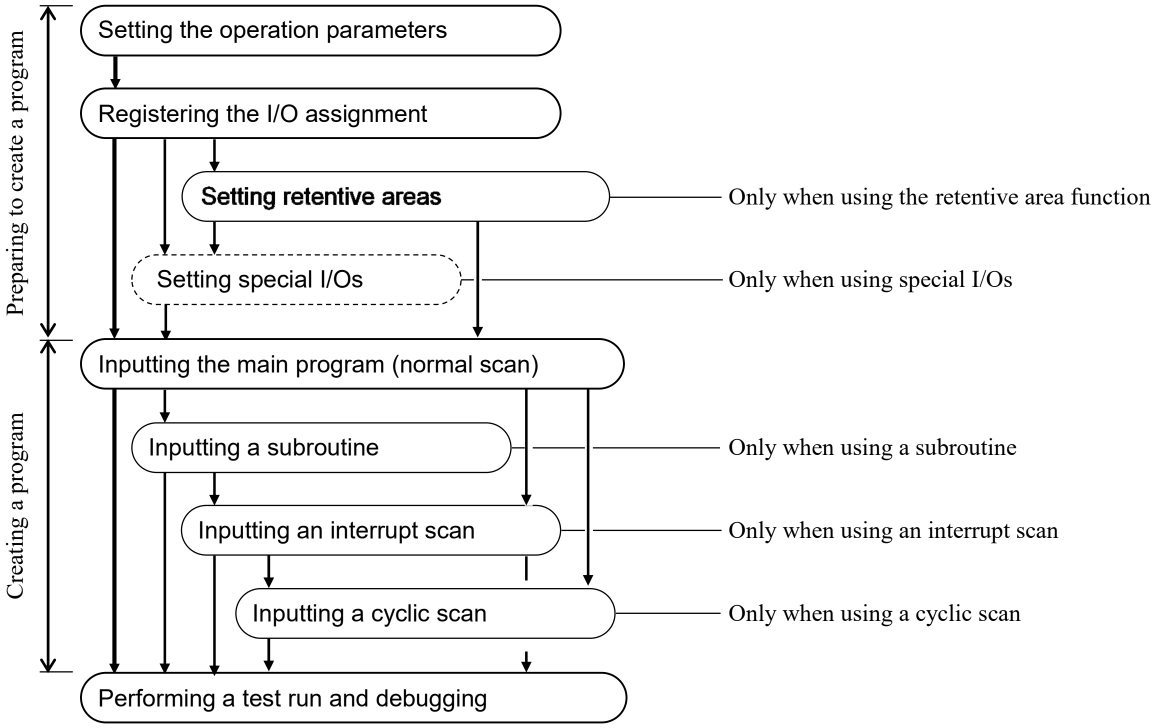


Figure 4.1 Flow of creating user program

Create a basic user program according to the following flow:

Setting the operation parameters ➔ Registering the I/O assignment ➔ Inputting a normal scan ➔ Performing a test run

I/O assignment and normal scan input are required. (You cannot create a program containing only a subroutine, cyclic scan, or interrupt scan.) Since the operation parameters are set to the default values when creating a new program, the setting is unnecessary if you do not change the values.

4.2 Preparation to create program

The following shows elements that need to be set before creating a user program:

(1) Operation parameters

The table below shows the parameters to be set as operation parameters.

Table 4.1 List of operation parameters

Item	Description		Default
Operation definition input	Select a bit to be used as an operation definition input from external inputs (X) or bit internal outputs (R, M). ☞ For details on the operation definition input function, see "2.2 Running and Stopping MICRO-EHV".		Unselected
Constant scan function	Enable or disable the constant scan function. ☞ For details on the constant scan function, see "3.2 Normal Scan".		Unselected
Digital filter	Set the number of digital filter samplings. Specify the number of samplings in increments of 0.5 ms (0 to 40, 0 to 20 ms). If 0 is set, it is considered no filtering. If a value equal to or greater than 41 is set, the number of samplings is treated as 40. This setting is applied to all input terminals in the basic unit.		4 (2 ms)
Congestion check time	Set the time before a normal scan causes a scan time error. ☞ For details on scan time errors, see "3.2 Normal Scan".		10 ms
CPU operation mode on errors	Under I/O configuration error	Enable or disable the operation (RUN) when I/O assignment is unmatched. ☞ For details, see the next page.	Disabled
	Under scan time error (normal scan)	Set whether to continue or stop the operation when a scan time error occurs in a normal scan. ☞ For details, see "3.2 Normal Scan".	Stop
	Under scan time error (cyclic scan)	Set whether to continue or stop the operation when a scan time error occurs in a cyclic scan. ☞ For details, see "3.3 Cyclic Scan".	Stop
	Under scan time error (interrupt scan)	Set whether to continue or stop the operation when a scan time error occurs in an interrupt scan. ☞ For details, see "3.4 Interrupt Scan".	Stop
Data memory backup	Enable or disable the data memory backup function. ☞ For details, see "Chapter 6 Data Memory Backup Function" in the User's Manual.		Disabled
Error display (LED control)	Display level	Set the OK LED status when an error 7x occurs. ☞ For details, see "Appendix 2 Error Codes".	Selected
Display level setting	When a warning error occurs	Set the error code level to be displayed on the OK LED. ☞ For details, see "Appendix 2 Error Codes".	Level 0

(2) I/O assignment

There are the following two methods to input I/O assignment:

i) Specify units to be installed on the I/O assignment setting screen.

ii) Use the installed I/O read function to create an I/O assignment table.

The MICRO-EHV has a function to read the types of installed units.

When all units to be used are ready, if you connect the programming tool to the MICRO-EHV with the units installed and read the installed I/Os on the I/O assignment setting screen, the assignment information of the installed I/Os is loaded, automatically creating the I/O assignment table.

Caution

I/O assignment information is managed using codes like "X16" and "Y32". When the installed I/O read function is used, the code corresponding to each unit is loaded. (The unit model is not loaded.)

For details on the I/O assignment code of each unit, see the MICRO-EHV Hardware Manual.

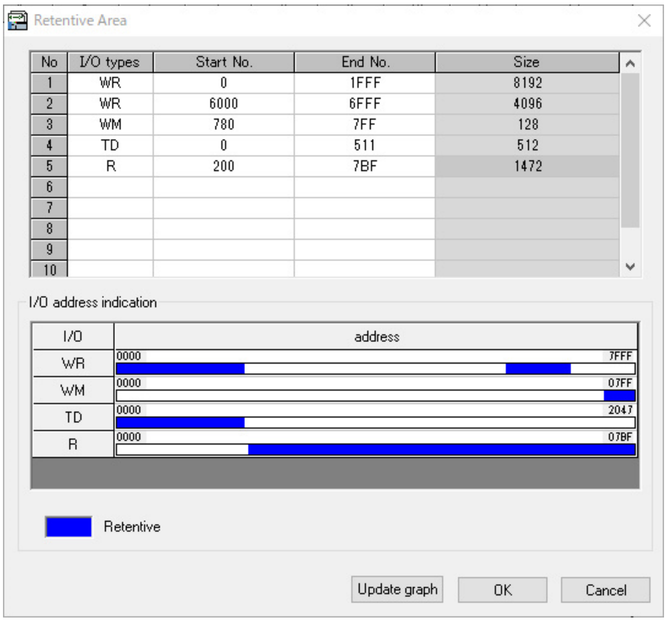
Reference

If you verify the program operation with not all units installed, set the operation to [RUN] when I/O assignment is unmatched in the operation parameters. This setting prevents the PLC from stopping with an error even if the I/O information written from the programming tool does not match the installed I/Os.

Be sure to set the operation to [STOP] when I/O assignment is unmatched in the operation parameters before going live with the system.

(3) Retentive area settings

To retain data when the PLC is turned OFF, specify the area of the target I/O as a power failure memory area. Up to 16 power failure memory areas of a specific range can be set for internal outputs and timers. You can also specify multiple areas for internal outputs of the same type.

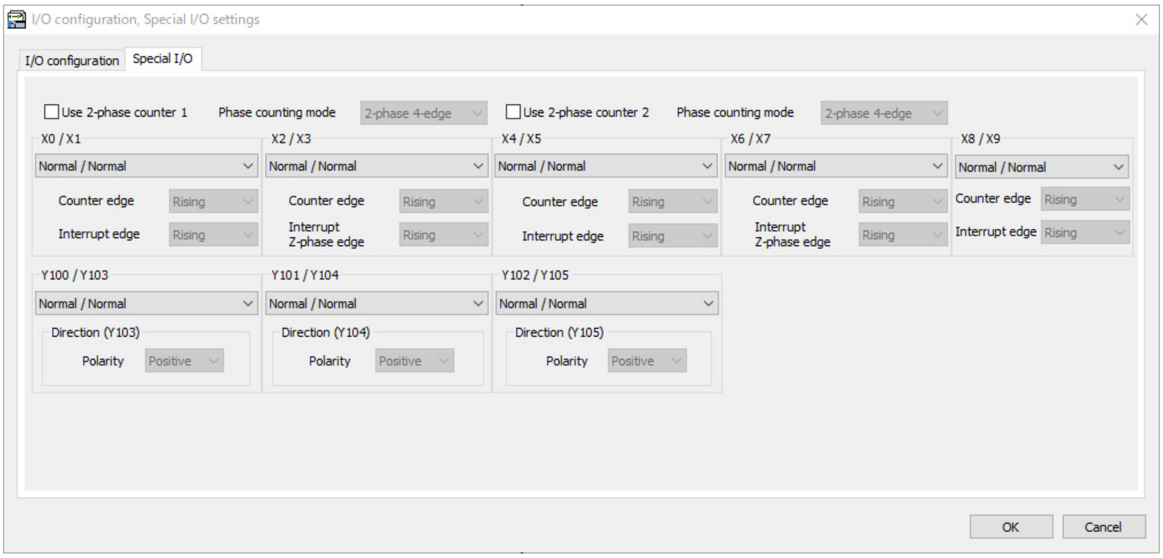


No	I/O types	Start No.	End No.	Size
1	WR	0	1FFF	8192
2	WR	6000	6FFF	4096
3	WM	780	7FF	128
4	TD	0	511	512
5	R	200	7BF	1472
6				
7				
8				
9				
10				

Figure 4.2 Retentive area setting

(4) Special I/O settings

Configure the settings when using the high-speed counter function and pulse output function for input and output terminals in the basic unit. A function can be assigned each to input and output terminals.



☐ Use 2-phase counter 1 Phase counting mode 2-phase 4-edge

☐ Use 2-phase counter 2 Phase counting mode 2-phase 4-edge

X0 / X1: Normal / Normal; Counter edge: Rising; Interrupt edge: Rising

X2 / X3: Normal / Normal; Counter edge: Rising; Interrupt Z-phase edge: Rising

X4 / X5: Normal / Normal; Counter edge: Rising; Interrupt edge: Rising

X6 / X7: Normal / Normal; Counter edge: Rising; Interrupt Z-phase edge: Rising

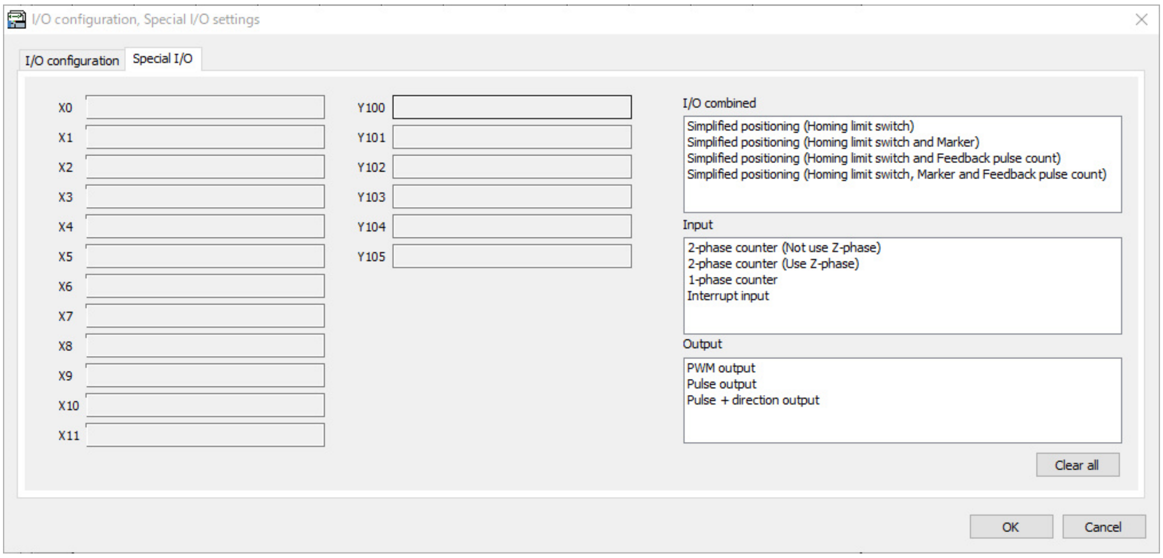
X8 / X9: Normal / Normal; Counter edge: Rising; Interrupt edge: Rising

Y100 / Y103: Normal / Normal; Direction (Y103) Polarity: Positive

Y101 / Y104: Normal / Normal; Direction (Y104) Polarity: Positive

Y102 / Y105: Normal / Normal; Direction (Y105) Polarity: Positive

Figure 4.3 I/O assignment and special I/O settings (Ver. 4.21 or earlier)



I/O combined
- Simplified positioning (Homing limit switch)
- Simplified positioning (Homing limit switch and Marker)
- Simplified positioning (Homing limit switch and Feedback pulse count)
- Simplified positioning (Homing limit switch, Marker and Feedback pulse count)

Input
- 2-phase counter (Not use Z-phase)
- 2-phase counter (Use Z-phase)
- 1-phase counter
- Interrupt input

Output
- PWM output
- Pulse output
- Pulse + direction output

Figure 4.4 I/O assignment and special I/O settings (Ver. 5.00 or later)

For details on how to set the special I/Os, see the User's Manual.

4.3 Description of basic program

Create a program in combinations of specific functional instructions. Instructions can be categorized into six types according to the description of executed processing.

Table 4.2 Instruction types

No.	Type	Description
1	Basic instruction	This instruction manipulates bit data and word data using ladder program-specific symbols. Timer and counters are also included in the basic instructions.
2	Arithmetic instruction (Written in a processing box)	This instruction performs a substitution, four arithmetic operations, logical operation, and data comparison.
3	Application instruction (Written in a processing box)	This instruction groups processes that cannot be completed without combining multiple basic instructions and arithmetic instructions.
4	Control instruction (Written in a processing box)	This instruction defines the end of the program and changes the program execution order, such as jump, repeat, and subroutine.
5	CPU communication instruction (Written in a processing box)	This instruction controls the CPU serial communication port and Ethernet communication port. It is used to establish data communication with an external device.

(1) Basic circuit configuration

The smallest unit of user program is "instruction". Write an instruction to connect between the left and right lines called power rails. This is called a "circuit". A circuit has a rule that conditions must be written on the left side, and an output (coil) or processing (processing box) must be written on the right end.

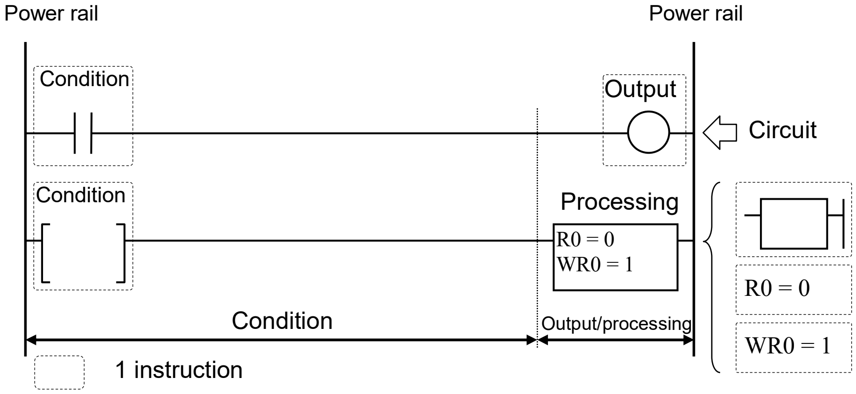


Figure 4.5 User program basic circuit

When all conditions on the left side of the circuit are met, the coil turns ON, and the instruction in the processing box is executed. If conditions are not met, the coil turns OFF, and the instruction in the processing box is not executed.

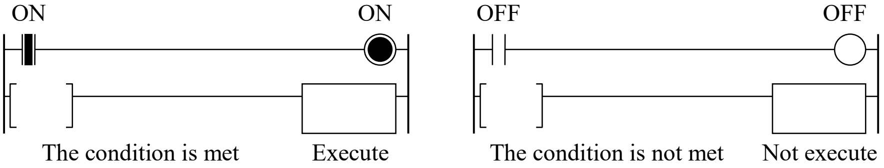


Figure 4.6 Basic circuit operation

Reference

When conditions are not met, a coil executes a process to turn itself OFF. Unlike a coil, a processing box executes nothing when conditions are not met.

(2) Instruction execution order

A program is executed from left to right and from top to bottom sequentially, and when all conditions are met in each circuit, an output turns ON. Otherwise, it turns OFF. Similarly, processing is executed when all conditions are met or not executed when they are not met.

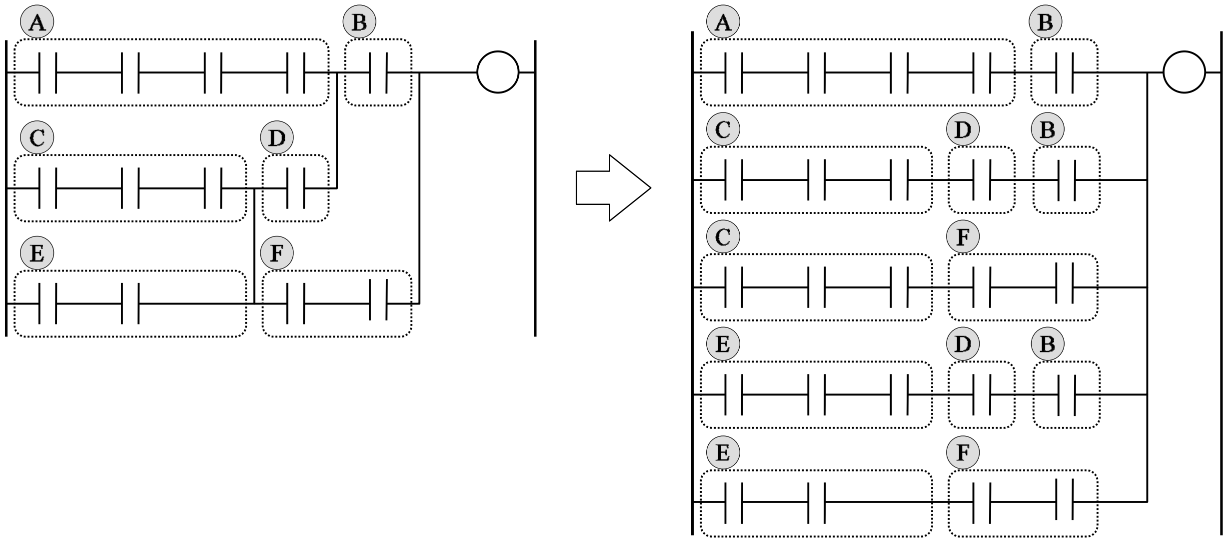


Figure 4.7 Concept of parallel (OR) connection circuit

If you break down the left circuit shown in Figure 4.7 by block and follow the from-left-to-right and from-top-to-bottom flow rules, the circuit will be as shown on the right.

Reference

You cannot place a contact or comparison box to the right of a coil or processing box.

A route of A ➔ D ➔ F ➔ coil shown in Figure 4.7 is called a " sneak circuit", but it is invalid as it is against the from-left-to-right flow rule.

(3) Description range of each circuit

As shown in the figure below, each circuit can contain 11 contacts and 32 coils. By using return symbols, you can write a circuit with 321 contacts and one coil within 32 lines.

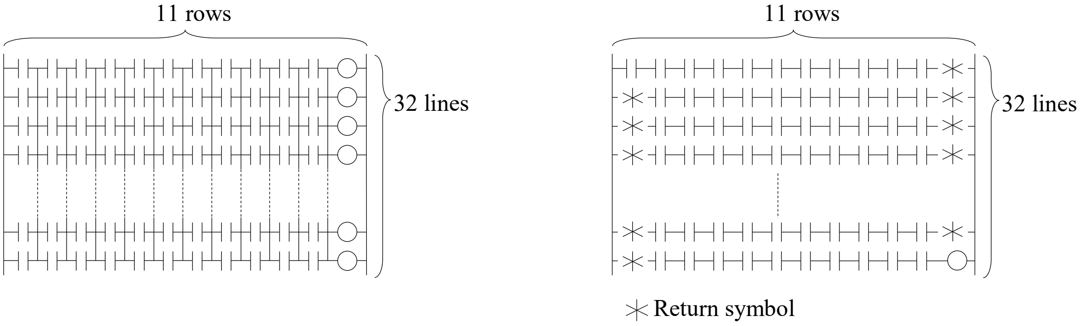


Figure 4.8 Description range of each circuit

You can write one comparison box using the width of three contacts. A comparison box can be considered a-contact, which turns ON when conditions in the box are met.

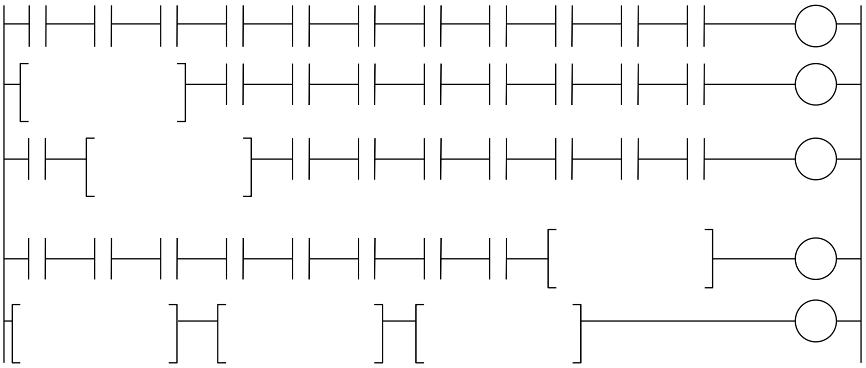


Figure 4.9 Description of comparison box

A processing box uses the width of two contacts and one coil. Write any other instruction than the basic instruction in a processing box. Each processing box contains up to 32 instructions.

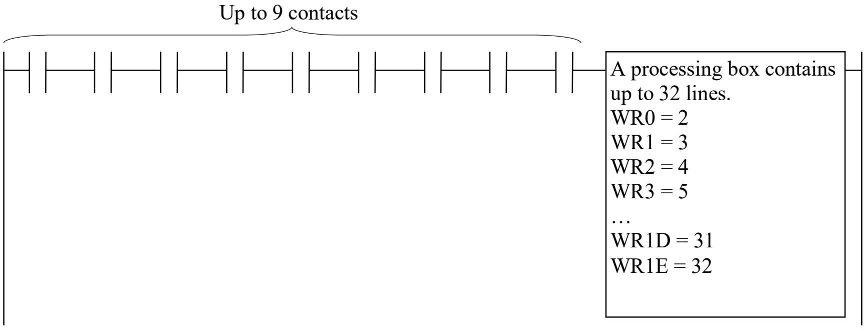


Figure 4.10 Description of processing box

In the MICRO-EHV, a processing box and a coil can be connected in parallel (OR).

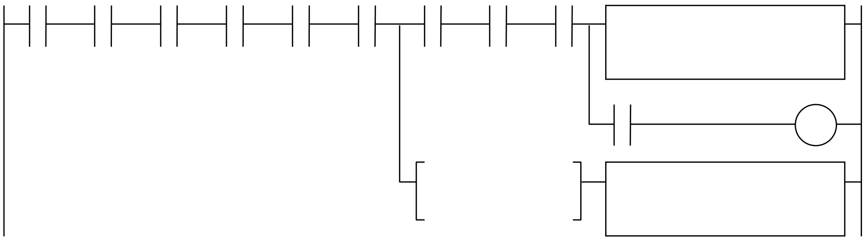
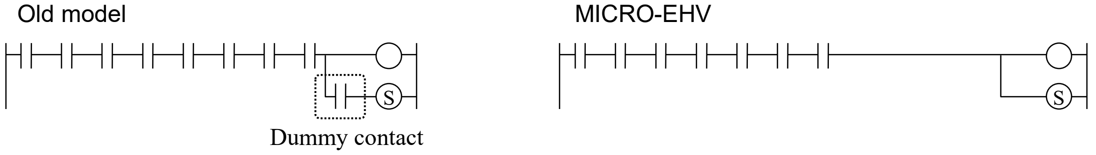


Figure 4.11 Parallel (OR) connection of processing box and coil

Reference

In models earlier than the EHV series (models using the LADDER EDITOR for programming), a dummy contact must be placed before the coil when a set or reset coil is connected in parallel, but this is no longer necessary in the MICRO-EHV.



4.4 Instruction on creating user program

(1) Timer

- **Elapsed value update**

The elapsed timer value is updated when a timer instruction is executed. Therefore, the timer may not turn ON correctly under a condition where the timer instruction is not scanned in a program using the JMP instruction or master control (MCS).

(If the non-scanning time of the timer instruction exceeds the time base × 65,535, the timer does not turn ON correctly.)

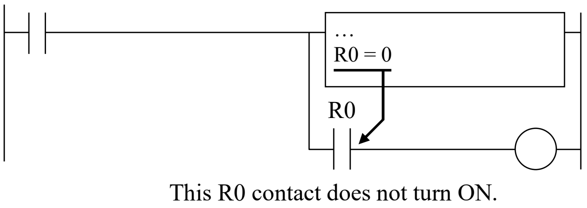
A value before the timer instruction is executed is retained as the elapsed timer value.

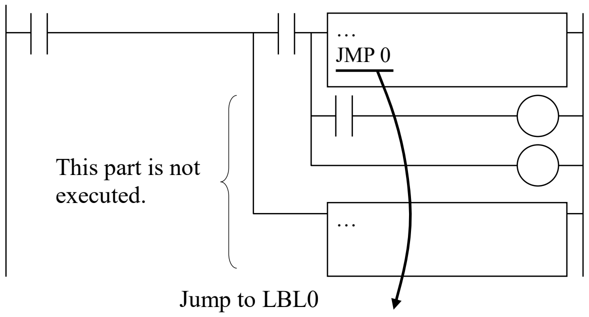
- **Timer start condition**

You cannot connect a timer instruction directly from the power rail. A condition is required before a timer instruction.

(2) Parallel (OR) connection of coil and processing box

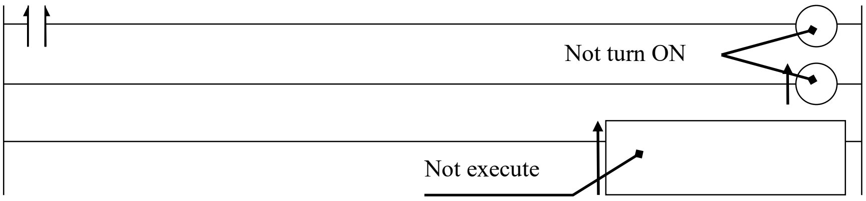
If you connect a coil and a processing box in parallel (OR), the operation of the bottom part of the parallel (OR) connection may not be executed depending on the operation of the top part because the MICRO-EHV scans from top to bottom.





(3) Edge, edge coil, and edge processing box

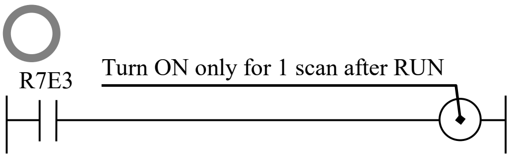
An edge, edge coil, or edge processing box instruction (rising/falling) detects the state change of a condition to the left of an edge instruction. Therefore, a condition is required to the left of an edge instruction.

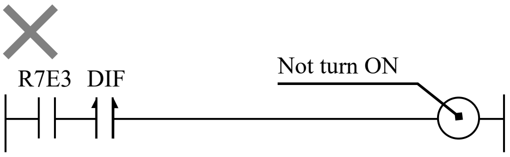


Caution

The special internal output (R7E3), which turns ON one scan after RUN, starts turned ON when the PLC starts running. (It only changes from ON to OFF.) Therefore, R7E3 cannot be used as a rising edge condition.

No edge instruction is required for R7E3 because it turns ON only one scan after RUN.

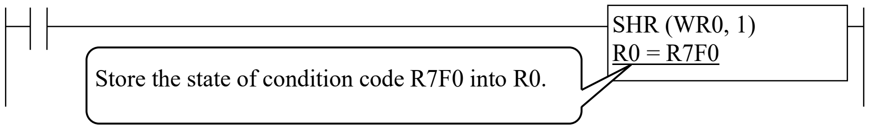




(4) Condition code

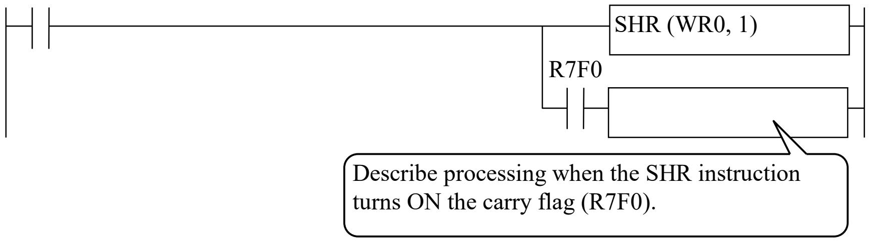
Most of instructions use a bit internal output called a condition code. A condition code is shared by instructions, so the state may change after an instruction is executed. For this reason, use the following program when accessing the condition code:

- Store the code into another internal output after the instruction is executed.



Store the state of condition code R7F0 into R0.

- Branch the output with the same condition and insert a condition code contact before the processing after the branch.



Describe processing when the SHR instruction turns ON the carry flag (R7F0).

(5) Floating point

There is a limit to the number of significant floating-point digits. Therefore, a difference occurs between the calculation result and the true value.

In a program where the floating-point data type is used and a calculation result is compared to a constant (particularly, comparison using "= = (match)" or "< > (unmatch)"), you may not obtain the expected result due to an error.

To compare a floating-point calculation result, it is recommended that the comparison should be conducted based on a range instead of match or unmatch.

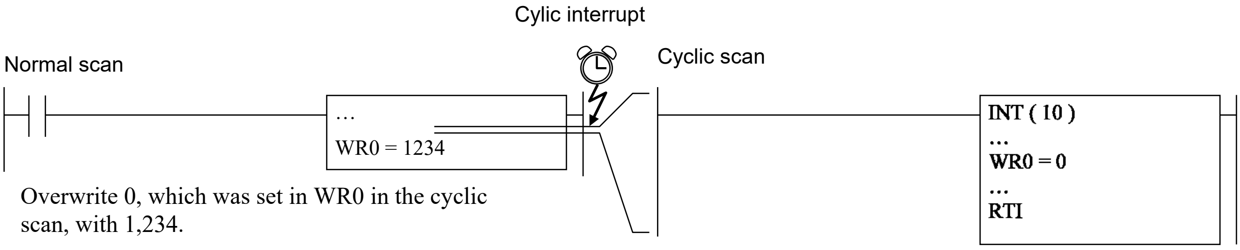
Reference

The table below shows the three types of floating-point calculation errors.

Name	Description
Round-off error	This error occurs because low-order digits are discarded by rounding up, down, or off to show the calculation result within the significant digits. Example: When decimal 0.1 is converted to binary, it will be a recurring decimal. The value will be close to 0.1 within the limited significant digits but never be 0.1.
Loss of significance	This error occurs because the smaller number is not reflected in the calculation result after addition or subtraction of two numbers that differ significantly in absolute magnitude. Example: When 0.0056 is added to 1,234, the expected result is 1,234.0056. However, the mantissa value with the smaller exponent is rounded off because the calculation is conducted based on the value with the larger exponent.
Cancellation	This error occurs because some of the significant digits are lost when calculating a difference between two very nearly equal absolute values. Example: When 1.23789 is subtracted from 1.23456, the result will be -0.00333, which represents three significant digits, although it is six before calculation.

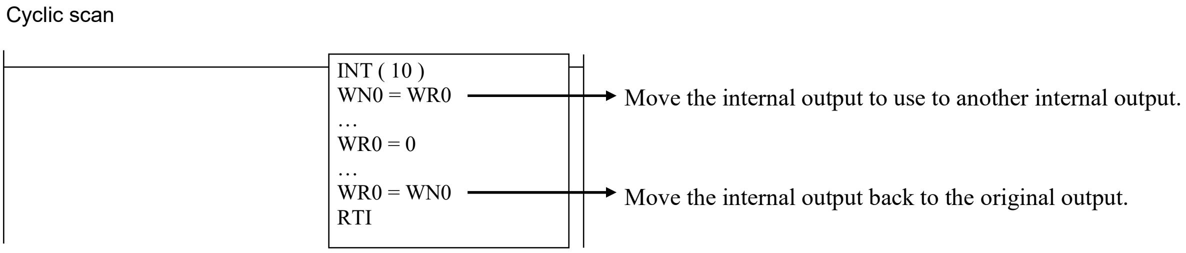
(6) Cyclic scan

If the same I/O is manipulated in scans with different priorities in a program using a cyclic scan and a normal scan or multiple cyclic scans, a value set in the scan with higher priority may be lost.



The following shows programming examples to prevent data loss:

- Do not use the same I/O in scans with different priorities.
- Only access an I/O set in a scan with higher priority in other scans.
- Move an I/O used at the beginning of a scan with higher priority to another I/O temporarily and move it back to the original I/O after the scan.

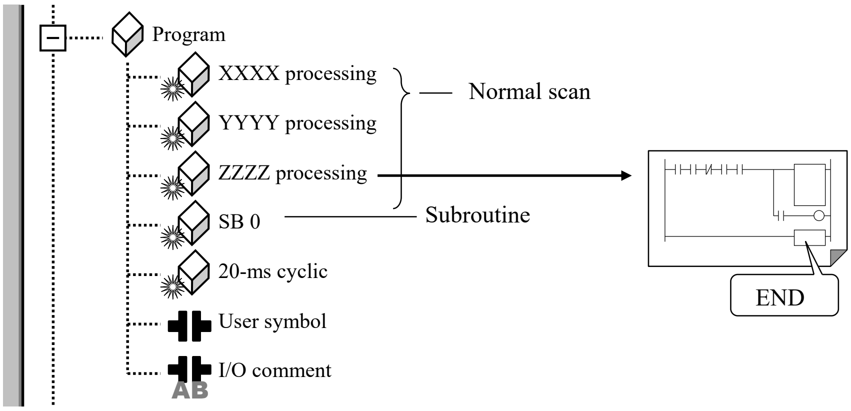


External inputs/outputs are refreshed at the end of the normal scan. Before accessing the latest input information in a cyclic scan or manipulating outputs, use the I/O refresh instruction.

(7) Sheet separation

When a subroutine or cyclic scan is used, the END instruction is required at the end of the normal scan.

You can write a subroutine and cyclic scan in separate sheets, but make sure that the END instruction is placed at the end of the normal scan sheet in the project tree because the MICRO-EHV executes sheets registered to the project tree in order from top to bottom.



Chapter 5 Command Specifications

5.1 Command classification

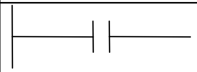
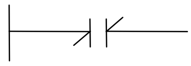
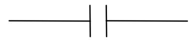
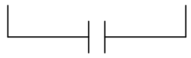
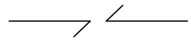
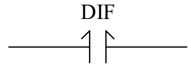
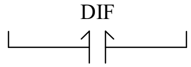
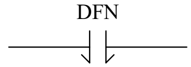
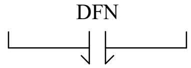
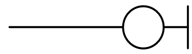
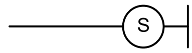
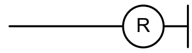
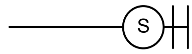
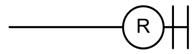
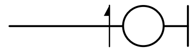
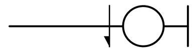
Usable commands in MICRO-EHV can be classified as follows.

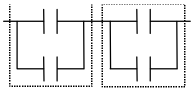
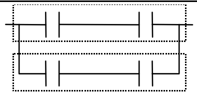
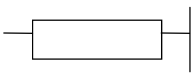
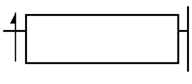
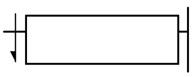
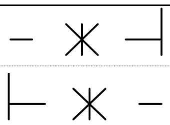
Table 5.1 Command classification table

No.	Command classification	Description	Types
1	Basic commands	Sequence	26
		Timer / Counter	11
		Comparison box	18
2	Arithmetic commands	Substitution expression (Array variable)	3
		Four arithmetical operations	16
		Logical operations	3
		Comparison expression	18
		Type conversion, Code	8
		Square root, Exponentiation	3
		Trigonometric function	12
		Exponent, Logarithm	3
3	Application commands	Command support	1
		Bit operation	4
		Shift / Rotate	14
		Character conversion	14
		Data operation	11
		Data search	3
		Exchange	2
		Transfer	4
		Decode, Encode	2
		Information memory / Indication	2
		I/O refresh	3
		PID control	3
		FIFO	3
		Communication support	2
		Others	3
		Special I/O	17
4	Control commands	END, JMP, CAL, FOR, NEXT, RTS, RTI, LBL, SB, INT, CEND, CJMP, XINT, XRTI, CINTP, CRTIP, CINTN, CRTIN	18
5	CPU communication commands	TRNS0, RECV0, MBMST, INV1,OMST1,OCTP1	6

5.2 Command list

(1) Basic commands (Sequence commands)

No.	Ladder symbol	Command name	Processing	Page
1		Logical operation start	Indicates the start of the contact A operation.	5-26
2		NOT operation start	Indicates the start of the contact B operation.	5-26
3		AND operation	Indicates the series connection of A contact.	5-27
4		NAND operation	Indicates the series connection of B contact.	5-27
5		OR operation	Indicates the parallel connection of A contact.	5-28
6		NOR operation	Indicates the parallel connection of B contact.	5-28
7		NOT operation	Inverts the operation result.	5-29
8	 	Rising edge detection	Indicates the rising detection of input.	5-30
9	 	Falling edge detection	Indicates the falling detection of input.	5-31
10		I/O output	Indicates the output coil.	5-32
11		I/O set	Indicates the set coil.	5-33
12		I/O reset	Indicates the reset coil.	5-33
13		Master control start	Indicates the set action of master control.	5-34
14		Master control cancellation	Indicates the reset action of master control.	5-34
15		Coil with edge (Rising)	Detects the rising of condition, and turns ON the output during only one scan.	5-35
16		Coil with edge (Falling)	Detects the falling of condition, and turns ON the output during only one scan.	5-36

No.	Ladder symbol	Command name	Processing	Page
17		Series connection of logical block	Indicates the series connection of two logical blocks.	5-37
18		Parallel connection of logical block	Indicates the parallel connection of two logical blocks.	5-38
19		Start and End of processing box	Indicates the start and the end of the processing box.	5-39
20		Start and End of processing box with edge (Rising)	Indicate the start and the end of the processing box. (Only a scan that a condition before the processing box turned ON is executed.)	5-40
21		Start and End of processing box with edge (Falling)	Indicates the start and the end of the processing box. (Only a scan that a condition before the processing box turned OFF is executed.)	5-41
22		Start and End of comparison box	Indicates the start and the end of the comparison box.	5-42
23		Turning symbol	Uses to divide a line when connecting 12 points of contact or more.	5-43

(2) Basic commands (Timer / Counter)

No.	Ladder symbol	Command name	Processing	Page
1	—○┤TD	On delay timer	Indicates acting of the on-delay timer.	5-45
2	—○┤TDN	Off delay timer	Indicates acting of the off-delay timer.	5-47
3	—○┤SS	Single shot	Indicates acting of the single shot.	5-49
4	—○┤MS	Mono stable timer	Indicates acting of the mono stable timer.	5-51
5	—○┤TMR	Integral timer	Indicates acting of the integral timer.	5-53
6	—○┤WDT	Watchdog timer	Indicates acting of the watchdog timer.	5-55
7	—○┤CU	Counter	Indicates acting of the counter.	5-57
8	—○┤RCU	Ring counter	Indicates acting of the ring counter.	5-59
9	—○┤CTU	Up down counter up	Indicates acting of the up down counter up.	5-61
10	—○┤CTD	Up down counter down	Indicates acting of the up down counter down.	5-61
11	—○┤CL	Counter clear	Indicates the clear acting of CU, RCU, CTU, CTD, and WDT.	5-64

Reference

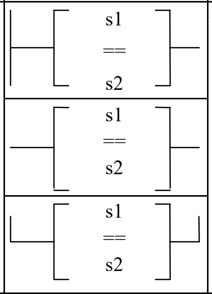
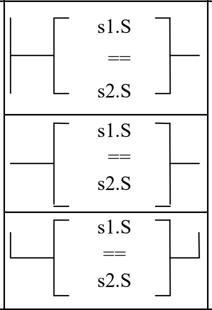
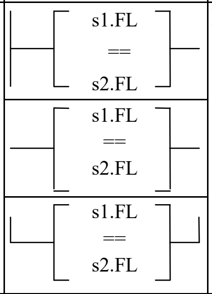
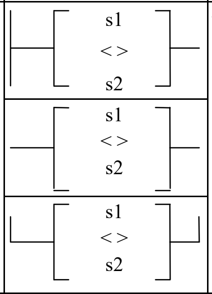
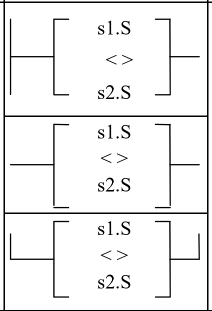
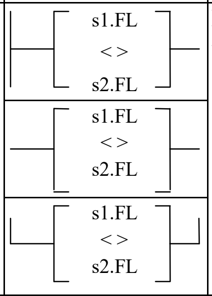
(1) Timer

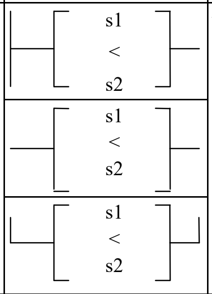
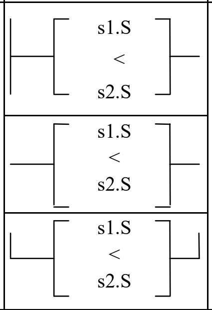
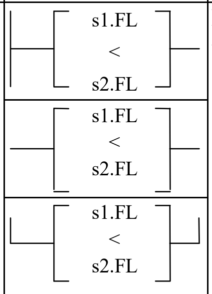
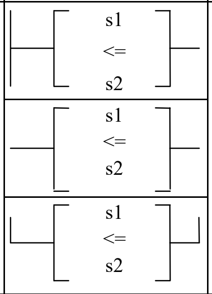
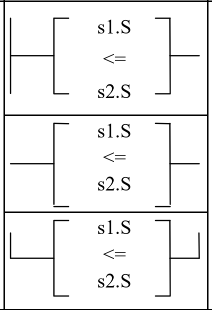
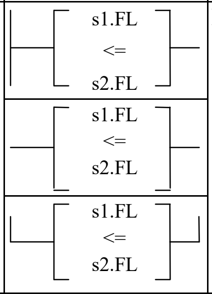
The total point of the timer is 2,048 points. The timer base can be selected from 1s, 100 ms, 10 ms, and 1ms. (There is no point limit on the timer base.)

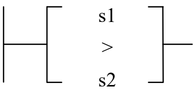
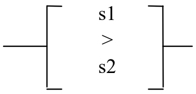
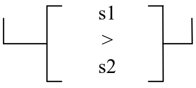
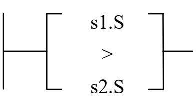
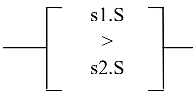
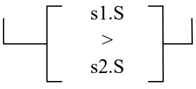
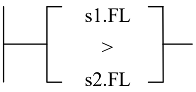
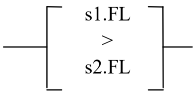
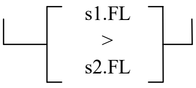
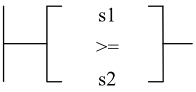
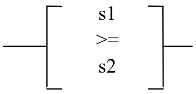
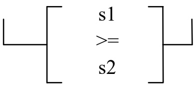
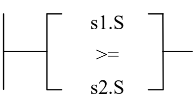
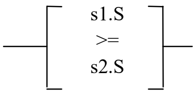
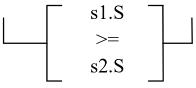
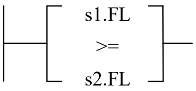
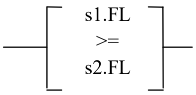
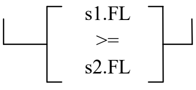
(2) Counter

The total point of the counter is 2,048 points. The same number cannot be used in the timer counter.

(3) Basic commands (Comparison box)

No.	Ladder symbol	Command name	Processing	Page
1		= Comparison box	s1 = s2 : Continuity s1 ≠ s2 : Discontinuity	5-65
2		Signed = Comparison box	s1 = s2 : Continuity s1 ≠ s2 : Discontinuity Compare s1 and s2 in signed 32-bit binary or signed 16-bit binary.	5-66
3		Floating point = Comparison box	s1 = s2 : Continuity s1 ≠ s2 : Discontinuity Compare s1 and s2 in floating point.	5-67
4		< > Comparison box	s1 = s2 : Discontinuity s1 ≠ s2 : Continuity	5-69
5		Signed < > Comparison box	s1 = s2 : Discontinuity s1 ≠ s2 : Continuity Compare s1 and s2 in signed 32-bit binary or signed 16-bit binary.	5-70
6		Floating point < > Comparison box	s1 = s2 : Discontinuity s1 ≠s2 : Continuity Compare s1 and s2 in floating point.	5-71

No.	Ladder symbol	Command name	Processing	Page
7		< Comparison box	s1 < s2 : Continuity s1 ≥ s2 : Discontinuity	5-73
8		Signed < Comparison box	s1 < s2 : Continuity s1 ≥ s2 : Discontinuity Compares s1 and s2 in signed 32-bit binary or signed 16-bit binary.	5-74
9		Floating point < Comparison box	s1 < s2 : Continuity s1 ≥ s2 : Discontinuity Compares s1and s2 in floating point.	5-75
10		≤ Comparison box	s1 ≤ s2 : Continuity s1 > s2 : Discontinuity	5-76
11		Signed ≤ Comparison box	s1 ≤ s2 : Continuity s1 > s2 : Discontinuity Compares s1 and s2 in signed 32-bit binary or signed 16-bit binary.	5-77
12		Floating point ≤ Comparison box	s1 ≤ s2 : Continuity s1 > s2 : Discontinuity Compares s1 and s2 in floating point.	5-78

No.	Ladder symbol	Command name	Processing	Page
13	  	> Comparison box	s1 > s2 : Continuity s1 ≤ s2 : Discontinuity	5-79
14	  	Signed > Comparison box	s1 > s2 : Continuity s1 ≤ s2 : Discontinuity Compares s1 and s2 in signed 32-bit binary or signed 16-bit binary.	5-80
15	  	Floating point > Comparison box	s1 > s2 : Continuity s1 ≤ s2 : Discontinuity Compares s1 and s2 in floating point.	5-81
16	  	≥ Comparison box	s1 ≥ s2 : Continuity s1 < s2 : Discontinuity	5-82
17	  	Signed ≥ Comparison box	s1 ≥ s2 : Continuity s1 < s2 : Discontinuity Compares s1 and s2 in signed 32-bit binary or signed 16-bit binary.	5-83
18	  	Floating point ≥ Comparison box	s1 ≥ s2 : Continuity s1 < s2 : Discontinuity Compares s1 and s2 in floating point.	5-84

(4) Arithmetic commands

No.	Ladder symbol	Command name	Processing	Page
1	d = s	Substitution	d ← s	5-86
	d = s.m2	Substitution (Bit🡐Bit cut down)	d ← s (the m2th bit) d is the bit I/O.	
	d.m1 = s	Substitution (Bit cut down 🡐Bit)	d (the m1th bit) ← s s is the bit I/O.	
	d.m1 = s.m2	Substitution (Bit cut down 🡐 Bit cut down)	d (the m1th bit) ← s (the m2 th bit)	
2	d.S = s.S	Substitution (Signed integer)	d.S ← s.S	5-89
3	d.FL = s.FL	Substitution (Floating point)	d.FL ← s.FL d.FL and s.FL are double word.	5-91
4	d = s1 + s2	Binary addition	d ← s1＋s2	5-93
5	d.S = s1.S + s2.S	Binary addition (Signed)	d.S ← s1.S＋s2.S	5-95
6	d = s1 B + s2	BCD addition	d ← s1＋s2	5-97
7	d.FL = s1.FL + s2.FL	Binary addition (Floating point)	d.FL ← s1.FL＋s2.FL [FUN 105(s)]	5-98
8	d = s1 - s2	Binary subtraction	d ← s1-s2	5-99
9	d.S = s1.S – s2.S	Binary subtraction (Signed)	d.S ← s1.S-s2.S	5-101
10	d = s1 B- s2	BCD subtraction	d ← s1-s2	5-103
11	d.FL = s1.FL – s2.FL	Binary subtraction (Floating point)	d.FL ← s1.FL-s2.FL [FUN 106(s)]	5-104
12	d=s1 × s2	Binary multiplication	d ← s1×s2	5-105
13	d.S = s1.S × s2.S	Signed binary multiplication	d.S ← s1.S×s2.S	5-106
14	d=s1 B × s2	BCD multiplication	d ← s1×s2	5-108
15	d.FL = s1.FL × s2.FL	Binary multiplication (Floating point)	d.FL ← s1.FL×s2.FL [FUN 107(s)]	5-110
16	d = s1 / s2	Binary division	[Word] d ← s1 / s2 WRF016 ← s1 mod s2 [Double word] d ← s1 / s2 DRF016 ← s1 mod s2 * Floating point operation has no remains. [Floating decimal point is FUN 107(s)]	5-112
17	d.S = s1.S / s2.S	Signed binary division		5-114
18	d = s1 B/ s2	BCD division		5-116
19	d.FL = s1.FL / s2.FL	Binary division (Floating point)		5-118

[ ] : Ladder symbol in MICRO-EHV series

No.	Ladder symbol	Command name	Processing	Page
20	d = s1 OR s2	Logical disjunction	d ← s1 + s2	5-120
	d = s1.m1 OR s2	Logical disjunction (Bit cut down)	d ← s1 (the m1th bit) + s2	5-120
	d = s1 OR s2.m2	Logical disjunction (Bit cut down)	d ← s1 + s2 (the m2th bit)	
	d = s1.m1 OR s2.m2	Logical disjunction (Bit cut down)	d ← s1 (the m1th bit) + s2 (the m2th bit)	
	d.m0 = s1 OR s2	Logical disjunction (Bit cut down)	d (the m0th bit) ← s1 + s2	5-120
	d.m0 = s1.m1 OR s2	Logical disjunction (Bit cut down)	d (the m0th bit) ← s1 (the m1th bit) + s2	
	d.m0 = s1 OR s2.m2	Logical disjunction (Bit cut down)	d (the m0th bit) ← s1 + s2 (the m2th bit)	
	d.m0 = s1.m1 OR s2.m2	Logical disjunction (Bit cut down)	d (the m0th bit) ← s1 (the m1th bit) + s2 (the m2th bit)	
21	d = s1 AND s2	Logical conjunction	d ← s1 ・ s2	5-122
	d = s1.m1 AND s2	Logical conjunction (Bit cut down)	d ← s1 (the m1th bit) ・ s2	5-122
	d = s1 AND s2.m2	Logical conjunction (Bit cut down)	d ← s1 ・ s2 (the m-th bit)	
	d = s1.m1 AND s2.m2	Logical conjunction (Bit cut down)	d ← s1 (the m1th bit) ・ s2 (the m-th bit)	
	d.m0 = s1 AND s2	Logical conjunction (Bit cut down)	d (the m0th bit) ← s1 ・ s2	5-122
	d.m0 = s1.m1 AND s2	Logical conjunction (Bit cut down)	d (m0 bit) ← s1 (the m1th bit) ・ s2	
	d.m0 = s1 AND s2.m2	Logical conjunction (Bit cut down)	d (the m0th bit) ← s1 ・ s2 (the m-th bit)	
	d.m0 = s1.m1 AND s2.m2	Logical conjunction (Bit cut down)	d (the m0th bit) ← s1 (the m1th bit) ・ s2 (the m-th bit)	
22	d = s1 XOR s2	Exclusive OR	d ← s1 ⊕ s2	5-124
	d = s1.m1 XOR s2	Exclusive OR (Bit cut down)	d ← s1 (the m1th bit) ⊕ s2	5-124
	d = s1 XOR s2.m2	Exclusive OR (Bit cut down)	d ← s1 ⊕ s2 (the m-th bit)	
	d = s1.m1 XOR s2.m2	Exclusive OR (Bit cut down)	d ← s1 (the m1th bit) ⊕ s2 (the m-th bit)	
	d.m0 = s1 XOR s2	Exclusive OR (Bit cut down)	d (the m0th bit) ← s1 ⊕ s2	5-124
	d.m0 = s1.m1 XOR s2	Exclusive OR (Bit cut down)	d (the m0th bit) ← s1 (the m1th bit) ⊕ s2	
	d.m0 = s1 XOR s2.m2	Exclusive OR (Bit cut down)	d (the m0th bit) ← s1 ⊕ s2 (the m-th bit)	
	d.m0 = s1.m1 XOR s2.m2	Exclusive OR (Bit cut down)	d (the m0th bit) ← s1 (the m1th bit) ⊕ s2 (the m-th bit)	

No.	Ladder symbol	Command name	Processing	Page
23	d = s1 == s2	= Comparison	When s1 = s2, d ← 1 / When s1 ≠ s2, d ← 0	5-126
	d.m = s1 == s2	= Comparison	When s1 = s2, d.m ← 1 / When s1 ≠ s2, d.m ← 0	
24	d = s1.S == s2.S	Signed = Comparison	When s1.S = s.S2, d ← 1 / When s1.S ≠ s2.S, d ← 0 Compares s1 and s2 in signed 16-bit binary or signed 32-bit binary.	5-128
	d.m = s1.S == s2.S	Signed = Comparison	When s1.S = s2.S, d.m ← 1 / When s1.S ≠ s2.S, d.m ← 0 Compares s1 and s2 in signed 16-bit binary or signed 32-bit binary.	
25	d = s1.FL == s2.FL	Floating point = Comparison	When s1.FL = s2.FL, d ← 1 / When s1.FL ≠ s2.FL, d ← 0 Compares s1 and s2 in floating point (32 bit).	5-130
	d.m = s1.FL == s2.FL	Floating point = Comparison	When s1.FL = s2.FL, d.m ← 1 / When s1.FL ≠ s2.FL, d.m ← 0 Compares s1 and s2 in floating point (32 bit).	
26	d = s1 <> s2	<> Comparison	When s1 = s2, d ← 0 / When s1 ≠ s2, d ← 1	5-131
	d.m = s1 <> s2	<> Comparison	When s1 = s2, d.m ← 0 / When s1 ≠ s2, d.m ← 1	
27	d = s1.S <> s2.S	Signed <> Comparison	When s1.S = s2.S, d. ← 0 / When s1.S ≠ s2.S, d ← 1 Compares s1 and s2 in signed 16-bit binary or signed 32-bit binary.	5-133
	d.m = s1.S <> s2.S	Signed <> Comparison	When s1.S = s2.S, d.m ← 0 / When s1.S ≠ s2.S, d.m ← 1 Compares s1 and s2 in signed 16-bit binary or signed 32-bit binary.	
28	d = s1.FL <> s2.FL	Floating point <> Comparison	When s1.FL = s2.FL, d ← 0 / When s1.FL ≠ s2.FL, d ← 1 Compares s1 and s2 in floating point (32 bit).	5-135
	d.m = s1.FL <> s2.FL	Floating point <> Comparison	When s1.FL = s2.FL, d.m ← 0 / When s1.FL ≠ s2.FL, d.m ← 1 Compare s1 and s2 in floating point (32 bit).	
29	d = s1 < s2	< Comparison	When s1 < s2, d ← 1 / When s1 ≥ s2, d ← 0	5-136
	d.m = s1 < s2	< Comparison	When s1 < s2, d.m ← 1 / When s1 ≥ s2 d.m ← 0	
30	d = s1.S < s2.S	Signed < Comparison	When s1.S< s2.S, d ← 1 / When s1.S ≥ s2.S, d ← 0 Compares s1 and s2 in signed 16-bit binary or signed 32-bit binary.	5-138
	d.m = s1.S < s2.S	Signed < Comparison	When s1.S < s2.S, d.m ← 1 / When s1.S ≥ s2.S, d.m ← 0 Compares s1 and s2 in signed 16-bit binary or signed 32-bit binary.	
31	d = s1.FL < s2.FL	Floating point < Comparison	When s1.FL < s2.FL, d ← 1 / When s1.FL ≥ s2.FL, d ← 0 Compares s1 and s2 in floating point (32 bit).	5-140
	d.m = s1.FL < s2.FL	Floating point < Comparison	When s1.FL < s2.FL, d.m ← 1 / When s1.FL ≥ s2.FL, d.m ← 0 Compares s1 and s2 in floating point (32 bit).	

No.	Ladder symbol	Command name	Processing	Page
32	d = s1 <= s2	<= Comparison	When s1 ≤ s2, d ← 1 / When s1 > s2, d ← 0	5-141
	d.m = s1 <= s2	<= Comparison	When s1 ≤ s2, d.m ← 1 / When s1 > s2, d.m ← 0	
33	d = s1.S <= s2.S	Signed <= Comparison	When s1.S ≤ s2.S, d ← 1 / When s1.S > s2.S, d ← 0 Compares s1 ands2 in signed 16-bit binary or signed 32-bit binary.	5-143
	d.m = s1.S <= s2.S	Signed <= Comparison	When s1.S ≤ s2.S, d.m ← 1 / when s1.S > s2.S, d.m ← 0 Compares s1 and s2 in signed 16-bit binary or signed 32-bit binary.	
34	d = s1.FL <= s2.FL	Floating point <= Comparison	When s1.FL ≤ s2.FL, d ← 1 / When s1.FL > s2.FL, d ← 0 Compares s1 and s2 in floating point (32 bit).	5-145
	d.m = s1.FL <= s2.FL	Floating point <= Comparison	When s1.FL ≤ s2.FL, d.m ← 1 / When s1.FL > s2.FL, d.m ← 0 Compares s1 and s2 in floating point (32 bit).	
35	d = s1 > s2	> Comparison	When s1 > s2, d ← 1 / When s1 ≤ s2, d ← 0	5-146
	d.m = s1 > s2	> Comparison	When s1 > s2, d.m ← 1 / When s1 ≤ s2, d.m ← 0	
36	d = s1.S > s2.S	Signed > Comparison	When s1.S > s2.S, d ← 1 / When s1.S ≤ s2.S, d ← 0 Compares s1 and s2 in signed 16-bit binary or signed 32-bit binary.	5-148
	d.m = s1.S > s2.S	Signed > Comparison	When s1.S > s2.S, d.m ← 1 / When s1.S ≤ s2.S, d.m ← 0 Compares s1 and s2 in signed 16-bit binary or signed 32-bit binary.	
37	d = s1.FL > s2.FL	Floating point > Comparison	When s1.FL > s2.FL, d ← 1 / When s1.FL ≤ s2.FL, d ← 0 Compares s1 and s2 in floating point (32 bit).	5-150
	d.m = s1.FL > s2.FL	Floating point > Comparison	When s1.FL > s2.FL, d.m ← 1 / When s1.FL ≤ s2.FL, d.m ← 0 Compares s1 and s2 in floating point (32 bit).	
38	d = s1 >= s2	>= Comparison	When s1 ≥ s2, d ← 1 / When s1 < s2, d ← 0	5-151
	d.m = s1 >= s2	>= Comparison	When s1 ≥ s2, d.m ← 1 / When s1 < s2, d.m ← 0	
39	d = s1.S >= s2.S	Signed >= Comparison	When s1.S ≥ s2.S, d ← 1 / When s1.S < s2.S, d ← 0 Compares s1 and s2 in signed 16-bit binary or signed 32-bit binary.	5-153
	d.m = s1.S >= s2.S	Signed >= Comparison	When s1.S ≥ s2.S, d.m ← 1 / When s1.S < s2.S, d.m ← 0 Compares s1 and s2 in signed 16-bit binary or signed 32-bit binary.	
40	d = s1.FL >= s2.FL	Floating point >= Comparison	When s1.FL ≥ s2.FL, d ← 1 / When s1.FL < s2.FL, d ← 0 Compares s1 and s2 in floating point (32 bit).	5-155
	d.m = s1.FL >= s2.FL	Floating point >= Comparison	When s1.FL ≥ s2.FL, d.m ← 1 / When s1.FL < s2.FL, d.m ← 0 Compares s1 and s2 in floating point (32 bit).	

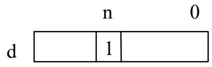
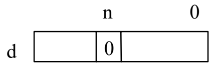
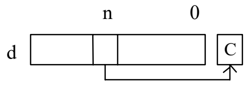
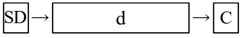
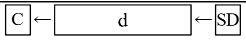
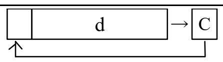
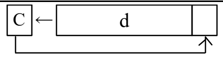
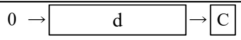
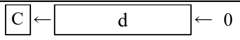
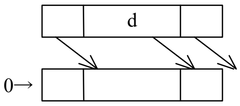
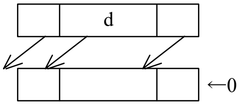
No.	Ladder symbol	Command name	Processing	Page
41	d.S = INTG (s.FL)	Floating point conversion (Floating point → Signed)	Converts the floating point into the signed. [FUN 100(s) / FUN 101(s)]	5-156
42	d.FL = FLOAT (s.S)	Floating point conversion (Signed → Floating point)	Converts the signed into the floating point. FUN 102(s) / FUN 103(s)]	5-158
43	d.FL = RAD (s.FL)	Radian conversion (Floating point)	Converts degree into radian. [FUN 108(s)]	5-160
44	d.FL = DEG (s.FL)	Degree conversion (Floating point)	Converts radian into degree. [FUN 109(s)]	5-162
45	d = ABS (s.S)	Absolute value	Stores the absolute value of s in d and stores the signed value of s in carry (R7F0). (0:Positive, 1:Negative) [ABS (d, s)]	5-164
46	d.S = SGET (s)	Sign addition	If the value of carry (R7F0) is 0, stores the value of s in d. If it is 1, stores the complementary value of 2 of s. [SGET(d, s)]	5-166
47	d.S = EXT (s.S, n)	Bit extension	Copies the value of the signed bit in s (the n-th bit) for from the n-th bit to MSB in d, and stores the value of s in the lower word in d. [EXT(d, s)]	5-168
48	d= NEG (s)	Two's complement	Stores the complementary of 2 of I/O No. s in d. [NEG(d)]	5-170
49	d = SQR (s) d.FL = SQR (s.FL)	Binary square root	Calculates the square root of 32-bit binary value. [FUN 60 (s) / FUN 116 (s)]	5-172
50	d = BSQR (s)	BCD square root	Calculates the square root of the value of s (8 digits BCD), and stores the result in d (4 digits BCD). [SQR(d, s)]	5-174
51	d = POW (s, n) d.FL = POW (s.FL, n.FL)	Exponentiation	Calculates the exponentiation.	5-176
52	d = SIN (s)	Trigonometric function SIN operation (Degree)	Stores the result after calculating SIN of the value indicated by s in d and d+1. [FUN 10 (s)]	5-178
53	d.FL = SINR (s.FL)	Trigonometric function SIN operation (Radian)	Calculates SIN of radian unit system in floating point. [FUN 110(s)]	5-180
54	d = COS (s)	Trigonometric function COS operation (Degree)	Stores the result after calculating COS of the value indicated by s in d and d+1. [FUN 11 (s)]	5-182
55	d.FL = COSR (s.FL)	Trigonometric function COS operation (Radian)	Calculates COS of radian unit system in floating point. [FUN 111(s)]	5-184
56	d = TAN (s)	Trigonometric function TAN operation (Degree)	Stores the result after calculating TAN of the value indicated by s in d and d+1. [FUN 12 (s)]	5-186
57	d.FL = TANR (s.FL)	Trigonometric function TAN operation (Radian)	Calculates TAN of radian unit system in floating point. [FUN 112(s)]	5-188
58	d = ASIN (s)	Trigonometric function ARC SIN operation (Degree)	Stores the result after calculating ARC SIN of the value indicated by s (decimal part) and s+1(integer part) in d. [FUN 13 (s)]	5-190
59	d.FL = ASINR (s.FL)	Trigonometric function ARC SIN operation (Radian)	Calculates ARC SIN of radian unit system in floating point. [FUN 113(s)]	5-192

[ ] : Ladder symbol in MICRO-EH series

No.	Ladder symbol	Command name	Processing	Page
60	d = ACOS (s)	Trigonometric function ARC COS operation (Degree)	Stores the result after calculating ARC COS of the value indicated by s (decimal part) and s+1 (integer part) in d. [FUN 14 (s)]	5-194
61	d.FL = ACOSR (s.FL)	Trigonometric function ARC COS operation (Radian)	Calculates ARC COS of radian unit system in floating point. [FUN 114(s)]	5-196
62	d = ATAN (s)	Trigonometric function ARC TAN operation (Degree)	Stores the result after calculating ARC TAN of the value indicated s (decimal part) and s+1 (integer part) in d. [FUN 15 (s)]	5-198
63	d.FL = ATANR (s.FL)	Trigonometric function ARC TAN operation (Radian)	Calculates ARC TAN of radian unit system in floating point. [FUN 115(s)]	5-200
64	d.FL = EXP (s.FL)	Exponential operation (Floating point)	Performs the exponential operation. [FUN 117(s)]	5-202
65	d.FL = LOG (s.FL)	Natural logarithm (Floating point)	Performs the logarithmic operation as the natural logarithm is the base.	5-204
66	d.FL = LOG10 (s.FL)	Common logarithm (Floating point)	Performs the logarithmic operation as the common logarithm is the base.	5-206

[ ] : Ladder symbol in MICRO-EH series

(5) Application commands

No.	Ladder symbol	Command name	Processing	Page
1	d = ADR (s)	Coding I/O address	Store a real address of I/O specified by s in d. [ADRIO (d, s)] * d requires 2 words in EHV.	5-210
2	BSET (d, n)	Bit set	 Sets the bit n to 1.	5-213
3	BRES (d, n)	Bit reset	 Resets the bit n to 0.	5-214
4	BTS (d, n)	Bit test	 Gets the value of the bit n in C(R7F0).	5-215
5	BCU (d, s)	Bit count	Stores the number of bits set to 1 in the content of s (word and double word) in I/O No. d.	5-217
6	SHR (d, n)	Shift to the right	 n bits shift the right	5-218
7	SHL (d, n)	Shift to the left	 n bits shift to the left.	5-220
8	ROR (d, n)	Rotate to the right	 n bits rotate to the right.	5-222
9	ROL (d, n)	Rotate to the left	 n bits rotate to the left.	5-224
10	LSR (d, n)	Logic shift to the right	 n bits shift to the right.	5-227
11	LSL (d, n)	Logic shift to the left	 n bits shift to the left.	5-228
12	BSR (d, n)	BCD shift to the right	 BCD n digits shift to the right.	5-230
13	BSL (d, n)	BCD shift to the left	 BCD n digits shift to the left.	5-232
14	WSHR (d, n)	Batch shift to the right	n bits (or n words) width from I/O No. d is shifted 1 bit (or 1 word) to the right.	5-234
15	WSHL (d, n)	Batch shift to the left	n bits (or n words) width from I/O No. d is shifted 1 bit (or 1 word) to the left.	5-236
16	WBSR (d, n)	Batch BCD shift to the right	BCD of n digits width from I/O No. d is shifted 1 digit to the right.	5-238
17	WBSL (d, n)	Batch BCD shift to the left	BCD of n digits width from I/O No. d is shifted 1 digit to the left.	5-240
18	BSHR (d, n)	Byte units shift to the right	The data string specified is shifted the number of bytes specified (8 bits *n) to the right. [FUN 48(s)]	5-242
19	BSHL (d, n)	Byte units shift to the left	The data string specified is shifted the number of bytes specified (8 bits *n) to the left. [FUN 49(s)]	5-244

[ ] : Ladder symbol in MICRO-EH series

No.	Ladder symbol	Command name	Processing	Page
20	BCD (d, s)	Binary → BCD conversion	Stores the result after converting the value of s into BCD in I/O No. d. If the value of s is abnormal, DER(R7F4) = 1.	5-246
21	BIN (d, s)	BCD → Binary conversion	Stores the result after converting the value of s into Binary in I/O No. d. If the value of s is abnormal, DER(R7F4) = 1.	5-248
22	GRY (d, s) (*1)	Binary → Gray code conversion	Stores the result after converting the value (Binary) of s into Gray code in I/O No. d.	5-250
23	GBIN (d, s) (*1)	Gray code → Binary conversion	Stores the result after converting the value (Gray code) of s into Binary in I/O No. d.	5-251
24	BINDA (d, s)	BIN (16 bits) → ASCII conversion	Stores after converting the 16-bit unsigned BIN data into decimal ASCII code. [FUN 30 (s)]	5-252
25	SBINDA (d, s.S)	BIN (signed 32bits) → ASCII conversion	Stores after converting the 32-bit signed BIN data into decimal ASCII code. [FUN 31 (s)]	5-254
26	BINHA (d, s)	BIN → ASCII conversion (16 bits/32 bits)	Stores after converting the 16-bit unsigned BIN data into decimal ASCII code. [FUN 32(s) / FUN 33(s)]	5-256
27	BCDDA (d, s)	BIN → ASCII conversion (16sbits/32bits)	Stores after converting 16/32-bit BCD (BCD 4/8-digit) data into decimal ASCII code. [FUN 34(s) / FUN 35(s)]	5-258
28	DABIN (d, s)	ASCII → BIN conversion (16 bits)	Stores after converting the 5-digit unsigned decimal ASCII data into the hexadecimal BIN data. [FUN 36 (s)]	5-260
29	SDABIN (d.S, s)	ASCII → BIN conversion (Signed 32 bits)	Stores after converting the10-digit signed decimal ASCII code into the 32-bit BIN data. [FUN 37 (s)]	5-262
30	HABIN (d, s)	ASCII → BIN conversion (16 bits/32 bits)	Stores after converting 4/8-digit hexadecimal ASCII code into 16/32-bit BIN data. [FUN 38 (s) / FUN 39(s)]	5-264
31	DABCD (d, s)	ASCII → BIN conversion (16 bits/32 bits)	Stores after converting 4/8-digit ASCII code into 4/8-digit BCD data. [FUN 40 (s) / FUN 41(s)]	5-266
32	ASC (d, s, n)	BIN → ASCII conversion (Specifying)	Stores after converting BIN data into ASCII code of the number of characters specified. [FUN 42 (s)]	5-268
33	HEX (d, s, n)	ASCII → BIN conversion (Specifying)	Stores after converting ASCII code of the number of characters specified into BIN data. [FUN 43 (s)]	5-270
34	WTOB (d, s, n)	Word → Byte conversion	Divides 16-bit word data and stores after converting into 8-bit byte data. [FUN 46(s)]	5-272
35	BTOW (d, s, n)	Byte → Word conversion	Divides 8-bit word data and stores after converting into 16-bit byte data. [FUN 47(s)]	5-274
36	NOT (d, s)	Invert	Stores the result after inverting bit of the value of the I/O No. s in d.	5-276
37	UNIT (d, s, n)	Unite	Stores the value of lower 4 bits of n words in s in every 4 bits from the lower in d (word).	5-278
38	DIST (d, s, n)	Distribute	Takes value of every 4 bits from the lower in s (word), and sets the taken value to every 1 word consisting of lower 4 bits in from the I/O No. d (word). The upper bits become 0.	5-280
39	SADD (d, s1, s2)	Character string unite	Unites the character string specifies (- NULL), and stores the result as d is the top address. [FUN 44 (s)]	5-282
40	SCMP (d, s1, s2)	Character string comparison	Stores the result after comparing between character strings specified (- NULL) in d. [FUN 45 (s)]	5-284

[ ] : Ladder symbol in MICRO-EH series

*1 CPU: Supported by Ver.x120 or newer, C/E: Supported by Ver.5.00 or newer

No.	Ladder symbol	Command name	Processing	Page
41	BITTOW (d, s, n)	Expand Bit data into Word data	Sets the number of bits from the I/O No. specified to the word I/O No. specified. [FUN 127(s)]	5-286
42	WTOBIT (d, s, n)	Expand Word data into Bit data	Sets the number of bits from the I/O No. specified to the bit I/O No. specified. [FUN 128(s)]	5-288
43	INC (d)	Increment	Increases the word I/O or the double word I/O specified by 1. [FUN 123(s) / FUN 124(s)]	5-290
44	DEC (d)	Decrement	Decreases the word I/O or the double word I/O specified by 1. [FUN 125(s) / FUN 126(s)]	5-292
45	DSRCH (d, s1, s2, n)	Data search	Searches the specified data from the data string, and set the data position and the number of data. [FUN 20 (s)]	5-294
46	TSRCH (d, s, n1, n2)	Data table search	Stores the data table of the specified No. taken from the data string in the specified position. [FUN 21 (s)]	5-296
47	VSRCH (d, s1, s2, n)	Average / Minimum / Maximum search	Calculates the average, the minimum, and the maximum value of the data table specified. [FUN 63 (s)]	5-298
48	SWAP (d, s)	Swapping	Stores the result after swapping upper 8-bit and lower 8-bit of the value of the I/O No. s in d.	5-301
49	XCG (d1, d2, n)	Block exchange	Exchanges the n-bit (or n-word) areas from the I/O No.d1 and from the I/O No. d2 with each other.	5-303
50	MOV (d, s, n)	Block transfer	Transfers (copy) the n-bit (or n-word) width data from the I/O No. s to the n-bit (or n-word) width area from I/O No. d.	5-305
51	BMOV (d, s, n1, n2)	Bit block transfer	Considers lower n1-bit of the I/O No. s to be 1 block and transfers the n2-blocks from the I/O No. d.	5-307
52	COPY (d, s, n)	Copy	Copies the bit data (or the word data) of the I/O No. s to the n-bit (or n-word) area from the I/O No. d.	5-310
53	BCOPY (d, s, n1, n2)	Bit block copy	Considers lower n1-bit of the I/O No. s to be 1 block, and copies the same block of n2 blocks from I/O No. d.	5-312
54	DECO (d, s, n)	Decode	Decodes the value indicated by lower n bits of s, and set the bit corresponding to the decoded result of the bit string in the I/O No. d to 1.	5-314
55	ENCO (d, s, n)	Encode	Stores the result after encoding the bit position set to 1 in the bit string of 2 to the n-th power in the I/O No. s in the I/O No. d. If there are several '1's, encode upper of the bit position.	5-316
56	RECSET (s, n)	Recording data (Initial setting)	Performs the initial setting for the RECEXE (Recording data) command.	5-318
57	RECEXE (s, n)	Recording data (Execution)	Memorizes the time data at a time of execution of the specified data and the command in the internal	5-320
58	ALREF	I/O refresh (All points)	Refreshes all external input and output areas. [FUN 80 (s)]	5-323
59	IOREF (s)	I/O refresh (Specifying input / output / link)	Refreshes only input area, only output area, or only link area. [FUN 81 (s)]	5-324
60	SLREF (s)	I/O refresh (Optional slot)	Refresh I/O of the slot specified. [FUN 82 (s)]	5-326

[ ] : Ladder symbol in MICRO-EH series

No.	Ladder symbol	Command name	Processing	Page
61	PIDIT (s)	Initializing PID calculation	Initializes the area for the PID calculation. [FUN 0(s)]	5-330
62	PIDOP (s)	Execution control of PID calculation	Controls the execution of PID calculation. [FUN 1(s)]	5-331
63	PIDCL (s)	PID calculation	Performs PID calculation. [FUN 2(s)]	5-332
64	FIFIT (p, n)	FIFO initial	Stores the value of n in the size area (p) of FIFO, and stores 0 in the area (p+1) of the number of uses of FIFO.	5-345
65	FIFWR (p, s)	FIFO write	Stores the value of the I/O No. s in the write position of FIFO, and add 1 to the value of the area (p+1) of the number of uses of FIFO.	5-346
66	FIFRD (p, d)	FIFO read	Stores data taken from the read position of FIFO in d. Pack data in FIFO for 1 piece and subtracts 1 from the value of the area (p+1) of the number of uses of FIFO.	5-347
67	CCCL (s)	Creation of check code	Creates the check code to add the data frame in a general-purpose communication. [FUN22 (s)]	5-349
68	CCCMP (s)	Collation of check code	Compares the check code of the receiving frame in a general-purpose communication. [FUN23 (s)]	5-353
69	IFR (s)	Process stepping	Performs the process stepping processing. [FUN 4 (s)]	5-357
70	TMRNGE(d, s1, s2)	Time range decision	When the time data (hour and minute data) of PLC is s1 or more and less than s2, I/O specified by d turns ON.	5-360
71	UFNC (s1, s2)	User-defined function	Define user-specific functions according to the table specified in s2.	5-362
72	CUSTA (s)	Counter control	Controls the start / stop of the specified single-phase / 2-phase counter. [FUN 140 (s)]	5-367
74	CURD (s)	Current counter value read	Reads the current value of the specified single-phase / 2-phase counter. [FUN 144 (s)]	5-369
75	CUWR (s)	Current counter value write	Writes the current value of the specified single-phase / 2-phase counter. [FUN 143 (s)]	5-370
76	CUPRE (s)	Counter comparison value setting	Sets the first and second comparison values for the specified single-phase / two-phase counter. [FUN 146 (s)]	5-371
77	PWMSTA (s)	PWM output start / change	Starts the PWM output with the specified output frequency and ON-duty from the specified output number. When run during output, change the frequency and ON-Duedie. [FUN 148 (s)]	5-373
78	PWMSTP (s)	PWM output stop	Stops PWM output of the specified PWM output number. [FUN 147 (s)]	5-375
79	PLSTA (s)	Pulse output start	Outputs the specified number of pulse traines from the specified output number. It is also possible to specify acceleration and deceleration. [FUN 151 (s)]	5-376
80	PLSTAR (s) *	Pulse output start	Outputs the specified number of pulse traines from the specified output number. It is also possible to specify acceleration and deceleration. The acceleration / deceleration can be set 10 times that of PLSTA (s).	5-380
81	PLSPD (s)	Pulse speed control start	Accelerates at the specified acceleration rate from the specified output number and continues to output the pulse train with the specified frequency. [FUN 149 (s)]	5-382
82	PLSPDR (s) *	Pulse speed control start	Accelerates at the specified acceleration rate from the specified output number and continues to output the pulse train with the specified frequency. The acceleration / deceleration can be set 10 times that of PLSPD (s).	5-384
83	PLCNG (s)	Pulse speed change	Changes the output frequency of the specified output number.This command is valid only during pulse output by the PLSPD command. [FUN 150 (s)]	5-386

* CPU: Supported by Ver.x120 or later, C/E: Supported by Ver.5.00 or later

No.	Ladder symbol	Command name	Processing	Page
84	PLCNGR (s) ※	Pulse speed change	Changes the output frequency of the specified output number.This command is valid only during pulse output by the PLSPD command. The acceleration / deceleration can be set 10 times that of PLCNG (s).	5-388
82	PLSTP (s)	Pulse output stop	Stops pulse output for the specified output number [FUN 149 (s)]	5-390
85	PLSTPR (s) ※	Pulse output stop	Stops pulse output for the specified output number Deceleration of 10 times can be set for PLSTP (s).	5-392
86	PLHM (s) ※	Hominh return	Returns the specified output number to the home position. (Input must be used depending on the output number and the homing return method.)	5-394
87	PLSRD (s)	Pulse position data read	Reads the current value position data (absolute position) of the specified pulse output number.	5-396
88	PLSWR (s)	Pulse position data write	Rewrites the current value position data (absolute position) of the specified pulse output number to the specified value.	5-397

[ ] : Ladder symbol in MICRO-EH series

* CPU: Supported by Ver.x120 or later, C/E: Supported by Ver.5.00 or later

(6) Control commands

No.	Ladder symbol	Command name	Processing	Page
1	END	Normal scan END	Indicates the end of the normal scan and executes the normal scan from the top again.	5-400
2	CEND (s)	Conditional END of scan	When s = 1, executes the normal scan from the top again. When s = 0, executes the next command.	5-402
3	JMP n	Unconditional jump	Jumps to LBL n of the same No. n.	5-403
4	CJMP n (s)	Conditional jump	When s = 1, jumps to LBL n of the same No. n. When s = 0, executes the next command.	5-405
5	LBL n	Label	Indicates the destin, ration for JMP and CJMP of the same No. n to jump.	5-409
6	FOR n (s)	FOR	When s = 0, jumps to the next to NEXT n. When s ≠ 0, executes the next command.	5-411
7	NEXT n	NEXT	Jumps to FOR n after subtracting 1 from the value of s of FOR n of the same No. n.	5-412
8	CAL n	Subroutine call	Performs the subroutine SB n of the same No. n.	5-415
9	SB n	START subroutine	Indicates the start of the subroutine of No. n.	5-417
10	RTS	RETURN from subroutine	Returns from the subroutine.	5-419
11	INT (s)	START cyclic scan	Indicates the start of the interrupt scan/the cyclic scan of the cycle s [ms]. [INT 0, INT 1, INT 2, INT 3]	5-421
12	RTI	RETURN from cyclic scan	Return from the interrupt scan/ the cyclic scan.	5-423
13	XINT n	START interrupt scan	Indicates the start of the No.n input interrupt scan.	5-425
14	XRTI	RETURN from interrupt scan	Returns from the input interrupt scan.	5-427
15	CINTP n	START counter interrupt scan (1st comparison value)	Indicates the start of the No.n counter input 1 comparison value match interrupt scan.	5-429
16	CRTIP	RETURN from counter interrupt scan (1st comparison value)	Return from counter input 1st comparison value match interrupt scan.	5-431
17	CINTN n	START counter interrupt scan (2nd comparison value)	Indicates the start of the No.n counter input 2 comparison value match interrupt scan.	5-433
18	CRTIN	RETURN from counter interrupt scan (2nd comparison value)	Return from counter input 2nd comparison value match interrupt scan.	5-435

[ ] : Ladder symbol in MICRO-EH series

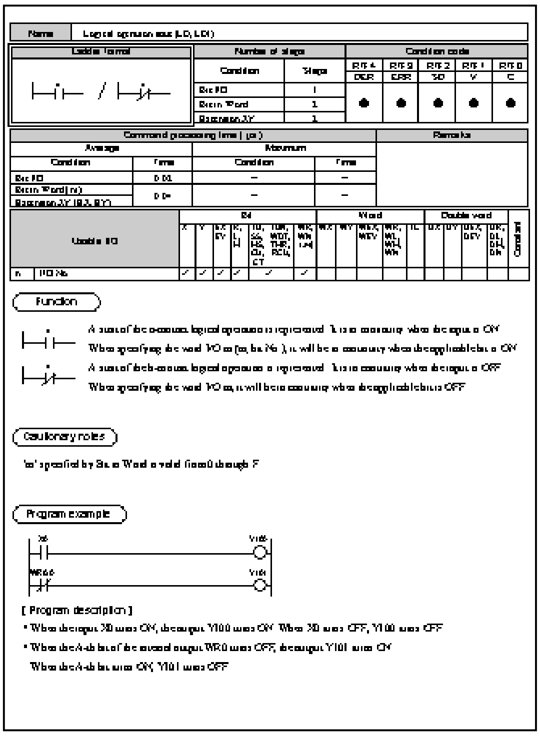
(7) CPU communication commands

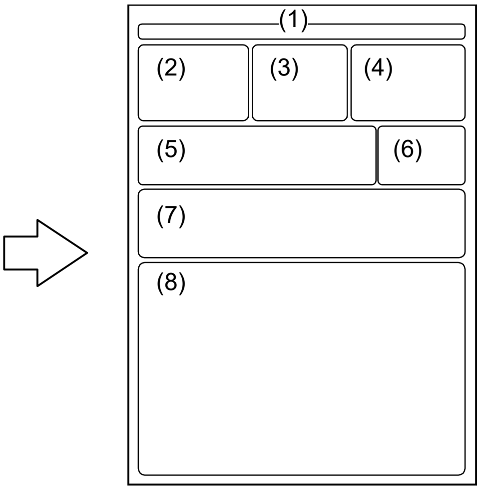
No.	Ladder symbol	Command name	Processing	Page
1	TRNS0 (s, t)	General-purpose port transmitting command	Transmits data from the CPU general-purpose port. [TRNS 0 (d, s, t)]	5-440
2	RECV0 (s, t)	General-purpose port receiving command	Receives data by the CPU general-purpose port. [RECV 0 (d, s, t)]	5-449
3	MBMST (s, t)	Modbus protocol query transmitting command	Performs serial communication with Modbus protocol by CPU serial port. [FUN 191 (s)]	5-452
4	INV1 (s)	Hitachi inverter control instruction 1	Controls the specified Hitachi inverter.	5-464
5	OMST1(s)	Oriental Motor stepping motor control instruction 1	Controls the specified Oriental Motor stepping motor.	5-470
6	OCTP1(s)	Omron temperature controller control instruction 1	Controls the specified Omron temperature controller.	5-491

[ ] : Ladder symbol in MICRO-EH series

5.3 About command specification details

The command specification details on each page are as follows.





(1) Name of command

The name of the command which is explained on the page is indicated.

(2) Ladder format

The format in case the command is written into the program is indicated. This format is written into the parameter part using symbols like d, s, and t. However, when writing this format into the program in practice, please replace these symbols with I/O and a constant you should use.

(3) Number of steps

The number of steps for the command is indicated. In case the number of steps changes according to conditions, the condition and the number of steps are indicated.

(4) Condition code

Condition code is bit to display the execution result of the command and the information attendant on the result. The meaning of each bit used as the condition code is as follows.

DER	Data error (Special internal output R7F4) When it exceeds the input and output number and when it is abnormal data as BCD, DER becomes "1" as data error. When it is not data error, DER becomes "0".
ERR	Error (Special internal output R7F3) When error occurs by executing the control command and the special command, it is set to "1" and the error code is set to WRF015. When there is no error, it remains unchanged.
SD	Shift data (Special internal output R7F2) The content of SD is shifted in on the SHR command and the SHL command.
V	Overflow (Special internal output R7F1) It means exceeding the range of signed data by overflow as a result of the operation of signed data.
C	Carry (Special internal output R7F0) It means the carry by addition, the borrow by subtraction, and the shift-out by shift.

[Meaning of symbol in table]

●	Holds the previous state.
[1]	Sets "1" when there is error in the operation result, and holds the previous state in other cases.
↕	It changes depending on the operation result.

(5) Processing speed of command

Command processing time of MICRO-EHV is indicated. There are commands of which processing time changes according to the parameter and the number of data.

(6) Remarks

Remarks about parameters used on the command and symbols written in the command processing time are described.

(7) Usable I/O on command

Usable I/O on parameters written with d, s, and t in columns of Ladder format and Command format is indicated. Usable I/O is marked "✓".

(8) Description

Processing of the command, explanation of the parameter, caution on use, and the sample program are indicated. And a method to convert the program for MICRO-EH to the program for MICRO-EHV is indicated according to commands.

Explanation of a headword for description

Function	The function (processing) of a command is explained.
Parameter	About a command to use several parameters, a meaning of the parameter and a set value are explained.
Cautionary notes	A matter you should care is written when using a command. * Be sure to read this before writing a command.
Program example	A sample program using the command and an action in executing the program are explained.
Return code	There is a command which represents the execution result using a 1-byte or 1-word code. This code is called a return code. When the command has a return code, the return code and the meaning are explained.
PRN ➔ PRJ	A method to convert the program for MICRO-EH and H series (Extension PRN) into the program for EHV is explained. Although it is possible to convert into the program for EHV using Convert Tool started from Control Editor, some commands cannot be converted into the program for EHV. Please modify the program referring to this part for these commands.

MEMO

[1] Basic commands

[2] Arithmetic commands

[3] Application commands

[4] Control commands

[5] CPU communication commands

Name	Logical operation start (LD, LDI)

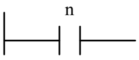
Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
	Bit I/O	1	●	●	●	●	●
	Bit in Word	2					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
Bit I/O	0.37	0.37	–	–	–
Bit in Word(.m)	0.44	0.45	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
n	I/O No.	✓	✓	✓	✓		✓								

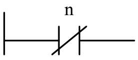
Remarks

Function



A start of the a-contact logical operation is represented. It is in continuity when the input is ON.

When specifying the word I/O.m (m: bit No.), it will be in continuity when the applicable bit is ON.



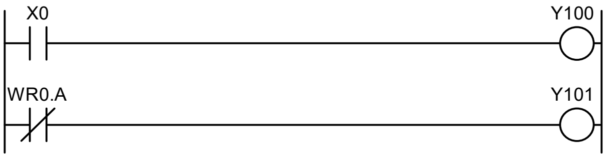
A start of the b-contact logical operation is represented. It is in contituity when the input is OFF.

When specifying the word I/O.m, it will be in contituity when the applicable bit is OFF.

Cautionary notes

‘m’ specified by Bit in Word is valid from 0 to F.

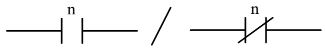
Program example



[Program description]

- When the input X0 turns ON, the output Y100 turns ON. When X0 turns OFF, Y100 turns OFF.
- When the A-th bit of the internal output WR0 turns OFF, the output Y101 turns ON.
  When the A-th bit turns ON, Y101 turns OFF.

Name	Series connection of contact (AND, ANI)

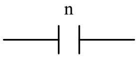
Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
	Bit I/O	1	●	●	●	●	●
	Bit in Word	2					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
Bit I/O	0.31	0.31	—	—	—
Bit in Word(.m)	0.45	0.45	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
n	I/O No.	✓	✓	✓	✓	✓								

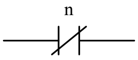
Remarks

Function



Performs the AND operation of the operation result preceding and the a-contact operation.

When specifying the word I/O.m (m: bit No.), performs the AND operation of the operation result preceding and the applicable bit (a-contact) in word.



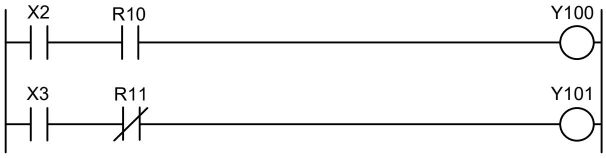
Performs the AND operation of the operation result preceding and the b-contact operation.

When specigying the word I/O.m, performes the AND operation of the operation result preceding and the applicable bit (b-contact) in word.

Cautionary notes

'm' specified by Bit in Word is valid from 0 to F.

Program example



[Program description]

- When both the input X2 and R10 are ON, the output Y100 turns ON. All other cases are turned OFF.
- When the input X3 is ON and R11 is OFF, the output Y101 turns ON. All other cases are turned OFF.

Name	Parallel connection of contact

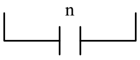
Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
	Bit I/O	2	●	●	●	●	●
	Bit in Word	3					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
Bit I/O	0.52	0.55	—	—	—
Bit in Word(.m)	0.69	0.69	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
n	I/O No.	✓	✓	✓	✓	✓								

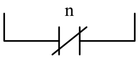
Remarks

Function



Performs the OR operation of the operation result preceding and the a-contact operation.

When specifying the word I/O.m (m: bit No.), performs the OR operation of the operation result preceding and the applicable bit (b-contact) in word.



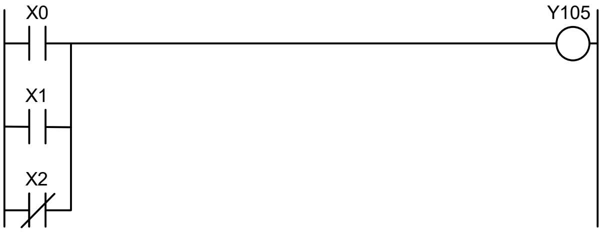
Performs the OR operation of the operation result preceding and the b-contact operation.

When specifying the word I/O.m, performs the OR operatino of the operation result preceding and the applicable bit (b-contact) in word.

Cautionary notes

'm' specified by Bit in Word is valid from 0 to F.

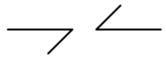
Program example



[Program description]

When X0 or X1 is ON, or when X2 is OFF, Y105 turns ON.

Name	Negation

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
	—	2	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
—	0.45	0.48	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
—	No argument														

Remarks

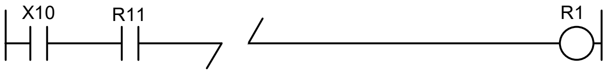
Function

The operation results obtained by then are inverted.

Cautionary notes

The negation command cannot be written into the top of the circuit.

Program example



[Program description]

When both the input X10 and X11 are ON, the operation becomes 1 but the operation becomes 0 because of the negation command. As a result, R1 is turned OFF. In all other cases, R1 is turned ON.

Name	Rising edge detection

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
	AND DIF	1	●	●	●	●	●
	OR DIF	2					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
AND DIF	0.37	0.37	—	—	—
OR DIF	0.37	0.37	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
—	No argument														

Remarks
Useable to 512 maximum.

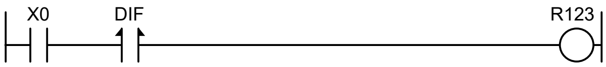
Function

The rising edge of the input signal is detected and the operation result for only one scan is held.

Cautionary notes

- A programming tool assigns the DIF number automatically.
- DIF cannot be used singly.
- DIF is a command to detect a change (0→1) of the operation result obtained by then.

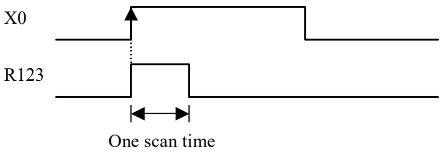
Program example



[Program description]

- R123 turns ON during only one scan at the rising of X0.

Time chart



- When X0 is the b-contact, the program is the same meaning as a-contact DFN operation of X0.

Name	Falling edge detection

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
	AND DFN	1	●	●	●	●	●
	OR DFN	2					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
AND DFN	0.37	0.38	—	—	—
OR DFN	0.37	0.40	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT, TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
—	No argument													

Remarks
Useable to 512 maximum.

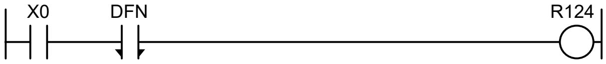
Function

The falling edge of the input signal is detected and the operation result for only one scan is held.

Cautionary notes

- A programming tool assigns the DIF number automatically.
- DFN cannot be used singly.
- DFN is a command to detect a change (1→0) of the operation result obtained by then.

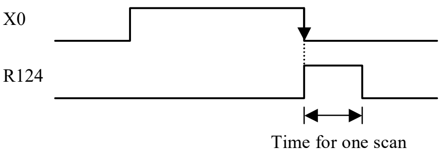
Program example



[Program description]

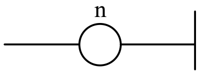
- R124 turns ON during only scan at the falling of X0.

Time chart



- When X0 is the b-contact, the program is the same meaning as a-contact DIF operation of X0.

Name	Output to Coil

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
 n	Bit I/O	1	●	●	●	●	●
	Bit in Word	2					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
Bit I/O	0.3	0.3	–	–	–
Bit in Word(.m)	0.43	0.51	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
n	I/O No.		✓	✓			✓								

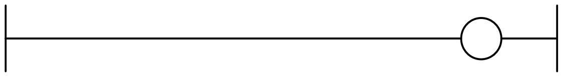
Remarks

Function

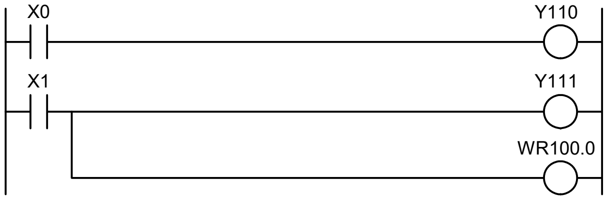
- When the operation result obtained by then is ‘1’, the coil is turned ON.
- When the operation result obtained by then is ‘0’, the coil is turned OFF.

Cautionary notes

- ‘m’ specified by bit in word is valid from 0 to F.
- In case of the circuit shown below, ‘1’ is added to the number of steps in the table mentioned above.



Program example



[Program description]

- When the input X0 is ON, the operation is set to ‘1’ and Y110 is turned ON.
- When the input X1 is ON, the operation is set to ‘1’ and Y111 is turned ON. Also the 0th bit of WR100 is set to ‘1’.

Name	Output Set and Reset to Coil

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
	Bit I/O	1	●	●	●	●	●
	Bit in Word	2					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
Bit I/O	0.32	0.32	—	—	—
Bit in Word(.m)	0.46	0.48	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
n	I/O No.		✓	✓			✓								

Remarks

Function

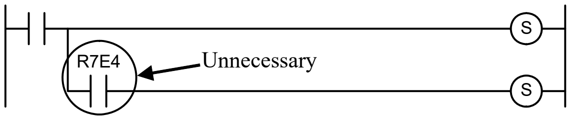
SET n When the operation result obtained by then is '1', the device is turned ON.

The device which is turned ON is not turned OFF even if the operation is set to '0'.

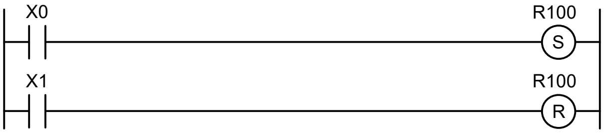
RES n When the operation result obtained by then is '1', the device is turned OFF.

Cautionary notes

- 'm' specified by bit in word is valid from 0 to F.
- Though the dummy contact is needed in front of the set reset coil connected as OR in MICRO-EH series, it is not needed in MICRO-EHV.



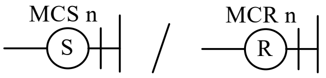
Program example



[Program description]

- When the input X0 turns ON, R100 turns ON. Even if X0 turns OFF, R100 remains unchanged from ON.
- When the input X1 turns ON, R100 turns OFF.
- If both inputs X0 and X1 turn ON, the later performance on the program has priority.

Name	Set and Reset of Master control

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
	MCS n	2	●	●	●	●	●
	MCR n	1					

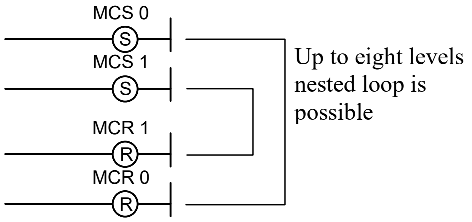
Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
MCS n	0.50	0.50	–	–	–
MCR n	0.18	0.18	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
n	No.														✓

Remarks
Useable No.0 to No.49 (decimal number)

Function

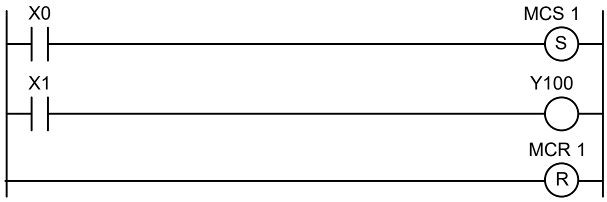
- The input of the circuit surrounded by Set (MCS n) and Reset (MCR n) of Master control is controlled. (Performs the AND operation with each input and MCS.)
- The master control can be used up to eight levels nested loop.



Cautionary notes

MCS and MCR of the master control should be always used together.

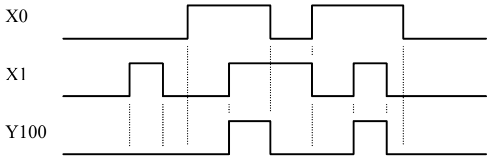
Program example



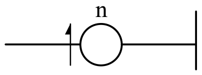
[Program description]

- When X0 is ON, Y100 is turned ON/OFF according to the state of X1.
- When X0 is OFF, Y100 is turned OFF regardless of the state of X1.

Time chart



Name	Coil with rising edge

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
	Bit I/O	2	●	●	●	●	●
	Bit in Word	3					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
Bit I/O	0.64	0.65	—	—	—
Bit in Word(.m)	0.78	0.79	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
n	I/O No.		✓	✓			✓								

Remarks
Useable to 1,024 maximum.

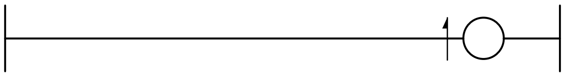
Function

When the operation result obtained by then changes from ‘0’ to ‘1’, the device is turned ON during one scan.

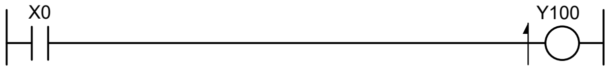
Cautionary notes

- ‘m’ specified by bit in word is valid from 0 to F.
- Unconditional circuit (the circuit of only a coil with edge) cannot be written.

[WRONG]

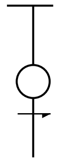


Program example

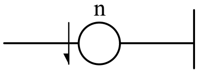


[Program description]

- Y100 turns ON during only one scan at the rising of ON of X0.
- If X0 is the b-contact, it is the same operation as the coil with a falling edge.



Name	Coil with falling edge

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
	Bit I/O	2	●	●	●	●	●
	Bit in Word	3					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
Bit I/O	0.65	0.65	—	—	—
Bit in Word(.m)	0.82	0.82	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
n	I/O No.		✓	✓		✓								

Remarks
Useable to 1,024 maximum.

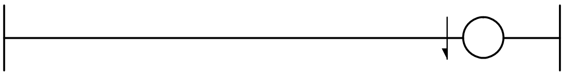
Function

When the operation result obtained by then changes from ‘1’ to ‘0’, the device is turned ON during one scan.

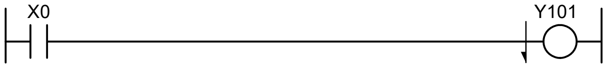
Cautionary notes

- ‘m’ specified by bit in word is valid from 0 to F.
- Unconditional circuit (the circuit of only a coil with edge) cannot be written.

[WRONG]



Program example



[Program description]

- Y101 turns ON during only one scan at the falling OFF of X0.
- If X0 is the b-contact, it is the same operation as the coil with a rising edge.

Name	Logical block series connection

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
(See function column)	—	0	●	●	●	●	●

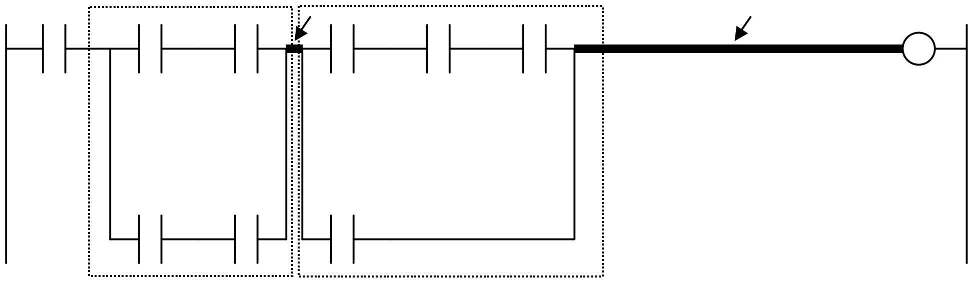
Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
—	—	—	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
—	No argument														

Remarks

Function

This command is used when the logical operation block is connected as AND.



Name	Logical block parallel connection

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
(See function column)	–	1	●	●	●	●	●

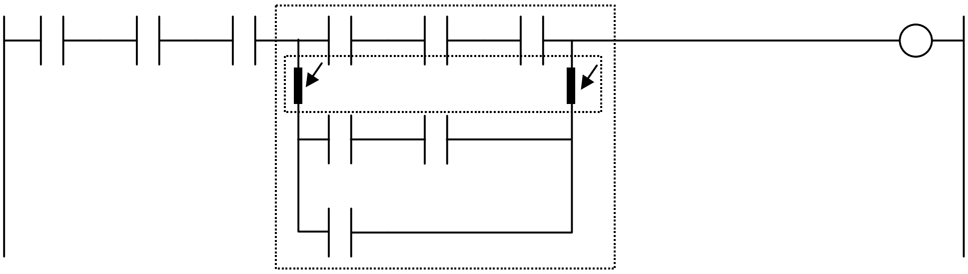
Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
–	–	–	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
–	No argument														

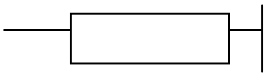
Remarks

Function

This command is used when the logical operation block is connected as OR.



Name	Start and End of processing box

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
	—	3	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
—	0.6	0.6	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
—	No argument													

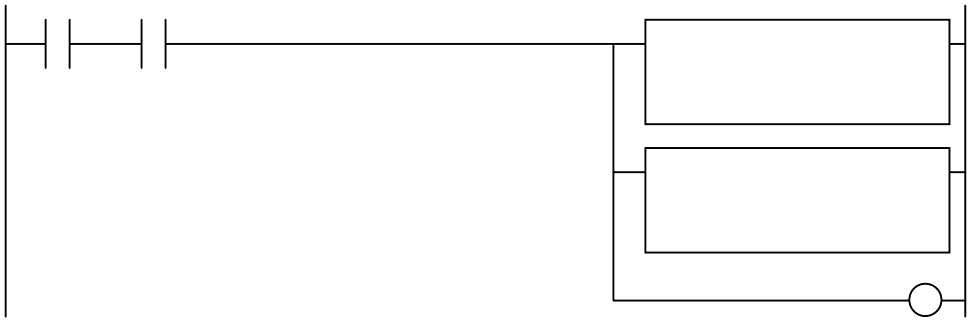
Remarks

Function

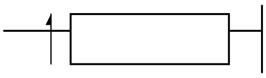
A start and an end of the processing box are represented.

Reference

- The content of the operation can be written up to 32 lines in the processing box.
- The processing box and the coil can be connected parallel.



Name	Start and End of processing box with rising edge

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
	—	2	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
—	0.6	0.6	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
—	No argument													

Remarks
Useable to 1,024 maximum.

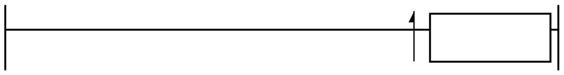
Function

- A start and an end of the processing box with the rising edge are represented.
- When the operation result obtained by then changes from ‘0’ to ‘1’, the operation in the processing box is performed.
- The processing box and the coil can be connected parallel.
- The content of the operation can be written up to 32 lines in the processing box.

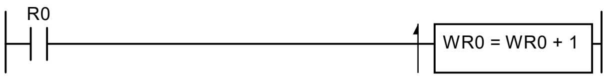
Cautionary notes

Unconditional circuit (the circuit of only a processing box with the rising edge) cannot be written.

[WRONG]

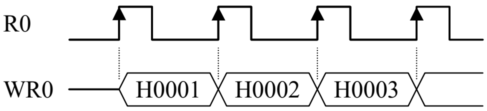


Program example

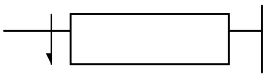


[Program description]

The operation in the processing box is performed only once at the rising of R0.



Name	Start and End of processing box with falling edge

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
	–	2	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
–	0.6	0.6	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
–	No argument	✓	✓	✓	✓	✓								

Remarks
Useable to 1,024 maximum.

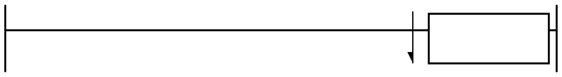
Function

- A start and an end of the processing with the falling edge are represented.
- When the operation result obtained by then changes from ‘1’ to ‘0’, the operation in the processing box is performed.
- The processing box and the coil can be connected parallel.
- The content of the operation can be written up to 32 lines in the processing box.

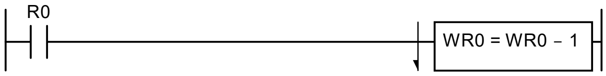
Cautionary notes

Unconditional circuit (the circuit of only the processing box with the falling edge) cannot be written.

[WRONG]

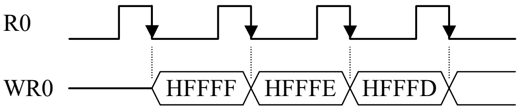


Program example



[Program description]

The operation in the processing box is performed only once at the falling of R0.



Name	Start and End of comparison box

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
	–	0	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
–	–	–	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
–	No argument													

Remarks

Function

A start and an end of the comparison box are represented.

Name	Turning symbol

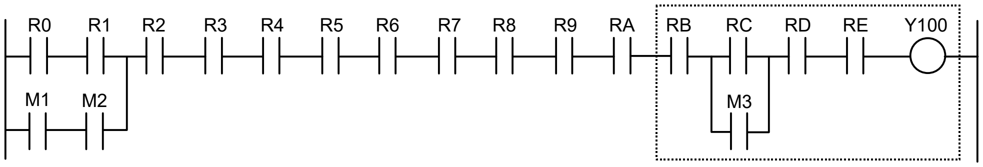
Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
－ ✕ ┤ / ├ ✕ －	—	1	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
—	0.15	0.31	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM
—	No argument													

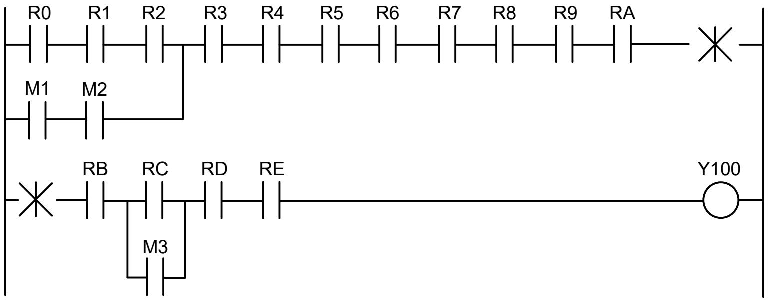
Remarks

Function



The circuit that is more than 12 contact mentioned above cannot be input as it is. In this case, the circuit can be turned back using the turning symbol.

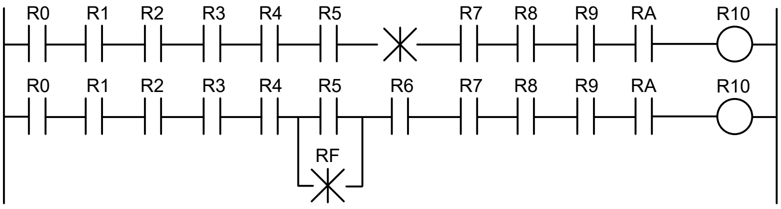
■ Program description using the turning symbol



Cautionary notes

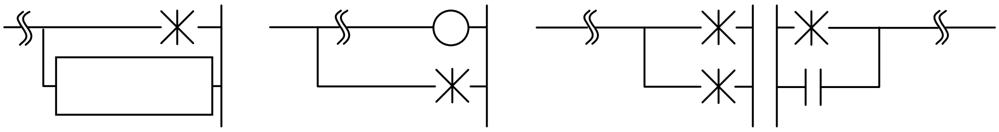
- The turning symbol does not operate by itself.
- It is always used in pairs.
- It can be used up to 32 times.
- The turning symbol can be input to only a position connecting with a master line.

[WRONG]



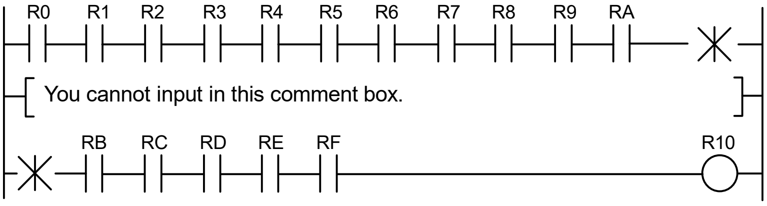
- The turning symbol cannot be connected as OR.

[WRONG]



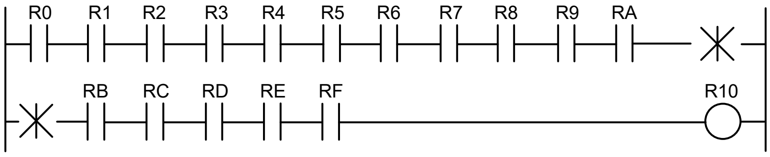
- The comment cannot be input between turning symbols.

[WRONG]



- The turning cannot be extended over the sheets.

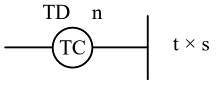
Program example



[Program description]

When all bits from R0 to RF turn ON, R10 turns ON.

名称	On delay timer

<table>
<tr><th rowspan="3">Ladder format</th><th colspan="2">Number of steps</th><th colspan="5">Condition code</th></tr>
<tr><td rowspan="2">Condition</td><td rowspan="2">Steps</td><td>R7F4</td><td>R7F3</td><td>R7F2</td><td>R7F1</td><td>R7F0</td></tr>
<tr><td>DER</td><td>ERR</td><td>SD</td><td>V</td><td>C</td></tr>
<tr><td> </td><td>—</td><td>6</td><td>●</td><td>●</td><td>●</td><td>●</td><td>●</td></tr>
</table>

<table>
<tr><th colspan="6">Command processing time (μs)</th></tr>
<tr><th colspan="3">Average</th><th colspan="3">Maximum</th></tr>
<tr><td rowspan="2">Condition</td><td colspan="2">Time</td><td rowspan="2">Condition</td><td colspan="2">Time</td></tr>
<tr><td>MVH
(High Function)</td><td>MVL
(Standard)</td><td>MVH
(High Function)</td><td>MVL
(Standard)</td></tr>
<tr><td>—</td><td>1.86</td><td>1.98</td><td>—</td><td>—</td><td>—</td></tr>
</table>

<table>
<tr><th colspan="2" rowspan="2">Usable I/O</th><th colspan="6">Bit</th><th colspan="4">Word</th><th colspan="3">Double word</th><th rowspan="2">Constant</th></tr>
<tr><td>X</td><td>Y</td><td>R,M</td><td>TD, TDN,
SS, WDT,
MS, TMR,
CU, RCU,
CT</td><td>WR,
(.m)</td><td>WX</td><td>WY</td><td>WR, WM</td><td>TC</td><td>DX</td><td>DY</td><td>DR,DM</td></tr>
<tr><td>n</td><td>Timer No.</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>✓</td></tr>
<tr><td>t</td><td>Time base</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>s</td><td>Set value</td><td></td><td></td><td></td><td></td><td></td><td>✓</td><td>✓</td><td>✓</td><td></td><td></td><td></td><td></td><td>✓</td></tr>
</table>

Remarks
Timer points are 2,048 points. (0 to 2,047 / decimal) Time base is selectable from 1, 10, 100, and 1,000 [ms]. Set value is 0 to 65,535.

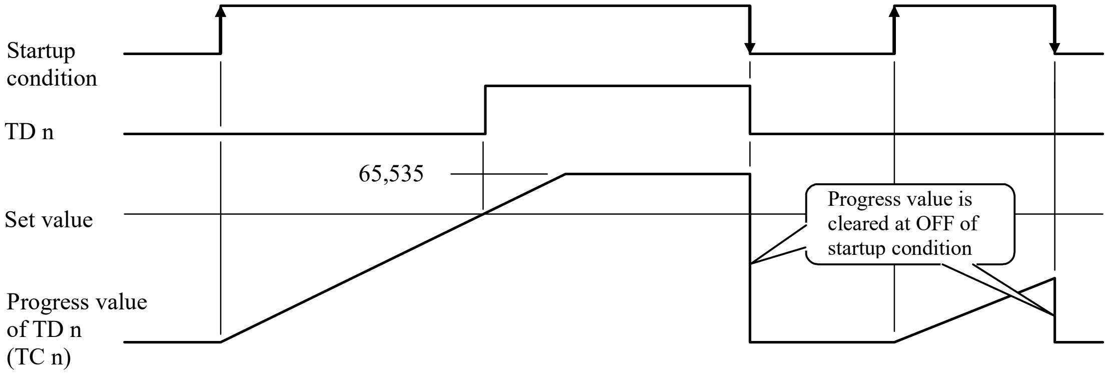
Function

- The elapsed value is updated while the startup condition is ON, and the coil turns ON if the elapsed value ≥ the set value.
- If the startup condition turns OFF, the coil turns OFF after the elapsed value is cleared.
- The elapsed value gets into TC n. The elapsed value does not exceed 65,535 (decimal number).
- If the elapsed value is updated during RUN, it operates according to a new elapsed value at that time.
- If the I/O is specified to the set value, the set value can be changed during operation by changing the I/O value because of taking in the set value at every scan.

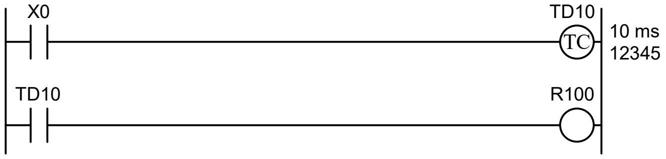
Cautionary notes

- The timer can be used up to 2,048 points including TD, TDN, SS, MS, TMR, and WDT.
  However, the same area as the counter is used. The timer No. and the counter No. cannot be used overlapping.
- The elapsed value of the timer is updated at the scan end.

Time chart

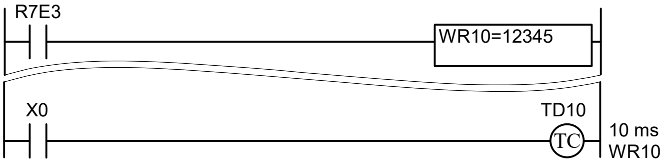


Program example

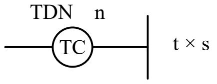


[Program description]

- If X0 turns ON, the elapsed value of TD10 is updated.
- If X0 turns OFF, the elapsed value of TD10 is cleared.
- If the elapsed value ≥ the set value, TD10 turns ON.
- While X 0 is ON, the elapsed value increases but it does not exceed 65,535.
- If X0 turns OFF when TD10 is ON, TD10 turns OFF.
- The set value of the timer can be specified by the word I/O.



Name	Off delay timer

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
	–	6	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
–	1.65	1.98	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
n	Timer No.													✓
t	Time base													
s	Set value						✓	✓	✓					✓

Remarks
Timer points are 2,048 points. (0 to 2,047 / decimal) Time base is selectable from 1, 10, 100, and 1,000 [ms]. Set value is 0 to 65,535.

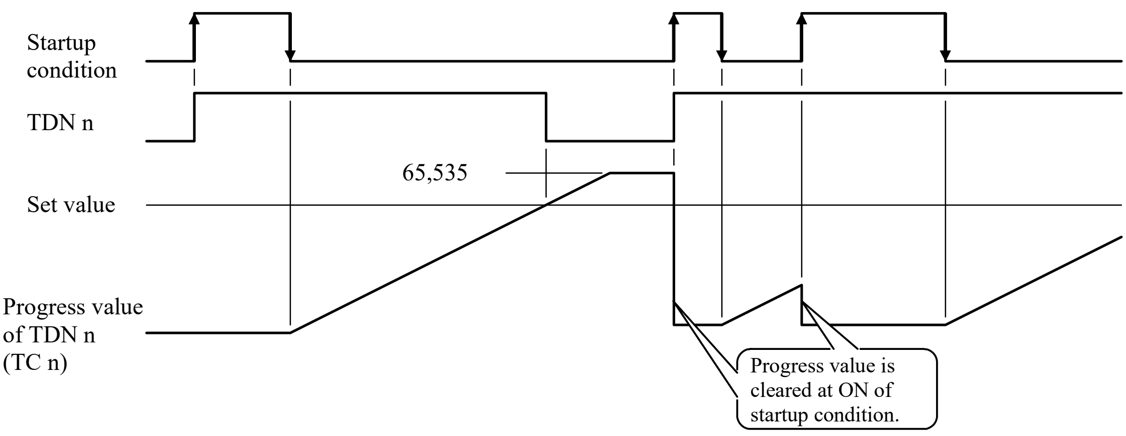
Function

- The rising edge of the startup condition is detected and the coil is turned ON.
- If the startup condition turns OFF, the elapsed value is updated and the coil turns OFF if the elapsed value ≥ the set value.
- If the startup condition turns ON, the elapsed value is cleared.
- The elapsed value gets into TC n. The elapsed value does not exceed 65,535 (decimal number).
- If the elapsed value is updated during RUN, it operates according to a new elapsed value at that time.
- If the I/O is specified to the set value, the set value can be changed during operation by changing the I/O value because of taking in the set value at every scan.

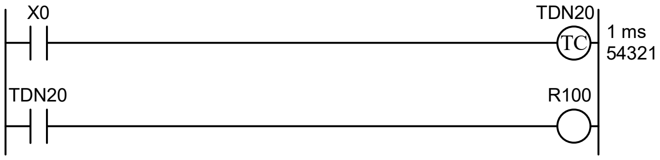
Cautionary notes

- The timer can be used up to 2,048 points including TD, TDN, SS, MS, TMR, and WDT.
  However, the same area as the counter is used. The timer No. and the counter No. cannot be used overlapping.
- The elapsed value of the timer is updated at the scan end.

Time chart

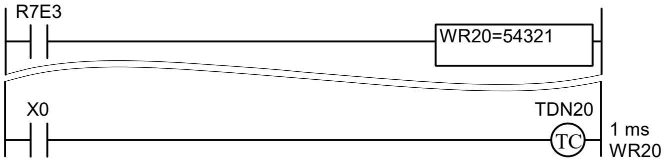


Program example

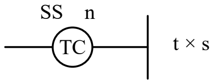


[Program description]

- If X0 turns ON, TDN20 turns ON. After that, if X0 turns OFF, TDN20 starts updating of the elapsed value with ON.
- If the elapsed value ≥ the set value, TDN20 turns OFF.
- When X0 changes from ON to OFF, the elapsed value of TDN20 does not exceed 65,535 although it increases while X0 is OFF.
- If X0 is tined ON while the elapsed value of TDN20 is updated (X0 is OFF), the elapsed value is cleared. (TDN20 holds the ON even if the elapsed value is cleared.)
- The set value is specified by Word I/O like TD.



Name	Single shot

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
	—	6	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
—	2.0	2.05	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
n	Timer No.													✓
t	Time base													
s	Set value						✓	✓	✓					✓

Remarks
Timer points are 2,048 points. (0 to 2,047 / decimal) Time base is selectable from 1, 10, 100, and 1,000 [ms]. Set value is 0 to 65,535.

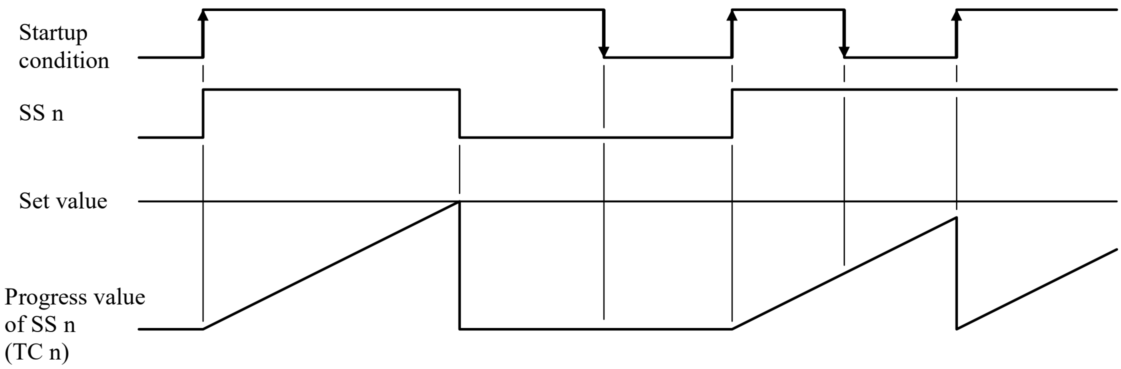
Function

- The rising edge of the startup condition is detected and the elapsed value starts updating. And the coil is turned ON.
- If the elapsed value ≥ the set value, the coil turns OFF. If the rising edge of the startup condition is detected further during the elapsed value < the set value, it is counted from the beginning again with considering the elapsed value to be 0.
- The elapsed value gets into TC n. The elapsed value does not exceed the set value.
- If the elapsed value is updated during RUN, it operates according to a new elapsed value at that time.
- If the I/O is specified to the set value, the set value can be changed during operation by changing the I/O value because of taking in the set value at every scan.

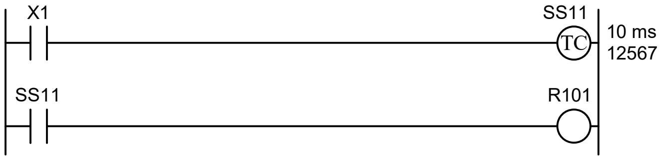
Cautionary notes

- Since the startup condition of the Single shot is the edge detection, it is impossible to detect under the condition of one scan after RUN.
- The timer can be used up to 2,048 points including TD, TDN, SS, MS, TMR, and WDT.
  However, the same area as the counter is used. The timer No. and the counter No. cannot be used overlapping.
- The elapsed value of the timer is updated at the scan end.

Time chart

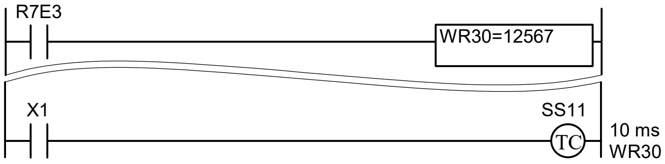


Program example

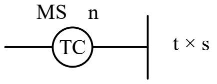


[Program description]

- The elapsed value is updated at the rising edge of X1 and SS11 turns ON.
- If the elapsed value ≥ the set value, SS11 turns OFF.
  The startup condition of the single shot is ignored because of the edge trigger although X1 is ON at this time.
- If the rising edge of X1 is detected before the elapsed value reaches the set value, the single shot timer is triggered again and the elapsed value starts to increase after returning to 0. SS11 holds the ON.
- The set value can be specified by Word I/O like TD.



Name	Mono stable timer

Ladder format


Number of steps		Condition code				
Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
–	6	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
–	1.89	1.94	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)		WX	WY	WR, WM	TC	DX	DY	DR,DM	
n	Timer No.														✓
t	Time base														
s	Set value							✓	✓	✓					✓

Remarks
Timer points are 2,048 points. (0 to 2,047 / decimal) Time base is selectable from 1, 10, 100, and 1,000 [ms]. Set value is 0 to 65,535.

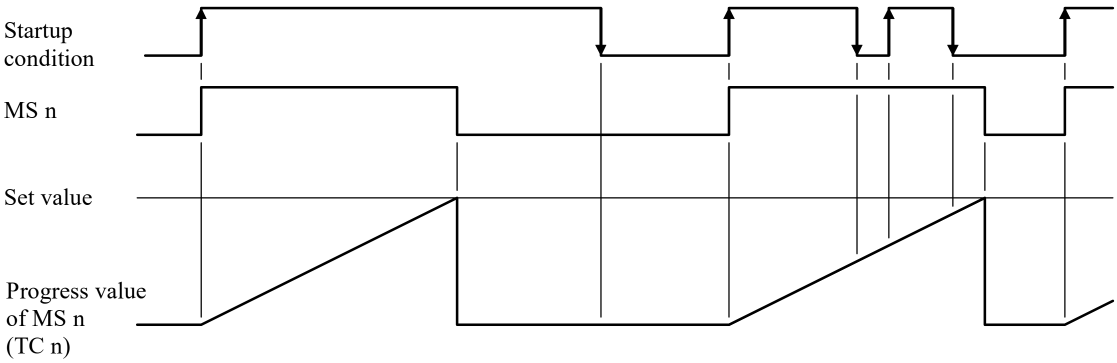
Function

- The rising edge of the startup condition is detected and the elapsed value starts updating. And the coil is turned ON.
- If the elapsed value ≥ the set value, the coil turns OFF. The rising edge of the startup condition is ignored while MS is ON.
- The elapsed value gets into TC n. The elapsed value does not exceed the set value.
- If the elapsed value is updated during RUN, it operates according to a new elapsed value at that time.
- If the I/O is specified to the set value, the set value can be changed during operation by changing the I/O value because of taking in the set value at every scan.

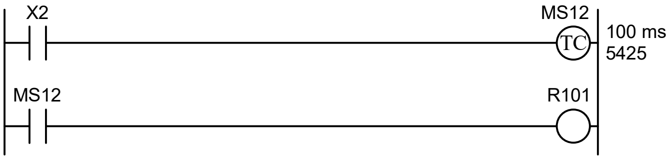
Cautionary notes

- Since the startup condition of the mono stable timer is the edge detection, it is impossible to detect under the condition of one scan after RUN.
- The timer can be used up to 2,048 points including TD, TDN, SS, MS, TMR, and WDT.
  However, the same area as the counter is used. The timer No. and the counter No. cannot be used overlapping.
- The elapsed value of the timer is updated at the scan end.

Time chart

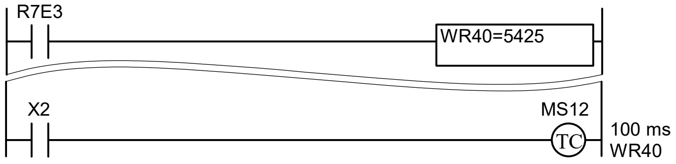


Program example

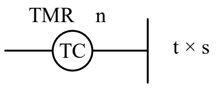


[Program description]

- The elapsed value is updated at the rising edge of X2 and MS12 turns ON.
- If the elapsed value ≥ the set value, MS12 turns OFF.
- The startup condition of the mono stable timer is ignored because of the edge trigger although X2 is ON at this time.
- Even if the rising edge of X2 is detected before the elapsed value reaches the set value, the mono stable timer ignores this rising.
- The set value can be specified by Word I/O like TD.



Name	Integral timer

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
	—	6	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
—	2.0	2.0	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
n	Timer No.														✓
t	Time base														
s	Set value							✓	✓	✓					✓

Remarks
Timer points are 2,048 points. (0 to 2,047 / decimal) Time base is selectable from 1, 10, 100, and 1,000 [ms]. Set value is 0 to 65,535.

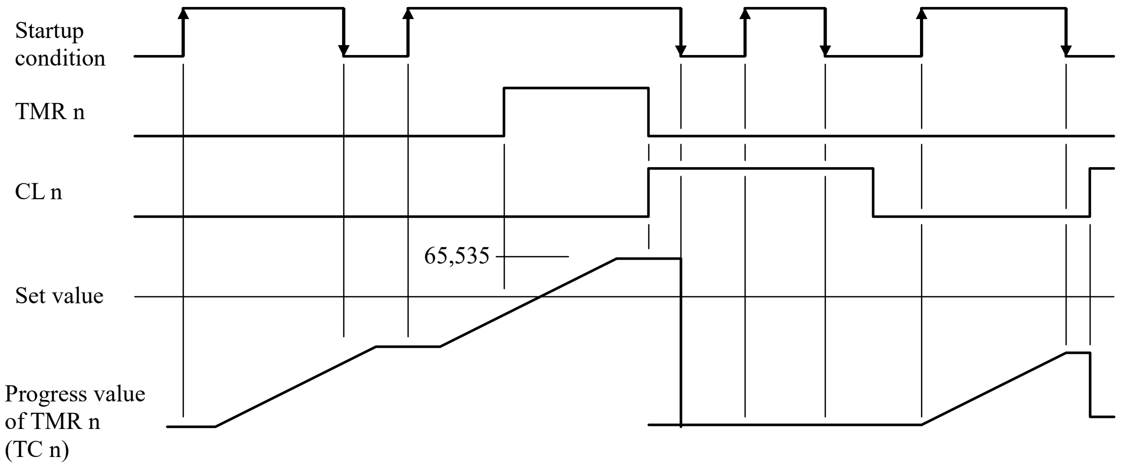
Function

- The elapsed value is updated while the startup condition is ON. The elapsed value restarts updating after the startup condition turns ON again without being cleared even if the condition turns OFF.
- If the elapsed value ≥ the set value, the coil turns ON. And the coil does not turn OFF until the clear input CL n turns ON.
- The elapsed value gets into TC n. The elapsed value does not exceed 65,535 (decimal number).
- If the elapsed value is updated during RUN, it operates according to a new elapsed value at that time.
- If the I/O is specified to the set value, the set value can be changed during operation by changing the I/O value because of taking in the set value at every scan.

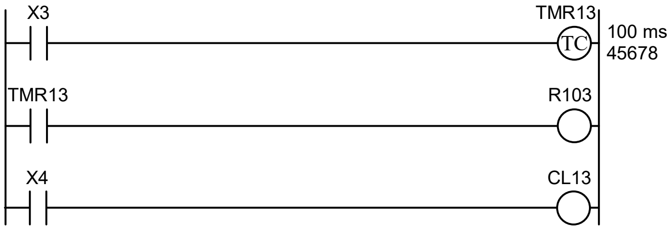
Cautionary notes

- ON of the startup condition is ignored while the clear input CL n is ON.
- The timer can be used up to 2,048 points including TD, TDN, SS, MS, TMR, and WDT.
  However, the same area as the counter is used. The timer No. and the counter No. cannot be used overlapping.
- The elapsed value of the timer is updated at the scan end.

Time chart

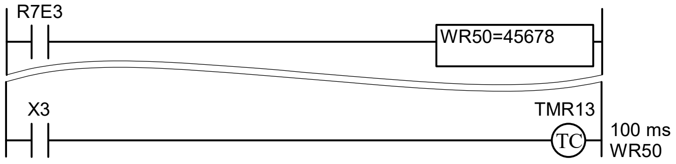


Program example

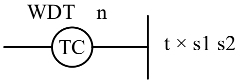


[Program description]

- The elapsed value is updated while X3 is ON.
- If X3 turns OFF, the elapsed value stops updating and is held.
- If X3 turns ON again, the elapsed value restarts updating.
- If the elapsed value ≥ the set value, TMR13 turns ON. TMR13 is held until the timer clear (CL13) turns ON.
- If the timer clear (CL13) turns ON, both the timer coil and the elapsed value are cleared.
- The startup condition is ignored while the timer clear (CL13) is ON.
- The set value can be specified by Word I/O like TD.



Name	Watchdog timer

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
	–	8	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
–	2.58	2.45	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
n	Timer No.													✓
t	Time base													
s1	The 1st set value						✓	✓	✓					✓
s2	The 2nd set value						✓	✓	✓					✓

Remarks
Timer points are 2,048 points. (0 to 2,047 / decimal) Time base is selectable from 1, 10, 100, and 1,000 [ms]. 1st / 2nd set value is 0 to 65,535.

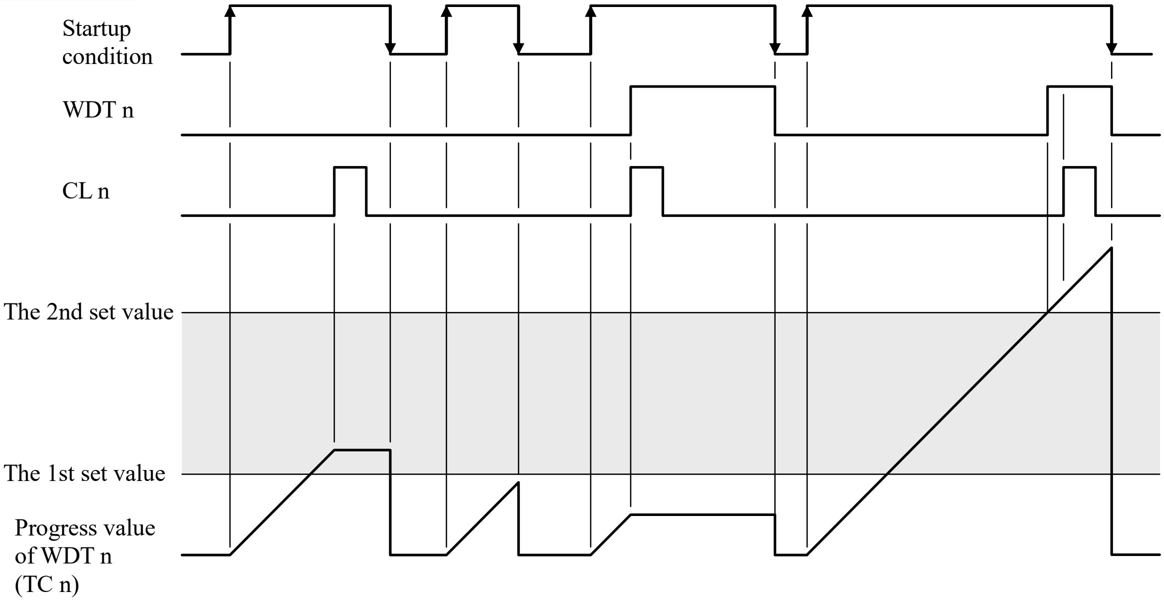
Function

- The elapsed value is updated while the startup condition is ON.
  If the clear input CL n is accessed during the 1st set value ≤ the elapsed value < the 2nd set value, the coil does not turn ON. If the clear input CL n is accessed during the elapsed value < the 1st set value and if the 2nd set value ≤ the elapsed value, the coil turns ON. If the startup condition turns OFF, all is cleared.
- The elapsed value gets into TC n. The elapsed value does not exceed 65,535(decimal number).
- If the elapsed value is updated during RUN, it operates according to a new elapsed value at that time
- If the I/O is specified to the set value, the set value can be changed during operation by changing the I/O value because of taking in the set value at every scan.

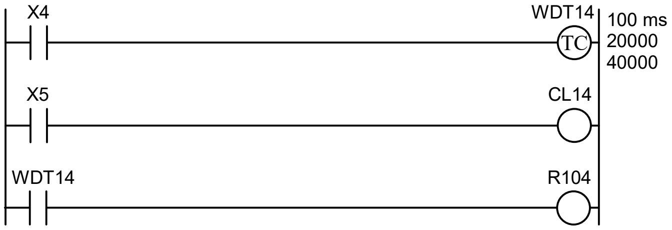
Cautionary notes

- The set value has to fulfill the following condition, s1 < s2.
  Otherwise, the coil turns ON just when the elapsed value reached s2.
- The timer can be used up to 2,048 points including TD, TDN, SS, MS, TMR, and WDT.
  However, the same area the counter is used. The timer No. and the counter No. cannot be used overlapping.
- The elapsed value of the timer is updated at the scan end.

Time chart

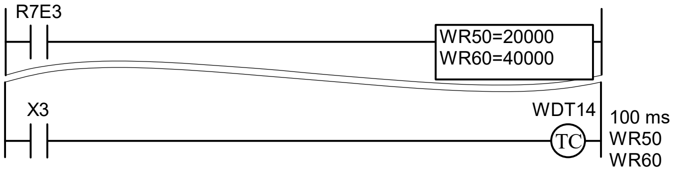


Program example

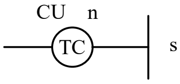


[Program description]

- The clear operates on condition just before the WDT coil command is executed.
- The elapsed value is updated while X4 is ON.
- If the watchdog clear (CL14) is turned ON before the elapsed value exceeds the 2nd set value after exceeding the 1st set value, WDT14 (R104) does not turn ON.
- If X4 turns OFF, the elapsed value and the output of WDT coil are cleared.
- If the startup condition is turned OFF before the elapsed value exceeds the 1st set value, the WDT coil does not turn ON and the elapsed value is cleared to 0 (zero clear).
- If the watchdog clear (CL14) is turned ON before the elapsed value exceeds the 1st set value, WDT14 (R104) turns ON. The elapsed value at that time is held.
- If the watchdog clear (SL14) is not turned ON even if the elapsed value exceeds the 2nd set value, WDT14 turns ON. The elapsed value is updated in succession.
- Even if the watchdog clear (CL14) is turned ON after the elapsed value exceeds the 2nd set value and then WDT coil turns ON, it is ignored.
- The set value can be specified by Word I/O like TD.



Name	Counter

<table>
<tr><th rowspan="3">Ladder format</th><th colspan="2">Number of steps</th><th colspan="5">Condition code</th></tr>
<tr><th rowspan="2">Condition</th><th rowspan="2">Steps</th><th>R7F4</th><th>R7F3</th><th>R7F2</th><th>R7F1</th><th>R7F0</th></tr>
<tr><th>DER</th><th>ERR</th><th>SD</th><th>V</th><th>C</th></tr>
<tr><td> </td><td>–</td><td>6</td><td>●</td><td>●</td><td>●</td><td>●</td><td>●</td></tr>
</table>

<table>
<tr><th colspan="6">Command processing time (μs)</th></tr>
<tr><th colspan="3">Average</th><th colspan="3">Maximum</th></tr>
<tr><th rowspan="2">Condition</th><th colspan="2">Time</th><th rowspan="2">Condition</th><th colspan="2">Time</th></tr>
<tr><th>MVH
(High Function)</th><th>MVL
(Standard)</th><th>MVH
(High Function)</th><th>MVL
(Standard)</th></tr>
<tr><td>–</td><td>2.0</td><td>2.0</td><td>–</td><td>–</td><td>–</td></tr>
</table>

<table>
<tr><th colspan="2" rowspan="2">Usable I/O</th><th colspan="6">Bit</th><th colspan="4">Word</th><th colspan="3">Double word</th><th rowspan="2">Constant</th></tr>
<tr><th>X</th><th>Y</th><th>R,M</th><th>TD, SS, MS, CU, CT</th><th>TDN, WDT, TMR, RCU,</th><th>WR, (.m)</th><th>WX</th><th>WY</th><th>WR, WM</th><th>TC</th><th>DX</th><th>DY</th><th>DR,DM</th></tr>
<tr><td>n</td><td>Counter No.</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>✓</td></tr>
<tr><td>s</td><td>Set value</td><td></td><td></td><td></td><td></td><td></td><td></td><td>✓</td><td>✓</td><td>✓</td><td></td><td></td><td></td><td></td><td>✓</td></tr>
</table>

Remarks
Counter points are 2,048 points. (0 to 2,047 / decimal) Set value is 0 to 65,535.

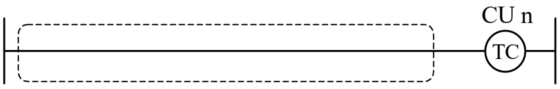
Function

- Whenever the rising edge of the startup condition is detected, the elapsed value increases by 1, and the coil turns ON if the elapsed value ≥ the set value.
  If the counter clear CL n turns ON, the coil turned ON turns OFF and the elapsed value is cleared to 0.
- The elapsed value gets into TC n. The elapsed value does not exceed 65,535 (decimal number).
- If the elapsed value is updated during RUN, it operates according to a new elapsed value at that time.
- If the I/O is specified to the set value, the set value can be changed during operation by changing the I/O value because of taking in the set value at every scan.

Cautionary notes

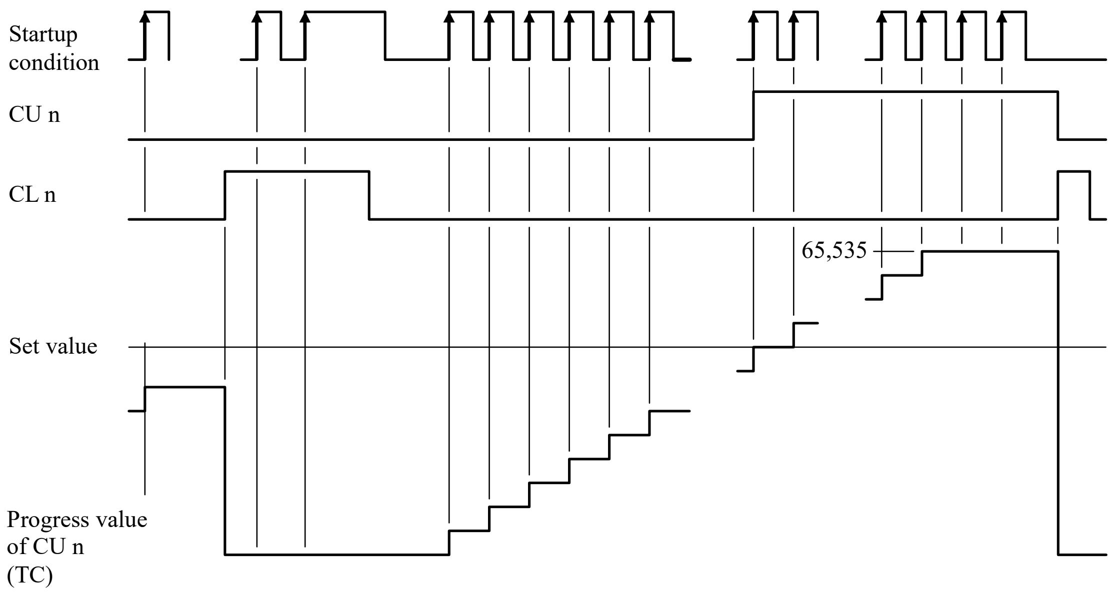
- The counter can be used up to 2,048 points (No.0 to No. 2,047), but the same area as the timer is used.
- The timer No. and the counter No. cannot be used overlapping.
- The counter cannot be used singly. (The condition is needed in front of the coil.)

[WRONG]

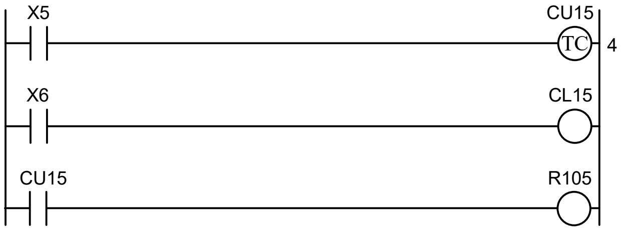


- The rising of the startup condition is ignored while the counter clear CL n is ON.
- Since the startup condition of the counter is the edge detection, it is impossible to detect the condition of one scan after RUN (R7E3).
- The elapsed value of the counter is updated when the counter coil is executed.
- If the set value is set to ‘0’, it is always ON and becomes the coil controlled by CL n.
- The counter is cleared in the counter coil. (The counter is monitored in the counter coil, and is cleared.) If the counter coil is cleared, the counter clear needs to be turned ON before the counter coil is executed.

Time chart

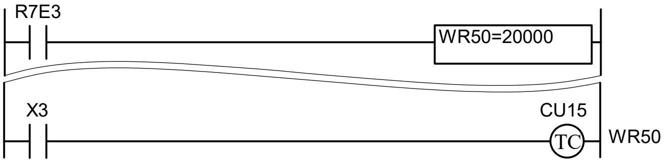


Program example

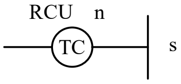


[Program description]

- The elapsed value is updated at the rising edge of X5.
- If the elapsed value ≥ the set value, the counter coil (CU15) turns ON.
- The counter value does not exceed 65,535.
- If the counter coil (CL15) is turned ON, the elapsed value and the counter coil are cleared.
- The set value can be specified by Word I/O like TD.



Name	Ring counter

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
	—	6	●	●	●	●	●

Command processing time (µs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
—	2.07	2.13	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
n	Counter No.													✓
s	Set value						✓	✓	✓					✓

Remarks
Counter points are 2,048 points. (0 to 2,047 / decimal) Set value is 0 to 65,535.

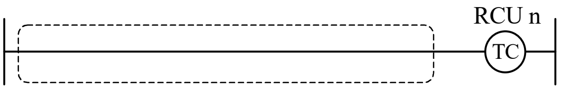
Function

- Whenever the rising edge of the startup condition is detected, the elapsed value increases by 1. If the elapsed value ≥ set value, the elapsed value is cleared to 0 and the coil for one scan is turned ON. If the counter clear CL n turns ON, the elapsed value becomes 0 and the coil also turns OFF.
- The elapsed value gets into TC n. The elapsed value does not exceed the set value.
- If the elapsed value is updated during RUN, it operates according to a new elapsed value at that time. If the I/O is specified to the set value, the set value can be changed during operation by changing the I/O value because of taking in the set value at every scan.

Cautionary notes

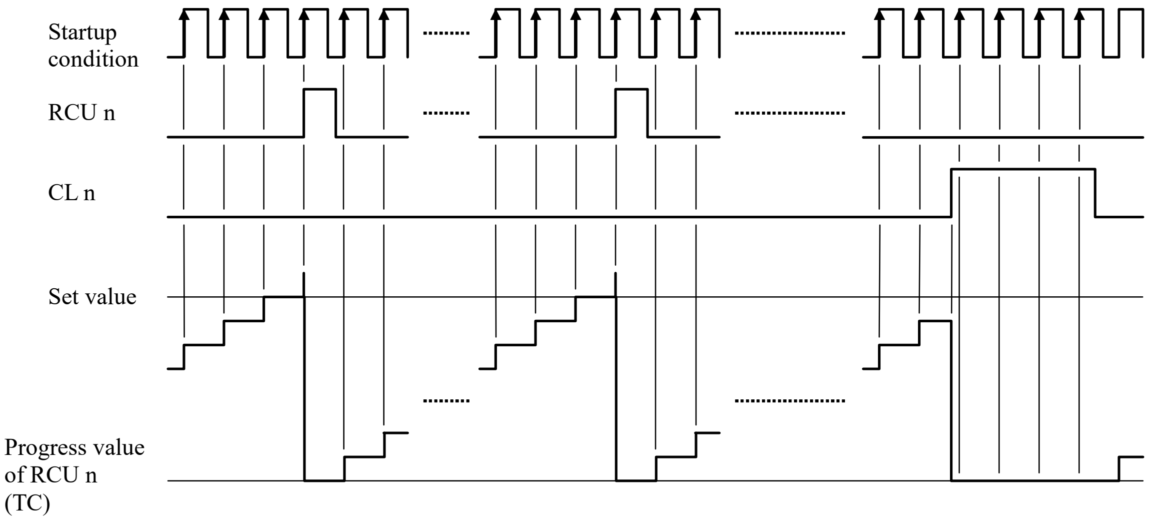
- The counter can be used up to 2,048 points (No.0 to No.2,047), but the same area as the timer is used.
- The timer No. and the counter No. cannot be used overlapping.
- The counter cannot be used singly. (The condition is needed in front of the coil.)

[WRONG]

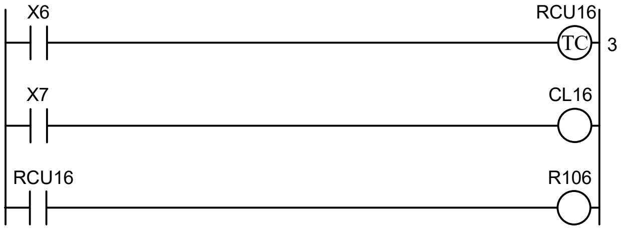


- The rising of the startup condition is ignored while the counter clear CL n is ON.
- Since the startup condition of the counter is the edge detection, it is impossible to detect the condition of one scan after RUN (R7E3).
- <u>The elapsed value of the counter is updated when the counter coil is executed.</u>
- If the set value is set to ‘0’, it is always ON and becomes the coil controlled by CL n.
- The counter is cleared in the counter coil. (The counter is monitored in the counter coil and cleared.) If the counter coil is cleared, the counter clear needs to be turned ON before the counter coil is executed.

Time chart

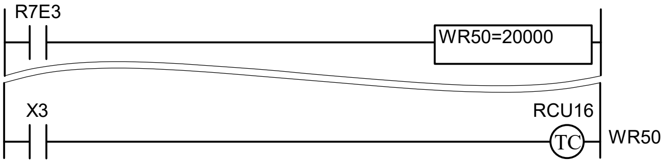


Program example



[Program description]

- The elapsed value (the count value) is updated at the rising edge of X6.
- If the elapsed value ≥ the set value, the count coil (RCU16) turns ON for one scanning time and the elapsed value is cleared to 0.
- If the counter clear (CL16) is turned ON, the elapsed value is cleared to 0. The elapsed value is not updated while the counter clear (CL16) is ON.
- The set value can be specified by Word I/O like TD.



Name	Up counter, Down counter

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
	CTU n s	6	●	●	●	●	●
	CTD n	4					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
CTU n s	2.66	2.91	—	—	—
CTD n	1.3	1.36	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT, TDN, WDT, TMR, RCU	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
n	Counter No.													✓
s	Set value						✓	✓	✓					✓

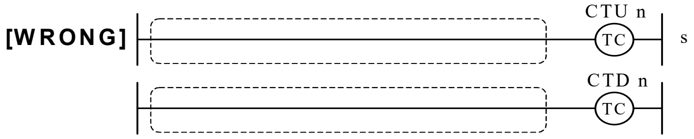
Remarks
Counter points are 2,048 points. (0 to 2,047 / decimal) Set value is 0 to 65,535.

Function

- Whenever the rising edge of the startup condition is detected, the up counter increases the elapsed value by 1and the down counter decreases the elapsed value by 1. The coil is turned ON if the elapsed value ≥ the set value and the coil is turned OFF if the elapsed value < set value. If the counter clear CL n turns ON, the elapsed value is cleared to 0 and the coil also turns OFF.
- The elapsed value gets into TC n. The elapsed value is 0 to 65,535 (decimal number).
- If the elapsed value is updated during RUN, it operates according to a new elapsed value at that time.
- If the I/O is specified to the set value, the set value can be changed during operation by changing the I/O value because of taking in the set value at every scan.

Cautionary notes

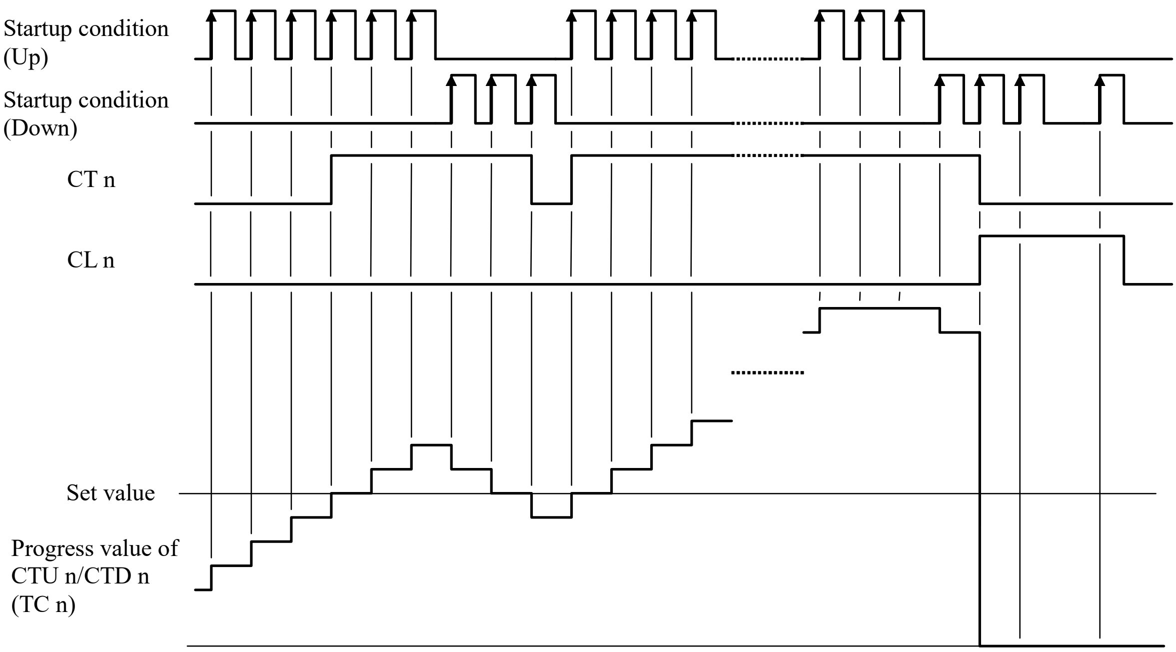
- The number of the up coil and the down coil should be same.
- The counter can be used up to 2,048 points (No.0 to No.2,047), but the same area as the timer is used.
- The timer No. and the counter No. cannot be used overlapping.
- The counter cannot be used singly. (The condition is needed in front of a coil.)



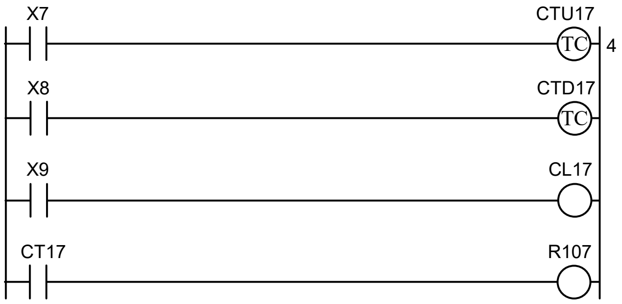
- The rising of the startup condition is ignored while the counter clear CL n is ON.
- Since the startup condition of the counter is the edge detection, it is impossible to detect the condition of one scan after RUN (R7E3).
- <u>The elapsed value of the counter is updated when the counter coil is executed.</u>

- If the set value is set to 0, it is always ON and becomes the coil controlled by CL n.
- The elapsed value is cleared when the counter coil is executed. (The counter clear CL n is monitored in the counter coil and cleared.) If the counter coil is cleared, the counter clear needs to be turned ON before the elapsed value is executed.

Time chart



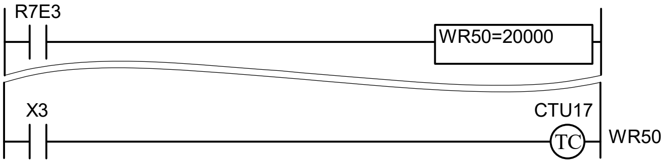
Program example



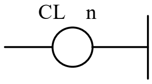
[Program description]

- The elapsed value (the count value) is up-counted at the rising edge of X7.
- The counter coil (CT17) turns ON if the elapsed value ≥ the set value.
- If the startup conditions of the up coil and the down coil turn ON simultaneously, the elapsed value does not change.
- The elapsed value is down-counted at the rising edge of X8.
- The counter coil turns OFF if the elapsed value < the set value.
- The elapsed value does not exceed 65,535 and does not fall below 0 either.

- If the counter clear (CL17) turns ON, the elapsed value and the counter coil are cleared. The elapsed value is not updated while the counter clear is ON.
- The set value can be specified by word I/O same as TD.



Name	Counter clear

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
	—	1	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
—	—	—	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
n	Counter No.													✓

Remarks
Counter points are 2,047 points.(0 to 2,047 / decimal)

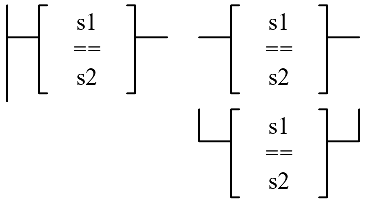
Function

- The elapsed value of the integral timer is cleared to 0 and the timer coil is turned OFF.
- In case of WDT, the time monitor is checked. (Refer to WDT for details.)
- In case of the counter, the elapsed value is cleared and the counter coil is turned OFF.
  It is cleared in the coils of the counter and the timer corresponding to the clear coil. (The clear coil is monitored in the coils of the counter and the timer, and cleared.)

Cautionary notes

- If the timer is turned OFF and the elapsed value is cleared, CL n with the same No. as the timer should be turned ON. This is the same also when clearing the counter.

Name	= Comparison box

Ladder format	Number of steps: Condition	Number of steps: Steps	Condition code R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
	(See Command processing time column)		●	●	●	●	●

Command processing time (μs)

Average

Condition	Time: MVH (High Function) Word		Double word		MVL (Standard) Word		Double word	
LD (s1 == s2) s1:I/O, s2: I/O	1.5	[5]	6.4	[6]	1.52	[5]	7.7	[6]
LD (s1 == s2) s1:I/O, s2:C	1.3	[5]	4.5	[7]	1.39	[5]	5.32	[7]
LD (s1 == s2) s1:C, s2:I/O	1.3	[5]	4.5	[7]	1.36	[5]	5.32	[7]
AND (s1 == s2) s1:I/O, s2: I/O	1.5	[5]	6.3	[6]	1.38	[5]	7.56	[6]
AND (s1 == s2) s1:I/O, s2:C	1.3	[5]	4.5	[7]	1.26	[5]	5.18	[7]
AND (s1 == s2) s1:C, s2: I/O	1.3	[5]	4.5	[7]	1.25	[5]	5.14	[7]
OR (s1 == s2) s1:I/O, s2: I/O	1.7	[6]	6.7	[7]	1.58	[6]	7.76	[7]
OR (s1 == s2) s1:I/O, s2:C	1.6	[6]	4.9	[8]	1.46	[6]	5.48	[8]
OR (s1 == s2) s1:C, s2: I/O	1.5	[6]	4.9	[8]	1.43	[6]	5.4	[8]

Usable I/O		Bit: X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU	WR, (.m)	Word: WX	WY	WR, WM	TC	Double word: DX	DY	DR,DM	Constant
s1	Comparison number 1						✓	✓	✓	✓	✓	✓	✓	✓
s2	Comparison number 2						✓	✓	✓	✓	✓	✓	✓	✓

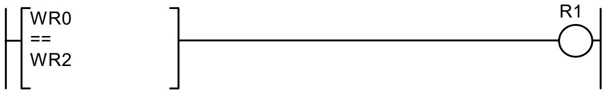
Remarks

C means a constant
[ ]: number of steps (in Processing time column)

Function

- s1 and s2 are compared as unsigned integers,
  when s1 = s2, it is in continuity (ON).
  when s1 ≠ s2, it is in discontinuity (OFF).
- When s1 and s2 are Word, 0 to 65,535 (decimal number),
  H0000 to HFFFF (hexadecimal number)
  When s1 and s2 are Double word, 0 to 4,294,967,295 (decimal number),
  H00000000 to HFFFFFFFF (hexadecimal number)

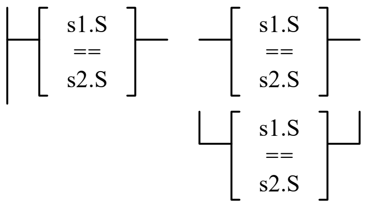
Program example



[Program description]

R1 turns ON when WR0 = WR2 and R1 turns OFF when WR0 ≠ WR2.

Name	= Comparison box (Signed integer)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
	(See Command processing time column)		●	●	●	●	●

Command processing time (μs)

Average

Condition	Processing Time							
	MVH (High Function)				MVL (Standard)			
	Word		Double word		Word		Double word	
LD (s1.S == s2.S) s1:I/O, s2: I/O	1.5	[5]	6.4	[6]	1.6	[5]	7.76	[6]
LD (s1.S == s2.S) s1:I/O, s2:C	1.4	[5]	4.6	[7]	1.47	[5]	5.4	[7]
LD (s1.S == s2.S) s1:C, s2:I/O	1.3	[5]	4.5	[7]	1.44	[5]	5.4	[7]
AND (s1.S == s2.S) s1:I/O, s2: I/O	1.5	[5]	6.4	[6]	1.41	[5]	7.56	[6]
AND (s1.S == s2.S) s1:I/O, s2:C	1.3	[5]	4.5	[7]	1.27	[5]	5.22	[7]
AND (s1.S == s2.S) s1:C, s2: I/O	1.3	[5]	4.5	[7]	1.25	[5]	5.2	[7]
OR (s1.S == s2.S) s1:I/O, s2: I/O	1.7	[6]	6.6	[7]	1.57	[6]	7.76	[7]
OR (s1.S == s2.S) s1:I/O, s2:C	1.6	[6]	4.8	[8]	1.46	[6]	5.4	[8]
OR (s1.S == s2.S) s1:C, s2: I/O	1.5	[6]	4.7	[8]	1.46	[6]	5.36	[8]

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s1.S	Comparison number 1							✓	✓	✓	✓	✓	✓	✓	✓
s2.S	Comparison number 2							✓	✓	✓	✓	✓	✓	✓	✓

Remarks
C means a constant [ ]: number of steps (in Processing time column)

Function

- s1.S and s2.S are compared as signed integers,

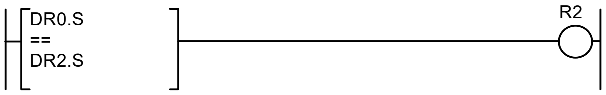
  when s1.S = s2.S, it is in continuity (ON).

  when s1.S ≠ s2.S, it is in discontinuity (OFF).

- When s1.S and s2.S are Word, −32,768 to 32,767 (decimal number),
  H8000 to H7FFF (hexadecimal number)

  When s1.S and s2.S are Double word, −2,147,483,648 to 2,147,483,647 (decimal number),
  H80000000 to H7FFFFFFF (hexadecimal number)

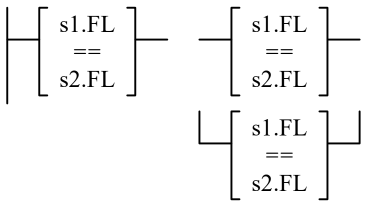
Program example



[Program description]

R2 turns ON when DR0.S = DR2.S and R2 turns OFF when DR0.S ≠ DR2.S.

Name	= Comparison box (Floating point)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
	(See Command processing time column)		●	●	●	●	●

Command processing time (μs)				
Average				
Condition	Processing Time			
	MVH (High Function)		MVL (Standard)	
	Double word		Double word	
LD (s1.FL == s2.FL) s1:I/O, s2: I/O	6.4	[6]	7.74	[6]
LD (s1.FL == s2.FL) s1:I/O, s2:C	4.6	[7]	5.36	[7]
LD (s1.FL == s2.FL) s1:C, s2:I/O	4.5	[7]	5.34	[7]
AND (s1.FL == s2.FL) s1:I/O, s2: I/O	6.4	[6]	7.52	[6]
AND (s1.FL == s2.FL) s1:I/O, s2:C	4.5	[7]	5.22	[7]
AND (s1.FL == s2.FL) s1:C, s2: I/O	4.5	[7]	5.2	[7]
OR (s1.FL == s2.FL) s1:I/O, s2: I/O	6.6	[7]	7.76	[7]
OR (s1.FL == s2.FL) s1:I/O, s2:C	4.8	[8]	5.4	[8]
OR (s1.FL == s2.FL) s1:C, s2: I/O	4.7	[8]	5.36	[8]

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s1.FL	Comparison number 1											✓	✓	✓	✓
s2.FL	Comparison number 2											✓	✓	✓	✓

Remarks
Floating decimal point is specified by Double word. C means a constant. Constant is 20 digit maximum. [ ]: number of steps (in Processing time column)

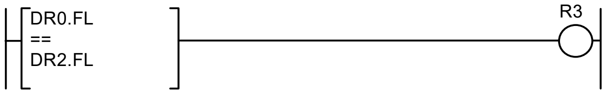
Function

- s1.FL and s2.FL are compared as floating decimal points,
  when s1.FL = s2.FL, it is in continuity (ON).
  when s1.FL ≠ s2.FL, it is in discontinuity (OFF).
- s1.FL, s2.FL: -3.40282×10^{38} to 3.40282×10^{38} (decimal number),
  HFF7FFFFF to H80800000, H00800000 to H7F7FFFFF (hexadecimal number)

Cautionary notes

Since there is an error in floating point, the error may cause the disagreement even if the value from the calculation is in agreement. We recommend deciding the comparison of floating point not in agreement and disagreement but in “range”.

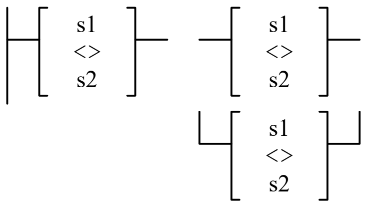
Program example



[Program description]

R3 turns ON when DR0.FL = DR2.FL and R3 turns OFF when DR0.FL ≠ DR2.FL.

Name	<> Comparison box

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
	(See Command processing time column)		●	●	●	●	●

Command processing time (μs)								
Average								
Condition	Processing Time							
	MVH (High Function)				MVL (Standard)			
	Word		Double word		Word		Double word	
LD (s1 <> s2) s1:I/O, s2: I/O	1.5	[5]	6.4	[6]	1.59	[5]	7.77	[6]
LD (s1 <> s2) s1:I/O, s2:C	1.4	[5]	4.5	[7]	1.48	[5]	5.41	[7]
LD (s1 <> s2) s1:C, s2:I/O	1.3	[5]	4.5	[7]	1.44	[5]	5.37	[7]
AND (s1 <> s2) s1:I/O, s2: I/O	1.5	[5]	6.4	[6]	1.4	[5]	7.56	[6]
AND (s1 <> s2) s1:I/O, s2:C	1.4	[5]	4.6	[7]	1.25	[5]	5.2	[7]
AND (s1 <> s2) s1:C, s2: I/O	1.3	[5]	4.6	[7]	1.25	[5]	5.2	[7]
OR (s1 <> s2) s1:I/O, s2: I/O	1.7	[6]	6.6	[7]	1.58	[6]	7.72	[7]
OR (s1 <> s2) s1:I/O, s2:C	1.6	[6]	4.6	[8]	1.46	[6]	5.36	[8]
OR (s1 <> s2) s1:C, s2: I/O	1.5	[6]	4.6	[8]	1.44	[6]	5.36	[8]

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s1	Comparison number 1							✓	✓	✓	✓	✓	✓	✓	✓
s2	Comparison number 2							✓	✓	✓	✓	✓	✓	✓	✓

Remarks
C means a constant [ ]: number of steps (in Processing time column)

Function

- s1and s2 are compared as unsigned integers,

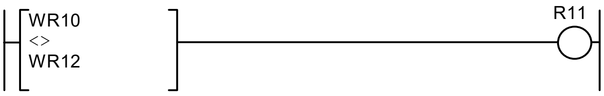
  when s1 ≠ s2, it is in continuity (ON).

  when s1 = s2, it is in discontinuity (OFF).

- When s1 and s2 are Word, 0 to 65,535 (decimal number),
  H0000 to HFFFF (hexadecimal number)

  When s1 and s2 are Double word, 0 to 4,294,967,295 (decimal number),
  H00000000 to HFFFFFFFF (hexadecimal number)

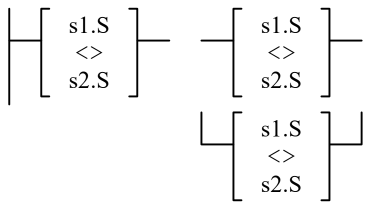
Program example



[Program description]

R11 turns ON when WR10 ≠ WR12 and R11 turns OFF when WR10 = WR12.

Name	≠ Comparison box (Signed integer)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
	(See Command processing time column)		●	●	●	●	●

Command processing time (μs)								
Average								
Condition	Processing Time							
	MVH (High Function)				MVL (Standard)			
	Word		Double word		Word		Double word	
LD (s1.S <> s2.S) s1:I/O, s2: I/O	1.5	[5]	6.4	[6]	1.6	[5]	7.73	[6]
LD (s1.S <> s2.S) s1:I/O, s2:C	1.4	[5]	4.5	[7]	1.46	[5]	5.41	[7]
LD (s1.S <> s2.S) s1:C, s2:I/O	1.4	[5]	4.5	[7]	1.44	[5]	5.39	[7]
AND (s1.S <> s2.S) s1:I/O, s2: I/O	1.4	[5]	6.4	[6]	1.39	[5]	7.56	[6]
AND (s1.S <> s2.S) s1:I/O, s2:C	1.3	[5]	4.6	[7]	1.27	[5]	5.18	[7]
AND (s1.S <> s2.S) s1:C, s2: I/O	1.3	[5]	4.5	[7]	1.25	[5]	5.12	[7]
OR (s1.S <> s2.S) s1:I/O, s2: I/O	1.7	[6]	6.5	[7]	1.58	[6]	7.8	[7]
OR (s1.S <> s2.S) s1:I/O, s2:C	1.6	[6]	4.7	[8]	1.46	[6]	5.44	[8]
OR (s1.S <> s2.S) s1:C, s2: I/O	1.5	[6]	4.7	[8]	1.45	[6]	5.48	[8]

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s1.S	Comparison number 1							✓	✓	✓	✓	✓	✓	✓	✓
s2.S	Comparison number 2							✓	✓	✓	✓	✓	✓	✓	✓

Remarks
C means a constant [ ]: number of steps (in Processing time column)

Function

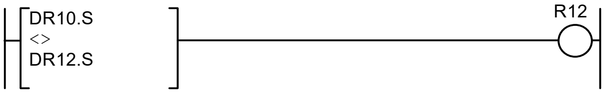
- s1.S and s2.S are compared as signed integers.

  When s1.S ≠ s2.S, it is in continuity (ON).

  When s1.S = s2.S, it is in discontinuity (OFF).

- When s1.S and s2.S are Word, −32,768 to32,767 (decimal number), H8000 to H7FFF (hexadecimal number) When s1.S and s2.S are Double word, −2,147,483,648 to 2,147,483,647 (decimal number), H80000000 to H7FFFFFFF (hexadecimal number)

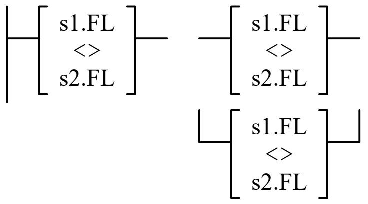
Program example



[Program description]

R12 turns ON when DR10.S ≠ DR12.S and R12 turns OFF when DR10.S = DR12.S.

Name	≠ Comparison box (Floating points)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
	(See Command processing time column)		●	●	●	●	●

Command processing time (μs)

Average

Condition	Processing Time			
	MVH (High Function)		MVL (Standard)	
	Double word		Double word	
LD (s1.FL <> s2.FL) s1:I/O, s2: I/O	6.4	[6]	7.76	[6]
LD (s1.FL <> s2.FL) s1:I/O, s2:C	4.5	[7]	5.6	[7]
LD (s1.FL <> s2.FL) s1:C, s2:I/O	4.5	[7]	5.58	[7]
AND (s1.FL <> s2.FL) s1:I/O, s2: I/O	6.4	[6]	7.56	[6]
AND (s1.FL <> s2.FL) s1:I/O, s2:C	4.6	[7]	5.24	[7]
AND (s1.FL <> s2.FL) s1:C, s2: I/O	4.5	[7]	5.2	[7]
OR (s1.FL <> s2.FL) s1:I/O, s2: I/O	6.5	[7]	7.68	[7]
OR (s1.FL <> s2.FL) s1:I/O, s2:C	4.7	[8]	5.36	[8]
OR (s1.FL <> s2.FL) s1:C, s2: I/O	4.8	[8]	5.36	[8]

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s1.FL	Comparison number 1											✓	✓	✓	✓
s2.FL	Comparison number 2											✓	✓	✓	✓

Remarks

Floating decimal point is specified by Double word.
C means a constant.
Constant is 20 digit maximum.
[ ]: number of steps (in Processing time column)

Function

- s1.FL and s2.FL are compared as floating points,

  when s1.FL ≠ s2.FL, it is in continuity (ON).

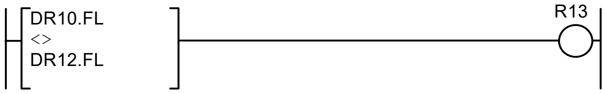
  when s1.FL = s2.FL, it is in discontinuity (OFF).
- s1.FL, s2.FL: -3.40282×10^{38} to 3.40282×10^{38} (decimal number),

  HFF7FFFFF to H80800000, H00800000 to H7F7FFFFF (hexadecimal number)

Cautionary notes

Since there is an error in floating point, the error may cause the disagreement even if the value from the calculation is in agreement. We recommend deciding the comparison of floating points not in agreement and disagreement but in “range”.

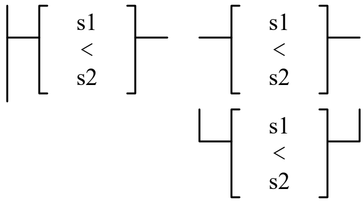
Program example



[Program description]

R13 turns ON when DR10.FL ≠ DR12.FL and R13 turns OFF when DR10.FL = DR12.FL.

Name	< Comparison box

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
	(See Command processing time column)		●	●	●	●	●

Command processing time (μs)								
Average								
Condition	Processing Time							
	MVH (High Function)				MVL (Standard)			
	Word		Double word		Word		Double word	
LD (s1 < s2) s1:I/O, s2: I/O	1.5	[5]	6.4	[6]	1.61	[5]	7.75	[6]
LD (s1 < s2) s1:I/O, s2:C	1.4	[5]	4.5	[7]	1.45	[5]	5.37	[7]
LD (s1 < s2) s1:C, s2:I/O	1.3	[5]	4.5	[7]	1.44	[5]	5.41	[7]
AND (s1 < s2) s1:I/O, s2: I/O	1.5	[5]	6.4	[6]	1.38	[5]	7.56	[6]
AND (s1 < s2) s1:I/O, s2:C	1.3	[5]	4.6	[7]	1.25	[5]	5.26	[7]
AND (s1 < s2) s1:C, s2: I/O	1.3	[5]	4.5	[7]	1.25	[5]	5.12	[7]
OR (s1 < s2) s1:I/O, s2: I/O	1.7	[6]	6.5	[7]	1.58	[6]	7.72	[7]
OR (s1 < s2) s1:I/O, s2:C	1.5	[6]	4.7	[8]	1.44	[6]	5.4	[8]
OR (s1 < s2) s1:C, s2: I/O	1.5	[6]	4.7	[8]	1.44	[6]	5.4	[8]

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s1	Comparison number 1						✓	✓	✓	✓	✓	✓	✓	✓
s2	Comparison number 2						✓	✓	✓	✓	✓	✓	✓	✓

Remarks
C means a constant [ ]: number of steps (in Processing time column)

Function

- s1 and s2 are compared as unsigned integers,

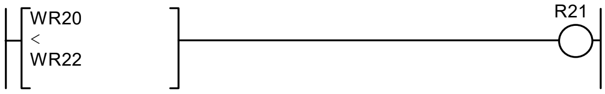
  when s1 < s2, it is in continuity (ON).

  when s1 ≥ s2, it is in discontinuity (OFF).

- When s1 and s2 are Word, 0 to 65,535 (decimal number),
  H0000 to HFFFF (hexadecimal number)

  When s1 and s2 are Double word, 0 to 4,294,967,295 (decimal number),
  H00000000 to HFFFFFFFF (hexadecimal number)

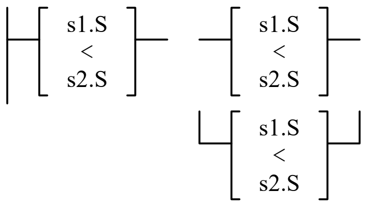
Program example



[Program description]

R21 turns ON when WR20 < WR22 and R21 turns OFF when WR20 ≥ WR22.

Name	< Comparison box (Signed integer)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
	(See Command processing time column)		●	●	●	●	●

Command processing time (μs)								
Average								
Condition	Processing Time							
	MVH (High Function)				MVL (Standard)			
	Word		Double word		Word		Double word	
LD (s1.S < s2.S) s1:I/O, s2: I/O	5.5	[5]	6.4	[6]	6.75	[5]	7.72	[6]
LD (s1.S < s2.S) s1:I/O, s2:C	4.0	[5]	4.5	[7]	4.71	[5]	5.4	[7]
LD (s1.S < s2.S) s1:C, s2:I/O	4.0	[5]	4.5	[7]	4.69	[5]	5.36	[7]
AND (s1.S < s2.S) s1:I/O, s2: I/O	5.5	[5]	6.3	[6]	6.6	[5]	7.56	[6]
AND (s1.S < s2.S) s1:I/O, s2:C	4.0	[5]	4.5	[7]	4.56	[5]	5.22	[7]
AND (s1.S < s2.S) s1:C, s2: I/O	4.0	[5]	4.5	[7]	4.58	[5]	5.2	[7]
OR (s1.S < s2.S) s1:I/O, s2: I/O	5.8	[6]	6.6	[7]	6.76	[6]	7.72	[7]
OR (s1.S < s2.S) s1:I/O, s2:C	4.0	[6]	4.8	[8]	4.76	[6]	5.36	[8]
OR (s1.S < s2.S) s1:C, s2: I/O	4.0	[6]	4.7	[8]	4.76	[6]	5.36	[8]

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s1.S	Comparison number 1							✓	✓	✓	✓	✓	✓	✓	✓
s2.S	Comparison number 2							✓	✓	✓	✓	✓	✓	✓	✓

Remarks
C means a constant. [ ]: number of steps (in Processing time column)

Function

- s1.S and s2.S are compared as signed integers,

  when s1.S < s2.S, it is in continuity (ON).

  when s1.S ≥ s2.S, it is in discontinuity (OFF).

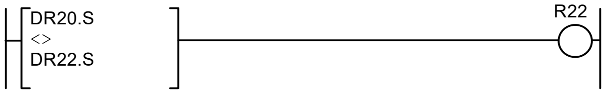
- When s1.S and s2.S are Word, −32,768 to 32,767 (decimal number),

  H8000 to H7FFF (hexadecimal number)

  When s1.S and s2.S are Double word, −2,147,483,648 to 2,147,483,647 (decimal number),

  H80000000 to H7FFFFFFF (hexadecimal number)

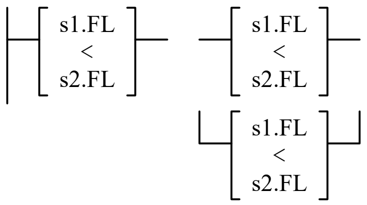
Program example



[Program description]

R22 turns ON when DR20.S < DR22.S and R22 turns OFF when DR20.S ≥ DR22.S.

Name	< Comparison box (Floating point)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
	(See Command processing time column)		●	●	●	●	●

Command processing time (μs)

Average

Condition	Processing Time			
	MVH (High Function)		MVL (Standard)	
	Double word		Double word	
LD (s1.FL < s2.FL) s1:I/O, s2: I/O	7.3	[6]	8.86	[6]
LD (s1.FL < s2.FL) s1:I/O, s2:C	5.3	[7]	6.44	[7]
LD (s1.FL < s2.FL) s1:C, s2:I/O	5.4	[7]	6.48	[7]
AND (s1.FL < s2.FL) s1:I/O, s2: I/O	7.2	[6]	8.68	[6]
AND (s1.FL < s2.FL) s1:I/O, s2:C	5.4	[7]	6.24	[7]
AND (s1.FL < s2.FL) s1:C, s2: I/O	5.3	[7]	6.28	[7]
OR (s1.FL < s2.FL) s1:I/O, s2: I/O	7.4	[7]	8.9	[7]
OR (s1.FL < s2.FL) s1:I/O, s2:C	5.5	[8]	6.48	[8]
OR (s1.FL < s2.FL) s1:C, s2: I/O	5.6	[8]	6.44	[8]

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s1.FL	Comparison number 1										✓	✓	✓	✓
s2.FL	Comparison number 2										✓	✓	✓	✓

Remarks

Floating decimal point is specified by Double word.
C means a constant.
Constant is 20 digit maximum.
[ ]: number of steps (in Processing time column)

Function

- s1.FL and s2.FL are compared as floating points,

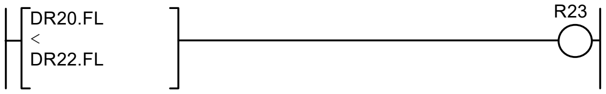
  when s1.FL < s2.FL, it is in continuity (ON).

  when s1.FL ≥ s2.FL, it is in discontinuity (OFF).

- s1.FL, s2.FL: -3.40282×10^{38} to 3.40282×10^{38} (decimal number),

  HFF7FFFFF to H80800000, H00800000 to H7F7FFFFF (hexadecimal number)

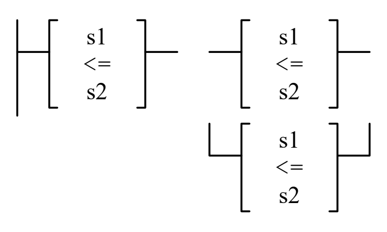
Program example



[Program description]

R23 turns ON when DR20.FL < DR22.FL and R23 turns OFF when DR20.FL ≥ DR22.FL.

Name	≤ Comparison box

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
	(See Command processing time column)		●	●	●	●	●

Command processing time (μs)								
Average								
Condition	Processing Time							
	MVH (High Function)				MVL (Standard)			
	Word		Double word		Word		Double word	
LD (s1 <= s2) s1:I/O, s2: I/O	1.5	[5]	6.3	[6]	1.6	[5]	7.7	[6]
LD (s1 <= s2) s1:I/O, s2:C	1.4	[5]	4.5	[7]	1.46	[5]	5.34	[7]
LD (s1 <= s2) s1:C, s2:I/O	1.4	[5]	4.5	[7]	1.46	[5]	5.38	[7]
AND (s1 <= s2) s1:I/O, s2: I/O	1.5	[5]	6.4	[6]	1.4	[5]	7.6	[6]
AND (s1 <= s2) s1:I/O, s2:C	1.5	[5]	4.6	[7]	1.25	[5]	5.22	[7]
AND (s1 <= s2) s1:C, s2: I/O	1.5	[5]	4.5	[7]	1.24	[5]	5.18	[7]
OR (s1 <= s2) s1:I/O, s2: I/O	1.7	[6]	6.6	[7]	1.58	[6]	7.68	[7]
OR (s1 <= s2) s1:I/O, s2:C	1.6	[6]	4.7	[8]	1.44	[6]	5.44	[8]
OR (s1 <= s2) s1:C, s2: I/O	1.6	[6]	4.8	[8]	1.44	[6]	5.36	[8]

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s1	Comparison number 1						✓	✓	✓	✓	✓	✓	✓	✓
s2	Comparison number 2						✓	✓	✓	✓	✓	✓	✓	✓

Remarks
C means a constant [ ]: number of steps (in Processing time column)

Function

- s1 and s2 are compared as unsigned integers,

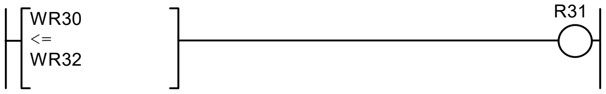
  when s1 ≤ s2, it is in continuity (ON).

  when s1 > s2, it is in discontinuity (OFF).

- When s1 and s2 are Word, 0 to 65,535 (decimal number), H0000 to HFFFF (hexadecimal number)

  When s1 and s2 are Double word, 0 to 4,294,967,295 (decimal number), H00000000 to HFFFFFFFF (hexadecimal number)

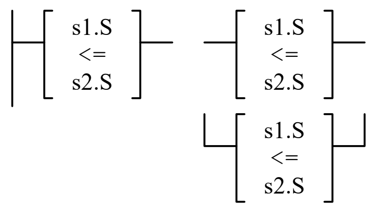
Program example



[Program description]

R31 turns ON when WR30 ≤ WR32 and R31 turns OFF when WR30 > WR32.

Name	≤ Comparison box (Signed integer)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
	(See Command processing time column)		●	●	●	●	●

Command processing time (μs)

Average

Condition	Processing Time							
	MVH (High Function)				MVL (Standard)			
	Word		Double word		Word		Double word	
LD (s1.S <= s2.S) s1:I/O, s2: I/O	5.5	[5]	6.3	[6]	6.73	[5]	7.71	[6]
LD (s1.S <= s2.S) s1:I/O, s2:C	3.9	[5]	4.5	[7]	4.71	[5]	5.43	[7]
LD (s1.S <= s2.S) s1:C, s2:I/O	3.9	[5]	4.5	[7]	4.73	[5]	5.37	[7]
AND (s1.S <= s2.S) s1:I/O, s2: I/O	5.5	[5]	6.3	[6]	6.58	[5]	7.52	[6]
AND (s1.S <= s2.S) s1:I/O, s2:C	3.9	[5]	4.5	[7]	4.54	[5]	5.2	[7]
AND (s1.S <= s2.S) s1:C, s2: I/O	4.0	[5]	4.5	[7]	4.52	[5]	5.18	[7]
OR (s1.S <= s2.S) s1:I/O, s2: I/O	5.9	[6]	5.9	[7]	6.8	[6]	7.68	[7]
OR (s1.S <= s2.S) s1:I/O, s2:C	4.2	[6]	4.7	[8]	4.72	[6]	5.4	[8]
OR (s1.S <= s2.S) s1:C, s2: I/O	4.0	[6]	4.7	[8]	4.74	[6]	5.36	[8]

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s1.S	Comparison number 1						✓	✓	✓	✓	✓	✓	✓	✓
s2.S	Comparison number 2						✓	✓	✓	✓	✓	✓	✓	✓

Remarks

C means a constant
[ ]: number of steps (in Processing time column)

Function

- s1.S and s2.S are compared as signed integers,

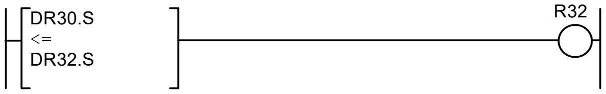
  when s1.S ≤ s2.S, it is in continuity (ON).

  when s1.S > s2.S, it is in discontinuity (OFF).

- When s1.S and s2.S are Word, −32,768 to 32,767 (decimal number),
  H8000 to H7FFF (hexadecimal number)

  When s1.S and s2.S are Double word, −2,147,483,648 to 2,147,483,647 (decimal number),
  H80000000 to H7FFFFFFF (hexadecimal number)

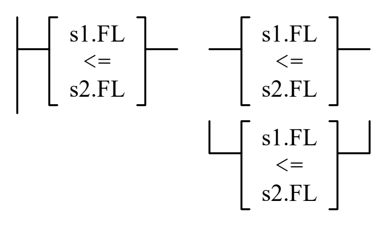
Program example



[Program description]

R32 turns ON when DR30.S ≤ DR32.S and R32 turns OFF when DR30.S > DR32.S.

Name	≤ Comparison box (Floating point)

<table>
<tr><th rowspan="3">Ladder format</th><th colspan="2">Number of steps</th><th colspan="5">Condition code</th></tr>
<tr><td rowspan="2">Condition</td><td rowspan="2">Steps</td><td>R7F4</td><td>R7F3</td><td>R7F2</td><td>R7F1</td><td>R7F0</td></tr>
<tr><td>DER</td><td>ERR</td><td>SD</td><td>V</td><td>C</td></tr>
<tr><td> </td><td colspan="2">(See Command processing time column)</td><td>●</td><td>●</td><td>●</td><td>●</td><td>●</td></tr>
</table>

<table>
<tr><th colspan="5">Command processing time (μs)</th></tr>
<tr><td colspan="5">Average</td></tr>
<tr><td rowspan="3">Condition</td><td colspan="4">Processing Time</td></tr>
<tr><td colspan="2">MVH
(High Function)</td><td colspan="2">MVL
(Standard)</td></tr>
<tr><td colspan="2">Double word</td><td colspan="2">Double word</td></tr>
<tr><td>LD (s1.FL <= s2.FL) s1:I/O, s2: I/O</td><td>7.2</td><td>[6]</td><td>7.7</td><td>[6]</td></tr>
<tr><td>LD (s1.FL <= s2.FL) s1:I/O, s2:C</td><td>5.4</td><td>[7]</td><td>5.34</td><td>[7]</td></tr>
<tr><td>LD (s1.FL <= s2.FL) s1:C, s2:I/O</td><td>5.4</td><td>[7]</td><td>5.38</td><td>[7]</td></tr>
<tr><td>AND (s1.FL <= s2.FL) s1:I/O, s2: I/O</td><td>7.2</td><td>[6]</td><td>7.6</td><td>[6]</td></tr>
<tr><td>AND (s1.FL <= s2.FL) s1:I/O, s2:C</td><td>5.3</td><td>[7]</td><td>5.22</td><td>[7]</td></tr>
<tr><td>AND (s1.FL <= s2.FL) s1:C, s2: I/O</td><td>5.4</td><td>[7]</td><td>5.18</td><td>[7]</td></tr>
<tr><td>OR (s1.FL <= s2.FL) s1:I/O, s2: I/O</td><td>7.4</td><td>[7]</td><td>7.68</td><td>[7]</td></tr>
<tr><td>OR (s1.FL <= s2.FL) s1:I/O, s2:C</td><td>5.6</td><td>[8]</td><td>5.44</td><td>[8]</td></tr>
<tr><td>OR (s1.FL <= s2.FL) s1:C, s2: I/O</td><td>5.6</td><td>[8]</td><td>5.36</td><td>[8]</td></tr>
</table>

<table>
<tr><th colspan="2" rowspan="2">Usable I/O</th><th colspan="5">Bit</th><th colspan="4">Word</th><th colspan="3">Double word</th><th rowspan="2">Constant</th></tr>
<tr><td>X</td><td>Y</td><td>R,M</td><td>TD, SS, MS, CU, CT
TDN, WDT, TMR, RCU,</td><td>WR, (.m)</td><td>WX</td><td>WY</td><td>WR, WM</td><td>TC</td><td>DX</td><td>DY</td><td>DR,DM</td></tr>
<tr><td>s1.FL</td><td>Comparison number 1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>✓</td><td>✓</td><td>✓</td><td>✓</td></tr>
<tr><td>s2.FL</td><td>Comparison number 2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>✓</td><td>✓</td><td>✓</td><td>✓</td></tr>
</table>

Remarks
Floating decimal point is specified by Double word. C means a constant. Constant is 20 digit maximum. [ ]: number of steps (in Processing time column)

Function

- s1.FL and s2.FL are compared as floating points,

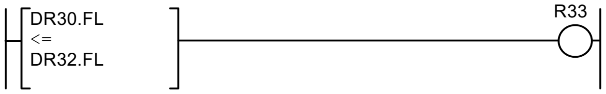
  when s1.FL ≤ s2.FL, it is in continuity (ON).

  when s1.FL > s2.FL, it is in discontinuity (OFF).

- s1.FL, s2.FL: -3.40282×10^{38} to 3.40282×10^{38} (decimal number),

  HFF7FFFFF to H80800000, H00800000 to H7F7FFFFF (hexadecimal number)

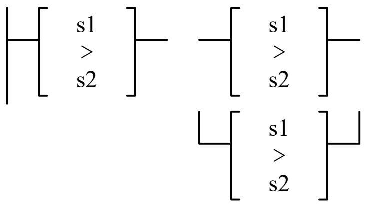
Program example



[Program description]

R33 turns ON when DR30.FL ≤ DR32.FL and R33 turns OFF when DR30.FL > DR32.FL.

Name	> Comparison box

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
	(See Command processing time column)		●	●	●	●	●

Command processing time (μs)								
Average								
Condition	Processing Time							
	MVH (High Function)				MVL (Standard)			
	Word		Double word		Word		Double word	
LD (s1 > s2) s1:I/O, s2: I/O	1.5	[5]	6.4	[6]	1.6	[5]	7.76	[6]
LD (s1 > s2) s1:I/O, s2:C	1.4	[5]	4.5	[7]	1.47	[5]	5.42	[7]
LD (s1 > s2) s1:C, s2:I/O	1.3	[5]	4.5	[7]	1.45	[5]	5.38	[7]
AND (s1 > s2) s1:I/O, s2: I/O	1.5	[5]	6.4	[6]	1.39	[5]	7.56	[6]
AND (s1 > s2) s1:I/O, s2:C	1.3	[5]	4.5	[7]	1.28	[5]	5.24	[7]
AND (s1 > s2) s1:C, s2: I/O	1.3	[5]	4.5	[7]	1.25	[5]	5.24	[7]
OR (s1 > s2) s1:I/O, s2: I/O	1.7	[6]	6.5	[7]	1.61	[6]	7.72	[7]
OR (s1 > s2) s1:I/O, s2:C	1.6	[6]	4.8	[8]	1.45	[6]	5.36	[8]
OR (s1 > s2) s1:C, s2: I/O	1.5	[6]	4.6	[8]	1.46	[6]	5.36	[8]

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s1	Comparison number 1						✓	✓	✓	✓	✓	✓	✓	✓
s2	Comparison number 2						✓	✓	✓	✓	✓	✓	✓	✓

Remarks
C means a constant [ ]: number of steps (in Processing time column)

Function

- s1 and s2 are compared as unsigned integers,

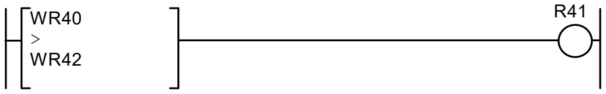
  when s1 > s2, it is in continuity (ON).

  when s1 ≤ s2, it is in discontinuity (OFF).

- When s1 and s2 are Word, 0 to 65,535 (decimal number), H0000 to HFFFF (hexadecimal number)

  When s1 and s2 are Double word, 0 to 4,294,967,295 (decimal number), H00000000 to HFFFFFFFF (hexadecimal number)

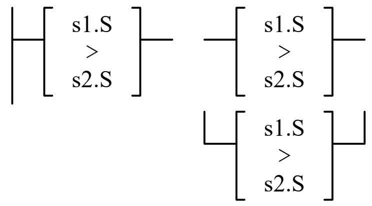
Program example



[Program description]

R41 turns ON when WR40 > WR42 and R41 turns OFF when WR40 ≤ WR42.

Name	> Comparison box (Signed integer)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
	(See Command processing time column)		●	●	●	●	●

Command processing time (μs)

Average

Condition	Processing Time							
	MVH (High Function)				MVL (Standard)			
	Word		Double word		Word		Double word	
LD (s1.S > s2.S) s1:I/O, s2: I/O	5.3	[5]	6.4	[6]	6.73	[5]	7.71	[6]
LD (s1.S > s2.S) s1:I/O, s2:C	4.0	[5]	4.5	[7]	4.71	[5]	5.39	[7]
LD (s1.S > s2.S) s1:C, s2:I/O	4.0	[5]	4.5	[7]	4.72	[5]	5.37	[7]
AND (s1.S > s2.S) s1:I/O, s2: I/O	5.6	[5]	6.3	[6]	6.6	[5]	7.52	[6]
AND (s1.S > s2.S) s1:I/O, s2:C	4.0	[5]	4.6	[7]	4.54	[5]	5.2	[7]
AND (s1.S > s2.S) s1:C, s2: I/O	4.0	[5]	4.5	[7]	4.56	[5]	5.2	[7]
OR (s1.S > s2.S) s1:I/O, s2: I/O	5.7	[6]	6.5	[7]	6.76	[6]	7.84	[7]
OR (s1.S > s2.S) s1:I/O, s2:C	4.1	[6]	4.8	[8]	4.72	[6]	5.4	[8]
7OR (s1.S > s2.S) s1:C, s2: I/O	4.0	[6]	4.8	[8]	4.72	[6]	5.4	[8]

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s1.S	Comparison number 1						✓	✓	✓	✓	✓	✓	✓	✓
s2.S	Comparison number 2						✓	✓	✓	✓	✓	✓	✓	✓

Remarks

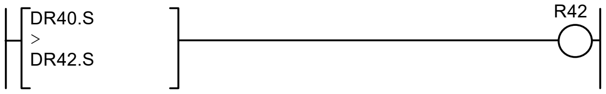
C means a constant
[ ]: number of steps (in Processing time column)

Function

- s1.Sand s2.S are compared as signed integers,
  when s1.S > s2.S, it is in continuity (ON).
  when s1.S ≤ s2.S, it is in discontinuity (OFF).
- When s1.S and s2.S are Word, −32,768 to 32,767 (decimal number), H8000 to H7FFF (hexadecimal number)

  When s1.S and s2.S are Double word, −2,147,483,648 to 2,147,483,647 (decimal number), H80000000 to H7FFFFFFF (hexadecimal number)

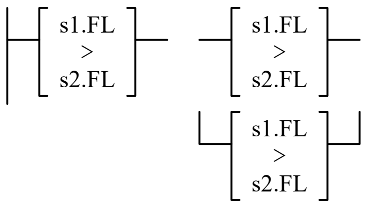
Program example



[Program description]

R42 turns ON when DR40.S > DR42.S and R42 turns OFF when DR40.S ≤ DR42.S.

Name	> Comparison box (Floating point)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
	(See Command processing time column)		●	●	●	●	●

Command processing time (μs)				
Average				
Condition	Processing Time			
	MVH (High Function)		MVL (Standard)	
	Double word		Double word	
LD (s1.FL > s2.FL) s1:I/O, s2: I/O	7.4	[6]	8.87	[6]
LD (s1.FL > s2.FL) s1:I/O, s2:C	5.3	[7]	6.47	[7]
LD (s1.FL > s2.FL) s1:C, s2:I/O	5.4	[7]	6.49	[7]
AND (s1.FL > s2.FL) s1:I/O, s2: I/O	7.1	[6]	8.72	[6]
AND (s1.FL > s2.FL) s1:I/O, s2:C	5.4	[7]	6.28	[7]
AND (s1.FL > s2.FL) s1:C, s2: I/O	5.4	[7]	6.3	[7]
OR (s1.FL > s2.FL) s1:I/O, s2: I/O	7.4	[7]	8.9	[7]
OR (s1.FL > s2.FL) s1:I/O, s2:C	4.8	[8]	6.52	[8]
OR (s1.FL > s2.FL) s1:C, s2: I/O	5.6	[8]	6.48	[8]

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s1.FL	Comparison number 1											✓	✓	✓	✓
s2.FL	Comparison number 2											✓	✓	✓	✓

Remarks
Floating decimal point is specified by Double word. C means a constant. Constant is 20 digit maximum. [ ]: number of steps (in Processing time column)

Function

- s1.FL and s2.FL are compared as floating points,

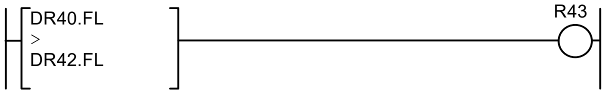
  when s1.FL > s2.FL, it is in continuity (ON).

  when s1.FL ≤ s2.FL, it is in discontinuity (OFF).

- s1.FL, s2.FL: -3.40282×10^{38} to 3.40282×10^{38} (decimal number),

  HFF7FFFFF to H80800000, H00800000 to H7F7FFFFF (hexadecimal number)

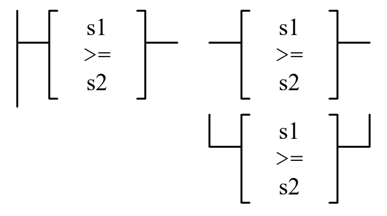
Program example



[Program description]

R43 turns ON when DR40.FL > DR42.FL and R43 turns OFF when DR40.FL ≤ DR42.FL.

Name	≥ Comparison box

<table>
<tr><th rowspan="2">Ladder format</th><th colspan="2">Number of steps</th><th colspan="5">Condition code</th></tr>
<tr><th rowspan="2">Condition</th><th rowspan="2">Steps</th><th>R7F4</th><th>R7F3</th><th>R7F2</th><th>R7F1</th><th>R7F0</th></tr>
<tr><td rowspan="2"> </td><th>DER</th><th>ERR</th><th>SD</th><th>V</th><th>C</th></tr>
<tr><td colspan="2">(See Command processing time column)</td><td>●</td><td>●</td><td>●</td><td>●</td><td>●</td></tr>
</table>

<table>
<tr><th colspan="9">Command processing time (μs)</th></tr>
<tr><th colspan="9">Average</th></tr>
<tr><th rowspan="3">Condition</th><th colspan="8">Processing Time</th></tr>
<tr><th colspan="4">MVH
(High Function)</th><th colspan="4">MVL
(Standard)</th></tr>
<tr><th colspan="2">Word</th><th colspan="2">Double word</th><th colspan="2">Word</th><th colspan="2">Double word</th></tr>
<tr><td>LD (s1 >= s2) s1:I/O, s2: I/O</td><td>1.5</td><td>[5]</td><td>6.4</td><td>[6]</td><td>1.6</td><td>[5]</td><td>7.74</td><td>[6]</td></tr>
<tr><td>LD (s1 >= s2) s1:I/O, s2:C</td><td>1.4</td><td>[5]</td><td>4.5</td><td>[7]</td><td>1.45</td><td>[5]</td><td>5.38</td><td>[7]</td></tr>
<tr><td>LD (s1 >= s2) s1:C, s2:I/O</td><td>1.4</td><td>[5]</td><td>4.5</td><td>[7]</td><td>1.43</td><td>[5]</td><td>5.42</td><td>[7]</td></tr>
<tr><td>AND (s1 >= s2) s1:I/O, s2: I/O</td><td>1.5</td><td>[5]</td><td>6.3</td><td>[6]</td><td>1.3</td><td>[5]</td><td>7.56</td><td>[6]</td></tr>
<tr><td>AND (s1 >= s2) s1:I/O, s2:C</td><td>1.3</td><td>[5]</td><td>4.6</td><td>[7]</td><td>1.24</td><td>[5]</td><td>5.22</td><td>[7]</td></tr>
<tr><td>AND (s1 >= s2) s1:C, s2: I/O</td><td>1.3</td><td>[5]</td><td>4.5</td><td>[7]</td><td>1.27</td><td>[5]</td><td>5.18</td><td>[7]</td></tr>
<tr><td>OR (s1 >= s2) s1:I/O, s2: I/O</td><td>1.7</td><td>[6]</td><td>6.7</td><td>[7]</td><td>1.59</td><td>[6]</td><td>7.76</td><td>[7]</td></tr>
<tr><td>OR (s1 >= s2) s1:I/O, s2:C</td><td>1.6</td><td>[6]</td><td>4.7</td><td>[8]</td><td>1.44</td><td>[6]</td><td>5.36</td><td>[8]</td></tr>
<tr><td>OR (s1 >= s2) s1:C, s2: I/O</td><td>1.5</td><td>[6]</td><td>4.7</td><td>[8]</td><td>1.46</td><td>[6]</td><td>5.36</td><td>[8]</td></tr>
</table>

<table>
<tr><th colspan="2" rowspan="2">Usable I/O</th><th colspan="5">Bit</th><th colspan="4">Word</th><th colspan="3">Double word</th><th rowspan="2">Constant</th></tr>
<tr><th>X</th><th>Y</th><th>R,M</th><th>TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,</th><th>WR, (.m)</th><th>WX</th><th>WY</th><th>WR, WM</th><th>TC</th><th>DX</th><th>DY</th><th>DR,DM</th></tr>
<tr><td>s1</td><td>Comparison number 1</td><td></td><td></td><td></td><td></td><td></td><td></td><td>✓</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td></tr>
<tr><td>s2</td><td>Comparison number 2</td><td></td><td></td><td></td><td></td><td></td><td></td><td>✓</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td></tr>
</table>

Remarks
C means a constant [ ]: number of steps (in Processing time column)

Function

- s1 and s2 are compared as unsigned integers,

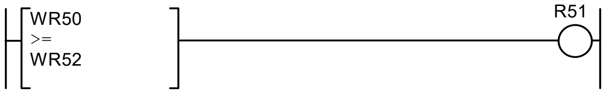
  when s1≥ s2, it is in continuity (ON).

  when s1 < s2, it is in discontinuity (OFF).

- When s1 and s2 are Word, 0 to 65,535 (decimal number),
  H0000 to HFFFF (hexadecimal number)

  When s1 and s2 are Double word, 0 to 4,294,967,295 (decimal number),
  H00000000 to HFFFFFFFF (hexadecimal number)

Program example

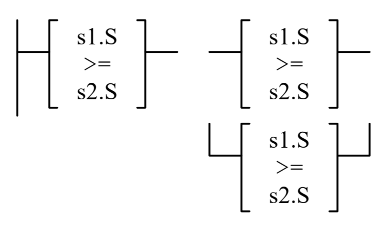


[Program description]

R51 turns ON when WR50 ≥ WR52 and R51 turns OFF when WR50 < WR52.

Name	≥ Comparison box (Signed integer)

Ladder format



Number of steps		Condition code				
Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
(See Command processing time column)		●	●	●	●	●

Command processing time (μs)

Average

Condition	Processing Time							
	MVH (High Function)				MVL (Standard)			
	Word		Double word		Word		Double word	
LD (s1.S >= s2.S) s1:I/O, s2: I/O	5.5	[5]	6.4	[6]	6.78	[5]	7.72	[6]
LD (s1.S >= s2.S) s1:I/O, s2:C	4.0	[5]	4.6	[7]	4.7	[5]	5.4	[7]
LD (s1.S >= s2.S) s1:C, s2:I/O	4.0	[5]	4.5	[7]	4.7	[5]	5.38	[7]
AND (s1.S >= s2.S) s1:I/O, s2: I/O	5.6	[5]	6.4	[6]	6.58	[5]	7.56	[6]
AND (s1.S >= s2.S) s1:I/O, s2:C	3.8	[5]	4.5	[7]	4.54	[5]	5.2	[7]
AND (s1.S >= s2.S) s1:C, s2: I/O	3.9	[5]	4.5	[7]	4.54	[5]	5.18	[7]
OR (s1.S >= s2.S) s1:I/O, s2: I/O	5.8	[6]	6.6	[7]	6.76	[6]	7.68	[7]
OR (s1.S >= s2.S) s1:I/O, s2:C	4.2	[6]	5.0	[8]	4.78	[6]	5.4	[8]
OR (s1.S >= s2.S) s1:C, s2: I/O	4.1	[6]	4.8	[8]	4.72	[6]	5.36	[8]

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s1.S	Comparison number 1							✓	✓	✓	✓	✓	✓	✓
s2.S	Comparison number 2							✓	✓	✓	✓	✓	✓	✓

Remarks

C means a constant

[ ]: number of steps (in Processing time column)

Function

- s1.S and s2.S are compared as signed integers,

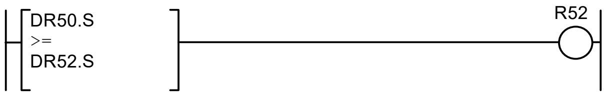
  when s1.S ≥ s2.S, it is in continuity (ON).

  when s1.S < s2.S, it is in discontinuity (OFF).

- When s1.S and s2.S are Word, −32,768 to 32,767 (decimal number),
  H8000 to H7FFF (hexadecimal number)

  When s1.S and s2.S are Double word, −2,147,483,648 to 2,147,483,647 (decimal number),
  H80000000 to H7FFFFFFF (hexadecimal number)

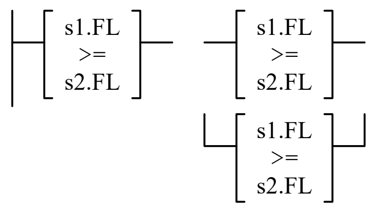
Program example



[Program description]

R52 turns ON when DR50.S ≥ DR52.S and R52 turns OFF when DR50.S < DR52.S.

Name	≥ Comparison box (Floating point)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
	(See Command processing time column)		●	●	●	●	●

Command processing time (μs)				
Average				
Condition	Processing Time			
	MVH (High Function)		MVL (Standard)	
	Double word		Double word	
LD (s1.FL >= s2.FL) s1:I/O, s2: I/O	7.3	[6]	8.84	[6]
LD (s1.FL >= s2.FL) s1:I/O, s2:C	5.3	[7]	6.48	[7]
LD (s1.FL >= s2.FL) s1:C, s2:I/O	5.4	[7]	6.48	[7]
AND (s1.FL >= s2.FL) s1:I/O, s2: I/O	7.2	[6]	8.64	[6]
AND (s1.FL >= s2.FL) s1:I/O, s2:C	5.3	[7]	6.2	[7]
AND (s1.FL >= s2.FL) s1:C, s2: I/O	5.1	[7]	6.28	[7]
OR (s1.FL >= s2.FL) s1:I/O, s2: I/O	7.5	[7]	8.84	[7]
OR (s1.FL >= s2.FL) s1:I/O, s2:C	5.7	[8]	6.48	[8]
OR (s1.FL >= s2.FL) s1:C, s2: I/O	5.9	[8]	6.48	[8]

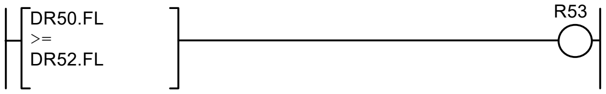
Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s1.FL	Comparison number 1										✓	✓	✓	✓
s2.FL	Comparison number 2										✓	✓	✓	✓

Remarks
Floating decimal point is specified by Double word. C means a constant. Constant is 20 digit maximum. [ ]: number of steps (in Processing time column)

Function

- s1.FL and s2.FL are compared as floating points,
  when s1.FL ≥ s2.FL, it is in continuity (ON).
  when s1.FL < s2.FL, it is in discontinuity (OFF).
- s1.FL, s2.FL: -3.40282×10^{38} to 3.40282×10^{38} (decimal number),
  HFF7FFFFF to H80800000, H00800000 to H7F7FFFFF (hexadecimal number)

Program example



[Program description]

R53 turns ON when DR50.FL ≥ DR52.FL and R53 turns OFF when DR50.FL < DR52.FL.

[2] Arithmetic commands

Name	Substitution statement

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
d = s d.m1 = s d.m1 = s.m2 d = s.m2	(See Command processing time column)		↕	●	●	●	●

Command processing time (μs)													
Maximum													
Condition		MVH(High function)						MVL(Standard)					
d	s	d, s:Bit		d, s:Word		d, s: Double word		d, s:Bit		d, s:Word		d, s: Double word	
I/O	I/O / C	1.01	[3]	0.87	[3]	4.38	[5]	1.0	[3]	0.83	[3]	5.15	[5]
I/O.m_1	I/O / C	6.6	[5]	—		—		8.23	[5]	—		—	
I/O	I/O.m_2	6.86	[5]	—		—		7.83	[5]	—		—	
I/O.m_1	I/O.m_2	6.58	[6]	—		—		7.83	[6]	—		—	
Array(W/C)	I/O / C	10.64	[5]	7	[5]	8.64	[6]	12.96	[5]	8.33	[5]	10.15	[6]
Array(DW)	I/O / C	10.96	[6]	7.44	[6]	8.96	[7]	13.24	[6]	8.63	[6]	10.43	[7]
I/O	Array(W/C)	12.32	[5]	8.68	[5]	10.39	[5]	15.12	[5]	10.33	[5]	12.29	[5]
I/O	Array(DW)	12.64	[6]	9	[6]	10.64	[6]	15.44	[6]	10.65	[6]	12.59	[6]
Array(W/C)	Array(W/C)	16.08	[6]	12.74	[6]	15.12	[6]	19.72	[6]	15.25	[6]	18.31	[6]
Array(DW)	Array(W/C)	16.44	[7]	13.10	[7]	15.62	[7]	20.0	[7]	15.59	[7]	18.63	[7]
Array(W/C)	Array(DW)	16.48	[7]	13.08	[7]	15.50	[7]	19.36	[7]	15.53	[7]	18.59	[7]
Array(DW)	Array(DW)	16.80	[8]	13.32	[8]	15.76	[7]	20.36	[8]	15.95	[8]	18.91	[7]

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Substitution destination		✓	✓				✓	✓	✓		✓	✓	
d.m_1	Substitution destination					✓								
s	Substitution source	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓
s.m_2	Substitution source					✓								
( )	Index value						✓	✓	✓					

Remarks
m_1 and m_2 are 0 to F. Bit in Word (.m) cannot be used in the array. W means Word I/O. DW means Double word I/O. C means a constant. [ ]: number of steps (in Processing time column)

Function

- d = s A content of s is substituted for d.
- d.m_1 = s A content of bit s is substituted for the m_1th bit of word data d.

  Example) WR0.4 = R0 A content of R0 is stored in the 4th bit of WR0.

  If WR0 = 0 and R0 = 1, WR0 = H0010.
- d = s.m_2 The m_2th bit of word data s is substituted for the bit d.

  Example) R10 = WR0.F A content of MSB (Most Significant Bit) of WR0 is stored in R10.

  If WR0 = HFFFF, R10 = 1.
- d.m_1 = s.m_2 The m_2th bit of word data s is substituted for the m_1th bit of word data d.

  Example) WR100.8 = WR10.0 A content of LSB (Less Significant Bit) of WR10 is stored in the 8th bit of WR100.

  If WR100 = H0F00 and WR10 = H0000, WR100 = H0E00.

- An array variable can be used for d and s.
- A constant can be used in the following range,

Word	0 to 65,535 (decimal number), H0000 to HFFFF (hexadecimal number)
Double word	0 to 4,294,967,295 (decimal number), H00000000 to HFFFFFFFF (hexadecimal number)

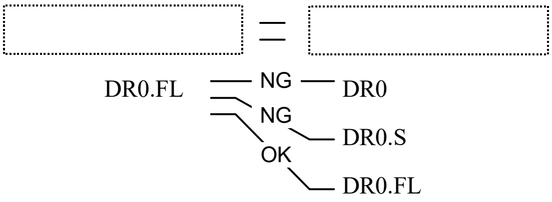
Cautionary notes

- The type is not converted in the substitution formula. Although it can be described that the internal output with an extension (.S, .FL) is substituted for the internal output with another extension by changing and/or deleting the extension temporarily, the16-bit data (or the 32-bit data) is substituted unchanged.

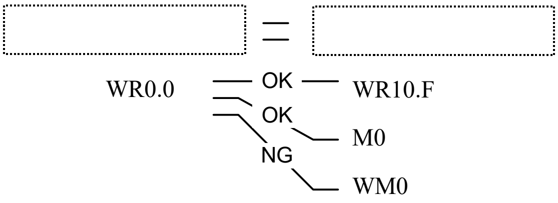
  Example) Case of DR0.S = −2000(HFFFFF830),

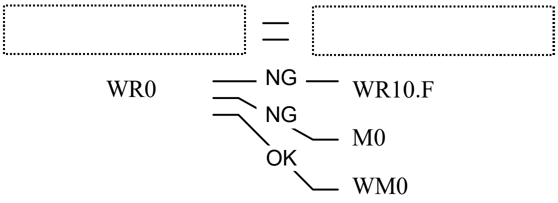
  DM0 = DR0

  (If DR0.S and DM0 are monitored in hexadecimal system, the result is both HFFFFF830. But if monitored in decimal system, DR0.S is displayed as -2000 and DM0 is displayed as 4294965296.)



- Please write so that the types of the left-hand side and the right-hand side are in agreement.





- When using the array variable, DER = 1 if it exceeds the maximum of usable I/O No., and DER = 0 if it is normal.
- When using a constant for the index of the array, the constant is valid from 0 to 65,535 (decimal number) and from H0000 to HFFFF (hexadecimal number).
- Only the word internal output can specify d.m_1 and s.m_2.
- d.m_1 and s.m_2 are the bit in the word internal output. Therefore if you use them for the substitution statement, please set the substitution destination or the substitution source to the bit.

Program example

```
R0 = M0
WM10 = WR100
DR200 = DY10
R1 = WR10.0
WR10.1 = R2
WR10.2 = WR20.A
```

[Program description]

At the rising edge of R0,

- a state of M0 is substituted for R0.
- a value of WR100 is substituted for WM10.
- a value of DY10 is substituted for DR200.

- a state of the 0th bit of WR10 is substituted for R1.
- a state of R2 is substituted for the 1st bit of WR10
- a state of the Ath bit (the 10th bit) of WR20 is substituted for the 2nd bit of WR10.

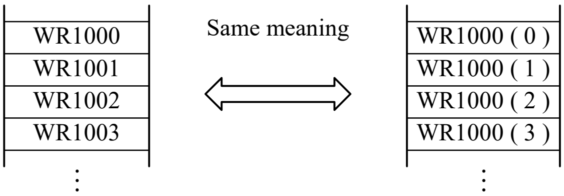
Reference Array specification for Substitution statement

Array is a means to change I/O specified by the index dynamically. This is convenient when setting a value, updating I/O address using FOR sentence.

Index value is specified by a constant and word I/O (WX, WY, WR, WM). In addition, the commands which the array variable can use are only a substitution statement, MOV, COPY, and I/O address conversion command.

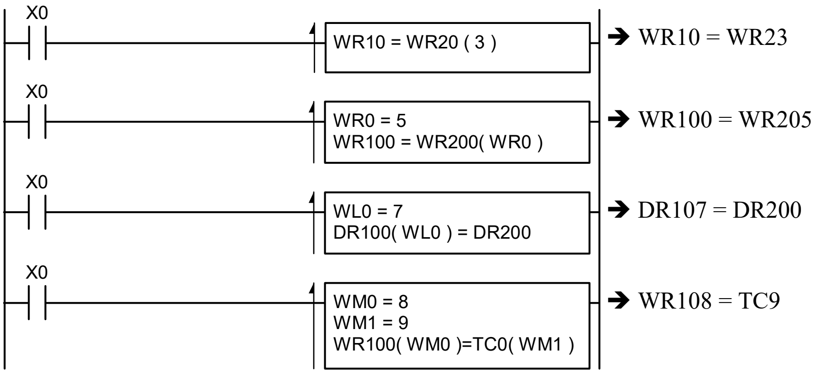
(1) Meaning of Array variable

Array variable is represented by the form of '☐ a(b)'. '☐' represents I/O type, and 'a' represents I/O address, and 'b' represents a constant or word I/O. And '☐ a' is called "I/O of array variable" and 'inner b of ( )' is called "Content of index".



(When the B part is a constant, '☐ a(b)' means '☐ a + b'.)

(2) Example



(3) Note

- The index is 0 or a positive integer. Negative cannot be specified.
- The array variable can be used only for the substitution statement. You cannot use as follows,

  WR10(WR20) = WR100 + 1

  R0 = WR10(WR20) < WR30
- The array of bit in Word cannot be used. You cannot use as follows,

  WR10.8 (WR20) = 1

  R0 = WR10.0 (WR20)

Name	Substitution statement (Signed integer)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
d.S = s.S	(See Command processing time column)		↕	●	●	●	●

Command processing time (μs)									
Maximum									
Condition		MVH(High function)				MVL(Standard)			
d.S	s.S	d.S, s.S:Word		d.S, s.S:Double word		d.S, s.S:Word		d.S, s.S:Double word	
I/O	I/O / C	0.85	[3]	4.5	[5]	0.83	[3]	5.15	[5]
Array(W/C)	I/O / C	7.02	[5]	8.68	[6]	8.33	[5]	10.19	[6]
Array(DW)	I/O / C	8.86	[6]	8.92	[7]	8.63	[6]	10.51	[7]
I/O	Array(W/C)	8.74	[5]	10.28	[5]	10.31	[5]	12.27	[5]
I/O	Array(DW)	8.96	[6]	10.60	[6]	10.57	[6]	12.59	[6]
Array(W/C)	Array(W/C)	12.69	[6]	15.22	[6]	15.25	[6]	18.27	[6]
Array(DW)	Array(W/C)	13.08	[7]	15.50	[7]	15.59	[7]	18.59	[7]
Array(W/C)	Array(DW)	13.08	[7]	15.36	[7]	15.57	[7]	18.55	[7]
Array(DW)	Array(DW)	13.38	[8]	15.8	[7]	15.91	[8]	18.95	[7]

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d.S	Substitution destination								✓	✓	✓		✓	✓	
s.S	Substitution source							✓	✓	✓	✓	✓	✓	✓	✓
( )	Index value							✓	✓	✓					✓

Remarks
W means Word I/O. DW means Double I/O. C means a constant. [ ]: number of steps (in Processing time column)

Function

- A content of s.S is substituted for d.S.
- An array variable can be used for d.S and s.S.
- A constant can be used in the following range.

Word	−32,768 to 32,767 (decimal number), H8000 to H7FFF (hexadecimal number)
Double word	−2,147,483,648 to 2,147,483,647 (decimal number), H80000000 to H7FFFFFFF (hexadecimal number)

- The combination of d.S and s.S are as follows.

d.S	s.S
Word	Word
Double word	Double word

Cautionary notes

- When using the array variable, DER = 1 if it exceeds the maximum of usable I/O No., and DER = 0 if it is normal.
- The type is not converted tin the substitution statement. Although it can be described that the internal output with an extension (.S, .FL) is substituted for the internal output with another extension by changing and/or deleting the extension temporarily, the16-bit data (or the 32-bit data) is substituted unchanged.

  Example) Case of DR0.FL = −259（HC3818000),

  DM0.S = DR0.S

  (If DR0.FL and DM0.S are monitored in hexadecimal system, the result is both HC3818000. But if monitored in decimal system, DR0.FL is displayed as –259 and DM0.S is displayed as –1014923264.)
- When using a constant for the index of the array, the constant is valid from 0 to 65,535 (decimal number) and from H0000 to HFFFF (hexadecimal number).

Name	Substitution (Floating point)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
d.FL = s.FL	(See Command processing time column)		↕	●	●	●	●

Command processing time (μs)					
Maximum					
Condition		MVH(High function)		MVL(Standard)	
d.FL	s.FL	d.FL, s.FL:Double word		d.FL, s.FL:Double word	
I/O	I/O / C	4.48	[5]	5.15	[5]
Array(W/C)	I/O / C	8.56	[6]	10.15	[6]
Array(DW)	I/O / C	8.96	[7]	10.43	[7]
I/O	Array(W/C)	10.28	[5]	12.27	[5]
I/O	Array(DW)	10.60	[6]	12.59	[6]
Array(W/C)	Array(W/C)	15.22	[6]	18.31	[6]
Array(DW)	Array(W/C)	15.50	[7]	18.59	[7]
Array(W/C)	Array(DW)	15.36	[7]	18.55	[7]
Array(DW)	Array(DW)	15.88	[7]	18.95	[7]

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)		WX	WY	WR, WM	TC	DX	DY	DR,DM	
d.FL	Substitution destination												✓	✓	
s.FL	Substitution source											✓	✓	✓	✓
( )	Index value							✓	✓	✓					✓

Remarks
Floating decimal point is specified by Double word W means Word I/O. DW means Double I/O. C means a constant. Constant is 20 digit maximum. [ ]: number of steps (in Processing time column)

Function

- A content of s.FL is substituted for d.FL.
- An array variable can be used for d.FL and s.FL.
- A constant can be use in the following range,

d.FL and s.FL -3.40282×10^{38} to 3.40282×10^{38} (decimal number),

HFF7FFFFF to H80800000, H00800000 to H7F7FFFFF (hexadecimal number)

- The combination of d.FL and s.FL are as follows.

d.FL	s.FL
Double word	Double word

Cautionary notes

- When using the array variable, DER = 1 if it exceeds the maximum of usable I/O No., and DER = 0 if it is normal.
- The type is not converted in the substitution statement. Although it can be described that the internal output with an extension (.S, .FL) is substituted for the internal output with another extension by changing and/or deleting the extension temporarily, the16-bit data (or the 32-bit data) is substituted unchanged.

  Example) Case of DR0.S = 1073741824 (H40000000),

  DM0.FL = DR0.FL

  (If DR0.S and DM0.FL are monitored in hexadecimal system, the result is both H40000000. But if monitored in decimal system, DR0.S is displayed as 1073741824 and DM0.FL is displayed as 2.)
- When using a constant for the index of the array, the constant is valid from 0 to 65,535 (decimal number), and from H0000 to HFFFF (hexadecimal number).

Name	Binary addition

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
d = s1 + s2	Word	4	●	●	●	↕	↕
	Double word	7					

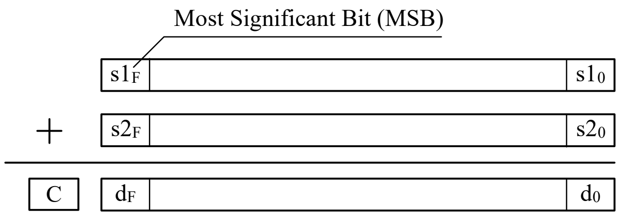
Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
Word	1.56	1.57	—	—	—
Double word	5.73	6.42	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Substitution destination							✓	✓	✓		✓	✓	
s1	Augend						✓	✓	✓	✓	✓	✓	✓	✓
s2	Addend						✓	✓	✓	✓	✓	✓	✓	✓

Remarks

Function

- s1 and s2 are added as binary data, and the result is substituted for d in binary data.



- If the C flag (Carry: R7F0) is within the following range, it is reset to 0 and it is set to 1 if not.

  when the operation result is Word, the range is 0 to 65,535 (decimal number) and H0000 to HFFFF (hexadecimal number).

  when the operation result is Double word, the range is 0 to 4,294,967,295 (decimal number) and H00000000 to HFFFFFFFF (hexadecimal number)

$$C = s1_F \cdot s2_F + s1_F \cdot \overline{d_F} + s2_F \cdot \overline{d_F}$$

- V flag (Overflow: R7F1) is set to 1 if the operation results are meaningless as signed binary data, and it is reset to 0 if it is meaningful. (See the following table)

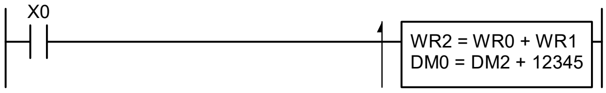
s1	s2	d	V
Positive	Positive	Positive	0
Positive	Positive	Negative	1
Positive	Negative	Positive / Negative	0
Negative	Positive	Negative / Positive	0
Negative	Negative	Positive	1
Negative	Negative	Negative	0

$$V = s1_F \cdot s2_F \cdot \overline{d_F} + \overline{s1_F} \cdot \overline{s2_F} \cdot d_F$$

- The combination of d and s are as follows.

d	s1	s2
Word	Word	Word
Double word	Double word	Double word

Program example



[Program description]

At the rising edge of X0,

- The value of WR1 is added to the value of WR0, and the result is substituted for WR2.
- 12345 is added to the value of DM2 and the result is substituted for DM0.

Name	Binary addition (Signed integer)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
d.S = s1.S + s2.S	Word	4	●	●	●	↕	↕
	Double word	7					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
Word	1.57	1.54	—	—	—
Double word	5.66	6.41	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)		WX	WY	WR, WM	TC	DX	DY	DR,DM	
d.S	Substitution destination								✓	✓	✓		✓	✓	
s1.S	Augend							✓	✓	✓	✓	✓	✓	✓	✓
s2.S	Addend							✓	✓	✓	✓	✓	✓	✓	✓

Remarks

Function

- s1.S and s2.S are added as signed binary data, and the result is substituted for d.S in binary data.
- If the C flag (Carry: R7F0) is within the following range, it is reset to 0 and it is set to 1 if not.

  when the operation result is Word, the range is −32,768 to 32,767 (decimal number) and H8000 to H7FFF (hexadecimal number).

  when the operation result is Double word, the range is −2,147,483,648 to 2,147,483,647 (decimal number) and H80000000 to H7FFFFFFF (hexadecimal number).

$$C = s1_F \cdot s2_F + s1_F \cdot \overline{d_F} + s2_F \cdot \overline{d_F}$$

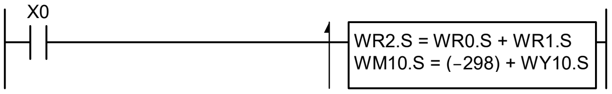
- A control of V flag (Overflow: R7F1) is same as the binary addition (d = s1 + s2).
- The combination of d and s are as follows.

D	s1	s2
Word	Word	Word
Double word	Double word	Double word

Cautionary notes

d, s1, and s2 need an extension “.S”.

Program example



[Program description]

At the rising edge of X0,

- The value of WR1.S is added to the value of WR0.S, and the result is substituted for WR2.S.
- The value of WY10.S is added to -298 and the result is substituted for WM10.S.
  (298 is subtracted from WY10.S and the result is substituted for WM10.S.)

Name	BCD addition

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
d = s1 B+ s2	Word	5	↕	●	●	●	↕
	Double word	7					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
Word	7.44	8.91	—	—	—
Double word	9.8	11.53	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Substitution destination							✓	✓	✓		✓	✓	
s1	Augend						✓	✓	✓	✓	✓	✓	✓	✓
s2	Addend						✓	✓	✓	✓	✓	✓	✓	✓

Remarks

Function

- s1 and s2 are added as BCD data, and the result is substituted for d in BCD data.
- If a carry is in the operation result, the C flag (Carry: R7F0) is set to 1, and if no carry is in, it is reset to 0.
- The DER flag (Data error: R7F4) is set to 1 if a content of s1 or s2 is not correct as BCD data. In this case, the operation is not performed and C holds the preceding state and the output to d is not performed.

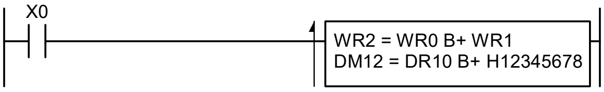
  If the calculation result is correct, the DER flag is reset to 0 and the operation result is output to d.
- s1 and s2 are valid in the following range,

when it is Word,	H0000 to 9999 (BCD)
when it is Double word,	H00000000 to 99999999 (BCD)

- The combination of d and s are as follows.

d	s1	s2
Word	Word	Word
Double word	Double word	Double word

Program example



WR2 = WR0 B+ WR1
DM12 = DR10 B+ H12345678

[Program description]

At the rising edge of X0,

- The value of WR1 is added to the value of WR0 and the result is substituted for WR2 in BCD data.
- H12345678 is added to the value of DR10 and the result is substituted for DM12.

Name	Floating point addition

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
d.FL = s1.FL + s2.FL	—	7	↕	●	●	●	●

Command processing time (μs)				
Average				
Condition	Processing time			
	MVH (High function)		MVL (Standard)	
	Double word		Double word	
—	10.66	[8]	12.86	[8]

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d.FL	Substitution destination												✓	✓	
s1.FL	Augend											✓	✓	✓	✓
s2.FL	Addend											✓	✓	✓	✓

Remarks
Floating decimal point is specified by Double word. Constant is 20 digit maximum.

Function

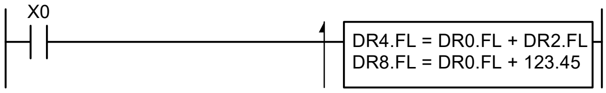
- s1.FL and s2.FL are added as floating point data, and the result is substituted for d.FL in floating point data.
- The DER flag (Data error: R7F4) is set to 1 if a content of s1.FL or s2.FL is not correct as floating point. In this case, the operation is not performed and the output to d.FL is not performed.

  If the calculation result is correct, the DER flag is reset to 0 and the operation result is output to d.FL.

Cautionary notes

- If the operation result is outside the range from −1e+37 to 1e+37, the operation is not performed because of DER = 1.
- d, s1 and s2 need an extension “.FL“.

Program example



[Program description]

At the rising edge of X0,

- The value of DR2.FL is added to the value of DR0.FL and the result is substituted for DR4.FL.
- 123.45 is added to the value of DR0.FL and the result is substituted for DR8.FL.

Name	Binary subtraction

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
d = s1 – s2	Word	4	●	●	●	↕	↕
	Double word	7					

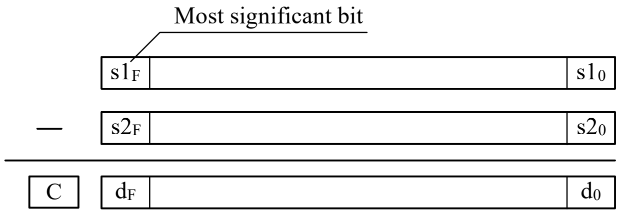
Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
Word	1.57	1.54	—	—	—
Double word	5.73	6.42	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, TDN, SS, WDT, MS, TMR, CU, RCU, CT	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Substitution destination							✓	✓	✓		✓	✓	
s1	Minuend						✓	✓	✓	✓	✓	✓	✓	✓
s2	Subtrahend						✓	✓	✓	✓	✓	✓	✓	✓

Remarks

Function

- s2 is subtracted from s1 as binary data in s1 and s2. And the result is substituted for d in binary data.



- The C flag (Carry: R7F0) is set to 1 if a carry-down occurs in the operation result and it is reset to 0 if a carry-down does not occur.

  When s1 < s2, the operation result and C are set to 1.

  When s1 ≥ s2, the operation result and C are set to 0.

$$C=\overline{s1_F}\cdot s2_F+\overline{s1_F}\cdot d_F+s2_F\cdot d_F$$

- The V flag (Overflow: R7F1) is set to 1 if the operation result is meaningless as signed binary data, and it is reset to 0 if it is meaningful. (See the following table)

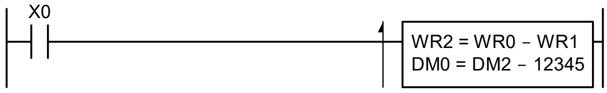
s1	s2	d	V
Positive	Positive	Positive / Negative	0
Negative	Negative	Positive / Negative	0
Positive	Negative	Positive	0
Positive	Negative	Negative	1
Negative	Positive	Positive	1
Negative	Positive	Negative	0

$$V=\overline{s1_F}\cdot s2_F\cdot d_F+\overline{s1_F}\cdot\overline{s2_F}\cdot d_F$$

- The combination of d and s are as follows.

D	s1	s2
Word	Word	Word
Double word	Double word	Double word

Program example



[Program description]

At the rising edge of X0,

- The value of WR1 is subtracted from the value of WR0 and the result is substituted for WR2.
- 12345 is subtracted from the value of DM2 and the result is substituted for DM0.

Name	Binary subtraction (Signed integer)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
d.S = s1.S − s2.S	Word	4	●	●	●	↕	↕
	Double word	7					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
Word	1.61	1.54	—	—	—
Double word	5.66	6.41	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d.S	Substitution destination							✓	✓	✓		✓	✓	
s1.S	Minuend						✓	✓	✓	✓	✓	✓	✓	✓
s2.S	Subtrahend						✓	✓	✓	✓	✓	✓	✓	✓

Remarks

Function

- s2.S is subtracted from s1.S as signed binary data in s1.S and s2.S and the result is substituted for d.S in signed binary data.
- The C flag (Carry: R7F0) is set to 1 if a carry-down occurs in the operation result, and it is reset to 0 if a carry-down does not occur.

  When $s1.S < s2.S$, the operation result and C are set to 1.

  When $s1.S \geq s2.S$, the operation result and C are set to 0.

$$C = \overline{s1_F} \cdot s2_F + \overline{s1_F} \cdot d_F + s2_F \cdot d_F$$

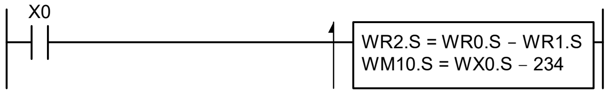
- The control of the V flag (Overflow: R7F1) is same as the binary subtraction ($d = s1 - s2$).
- The combination of d and s are as follows.

d	s1	s2
Word	Word	Word
Double word	Double word	Double word

Cautionary notes

d, s1, and s2 need an extension “.S”.

Program example



```
WR2.S = WR0.S − WR1.S
WM10.S = WX0.S − 234
```

[Program description]

At the rising edge of X0,

- The value of WR1.S is subtracted from the value of WR0.S and the result is substituted for WR2.S.
- 234 is subtracted from the value of WX0.S and the result is substituted for WM10.S.

Name	BCD subtraction

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
d = s1 B− s2	Word	5	↕	●	●	●	↕
	Double word	7					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
Word	7.24	8.73	—	—	—
Double word	9.76	11.43	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Substitution destination								✓	✓	✓		✓	✓	
s1	Minuend							✓	✓	✓	✓	✓	✓	✓	✓
s2	Subtrahend							✓	✓	✓	✓	✓	✓	✓	✓

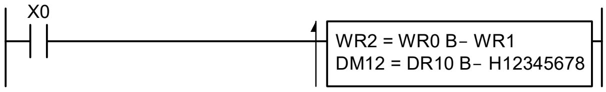
Remarks

Function

- s2 is subtracted from s1 as BCD data in s1 and s2, and the result is substituted for ‘d’ in BCD data.
- The C flag (Carry: R7F0) is set to 1 if a carry-down occurs in the operation result, and it is reset to 0 if a carry-down does not occur.
- The DER flag (Data error: R7F4) is set to 1 if a content of s1 or s2 is not correct as BCD data. In this case, the operation is not performed and C holds the preceding state and the output to ‘d’ is not performed.
  If the calculation result is correct, the DER flag is reset to 0 and the operation result is output to d.
- s1 and s2 are valid in the following range,
  when it is Word, the range is H0000 to 9999 (BCD)
  when it is Double word, the range is H00000000 to 99999999 (BCD)
- The combination of d and s are as follows.

d	s1	s2
Word	Word	Word
Double word	Double word	Double word

Program example



[Program description]

At the rising edge of X0,

- The value of WR1 is subtracted from the value of WR0 and the result is substituted for WR2 in BCD data.
- H12345678 is subtracted from the value of DR10 and the result is substituted for DM12.

Name	Floating point subtraction

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
d.FL = s1.FL − s2.FL	—	7	↕	●	●	●	●

Command processing time (μs)				
Average				
Condition	処理 Time			
	MVH (High function)		MVL (Standard)	
	Double word		Double word	
—	10.64	[8]	12.91	[8]

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d.FL	Substitution destination											✓	✓	
s1.FL	Minuend										✓	✓	✓	✓
s2.FL	Subtrahend										✓	✓	✓	✓

Remarks
Floating decimal point is specified by Double word. Constant is 20 digit maximum.

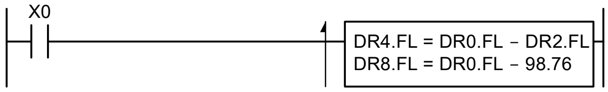
Function

- s2.FL is subtracted from s1.FL as floating point data in s1.FL and s2.FL, and the result is substituted for d.FL in floating point data.
- The DER flag (Data error: R7F4) is set to 1 if a content of s1.FL or s2.FL is not correct as floating point data. In this case, the operation is not performed and the output to d.FL is not performed.
  If the calculation result is correct, the DER flag is reset to 0 and the operation result is output to d.FL.

Cautionary notes

- If the operation result is outside the range from −1e+37 to 1e+37, the operation is not performed because of DER = 1.
- d, s1, and s2 need an extension ".FL".

Program example



[Program description]

At the rising edge of X0,

- The value of DR2.FL is subtracted from the value of DR0.FL and the result is substituted for DR4.FL.
- 98.76 is subtracted from the value of DR0.FL and the result is substituted for DR8.FL.

Name	Binary multiplication

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
d = s1 * s2	Word	5	●	●	●	●	●
	Double word	7					

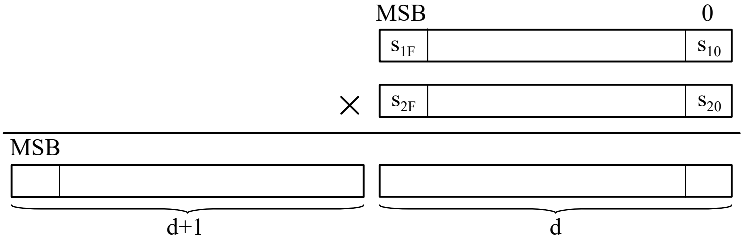
Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
Word	4.5	5.19	—	—	—
Double word	7.7	9.15	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Substitution destination							✓	✓	✓		✓	✓	
s1	Multiplicand						✓	✓	✓	✓	✓	✓	✓	✓
s2	Multiplier						✓	✓	✓	✓	✓	✓	✓	✓

Remarks

Function

- s1 and s2 are multiplied together as binary data, and the result is substituted for d+1 (upper) and d (lower) in binary data.



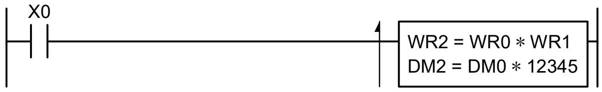
- The combination of d and s are as follows.

d	s1	s2	Remarks
Word	Word	Word	Calculation result is stored in 2 words.
Double word	Double word	Double word	Calculation result is stored in 2 double words.

Cautionary notes

- The operation result is substituted for d and d+1. Please pay attention when using Word or Double word of d+1 for other purpose.
- If d+1 exceeds the I/O range, the circuit cannot be input.

Program example



[Program description]

At the rising edge of X0,

- The value of WR0 and the value of WR1 are multiplied together, and the result is substituted for WR2 and WR3 (DR2).
- The value of DM0 and 12345 are multiplies together, and the result is substituted for DM2 and DM4.

Name	Binary multiplication (Signed integer)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
d.S = s1.S ＊ s2.S	Word	5	●	●	●	●	●
	Double word	7					

Command processing time (μs)

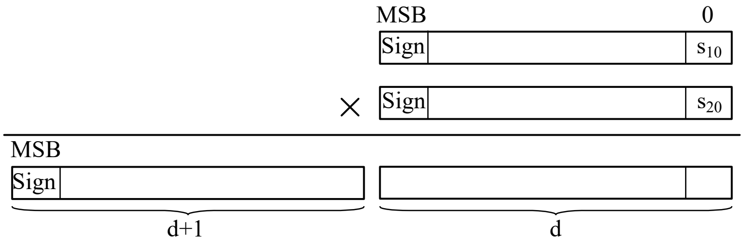
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
Word	4.5	5.19	—	—	—
Double word	7.7	9.17	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d.S	Substitution destination								✓	✓	✓		✓	✓	
s1.S	Multiplicand							✓	✓	✓	✓	✓	✓	✓	✓
s2.S	Multiplier							✓	✓	✓	✓	✓	✓	✓	✓

Remarks

Function

- s1.S and s2.S are multiplies together as signed binary data, and the result is substituted for d+1.S (upper) and d.S (lower) in signed binary data.



- Sign of the operation result is stored in MSB of d+1.
- s1.S and s2.S are valid in the following range.

  when it is Word , the range is −32,768 to 32,767 (decimal number) and H8000 to H7FFF (hexadecimal number).

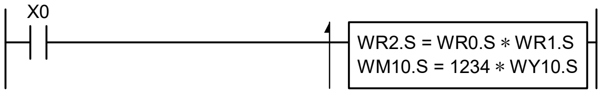
  when it is Double word, the range is −2,147,483,648 to 2,147,483,647 (decimal number) and H80000000 to H7FFFFFFF (hexadecimal number).
- The combination of d and s are as follows.

d	s1	s2	Remarks
Word	Word	Word	Calculation is stored in 2 words.
Double word	Double word	Double word	Calculation is stored in 2 double words.

Cautionary notes

- d, s1, and s2 need an extension ".S".
- The operation result is substituted for d and d+1. Please pay attention when using Word or Double word of d+1 for other purpose.
- If d+1 exceeds the I/O range, the circuit cannot be input.

Program example



[Program description]

At the rising edge of X0,

- The value of WR0.S and the value of WR1.S are multiplies together, and the result is substituted for WR2 and WR3 (DR2.S).
- 1234 and the value of WY10.S are multiplies together, and the result is substituted for WM10 and WM12 (DM10.S).

Name	BCD multiplication

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
d = s1 B* s2	Word	5	↕	●	●	●	●
	Double word	7					

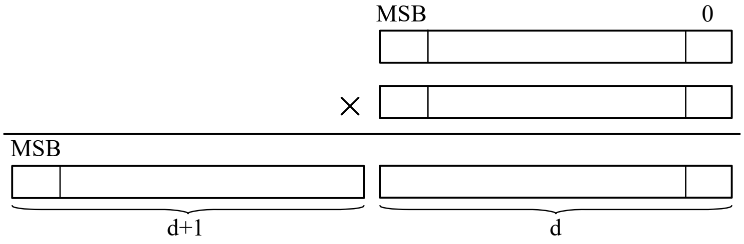
Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
Word	8.32	10.07	—	—	—
Double word	26.42	31.25	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Substitution destination							✓	✓	✓		✓	✓	
s1	Multiplicand						✓	✓	✓	✓	✓	✓	✓	✓
s2	Multiplier						✓	✓	✓	✓	✓	✓	✓	✓

Remarks

Function

- s1 and s2 are multiplies together as BCD data, and the result is substituted for d+1 (upper) and d (lower) in BCD data.



- If a content of s1 or s2 is not correct as BCD data, the DER flag (Data error: R7F4) is set to 1 and the operation is not performed. If s1 and s2 are correct as BCD data, it is reset to 0 and the operation result is output to d.
- s1 and s2 are valid in the following range,

  when it is Word, the range is H0000 to 9999 (BCD).

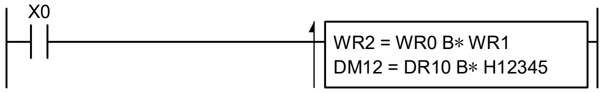
  when it is Double word, the range is H00000000 to 99999999 (BCD).
- The combination of d and s are as follows.

d	s1	s2	Remarks
Word	Word	Word	Calculation is stored in 2 words.
Double word	Double word	Double word	Calculation is stored in 2 double words.

Cautionary notes

- The operation result is substituted for d and d+1. Please pay attention when using Word or Double word of d+1 for other purpose.
- If d+1 exceeds the I/O range, the circuit cannot be input.

Program example



[Program description]

At the rising edge of X0,

- The value of WR0 and the value of WR1 are multiplies together, and the result is substituted for WR2 and WR3 (DR2) in BCD data.
- The value of DR10 and 12345 are multiplies together, and the result is substituted for DM12 and DM14 in BCD data.

Name	Floating point multiplication

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
d.FL = s1.FL ＊ s2.FL	—	7	↕	●	●	●	●

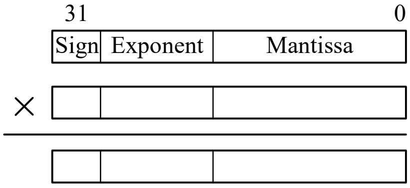
Command processing time (μs)				
Average				
Condition	Processing Time			
	MVH (High function)		MVL (Standard)	
	Double word		Double word	
—	10.7	[8]	12.91	[8]

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d.FL	Substitution destination											✓	✓	
s1.FL	Multiplicand										✓	✓	✓	✓
s2.FL	Multiplier										✓	✓	✓	✓

Remarks
Floating decimal point is specified by Double word. Constant is 20 digit maximum.

Function

- s1.FL and s2.FL are multiplied together as floating point data, and the result is substituted for d.FL in floating points data.

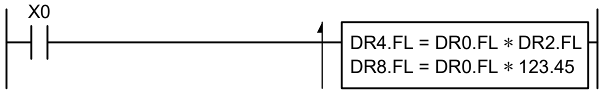


- The DER flag (Data error: R7F4) is set to 1 if a content of s1.FL or s2.FL is not correct as floating point data. In this case, the operation is not performed and the output to d.FL is not performed.
  If the calculation result is correct, it is reset to 0 and the operation is output to d.FL.

Cautionary notes

- If the operation result is outside the range from −1e+37 to 1e+37, the operation is not performed because of DER = 1.
- d, s1, and s2 need an extension ".FL".

Program example



[Program description]

At the rising edge of X0,

- The value of DR0.FL and the value of DR2.FL are multiplied together, and the result is substituted for DR4.FL.
- The value of DR0.FL and 123.45 are multiplied together, and the result is substituted for DR8.FL.

Name	Binary division

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
d = s1 / s2	Word	5	↕	●	●	●	●
	Double word	7					

Command processing time (μs)

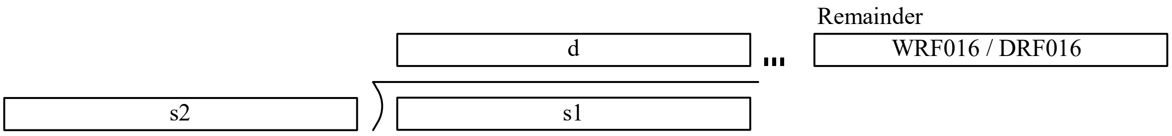
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
Word	3.9	4.41	—	—	—
Double word	4.7	5.28	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Substitution destination							✓	✓	✓		✓	✓	
s1	Dividend						✓	✓	✓	✓	✓	✓	✓	✓
s2	Divisor						✓	✓	✓	✓	✓	✓	✓	✓

Remarks

Function

- s1 is divided by s2 as binary data in s1 and s2. And the result is substituted for d in binary data.
  The remainder (s1 mod s2) is stored in the special internal output WRF016 at Word and DRF016 at Double word.
- The DER flag (Data error: R7F4) is set to 1 if s2 is 0. The operation is not performed.
  If s2≠0, it is reset to 0 and the operation is performed.



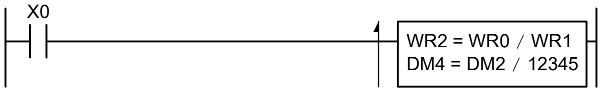
- The combination of d and s are as follows.

d	s1	s2
Word	Word	Word
Double word	Double word	Double word

Cautionary notes

- If s1 and s2 are Word, the special internal output WRF017 in which the remainder of division is stored is not used. (The value before operation remains unchanged.)

Program example



[Program description]

At the rising edge of X0,

- The value of WR0 is divided by the value of WR1, and the result is substituted for WR2. The remainder is substituted for the special internal output WRF016.
- The value of DM2 is divided by 12345, and the result is substituted for DM4. The remainder is substituted for the special internal output DRF016.

Name	Binary division (Signed integer)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
d.S = s1.S / s2.S	Word	5	↕	●	●	↕	●
	Double word	7					

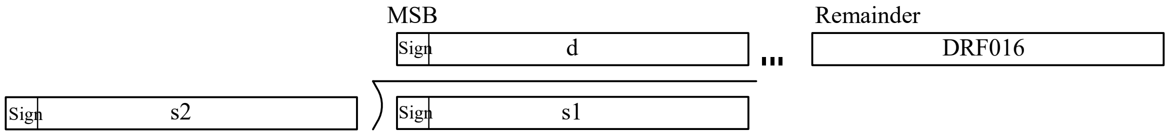
Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
Word	3.8	4.41	—	—	—
Double word	5.6	6.44	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d.S	Substitution destination							✓	✓	✓		✓	✓	
s1.S	Dividend						✓	✓	✓	✓	✓	✓	✓	✓
s2.S	Divisor						✓	✓	✓	✓	✓	✓	✓	✓

Remarks

Function

- s1 is divided by s2 as signed binary data in s1 and s2. And the result is substituted for d in signed binary data. The remainder (s1 mod s2) is stored in the special internal output WRF016 at Word operation and DRF016 at Double word operation. (Sign is stored in MSB.)



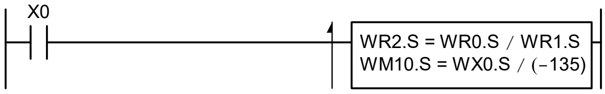
- The DER flag (Data error: R7F4) is set to 1 if s2 = 0. The operation is not performed.
  If s2 ≠ 0, it is reset to 0 and the operation result is performed.
- V (Overflow: R7F0) is set to 1 if the quotient is positive and also exceeds H7FFF or H7FFFFFFF (hexadecimal number). In all other cases, it is reset to 0.
- The range of s1.S and s2.S are as follows,
  when it is Word , the range is −32,768 to 32,767 (decimal number) and H8000 to H7FFF (hexadecimal number).
  when it is Double word, the range is −2,147,483,648 to 2,147,483,647 (decimal number) and H80000000 to H7FFFFFFF (hexadecimal number).
- The combination of d and s are as follows.

d	s1	s2
Word	Word	Word
Double word	Double word	Double word

Cautionary notes

- d, s1, and s2 need an extension ".S".
- If s1 and s2 are Word, the special internal output WRF017 in which a remainder of division is stored is not used. (The value before operation remains unchanged.)

Program example



[Program description]

At the rising edge of X0,

- The value of WR0.S is divided by the value of WR1.S, and the result is substituted for WR2.S. A remainder is substituted for the special internal output WRF016 in signed binary data.
- The value of WX0.S is divided by −135, and the result is substituted for WM10.S. A remainder is substituted for the special internal output WRF016 in signed binary data.

Name	BCD division

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
d = s1 B/ s2	Word	5	↕	●	●	●	●
	Double word	7					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
Word	4.72	5.74	—	—	—
Double word	6.64	7.82	—	—	—

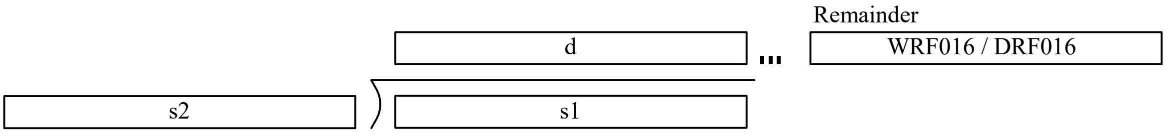
Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Substitution destination							✓	✓	✓		✓	✓	
s1	Dividend						✓	✓	✓	✓	✓	✓	✓	✓
s2	Divisor						✓	✓	✓	✓	✓	✓	✓	✓

Remarks

Function

- s1 is divided by s2 as BCD data in s1 and s2. And the result is substituted for d in BCD data.

  A remainder (s1 mod s2) is stored in the special internal output WRF016 at Word and DRF016 at Double word in BCD data



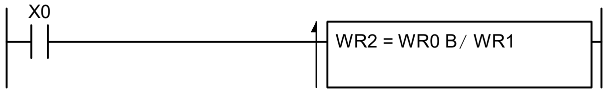
- If a content of s1 or s2 is not correct, or if s2 = 0, the DER flag (Data error: R7F4) is set to 1 and the operation is not performed. If s1 and s2 are correct as BCD data and also s2 ≠ 0, it is reset to 0 and the operation is output to d.
- s1 and s2 are valid in the following ranges,

  when it is Word, the range is H0000 to 9999 (BCD).

  when it is Double word, the range is H00000000 to 99999999 (BCD).
- The combination of d and s are as follows.

d	s1	s2
Word	Word	Word
Double word	Doube word	Double word

Program example



[Program description]

The value of WR0 is divided by the value of WR1 at the rising edge of X0, and the result is substituted for WR2 in BCD data. A remainder is substituted for the special internal output WRF016 in BCD data.

Name	Floating point division

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
d.FL = s1.FL / s2.FL	—	7	↕	●	●	●	●

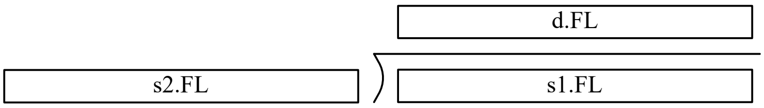
Command processing time (μs)				
Average				
Condition	Processing Time			
	MVH (High function)		MVL (Standard)	
	Double word		Double word	
—	5.6	[8]	6.44	[8]

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT / TDN, WDT, TMR, RCU	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d.FL	Substitution destination											✓	✓	
s1.FL	Dividend										✓	✓	✓	✓
s2.FL	Divisor										✓	✓	✓	✓

Remarks
Floating decimal point is specified by Double word. Constant is 20 digit maximum.

Function

- s1.FL is divided by s2.FL as floating point data in s1.FL and s2.FL. And the result is substituted for d.FL in floating decimal point data.

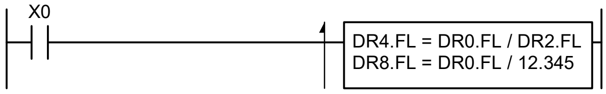


- The DER flag (Data error: R7F4) is set to 1 if a content of s1.FL or s2.FLis not correct as floating point data, or if s2 = 0. In this case, the operation is not performed and the output to d.FL is not performed.
  If s1.FL and s2.FL are correct data and also s2 ≠ 0, it is reset to 0 and the operation result is output to d.FL.

Cautionary notes

- If the operation result is outside the range from −1e+37 to 1e+37, the operation is not performed because of DER = 1.
- d, s1, and s2 need an extension ".FL".
- There is no remainder in floating point division.

Program example



[Program description]

At the rising edge of X0,

- The value of DR0.FL is divided by the value of DR2.FL, and the result is substituted for DR4.FL.
- The value of DR0.FL is divided by 12.345, and the result is substituted for DR8.FL.

Name	Logical sum (OR)

Ladder format

d = s1 OR s2

d = $s1.m_1$ OR s2
d = s1 OR $s2.m_2$
d = $s1.m_1$ OR $s2.m_2$
$d.m_0$ = s1 OR s2
$d.m_0$ = $s1.m_1$ OR s2
$d.m_0$ = s1 OR $s2.m_2$
$d.m_0$ = $s1.m_1$ OR $s2.m_2$

Number of steps				Condition code				
Condition			Steps ( ) indicates DW	R7F4	R7F3	R7F2	R7F1	R7F0
d	s1	s2		DER	ERR	SD	V	C
I/O	I/O	I/O	3 (6)	●	●	●	●	●
I/O	I/O.m	I/O	6 (–)					
I/O	I/O	I/O.m	6 (–)					
I/O	I/O.m	I/O.m	7 (–)					
I/O.m	I/O	I/O	6 (–)					
I/O.m	I/O.m	I/O	7 (–)					
I/O.m	I/O	I/O.m	7 (–)					
I/O.m	I/O.m	I/O.m	8 (–)					

Command processing time (μs)

Average					Maximum				
Condition			Time ( ) indicates DW		Condition			Time ( ) indicates DW	
d	s1	s2	MVH (High function)	MVL (Standard)	d	s1	s2	MVH (High function)	MVL (Standard)
I/O	I/O	I/O	0.96 (5.52)	0.96(6.47)	I/O	I/O	I/O	–	–
I/O	I/O.m	I/O	6.92	9.55	I/O	I/O.m	I/O	–	–
I/O	I/O	I/O.m	6.92	9.55	I/O	I/O	I/O.m	–	–
I/O	I/O.m	I/O.m	6.92	9.55	I/O	I/O.m	I/O.m	–	–
I/O.m	I/O	I/O	5.48	8.95	I/O.m	I/O	I/O	–	–
I/O.m	I/O.m	I/O	7.64	9.14	I/O.m	I/O.m	I/O	–	–
I/O.m	I/O	I/O.m	7.64	9.14	I/O.m	I/O	I/O.m	–	–
I/O.m	I/O.m	I/O.m	7.64	9.14	I/O.m	I/O.m	I/O.m	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Substitution destination		✓	✓			✓	✓	✓	✓		✓	✓	
$d.m_0$	Substitution destination					✓								
s1	Comparand	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓
$s1.m_1$	Comparand					✓								
s2	Comparative value	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓
$s2.m_2$	Comparative value					✓								

Remarks

Function

- Logical sum (OR) of s1 and s2 is substituted for ‘d’.

s1	s2	d
0	0	0
0	1	1
1	0	1
1	1	1

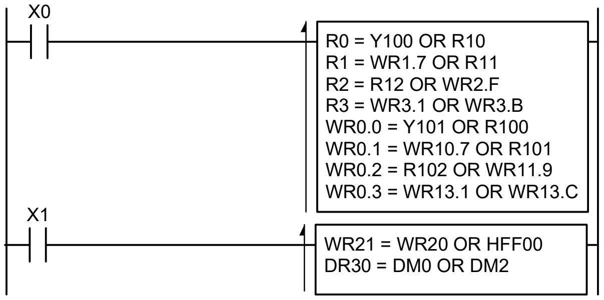
- The combination of d and s are as follows.

d	s1	s2
Bit	Bit	Bit
	Bit in Word	Bit
	Bit	Bit in Word
	Bit in Word	Bit in Word
Bit in Word	Bit	Bit
	Bit in Word	Bit
	Bit	Bit in Word
	Bit in Word	Bit in Word
Word	Word	Word
Double word	Double word	Double word

Cautionary notes

- Only the word internal output of WR can be specified to d.m_0, s1.m_1, and s2.m_2.
- m_0, m_1, and m_2 are from 0 to F.

Program example



```
R0 = Y100 OR R10
R1 = WR1.7 OR R11
R2 = R12 OR WR2.F
R3 = WR3.1 OR WR3.B
WR0.0 = Y101 OR R100
WR0.1 = WR10.7 OR R101
WR0.2 = R102 OR WR11.9
WR0.3 = WR13.1 OR WR13.C
```

```
WR21 = WR20 OR HFF00
DR30 = DM0 OR DM2
```

[Program description]

At the rising edge of X0,

- Logical sum (OR) of Y100 and R10 is substituted for R0.
- Logical sum (OR) of the 7th bit of WR1and R11is substituted for R1.
- Logical sum (OR) of R12 and the Fth bit (the 15th bit) of WR2 is substituted for R2.
- Logical sum (OR) of the 1st bit of WR3 and the Bth bit (the 11th bit) of WR3 is substituted for R3.
- Logical sum (OR) of Y101and R100 is substituted for the 0th bit of WR0.
- Logical sum (OR) of the 7th bit of WR10 and R101 is substituted for the 1st bit of WR0.
- Logical sum (OR) of R102 and the 9th bit of WR11 is substituted for the 2nd bit of WR0.
- Logical sum (OR) of the 1st bit of WR13 and the Cth bit (the 12th bit) of WR13 is substituted for the 3rd bit of WR0.

At the rising edge of X1,

- Logical sum (OR) of WR20 and HFF00 is substituted for WR21.
- Logical sum (OR) of DM0 and DM2 is substituted for DR30.

Name	Logical conjunction (AND)

Ladder format

d = s1 AND s2

d = $s1.m_1$ AND s2
d = s1 AND $s2.m_2$
d = $s1.m_1$ AND $s2.m_2$
$d.m_0$ = s1 AND s2
$d.m_0$ = $s1.m_1$ AND s2
$d.m_0$ = s1 AND $s2.m_2$
$d.m_0$ = $s1.m_1$ AND $s2.m_2$

Number of steps				Condition code				
Condition			Steps ( ) indicates DW	R7F4	R7F3	R7F2	R7F1	R7F0
d	s1	s2		DER	ERR	SD	V	C
I/O	I/O	I/O	3 (6)	●	●	●	●	●
I/O	I/O.m	I/O	6 (−)					
I/O	I/O	I/O.m	6 (−)					
I/O	I/O.m	I/O.m	7 (−)					
I/O.m	I/O	I/O	6 (−)					
I/O.m	I/O.m	I/O	7 (−)					
I/O.m	I/O	I/O.m	7 (−)					
I/O.m	I/O.m	I/O.m	8 (−)					

Command processing time (μs)

Average					Maximum				
Condition			Time ( ) indicates DW		Condition			Time ( ) indicates DW	
d	s1	s2	MVH (High function)	MVL (Standard)	d	s1	s2	MVH (High function)	MVL (Standard)
I/O	I/O	I/O	0.96(5.5)	0.96(6.41)	I/O	I/O	I/O	—	—
I/O	I/O.m	I/O	7.92	9.51	I/O	I/O.m	I/O	—	—
I/O	I/O	I/O.m	7.92	9.51	I/O	I/O	I/O.m	—	—
I/O	I/O.m	I/O.m	7.92	9.51	I/O	I/O.m	I/O.m	—	—
I/O.m	I/O	I/O	7.5	8.91	I/O.m	I/O	I/O	—	—
I/O.m	I/O.m	I/O	7.64	9.11	I/O.m	I/O.m	I/O	—	—
I/O.m	I/O	I/O.m	7.64	9.11	I/O.m	I/O	I/O.m	—	—
I/O.m	I/O.m	I/O.m	7.64	9.11	I/O.m	I/O.m	I/O.m	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Substitution destination		✓	✓			✓	✓	✓	✓		✓	✓	
$d.m_0$	Substitution destination					✓								
s1	Comparand	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓
$s1.m_1$	Comparand					✓								
s2	Comparative value	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓
$s2.m_2$	Comparative value					✓								

Remarks

Function

- Logical conjunction (AND) of s1 and s2 is substituted for ‘d’.

s1	s2	d
0	0	0
0	1	0
1	0	0
1	1	1

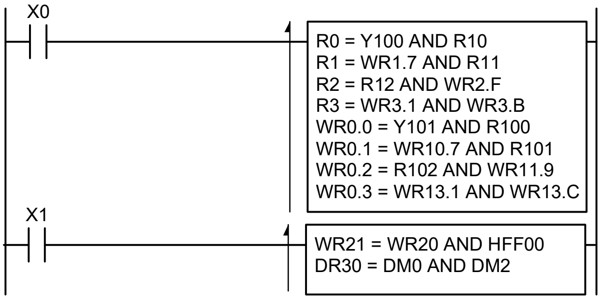
- The combination of d and s are as follows.

d	s1	s2
Bit	Bit	Bit
	Bit in Word	Bit
	Bit	Bit in Word
	Bit in Word	Bit in Word
Bit in Word	Bit	Bit
	Bit in Word	Bit
	Bit	Bit in Word
	Bit in Word	Bit in Word
Word	Word	Word
Double word	Double word	Double word

Cautionary notes

- Only the word internal output of WR can be specified to d.m_0, s1.m_1, and s2.m_2.
- m_0, m_1, and m_2 are from 0 to F.

Program example



[Program description]

At the rising edge of X0,

- Logical conjunction (AND) of Y100 and R10 is substituted for R0.
- Logical conjunction (AND) of the 7th bit of WR1 and R11 is substituted for R1.
- Logical conjunction (AND) of R12 and the Fth bit (the 15th bit) of WR2 is substituted for R2.
- Logical conjunction (AND) of the 1st bit of WR3 and the Bth bit (the 11th bit) of WR3 is substituted for R3.
- Logical conjunction (AND) of Y101 and R100 is substituted for the 0th bit of WR0.
- Logical conjunction (AND) of the 7th bit of WR10 and R101 is substituted for the 1st bit of WR0.
- Logical conjunction (AND) of R102 and the 9th bit of WR11 is substituted for the 2nd bit of WR0.
- Logical conjunction (AND) of the 1st bit of WR13 and the Cth bit of WR13 is substituted for the 3rd bit of WR0.

At the rising edge of X1,

- Logical conjunction (AND) of WR20 and HFF00 is substituted for WR21.
- Logical conjunction (AND) of DM0 and DM2 is substituted for DR30.

Name	Exclusive disjunction (XOR)

Ladder format

d = s1 XOR s2

d = s1.m$_1$ XOR s2
d = s1 XOR s2.m$_2$
d = s1.m$_1$ XOR s2.m$_2$
d.m$_0$ = s1 XOR s2
d.m$_0$ = s1.m$_1$ XOR s2
d.m$_0$ = s1 XOR s2.m$_2$
d.m$_0$ = s1.m$_1$ XOR s2.m$_2$

Number of steps				Condition code				
Condition			Steps ( ) indicates DW	R7F4	R7F3	R7F2	R7F1	R7F0
d	s1	s2		DER	ERR	SD	V	C
I/O	I/O	I/O	3 (6)	●	●	●	●	●
I/O	I/O.m	I/O	6 (−)					
I/O	I/O	I/O.m	6 (−)					
I/O	I/O.m	I/O.m	7 (−)					
I/O.m	I/O	I/O	6 (−)					
I/O.m	I/O.m	I/O	7 (−)					
I/O.m	I/O	I/O.m	7 (−)					
I/O.m	I/O.m	I/O.m	8 (−)					

Command processing time (µs)									
Average					Maximum				
Condition			Time ( ) indicates DW		Condition			Time ( ) indicates DW	
d	s1	s2	MVH (High function)	MVL (Standard)	d	s1	s2	MVH (High function)	MVL (Standard)
I/O	I/O	I/O	0.97(5.5)	0.96(6.43)	I/O	I/O	I/O	−	−
I/O	I/O.m	I/O	5.95	9.51	I/O	I/O.m	I/O	−	−
I/O	I/O	I/O.m	5.95	9.51	I/O	I/O	I/O.m	−	−
I/O	I/O.m	I/O.m	5.95	9.51	I/O	I/O.m	I/O.m	−	−
I/O.m	I/O	I/O	7.5	8.93	I/O.m	I/O	I/O	−	−
I/O.m	I/O.m	I/O	7.64	9.11	I/O.m	I/O.m	I/O	−	−
I/O.m	I/O	I/O.m	7.64	9.11	I/O.m	I/O	I/O.m	−	−
I/O.m	I/O.m	I/O.m	7.64	9.11	I/O.m	I/O.m	I/O.m	−	−

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Substitution destination		✓	✓			✓	✓	✓	✓		✓	✓	
d.m$_0$	Substitution destination					✓								
s1	Comparand	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓
s1.m$_1$	Comparand					✓								
s2	Comparative value	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓
s2.m$_2$	Comparative value					✓								

Remarks

Function

- Exclusive disjunction (XOR) of s1 and s2 is substituted for d.

s1	s2	D
0	0	0
0	1	1
1	0	1
1	1	0

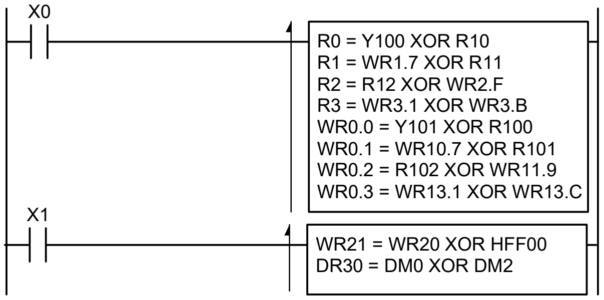
- The combination of d and s are as follows.

d	s1	s2
Bit	Bit	Bit
	Bit in Word	Bit
	Bit	Bit in Word
	Bit in Word	Bit in Word
Bit in Word	Bit	Bit
	Bit in Word	Bit
	Bit	Bit in Word
	Bit in Word	Bit in Word
Word	Word	Word
Double word	Double word	Double word

Cautionary notes

- Only the word internal output of WR can be specified to $d.m_0$, $s1.m_1$, and $s2.m_2$.
- m_0, m_1, and m_2 are from 0 to F.

Program example



```
R0 = Y100 XOR R10
R1 = WR1.7 XOR R11
R2 = R12 XOR WR2.F
R3 = WR3.1 XOR WR3.B
WR0.0 = Y101 XOR R100
WR0.1 = WR10.7 XOR R101
WR0.2 = R102 XOR WR11.9
WR0.3 = WR13.1 XOR WR13.C
```

```
WR21 = WR20 XOR HFF00
DR30 = DM0 XOR DM2
```

[Program description]

At the rising edge of X0,

- Exclusive disjunction (XOR) of Y100 and R10 is substituted for R0.
- Exclusive disjunction (XOR) of the 7th bit of WR1 and R11 is substituted for R1.
- Exclusive disjunction (XOR) of R12 and the Fth bit (the 15th bit) of WR2 is substituted for R2.
- Exclusive disjunction (XOR) of the 1st bit of WR3 and the Bth bit (the 11th bit) of WR3 is substituted for R3.
- Exclusive disjunction (XOR) of Y101 and R100 is substituted for the 0th bit of WR0.
- Exclusive disjunction (XOR) of the 7th bit of WR10 and R101 is substituted for the 1st bit of WR0.
- Exclusive disjunction (XOR) of R102 and the 9th bit of WR11 is substituted for the 2nd bit of WR0.
- Exclusive disjunction (XOR) of the 1st bit of WR13 and the Cth bit (the 12th bit) of WR13 is substituted for the 3rd bit of WR0.

At the rising edge of X1,

- Exclusive disjunction (XOR) of WR20 and HFF00 is substituted for WR21.
- Exclusive disjunction (XOR) of DM0 and DM2 is substituted for DR30.

Name	= Comparison expression

Ladder format	Number of steps				Condition code				
	Condition			Steps	R7F4	R7F3	R7F2	R7F1	R7F0
	d	s1	s2		DER	ERR	SD	V	C
d = s1 == s2 d.m$_0$ = s1 == s2	B	W	W	3	●	●	●	●	●
	B	DW	DW	7					
	B(.m)	W	W	6					
	B(.m)	DW	DW	8					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d:B / s1, s2:W	0.94	0.94	—	—	—
d:B / s1, s2:DW	8.08	9.61	—	—	—
d:B(.m) / s1, s2:W	6.4	7.65	—	—	—
d:B(.m) / s1, s2:DW	7.76	9.19	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Substitution destination		✓	✓											
d.m$_0$	Substitution destination						✓								
s1	Comparand							✓	✓	✓	✓	✓	✓	✓	✓
s2	Comparative value							✓	✓	✓	✓	✓	✓	✓	✓

Remarks
B means Bit I/O, W means Word I/O, and DW means Double word I/O.

Function

- As binary data in s1 and s2, 1 is substituted for d when s1 = s2, and 0 is substituted in all other cases.
- As binary data in s1 and s2, 1 is substituted for the mth bit of word data d when s1 = s2, and 0 is substituted in all other cases.
- The combinations of d, s1, and s2 are as follows.

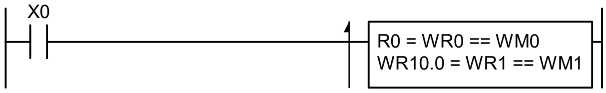
d	s1	s2
Bit	Word	Word
Bit	Double word	Double word

d.m	s1	s2
Word (the mth bit)	Word	Word
Word (the mth bit)	Double word	Double word

Cautionary notes

- Only the word internal output of WR can be specified to d.m$_0$.
- m$_0$ is from 0 to F.

Program example



[Program description]

At the rising edge of X0,

- When the value of WR0 and WM0 are the same, R0 is set to 1. In all other cases, it is reset to 0.
- When the value of WR1 and WM1 are the same, the 0th bit of WR10 is set to 1. In all other cases, it is reset to 0.

Name	= Comparison expression (Signed integer)

Ladder format	Number of steps				Condition code				
	Condition			Steps	R7F4	R7F3	R7F2	R7F1	R7F0
	d	s1	s2		DER	ERR	SD	V	C
d = s1.S == s2.S $d.m_0$ = s1.S == s2.S	B	W	W	3	●	●	●	●	●
	B	DW	DW	7					
	B(.m)	W	W	6					
	B(.m)	DW	DW	8					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d:B / s1, s2:W	0.94	0.94	—	—	—
d:B / s1, s2:DW	8.04	9.67	—	—	—
d:B(.m) / s1, s2:W	6.34	7.67	—	—	—
d:B(.m) / s1, s2:DW	7.76	9.23	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT, TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Substitution destination		✓	✓										
$d.m_0$	Substitution destination					✓								
s1.S	Comparand						✓	✓	✓	✓	✓	✓	✓	✓
s2.S	Comparative value						✓	✓	✓	✓	✓	✓	✓	✓

Remarks
B means Bit I/O, W means Word I/O, and DW means Double word I/O.

Function

- As signed binary data in s1 and s2, 1 is substituted for d when s1.S = s2.S, and in all other cases, 0 is substituted.
- As signed binary data in s1 and s2, 1 is substituted for the mth bit of word data d when s1.S = s2.S, and in all other cases, 0 is substituted.
- The range of s1.S and s2.S are as follows,

  when it is Word, the range is −32,768 to 32,767 (decimal number) and H8000 to H7FFF (hexadecimal number).

  when it is Double word, the range is −2,147,483,648 to 2,147,483,647 (decimal number) and H80000000 to H7FFFFFFF (hexadecimal number).
- The combination of d, s1, and s2 are as follows.

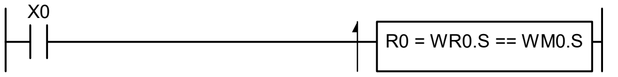
d	s1	s2
Bit	Word	Word
Bit	Double word	Double word

d.m	s1	s2
Word (the mth bit)	Word	Word
Word (the mth bit)	Double word	Double word

Cautionary notes

- Only the word internal output can be specified to $d.m_0$.
- m_0 is from 0 to F.

Program example



[Program description]

The value of WR0.S and WM0.S are the same at the rising edge of X0, R0 is set to 1. In all other cases, it is reset to 0.

Name	= Comparison expression (Floating point)

Ladder format	Number of steps				Condition code				
	Condition			Steps	R7F4	R7F3	R7F2	R7F1	R7F0
	d	s1	s2		DER	ERR	SD	V	C
d = s1.FL == s2.FL d.m$_0$ = s1.FL == s2.FL	B	DW	DW	7	●	●	●	●	●
	B(.m)	DW	DW	8					

Command processing time (μs)		
Average		
Condition	ProcessingTime	
	MVH (High function)	MVL (Standard)
	Double word	Double word
d:B / s1, s2:DW	8.02	9.65
d:B.m / s1, s2:DW	7.76	9.21

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Substitution destination		✓	✓										
d.m$_0$	Substitution destination					✓								
s1.FL	Comparand										✓	✓	✓	✓
s2.FL	Comparative value										✓	✓	✓	✓

Remarks
B means Bit I/O, and DW means Double word I/O. Floating point is specified by double-word. Constant is 20 digit maximum.

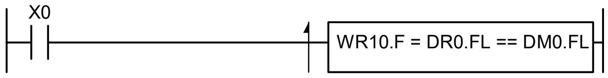
Function

- As floating point data in s1 and s2, 1 is substituted for d when s1.FL = s2.FL, and in all other cases, 0 is substituted.
- As floating point data in s1 and s2, 1 is substituted for the mth bit of word data d when s1.FL = s2.FL, and in all other cases, 0 is substituted.
- The range of s1 and s2 -3.40282×10^{38} to 3.40282×10^{38} (decimal number),
  HFF7FFFFF to H80800000 and H00800000 to H7F7FFFFF (hexadecimal number)

Cautionary notes

- Since there is an error in floating point, the error may cause the disagreement even if the value from the calculation is in agreement. We recommend deciding the comparison of floating point not in agreement and disagreement but in “range”.
- Only the word internal output of WR can be specified to d.m$_0$.
- m$_0$ is from 0 to F.

Program example



[Program description]

When the value of DR0.FL and DM0.FL are the same at the rising edge of X0, the Fth bit of WR10 is set to 1. In all other cases, it is reset to 0.

Name	<> Comparison expression

Ladder format	Number of steps				Condition code				
	Condition			Steps	R7F4	R7F3	R7F2	R7F1	R7F0
	d	s1	s2		DER	ERR	SD	V	C
d = s1 <> s2 d.m$_0$ = s1 <> s2	B	W	W	3	●	●	●	●	●
	B	DW	DW	7					
	B(.m)	W	W	6					
	B(.m)	DW	DW	8					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d:B / s1, s2:W	0.94	0.94	—	—	—
d:B / s1, s2:DW	8.04	9.64	—	—	—
d:B(.m) / s1, s2:W	6.34	7.65	—	—	—
d:B(.m) / s1, s2:DW	7.76	9.18	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Substitution destination		✓	✓										
d.m$_0$	Substitution destination					✓								
s1	Comparand						✓	✓	✓	✓	✓	✓	✓	✓
s2	Comparative value						✓	✓	✓	✓	✓	✓	✓	✓

Remarks
B means Bit I/O, W means Word I/O, and DW means Double word I/O.

Function

- As binary data in s1 and s2, 1 is substituted for d when s1 ≠ s2, and in all other cases, 0 is substituted.
- As binary data in s1 and s2, 1 is substituted for the mth bit of word data d when s1 ≠ s2, and in all other cases, 0 is substituted.
- The combination of d, s1 and s2 are as follows.

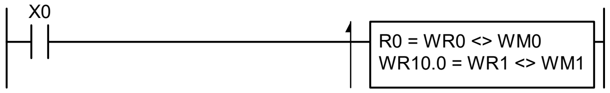
d	s1	s2
Bit	Word	Word
Bit	Double word	Double word

d.m	s1	s2
Word (the mth bit)	Word	Word
Word (the mth bit)	Double word	Double word

Cautionary notes

- Only the word internal output of WR can be specified to d.m$_0$.
- m$_0$ is from 0 to F.

Program example



[Program description]

At the rising edge of X0,

- When the value of WR0 and WM0 are different, R0 is set to 1. When the value is the same, it is reset to 0.
- When the value of WR1 and WM1 are different, the 0th bit of WR10 is set to 1. When the value is the same, it is reset to 0.

Name	<> Comparison expression (Signed integer)

Ladder format	Number of steps				Condition code				
	Condition			Steps	R7F4	R7F3	R7F2	R7F1	R7F0
	d	s1	s2		DER	ERR	SD	V	C
d = s1.S <> s2.S d.m$_0$ = s1.S <> s2.S	B	W	W	3	●	●	●	●	●
	B	DW	DW	7					
	B(.m)	W	W	6					
	B(.m)	DW	DW	8					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d:B / s1, s2:W	0.94	0.95	—	—	—
d:B / s1, s2:DW	8.12	9.65	—	—	—
d:B(.m) / s1, s2:W	6.42	7.67	—	—	—
d:B(.m) / s1, s2:DW	7.78	9.19	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Substitution destination		✓	✓											
d.m$_0$	Substitution destination						✓								
s1.S	Comparand							✓	✓	✓	✓	✓	✓	✓	✓
s2.S	Comparative value							✓	✓	✓	✓	✓	✓	✓	✓

Remarks
B means Bit I/O, W meand Word I/O, and DW means Double word I/O.

Function

- As signed binary data in s1 and s2, 1 is substituted for d when s1.S ≠ s2.S, and in all other cases, 0 is substituted.
- As signed binary data in s1 and s2, 1 is substituted for the mth bit of word data d when s1.S ≠ s2.S, and in all other cases, 0 is substituted.
- The range of s1.S and s2.S are as follows,

  when it is Word, the range is −32,768 to 32,767 (decimal number) and H8000 to H7FFF (hexadecimal number).

  when it is Double word, the range is −2,147,483,648 to 2,147,483,647 (decimal number) and H80000000 to H7FFFFFFF (hexadecimal number).
- The combination of d, s1 and s2 are as follows.

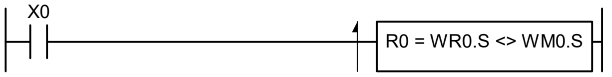
d	s1	s2
Bit	Word	Word
Bit	Double word	Double word

d.m	s1	s2
Word (the mth bit)	Word	Word
Word (the mth bit)	Double word	Double word

Cautionary notes

- Only the word internal output of WR can be specified to d.m$_0$.
- m$_0$ is from 0 to F.

Program example



[Program description]

When the value of WR0.S and WM0.S are different at the rising edge of X0, R0 is set to 1. When the value is the same, it is reset to 0.

Name	<> Comparison expression (Floating point)

Ladder format
d = s1.FL <> s2.FL d.m0 = s1.FL <> s2.FL

Number of steps				Condition code				
Condition			Steps	R7F4	R7F3	R7F2	R7F1	R7F0
d	s1	s2		DER	ERR	SD	V	C
B	DW	DW	7	●	●	●	●	●
B(.m)	DW	DW	8					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d:B / s1, s2:DW	8.1	9.63	—	—	—
d:B.m / s1, s2:DW	7.78	9.19	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Substitution destination		✓	✓										
d.m0	Substitution destination					✓								
s1.FL	Comparand										✓	✓	✓	✓
s2.FL	Comparative value										✓	✓	✓	✓

Remarks
B means Bit I/O, and DW means Double word I/O.

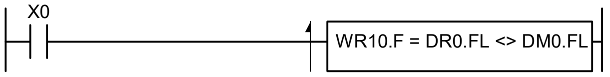
Function

- As floating point data in s1 and s2, 1 is substitutetd for d when s1.FL ≠ s2.FL, and in all other cases, 0 is substituted.
- As floating point data in s1 and s2, 1 is substituted for the mth bit of word data d when s1.FL ≠ s2.FL, and in all other cases, 0 is substituted.
- The range of s1 and s2 -3.40282×10^{38} to 3.40282×10^{38} (decimal number),
  HFF7FFFFF to H80800000 and H00800000 to H7F7FFFFF (hexadecimal number)

Cautionary notes

- Since there is an error in floating point, the error may cause the disagreement even if the value from the calculation is in agreement. We recommend deciding the comparison of floating point not in agreement or disagreement but in “range”.
- Only the word internal output of WR can be specified to d.m$_0$.
- m$_0$ is from 0 to F.

Program example



[Program description]

When the value of DR0.FL and DM0.FL are different at the rising edge of X0, the Fth bit of WR10 is set to 1. When the value is the same, it is reset to 0.

Name	< Comparison expression

Ladder format	Number of steps				Condition code				
	Condition			Steps ( )indicates DW	R7F4	R7F3	R7F2	R7F1	R7F0
	d	s1	s2		DER	ERR	SD	V	C
d = s1 < s2 d.m_0 = s1 < s2	B	W	W	3	●	●	●	●	●
	B	DW	DW	7					
	B(.m)	W	W	6					
	B(.m)	DW	DW	8					

Command processing time (μs)

Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d:B / s1, s2:W	0.95	0.95	—	—	—
d:B / s1, s2:DW	8.07	9.61	—	—	—
d:B(.m) / s1, s2:W	6.41	7.61	—	—	—
d:B(.m) / s1, s2:DW	7.81	9.18	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Substitution destination		✓	✓										
d.m_0	Substitution destination					✓								
s1	Comparand						✓	✓	✓	✓	✓	✓	✓	✓
s2	Comparative value						✓	✓	✓	✓	✓	✓	✓	✓

Remarks

B means Bit I/O, W means Word I/O, and DW means Double word I/O.

Function

- As binary data in s1 and s2, ‘1’ is substituted for d when s1 < s2, and in all other cases, 0 is substituted.
- As binary data in s1 and s2, ‘1’ is substituted for the mth bit of word data d when s1 < s2, and in all other cases, 0 is substituted.
- The combination of d, s1 and s2 are as follows.

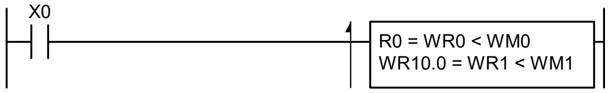
d	s1	s2
Bit	Word	Word
Bit	Double word	Double word

d.m	s1	s2
Word (the mth bit)	Word	Word
Word (the mth bit)	Double word	Double word

Cautionary notes

- Only the word internal output of WR can be specified to d.m_0.
- m_0 is from 0 to F.

Program example



[Program description]

At the rising edge of X0,

- When WR0 < WM0, R0 is set to 1. When WR0 ≥ WM0, R0 is reset to 0.
- When WR1 < WM1, the 0th bit of WR10 is set to 1. When WR1 ≥ WM1, the 0th bit of WR10 is reset to 0.

Name	< Comparison expression (Signed integer)

Ladder format	Number of steps				Condition code				
	Condition			Steps ( )indicates DW	R7F4	R7F3	R7F2	R7F1	R7F0
	d	s1	s2		DER	ERR	SD	V	C
d = s1.S < s2.S $d.m_0$ = s1.S < s2.S	B	W	W	5	●	●	●	●	●
	B	DW	DW	7					
	B(.m)	W	W	6					
	B(.m)	DW	DW	8					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d:B / s1, s2:W	0.95	0.95	–	–	–
d:B / s1, s2:DW	8.11	9.64	–	–	–
d:B(.m) / s1, s2:W	6.43	7.66	–	–	–
d:B(.m) / s1, s2:DW	7.83	9.2	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Substitution destination		✓	✓											
$d.m_0$	Substitution destination						✓								
s1.S	Comparand							✓	✓	✓	✓	✓	✓	✓	✓
s2.S	Comparative value							✓	✓	✓	✓	✓	✓	✓	✓

Remarks
B means Bit I/O, and DW means Double word I/O.

Function

- As signed binary data in s1 and s2, 1 is substituted for d when s1.S < s2.S, and in all other cases, 0 is substituted.
- As signed binary data in s1 and s2, 1 is substituted for the mth bit of word data d when s1.S < s2.S, and in all other cases, 0 is substituted.
- The range of s1.S and s2.S are as follows,
  when it is Word, the range is −32,768 to 32,767 (decimal number) and H8000 to H7FFF (hexadecimal number).
  when it is Double word, the range is −2,147,483,648 to 2,147,483,647 (decimal number) and H80000000 to H7FFFFFFF (hexadecimal number).
- The combination of d, s1 and s2 are as follows.

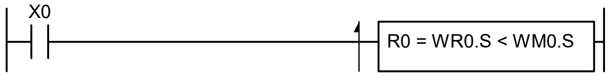
d	s1	s2
Bit	Word	Word
Bit	Double word	Double word

d.m	s1	s2
Word (the mth bit)	Word	Word
Word (the mth bit)	Double word	Double word

Cautionary notes

- Only the word internal output of WR can be specified to $d.m_0$.
- m_0 is from 0 to F.

Program example



[Program description]

When WR0.S < WM0.S at the rising edge of X0, R0 is set to 1. When WR0.S ≥ WM0.S, R0 is reset to 0.

Name	< Comparison expression (Floating point)

Ladder format	Number of steps				Condition code				
	Condition			Steps	R7F4	R7F3	R7F2	R7F1	R7F0
	d	s1	s2		DER	ERR	SD	V	C
d = s1.FL < s2.FL	B	DW	DW	7	●	●	●	●	●
$d.m_0$ = s1.FL < s2.FL	B(.m)	DW	DW	8					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d:B / s1, s2:DW	8.09	9.64	–	–	–
d:B.m / s1, s2:DW	7.87	9.2	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Substitution destination		✓	✓										
$d.m_0$	Substitution destination					✓								
s1.FL	Comparand										✓	✓	✓	✓
s2.FL	Comparative value										✓	✓	✓	✓

Remarks
B means Bit I/O, and DW means Double word I/O.

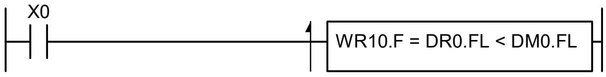
Function

- As floating point data in s1 and s2, 1 is substituted for d when s1.FL < s2.FL, and in all other cases, 0 is substituted.
- As floating point data in s1 and s2, 1 is substituted for the mth bit of word data d when s1.FL < s2.FL, and in all other cases, 0 is substituted.
- The range of s1 and s2 -3.40282×10^{38} to 3.40282×10^{38} (decimal number),
  HFF7FFFFF to H80800000 and H00800000 to H7F7FFFFF (hexadecimal number)

Cautionary notes

- Since there is an error in floating point, the error may cause the disagreement even if thevalue from the calculation is in agreement. We recommend deciding the comparison of floating point not in agreement and disagreement but in "range".
- Only the word internal output of WR can be specified to $d.m_0$.
- m_0 is from 0 to F.

Program example



[Program description]

When DR0.FL < DM0.FL, the Fth bit of WR10 is set to 1 at the rising edge of X0.

When DR0.FL ≥ DM0.FL, the Fth bit of WR10 is reset to 0.

Name	<= Comparison expression

Ladder format

d = s1 <= s2

$d.m_0$ = s1 <= s2

Number of steps			
Condition			Steps
d	s1	s2	
B	W	W	3
B	DW	DW	7
B(.m)	W	W	6
B(.m)	DW	DW	8

Condition code				
R7F4	R7F3	R7F2	R7F1	R7F0
DER	ERR	SD	V	C
●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d:B / s1, s2:W	0.95	0.95	—	—	—
d:B / s1, s2:DW	8.11	9.64	—	—	—
d:B(.m) / s1, s2:W	6.45	7.66	—	—	—
d:B(.m) / s1, s2:DW	7.79	9.19	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Substitution destination		✓	✓											
$d.m_0$	Substitution destination						✓								
s1	Comparand							✓	✓	✓	✓	✓	✓	✓	✓
s2	Comparative value							✓	✓	✓	✓	✓	✓	✓	✓

Remarks
B means Bit I/O, W means Word I/O, and DW means Double word I/O.

Function

- As binary data in s1 and s2, 1 is substituted for d when s1 ≤ s2, and in all other cases, 0 is substituted.
- As binary data in s1 and s2, 1 is substituted for the mth bit of word data d when s1 ≤ s2, and in all other cases, 0 is substituted.
- The combination of d, s1 and s2 are as follows.

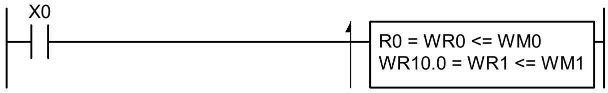
d	s1	s2
Bit	Word	Word
Bit	Double word	Double word

d.m	s1	s2
Word (the mth bit)	Word	Word
Word (the mth bit)	Double word	Double word

Cautionary notes

- Only the word internal output of WR can be specified to $d.m_0$.
- m_0 is from 0 to F.

Program example



[Program description]

At the rising edge of X0,

- When WR0 ≤ WM0, R0 is set to 1. When WR0 > WM0, R0 is reset to 0.
- When WR1 ≤ WM1, the 0th bit of WR10 is set to 1. When WR1 > WM1, the 0th bit of WR10 is reset to 0.

Name	<= Comparison expression (Signed integer)

Ladder format	Number of steps				Condition code				
	Condition			Steps	R7F4	R7F3	R7F2	R7F1	R7F0
	d	s1	s2		DER	ERR	SD	V	C
d = s1.S <= s2.S d.m_0 = s1.S <= s2.S	B	W	W	5	●	●	●	●	●
	B	DW	DW	7					
	B(.m)	W	W	6					
	B(.m)	DW	DW	8					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d:B / s1, s2:W	6.75	8.1	—	—	—
d:B / s1, s2:DW	8.11	9.64	—	—	—
d:B(.m) / s1, s2:W	6.45	7.64	—	—	—
d:B(.m) / s1, s2:DW	7.81	9.2	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Substitution destination		✓	✓										
d.m_0	Substitution destination					✓								
s1.S	Comparand						✓	✓	✓	✓	✓	✓	✓	✓
s2.S	Comparative value						✓	✓	✓	✓	✓	✓	✓	✓

Remarks
B means Bit I/O, W means Word I/O, and DW means Double word I/O.

Function

- As signed binary data in s1 and s2, 1 is substituted for d when s1.S ≤ s2.S, and in all other cases, 0 is substituted.
- As signed binry data in s1 and s2, 1 is substituted for the mth bit of word data d when s1.S ≤ s2.S, and in all other cases, 0 is substituted.
- The range of s1.S and s2.S are as follows,

  when it is Word, the range is −32,768 to 32,767 (decimal number) and H8000 to H7FFF (hexadecimal number).

  when it is Double word, the range is −2,147,483,648 to 2,147,483,647 (decimal number) and H80000000 to H7FFFFFFF (hexadecimal number).
- The combination of d, s1 and s2 are as follows.

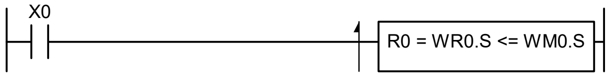
d	s1	s2
Bit	Word	Word
Bit	Double word	Double word

d.m	s1	s2
Word (the mth bit)	Word	Word
Word (the mth bit)	Double word	Double word

Cautionary notes

- Only the word internal output of WR can be specified to d.m_0.
- m_0 is from 0 to F.

Program example



[Program description]

When WR0.S ≤ WM0.S, R0 is set to 1 at the rising edge of X0. When WR0.S > WM0.S, R0 is reset to 0.

Name	<= Comparison expression (Floating point)

Ladder format	Number of steps				Condition code				
	Condition			Steps	R7F4	R7F3	R7F2	R7F1	R7F0
	d	s1	s2		DER	ERR	SD	V	C
d = s1.FL <= s2.FL d.m$_0$ = s1.FL <= s2.FL	B	DW	DW	7	●	●	●	●	●
	B(.m)	DW	DW	8					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d:B / s1, s2:DW	7.57	8.92	—	—	—
d:B.m / s1, s2:DW	7.29	8.46	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Substitution destination		✓	✓										
d.m$_0$	Substitution destination					✓								
s1.FL	Comparand										✓	✓	✓	✓
s2.FL	Comparative value										✓	✓	✓	✓

Remarks
B means Bit I/O, and DW means Double word I/O.

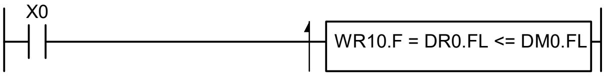
Function

- As floating point data in s1 and s2, 1 is substituted for d when s1.FL ≤ s2.FL, and in all other cases, 0 is substituted.
- As floating point data in s1 and s2, 1 is substituted for the mth bit of word data d when s1.FL ≤ s2.FL, and in all other cases, 0 is substituted.
- The range of s1 and s2 -3.40282×10^{38} to 3.40282×10^{38} (decimal number),
  HFF7FFFFF to H80800000 and H00800000 to H7F7FFFFF (hexadecimal number)

Cautionary notes

- Since there is an error in floating point, the error may cause the disagreement even if the value from the calculation is in agreement. We recommend deciding the comparison of floating point not in agreement and disagreement but in "range".
- Only the word internal output of WR can be specified to d.m$_0$.
- m$_0$ is from 0 to F.

Program example



[Program description]

When DR0.FL ≤ DM0.FL, the Fth bit of WR10 is set to 1 at the rising edge of X0.

When DR0.FL > DM0.FL, the Fth bit of WR10 is reset to 0.

Name	> Comparison expression

Ladder format	Number of steps				Condition code				
	Condition			Steps	R7F4	R7F3	R7F2	R7F1	R7F0
	d	s1	s2		DER	ERR	SD	V	C
d = s1 > s2 $d.m_0$ = s1 > s2	B	W	W	3	●	●	●	●	●
	B	DW	DW	7					
	B(.m)	W	W	6					
	B(.m)	DW	DW	8					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d:B / s1, s2:W	0.95	0.95	—	—	—
d:B / s1, s2:DW	8.07	9.63	—	—	—
d:B(.m) / s1, s2:W	6.37	7.65	—	—	—
d:B(.m) / s1, s2:DW	7.77	9.21	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Substitution destination		✓	✓										
$d.m_0$	Substitution destination					✓								
s1	Comparand						✓	✓	✓	✓	✓	✓	✓	✓
s2	Comparative value						✓	✓	✓	✓	✓	✓	✓	✓

Remarks
B means Bit I/O, W means Word I/O, and DW means Double word I/O.

Function

- As binary data in s1 and s2, 1 is substituted for d when s1 > s2, and in all other cases, 0 is substituted.
- As binary data in s1 and s2, 1 is substituted for the mth bit of word data d when s1 > s2, and in all other cases, 0 is substituted.
- The combination of d, s1 and s2 are as follows.

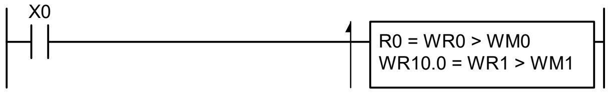
d	s1	s2
Bit	Word	Word
Bit	Double word	Double word

d.m	s1	s2
Word (the mth bit)	Word	Word
Word (the mth bit)	Double word	Double word

Cautionary notes

- Only the word internal output of WR can be specified to $d.m_0$.
- m_0 is from 0 to F.

Program example



[Program description]

At the rising edge of X0,

When WR0 > WM0, R0 is set to 1. When WR0 ≤ WM0, R0 is reset to 0.

When WR1 > WM1, the 0th bit of WR10 is set to 1. When WR1 ≤ WM1, the 0th bit of WR10 is reset to 0.

Name	> Comparison expression (Signed integer)

Ladder format	Number of steps				Condition code				
	Condition			Steps	R7F4	R7F3	R7F2	R7F1	R7F0
	d	s1	s2		DER	ERR	SD	V	C
d = s1.S > s2.S d.m_0 = s1.S > s2.S	B	W	W	5	●	●	●	●	●
	B	DW	DW	7					
	B(.m)	W	W	6					
	B(.m)	DW	DW	8					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d:B / s1, s2:W	6.69	8.09	—	—	—
d:B / s1, s2:DW	8.09	9.63	—	—	—
d:B(.m) / s1, s2:W	6.43	7.65	—	—	—
d:B(.m) / s1, s2:DW	7.79	9.21	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Substitution destination		✓	✓											
d.m_0	Substitution destination						✓								
s1.S	Comparand							✓	✓	✓	✓	✓	✓	✓	✓
s2.S	Comparative value							✓	✓	✓	✓	✓	✓	✓	✓

Remarks
B means Bit I/O, W means Word I/O, and DW means Double word I/O.

Function

- As signed binary data in s1 and s2, 1 is substituted for d when s1.S > s2.S, and in all other cases, 0 is substituted.
- As signed binary data in s1 and s2, 1 is substituted for the mth bit of word data d when s1.S > s2.S, and in all other cases, 0 is substituted.
- The range of s1.S and s2.S are as follows,
  when it is Word, the rnage is −32,768 to 32,767 (decimal number) and H8000 to H7FFF (hexadecimal number)
  when it is Double word, the range is −2,147,483,648 to 2,147,483,647 (decimal number), H80000000 to H7FFFFFFF (hexadecimal number).
- The combination of d, and s2 are as follows.

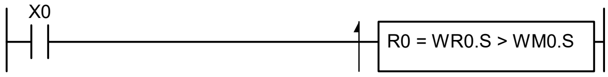
d	S1	s2
Bit	Word	Word
Bit	Double word	Double word

d.m	s1	s2
Word (the mth bit)	Word	Word
Word (the mth bit)	Double word	Double word

Cautionary notes

- Only the word internal output of WR can be specified to d.m_0.
- m_0 is from 0 to F.

Program example



[Program description]

When WR0.S > WM0.S, R0 is set to 1 at the rising edge of X0. When WR0.S ≤ WM0.S, R0 is reset to 0.

Name	> Comparison expression (Floating point)

Ladder format	Number of steps				Condition code				
	Condition			Steps	R7F4	R7F3	R7F2	R7F1	R7F0
	d	s1	s2		DER	ERR	SD	V	C
d = s1.FL > s2.FL d.m_0 = s1.FL > s2.FL	B	DW	DW	7	●	●	●	●	●
	B(.m)	DW	DW	8					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d:B / s1, s2:DW	7.53	8.93	—	—	—
d:B.m / s1, s2:DW	7.21	8.49	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Substitution destination		✓	✓											
d.m_0	Substitution destination						✓								
s1.FL	Comparand											✓	✓	✓	✓
s2.FL	Comparative value											✓	✓	✓	✓

Remarks
B means Bit I/O, and DW means Double word I/O.

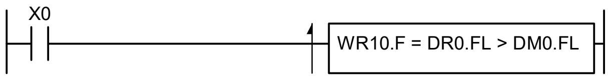
Function

- As floating point data in s1 and s2, 1 is substituted for d when s1.FL > s2.FL, and in all other cases, 0 is substituted.
- As floating point data in s1 and s2, 1 is substituted for the mth bit of word data d when s1.FL > s2.FL, and in all other cases, 0 is substituted.
- The range of s1 and s2 -3.40282×10^{38} to 3.40282×10^{38} (decimal number),
  HFF7FFFFF to H80800000 and H00800000 to H7F7FFFFF (hexadecimal number)

Cautionary notes

- Since there is an error in floating point, the error may cause the disagreement even if the value from the calculation is in agreement. We recommend deciding the comparison of floating point not in agreement and disagreement but in “range”.
- Only the word internal output of WR can be specified to d.m_0.
- m_0 is from 0 to F.

Program example



[Program description]

When DR0.FL > DM0.FL, the Fth bit of WR10 is set to 1 at the rising edge of X0.

When DR0.FL ≤ DM0.FL, the Fth bit of WR10 is reset to 0.

Name	>= Comparison expression

Ladder format	Number of steps				Condition code				
	Condition			Steps	R7F4	R7F3	R7F2	R7F1	R7F0
	d	s1	s2		DER	ERR	SD	V	C
d = s1 >= s2 d.m$_0$ = s1 >= s2	B	W	W	3	●	●	●	●	●
	B	DW	DW	7					
	B(.m)	W	W	6					
	B(.m)	DW	DW	8					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d:B / s1, s2:W	0.94	0.95	—	—	—
d:B / s1, s2:DW	8.12	9.65	—	—	—
d:B(.m) / s1, s2:W	6.38	7.65	—	—	—
d:B(.m) / s1, s2:DW	7.8	9.19	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Substitution destination		✓	✓											
d.m$_0$	Substitution destination						✓								
s1	Comparand							✓	✓	✓	✓	✓	✓	✓	✓
s2	Comparative value							✓	✓	✓	✓	✓	✓	✓	✓

Remarks
B means Bit I/O, W means Word I/O, and DW means Double word I/O.

Function

- As binary data in s1 and s2, 1 is substituted for d when s1 ≥ s2, and in all other cases, 0 is substituted.
- As binary data in s1 and s2, 1 is the mth bit of word data d when s1 ≥ s2, and in all other cases, 0 is substituted.
- The combination of d, s1 and s2 are as follows.

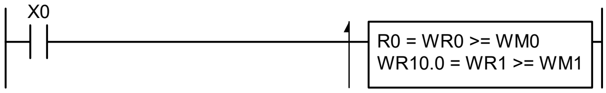
d	s1	s2
Bit	Word	Word
Bit	Double word	Double word

d.m	s1	s2
Word (the mth bit)	Word	Word
Word (the mth bit)	Double word	Double word

Cautionary notes

- Only the word internal output of WR can be specified to d.m$_0$.
- m$_0$ is from 0 to F.

Program example



[Program description]

At the rising edge of X0,

- When WR0 ≥ WM0, R is set to 1. When WR0 < WM0, R0 is reset to 0.
- When WR1 ≥ WM1, the 0th bit of WR10 is set to 1. When WR1 < WM1, the 0th bit of WR10 is reset to 0.

Name	>= Comparison expression (Signed integer)

Ladder format	Number of steps				Condition code				
	Condition			Steps	R7F4	R7F3	R7F2	R7F1	R7F0
	d	s1	s2		DER	ERR	SD	V	C
d = s1.S >= s2.S d.m_0 = s1.S >= s2.S	B	W	W	5	●	●	●	●	●
	B	DW	DW	7					
	B(.m)	W	W	6					
	B(.m)	DW	DW	8					

Command processing time (µs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d:B / s1, s2:W	6.7	8.09	—	—	—
d:B / s1, s2:DW	8.08	9.63	—	—	—
d:B(.m) / s1, s2:W	6.38	7.65	—	—	—
d:B(.m) / s1, s2:DW	7.82	9.19	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Substitution destination		✓	✓										
d.m_0	Substitution destination					✓								
s1.S	Comparand						✓	✓	✓	✓	✓	✓	✓	✓
s2.S	Comparative value						✓	✓	✓	✓	✓	✓	✓	✓

Remarks
B means Bit I/O, W means Word I/O, and DW means Double word I/O.

Function

- As signed binary data in s1 and s2, 1 is substituted for d when s1.S ≥ s2.S, and in all other cases, 0 is substituted.
- As signed binary data in s1 and s2, 1 is substituted for the mth bit of word data d when s1.S ≥ s2.S, and in all other cases, 0 is substituted.
- The range of s1.S and s2.S are as follows,

  when it is Word, the range is −32,768 to 32,767 (decimal number) and H8000 to H7FFF (hexadecimal number).

  when it is Double word, the range is −2,147,483,648 to 2,147,483,647 (decimal number) and H80000000 to H7FFFFFFF (hexadecimal number).
- The combination of d, s1 and s2 are as follows.

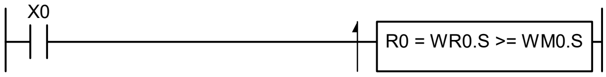
d	s1	s2
Bit	Word	Word
Bit	Double word	Double word

d.m	s1	s2
Word (the mth bit)	Word	Word
Word (the mth bit)	Double word	Double word

Cautionary notes

- Only the word internal output of WR can be specified to d.m_0.
- m_0 is from 0 to F.

Program example



[Program description]

When WR0.S ≥ WM0.S, R0 is set to 1 at the rising edge of X0. When WR0.S < WM0.S, R0 is reset to 0.

Name	>= Comparison expression (Floating point)

Ladder format	Number of steps				Condition code				
	Condition			Steps	R7F4	R7F3	R7F2	R7F1	R7F0
	d	s1	s2		DER	ERR	SD	V	C
d = s1.FL >= s2.FL d.m_0 = s1.FL >= s2.FL	B	DW	DW	7	●	●	●	●	●
	B(.m)	DW	DW	8					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d:B / s1, s2:DW	7.5	8.91	—	—	—
d:B.m / s1, s2:DW	7.18	8.47	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Substitution destination		✓	✓											
d.m_0	Substitution destination						✓								
s1.FL	Comparand											✓	✓	✓	✓
s2.FL	Comparative value											✓	✓	✓	✓

Remarks
B means Bit I/O, W means Word I/O, and DW means Double word I/O.

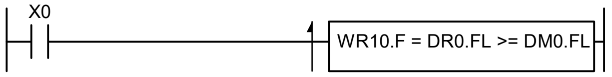
Function

- As floating point data in s1 and s2, 1 is substituted for d when s1.FL ≥ s2.FL, and in all other cases, 0 is substituted.
- As floating point data in s1 and s2, 1 is substituted for the mth bit of word data d when s1.FL ≥ s2.FL, and in all other cases, 0 is substituted.
- The range of s1 and s2 -3.40282×10^{38} to 3.40282×10^{38} (decimal number),
  HFF7FFFFF to H80800000 and H00800000 to H7F7FFFFF (hexadecimal number)

Cautionary notes

- Since there is an error in floating point, the error may cause the disagreement even if the value from the calculation is in agreement. We recommend deciding the comparison of floating point not in agreement and disagreement but in “range”.
- Only the word internal output of WR can specified to d.m_0.
- m_0 is from 0 to F.

Program example



[Program description]

When DR0.FL ≥ DM0.FL, the Fth bit of WR10 is set to 1 at the rising edge of X0.

When DR0.FL < DM0.FL, the Fth bit of WR10 is reset to 0.

Name	Data type conversion (Floating point ➔ Signed)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
d.S = INTG (s.FL)	—	5	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d : Word	3.85	4.47	—	—	—
d : Double word	7.51	9.12	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT, TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d.S	Conversion result							✓	✓	✓		✓	✓	
s.FL	Conversion source										✓	✓	✓	✓

Remarks

Function

- Floating point specified by s.FL is converted into signed integer.
  If d.S is Word, it is converted into word data. If d.S is Double word, it is converted into double word data.
- A form of substitution statement is written and the operation result is stored in d.S.
  Example1) WR10.S = INTG (DR0.FL)
  Converts DR0.FL (Floating point) into signed integer, and stores the result in WR10.S.
  Example2) DR10.S = INTG (DR0.FL)
  Converts DR0.FL (Floating point) into signed integer, and stores the result in DR10.S.

Parameter

d.S: Specifies the internal output (Word or Double word) to store the calculation result.

s.FL: Specifies an argument.

When d is Word,

if the value in outside of the range -32,768 < s.FL < 32,767 is specified, the operation is not performed because of DER = 1.

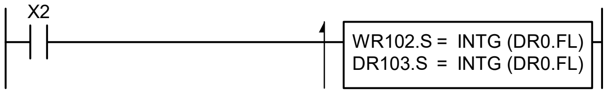
When d is Double word,

if the value in outside the range -2,147,483,648 < s.FL < 2,147,483,647 is specified, the operation is not performed because of DER = 1.

Cautionary notes

- Please specify the internal output used for argument and the internal output to store the calculation result within the range of the I/O No..
- The internal output of the conversion source needs an extension ".FL".
- The internal output of the conversion result needs an extension ".S".
- A format of floating point conforms to IEEE754.

Program example



[Program description]

Converts the floating point specified by DR0.FL into signed integer at the rising edge of X2, and sets the result in WR102 and DR103. (The figures below a decimal point are omitted.)

If X2 turns ON when DR0.FL is 123.456, both WR102 and DR103 are set to 123.

PRN ➔ PRJ

This command is equivalent to FUN100(s) / FUN101(s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 100 (s) / FUN101 (s) into the program for MICRO-EHV is as follows.

(1) FUN 100 (s) ➔ [s+2].S = INTG (s.FL), provided that s is Double word.

Example) FUN 100 (WR100) ➔ WR102.S = INTG (DR100.FL)

(2) FUN 101 (s) ➔ [s+2].S = INTG (s.FL), provided that s and s+2 are Double word.

Example) FUN 101 (WM10) ➔ DM12.S = INTG (DM10.FL)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	Data type conversion (Signed ➔ Floating point)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
d.FL = FLOAT (s.S)	d : Word	4	●	●	●	●	●
	d : Double word	5					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d : Word	3.93	4.63	—	—	—
d : Double word	4.59	5.43	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d.FL	Conversion result												✓	✓	
s.S	Conversion source							✓	✓	✓	✓	✓	✓	✓	✓

Remarks

Function

- Integer specified by s.S is converted into floating point.
- A form of substitution statement is written and the operation result is stored in d.FL.

Example 1) DR10.FL = FLOAT (WR0.S)

Converts WR0.S (signed integer) into floating point, and stores the result in DR10.FL.

Example 2) DR10.FL = FLOAT (DR0.S)

Converts DR0.S (signed integer) into floating point, and stores the result in DR10.FL.

Parameter

d.FL: Specifies the internal output (Double word) to store the calculation result.

s.S: Specifies an argument.

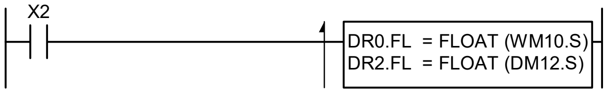
Since negative number is handled as two’s complement, convertible integer is the following ranges,

when it is Word,	−32,768 to 32,767
when it is Double word,	−2,147,483,648 to 2,147,483,647

Cautionary notes

- Please specify the internal outputs used for argument and the internal output to store the calculation result within the range of the I/O No..
- The internal output of the conversion source needs an extension “.S”.
- The internal output to store the calculation result needs an extension “.FL”.
- A format of floating point conforms to IEEE754.

Program example



[Program description]

Converts signed integer specified by WM10.S into floating point at the rising edge of X2, and sets the result in DR0.FL.

And converts signed integer specified by DM12.S into floating point, and sets the result in DR2.FL.

If X2 turns ON when WM10.S is −123 and DM12.S is 4,567,890, DR0.FL is set to −123 and DR2.FL is set to 4,567,890 in floating point format.

PRN ➔ PRJ

This command is equivalent to FUN102(s) / FUN103(s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 102 (s) / FUN103 (s) into the program for MICRO-EHV is as follows.

(1) FUN 102 (s) ➔ [s+1].FL = FLOAT (s.S), provided that s+1 is Double word.

Example) FUN 102 (WR100) ➔ DR101.FL = FLOAT (WR100.S)

(2) FUN 103 (s) ➔ [s+2].FL = FLOAT (s.S), provided that s and s+2 are Double word.

Example) FUN 103 (WM10) ➔ DM12.S = FLOAT (DM10.FL)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	Radian conversion (Floating point)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
d.FL = RAD (s.FL)	—	5	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	3.91	4.67	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT, TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d.FL	Conversion result (Radian)											✓	✓	
s.FL	Conversion source (Degree)										✓	✓	✓	✓

Remarks

Function

- Degree handling the floating point specified by s.FL as an argument is converted into Radian, and the result is set in d.FL.
- If the operation is performed normally, DER = 0.

Parameter

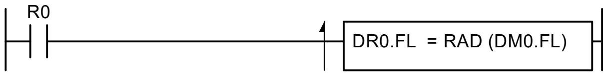
d.FL: Specifies the internal output (Double word) to store the calculation result.

s.FL: Specifies an argument.

Cautionary notes

- Please specify the internal output used for argument and the internal output to store the calculation result within the range of the I/O No.
- The internal outputs for the argument and for storing the calculation result need the extension ".FL".
- When argument is integer, please convert the integer into floating point before executing the operation. (Otherwise, you cannot get the correct calculation result.)
- When operation results are outside the range from −1e+37 to 1e+37, DER = 1.
- A format of floating point conforms to IEEE754.

Program example



[Program description]

Converts Degree specified by DM0.FL into Radian at the rising edge of R0, and sets the result in DR0.FL.

When DM0.FL is 30, if R0 turns ON, 0.5235 is stored in DR0.FL.

PRN ➔ PRJ

This command is equivalent to FUN108(s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 108 (s) into the program for MICRO-EHV is as follows.

FUN 108 (s) ➔ [s+2].FL = RAD (s.FL), provided that s and s+2 are Double word.

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	Degree conversion (Floating point)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
d.FL = DEG (s.FL)	—	5	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	4.63	5.43	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d.FL	Conversion result (Degree)											✓	✓	
s.FL	Conversion source (Radian)										✓	✓	✓	✓

Remarks

Function

- Radian handling the floating point specified by s.FL as an argument is converted into Degree, and the result is set in d.FL.
- If the operation is performed normally, DER = 0.

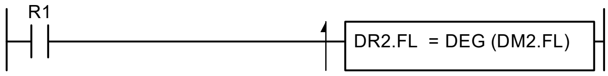
Parameter

d.FL: Specifies the internal output (Double word) to store the calculation result.
s.FL: Specifies an argument.

Cautionary notes

- Please specify the internal output used for argument and the internal output to store the calculation result within the range of the I/O No.
- The internal outputs for argument and for storing the calculation result need an extension ".FL".
- When argument is integer, please convert the integer into floating point before executing the operation. (Otherwise, you cannot get correct result of calculation.)
- When the operation result is outside the range from −1e+37 to 1e+37, DER = 1.
- A format of floating point conforms to IEEE754.

Program example



[Program description]

Converts Radian specified by DM2.FL into Degree at the rising edge of R1, and sets the result in DR2.FL. When DM2.FL is 3.14, if R1 turns ON, 179.9 is stored in DR2.FL.

PRN ➔ PRJ

This command is equivalent to FUN109(s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 109 (s) into the program for MICRO-EHV is as follows.

FUN 109 (s) ➔ [s+2].FL = DEG (s.FL), provide that s and s+2 are Double word.

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	Absolute value

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
d = ABS (s.S)	Word	4	●	●	●	●	↕
	Double word	5					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d : Word	3.71	4.43	–	–	–
d : Double word	4.71	5.63	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O to store absolute value							✓	✓	✓		✓	✓	
s.S	I/O to take absolute value						✓	✓	✓	✓	✓	✓	✓	✓

Remarks

Function

- Handling s.S as the signed parameter, an absolute value is set in d.
- When s.S is positive or 0, : a content of s is stored in d. C(R7F0) is set to '0'.
- When s.S is negative, : two's complement of a content of s is stored in d. C(R7F0) is set to '1'.
- The combination of d and s.S are as follows.

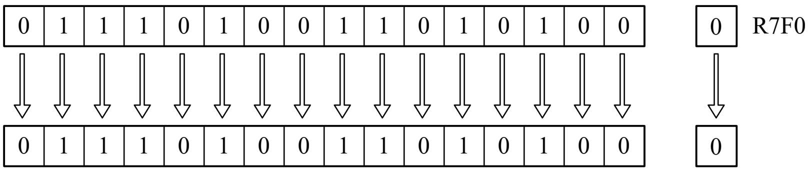
d	s.S
Word	Word
Double word	Double word

- Case of Word: From −32,768 to −1 correspond to from H8000 to HFFFF.

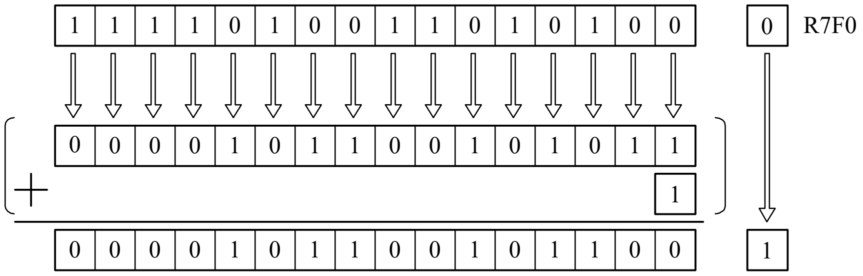
  From 0 to 32,767 correspond to from H0000 to H7FFF.
- Case of Double word: From −2,147,483,648 to −1 correspond to from H80000000 to HFFFFFFFF.

  From 0 to 2,147,483,647 correspond to from H00000000 to H7FFFFFFF.

When a value of 's' is positive or 0,



When a value of 's' is negative,



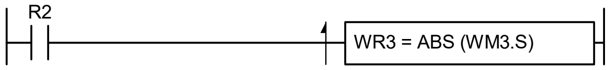
Parameter

d: Specifies the internal output to store the calculation result.
s.S: Specifies an argument.

Cautionary notes

Please set a startup condition to an edge trigger.

Program example



[Program description]

Sets the absolute value of WM3.S in WR3 at the rising edge of R2.

When WM3.S is −12,345, if R2 turns ON, 12,345 is stored in WR3.

PRN ➔ PRJ

This command is equivalent to ABS (d, s) in the program (PRN file) of MICRO-EH.
How to convert the program whish has used ABS (d, s) into the program for MICRO-EHV is as follows.

ABS (d, s) ➔ d = ABS (s.S)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	Sign addition

Ladder format
d.S = SGET (s)

Number of steps	
Condition	Steps
Word	4
Double word	5

Condition code				
R7F4	R7F3	R7F2	R7F1	R7F0
DER	ERR	SD	V	C
●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d : Word	3.5	4	—	—	—
d : Double word	4.22	4.73	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d.S	Storage I/O after adding sign								✓	✓	✓		✓	✓	
s	I/O to add sign							✓	✓	✓	✓	✓	✓	✓	✓

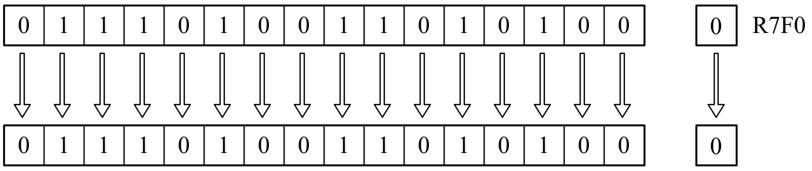
Remarks

Function

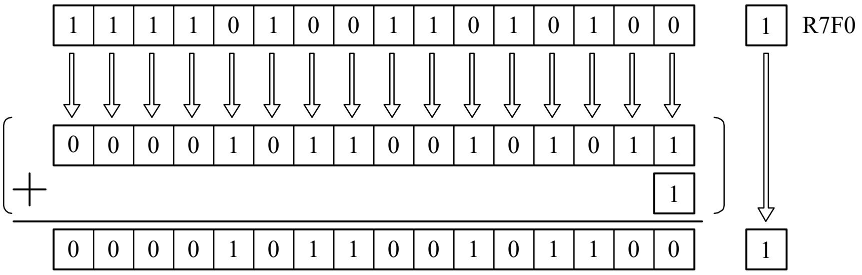
- When C(R7F0) is 0: a content of s is stored in d.S.
- When C(R7F0) is 1: two's complement of a content of s is stored in d.S.
- C(R7F0) remains unchanged.
- The combination of d and s are as follows.

d	S
Word	Word
Double word	Double word

When C(R7F0) is 0,



When C(R7F0) is 1,



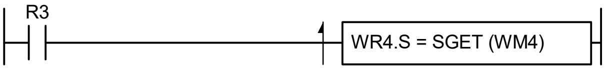
Parameter

d.S: Specifies the internal output to store the calculation result.
s: Specifies an argument.

Cautionary notes

Please set a startup condition to an edge trigger.

Program example



[Program description]

Adds the sign to the value of WM4, and sets the result in WR4.S at the rising edge of R3.

When WM4 is 12,345 and C(R7F0) is 0, if R3 turns ON, 12,345 is stored in WR4.S.

When WM4 is 12,345 and C(R7F0) is 1, if R3 turns ON, -12,345 is stored in WR4.S.

PRN ➔ PRJ

This command is equivalent to SGET (d, s) in the program (PRN file) of MICRO-EH.
How to convert the program which has used SGET (d, s) into the program for MICRO-EHV is as follows.

SGET (d, s) ➔ d.S = SGET (s)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	Bit extension

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
d.S = EXT (s.S, n)	Word	5	●	●	●	●	●
	Double word	5					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d : Word	4.22	4.62	—	—	—
d : Double word	4.22	4.62	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d.S	Storage I/O after bit extension							✓	✓	✓		✓	✓	
s.S	I/O before bit extension						✓	✓	✓	✓				✓
n	Position of sign bit													✓

Remarks

Function

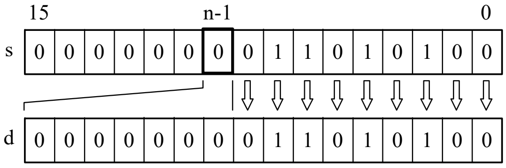
- The signed bit (the n-1th bit) of s.S is extended to d.S.

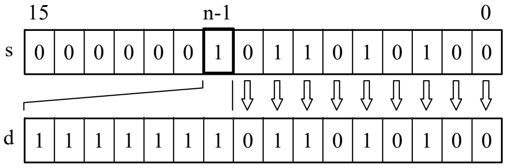
When d.S is Word:	the value of the n-1th bit is stored in from the n-1th bit to MSB of d.S.
When d.S is Double word:	the value of the n-1th bit is stored in from the n-1th bit of d.S to MSB of upper word.

- The combination of d.S and s.S are as follows.

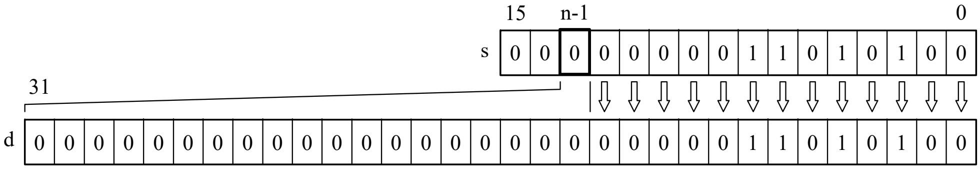
d.S	s.S
Word	Word
Double word	

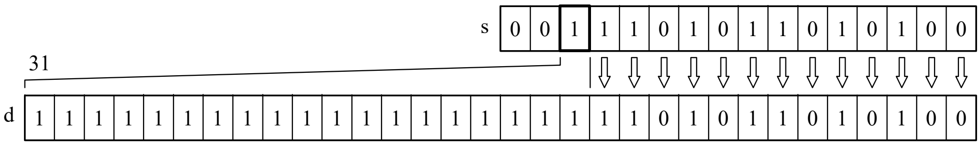
When s.S is Word and d.S is Word,





when s.S is Word and d.S is Double word,





Parameter

d.S: Specifies the internal output to store the calculation result.

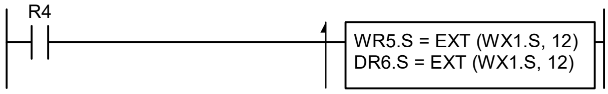
s.S: Specifies an argument.

n: Specifies the bit position of the signed bit.

Cautionary notes

n specifies the number of bits to extend. Please specify 12 when extending the 12-bit analog data and specify 14 when extending the 14-bit analog data.

Program example



[Program description]

Extends the signed bit (the 12th bit, b11) of the value of WX1.S to the upper bit and stores the result in WR5.S and DR6.S at the rising edge of R4.

When WX1.S is H7FF, if R4 turns ON, H07FF is stored in WR5.S.

When WX1.S is H800, if R4 turns ON, HFFFFF800 is stored in DR6.S.

PRN ➔ PRJ

This command is equivalent to EXT (d, s) in the program (PRN file) of MICRO-EH.
How to convert the program which has used EXT (d, s) into the program for MICRO-EHV is as follows.

EXT (d, s) ➔ d.S = EXT (s.S, 16), provided that d.S is Double word,

* if converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	Two's complement

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
d = NEG (s)	Word	4	●	●	●	●	●
	Double word	5					

Command processing time (µs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d : Word	2.96	3.73	—	—	—
d : Double word	3.58	4.46	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O after taking two'scomplement							✓	✓	✓		✓	✓	
s	I/O to take two's complement						✓	✓	✓	✓	✓	✓	✓	✓

Remarks

Function

- Two's complement of d is calculated. (1 is added after reversing a content of d. C(R7F0) remains unchanged.)

	15															0
s	1	1	1	1	0	0	0	0	1	1	0	0	1	0	1	0
	⇩	⇩	⇩	⇩	⇩	⇩	⇩	⇩	⇩	⇩	⇩	⇩	⇩	⇩	⇩	⇩
	0	0	0	0	1	1	1	1	0	0	1	1	0	1	0	1
+																1
d	0	0	0	0	1	1	1	1	0	0	1	1	0	1	1	0

Parameter

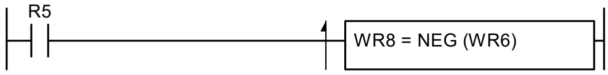
d: Specifies the internal output to store the calculation result.

s: Specifies an argument.

Cautionary notes

- Please set a startup condition to the edge trigger.
- When you want to set the I/O to find two's complement and the I/O to substitute the result to the same I/O, please set d and s to the same I/O.

Program example



[Program description]

Finds two’s complement of the value of WR6, and substituted the result for WR8 at the rising edge of R5.

When WR6 is H1234, if R5 turns ON, HEDCC is stored in WR8.

PRN ➔ PRJ

This command is equivalent to NEG (d) in the program (PRN file) of MICRO-EH.
How to convert the program which has used NEG (d) into the program for MICRO-EHV is as follows.

NEG (d) ➔ d = NEG (d)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	Binary square root

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
d = SQR (s) d.FL = SQR (s.FL)	Integer	5	↕	●	●	●	●
	Floating point	5					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
Integer	11.66	15.24	—	—	—
Floating point	17.10	21.46	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Calculation result								✓	✓					
d.FL	Calculation result												✓	✓	
s	Object to be calculated											✓	✓	✓	✓
s.FL	Object to be calculated											✓	✓	✓	✓

Remarks

Function

- If s is specified by Double word without an extension, a square root handling the 32-bit unsigned binary value as an argument is calculated. (Figures below a decimal point are omitted.)
- If s is specified by Double word of floating point (with an extension '.FL'), a square root handling floating point as an argument is calculated.
- A form of substitution statement is written and the operation result is stored in d or d.FL.

  Example) WR0 = SQR (DR10) Calculates a square root handling DR10 as an argument, and stores the result in WR0.
- If the operation is performed normally, DER = 0.

Parameter

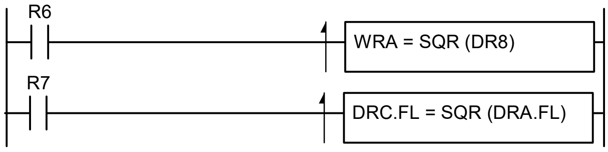
d: Specifies an internal output to store the calculation result.

s: Specifies an argument.

Cautionary notes

- Please specify the internal outputs to store object to be calculated and the calculation result within the range of the I/O No.
- If object to be calculated is specified to floating point, please specify the calculation result to floating point also. And if object to be calculated is specified to integer, the calculation result is also stored with integer.
- When computing with floating point, please convert object to be calculated from integer into floating point before executing the operation.
- When computing with floating point, if the operation results is outside the range from −1e+37 to 1e+37, DER = 1.
- A format of floating point conforms to IEEE754.

Program example



[Program description]

- Finds a square root of DR8 and sets the result in WRA at the rising edge of R6.
  (DR8 is handled as integer. The calculation result also becomes integer.)
- Finds a square root of DRA.FL and sets the result in DRC.FL at the rising edge of R7.
  (DRC.FL is handled as floating point. The calculation result also becomes floating point.)

PRN ➔ PRJ

This command is equivalent to FUN 60(s) [square root of integer] / FUN 116 (s) [square root of floating point] in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 60 (s) / FUN 116 (s) the program for MICRO-EHV is as follows.

(1) FUN 60 (s) ➔ s+2 = SQR (s), provided that s is Double word.

Example) FUN 60 (WR100) ➔ WR102 = SQR (DR100)

(2) FUN 116 (s) ➔ [s+2].FL = SQR (s.FL), provided that s and s+2 are Double word.

Example) FUN 116 (WR100) ➔ DR102.FL = SQR (DR100.FL)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	BCD Square root

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
d = BSQR (s)	–	5	↕	●	●	●	●

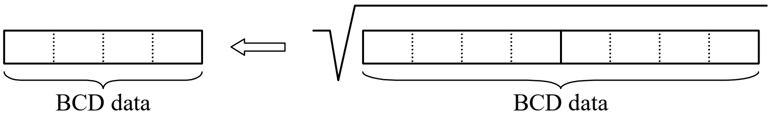
Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	14.96	17.68	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Calculation result (BCD)								✓	✓					
s	Object to be calculated (BCD)											✓	✓	✓	✓

Remarks

Function

- A square root of a content of s is calculated and the result is output to d.
- Please set BCD data in s.
- Figures below a decimal point are omitted.



Parameter

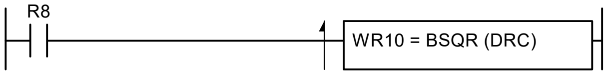
d: Specifies an internal output to store calculation result.

s: Specifies an argument.

Cautionary notes

When s is BCD data error (including values from HA to HF), the operation is not performed because DER(R7F4) becomes 1.

Program example



[Program description]

Finds a square root of DRC and sets the result in WR10 in BCD data at the rising edge of R8.

PRN ➔ PRJ

This command is equivalent to SQR (d, s) in the program (PRN file) of MICRO-EH.

How to convert the program whish has used SQR (d, s) into the program for MICRO-EHV is as follows.

SQR (d, s) ➔ d = BSQR (s), provided that s is Double word.

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	Exponentiation

Ladder format
d = POW (s, n) d.FL = POW (s.FL, n.FL)

Number of steps	
Condition	Steps
Integer	5
Floating point	7

Condition code				
R7F4	R7F3	R7F2	R7F1	R7F0
DER	ERR	SD	V	C
↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
Integer	6.2	7.12	—	—	—
Floating point	17.42	21.48	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT, TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Calculation result												✓	
d.FL	Calculation result												✓	
s	Object to be calculated (Base)						✓	✓	✓	✓				✓
s.FL	Object to be calculated (Base)										✓	✓	✓	✓
n	Object to be calculated (Exponent)						✓	✓	✓	✓				✓
n.FL	Object to be calculated (Exponent)										✓	✓	✓	✓

Remarks

Function

- Handling the unsigned binary value specified by s and the exponent (binary) specified by n as argument, exponentiation is calculated.
- Handling the floating point specified by s.FL and the exponent (floating point) specified by n.FL, exponentiation is calculated.
- A form of substitution statement is written and the operation result is stored in d or d.FL.

  Example 1) DR10 = POW (WR1, WR2): calculates WR1^WR2 and stores the result in DR10.

  Example 2) DR10.FL = POW (DR0.FL, DR2.FL): calculates DR0.FL^DR2.FL and stores the result in DR10.FL.
- If the operation is performed normally, DER = 0.

Parameter

d / d.FL: Specifies the internal output to store the calculation result.

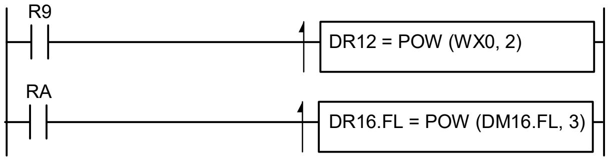
s / s.FL: Specifies the internal output which has stored object to be calculated (base).

n / n.FL: Specifies the internal output which has stored object to be calculated (exponent).

Cautionary notes

- Please specify the internal output used for argument and the internal output for the calculation result within the range of the I/O No.
- In the floating point operation, the internal outputs to store object to be calculated and the calculation result need an extension ".FL".
- In the floating point operation, please convert object to be calculated from integer into floating point before executing the operation. (Otherwise, you cannot get the correct calculation result.)
- In the unsigned binary value, if the operation result is outside the range from 0 to 4,294,967,295, DER = 1.
- In the floating point, if the operation result is outside the range from −1e+37 to 1e+37, DER = 1.
- A format of floating point conforms to IEEE754.
- When s.FL and n.FL are s.FL = 0 and n.FL ≤ 0 in floating point arithmetic, DER = 1 because calculation is not possible.
- When s.FL and n.FL are s.FL < 0 and n.FL is not an integer in floating point arithmetic, DER = 1 because calculation is not possible.

Program example



[Program description]

- Squares the value of WX0 and substitutes the result for DR12 at the rising edge of R9.
  (WX0 is handled as integer. The calculation result also becomes integer.)
- Cubes the value of DM16.FL and substitutes the result for DR16.FL at the rising edge of RA.
  (DM16.FL is handled as floating point. The calculation result also becomes floating point.)

Name	Sine operation (Degree)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
d = SIN (s)	—	4	↕	●	●	●	●

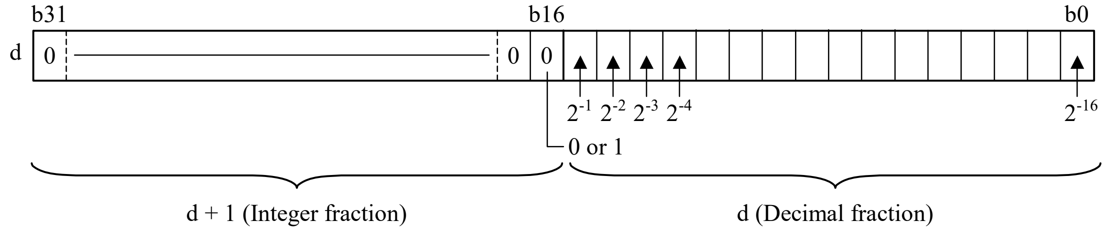
Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	3.98	4.87	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Calculation result												✓	✓	
s	Argument							✓	✓	✓	✓				✓

Remarks

Function

- Sine function handling the unsigned binary value (Degree) specified by s as an argument is calculated.
- A form of substitution statement is written and the operation result is stored in d (Double word: upper word is stored in the integer fraction and lower word is stored in the decimal fraction).



Example) DR100 = SIN (WR0)

Calculates the sine function handling WR0 as an argument and stores the result in DR100.

- The operation result is represented by the binary value. And the negative number is represented by two's complement. (The operation is performed normally, DER = 0.)
- A decimal fraction data (d parameter lower word) = the real number value × 65,535.

Parameter

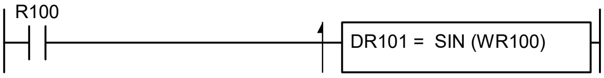
d: Specifies the internal output to store the calculation result.

s: Specifies an argument (Degree). The range is $0 \leq s \leq 360$.
(If the value of s parameter is outside the range, the operation is not performed because of DER = 1.)

Cautionary notes

- Please specify the internal output used for argument and the internal output to store the calculation result within the I/O No.
- Although a decimal point fraction is contained in the calculation result, it differs from the floating point format of IEEE754.

Program example



[Program description]

Calculates the sine function of WR100 at the rising edge of R100, and substitutes the result for DR101 (WR101: decimal fraction, WR102: integer fraction).

PRN ➔ PRJ

This command is equivalent to FUN 10 (s) in the program of MICRO-EH (PRN file).

How to convert the program which has used FUN 10 (s) into the program for MICRO-EHV is as follows.

FUN 10 (s) ➔ s+1 = SIN (s), provided that s+1 is Double word.

Example) FUN 10 (WR100) ➔ DR101 = SIN (WR100)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	Sine operation (Radian)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
d.FL = SINR (s.FL)	—	5	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	19.84	25.04	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d.FL	Calculation result												✓	✓	
s.FL	Argument											✓	✓	✓	✓

Remarks

Function

- Sine function handling floating point (Radian) specified by s.FL as an argument is calculated.
- A form of substitution statement is written and the result is stored in d.FL.

  Example) DR100.FL = SINR (DR0.FL)

  Calculates the sine function handling DR0.FL as an argument and stores the result in DR100.FL
- The operation result is represented by floating point. (If the operation is performed normally, DER = 0.)

Parameter

d.FL: Specifies the internal output to store the calculation result.

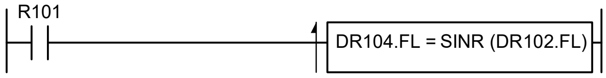
s.FL: Specifies an argument (Radian).

If the value which becomes s > 1.4148475504056880000e+16 is specified, the operation is not performed because of DER=1.

Cautionary notes

- Please specify the internal output used for argument and the internal output to store the calculation result within the range of the I/O No.
- The internal outputs for argument and to store the calculation result need an extension ".FL"
- If an argument is integer, please convert it from integer into the floating point before executing the operation. (Otherwise, you cannot get the correct calculation result.)
- If the operation result is outside the range form −1e+37 to 1e+37, DER = 1.
- If the value which becomes s.FL > 2.9815682600000000000e+08 is specified, the operation is performed but the accuracy goes down. (The operation result comes out but DER = 1.)
- A format of floating point conforms to IEEE754.

Program example



[Program description]

Calculates the sine function of DR102.FL at the rising edge of R101, and substitutes the result for DR104.FL.

PRN ➔ PRJ

This command is equivalent to FUN 110 (s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 110 (s) into the program for MICRO-EHV is as follows.

FUN 110 (s) ➔ [s+2].FL = SINR (s.FL), provided that s and s+2 are Double word.

Example) FUN 110 (WR100) ➔ DR102.FL = SINR (DR100.FL)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	Cosine operation (Degree)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
d = COS (s)	—	4	↕	●	●	●	●

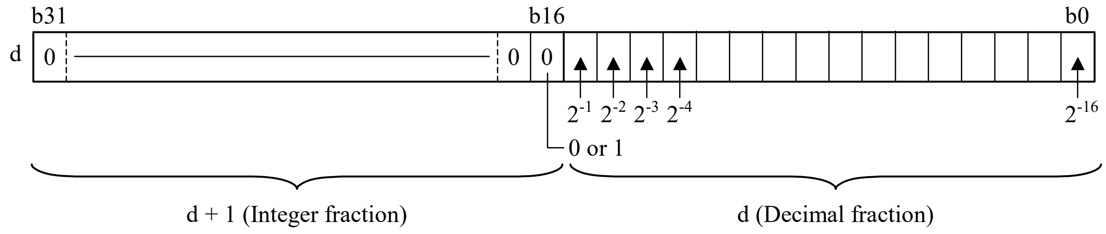
Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	4.3	4.87	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Calculation result												✓	✓	
s	Argument							✓	✓	✓	✓				✓

Remarks

Function

- Cosine function handling the unsigned binary value (Degree) specified by s as an argument is calculated.
- A form of substitution statement is written and the operation result is stored in d (Double word: upper word is stored in an integer fraction and lower is stored in a decimal fraction).



Example) DR100 = COS (WR0)

Calculates the cosine function handling WR0 as an argument, and stores the result in DR100.

- The operation result is represented by the binary value. And the negative number is represented by two's complement. (If the operation is performed normally, DER=0.)
- A decimal fraction data (d parameter lower word) = the real number value × 65,535.

Parameter

d: Specifies the internal output to store the calculation result.

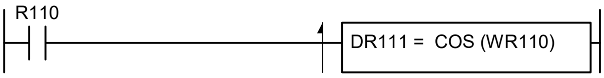
s: Specifies an argument (Degree). The range is $0 \leqq s \leqq 360$.

(If the value of s parameter is outside the range, the operation is not performed because of DER=1.)

Cautionary notes

- Please specify the internal output used for argument and the internal output to store the calculation result within the range of the I/O No.
- Although a decimal point fraction is contained in a result of calculation, it differs from the floating point format of IEEE754.

Program example



[Program description]

Calculates the cosine function of WR110 at the rising edge of R110, and substitutes the result for DR111 (WR111: a decimal fraction, WR112: an integer fraction).

PRN ➔ PRJ

This command is equivalent to FUN 11 (s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 11 (s) into the program for MICRO-EHV is as follows.

FUN 11 (s) ➔ s+1 = COS (s), provided that s+1 is Double word.

Example) FUN 11 (WR100) ➔ DR101 = COS (WR100)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	Cosine operation (Radian)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
d.FL = COSR (s.FL)	—	5	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	54.9	68.1	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d.FL	Calculation result												✓	✓	
s.FL	Argument											✓	✓	✓	✓

Remarks

Function

- Cosine function handling floating point (radian) specified by s.FL as an argument is calculated.
- A form of substitution statement is written and the operation result is stored in d.FL.

  Example) DR100.FL = COSR (DR0.FL)

  Calculates the cosine function handling DR0.FL as an argument, and stores the result in DR100.FL.
- The operation result is represented by floating point. (The operation is performed normally, DER = 0.)

Parameter

d.FL: Specifies the internal output to store the calculation result.

s.FL: Specifies an argument (Radian).

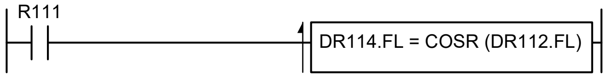
If the value which becomes s > 1.414847550405688000e+16 is specified, the operation is not performed because of DER = 1.

Cautionary notes

- Please specify the internal output used for argument and the internal output to store the calculation result within the range of the I/O No.
- The internal outputs for argument and to store the calculation result need an extension ".FL".
- If an argument is integer, please convert it from integer into the floating point before executing the operation. (Otherwise, you cannot get the correct calculation result.)
- If the operation result is outside the range from −1e+37 to 1e+37, DER = 1.
- If the value which becomes s.FL > 2.981568260000000000e+08 is specified, the accuracy goes down but the accuracy goes down.

  (Although the operation result comes out, DER = 1.)
- A format of floating point conforms to IEEE754.

Program example



[Program description]

Calculates the cosine function of DR112.FL at the rising edge of R111, and substitutes the result for DR114.FL.

PRN ➔ PRJ

This command is equivalent to FUN 111 (s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 111 (s) into the program for MICRO-EHV is as follows.

FUN 111 (s) ➔ [s+2].FL = COSR (s.FL), provided that s and s+2 are Double word.

Example) FUN 111 (WR100) ➔ DR102.FL = COSR (DR100.FL)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	Tangent operation (Degree)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
d = TAN (s)	–	4	↕	●	●	●	●

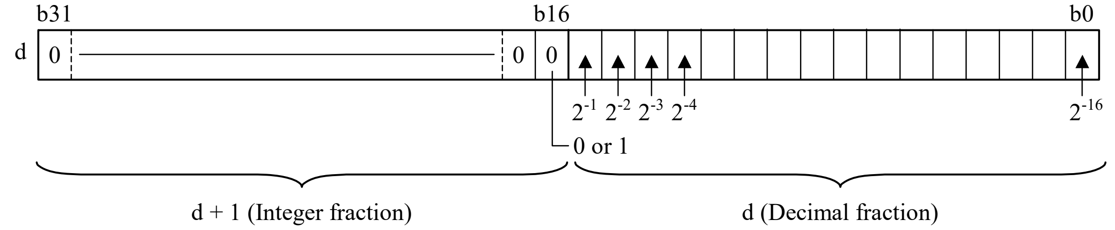
Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	4.26	4.91	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Calculation result											✓	✓	
s	Argument						✓	✓	✓	✓				✓

Remarks

Function

- Tangent function handling the unsigned binary value (Degree) specified by s as an argument is calculated.
- A form of substitution statement is written and the operation result is stored in d (Double word: upper word is stored in a integer fraction and lower is stored in a decimal fraction).



Example) DR100 = TAN (WR0)

Calculates the tangent function handling WR0 as an argument, and stores the result in DR100.

- The operation result is represented by binary value. And the negative number is represented by two's complement. (If the operation is performed normally, DER = 0.)
- A decimal fraction data ('d' parameter lower word) = the real number value × 65,535.

Parameter

d: Specifies the internal output to store the calculation result.

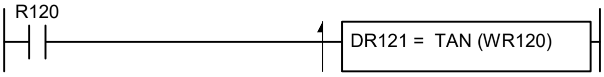
s: Specifies an argument (Degree). The range is $0 \leq s \leq 360$ (except 90 and 270).

(The value of s parameter is outside the range, the operation is not performed because DER = 1. And if s = 90 or s = 270, H7FFFFFFF is stored in the internal output to store the calculation result because of DER = 1.)

Cautionary notes

- Please specify the internal outputs used for argument and the internal output to store the calculation result within the range of the I/O No..
- Although a decimal points fraction is contained in the calculation result, it differs from a floating point format of IEEE754.

Program example



[Program description]

Calculates the tangent function of WR120 at the rising edge of R120, and substitutes the result for DR121 (WR121: a decimal fraction, WR122: an integer fraction).

PRN ➔ PRJ

This command is equivalent to FUN 12 (s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 12 (s) into the program for MICRO-EHV is as follows.

FUN 12 (s) ➔ s+1 = TAN (s), provided that s+1 is Double word.

Example) FUN 12 (WR100) ➔ DR101 = TAN (WR100)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	Tangent operation (Radian)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
d.FL = TANR (s.FL)	—	5	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	19.48	24.86	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d.FL	Calculation result											✓	✓	
s.FL	Argument										✓	✓	✓	✓

Remarks

Function

- Tangent function handling floating point (Radian) specified by s.FL as an argument is calculated.
- A form of substitution statement is written and the operation result is stored in d.FL.

  Example) DR100.FL = TANR (DR0.FL)

  Calculates the tangent function handling DR0.FL as an argument, and stores the result in DR100.FL.
- The operation result is represented by floating point. (If the operation is performed normally, DER = 0.)

Parameter

d.FL: Specifies the internal output to store the calculation result.

s.FL: Specifies an argument (Radian).

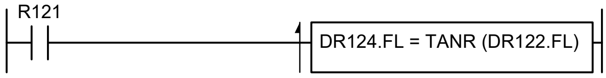
If the value which becomes s.FL > 1.414847550405688000e+16 is specified, the operation is not performed because of DER=1.

Cautionary notes

- Please specify the internal output used for argument and the internal output to store the calculation result within the range of the I/O No.
- The internal outputs for argument and to store the calculation result need an extension ".FL".
- If an argument is integer, please convert it from integer into the floating point before executing the operation. (Otherwise, you cannot get the correct calculation result.)
- If the operation result is outside the range from −1e+37 to 1e+37, DER = 1.
- If the value which becomes s.FL > 2.981568260000000000e+08 is specified, the operation is performed but the accuracy goes down .

  (Although the operation result comes out, DER = 1.)
- A format of floating point conforms to IEEE754.

Program example



[Program description]

Calculates the tangent function of DR122.FL at the rising edge of R121, and substitutes the result for DR124.FL.

PRN ➔ PRJ

This command is equivalent to FUN 112 (s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 112 (s) into the program for MICRO-EHV is as follows.

FUN 112 (s) ➔ [s+2].FL = TANR (s.FL), provided that s and s+2 are Double word.

Example) FUN 112 (WR100) ➔ DR102.FL = TANR (DR100.FL)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	Arc sine operation (Degree)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
d = ASIN (s)	—	5	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	8.10	9.31	—	—	—

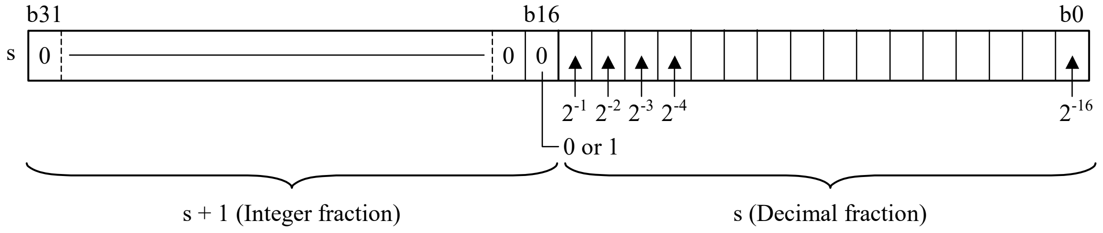
Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Calculation result								✓	✓	✓				
s	Argument											✓	✓	✓	

Remarks

Function

- Arc sine function (SIN^{-1}) handling the unsigned binary value specified by s (Double word: upper word is an integer fraction and lower is a decimal fraction) as an argument is calculated.

A decimal fraction data (s parameter lower word) = the real number value × 65,535.



- A form of substitution statement is written and the operation result is stored in d.

Example) WR200 = ASIN (DR2)

Calculates SIN^{-1} handling DR2 as an argument, and stores the result in WR200.

- If the operation is performed normally, DER = 0.
- The calculation result is the binary value and an angle (Degree) from 0 to 90 and from 180 to 270.

Parameter

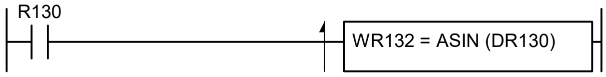
d: Specifies the internal output to store the calculation result.

s: Specifies an argument. Please set a decimal point data (s parameter lower word) = the real number value × 65,535. If |s| > 1, the operation is not performed because of DER = 1.

Cautionary notes

- Please specify the internal output used for argument and the internal output to store the calculation result within the range of the I/O No.
- Although a decimal point fraction is also contained in the argument, it differs from a floating point format of IEEE754.

Program example



[Program description]

Calculates SIN^{-1} of DR130 (WR130: a decimal fraction, WR131: an integer fraction) at the rising edge of R130, and substitutes the result for WR132.

PRN ➔ PRJ

This command is equivalent to FUN 13 (s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 13 (s) into the program for MICRO-EHV is as follows.

FUN 13 (s) ➔ s+2 = ASIN (s), provided that s is Double word.

Example) FUN 13 (WR100) ➔ WR102 = ASIN (DR100)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	Arc sine operation (Radian)

<table>
<tr><th rowspan="3">Ladder format</th><th colspan="2">Number of steps</th><th colspan="5">Condition code</th></tr>
<tr><th rowspan="2">Condition</th><th rowspan="2">Steps</th><th>R7F4</th><th>R7F3</th><th>R7F2</th><th>R7F1</th><th>R7F0</th></tr>
<tr><th>DER</th><th>ERR</th><th>SD</th><th>V</th><th>C</th></tr>
<tr><td>d.FL = ASINR (s.FL)</td><td>—</td><td>5</td><td>↕</td><td>●</td><td>●</td><td>●</td><td>●</td></tr>
</table>

<table>
<tr><th colspan="6">Command processing time (μs)</th></tr>
<tr><th colspan="3">Average</th><th colspan="3">Maximum</th></tr>
<tr><th rowspan="2">Condition</th><th colspan="2">Time</th><th rowspan="2">Condition</th><th colspan="2">Time</th></tr>
<tr><th>MVH
(High function)</th><th>MVL
(Standard)</th><th>MVH
(High function)</th><th>MVL
(Standard)</th></tr>
<tr><td>—</td><td>15.28</td><td>19.16</td><td>—</td><td>—</td><td>—</td></tr>
</table>

<table>
<tr><th colspan="2" rowspan="2">Usable I/O</th><th colspan="6">Bit</th><th colspan="4">Word</th><th colspan="3">Double word</th><th rowspan="2">Constant</th></tr>
<tr><th>X</th><th>Y</th><th>R,M</th><th>TD,
SS,
MS,
CU,
CT</th><th>TDN,
WDT,
TMR,
RCU,</th><th>WR,
(.m)</th><th>WX</th><th>WY</th><th>WR, WM</th><th>TC</th><th>DX</th><th>DY</th><th>DR,DM</th></tr>
<tr><td>d.FL</td><td>Calculation result</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>✓</td><td>✓</td><td></td></tr>
<tr><td>s.FL</td><td>Argument</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>✓</td><td>✓</td><td>✓</td><td>✓</td></tr>
</table>

Remarks

Function

- Arc sine function (SIN^{-1}) handling floating point specified by s.FL as an argument is calculated.
- A form of substitution statement is written and the operation result is stored in d.FL with Radian unit.

  Example) DR200.FL = ASINR (DR2.FL)

  Calculates SIN^{-1} handling DR2.FL as an argument, and stores the result in DR200.FL.
- The operation result is an angle system of Radian units, and represented by floating point.
- If the operation is performed normally, DER = 0.

Parameter

d.FL: Specifies the internal output to store the calculation result.

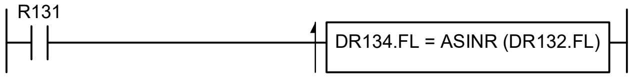
s.FL: Specifies an argument which has stored the real number to find SIN^{-1}.

If the value which becomes s.FL > |1.0| is specified, the operation is not performed because of DER = 1.

Cautionary notes

- Please specify the internal output used for argument and the internal output to store the calculation result within the range of the I/O No.
- The internal outputs for argument and to store the calculation result need an extension ".FL".
- If an argument is integer, please convert it from the integer into the floating point before executing the operation. (Otherwise, you cannot get the correct calculation result.)
- If the operation result is outside the range from −1e+37 to 1e+37, DER = 1.
- A format of floating point conforms to IEEE754.

Program example



[Program description]

Calculates SIN^{-1} of DR132.FL at the rising edge of R131, and substitutes the result for DR134.FL.

PRN ➔ PRJ

This command is equivalent to FUN 113 (s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 113 (s) into the program for MICRO-EHV is as follows.

FUN 113 (s) ➔ [s+2].FL = ASINR (s.FL), provided that s and s+2 are Double word.

Example) FUN 113 (WR100) ➔ DR102.FL = ASINR (DR100.FL)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	Arc cosine operation (Degree)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
d = ACOS (s)	—	5	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	8.48	9.38	—	—	—

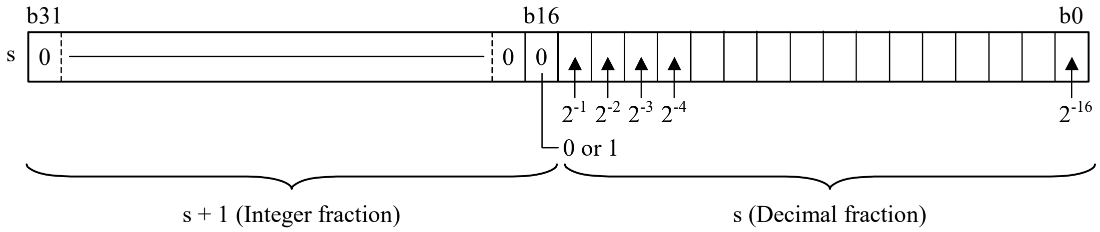
Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Calculation result								✓	✓	✓				
s	Argument											✓	✓	✓	

Remarks

Function

- Arc cosine function (COS^{-1}) handling the unsigned binary value specified by s (Double word: upper word is integer fraction, lower is decimal fraction) as an argument is calculated.

  A decimal fraction data (s parameter lower word) = the real number value × 65,535.



- A form of substitution statement is written and the operation result is stored in d.

  Example) WR200 = ACOS (DR2)

  Calculates COS^{-1} function handling DR2 as an argument, and stores the result in WR200.
- If the operation is performed, DER = 0.
- The calculation result is the binary value and an angle (Degree) from 0 to 180.

Parameter

d: Specifies the internal output to store the calculation result.

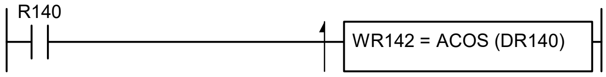
s: Specifies an argument. A decimal point data (s parameter lower word) = the real number value × 65,535.

If |s| > 1, the operation is not performed because of DER = 1.

Cautionary notes

- Please specify the internal output used for argument and the internal output to store the calculation result within the range of the I/O No.
- Although a decimal point fraction is contained in an argument, it differs from a floating point format of IEEE754.

Program example



[Program description]

Calculates COS^{-1} of DR140 (WR140: a decimal fraction, WR141: an integer fraction) at the rising edge of R140, and substitutes the result for WR142.

PRN ➔ PRJ

This command is equivalent to FUN 14 (s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 14 (s) into the program for MICRO-EHV is as follows.

FUN 14 (s) ➔ s+2 = ACOS (s), provided that s is Double word.

Example) FUN 14 (WR100) ➔ WR102 = ACOS (DR100)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	Arc cosine operation (Radian)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
d.FL = ACOSR (s.FL)	—	5	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	22.34	27.62	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d.FL	Calculation result												✓	✓	
s.FL	Argument											✓	✓	✓	✓

Remarks

Function

- Arc cosine function (COS^{-1}) handling a floating point specified by s.FL as an argument is calculated.
- A form of substitution statement is written and the operation result is stored in d.FL with Radian units.

  Example) DR200.FL = ACOSR (DR2.FL)

  Calculates COS^{-1} function handling DR2.FL as an argument, and stores the result in DR200.FL.
- The operation result is an angle of Radian units system and represented by floating point.
- If the operation is performed normally, DER = 0.

Parameter

d.FL: Specifies the internal output to store the calculation result.

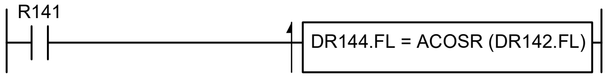
s.FL: Specifies an argument which has stored the floating point to find COS^{-1}.

If the value which becomes s.FL > |1.0| is specified, the operation is not performed because of DER = 1.

Cautionary notes

- Please specify the internal output used for argument and the internal output to store the calculation result within the range of the I/O No.
- The internal outputs for argument and to store the calculation result need an extension ".FL".
- If an argument is integer, please convert it from integer into the floating point before executing the operation. (Otherwise, you cannot get the correct calculation result.)
- If the operation result is outside the range from −1e+37 to 1e+37, DER = 1.
- A format of floating point conforms to IEEE754.

Program example



[Program description]

Calculates COS^{-1} of DR142.FL at the rising edge of R141, and substitutes the result for DR144.FL.

PRN ➔ PRJ

This command is equivalent to FUN 114 (s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 114 (s) into the program for MICRO-EHV is as follows.

FUN 114 (s) ➔ [s+2].FL = ACOSR (s.FL), provided that s and s+2 are Double word.

Example) FUN 114 (WR100) ➔ DR102.FL = ACOSR (DR100.FL)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	Arc tangent operation (Degree)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
d = ATAN (s)	–	5	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	7.52	8.54	–	–	–

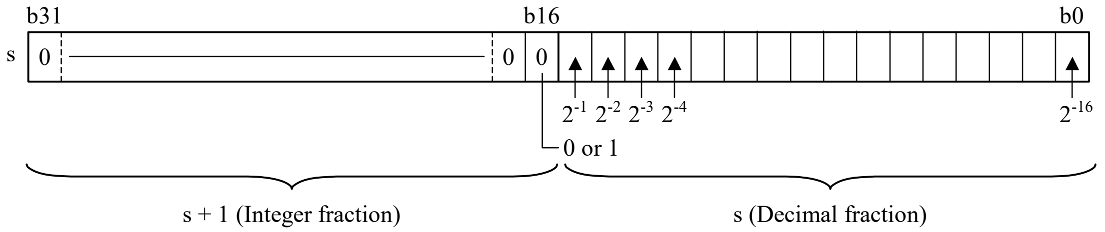
Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Calculation result								✓	✓	✓				
s	Argument											✓	✓	✓	

Remarks

Function

- TAN^{-1} function handling the unsigned binary value specified by s (Double word: upper word is an integer fraction, lower is a decimal fraction) as an argument is calculated.

  A decimal point data (s parameter lower word) = the real number value × 65,535.



- A form of substitution statement is written and the operation result is stored in d.

  Example) WR200 = ATAN (DR2)

  Calculates TAN^{-1} function handling DR2 as an argument and stores the result in WR200.
- If the operation is performed normally, DER=0.
- The calculation result is the binary value and an angle (Degree) from 0 to 180.

Parameter

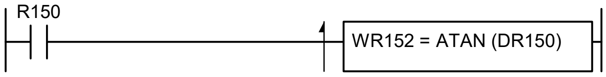
d: Specifies the internal output to store the calculation result.

s: Specifies an argument. A decimal point data (s parameter lower word) = the real number value × 65,535.

Cautionary notes

- Please specify the internal output used for argument and the internal output to store the calculation result within the range of the I/O No.
- Although a decimal point fraction is contained in an argument, it differs from a floating point format of IEEE754.

Program example



[Program description]

Calculates TAN^{-1} of DR150 (WR150: a decimal fraction, WR151: an integer fraction) at the rising edge of R150, and substitutes the result for WR152.

PRN ➔ PRJ

This command is equivalent to FUN 15 (s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 15 (s) into the program for MICRO-EHV is as follows.

FUN 15 (s) ➔ s+2 = ATAN (s), provided that s is Double word.

Example) FUN 15 (WR100) ➔ WR102 = ATAN (DR100)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	Arc tangent operation (Radian)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
d.FL = ATANR (s.FL)	—	5	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	15.54	19.2	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d.FL	Calculation result												✓	✓	
s.FL	Argument											✓	✓	✓	✓

Remarks

Function

- TAN^{-1} function handling floating point specified by s.FL as an argument is calculated.
- A form of substitution statement is written and the operation result is stored in d.FL with Radian units system.

  Example) DR200.FL = ATANR (DR2.FL)

  Calculates TAN^{-1} function handling DR2.FL as an argument and stores the result in DR200.FL.
- The operation result is an angle of Radian units system and represented by floating point.
- If the operation is performed normally, DER=0.

Parameter

d.FL: Specifies the internal output to store the calculation result.

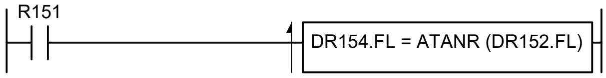
s.FL: Specifies an argument which has stored the floating point to find TAN^{-1}.

If the value which becomes s.FL > 1 is specified, the operation is not performed because of DER = 1.

Cautionary notes

- Please specify the internal output used for argument and the internal output to store the calculation result within the range of the I/O No.
- The internal outputs for argument and to store the calculation result need an extension ".FL".
- If an argument is integer, please convert it from integer into the floating point before executing the operation. (Otherwise, you cannot get the correct calculation result.)
- If the operation result is outside the range from −1e+37 to 1e+37, DER = 1.
- A format of floating point conforms to IEEE754.

Program example



[Program description]

Calculates TAN^{-1} of DR152.FL at the rising edge of R151, and substitutes the result for DR154.FL.

PRN ➔ PRJ

This command is equivalent to FUN 115 (s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 115 (s) into the program for MICRO-EHV is as follows.

FUN 115 (s) ➔ [s+2].FL = ATANR (s.FL), provided that s and s+2 are Double word.

Example) FUN 115 (WR100) ➔ DR102.FL = ATANR (DR100.FL)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	Exponent (Floating point)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
d.FL = EXP (s.FL)	–	5	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	67.2	83.2	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, TDN, SS, WDT, MS, TMR, CU, RCU, CT	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d.FL	Calculation result											✓	✓	
s.FL	Argument										✓	✓	✓	✓

Remarks

Function

- Handling the floating point specified by s.FL as an argument, Exponent is calculated.
- A form of substitution statement is written and the operation result is stored in d.FL.

  Example) DR10.FL = EXP (DR0.FL) Calculates exponent of DR0.FL and stores the result in DR10.FL.
- If the operation is performed normally, DER = 0.

Parameter

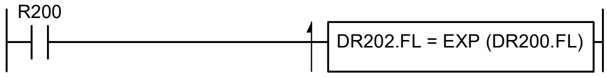
d.FL: Specifies the internal output to store the calculation result.

s.FL: Specifies an argument.

Cautionary notes

- Please specify the internal output used for argument and the internal output to store the calculation result within the range of the I/O No.
- The internal outputs for argument and to store the calculation result need an extension ".FL".
- If an argument is integer, please convert it form the integer into the floating point before executing the operation. (Otherwise, you cannot get the correct calculation result.)
- If the operation result is outside the range from −1e+37 to 1e+37, DER = 1.
- If s.FL < -7.0839639e+02, DER = 1 because the calculation is impossible.
- A format of floating point conforms to IEEE754.

Program example



[Program description]

Calculates the exponent of the floating point specified by DR200.FL at the rising edge of R200, and substitutes the result for DR202.FL.

PRN ➔ PRJ

This command is equivalent to FUN 117 (s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 117 (s) into the program for MICRO-EHV is as follows.

FUN 117 (s) ➔ [s+2].FL = EXP (s.FL), provided that s and s+2 are Double word.

Example) FUN 117 (WR100) ➔ DR102.FL = EXP (DR100.FL)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	Natural logarithm (Floating point)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
d.FL = LOG (s.FL)	−	5	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
−	12.8	16.06	−	−	−

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d.FL	Calculation result												✓	✓	
s.FL	Argument											✓	✓	✓	✓

Remarks

Function

- Handling the floating point specified by s.FL as an argument, the logarithm with base the natural logarithm (e) is calculated.
- A form of substitution statement is written and the operation results is stored in d.FL.
  Example) DR10.FL = LOG (DR0.FL) calculates the natural logarithm of DR0.FL and stores the result in DR10.FL.
- If the operation is performed normally, DER = 0.

Parameter

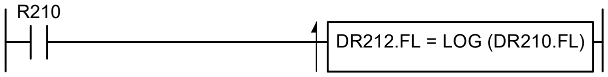
d.FL: Specifies the internal output to store the calculation result.

s.FL: Specifies an argument.

Cautionary notes

- Please specify the internal output used for argument and the internal output to store the calculation result within the range of the I/O No.
- The internal output for argument and to store the calculation result need an extension ".FL".
- If an argument is integer, please convert it from integer into the floating point before executing the operation. (Otherwise, you cannot get the correct calculation result.)
- If the operation result is outside the range from −1e+37 to 1e+37, DER = 1.
- If s.FL ≤ 0, DER = 1 because the calculation is impossible.
- A format of floating point conforms to IEEE754.

Program example



[Program description]

Calculates the logarithm of the floating point specified by DR210.FL at the rising edge of R210, and substitutes the result for DR212.FL.

PRN ➔ PRJ

This command is equivalent to FUN 118 (s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 118 (s) into the program for MICRO-EHV is as follows.

FUN 118 (s) ➔ [s+2].FL = LOG (s.FL), provided that s and s+2 are Double word.

Example) FUN 118 (WR100) ➔ DR102.FL = LOG (DR100.FL)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	Common logarithm (Floating point)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
d.FL = LOG10 (s.FL)	–	5	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	12.8	16.06	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d.FL	Calculation result											✓	✓	
s.FL	Argument										✓	✓	✓	✓

Remarks

Function

- Handling the floating point value specified by s.FL as an argument, the common logarithm with base 10 is calculated.
- A form of substitution statement is written and the operation result is stored in d.FL.
  Example) DR10.FL = LOG10 (DR0.FL) calculates the common logarithm of DR0.FL and stores the result in R10.FL.
- If the operation is performed normally, DER = 0.

Parameter

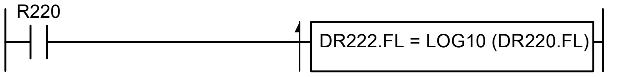
d.FL: Specifies the internal output to store the calculation result.

s.FL: Specifies an argument.

Cautionary notes

- Please specify the internal output used for argument and the internal output to store the calculation result within the range of the I/O No.
- The internal outputs for argument and to store the calculation result need an extension ".FL".
- If an argument is integer, please convert it from integer into the floating point before executing the operation. (Otherwise, you cannot get the correct calculation result.)
- If the operation result is outside the range from −1e+37 to 1e+37, DER = 1.
- If s.FL ≤ 0, DER = 1 because the calculation is impossible.
- A format of floating point conforms to IEEE754.

Program example



[Program description]

Calculates the logarithm of the floating point specified by DR220.FL at the rising edge of R220 and substitutes the result for DR222.FL.

MEMO

[3] Application commands

Name	Coding I/O address

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
d = ADR (s)	—	4	↕	●	●	●	●

Command processing time (μs)							
Average				Maximum			
Condition		Time		Condition		Time	
d	s	MVH (High function)	MVL (Standard)	d	s	MVH (High function)	MVL (Standard)
—		—	—	I/O	I/O	8.6	9.31
—		—	—	I/O	Array	15.82	16.63
—		—	—	Array	I/O	16.48	17.75
—		—	—	Array	Array	23	25.31

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O to store address value												✓	
s	I/O to be coded	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓	
( )	Index value								✓					

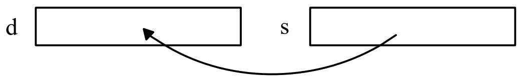
Remarks

Function

- I/O address specified by s is coded to store in d.

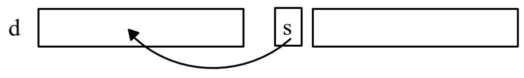
  (This is used combining with commands that require registering the I/O address.)

  Case of substitution statement d = s



Stores data of s in d.

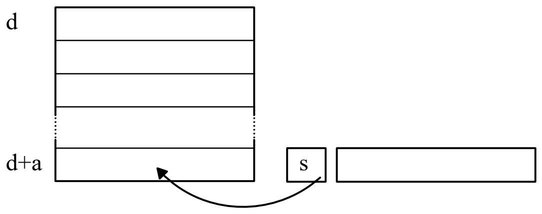
Case of specify address d = ADR (s)



Codes address of s to store in d.

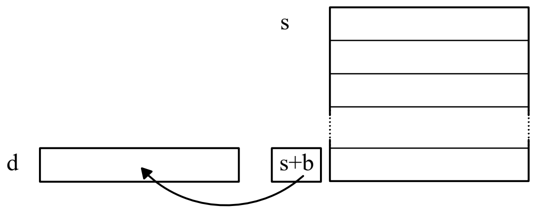
- An array variable is available for d and s.

Case of d (a) = ADR (s)



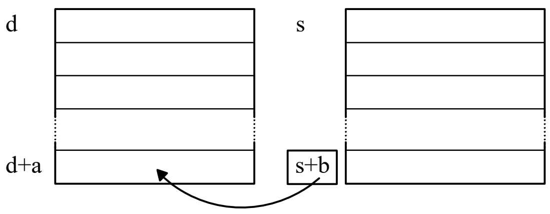
Codes address of s to store in d+a.

Case of d = ADR (s (b))



Codes address of s+b to store in d.

Case of d (a) = ADR (s (b))



Codes address of s+b to store in d+a.

Parameter

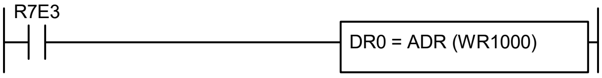
d: Specifies the internal output (Double word) to store the value which coded the I/O address.

s: Specifies the I/O to code.

Cautionary notes

- d can specify only the internal output of double word.
- The address before coding cannot be distinguished even if the coded address (the stored value in d) is monitored. Therefore, please check it at the spot where the I/O address coding command in a program was described.
- When using the array variable, DER=1 if it exceeds the maximum of the available I/O No.

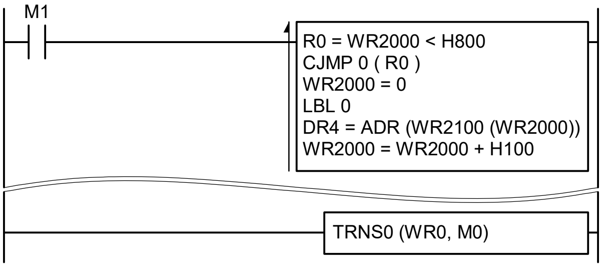
Program example



```
DR0 = ADR (WR1000)
```

[Program description]

WR1000 is coded at the first scan after RUN and the result is stored in DR0.



```
R0 = WR2000 < H800
CJMP 0 ( R0 )
WR2000 = 0
LBL 0
DR4 = ADR (WR2100 (WR2000))
WR2000 = WR2000 + H100
```

```
TRNS0 (WR0, M0)
```

[Program description]

When transmitting the different data by the TRNS0 command one after another, there is a method to change the top I/O in the transmitting data area specified by s parameter of TRNS0, preparing the transmitting data areas are at some spots.

WR2100	Data 1	Transmitting data at the N+1th times
WR2200	Data 2	Transmitting data at the N+2th times
WR2300	Data 3	Transmitting data at the N+3th times
		⋮
WR2800	Data 4	Transmitting data at the N+8th times

The program which is mentioned above moves the top I/O in the transmitting data area for 256 (H100) words when ever the TRNS0 is executed normally. The top I/O in the transmitting data area increases H100 at a time, and the top I/O will return to WR2100 if it becomes WR2800.

PRN ➔ PRJ

This command is equivalent to ADRIO(d, s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used ADRIO (d, s) into the program for EHV is as follows.

ADRIO (d, s) ➔ d = ADR (s), provided that s is Double word.

Example) ADRIO (WR100, WR0) ➔ DR100 = ADR (WR0)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

[Note at the time of program conversion]

The I/O address after code conversion can be stored in one word in MICRO-EH but stored in one double word (2-words) in MICRO-EHV.

Please pay attention that d parameter does not overlap with an area used currently for another purpose.

Reference

This command is used combining with other command.

Commands that use the coded I/O address are shown in following table.

No.	Command format	Command description
1	BSHR (d, n)	Byte right shift
2	BSHL (d, n)	Byte left shift
3	ASC (d, s, n)	conversion Binary to ASCII
4	HEX (d, s, n)	conversion ASCII to Binary
5	WTOB (d, s, n)	Conversion words to bytes
6	BTOW (d, s, n)	Conversion bytes to words
7	BITTOW (d, s, n)	Expand bit data to word data
8	WTOBIT (d, s, n)	Compress word data to bit data
9	SADD (d, s1, s2)	Character string unite
10	SCMP (d, s1, s2)	Character string comparison
11	INTPL (s)	Linear interpolation
12	RECSET (s, n)	Data storage (Initial setting)
13	PIDIT (s)	PID Operation (Initialization)
14	PIDOP (s)	PID Operation (Execution control)
15	PIDCL (s)	PID Operation (Calculation)
16	CCCL (s)	Generating check code
17	CCCMP (d, s)	Collating check code
18	IFR (s)	Process stepping
19	PGEN (s)	Generation of scan pulse
20	TRNS 0 (s, t)	CPU serial communication port Sending data
21	RECV 0 (s, t)	CPU serial communication port Receiving data
22	MBMST (s, t)	Modbus protocol query communication command
23	MBTCL (s, t)	Modbus-TCP client command
24	TRNS 4 (s, t)	Positioning expansion unit setting command
25	MPOSCTRL (s, t)	Positioning expansion unit control command

Name	[Bit operation] Bit set

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
BSET (d, n)	–	4	●	●	●	●	●

Command processing time (μs)

Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d : Word	14.68	15.63	–	–	–
d : Double word	16.76	17.43	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O to set bit							✓	✓	✓		✓	✓	
n	Bit position to set						✓	✓	✓	✓				✓

Remarks

Function

- The nth bit in the I/O (Word or Double word) specified by d is set to 1.
- Other bits remain unchanged.
- When d is Word,

  A bit position is specified with the content (0 to 15) of the lower 4 bits (b3 to b0) in n (WX, WY, WR, WM, TC).

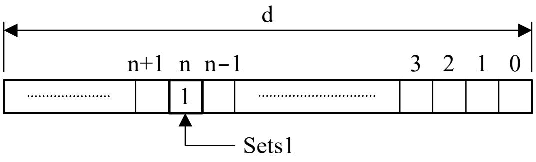
  (The upper bits are ignored and it is considered to be 0.)

  It can specify from 0 to 15 to n (a constant). (Decimal system)
- When d is Double word,

  A bit position is specified with the content (0 to 31) of the lower 5 bits (b4 to b0) in n (WX, WY, WR, WM, TC).

  (The upper bits are ignored and it is considered to be 0.)

  It can specify from 0 to 31 to n (a constant). (Decimal system)



Parameter

d: Specifies the I/O (word or double word) to set bit.

n: Specifies the bit position to set.

Program example

Refer to the explanation pages in ”BTS (d, n)”.

Name	[Bit operation] Bit reset

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
BRES (d, n)	—	4	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d : Word	15.16	15.67	—	—	—
d : Double word	17.16	17.99	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O to set bits							✓	✓	✓		✓	✓	
n	Bit position to set						✓	✓	✓	✓				✓

Remarks

Function

- The nth bit in the I/O (Word or Double word) specified by d is set to 0.
- Other bits remain unchanged.
- When d is Word,

  A bit position is specified with the content (0 to 15) of the lower 4 bits (b3 to b0) in n (WX, WY, WR, WM, TC).

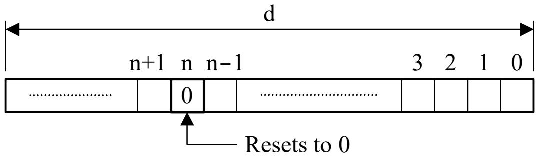
  (The upper bits are ignored and it is considered to be 0.)

  It can specify from 0 to 15 to n (a constant). (Decimal system)
- When d is Double word,

  A bit position is specified with the content (0 to 31) of the lower 5 bits (b4 to b0) in n (WX, WY, WR, WM, TC).

  (The upper bits are ignored and it is considered to be 0.)

  It can specify from 0 to 31 to n (a constant). (Decimal system)



Parameter

d: Specifies the I/O (Word or Double word) to set bit.

n: Specifies the bit position to reset.

Program example

Refer to the explanation pages in “BTS (d, n)”.

Name	[Bit operation] Bit Test

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
BTS (d, n)	–	4	●	●	●	●	↕

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d : Word	7.72	8.39	–	–	–
d : Double word	8.92	9.51	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O to test bits							✓	✓	✓		✓	✓	
n	Bit position to test						✓	✓	✓	✓				✓

Remarks

Function

- A content of the nth bit in the I/O(Word or Double word) specified by d is checked. If it is 1, C(R7F0) is set to 1 and if it is 0, C is reset to 0.
- Other bits remain unchanged.
- When d is Word,

  The bit position is specified with the content (0 to 15) of the lower 4 bits (b3 to b0) in n (WX, WY, WR, WM, TC).

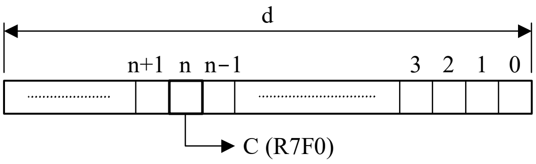
  (The upper bits are ignored and it is considered to be 0.)

  It can specify from 0 to 15 to n (a constant). (Decimal system)
- When d is Double word,

  The bit position is specified with the content (0 to 31) of the lower 5 bits b4 to b0) in n (WX, WY, WR, WM, TC).

  (The upper bits are ignored and it is considered to be 0.)

  It can specify from 0 to 31 to n (a constant). (Decimal system)

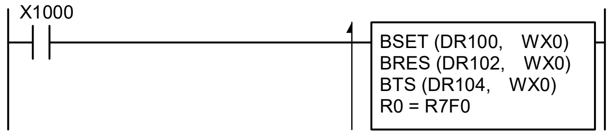


Parameter

d: Specifies the I/O (Word or Double word) to test bit.

n: Specifies the bit position to test.

Program example

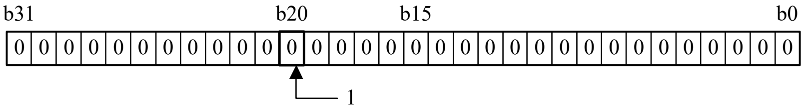


[Program description]

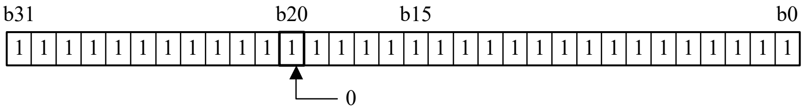
The bit operation in the processing box is performed at the rising edge of X1000.

If X1000 turns ON when WX0 is 20 (H0014), DR100 is 0, DR102 is HFFFFFFFF, and DR104 is H5555AAAA,

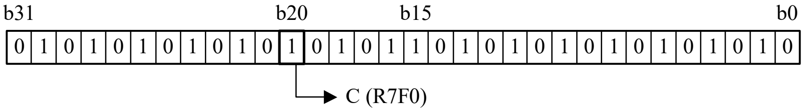
(1) The 20th bit in DR100 is set to 1 by BSET.



(2) The 20th bit in DR102 is reset to 0 by BRES.



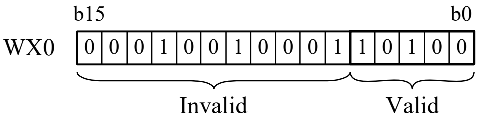
(3) The content of the 20th bit in DR104 is set to R7F0 by BTS.



[Reference]

The bit position is specified with the value of the lower 4 bits or 5 bits in the I/O used as n parameter.

If the value of WX0 is H1234 in the program example mentioned above, since the lower 5 bits are valid, the bit position will be the same result because of being 20(H0014).



Name	[Bit operation] Bit Count

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
BCU (d, s)	Word	4	●	●	●	●	●
	Double word	5					

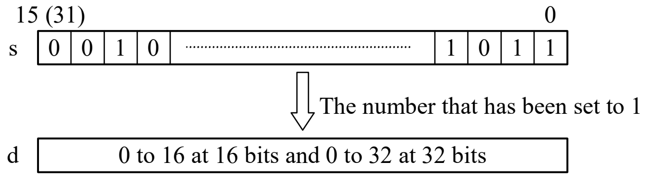
Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d : Word	10.08	10.31	—	—	—
d : Double word	11.06	12.59	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT, TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O to store number of 1							✓	✓	✓				
s	I/O to count bit of 1						✓	✓	✓	✓	✓	✓	✓	✓

Remarks

Function

The number of bits set to 1 with a content of ‘s’ (16 bits at Word, 32 bits at Double word) is stored in ‘d’.

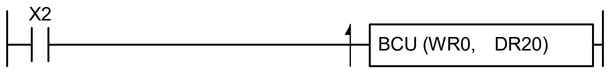


Parameter

d: Specifies the I/O to store the number of 1 contained in the I/O specified by s.

s: Specifies the I/O to count the number of 1 or a constant.

Program example



[Program description]

Counts the number of 1 in data of DR20 at the rising edge of X2 and sets the result in WR0.

If X2 turns ON when DR20 is H12345678, WR0 is set to 13.

Name	[Shift / Rotate] Shift right

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
SHR (d, n)	–	4	●	●	●	●	↕

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d : Word	4.58	5.63	–	–	–
d : Double word	5.9	5.58	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O to shift							✓	✓	✓		✓	✓	
n	Number of bits to shift						✓	✓	✓	✓				✓

Remarks

Function

- The content of d is shifted to the right (the lower direction) n bits.
- Data of SD(R7F2) is stored in n bits from the MSB.
- The content of the nth bit from the LSB is stored in C(R7F0).
- When d is Word,

  Volume to shift is specified with the content (0 to 15) of the lower 4 bits (b3 to b0) in n (WX, WY, WR, WM, TC).

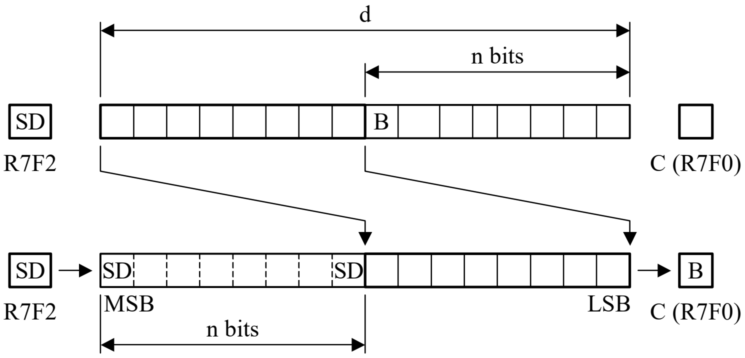
  (The upper bits are ignored and it is considered to be 0.)

  It can specify from 0 to 15 to n (a constant). (Decimal system)
- When d is Double word,

  Volume to shift is specified with the content (0 to 31) of the lower 5 bits (b4 to b0) in n (WX, WY, WR, WM, TC).

  (The upper bits are ignored and it is considered to be 0.)

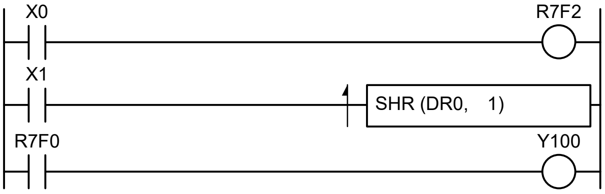
  It can specify from 0 to 31 to n (a constant). (Decimal system)



Cautionary notes

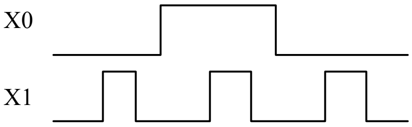
If n=0, it does not shift. C holds the preceding state.

Program example

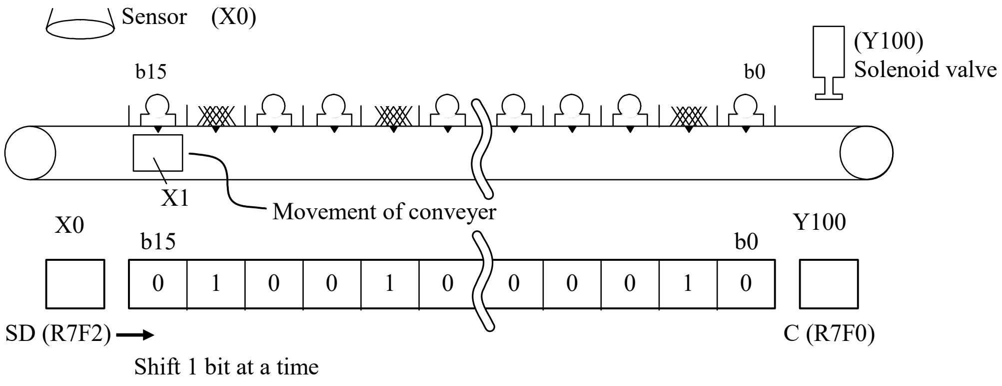


[Program description]

- There is a conveyer with 16 stands and it is moving to the right direction.
- Whenever a stand moves to the one right, one pulse input goes into X1.
- There is a sensor on the left end of the conveyer and X0 turns ON if inferior goods are put on the conveyer.
  The signal of X0 (sensor input) and X1 (conveyer move) is as follows.



- Data is also shifted 1 bit at a time with the movement to the right, and inferior goods are expelled out at the place (the right end of the conveyer) where data has come out to the carry because a solenoid valve (Y100) turns ON.



Name	[Shift / Rotate] Shift left

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
SHL (d, n)	–	4	●	●	●	●	↕

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d : Word	5.06	5.19	–	–	–
d : Double word	4.9	5.67	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O to shift							✓	✓	✓		✓	✓	
n	Number of bits to shift						✓	✓	✓	✓				✓

Remarks

Function

- The content of d is shifted to the left (the upper direction) n bits.
- Data of SD(R7F2) is stored in n bits from the LSB.
- The content of the nth bit from MSB is stored in C(R7F0).
- When d is Word,

  Volume to shift is specified with the content (0 to 15) of the lower 4 bits (b3 to b0) in n (WX, WY, WR, WM, TC)).

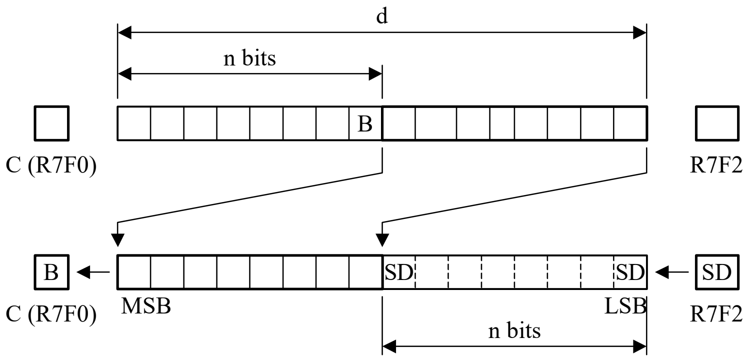
  (The upper bits are ignored and it is considered to be 0.)

  It can specify from 0 to 15 to n (a constant). (Decimal system)
- When d is Double word,

  Volume to shift is specified with the content (0 to 31) of the lower 5 bits (b4 to b0) in n (WX, WY, WR, WM, TC).

  (The upper bits are ignored and it is considered to be 0.)

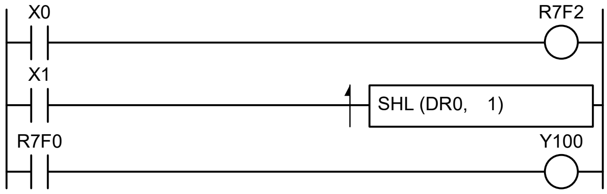
  It can specify from 0 to 31 to n (a constant). (Decimal system)



Cautionary notes

If n=0, it does not shift. C holds the preceding state.

Program example



[Program description]

- The value of R7F2 is determined by ON/OFF of X0.
- The content of DR0 is shifted to the left 1 bit at the rising of X1.

  In this case, the value of R7F2 is put into b0 and the value of b31 (b15 of WR1) is put into R7F0.

  Y100 turns ON/OFF from the value of b31 (b15 of WR1) of DR0 before the shift.

Name	[Shift / Rotate] Rotate right

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
ROR (d, n)	−	4	●	●	●	●	↕

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d : Word	5.66	6.51	−	−	−
d : Double word	6.54	6.99	−	−	−

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O to rotate							✓	✓	✓		✓	✓	
n	Number of bits to rotate						✓	✓	✓	✓				✓

Remarks

Function

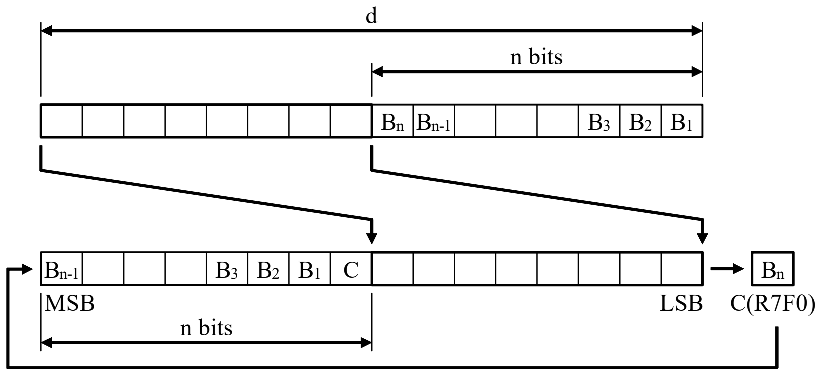
- The content of d is rotated to the right (the lower direction) n bits.
- The content of C(R7F0) is put into MSB, at a same time, the content of LSB is stored in C(R7F0). This processing is repeated n times.
- The content of the nth bit from LSB is stored in C(R7F0).
- When d is Word,

  Volume to shift is specified with the content (0 to 15) of the lower 4 bits (b3 to b0) in n (WX, WY, WR, WM, TC). (The upper bits are ignored and it is considered to be 0.)

  It can specify from 0 to 15 to n (a constant). (Decimal system)
- When d is Double word,

  Volume to shift is specified with the content (0 to 31) of the lower 5 bits (b4 to b0) in n (WX, WY, WR, WM, TC). (The upper bits are ignored and it is considered to be 0.)

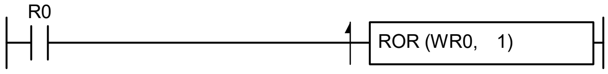
  It can specify from 0 to 31 to n (a constant). (Decimal system)



Cautionary notes

If n=0, it does not rotate. C holds the preceding state.

Program example



[Program description]

- WR0 is shifted to the right 1 bit at the rising of R0.
  In this case, the value of LSB b0 is put into R7F0 and the value of R7F0 just before shifting is put into MSB b15.

Name	[Shift / Rotate] Rotate left

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
ROL (d, n)	–	4	●	●	●	●	↕

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d : Word	5.70	6.47	–	–	–
d : Double word	6.02	6.99	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O to rotate							✓	✓	✓		✓	✓	
n	Number of bits to rotate						✓	✓	✓	✓				✓

Remarks

Function

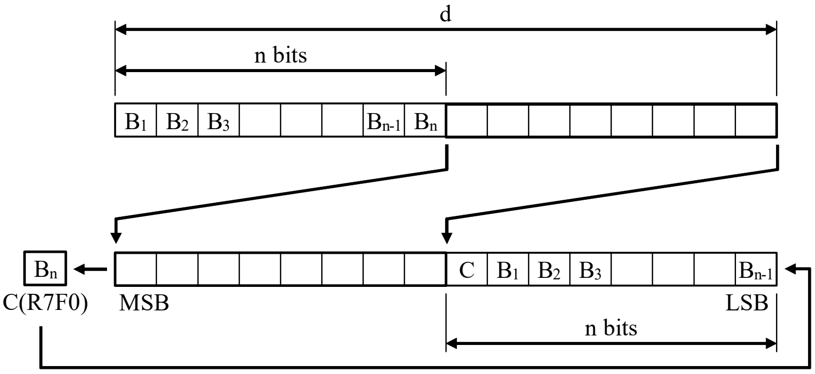
- The content of d is rotated to the left (the upper direction) n bits.
- The content of C(R7F0) is put into LSB, and at a same time, the content of MSB is stored into C(R7F0). This processing is repeated n times.
- The content of the nth bit from MSB can be stored in C(R7F0).
- When d is Word,

  Volume to shift is specified with the content (0 to 15) of the lower 4 bits (b3 to b0) in n ((X, WY, WR, WM, TC)). (The upper bits are ignored and it is considered to be 0.)

  It can specify from 0 to 15 to n (a constant). (Decimal system)
- When d is Double word,

  Volume to shift is specified with the content (0 to 31) of the lower 5 bits (b4 to b0) in n (WX, WY, WR, WM, TC). (The upper bits are ignored and it is considered to be 0.)

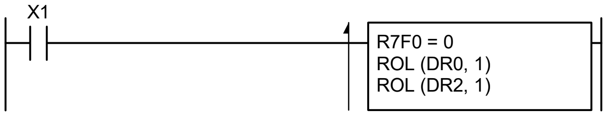
  It can specify from 0 to 31 to n (a constant). (Decimal system)



Cautionary notes

If n=0, it does not rotate. C holds the preceding state.

Program example

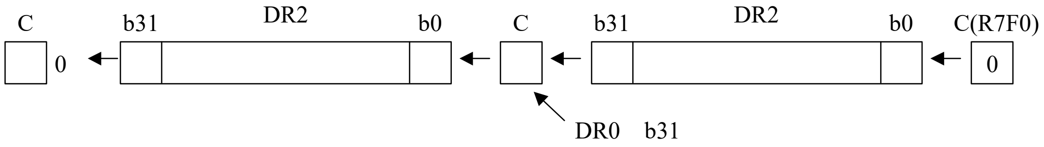


[Program description]

- 64-bit data is shifted 1 bit at a time at the rising of X1.

  0 is put into the opened space by shifting.

  The whole movement



Name	[Shift / Rotate] Logic shift right

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
LSR (d, n)	–	4	●	●	●	●	↕

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d : Word	5.06	5.63	–	–	–
d : Double word	5.46	6.11	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O to shift								✓	✓	✓		✓	✓	
n	Number of bits to shift							✓	✓	✓	✓				✓

Remarks

Function

- The content of d is shifted to the right (the lower direction) n bits.
- n bits from MSB store 0 respectively.
- The content of the nth bit from LSB is stored in C(R7F0).
- When d is Word,

  Volume to shift is specified with the content (0 to 15) of the lower 4 bits (b3 to b0) in n (WX, WY, WR, WM, TC).

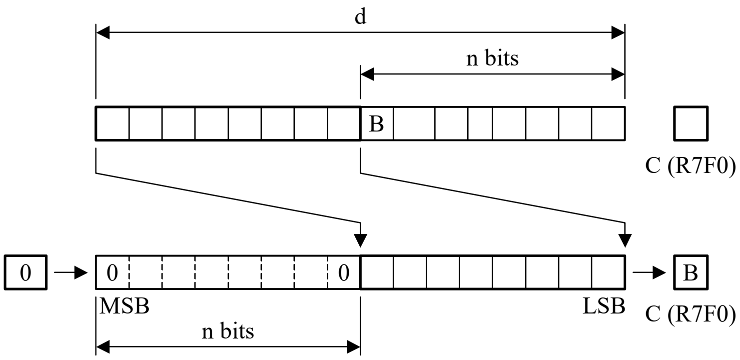
  (The upper bits are ignored and it is considered to be 0.)

  It can specify from 0 to 15 to n (a constant). (Decimal system)
- When d is Double word,

  Volume to shift is specified with the content (0 to 31) of the lower 5 bits (b4 to b0) in n (WX, WY, WR, WM, TC).

  (The upper bits are ignored and it is considered to be 0.)

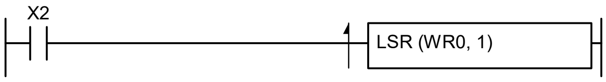
  It can specify from 0 to 31 to n (a constant). (Decimal system)



Cautionary notes

If n=0, it does not shift. C holds the preceding state.

Program example



[Program description]

- The value of WR is shifted to the right 1 bit at the rising of X2.
  In this case, 0 is put into b15 and the value of b0 before a shift is put into R7F0.

Name	[Shift / Rotate] Logic shift left

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
LSL (d, n)	–	4	●	●	●	●	↕

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d : Word	4.54	5.15	–	–	–
d : Double word	4.9	5.63	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O to shift							✓	✓	✓		✓	✓	
n	Number of bits to shift						✓	✓	✓	✓				✓

Remarks

Function

- The content of d is shifted to the left (the upper direction) n bits.
- n bits from LSB store 0 respectively.
- The content of the nth bit from MSB is stored in C(R7F0).
- When d is Word,

  Volume to shift is specified with the content (0 to 15) of the lower 4 bits (b3 to b0) in n (WX, WY, WR, WM, TC).

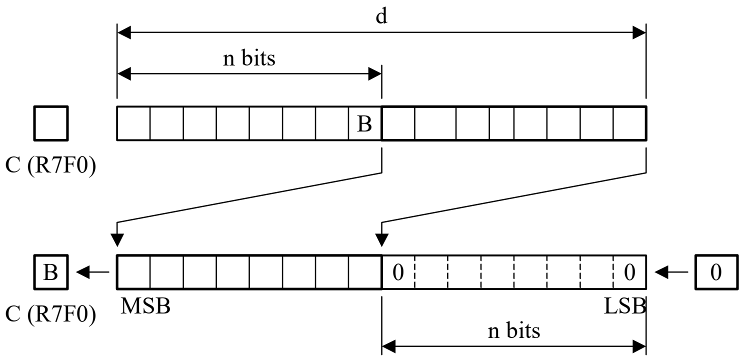
  (The upper bits are ignored and it is considered to be 0.)

  It can specify from 0 to 15 to n (a constant). (Decimal system)
- When d is Double word,

  Volume to shift is specified with the content (0 to 31) of the lower 5 bits (b4 to b0) in n (WX, WY, WR, WM, TC)..

  (The upper bits are ignored and it is considered to be 0.)

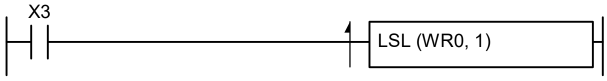
  It can specify from 0 to 31 to n (a constant). (Decimal system)



Cautionary notes

If n=0, it does not shift. C holds the preceding state.

Program example



[Program description]

- The value of WR0 is shifted to the left 1 bit.
  In this case, 0 is put into b0 and the value of b15 before a shift is put into R7F0.

Name	[Shift / Rotate] BCD shift right

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
BSR (d, n)	—	4	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d : Word	4.54	5.15	—	—	—
d : Double word	4.98	6.11	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O to shift							✓	✓	✓		✓	✓	
n	Number of digits to shift						✓	✓	✓	✓				✓

Remarks

Function

- The content of d is shifted to the right (the lower direction) n digits. (One digit has 4 bits.)
- n digits from the most significant digits store 0 respectively.
- n digits from the least significant digits are deleted.
- When d is Word,

  Volume to shift is specified with the content (0 to 3) of the lower 2 bits (b1 and b0) in n (WX, WY, WR, WM, TC).

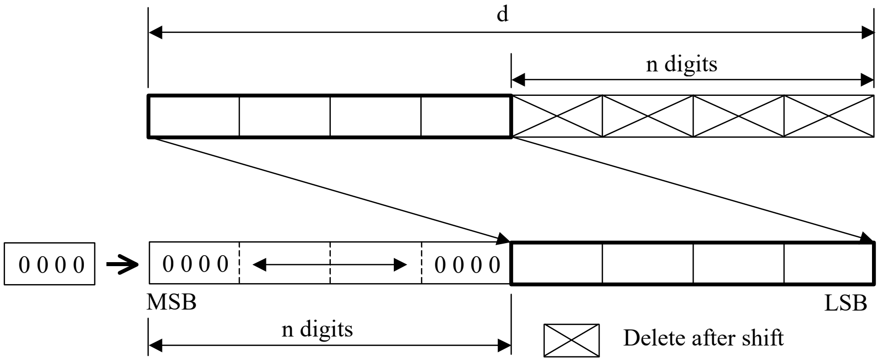
  (The upper bits are ignored and it is considered to be 0.)

  It can specify from 0 to 3 to n (a constant). (Decimal system)
- When d is Double word,

  Volume to shift is specified with the content (0 to 7) of the lower 3 bits (b2 to b0) in n (WX, WY, WR, WM, TC).

  (The upper bits are ignored and it is considered to be 0.)

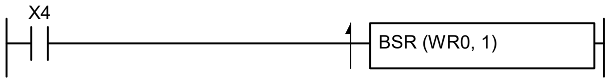
  It can specify from 0 to 7 to n (a constant). (Decimal system)



Cautionary notes

If n = 0, it does not shift.

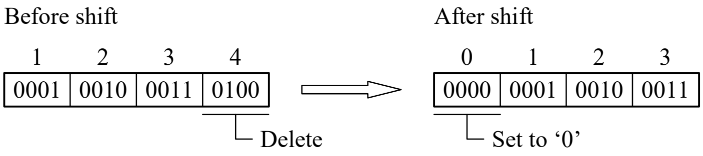
Program example



[Program description]

Considering the content of WR0 to be the BCD code, it is shifted t to the right one digit (4 bits) at the rising of X4.

In this case, data in lower 4 bits (b3 to b0) is deleted and the upper 4 bits (b15 to b12) are set to 0000.



Name	[Shift / Rotate] BCD Shift left

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
BSL (d, n)	—	4	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d : Word	4.98	5.13	—	—	—
d : Double word	5.46	5.59	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O to shift							✓	✓	✓		✓	✓	
n	Number of digits to shift						✓	✓	✓	✓				✓

Remarks

Function

- The content of d is shifted to the left (the upper direction) n digits. (One digit has 4 bits.)
- n digits from the least significant digit store 0 respectively.
- n digits form the most significant digit are deleted.
- When d is Word,

  Volume to shift is specified with the content (0 to 3) of lower 2 bits (b1 and b0) in n (WX, WY, WR, WM, TC) TC).

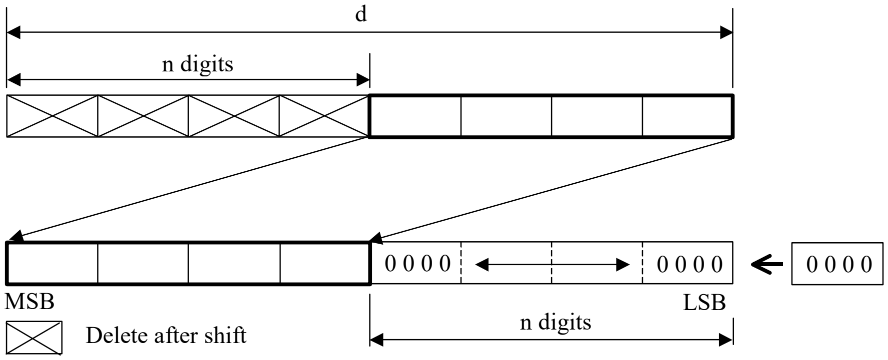
  (The upper bits are ignored and it is considered to be 0.)

  It can specify from 0 to 3 to n (a constant). (Decimal system)
- When d is Double word,

  Volume to shift is specified with the content (0 to 7) of the lower 3 bits (b2 to b0) in n (WX, WY, WR, WM, TC).

  (The upper bits are ignored and it is considered to be 0.)

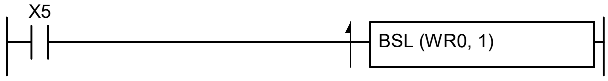
  It can specify from 0 to 7 to n (a constant). (Decimal system)



Cautionary notes

If n = 0, it does not shift.

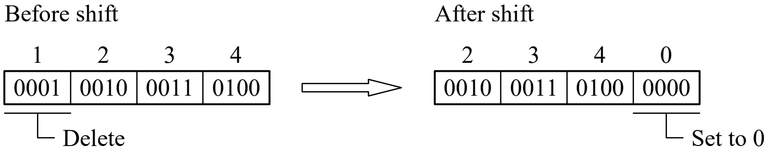
Program example



[Program description]

Considering the content of WR0 to be the BCD code, it is shifted to the left one digit (4 bits) at the rising of X5.

In this case, data in the upper 4 bits is deleted and the lower 4 bits are set to 0000.



Name	[Shift / Rotate] Batch shift right

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
WSHR (d, n)	—	4	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d : Bit	5.55+0.396*n	6.15+0.38*n	—	—	—
d : Word	4.51+1.773*n	4.77+2.28*n	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT, TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O to shift			✓					✓					
n	Number of bits to shift Number of words to shift						✓	✓	✓	✓				✓

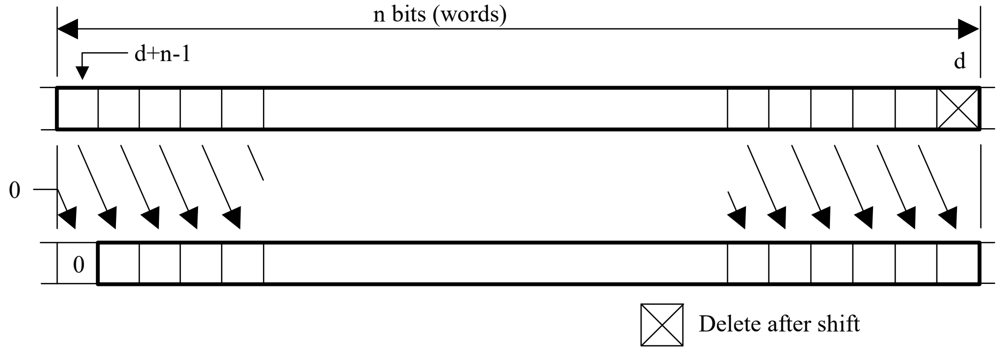
Remarks

Function

- n bits (word) from d is shifted to the right (the direction where I/O number is small) one bit (word).
- The content of the bit (word) specified by d is deleted.
- d+n-1, which is the bit in n bits (words) ahead from d, stores 0.
- When n is Word,

  Bit (word) volume to shift is specified with the content (0 to 255) of lower 8 bits (b7 to b0) in n (WX, WY, WR, WM, TC). (The upper bits are ignored and it is considered to be 0.)
- When n is a constant,

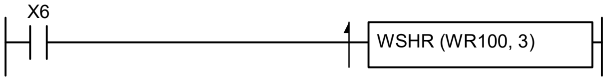
  Bit (word) volume to shift is specified. From 0 to 255 are valid.



Cautionary notes

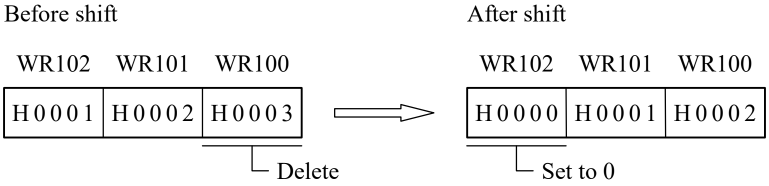
- If n = 0, it does not shift. DER is set to 0.
- Please use d+n−1 within the I/O range. If exceeded, DER = 1 and it is shifted from d to the maximum range.

Program example



[Program description]

Shifts the contents of WR100, WR101 and WR102 to the right one word at the rising of X6.



Name	[Shift / Rotate] Batch shift left

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
WSHL (d, n)	—	4	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d : Bit	5.9+0.372*n	7.11+0.35*n	—	—	—
d : Word	4.51+2.27*n	4.75+2.28*n	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O to shift			✓						✓					
n	Number of bits to shift Number of words to shift							✓	✓	✓	✓				✓

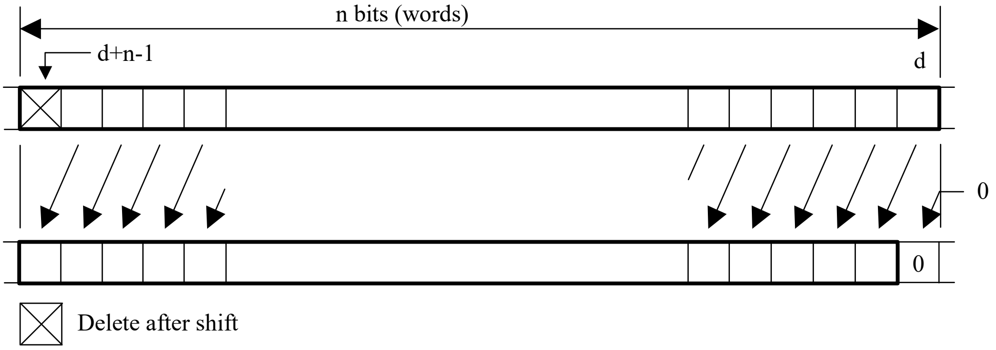
Remarks

Function

- n bits (words) from d are shifted to the left (the direction where I/O number is large) one bit (word).
- The bit (word) specified by d stores 0.
- The content of d+n-1, which is the bit in n bits ahead from d is deleted.
- When n is Word,

  Bit (word) volume to shift is specified the content (0 to 255) of lower 8 bits (b7 to b0) in n (WX, WY, WR, WM, TC). (The upper bits are ignored and it is considered to be 0.)
- When n is a constant,

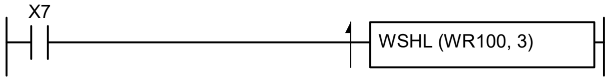
  Bit (word) volume to shift is specified. From 0 to 255 are valid.



Cautionary notes

- If n = 0, it does not shift and DER is set to 0.
- Please use d+n−1 within the I/O range. If exceeded, DER = 1 and it is shifted from d to the maximum range.

Program example



[Program description]

Shifts the contents of WR100, WR101 and WR102 to the left one word at the rising of X7.

Before shift

WR102	WR101	WR100
H0001	H0002	H0003

Delete (WR102)

After shift

WR102	WR101	WR100
H0002	H0003	H0000

Set to 0 (WR100)

Name	[Shift / Rotate] BCD batch shift right

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
WBSR (d, n)	—	4	↕	●	●	●	●

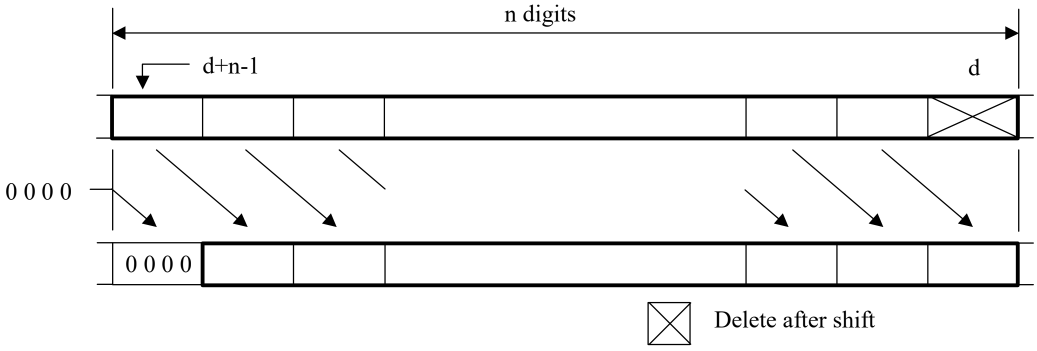
Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	5+2.39*n	4.79+2.4*n	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O to shift								✓					
n	Number of words for shifting						✓	✓	✓	✓				✓

Remarks
A constant is specified with decimal system.

Function

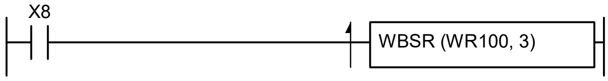
- n words from d which is considered to be the 4n-digit BCD data is shifted to the right (the direction where I/O number is small) one digit. (One digit has 4 bits.)
- The content of the least significant digit which is in the specified words ahead of d is deleted.
- The most significant digit (the upper 4 bits), which is in n words ahead from d, stores 0.
- When n is Word,
  The number of digits to shift is specified with the content (0 to 255) of lower 8 bits (b7 to b0) in n (WX, WY, WR, WM, TC). (The upper bits are ignored and it is considered to be 0.)
- When n is a constant,
  The number of digits to shift is specified. From 0 to 255 are valid.



Cautionary notes

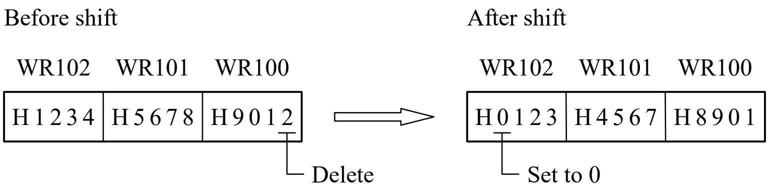
- If n = 0, it does not shift, and DER is set to 0.
- Please use d+n−1 within the I/O range. If exceeded, DER = 1 and it is shifted from d to the maximum range.

Program example



[Program description]

Considering the contents of WR100, WR101 and WR102 to be BCD code, it is shifted to the right 4 bits at the rising of X8.



Name	[Shift / Rotate] BCD batch shift left

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
WBSL (d, n)	–	4	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	4.5+2.89*n	4.77+2.9*n	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	The top I/O of character string for shifting								✓					
n	Number of bytes for shifting						✓	✓	✓	✓				✓

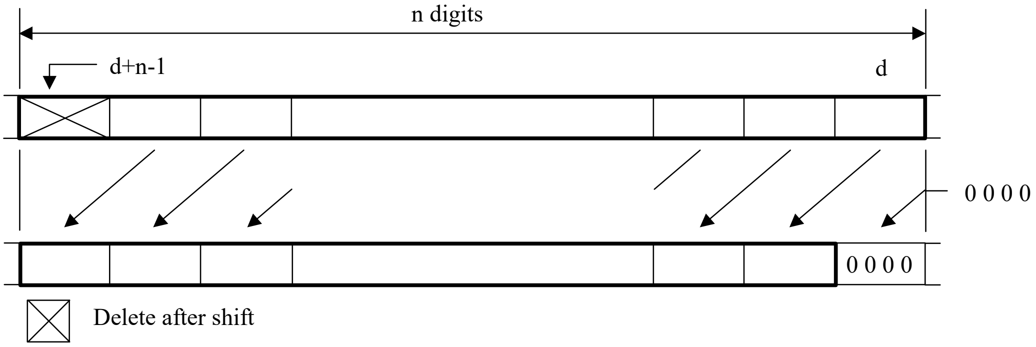
Remarks
A constant is specified with decimal system.

Function

- n words from d which is considered to be the 4n-digit BCD data is shifted to the left (the direction where I/O number is large) one digit. (One digit has 4 bits.)
- The content of the least significant digit in words specified by d stores 0.
- The most significant digit (the upper 4 bits) which is in n words ahead from d is deleted.
- When n is Word,

  The number of digits to shift is specified with the content (0 to 255) of the lower 8 bits (b7 to b0) in n (WX, WY, WR, WM, TC). (The upper bits are ignored and it is considered to be 0.)
- When n is a constant,

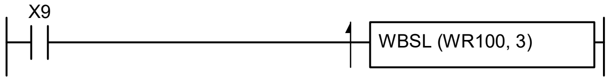
  The number of digits to shift is specified. From 0 to 255 are valid.



Cautionary notes

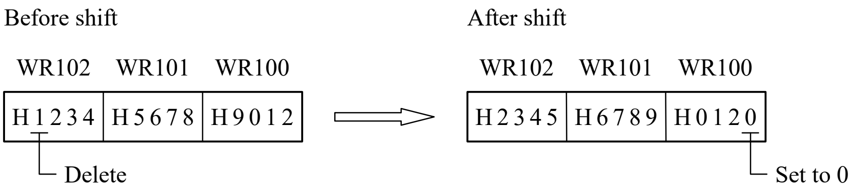
- If n = 0, it does not shift, and DER is set to 0.
- Please use d+n-1 within the I/O range. If exceeded, DER = 1 and it is shifted from d to the maximum range.

Program example



[Program description]

Considering the contents of WR100, WR101 and WR102 to be BCD code, it is shifted to the left 4 bits at the rising of X9.



Before shift

WR102	WR101	WR100
H1234	H5678	H9012

Delete

After shift

WR102	WR101	WR100
H2345	H6789	H0120

Set to 0

Name	[Shift / Rotate] Character data One byte right shift

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
BSHR (d, n)	–	4	↕	●	●	●	●

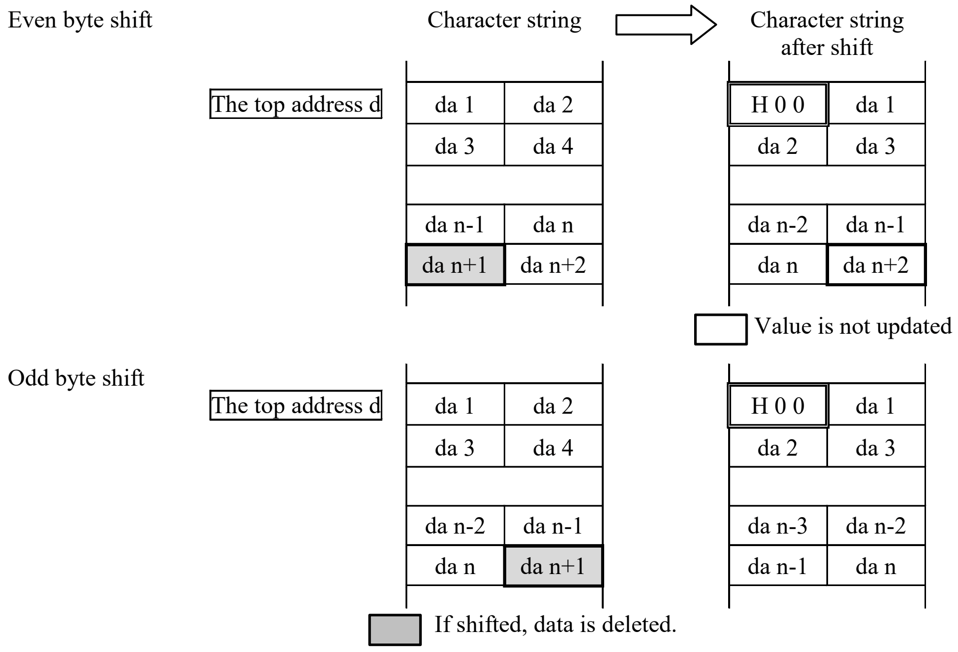
Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	4.88+1.42*n	5.2+1.43*n	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	The start I/O of character string for shifting									✓					
n	Number of bytes for shifting							✓	✓	✓	✓				✓

Remarks
A constant is specified with decimal system.

Function

- The n-byte character string data whose top is the address d is shifted to the right one byte.

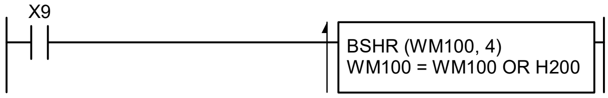


- An opened space after the shift stores H00.
- The next data to the specified number of bytes is deleted by the shift.

Cautionary notes

- Please specify the internal output used for the character string within the I/O number. If exceeded the maximum of the I/O number, DER = 1 and the operation is not performed.
- If n = 0, it does not shift and DER = 0.

Program example



[Program description]

Assume that 4 bytes of the sending data are stored at WM100 or after WM100.

- Shifts the 4-byte data whose top is WM100 to the right one byte at the rising of X9.
- Adds the communication control code STX (H02) to the top of data.

WM100	“ T ”	“ E ”
WM101	“ X ”	“ T ”
WM102		

⇨

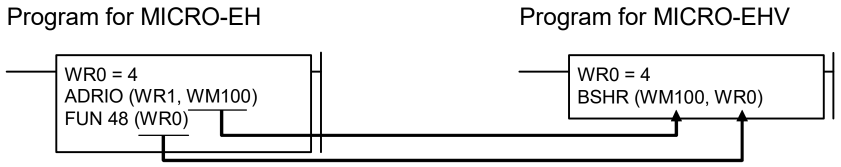
WM100	STX	“ T ”
WM101	“ E ”	“ X ”
WM102	“ T ”	

PRN ➔ PRJ

This command is equivalent to FUN 48 (s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 48 (s) into the program for MICRO-EHV is as follows.

FUN 48 (s) ➔ BSHR ([I/O specified by s+1], s)



* If converted by Convert Tool started from Control Editor of software Ver.6.00 or newer, it is converted as mentioned above. Because it cannot be converted by Control Editor of software Ver.5.01 or before, please use Control Editor of the newest software version.

Name	[Shift / Rotate] Character data One byte left shift

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
BSHL (d, n)	—	4	↕	●	●	●	●

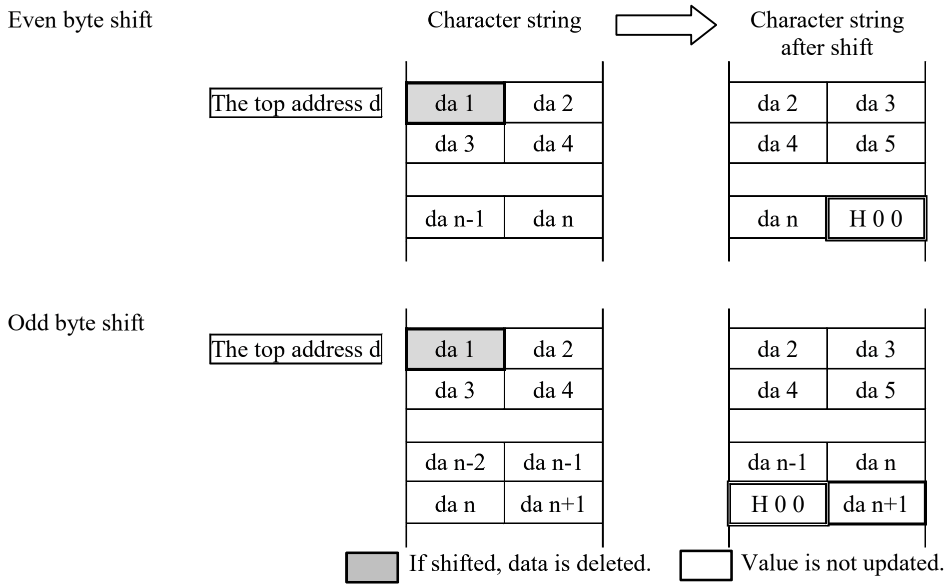
Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	4.8+1.978*n	4.74+1.98*n	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	The start I/O of character string for shifting									✓					
n	Number of bytes for shifting							✓	✓	✓	✓				✓

Remarks
A constant is specified with decimal system.

Function

- The n-byte character string data whose top is the address d is shifted to the left one byte.



Even byte shift

Character string		Character string after shift	
da 1	da 2	da 2	da 3
da 3	da 4	da 4	da 5
da n-1	da n	da n	H 0 0

Odd byte shift

Character string		Character string after shift	
da 1	da 2	da 2	da 3
da 3	da 4	da 4	da 5
da n-2	da n-1	da n-1	da n
da n	da n+1	H 0 0	da n+1

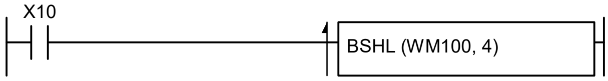
▨ If shifted, data is deleted. ☐ Value is not updated.

- An opened space after the shift stores H00.
- The top data in character string is deleted by the shift.

Cautionary notes

- Please the internal output used for the character string within the I/O number. If exceeded the maximum of I/O number, DER = 1 and the operation is not performed.
- If n = 0 or n = 1, it does not shift and DER = 0.

Program example



[Program description]

Assume that 5 bytes of the receiving data with a control code are stored at WM100 or after WM100.

- Shifts the 5-byte data whose top is WM100 to the left one byte at the rising of X10.
- Since the control code is deleted, only the receiving data remains.

WM100	STX	" T "
WM101	" E "	" X "
WM102	" T "	

⇨

WM100	" T "	" E "
WM101	" X "	" T "
WM102	H00	

PRN ➔ PRJ

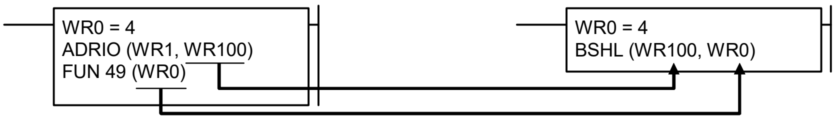
This command is equivalent to RUN 49 (s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 49 (s) into the program for MICRO-EHV is as follows.

FUN 49 (s) ➔ BSHL ([I/O specified by s+1], s)

Program for MICRO-EH

Program for MICRO-EHV


```
WR0 = 4
ADRIO (WR1, WR100)
FUN 49 (WR0)
```

```
WR0 = 4
BSHL (WR100, WR0)
```

* If converted by Convert Tool started from Control Editor of software Ver.6.00 or newer, it is converted as mentioned above. Because it cannot be converted by Control Editor of software Ver.5.01 or before, please use Control Editor of the newest software version.

Name	[Conversion of Character] Conversion Binary ➔ BCD

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
BCD (d, s)	Word	4	↕	●	●	●	●
	Double word	5					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
Word	10.26	11.7	—	—	—
Double word	12.18	13.3	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O after conversion (BCD)							✓	✓			✓	✓	
s	I/O before conversion (Binary)						✓	✓	✓	✓	✓	✓	✓	✓

Remarks

Function

- The content of s is converted from the binary into BCD and the result is stored in d.
- If the converted result of s exceeds the BCD digits of d, it is not performed because DER(R7F4) is set to 1.

  When s is Word, H0000 ≤ s ≤ H270F (0 to 9,999).

  When s is Double word, H00000000 ≤ s ≤ H5F5E0FF (0 to 99,999,999).
- The combinations of d and s are as follows.

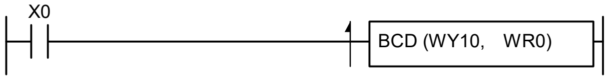
d	s
Word	Word
Double word	Double word

s	0	0	1	0	0	1	1	0	1	0	0	1	0	1	0	0	H2694 (9,876)
								⇩									
d	1	0	0	1	1	0	0	0	0	1	1	1	0	1	1	0	H9876
	15															0	
	9				8				7				6				

Cautionary notes

When the data error occurs, the content of d are held with being before the command execution.

Program example



[Program description]

Converts the content of WR0 from the binary into BCD when X0 is turned ON and outputs the result to WY10.

WR0 H1B4F (6991) ⟹ WY10 H6991

Name	[Conversion of Character] Conversion BCD ➔ Binary

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
BIN (d, s)	Word	4	↕	●	●	●	●
	Double word	5					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
Word	19.2	20.38	—	—	—
Double word	31	32.58	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT / TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O after conversion (Binary)							✓	✓			✓	✓	
s	I/O before conversion (BCD)						✓	✓	✓	✓	✓	✓	✓	✓

Remarks

Function

- The content of s is converted from BCD into the binary and the result is stored in d.
- If the content of s is not BCD data (if the symbols from A to F are in the data), it is not performed because DER(R7F4) is set to 1.
- The combinations of d and s are as follows.

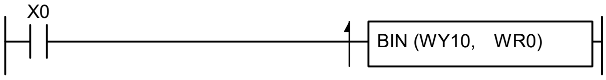
d	s
Word	Word
Double word	Double word

s	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	H1234
								⇩									
d	0	0	0	0	0	1	0	0	1	1	0	1	0	0	1	0	1,234 (H04D2)
	15							||								0	
	1,234 (Decimal)																

Cautionary notes

When the data error occurs, the contents of d are held with being before the command execution.

Program example



[Program description]

Converts the content of WR0 from BCD into the binary when X0 is turned ON and outputs the result to WY10.

WR0 H6991 ⟹ WY10 H1B4F (6991)

Name	[Conversion of Character] Conversion Binary code ➔ Gray code

Ladder format	Number of steps		Condition code				
GRY (d, s)	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
	Word	4	●	●	●	●	●
	Double word	5					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
Word	7.2	7.8	—	—	—
Double word	8.84	8.56	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT, TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O after conversion (Gray code)							✓	✓			✓	✓	
s	I/O before conversion (Binary code)						✓	✓	✓	✓	✓	✓	✓	✓

Remarks

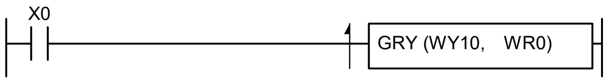
Function

- The content of s is converted from the binary code into the gray code and the result is stored in d.
- The combinations of d and s are as follows.

d	s
Word	Word
Double word	Double word

s	0	0	1	0	0	1	1	0	1	0	0	1	0	1	0	0	H2694
									⇩								
d	0	0	1	1	0	1	0	1	1	1	0	1	1	1	1	0	H35DE
	15															0	

Program example



[Program description]

Converts the content of WR0 from the binary code into the gray code at the rising edge of ON of X0 and outputs the result to WY10.

WR0 H2694 ⟹ WY10 H35DE

Name	[Conversion of Character] Conversion Gray code ➔ Binary code

Ladder format	Number of steps		Condition code				
GBIN (d, s)	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
	Word	4	●	●	●	●	●
	Double word	5					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
Word	7.64	8.3	–	–	–
Double word	8.4	9.54	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O after conversion (Binary code)							✓	✓			✓	✓	
s	I/O before conversion (Gray code)						✓	✓	✓	✓	✓	✓	✓	✓

Remarks

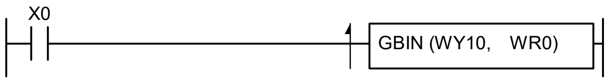
Function

- The content of s is converted from the gray code into the binary code and the result is stored in d.
- The combinations of d and s are as follows.

d	s
Word	Word
Double word	Double word

s	0	0	1	1	0	1	0	1	1	1	0	1	1	1	1	0	H35DE
								⇩									
d	0	0	1	0	0	1	1	0	1	0	0	1	0	1	0	0	H2694
	15															0	

Program example



[Program description]

Converts the content of WR0 from the gray code into the binary code at the rising edge of ON of X0 and outputs the result to WY10.

WR0 H35DE ⟹ WY10 H2694

Name	[Conversion of Character] Conversion 16-bit unsigned binary ➔ Decimal-ASCII

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
BINDA (d, s)	–	4	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
–	33.2	33.84	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O after conversion							✓	✓					
s	I/O before conversion						✓	✓	✓	✓				✓

Remarks
d parameters are occupied up to d+2.

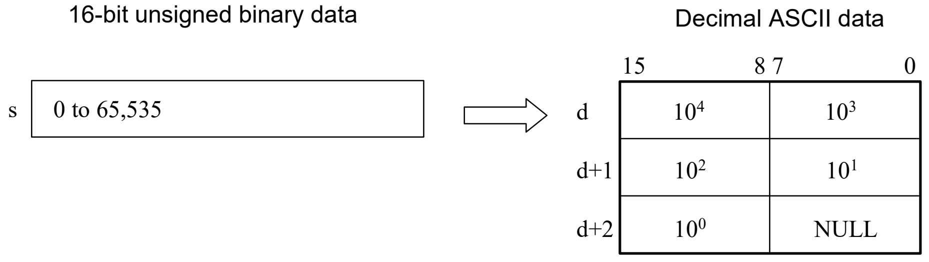
Function

- The 16-bit unsigned binary data is converted into the 5-digit decimal-ASCII code.
- The digits which the numerical value is not in is set to H20 (space) as a result of the zero suppression to the converted result. And the rest one byte after conversion to ASCII is set to NULL and it means a termination of the character string.

Parameter

d : Specifies the top I/O in the table to store the decimal-ASCII data after conversion.

s : Specifies the internal output in which the16-bit unsigned binary data to convert is stored or a constant.



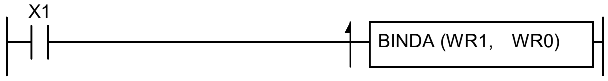
	15 8	7 0
d	10^4	10^3
d+1	10^2	10^1
d+2	10^0	NULL

10^n : ASCII code of 10^n positions

Cautionary notes

Please specify the internal output used for d and s parameters within the range of the I/O No.

Program example



[Program description]

Converts the value (16-bit unsigned binary data) of WR0 into the decimal-ASCII data at the rising of X1, and sets the result from WR1 to WR3.

If WR0 = 12,345, WR1 = H3132, WR2 = H3334, and WR3 = H3500

PRN ➔ PRJ

This command is equivalent to FUN 30 (s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 30 (s) into the program for MICRO-EHV is as follows.

FUN 30 (s) ➔ BINDA (s+1, s)

Example) FUN 30 (WR100) ➔ BINDA (WR101, WR100)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	[Conversion of Character] Conversion 32-bit singed binary ➔ Decimal-ASCII

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
SBINDA (d, s.S)	–	5	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
–	57.8	61.04	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O after conversion							✓	✓					
s.S	I/O before conversion										✓	✓	✓	✓

Remarks
d parameters are occupied up to d+5.

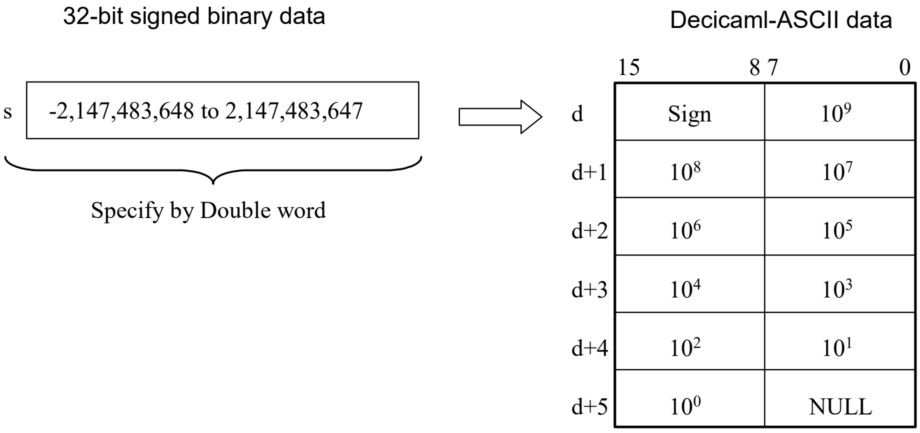
Function

- The 32-bit signed binary data is converted into the 10-digit decimal ASCII code.
- The digits which the numerical value is not in is set to H20 (space) as a result of the zero suppression to the converted result. And the rest one byte after conversion to ASCII is set to NULL and it means a termination of the character string.

Parameter

d : Specifies the top I/O in the table to store the decimal-ASCII data after conversion.

s.S : Specifies the internal output in which the 32-bit signed binary data to convert is stored or a constant.



32-bit signed binary data

s -2,147,483,648 to 2,147,483,647

Specify by Double word

Decicaml-ASCII data

	15 8	7 0
d	Sign	10^9
d+1	10^8	10^7
d+2	10^6	10^5
d+3	10^4	10^3
d+4	10^2	10^1
d+5	10^0	NULL

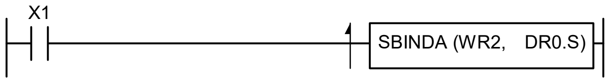
Sign Positive : H20 (space)
Negative : H2D (“-“)

10^n : ASCII code of 10^n position

Cautionary notes

Please specify the internal output used for d and s parameters within the range of the I/O No.

Program example



[Program description]

Converts the value (32-bit signed binary data) of DR0.S into the decimal-ASCII data at the rising of X1, and sets the result from WR2 to WR7.

If DR0.S = −1234,567, WR2 = H2D20, WR3 = H2020, WR4 = H3132, WR5 = H3334, WR6 = H3536, and WR7 = H3700.

PRN ➔ PRJ

This command is equivalent to FUN 31 (s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 31 (s) into the program for MICRO-EHV is as follows.

FUN 31 (s) ➔ SBINDA (s+2, s.S) , provided that s.S is double word.

Example) FUN 31 (WR100) ➔ SBINDA (WR102, DR100.S)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	[Conversion of Character] Conversion Unsigned binary ➔ Hexadecimal-ASCII

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
BINHA (d, s)	Word	4	↕	●	●	●	●
	Double word	5					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
Word	23.9	25.24	—	—	—
Double word	91.76	112.13	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O after conversion								✓	✓	✓				
s	I/O before conversion							✓	✓	✓	✓	✓	✓	✓	✓

Remarks
d parameters are occupied up to d+2 or d+4.

Function

- The unsigned binary data specified by s is converted into the hexadecimal-ASCII code.

  When s is Word, the 16-bit unsigned binary data is converted into the 4-digit hexadecimal-ASCII code.

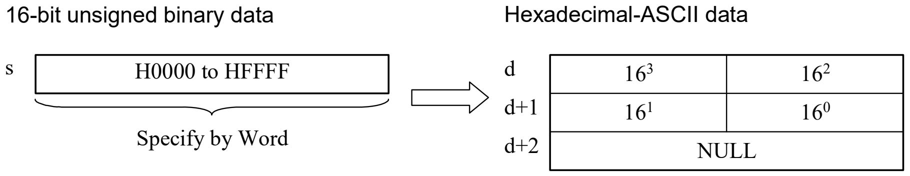
  When s is Double word, the 32-bit unsigned binary data is converted into the 8-digit hexadecimal-ASCII code.
- The zero suppression is not performed to the converted result. And NULL behind ASCII data means a termination of the character string.

Parameter

d : Specifies the top I/O in the table to store the hexadecimal-ASCII data after conversion.

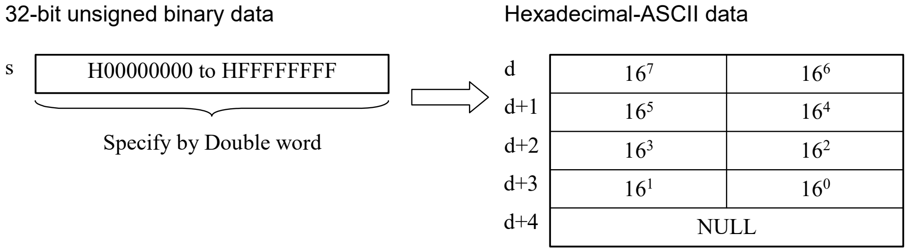
s : Case of Word: Specifies the word internal output in which the 16-bit unsigned binary data to convert is stored or a constant.

Case of Double word: Specifies the double word internal output in which the 32-bit unsigned binary data to convert is stored or a constant.



d	16^3	16^2
d+1	16^1	16^0
d+2	NULL	

16^n : ASCII code of 16^n position



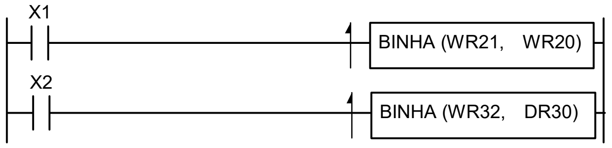
d	16^7	16^6
d+1	16^5	16^4
d+2	16^3	16^2
d+3	16^1	16^0
d+4	NULL	

16^n : ASCII code of 16^n position

Cautionary notes

- Please specify the internal output used for d and s parameters within the range of the I/O No.
- This command changes a size of d parameter by types of s parameter to be converted.
  (If s is word, it is used up to d+2. If s is double word, it is used up to d+4.)

Program example



[Program description]

- Converts the value (16-bit unsigned binary data) of WR20 into the hexadecimal-ASCII data at the rising of X1, and sets the result from WR21 to WR22. (WR23 is set to NULL.)
  If WR20 = H1234, WR21 = H3132, WR22 = H3334, and WR13 = H0000.
- Converts the value (32-bit unsigned binary data) of DR30 into the hexadecimal-ASCII data at the rising of S2, and sets the result from WR32 to WR35. (WR36 is set to NULL.)
  If DR30 = H001289AB, WR32 = H3030, WR33 = H3132, WR34 = H3839, WR35 = H4142, and WR36 = H0000.

PRN ➔ PRJ

This command is equivalent to FUN 32 (s) and FUN 33 (s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 32 (s) and FUN 33 (s) into the program for MICRO-EHV is as follows.

FUN 32 (s) ➔ BINHA (s+1, s)

Example) FUN 32 (WR100) ➔ BINHA (WR101, WR100)

FUN 33 (s) ➔ BINHA (s+2, s), provided that s is double word.

Example) FUN 33 (WR100) ➔ BINHA (WR102, DR100)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	[Conversion of Character] Conversion BCD ➔ Decimal-ASCII

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
BCDDA (d, s)	Word	4	↕	●	●	●	●
	Double word	5					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
Word	30.3	31.74	—	—	—
Double word	102.4	123.14	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O after conversion								✓	✓	✓				
s	I/O before conversion							✓	✓	✓	✓	✓	✓	✓	✓

Remarks
d parameters are occupied up to d+2 or d+4.

Function

- The BCD data is converted into the decimal-ASCII code.

  When s is Words, the 16-bit BCD data is converted into the 4-digit decimal-ASCII code.

  When s is Double word, the 32-bit BCD data is converted into the 8-digit decimal-ASCII code.
- The digits which the numerical value is not in is set to H20 (space) as a result of the zero suppression to the converted result. And NULL behind ASCII data means a termination of the character string.

Parameter

d : Specifies the top I/O in the table to store the decimal-ASCII data after conversion.

s : Case of Word: Specifies the internal output (word) in which the 16-bit BCD data to convert is stored or a constant.

Case of Double word: Specifies the internal output (double word) in which the 32-bit BCD data to convert is stored or a constant.

16-bit BCD data

s	10^3	10^2	10^1	10^0

Specify by Word

⇨

Decimal-ASCII data

d	10^3	10^2
d+1	10^1	10^0
d+2	NULL	

10^n : ASCII code of 10^n position

32-bit BCD data

s	10^7	10^6	10^5	10^4	10^3	10^2	10^1	10^0

Specify by Double word

⇨

Decimal-ASCII data

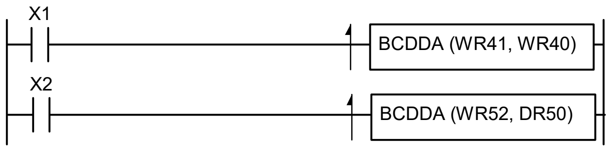
d	10^7	10^6
d+1	10^5	10^4
d+2	10^3	10^2
d+3	10^1	10^0
d+4	NULL	

10^n : ASCII code of 10^n position

Cautionary notes

- Please specify the internal output used for d and s parameters within the range of the I/O No.
- If the conversion data specified by s parameter is not the BCD data (if it is from A to F), the operation is not performed because of DER = 1.
- This command changes a size of d parameter by types of s parameter to be converted.
  (If s is Word, it is used up to d+2. If s is Double word, it is used up to d+4.)

Program example



[Program description]

- Converts the value (16-bit BCD data) of WR40 into the decimal-ASCII data at the rising of X1, and sets the result from WR41 to WR42. (WR43 is set to NULL.)
  If WR40 = H0123, WR41 = H2031, WR42 = H3233, and WR43 = H0000.
- Converts the value (32-bit BCD data) of DR50 into the decimal-ASCII data at the rising of X2, and sets the result from WR52 to WR55. (WR56 is set to NULL.)
  If DR50 = H00120567, WR52 = H2020, WR53 = H3132, WR54 = H3035, WR55 = H3637, and WR56 = H0000.

PRN ➔ PRJ

This command is equivalent to FUN 34 (s) and FUN 35 (s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 34 (s) and FUN 35 (s) into the program for MICRO-EHV is as follows.

FUN 34 (s) ➔ BCDDA (s+1, s)

Example) FUN 34 (WR100) ➔ BCDDA (WR101, WR100)

FUN 35 (s) ➔ BCDDA (s+2, s), provided that s is Double word.

Example) FUN 35 (WR100) ➔ BCDDA (WR102, DR100)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	[Conversion of Character] Conversion 5-digit unsigned decimal-ASCII ➔ 16-bit binary

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
DABIN (d, s)	–	4	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
–	73.88	78.28	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O after conversion							✓	✓	✓				
s	I/O before conversion						✓	✓	✓	✓				

Remarks
s parameters are occupied up to s+2.

Function

- The 5-digit unsigned decimal-ASCII data is converted into the 16-bit binary data.
  H00 and H20 (NULL and space) in upper digits are processed as H30 (“0”). (digit for zero suppression)

Parameter

d : Specifies the internal output to store the 16-bit binary data after conversion.

s : Specifies the top I/O in the table in which the unsigned decimal-ASCII data to convert is stored.

Unsigned decimal-ASCII data

	15 8	7 0
s	10^4	10^3
s+1	10^2	10^1
s+2	10^0	H00

⇨

16-bit binary data

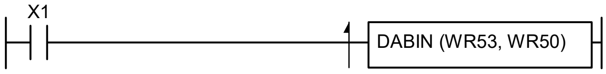
d	0 to 65,535

10^n : ASCII code of 10^n position

Cautionary notes

- Please specify the internal output used for d and s parameters within the range of the I/O No.
- If the 5-digit ASCII code specified by s parameter is not from H30 to H39 (0 to 9), the operation is not performed because of DER = 1.
- If the operation result becomes 65,535 or more, the operation is not performed because of DER = 1.

Program example



[Program description]

Converts the value (unsigned decimal-ASCII data) of WR50 to WR52 into the 16-bit binary data and sets the result in WR53.

If WR50 = H3132, WR51 = H3334 and WR52 = H3500, WR53 = 12,345.

PRN ➔ PRJ

This command is equivalent to FUN 36 (s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 36 (s) into the program for MICRO-EHV is as follows.

FUN 36 (s) ➔ DABIN (s+3, s)

Example) FUN 36 (WR100) ➔ DABIN (WR103, WR100)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	[Conversion of Character] Conversion 10-digit signed decimal-ASCII ➔ 32-bit binary

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
SDABIN (d.S, s)	–	4	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
–	17.68	19.74	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d.S	I/O after conversion											✓	✓	
s	I/O before conversion						✓	✓	✓	✓				

Remarks
s parameters are occupied up to s+5.

Function

- The 10-digit signed decimal-ASCII data is converted into the 32-bit binary data.
- Argument should be combined among H00, H20 (NULL and space), H30 to H39 and H2D (“-“).
- H00 and H20 (NULL and space) in upper digits are processed as H30 (“0”). (digit for zero suppression)

Parameter

d.S : Specifies the internal output to store the 32-bit binary data after conversion.

s : Specifies the top I/O in the table in which the signed decimal-ASCII data to convert is stored or a constant.

Signed decimal-ASCII data

	15 8	7 0
s	Sign	10^9
s+1	10^8	10^7
s+2	10^6	10^5
s+3	10^4	10^3
s+4	10^2	10^1
s+5	10^0	H00

⇨

32-bit signed binary data

d: −2,147,483,648 to 2,147,483,647

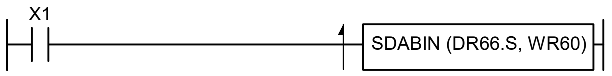
Sign Positive : H20 (space)
Negative : H2D(“−“)

10^n : ASCII code of 10^n position

Cautionary notes

- Please specify the internal output used for d and s parameters with the range of the I/O No.
- If the 10-digit ASCII code specified by s parameter is not from H30 to H39 (0 to 9) and if the sign is not H20 and H2D, the operation is not performed because of DER = 1. But this need not apply in H00 and H20 (NULL and space) in digits which performed the zero suppression.
- If the operation result is not from −2,147,483,648 to 2,147,483,647, the operation is not performed because of DER = 1.

Program example



[Program description]

Converts the value (signed decimal-ASCII data) of WR60 to WR65 into the 32-bit signed binary data at the rising of X1 and sets the result in DR66.S.

If WR60 = H2D32, WR61 = H3134, WR62 = H3734, WR63 = H3833, WR64 = H3634 and WR65 = H3800, DR66.S = −2,147,483,648.

PRN ➔ PRJ

This command is equivalent to FUN 37 (s) in the program (PRN file) of the MICRO-EH.

How to convert the program which has used FUN 37 (s) into the program for MICRO-EHV is as follows.

FUN 37 (s) ➔ SDABIN ([s+6].S, s), provided that s+6 is double word.

Example) FUN 37 (WR100) ➔ SDABIN (DR106.S, WR100)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	[Conversion of Character] Conversion Hexadecimal-ASCII ➔ 16-bit / 32-bit binary

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
HABIN (d, s)	―	4	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
Word	69.6	72.94	―	―	―
Double word	103.68	104.25	―	―	―

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O after conversion							✓	✓	✓		✓	✓	
s	I/O before conversion						✓	✓	✓	✓				

Remarks
s parameters are occupied up to s+1 or s+3.

Function

- The hexadecimal-ASCII data is converted into the 16-bit / 32-bit binary data.

  When d is Word, the 4-digit hexadecimal-ASCII data is converted into the16-bit binary data.

  When d is Double word, the 8-digit hexadecimal-ASCII data is converted into the 32-bit binary data.
- H00 and H20 (NULL and space) in upper digits are processed as H30 (“0”). (Digit for zero suppression)

Parameter

d : Specifies the internal output (word) to store the16-bit binary or the internal output (double word) to store the 32-bit binary data after conversion.

s : Specifies the top I/O in the table in which the hexadecimal-ASCII data to convert is stored.

Hexadecimal-ASCII data

	15 8	7 0
s	16^3	16^2
s+1	16^1	16^0

16^n : ASCII code of 16^n position

⇨

16-bit binary data

d	H0000 to HFFFF

Hexadecimal-ASCII data

	15 8	7 0
s	16^7	16^6
s+1	16^5	16^4
s+2	16^3	16^2
s+3	16^1	16^0

16^n : ASCII code of 16^n position

⇨

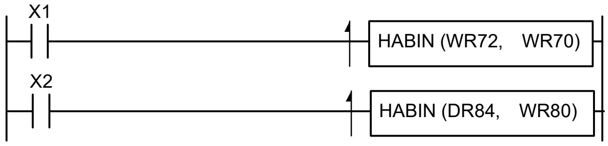
32-bit binary data

d	H00000000 to HFFFFFFFF

Cautionary notes

- Please specify the internal output used d and s parameters within the range of the I/O No.
- If ASCII code specified by s parameter is not from H30 to H39 (0 to 9) and from H41 to H46 (A to F), the operation is not performed because of DER = 1. But this need not apply in H00 and H20 (NULL and space) in digits which performed the zero suppression.
- This command changes a size of s parameter by types of d parameter to store the operation result.
  (If d is Word, ASCII code up to s+1 is converted. If d is Double word, ASCII code up to s+3 is converted.)

Program example



[Program description]

- Converts the value (hexadecimal-ASCII data) of WR70 and WR71 into the 16-bit binary data at the rising of X1 and sets the result in WR72.
  If WR70 = H3132 and WR71 = H4142, WR72 = H12AB.
- Converts the value (hexadecimal-ASCII data) of WR80 to WR83 into the 16-bit binary data at the rising of X2 and sets the result in DR84.
  If WR80 = H4645, WR81 = H4443, WR82 = H4241 and WR83 = H3938, DR80 is set to HFEDCBA98.

PRN ➔ PRJ

This command is equivalent to Fun 38 (s) and FUN 39 (s) in the program (PRN file) of MICRO-EH.

How to convert the program whish has used FUN 38 (s) and FUN 39 (s) into the program for MICRO-EHV is as follows.

FUN 38 (s) ➔ HABIN (s+2, s)

Example) FUN 38 (WR100) ➔ HABIN (WR102, WR100)

FUN 39 (s) ➔ HABIN (s+4, s), provided that s+4 is double word.

Example) FUN 39 (WR100) ➔ HABIN (DR104, WR100)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	[Conversion of Character] Conversion Decimal-ASCII ➔ 16-bit / 32-bit BCD

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
DABCD (d, s)	–	4	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High Function)	MVL (Standard)		MVH (High Function)	MVL (Standard)
Word	66.8	72.44	–	–	–
Double word	103.68	104.25	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O after conversion								✓	✓	✓		✓	✓	
s	I/O before conversion							✓	✓	✓	✓				

Remarks
s parameters are occupied up to s+1 or s+3.

Function

- The decimal-ASCII data is converted into the 16-bit / 32-bit BCD data.

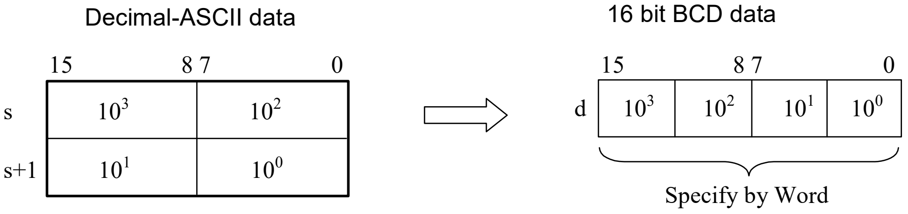
  When d is Word, the 4-digit decimal-ASCII data is converted into the 16-bit BCD data.

  When d is Double word, the 8-digit decimal-ASCII data is converted into the 32-bit BCD data.
- H00 and H20 (NULL and space) in upper digits are processed as H30 (“0”). (Digit for zero suppression)

Parameter

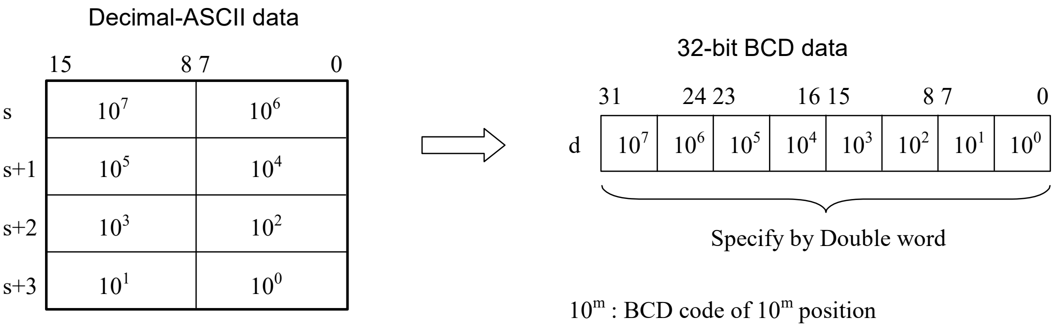
d : Specifies the internal output (word) to store the 16-bit BCD data or the internal output (double word) to store the 32-bit BCD data after conversion.

s : Specifies the top I/O in the table in which the 4-digit decimal-ASCII data to convert is stored.



10^n : ASCII code of 10^n position

10^m : BCD code of 10^m position



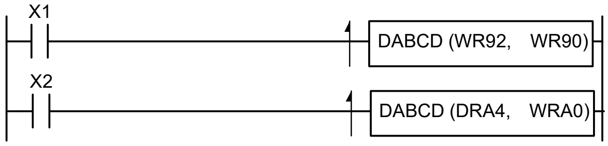
10^m : BCD code of 10^m position

10^n : ASCII code of 10^n position

Cautionary notes

- Please the internal output used for d and s parameters within the range of the I/O No.
- If ASCII code specified by s parameter is from H30 to H39 (0 to 9), the operation is not performed because of DER = 1.
- This command changes a size of s parameter by types of d parameter to store the operation result.
  (If d is Word, ASCII code up to s+1 is converted. If d is Double word, ASCII code up to s+3 is converted.)

Program example



[Program description]

- Converts the value (decimal-ASCII data) of WR90 and WR91 into the 16-bit BCD data at the rising of X1and sets the result in WR92.
  If WR90 = H2020 and WR91 = H3031, WR92 = H0001.
- Converts the value (decimal-ASCII data) of from WRA0 to WRA3 into the 16-bit BCD data at the rising of X2 and sets the result in DRA4.
  If WRA0 = H3938, WRA1 = H3736, WRA2 = H3534 and WRA3 = H3332, DRA4 is set to H98765432.

PRN ➔ PRJ

This command is equivalent to FUN 40 (s) and FUN 41 (s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 40 (s) and FUN 41 (s) into the program for MICRO-EHV is as follows.

FUN 40 (s) ➔ DABCD (s+2, s)

Example) FUN 40 (WR100) ➔ DABCD (WR102, WR100)

FUN 41 (s) ➔ DABCD (s+4, s), provided that s+4 is Double word.

Example) FUN 41 (WR100) ➔ DABCD (DR104, WR100)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	[Conversion of Character] Conversion Binary character string ➔ Hexadecimal-ASCII character string

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
ASC (d, s, n)	—	5	↕	●	●	●	●

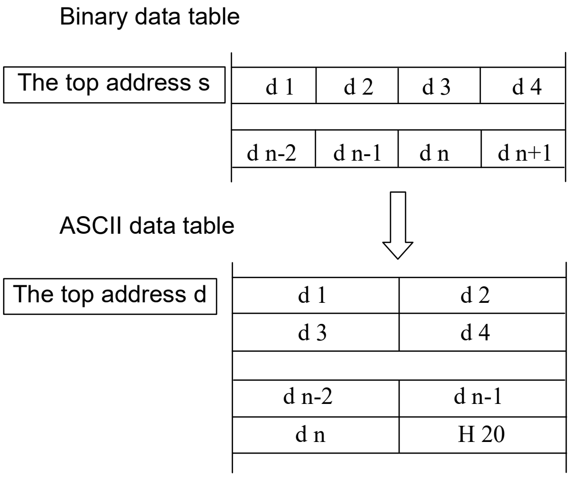
Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	6.76+6.2*n	6.76+6.74*n	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	The top I/O in ASCII data table									✓	✓				
s	The top I/O in binary data table							✓	✓	✓	✓				
n	Number of conversions							✓	✓	✓	✓				✓

Remarks

Function

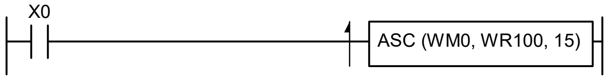
The number of characters specified by n from the top in the binary data table specified by s is converted into the hexadecimal-ASCII code and the converted data is stored in sequence from the internal output specified by d.



Cautionary notes

- Please specify the internal output used for binary data table and ASCII data table within the range of the I/O No. And if areas of binary data table and ASCII data table are overlapping, the operation is not performed because of DER = 1.
- The converted ASCII data are stored in sequence from the top in the word-unit. If the number of characters to convert is odd numbers, H20 (space) is stored in the end of the table.

Program example



[Program description]

Converts 15 characters from WR100 of the hexadecimal-binary data into the hexadecimal-ASCII data at the rising of X0 and stores the result in sequence from WM0.

WR100	H1 2	3 4
WR101	H5 6	7 8
WR102	H9 A	B C
WR103	HD E	F 0

⇨

WM0	H3 1	3 2
WM1	H3 3	3 4
WM2	H3 5	3 6
WM3	H3 7	3 8
WM4	H3 9	4 1
WM5	H4 2	4 3
WM6	H4 4	4 5
WM7	H4 6	2 0

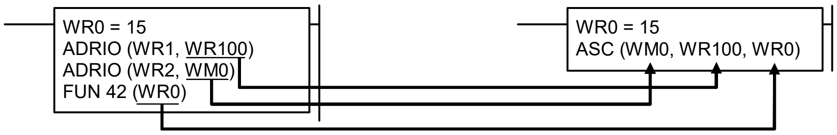
PRN ➔ PRJ

This command is equivalent to FUN 42 (s) in the program (PRN file) of EH-CPU.

How to convert the program which has used FUN 42 (s) into the program for MICRO-EHV is as follows.

FUN 42 (s) ➔ ASC ([I/O specified by s+2], [I/O specified by s+1], s)

Program for MICRO-EH Program for MICRO-EHV



```
WR0 = 15
ADRIO (WR1, WR100)
ADRIO (WR2, WM0)
FUN 42 (WR0)
```

```
WR0 = 15
ASC (WM0, WR100, WR0)
```

* If converted by Convert Tool started from Control Editor of software Ver.6.00 or newer, it is converted as mentioned above. Because it cannot be converted by Control Editor of software Ver.5.01 or before, please use Control Editor of the newest software version.

Name	[Conversion of character] Conversion Hexadecimal-ASCII character string ➔ Binary character string

Ladder format	Number of steps		Condition code				
HEX (d, s, n)	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
	—	5	↕	●	●	●	●

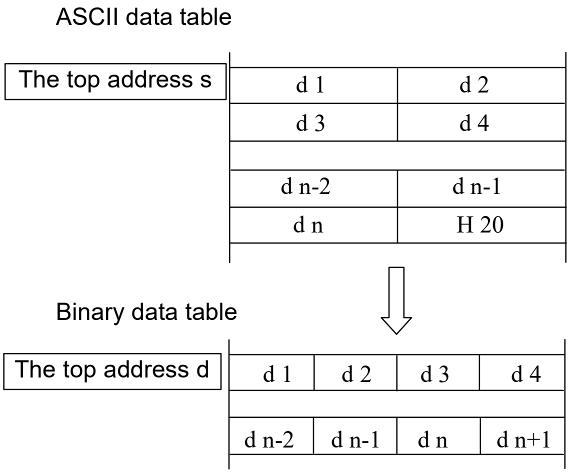
Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	7.07+6.46*n	8.13+6.51*n	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	The top I/O of binary data table								✓	✓				
s	The top I/O of ASCII data table						✓	✓	✓	✓				
n	Number of conversions						✓	✓	✓	✓				✓

Remarks

Function

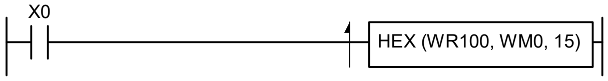
The number of characters specified by n from the top in the hexadecimal-ASCII code table specified by s is converted into the binary data and the converted data is stored in sequence from the internal output specified by d.



Cautionary notes

- Please specify the internal output used for ASCII data table and binary data table within the range of the I/O No. And if areas of ASCII data table and binary data table are overlapping, the operation is not performed because of DER = 1.
- The converted binary data is stored in sequence from the top in word units. If the number of characters to convert is not a multiple of 4, a data part less than 1 word stores 0.

Program example



[Program description]

Converts 15 characters from WM0 of the hexadecimal-ASCII data are converted into the hexadecimal-binary data at the rising of X0 and stores the result in sequence from WR100.

WM0	H3 1	3 2
WM1	H3 3	3 4
WM2	H3 5	3 6
WM3	H3 7	3 8
WM4	H3 9	4 1
WM5	H4 2	4 3
WM6	H4 4	4 5
WM7	H4 6	2 0

⇨

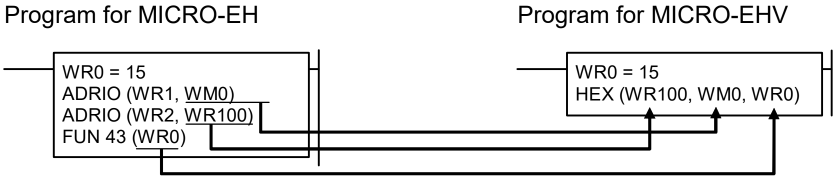
WR100	H1 2	3 4
WR101	H5 6	7 8
WR102	H9 A	B C
WR103	HD E	F 0

PRN ➔ PRJ

This command is equivalent to FUN 43 (s) in the program (PRN file) of EH-CPU.

How to convert the program which has used FUN 43 (s) into the program for MICRO-EHV is as follows.

FUN 43 (s) ➔ HEX ([I/O specified by s+2], [I/O specified by s+1], s)



* If converted by Convert Tool started from Control Editor of software Ver.6.00 or newer, it is converted as mentioned above. Because it cannot be converted by Control Editor of software Ver.5.01 or before, please use Control Editor of the newest software version.

Name	[Data operation] Conversion Word units ➔ Byte units

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
WTOB (d, s, n)	–	5	↕	●	●	●	●

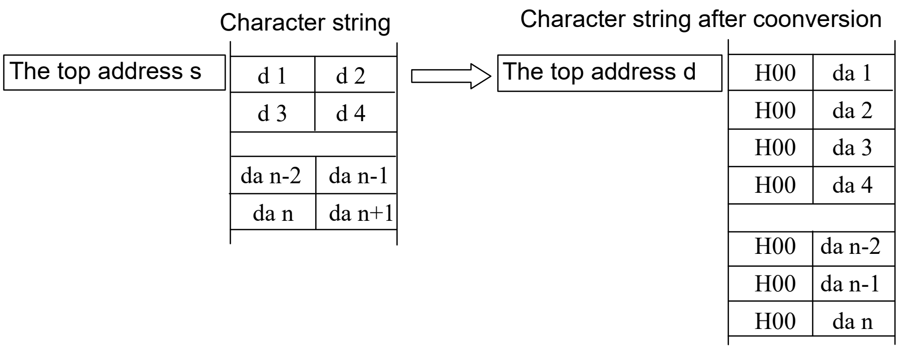
Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	5.89+6.27*n	6.82+6.58*n	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	The top I/O of character string after conversion								✓					
s	The top I/O of character string before conversion								✓					
n	Number of conversion bytes						✓	✓	✓	✓				✓

Remarks

Function

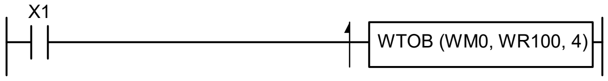
The n-byte data is picked out from the character data of which the top is the address s, and stores the data picked out in sequence from I/O specified by d as 1 byte per 1 word.



Cautionary notes

- Please specify the internal output used for the character string and the character string after conversion within the range of the I/O No.
- If areas of the character string and the character string after conversion are overlapping, the operation is not performed because of DER = 1.
- If n = 0, it is not converted and DER = 0.

Program example



[Program description]

Picks out 4 bytes from WR100 in order of the upper byte then the lower byte at the rising of X1, and sets the result in sequence from WM0.

WR100	H1 2	3 4
WR101	H5 6	7 8

⇨

WM0	H0 0	1 2
WM1	H0 0	3 4
WM2	H0 0	5 6
WM3	H0 0	7 8

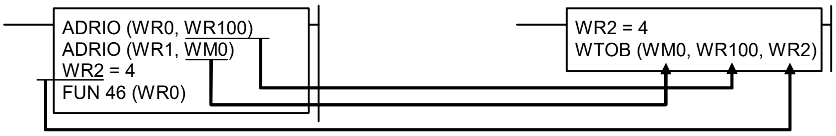
PRN ➔ PRJ

This command is equivalent to FUN 46 (s) in the program (PRN file) of EH-CPU.

How to convert the program which has used FUN 46 (s) into the program for MICRO-EHV is as follows.

FUN 46 (s) ➔ WTOB ([I/O specified by s+1], [I/O specified by s], s+2)

Program for MICRO-EH Program for MICRO-EHV



* If converted by Convert Tool started from Control Editor of software Ver.6.00 or newer, it is converted as mentioned above. Because it cannot be converted by Control Editor of software Ver.5.01 or before, please use Control Editor of the newest software version.

Name	[Data operation] Conversion Byte units ➔ Word units

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
BTOW (d, s, n)	–	5	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	6.4+6.19*n	6.8+7.25*n	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	The top I/O of character string after conversion								✓					
s	The top I/O of character string before conversion								✓					
n	Number of conversion bytes						✓	✓	✓	✓				✓

Remarks

Function

The lower byte (n bytes) are picked out from the character string data of which the top is the address s, and the bytes picked out are stored in sequence from I/O specified by d as 2 bytes per 1 word.

Character string

The top address s

H00	da 1
H00	da 2
H00	da 3
H00	da 4
H00	da n-2
H00	da n-1
H00	da n

⇨

Character string after conversion

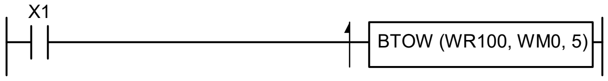
The top address d

d 1	d 2
d 3	d 4
da n-2	da n-1
da n	da n+1

Cautionary notes

- Please specify the internal output used for the character string and the character string after conversion within the range of the I/O No.
- If area of the character string and the character string after conversion are overlapping, the operation is not performed because of DER = 1.
- If n = 0, it is not converted and DER = 0.
- If the number of conversion bytes is odd number, the lower 8 bits at the end of the output destination are set to H00.

Program example



[Program description]

Picks out the lower bytes of 5 words from WM0 at the rising of X1, and sets the bytes picked out in order of the upper byte, a next lower byte, from WR100.

WM0	Any	H1 2
WM1	Any	H3 4
WM2	Any	H5 6
WM3	Any	H7 8
WM4	Any	H9 A

⇨

WR100	H1 2	3 4
WR101	H5 6	7 8
WR102	H9 A	0 0

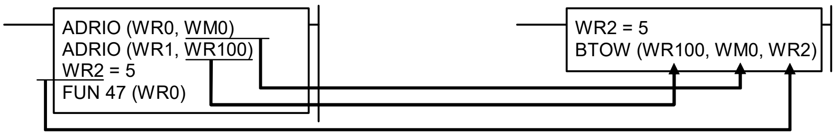
PRN ➔ PRJ

This command is equivalent to FUN 47 (s) in the program (PRN file) of EH-CPU.

How to convert the program which has used FUN 47 (s) into the program for MICRO-EHV is as follows.

FUN 47 (s) ➔ BTOW ([I/O specified by s+1], [I/O specified by s], s+2)

Program for MICRO-EH Program for MICRO-EHV



* If converted by Convert Tool started from Control Editor of software Ver.6.00 or newer, it is converted as mentioned above. Because it cannot be converted by Control Editor of software Ver.5.01 or before, please use Control Editor of the newest software version.

Name	[Data operation] Invert

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
NOT (d, s)	Bit	4	●	●	●	●	●
	Word	4					
	Double word	5					

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
Bit	17.10	20.18	—	—	—
Word	6.74	7.30	—	—	—
Double word	8.42	8.54	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)		WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O after reverse		✓	✓					✓	✓			✓	✓	
s	I/O to reverse	✓	✓	✓				✓	✓	✓	✓	✓	✓	✓	

Remarks

Function

- The contents of s are reversed and the result is stored in d.
- The combinations of d and s are as follows.

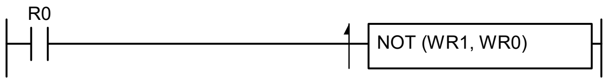
d	s
Bit	Bit
Word	Word
Double word	Double word

1	1	1	1	0	0	0	0	1	1	0	0	1	0	1	0
⇩	⇩	⇩	⇩	⇩	⇩	⇩	⇩	⇩	⇩	⇩	⇩	⇩	⇩	⇩	⇩
0	0	0	0	1	1	1	1	0	0	1	1	0	1	0	1

Cautionary notes

Please set a startup condition of this command to the edge trigger.

Program example



[Program description]

Reverses the contents of WR0 at the rising of R0 and stores the result in WR1.

Example) If WR0 is H1234, WR1 = HEDCB after the command execution.

PRN ➔ PRJ

This command is equivalent to NOT (d) in the program (PRN file) of EH-CPU.

How to convert the program which has used NOT (d) into the program for MICRO-EHV is as follows.

NOT (d) ➔ NOT (d, d) All d are the same I/O.

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	[Data operation] Unite

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
UNIT (d, s, n)	–	5	↕	●	●	●	●

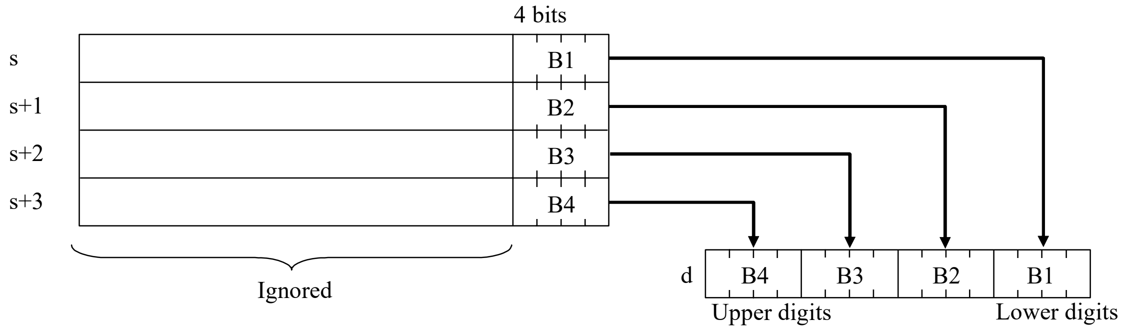
Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	13.14	14.54	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O for writing in the united result							✓	✓					
s	The top I/O to unite						✓	✓	✓					
n	Number of words to unite													✓

Remarks
n is from 0 to 4.

Function

- The value of the lower 4 bits in n (1 to 4) words from s is set to each4 bits from the lower in d.
- If n is from 1 to 3, the bit not to be set to d is set to 0.
- Data from s to s+n-1 does not change the value even if this command is executed.

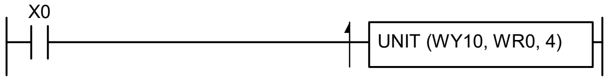


n=0 B4 to B1 are 0.
n=1 B4 to B2 are 0.
n=2 B4 to B3 are 0.
n=3 B4 is 0.

Cautionary notes

- Please use s+n−1 within the I/O range.
- If n = 0, the writing destination I/O is set to 0 because of DER = 0.
- If n ≥ 5, it is not executed.

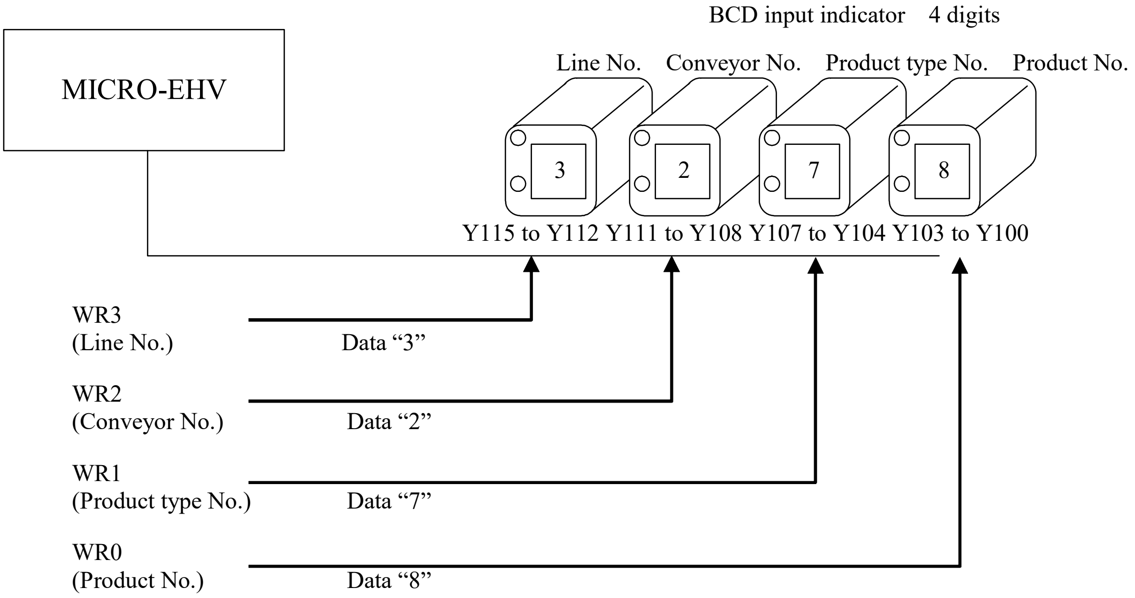
Program example



[Program description]

Connects the 4-digit BCD input indicator to WY10, and displays the individual data from WR0 to WR3 to each digit.

(Only data in the lower 4 bits from WR0 to WR3 are handled as valid data.)



Name	[Data operation] Distribute

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
DIST (d, s, n)	–	5	↕	●	●	●	●

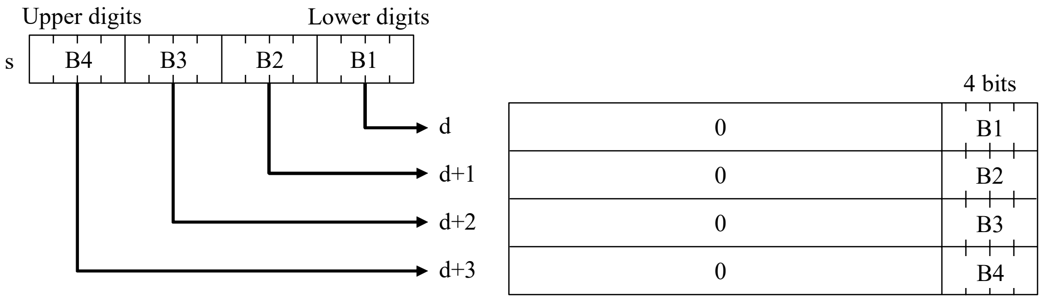
Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	9.46	10.74	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Writing destination I/O of the distributed result								✓	✓					
s	The top I/O to distribute							✓	✓	✓					
n	Number of words to distribute														✓

Remarks
n is from 0 to 4.

Function

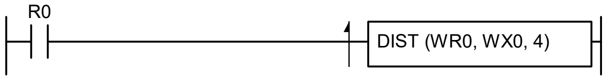
- s is distributed into each 4 bits, and the distributed value is set to the lower 4 bits in n words in sequence from d.
- The each upper 12 bits from d to d+n-1 is set to 0.
- The value of s does not change even if this command is executed.



Cautionary notes

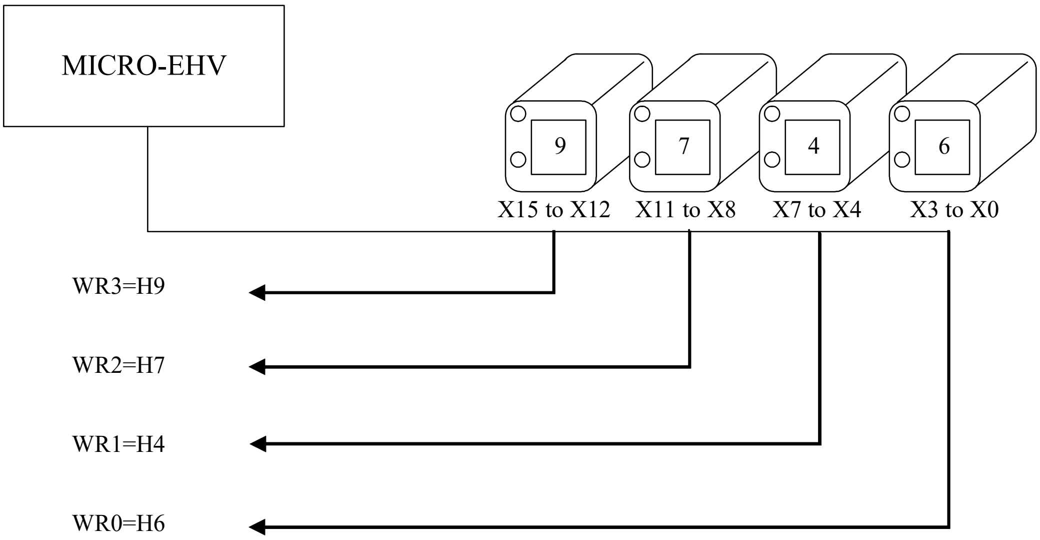
- Please user d+n−1 within the I/O range.
- If n = 0, the writing destination I/O is set to 0 because of DER = 0.
- If n ≥ 5, it is not executed.

Program example



[Program description]

Connects the input of the 4-bit 4-digit Digit switch to WX0 and stores data in each digit from WR0 to WR3 as an independent data.



Name	[Data operation] Combination of character data

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
SADD (d, s1, s2)	–	5	↕	●	●	●	●

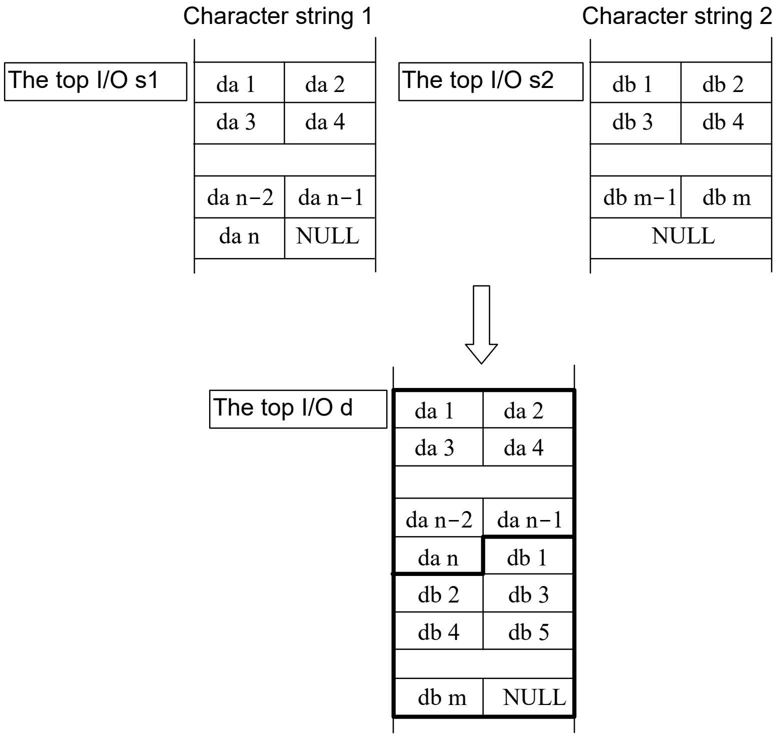
Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	45.46	51.24	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Writing destination top I/O of character string after uniting									✓					
s1	Character string 1 Top I/O									✓					
s2	Character string 2 Top I/O									✓					

Remarks

Function

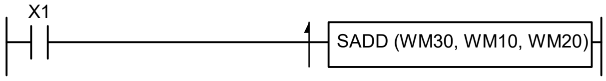
Two different tables are united to make one table.



Cautionary notes

- Please specify the internal outputs used for character string 1 and 2 within the range of the I/O No.
- If areas of character string 1 and 2 are overlapping, the operation is not performed because of DER = 1.
- Judges the end of data by NULL (H00) in character string 1 and 2 both. And NULL is set behind the character string after uniting.

Program example



[Program description]

Unites data from WM10 to NULL(H00) and data from WM20 to NULL at the rising of X1 and sets the result in and after WM30.

WM10	H4 8	4 9
WM11	H5 4	4 1
WM12	H4 3	4 8
WM13	H4 9	0 0

WM20	H4 E	4 8
WM21	H5 3	4 E
WM22	H5 2	4 9
WM23	H4 E	5 3
WM24	H0 0	0 0

⇨

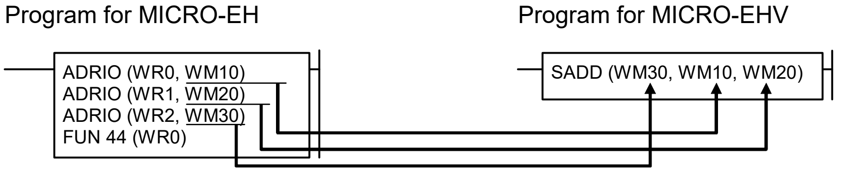
WM30	H4 8	4 9
WM31	H5 4	4 1
WM32	H4 3	4 8
WM33	H4 9	4 E
WM34	H4 8	5 3
WM35	H4 E	5 2
WM36	H4 9	4 E
WM37	H5 3	0 0

PRN ➔ PRJ

This command is equivalent to FUN 44 (s) in the program (PRN file) of EH-CPU.

How to convert the program which has used FUN 44 (s) into the program for MICRO-EHV is as follows.

FUN 44 (s) ➔ SADD ([I/O specified by s+2], [I/O specified by s], [I/O specified by s+1])



* If converted by Convert Tool started from Control Editor of software Ver.6.00 or newer, it is converted as mentioned above. Because it cannot be converted by Control Editor of software Ver.5.01 or before, please use Control Editor of the newest software version.

Name	[Data operation] Comparison of character data

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
SCMP (d, s1, s2)	–	5	↕	●	●	●	●

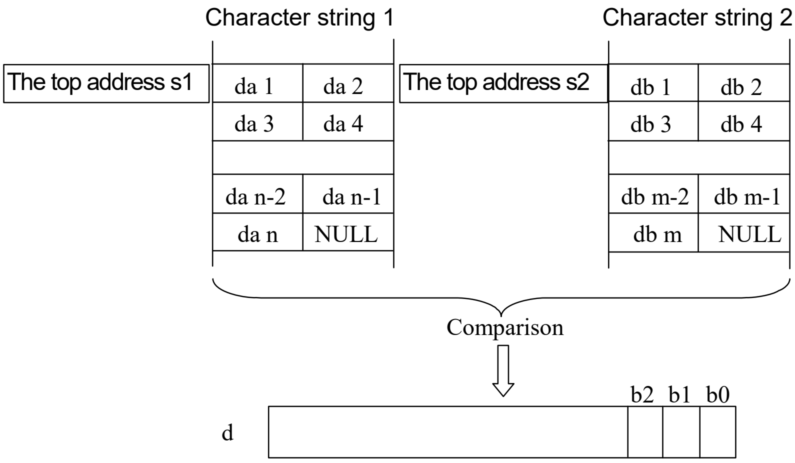
Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	109.86	118.44	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Comparison result									✓					
s1	Character string 1 Top I/O							✓	✓	✓					
s2	Character string 2 Top I/O							✓	✓	✓					

Remarks

Function

Data between the top and NULL(H00) in two different tables of which the top is the specified I/O are collated. The number of characters is compared first and then the character string is compared. If the number of characters is not matched, the character string is not compared because the comparison is terminated.

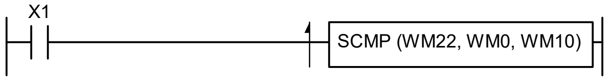


	b2	b1	b0
Number of character is not matched	1	0	0
Character string is not matched	0	1	0
Character string is matched	0	0	1

Cautionary notes

- Please specify the internal outputs used for the character string 1 and 2 within the range of the I/O No.
- If areas of the character string 1 and 2 are overlapping, the operation is not performed because of DER = 1.

Program example



[Program description]

Compares Data in and after WM0 with data in and after WM10 at the rising of X1 and sets the result in WM22.

PRN ➔ PRJ

This command is equivalent to FUN 45 (s) in the program (PRN file) of EH-CPU.

How to convert the program which has used FUN 45 (s) into the program for MICRO-EHV is as follows.

FUN 45 (s) ➔ SCMP (s+2, [I/O specified by s], [I/O specified by s+1])

Program for MICRO-EH

Program for MICRO-EHV

* If converted by Convert Tool started from Control Editor of software Ver.6.00 or newer, it is converted as mentioned above. Because it cannot be converted by Control Editor of software Ver.5.01 or before, please use Control Editor of the newest software version.

Name	[Data operation] Conversion Bit units ➔ Word unit

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
BITTOW (d, s, n)	–	5	↕	●	●	●	●

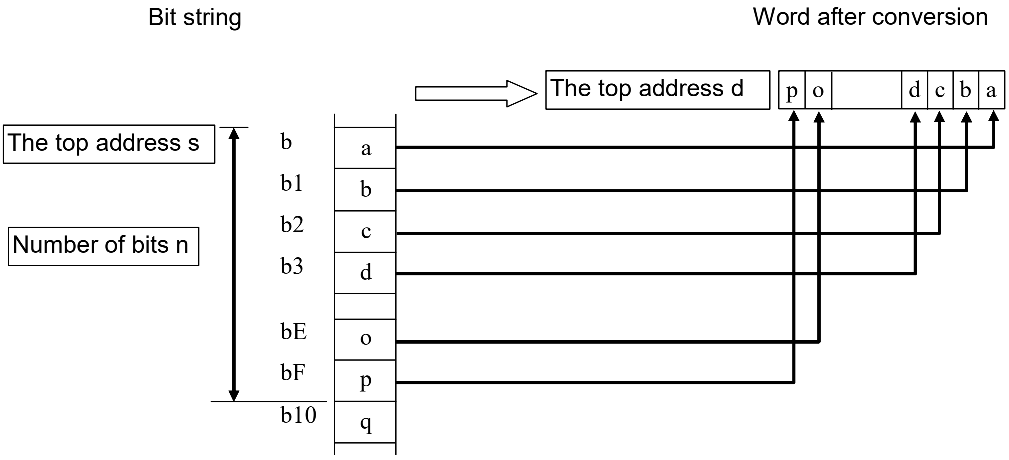
Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	5.78+6.4*n	6.3+6.36*n	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Word I/O to store bit data								✓					
s	Top bit I/O to store in word data	✓	✓	✓										
n	Number of conversion bits						✓	✓	✓	✓				✓

Remarks
n is from 0 to 16.

Function

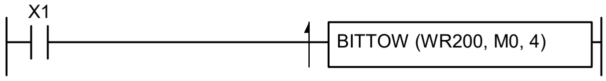
n bits from the bit I/O specified by s are set in the word I/O specified by d in sequence from the lower bit. If the number of bits specified is less than 16 (if from 1 to 15 is specified), the upper bits in the word I/O are set to 0.



Cautionary notes

- Please specify the internal outputs used for the bit string and the word after conversion within the range of the I/O No. If the bit I/O address exceeds the maximum of the I/O No., data is developed within the range of the I/O's specification, but DER = 1.
- If areas of the bit string and the word after conversion are overlapping, the operation is not performed because of DER = 1.
- If the number of bits exceeds 16, it is not processed because of DER = 1.
- If the number of bits is 0, it is not processed, and DER = 0.

Program example



[Program description]

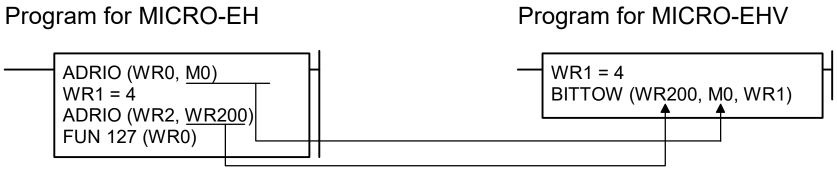
Sets the bit data from M0 to M3 (4 bits) in WR200 in sequence from the lower bit and sets 0 to other bits at the rising of X1.

PRN ➔ PRJ

This command is equivalent to FUN 127 (s) in the program (PRN file) of EH-CPU.

How to convert the program which has used FUN 127 (s) into the program for MICRO-EHV is as follows.

FUN 127 (s) ➔ BITTOW ([I/O specified by s+2], [I/O specified by s], s+1)



* If converted by Convert Tool started from Control Editor of software Ver.6.00 or newer, it is converted as mentioned above. Because it cannot be converted by Control Editor of software Ver.5.01 or before, please use Control Editor of the newest software version.

Name	[Data operation] Conversion Word unit ➔ Bit units

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
WTOBIT (d, s, n)	–	5	↕	●	●	●	●

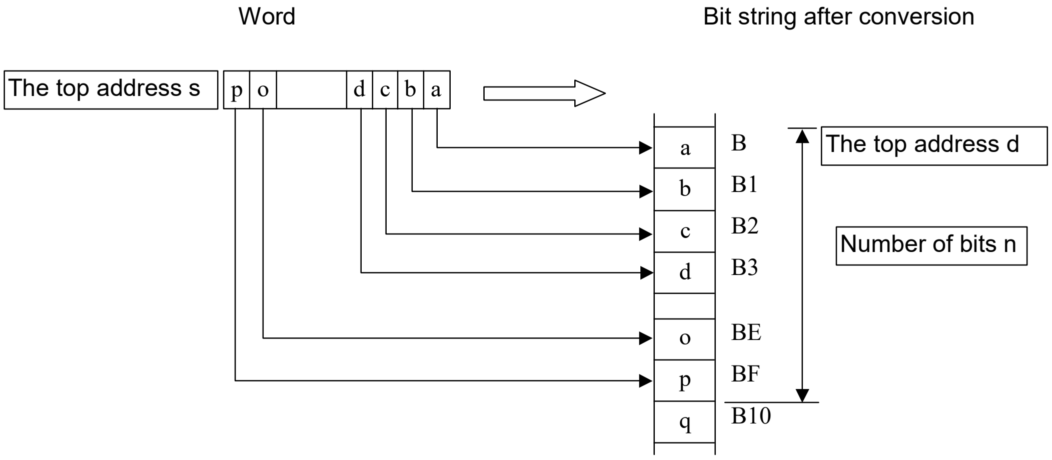
Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	5.38+7.95*n	5.84+8.44*n	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Top I/O of bit string after conversion	✓	✓	✓											
s	Word I/O to develop to bit data									✓					
n	Number of conversion bits							✓	✓	✓	✓				✓

Remarks
n is from 0 to 16.

Function

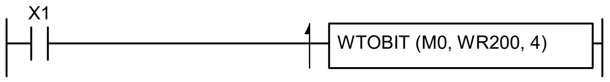
n bits from the 0th bit in the word I/O specified by s are developed, of which the top is the bit I/O specified by d.



Cautionary notes

- Please specify the internal outputs used for word data and the bit string after conversion within the range of the I/O No. If the bit I/O address exceeds the maximum of the I/O No., data is developed within the range of the I/O's specification, but DER = 1.
- If areas of word data and the bit string after conversion are overlapping, the operation is not performed because of DER = 1.
- If the number of bits exceeds 16, it is not processed because of DER = 1.
- If the number of bits is 0, it is not processed, and DER = 0.

Program example



```
WTOBIT (M0, WR200, 4)
```

[Program description]

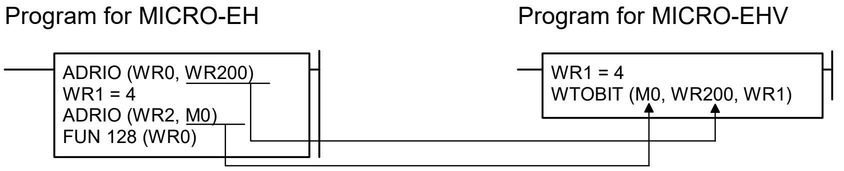
Picks out the lower 4 bits of WR200 and sets the result from M0 to M3 at the rising of X1. (The least significant bit of WR200 is stored in M0.)

PRN ➔ PRJ

This command is equivalent to FUN 128 (s) in the program (PRN file) of EH-CPU.

How to convert the program which has used FUN 128 (s) into the program for MICRO-EHV is as follows.

FUN 129 (s) ➔ WTOBIT ([I/O specified by s], [I/O specified by s+2], s+1)



```
ADRIO (WR0, WR200)
WR1 = 4
ADRIO (WR2, M0)
FUN 128 (WR0)
```

```
WR1 = 4
WTOBIT (M0, WR200, WR1)
```

* If converted by Convert Tool started from Control Editor of software Ver.6.00 or newer, it is converted as mentioned above. Because it cannot be converted by Control Editor of software Ver.5.01 or before, please use Control Editor of the newest software version.

Name	[Data operation] Increment

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
INC (d)	—	3	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
Word	6.16	6.66	—	—	—
Double word	7.92	9.18			

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Incremental I/O								✓	✓	✓		✓	✓	

Remarks

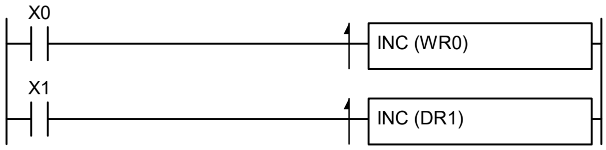
Function

The value of the internal output specified by d parameter increases by 1 whenever the command is executed.

Cautionary notes

- If the internal output specified to d parameter is HFFFF in word, it becomes H0 by adding 1.
- If the internal output specified to d parameter is HFFFFFFFF in double word, it becomes H0 by adding 1.

Program example



[Program description]

- Adds 1 to WR0 at the rising of X0.
- Adds 1 to DR1 at the rising of X1.

PRN ➔ PRJ

This command is equivalent to FUN 123 (s) / FUN 124 (s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 123 (s) / FUN 124 (s) into the program for MICRO-EHV is as follows.

FUN 123 (s) ➔ INC (s)

Example) FUN 123 (WR100) ➔ INC (WR100)

FUN 124 (s) ➔ INC (s), provided that s is double word.

Example) FUN 124 (WR100) ➔ INC (DR100)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	[Data operation] Decrement

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
DEC (d)	—	3	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
Word	6.18	6.66	—	—	—
Double word	8.46	9.16	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT, TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Decremental I/O							✓	✓	✓		✓	✓	

Remarks

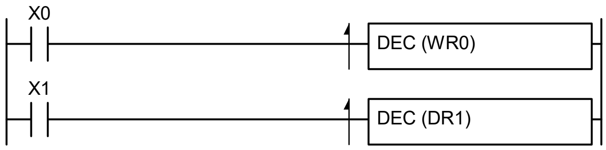
Function

The value of the internal output specified to d parameter decreases by 1 whenever the command is executed.

Cautionary notes

- If the internal output specified to d parameter is H0 in word, it becomes HFFFF by subtracting 1.
- If the internal output specified to d parameter is H0 in double word, it becomes HFFFFFFFF by subtracting 1.

Program example



[Program description]

- Subtracts 1 from WR0 at the rising of X0.
- Subtracts 1 from DR1 at the rising of X1.

PRN ➔ PRJ

This command is equivalent to FUN 125 (s) / FUN 126 (s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 125 (s) / FUN 126 (s) into the program for MICRO-EHV is as follows.

FUN 125 (s) ➔ DEC (s)

Example) FUN 125 (WR100) ➔ DEC (WR100)

FUN 126 (s) ➔ DEC (s), provided that s is double word.

Example) FUN 126 (WR100) ➔ DEC (DR100)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	[Data search] Search for word data

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
DSRCH (d, s1, s2, n)	–	6	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	6.88+8.39*n	6.68+8.66*n	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Search result								✓	✓	✓				
s1	Data to be searched							✓	✓	✓	✓				✓
s2	Top I/O in search area							✓	✓	✓	✓				
n	Number of search data							✓	✓	✓					✓

Remarks
d parameters are occupied use up to d+1.

Function

The specified data is searched from a data group of the specified range. The first searched position and the number of data searched from the specified range are output to from d to d+1.

Parameter

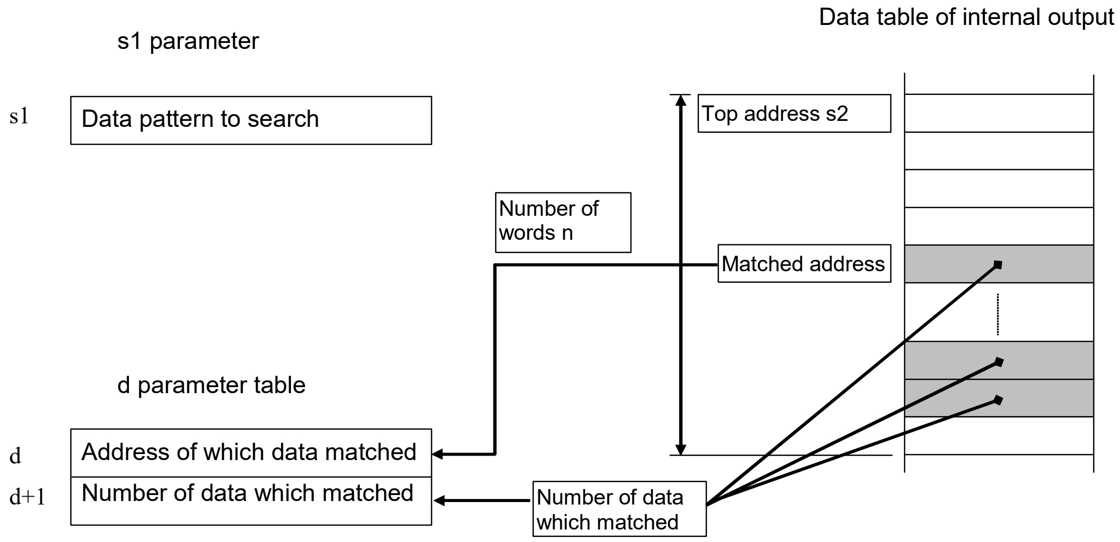
d : The data position searched first (a relative position from the top I/O) is stored.

d+1 : The number matched to data to be searched in the specified area is stored.

s1 : Specifies Data to search or the internal output in which data is stored.

s2 : Specifies the top I/O in the area to search.

n : Specifies the number of words in the are to search.



Cautionary notes

- Please specify the internal output used for s1, s2, n, and d and the search area within the range of the I/O No.
- Please pay attention that the search area does not overlap with s1, s2, and d parameters.
  If the area overlaps, the command is not executed because of DER = 1.

Program example

```
DSRCH (WR103, H1010, WM0, H100)
```

[Program description]

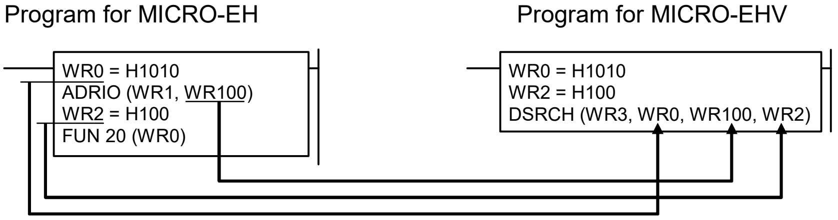
Searches 256 words (H100 word) from WM0 for the data H1010 at the rising of X0. The search result is set in WR103 (data position) and WR104 (number of data).

PRN ➔ PRJ

This command is equivalent to FUN 20 (s) in the program (PRN file) of MICRO-EH.

How to convert the program whish has used FUN 20 (s) into the program for MICRO-EHV is as follows.

FUN 20 (s) ➔ DSRCH (s+3, s, [I/O specified by s+1], s+2)


Program for MICRO-EH

```
WR0 = H1010
ADRIO (WR1, WR100)
WR2 = H100
FUN 20 (WR0)
```

Program for MICRO-EHV

```
WR0 = H1010
WR2 = H100
DSRCH (WR3, WR0, WR100, WR2)
```

* If converted by Convert Tool started from Control Editor of software Ver.6.00 or newer, it is converted as mentioned above. Because it cannot be converted by Control Editor of software Ver.5.01 or before, please use Control Editor of the newest software version.

Name	[Data search] Extract data table

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
TSRCH (d, s, n1, n2)	—	6	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	6.84	7.28	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Top I/O to store in extract table							✓	✓	✓				
s	Top I/O in data table						✓	✓	✓	✓				
n1	Size of 1 block						✓	✓	✓	✓				✓
n2	Block No. to extract						✓	✓	✓					✓

Remarks

Function

The data with the specified block number is extracted from the data block group of the specified range. The extracted data is copied to the specified drawing area.

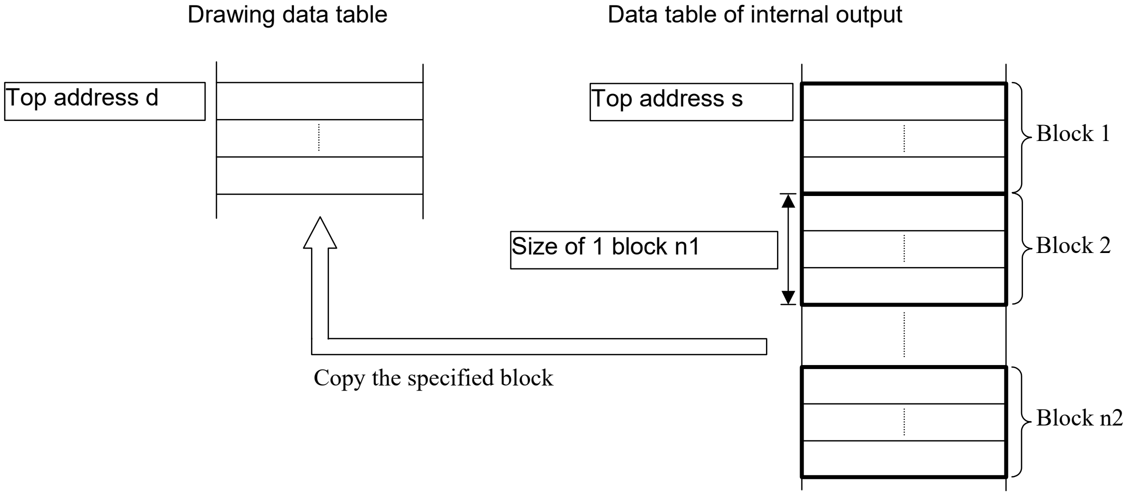
Parameter

d : Specifies the internal output stored the extracted table.

s : Specifies the top I/O in the table to extract data.

n1 : Specifies the size of data block (number of words).

n2 : Specifies the number of data blocks.



Cautionary notes

- Please specify the internal output used for d and s parameters, the data table, and the drawing data table within the range of the I/O No.
- Please pay attention that all kinds of table area do not overlap with d and s parameters. If the area overlaps, the command is not executed because of DER = 1.

Program example

[Program description]

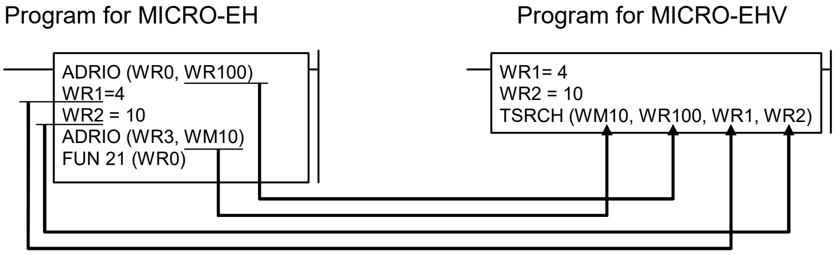
Draws the 10th data block counting from WM0 out of the data table consisting of 1 block with 2 words, and sets the result in and after WM100.

PRN ➔ PRJ

This command is equivalent to FUN 21 (s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 21 (s) into the program for MICRO-EHV is as follows.

FUN 21 (s) ➔ TSRCH ([I/O specified by s+3], [I/O specified by s], s+1, s+2)



* If converted by Convert Tool started from Control Editor of software Ver.6.00 or newer, it is converted as mentioned above. Because it cannot be converted by Control Editor of software Ver.5.01 or before, please use Control Editor of the newest software version.

Name	[Data search] Search for Maximum / Minimum / Average

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
VSRCH (d, s1, s2, n)	–	6	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	8.06+0.35*n	8.96+0.3*n	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Search result								✓	✓	✓				
s1	Search types							✓	✓	✓	✓				✓
s2	Top I/O in area to be searched							✓	✓	✓	✓				
n	Number of data to be searched							✓	✓	✓					✓

Remarks
d parameters are occupied use up to d+2 or d+5.

Function

The average, minimum and maximum are computed from the specified data table. (It can be searched selecting the classification from integer [word / double word] and real number by specifying the s parameter.)

Parameter

d : The search result is stored .

s1 : Specifies the classification of the numerical value to search and the classification of search.

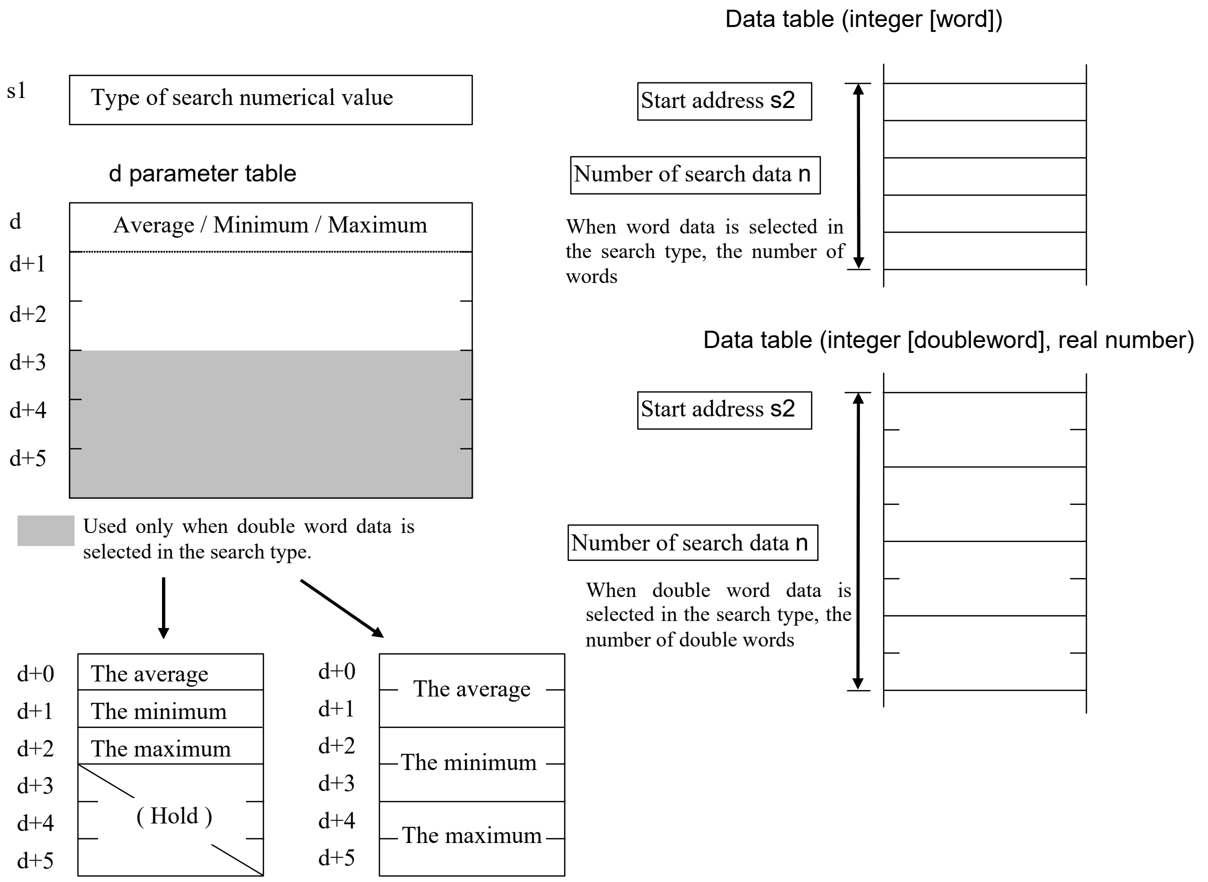
Classification of the number		Set value
Integer data (At word)	Unsigned	H0001
	Signed	H0002
Integer data (At double word)	Unsigned	H0004
	Signed	H0008
Floating point data		H000F

s2 : The top I/O in the area to be searched

n : Specifies the number of data in the area to be searched. The valid ranges of n are as follows.

At specified word: 1 to 65,535 (in decimal), H0001 to HFFFF (in hexadecimal)

At specified double word and floating point: 1 to 32,767 (in decimal), H0001 to H7FFF (in hexadecimal)



Cautionary notes

- When the specified value of search classification is abnormal, the operation is not performed because of DER = 1.
- When specifying the integer (word) at classification of search number, the calculation result of only from d to d+2 is stored.(From d+3 to d+5 holds the value before the command execution.)
- When specifying the integer at classification of search number, the average is the value of which a fraction to a decimal point is rounded down.
- If the area (s2) of the data table overlaps with the area (d) of s parameter, the operation is not performed because of DER = 1.
- If the number of data to be searched (n) is 0, the operation is not performed because of DER = 1.
- If the result is without the range from –1e+37 to 1e+37 at the operation of floating decimal data, the result is not output because of DER = 1.
- If the value of s parameter or the data table is changed during this command execution, the correct operation result cannot be obtained.

Program example

[Program description]

Calculates the average, the minimum, and the maximum of the 36-word unsigned integer data from WR100 at the rising of X0 and sets the result in WR0 (the average), WR1 (the minimum), and WR2 (the maximum) respectively.

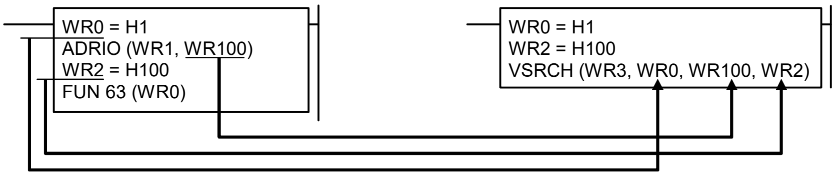
PRN ➔ PRJ

This command is equivalent to FUN 63 (s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 63 (s) into the program for MICRO-EHV is as follows.

FUN 63 (s) ➔ VSRCH (s+3, s, [I/O specified by s+1], s+2)

Program for MICRO-EH   Program for MICRO-EHV



```
WR0 = H1
ADRIO (WR1, WR100)
WR2 = H100
FUN 63 (WR0)
```

```
WR0 = H1
WR2 = H100
VSRCH (WR3, WR0, WR100, WR2)
```

* If converted by Convert Tool started from Control Editor of software Ver.6.00 or newer, it is converted as mentioned above. Because it cannot be converted by Control Editor of software Ver.5.01 or before, please use Control Editor of the newest software version.

Name	[Data exchange] Swapping

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
SWAP (d, s)	Word	4	●	●	●	●	●
	Double word	5					

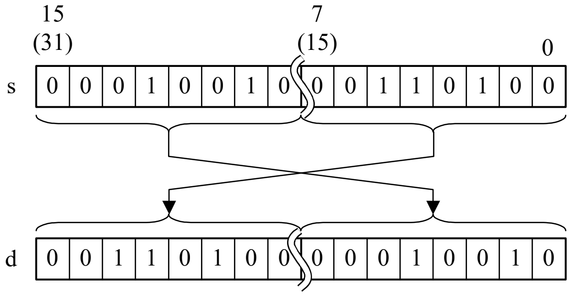
Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
Word	7.28	7.72	—	—	—
Double word	8.36	9	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT, TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O after exchange							✓	✓	✓		✓	✓	
s	I/O to exchange						✓	✓	✓	✓	✓	✓	✓	

Remarks

Function

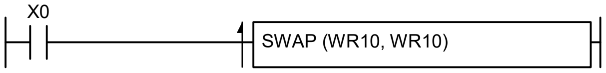
- When s is word, the lower 8 bits are exchanged for the upper 8 bits in the content of s and the result is stored in d.
- When s id double word, the lower word is exchanged for the upper word in the content of s and the result is stored in d.



Cautionary notes

Please set a start up of this command to the edge trigger.

Program example



[Program description]

Exchanges the lower 8 bits for the upper 8 bits in WR10 at the rising of X0 and stores the result in WR10.

WR10 H 1 2 3 4 ⟹ WR10 H 3 4 1 2

* Since this command is executed at every scan if the start up is not set to the edge trigger, the upper and the lower in WR10 are exchange at every scan.

PRN ➔ PRJ

This command is equivalent to SWAP (d) in the program (PRN file) of MICRO-EH.

How to convert the program which has used SWAP (d) into the program for MICRO-EHV is as follows.

SWAP (d) ➔ SWAP (d, d) All d are the same I/O.

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	[Data exchange] Block exchange

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
XCG (d1, d2, n)	–	5	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
Bit	8.44+3.58*n	8.68+3.78*n	–	–	–
Word	6.04+2.656*n	6.84+2.98*n	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d1	Top I/O of exchange destination			✓						✓					
d2	Top I/O of exchange source			✓						✓					
n	Number of bits to exchange Number of words to exchange							✓	✓	✓	✓				✓

Remarks

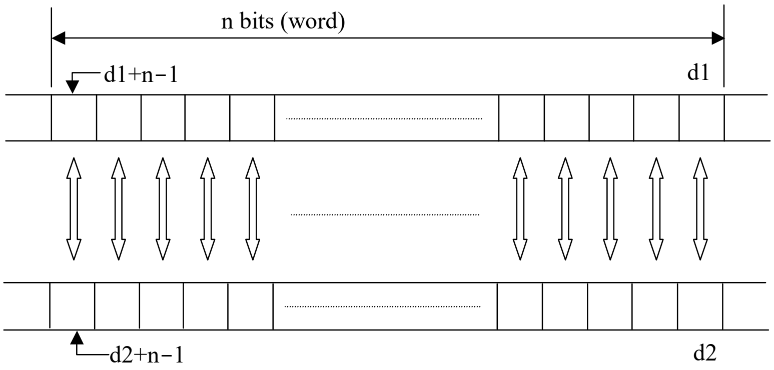
Function

- The content from d1 to n bits (words) is exchanged for the content from d2 to n bits (words).
- When n is words,

  specifies the number of bits (words) exchanged for the content (0 to 255) of the lower 8 bits (b7 to b0) in n (WX, WY, WR, WM, TC). (The upper bits are ignored and it is considered to be '0'.)

  n (a constant) can specify from 0 to 255. (Decimal number)
- The combination of d1 and d2 are as follows.

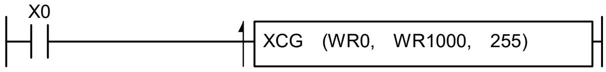
d1	d2
Bit	Bit
Word	Word



Cautionary notes

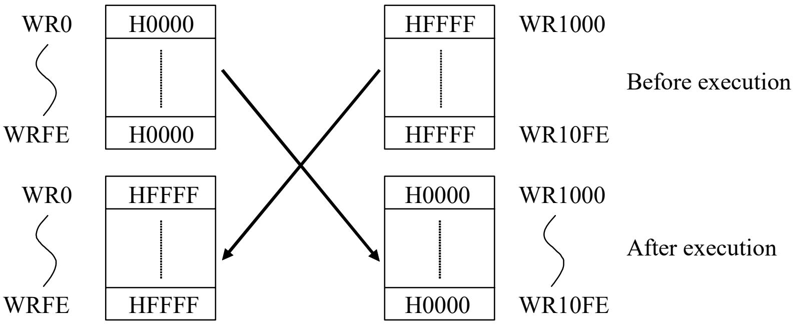
- If n = 0, the batch exchange is not performed. DER becomes 0.
- Please use d1+n−1 and d2+n−1 within the I/O range. If exceeded, it is exchanged up to the maximum range of the number of bits (words) of smaller one of the number of bits (words) specified to d1 and d2 because of DER = 1.

Program example



[Program description]

Exchanges the content from WR1000 to WR10FE for the content from WR0 to WRFE at the rising of X0.



Name	[Data transfer] Block transfer

Ladder format	Number of steps			Condition code				
	Condition		Steps	R7F4	R7F3	R7F2	R7F1	R7F0
	d	s		DER	ERR	SD	V	C
MOV (d, s, n)	I/O	I/O	5	↕	●	●	●	●
	I/O	Array(I/O / C)	6					
	Array(I/O)	I/O	6					
	Array(I/O)	Array(I/O / C)	7					
	Array(C)	I/O	6					
	Array(C)	Array(I/O / C)	7					

Command processing time (μs)

Condition	Time			
	MVH (High function)		MVL (Standard)	
	Bit	Word	Bit	Word
d : I/O, s : I/O	6.5+5.34*n	5.02+4.14*n	7.2+5.74*n	5.34+4.36*n
d : I/O, s : Array(I/O / C)	7.16+5.86*n	5.35+4.69*n	7.94+6.44*n	5.67+4.98*n
d : Array(I/O / C), s : I/O	6.9+6.09*n	5.86+5.11*n	7.06+6.48*n	6.14+5.42*n
d : Array(I/O / C), s : Array(I/O / C)	7.2+6.51*n	5.63+5.67*n	7.3+7.16*n	5.94+6.06*n

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Transfer destination Top I/O		✓	✓				✓	✓	✓				
s	Transfer source Top I/O	✓	✓	✓			✓	✓	✓	✓				
n	Number of transfer bits (words)						✓	✓	✓	✓				✓
( )	Index value						✓	✓	✓					✓

Remarks

- C means a constant.
- n is the number of words.

Function

- n bits (or words) of I/O data specified by s are transferred to the I/O specified by d.

  The combinations of d, s, and n are shown below.

d	s	n	Remarks
Bit	Bit	A constant	n is from 0 to 1023.
		Word I/O	n is data from b0 to b9.
Word	Word	A constant	n is from 0 to 1023.
		Word I/O	n is data from b0 to b9.

- The value form s to s+n-1 is held.
- If ranges of a transferring source and a transferring destination overlap, it changes to a transferred value.
- An array constant can be used for d and s parameters.

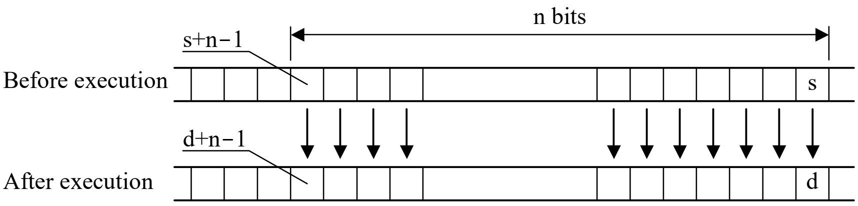
  Example) MOV (WR100(WR0), WR1000(WR0), 32)

  If WR0 = H10, the 32-word data from WR110 is transferred to the 32-word from WR1010.

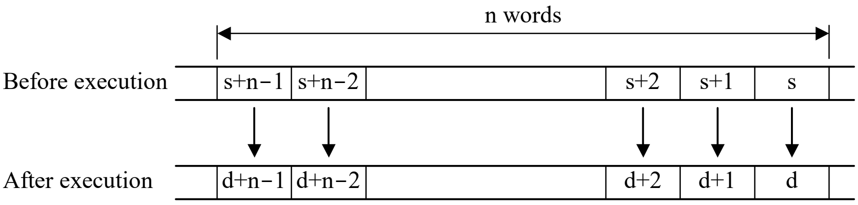
  * MICRO-EHV has no command which is equivalent to FUN 120 (Index setting / Argument d), FUN 121 (Index setting / Argument s) and FUN122 (Index canceling) in MICRO-EH.

- Bit ➔ Bit

Data is transferred from the bit I/O to the bit I/O.



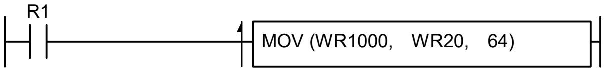
- Word ➔ Word



Cautionary notes

- Please use d+n−1 and s+n−1 within the range of the I/O for MICRO-EHV. If exceeded, it is transferred up to the maximum range because of DER(R7F4) = 1.
- If n = 0, the batch transfer is not performed. DER becomes 0.
  If usable maximum I/O No. is exceeded when using the array, it is transferred up to the maximum range because of DER = 1.

Program example

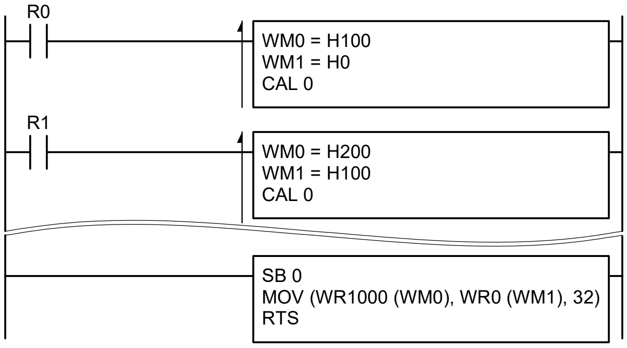


```
MOV (WR1000,   WR20,   64)
```

[Program description]

Transfers the 64 word data from the rising of R1.

The transfer area is set to from WR20 to WR5F and from WR1000 to WR103F, respectively.



```
WM0 = H100
WM1 = H0
CAL 0
```

```
WM0 = H200
WM1 = H100
CAL 0
```

```
SB 0
MOV (WR1000 (WM0), WR0 (WM1), 32)
RTS
```

[Program description]

- Calls SB 0 at the rising of R0, and transfers the 32 words from WR0 to the 32 words from WR1100.
- Calls SB 0 at the rising of R1, and transfers the 32 words from WR100 to the 32 words from WR1200.

Name	[Data transfer] Bit block transfer

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
BMOV (d, s, n1, n2)	–	6	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d : Bit / s : Bit	26+15.2*n	32.7+23*n	–	–	–
d : Bit / s : Word	37.1+15.3*n	54.4+13.4*n	–	–	–
d : Word / s : Bit	44.4+15.5*n	64.9+15.5*n	–	–	–
d : Word / s : Word	46+8.2*n	41.3+26.9*n	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Transfer destination Top I/O		✓	✓				✓	✓	✓				
s	Transfer source Top I/O	✓	✓	✓			✓	✓	✓	✓				
n1	Number of bits for 1 block						✓	✓	✓	✓				✓
n2	Number of blocks for transferring						✓	✓	✓	✓				✓

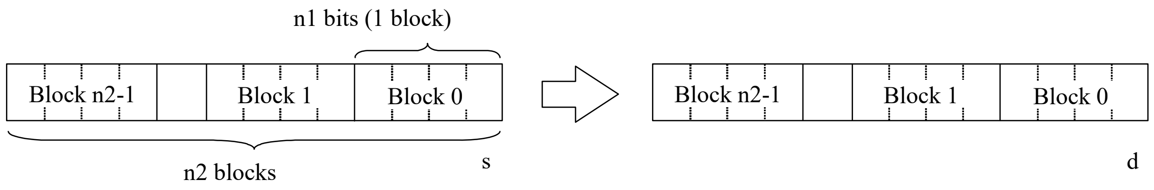
Remarks
n1 is from 1 to 16. n2 is from 1 to 65535. n for processing time is the number of blocks for transferring of n2.

Function

This command transfers the specified number of blocks to another area, considering some bits to be one block.

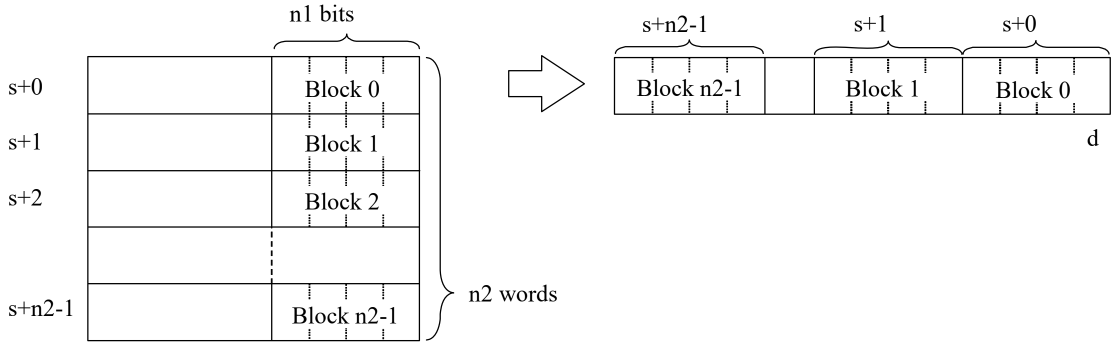
(1) When specifying both d and s to the bit I/O.

Handling n1 bits as one block from the bit I/O specified by s, n2 blocks are transferred setting the bit I/O specified by d as the top.



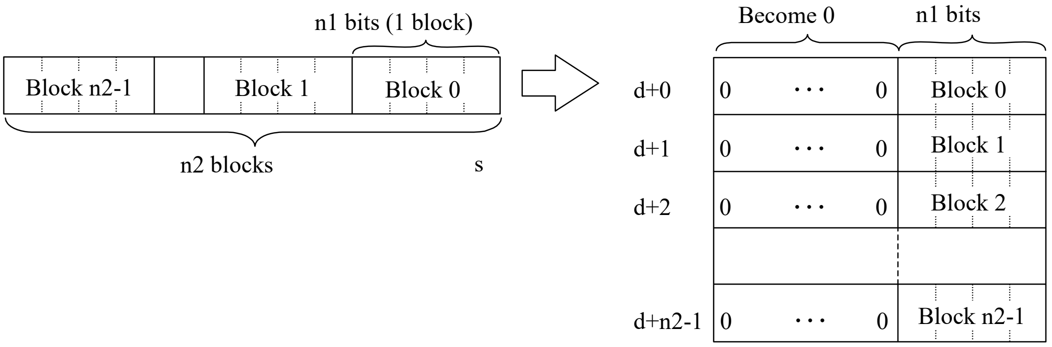
(2) When specifying d to the bit I/O and specifying s to the word I/O.

The lower n1 bits in n2 words from the word I/O specified bys are transferred setting the bit I/O specified by d as the top.



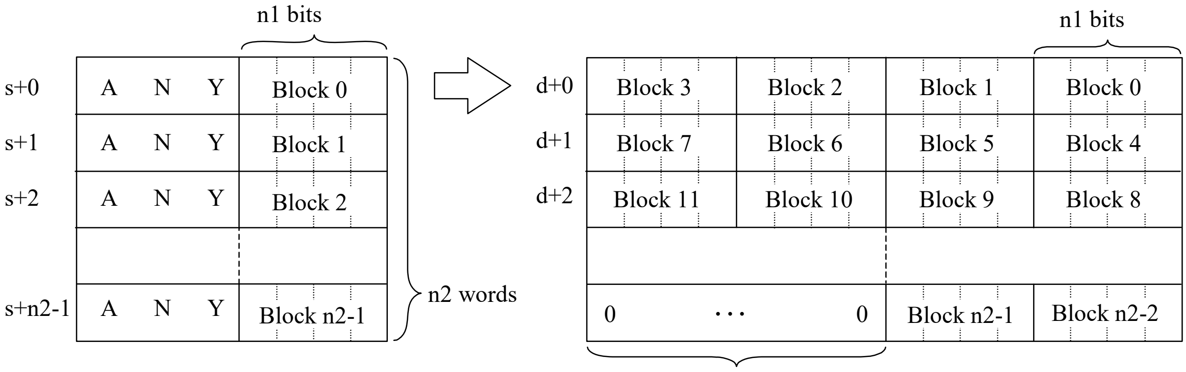
(3) When specifying d to the word I/O and specifying s to the bit I/O.

Handling n1 bits from the bit I/O specified by s as one block, n2 blocks are transferred to the lower in the word I/O setting the word I/O specified by d as the top.



(4) When specifying both d and s to the word I/O.

Handling the lower n1 bits in the word I/O specified by s as one block, n2 blocks are transferred setting the word I/O specified by d as the top. (Each block is stored consecutively.)

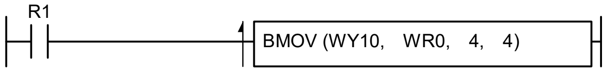


If the transfer result is less than 1 word, the upper part stores 0.

Cautionary notes

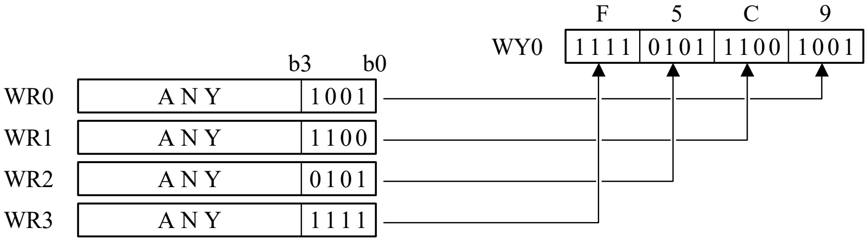
- Holds the value of s which is the transfer source. But if you specify so that the transfer destination and the transfer source are overlapped, it changes to the transferred value.
- When d is the word I/O and s is the bit I/O, the upper bits in d after transferring becomes 0.
- Please use the I/O of the transfer source and the transfer destination within the I/O range of MICRO-EHV. If exceeded, it is transferred up to the maximum range because of DER(R7F4) = 1.
- n1 is valid from 0 to 16. If the value outside the valid range is set, the command is not performed because of DER = 1.
- n2 is valid from 0 to 65535. If the value outside the valid range is set, the command is not performed because of DER = 1.
- If n1 = 0 or n2 = 0, the batch transfer is not performed. DER becomes 0.

Program example



[Program description]

Transfers 4 blocks to WY10 at the rising of R1, handling 4 bits from the lower from WR0 to WR3 as one block.



Name	[Data transfer] Copy

Ladder format	Number of steps			Condition code				
	Condition		Steps	R7F4	R7F3	R7F2	R7F1	R7F0
	d	s		DER	ERR	SD	V	C
COPY (d, s, n)	I/O	I/O / C	5	↕	●	●	●	●
	I/O	Array(I/O / C)	6					
	Array(I/O)	I/O / C	6					
	Array(I/O)	Array(I/O / C)	7					
	Array(C)	I/O / C	6					
	Array(C)	Array(I/O / C)	7					

Command processing time (μs)				
Condition	Time			
	MVH (High function)		MVL (Standard)	
	Bit	Word	Bit	Word
d : I/O, s : I/O / C	9.19+4.69*n	4.83+2.74*n	10.08+2.2*n	5.12+2.84*n
d : I/O, s : Array(I/O / C)	6.75+3.21*n	5.87+3.73*n	7.52+3.52*n	6.14+3.96*n
d : I/O, s : Array(I/O / C)	9.53+2.77*n	5.59+3.95*n	10.4+3.02*n	5.82+4.12*n
d : Array(I/O / C), s : Array(I/O / C)	7.08+3.8*n	4.65+4.39*n	7.78+4.22*n	5.96+4.6*n

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Transfer destination Top I/O		✓	✓				✓	✓	✓				
s	Transfer source Value	✓	✓	✓			✓	✓	✓	✓				✓
n	Number of transfer bits (words)						✓	✓	✓	✓				✓
( )	Index value						✓	✓	✓					✓

Remarks
C means a constant. n is the number of words.

Function

- n bits (or words) of the I/O data specified by s is copied to the I/O specified by d.

  The combinations of d, s, and n are shown below.

d	s	n	Remarks
Bit I/O	Bit I/O or bit data (0 or 1)	A constant	n is from 0 to 1023.
		Word I/O	n is data from b0 to b9.
Word I/O	Word I/O or word data (0 to 65535)	A constant	n is from 0 to 1023.
		Word I/O	n is data from b0 to b9.

- The value of s is held.
- If ranges of the transfer source (s) and the transfer destination (d) overlap, it changes to the copied value.
- An array variable can be used for d and s parameters.

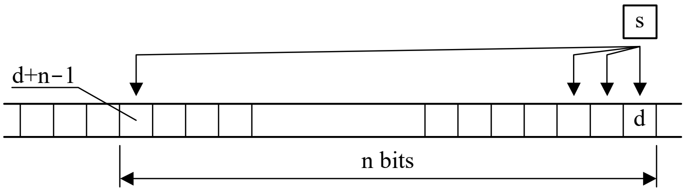
  Example) COPY (R100(WR10), M0(WR10), 16)

  If WR10 = H20, data in M20 is copied to 16 bits from R120 to R12F.

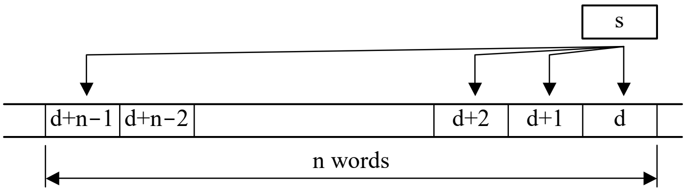
  * MICRO-EHV has no command which is equivalent to FUN 120 (Index setting / Argument d), FUN 121 (Index setting / Argument s), and FUN122 (Index canceling) in MICRO-EH.

- Bit ➔ Bit

Data is transferred from the bit I/O to the bit I/O.



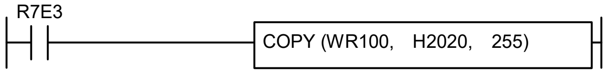
- Word ➔ Word



Cautionary notes

- Please use d+n-1 and s+n-1 within the I/O range of MICRO-EHV. If exceeded, it is transferred up to the maximum range because of DER = 1.
- If n = 0, the batch transfer is not performed. DER(R7F4) becomes 0.
- If usable maximum I/O No. is exceeded in using the array, it is transferred up to the maximum range because of DER = 1.

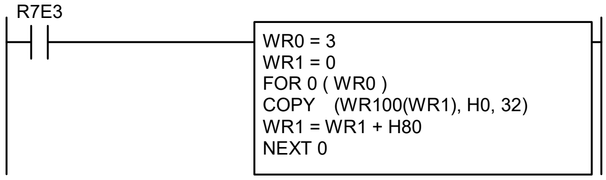
Program example



```
COPY (WR100,   H2020,   255)
```

[Program description]

Covers the communication data area set from WR100 to WR1FE with the space code (H20) as the default value at the first scan after RUN.



```
WR0 = 3
WR1 = 0
FOR 0 ( WR0 )
COPY   (WR100(WR1), H0, 32)
WR1 = WR1 + H80
NEXT 0
```

[Program description]

Using the array, sets 0 in the three areas.

Sets 0 from WR100 to WR11F, from WR180 to WR19F, and from WR200 to WR21F at the first scan after RUN.

Name	[Data transfer] Bit block copy

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
BCOPY (d, s, n1, n2)	–	6	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d : Bit / s : Bit	39.2+9.6*n	52.1+12.4*n	–	–	–
d : Bit / s : Word	32.2+10.6*n	42.3+13.3*n	–	–	–
d : Word / s : Bit	43.1+2.2*n	56.6+2.2*n	–	–	–
d : Word / s : Word	37.6+7.6*n	45.7+11.2*n	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT, TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Transfer destination Top I/O		✓	✓				✓	✓	✓				
s	Transfer source Top I/O	✓	✓	✓			✓	✓	✓	✓				✓
n1	Number of bits in 1 block						✓	✓	✓	✓				✓
n2	Number of copy blocks						✓	✓	✓	✓				✓

Remarks

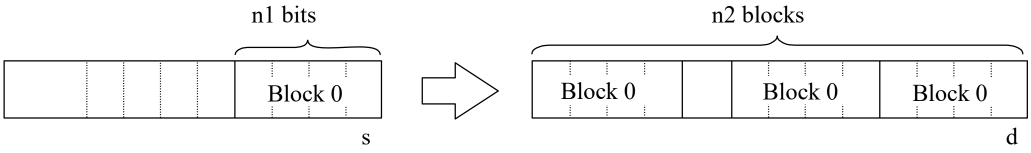
n1 is from 1 to 16.
n2 is form 1 to 65535
n for processing time is the number of copy blocks of n2.

Function

This command copies the specified block to another area handling some bits as one block.

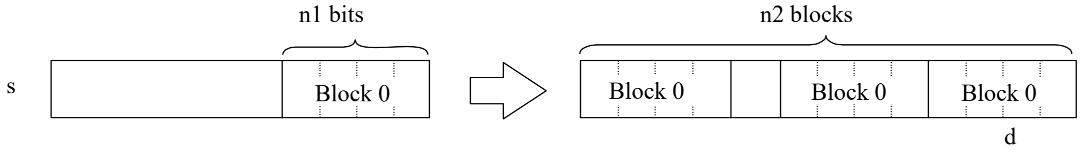
(1) When specifying both d and s to the bit I/O.

Handling n1 bits from the bit I/O specified by s as one block, the same block is copied n2 times, setting the bit I/O specified by d as the top.



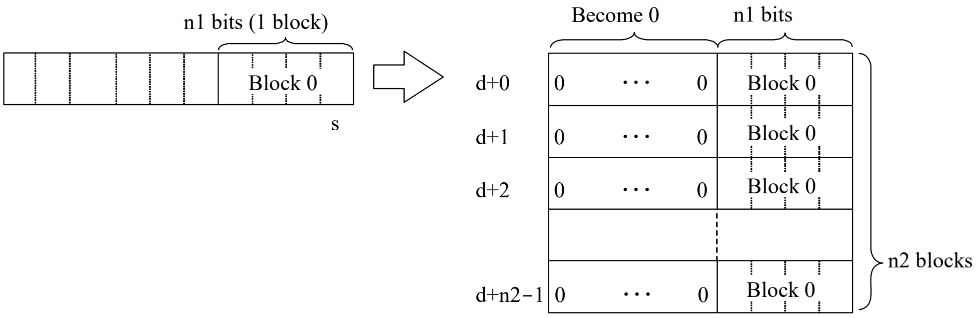
(2) When specifying d to the bit I/O and specifying s to the word I/O.

Handling the lower n1 bits in the word I/O specified by s as one block, the same block is copied n2 times, setting the bit I/O specified by d as the top.



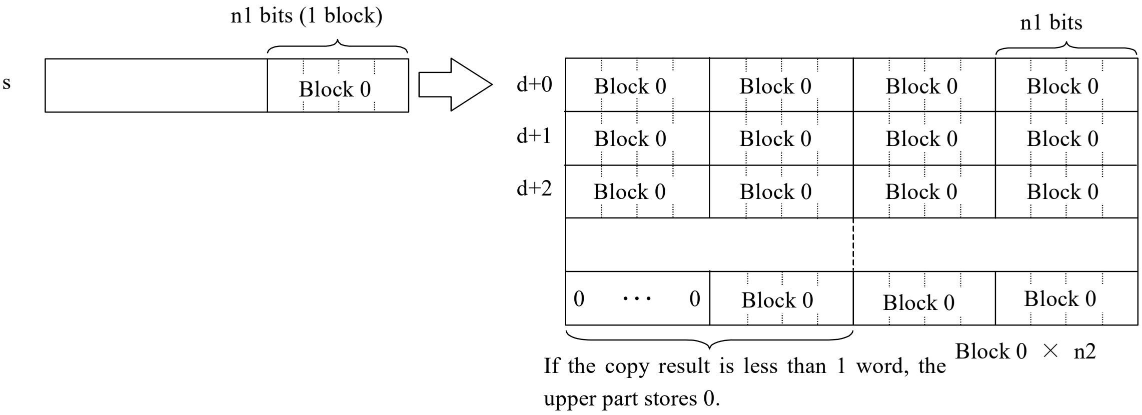
(3) When specifying d to the word I/O and specifying s to the bit I/O.

Handling n1 bits from the bit I/O specified by s as one block, the same block is copied to the lower part in the word I/O n2 times, setting the word I/O specified by d as the top.



(4) When specifying both d and s to the word I/O.

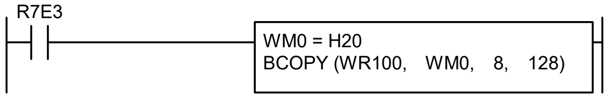
Handling the lower n1 bits in the word I/O specified by s as one block, the same block is copied n2 times, setting the word I/O specified by d as the top. (Each block is stored consecutively.)



Cautionary notes

- The value of s which is a transfer source is held. But if you specify so that the transfer destination and the transfer source are overlapped, it changes to the transferred value.
- When d is the word I/O and s is the bit I/O, the upper bits in d after transferring becomes 0.
- Please use the I/O of the transfer source and the transfer destination within the I/O range of MICRO-EHV. If exceeded, it is transferred up to the maximum range because of DER(R7F4) = 1.
- n1 is valid from 0 to 16. If the value outside the valid range is set, the command is not performed because of DER = 1.
- n2 is valid from 0 to 65535. If the value outside the valid range is set, the command is not performed because of DER = 1.
- If n1 = 0 or n2 = 0, the batch transfer is not performed. DER becomes 0.

Program example



```
WM0 = H20
BCOPY (WR100,   WM0,   8,   128)
```

[Program description]

Covers the communication data area set from WR100 to WR17F with the space code (H20) as the default value at the first scan after RUN. (Copies 1 byte data 128 blocks.)

Name	[Decode / Encode] Decode

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
DECO (d, s, n)	—	5	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
d : Bit/n : 1 to 3	38.71	41.56	—	—	—
d : Bit/n : 4 to16	38.21+2^(n-4)	41.56+2^(n-4)	—	—	—
d : Word/n : 1 to 3	21.83	22.76	—	—	—
d : Word/n : 4 to 16	21.83+2^(n-4)	22.76+2^(n-4)	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Top I/O of decoding destination			✓					✓	✓	✓				
s	Value to decode							✓	✓	✓	✓				✓
n	Number of bits to decode														✓

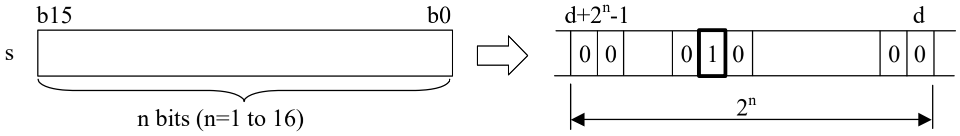
Remarks

Function

(1) When the I/O of the decoding destination is the bit area.

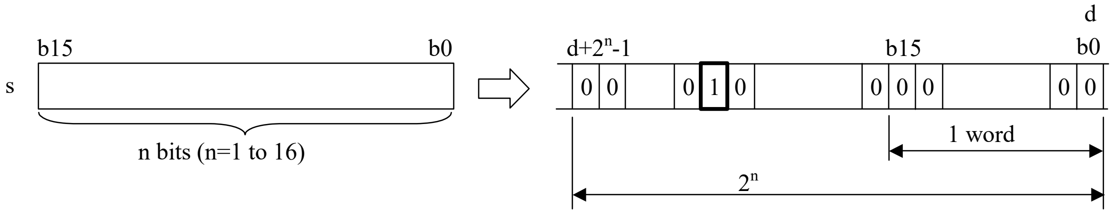
The lower n bits is decoded to 2^n and 1 is output to the decoded bit in the bit string from d to d+2^n-1.

(n = 1 to 16)



(2) When the I/O of the decoding destination is the word area.

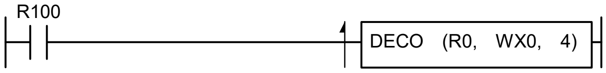
The lower n bits is decoded to 2^n and 1 is output to the decoded bit in the bit string from the 0th bit in d to d+2^n-1.



Cautionary notes

- Please used d+2^n-1 within the I/O range.
- When n is 0, the command is not performed. The content from d to d+2^n-1 holds the original value.
- Please specify n between 1 and 16.

Program example



[Program description]

If WX0 = HFFFF, RF, which is the 15 bit counting from R0 and the bit indicated with the value of the lower 4 bits in WX, is set to 1 at the rising of R100.

Name	[Decode / Encode] Encode

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
ENCO (d, s, n)	—	5	●	●	●	●	↕

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
s : Bit/n : 1～3	38.43+2^n	40.46+2^n	—	—	—
s : Bit/n : 4～16	74.93+2^(n-4)	76.96+2^(n-4)	—	—	—
s : Word/n : 1～3	24.43+2^n	25.86+2^n	—	—	—
s : Word/n : 4～16	30.37+2^(n-4)	32.06*2^(n-4)	—	—	—

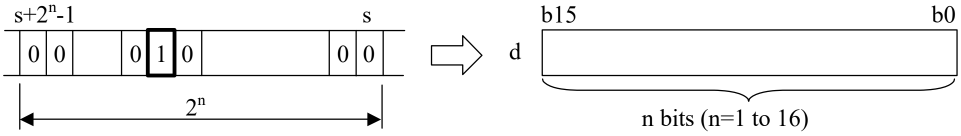
Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Top I/O of encoding destination							✓	✓					
s	Top I/O in bit string to encode			✓			✓	✓	✓	✓				
n	Number of bits to encode													✓

Remarks

Function

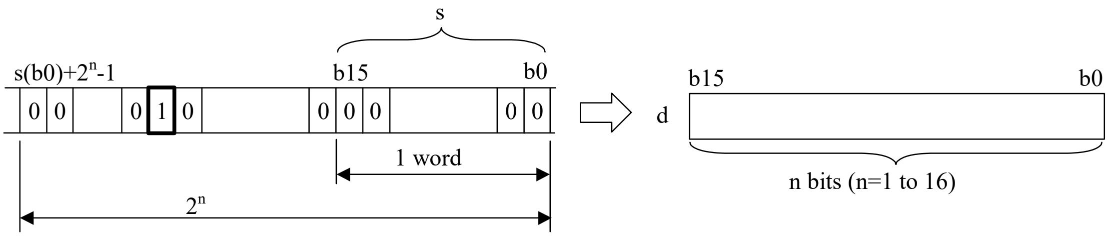
(1) When the encoding I/O is the bit area.

The bit position 2^n is encoded to n bits between s and s+2^n-1 and the result is output to d. (n = 1 to 16)



(2) When the encoding I/O is the word area.

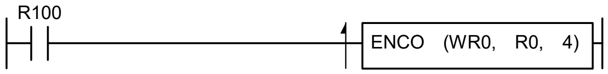
The bit position 2^n is encoded to n bits between 0 bit and 2^n-1 in s and the result is output to d. (n = 1 to 16)



Cautionary notes

- Please use $s+2^n-1$ within the I/O range.
- If there are several 1 between s and $s+2^n-1$ or 0 bit and 2^n-1 bits in s, the larger bit position is encoded.
- If n is 0, the command is not performed. d holds the original value.
- Please n between 1 and 16.
- If all bits between s and $s+2^n-1$ are 0, 0 is output to d and C(R7F0) is set to 1. In other cases, C(R7F0) is set to 0.

Program example



[Program description]

Detects the most significant bit which is set to 1 from the bit string which is from R0 to RF ($2^4-1 = 15$ bits) at the rising of R100, and sets the number of binaries of 4 bits to the word I/O of d.

Example) If the 7th bits and 6th bits between R0 and RF are set to 1, WR0 is set to H0007 .

Name	[Information storage / Display] Recording data (Initial setting)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
RECSET (s, n)	–	4	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	11.71	14	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	Top I/O in parameter table								✓					
n	Control number													✓

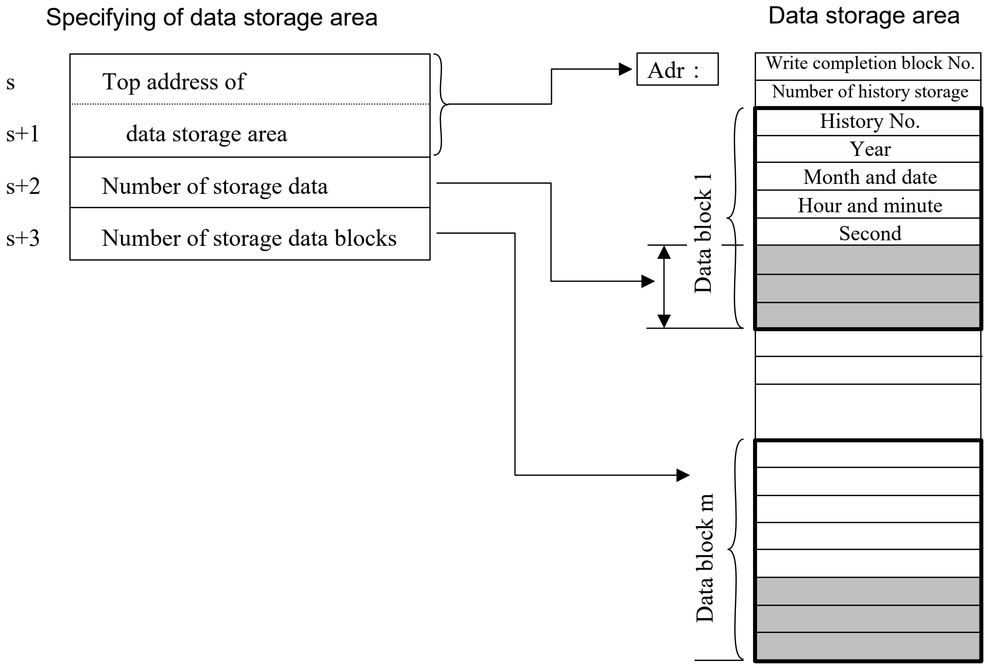
Remarks
n is from 1 to 32 (in decimal system). s parameters are occupied up to s+3.

Function

- The history number, the date and the time at executing, and data specified by users (hereafter called data block) can be stored in the specified internal output by combining RECEXE (s, n) with this command. This command performs an initial setting to memorize data block.
- The history storage can be controlled dividing to the maximum 32 sections, such as the history for event A and the history for event B.
- The user can specify the number of data to memorize. The number of words specified by the user and 5 words (the history number, the date and the time at a time of execution) are memorized at a time.
- If this command is executed, the write completion block number in the specified data storage area and the area for the number of history storage are cleared to 0.

Parameter

Specifies the top I/O No. of the parameter table to specify the data storage area by s.



The top address of the data storage area is specified by the I/O address coding command.

Cautionary notes

- Please specify the internal output used for s parameter and the data storage within the I/O No.
- If s parameter and the data storage area to the same control No. overlap, the operation is not performed because of DER = 1. But the overlap with the data storage area with different control number is not checked.
- If the number of storage data is 0, only the history number and the time data are memorized.
- The history storage cannot be executed until RECSET(s, n) to the control number of RECEXE(s, n) is executed.

Program example

Refer to description pages of ”RECEXE (s)”.

Name	[Information storage / Display] Recording data (Execution)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
RECEXE (s, n)	–	4	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	6.92	7.6	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	Data to memorize								✓					
n	Control number													✓

Remarks
s uses the number of data to memorize. n is from 1 to 32 (in decimal system).

Function

- The history number, the date and the time, and data specified by user are stored in the data storage area specified by RECSET(s, n), when this command is executed, and the write completion block number is updated. Also the number of history storage is added.
- The address to memorize the data block is computed on the system. And since the data storage area is the ring buffer, if the number of data blocks specified is stored, the next data block is overwritten from the top of data storage area. (The number of history storage is added.)
- n parameter relates this command and RECSET. This runs according to the initial setting of RECSET(s, n) which is executed by the same value as n parameter of this command.

Example) When there are RECSET (WR0, 1) and RECSET (WR4, 2)

If RECEXE (WM0, 1) is written, data in and after WM0 are memorized according to the initial setting by RECSET (WR0, 1).

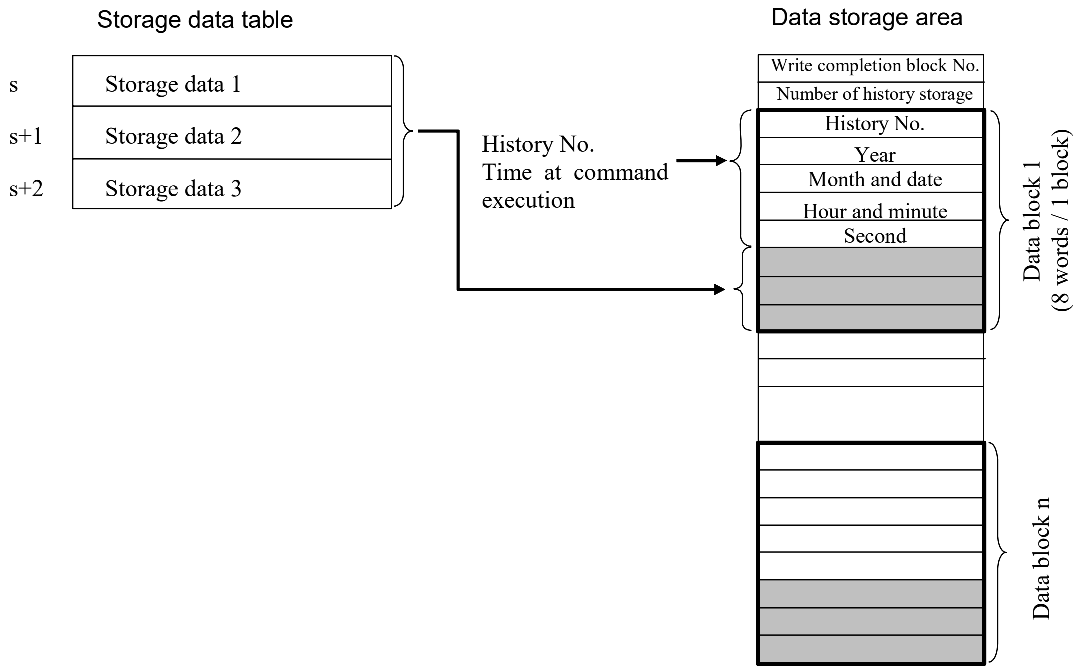
If RECEXE (WM0, 2) is written, data in after WM0 are memorized according to the initial setting by RECSET (WR4, 2).

Parameter

Specifies the top I/O No. in the table stored data to memorize by s.

The number of storage data specified by RECSET (s, n) determines the size of s. (If the number of data storage is 0, please allocate the internal output of dummy.)

Example) If the number of storage data is specified to 3 by RECSET(s, n), s can be used up to s+2.



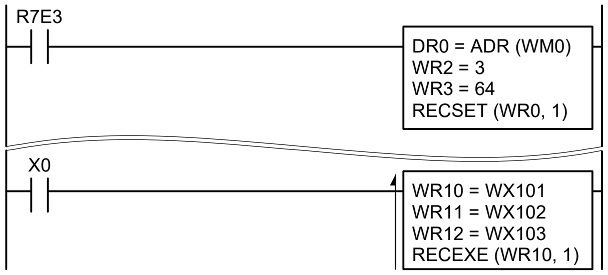
Cautionary notes

- Please specify the internal output used for s parameter within the I/O No.
- This command specifies the top I/O of the internal output in which data to store in the data storage area by s parameter is stored. Please pay attention since the purpose is different from s parameter of RECSET (s, n).
- Please program as RECSET (s, n) is executed before this command is executed. Even if this command is executed before RECSET (s, n) is executed, the operation is not performed because of DER = 1.
- Although the number of times to memorize the history is added even if data is overwritten from the top of the area because the data storage area was filled, the write completion block number is back to 1 when the data was overwritten.

Example) When the write block is 3.

The number of times to memorize the history	1	→	2	→	3	→	4	→	5	→	6	→	7	···
The write completion block No.	1	→	2	→	3	→	1	→	2	→	3	→	1	···

Program example



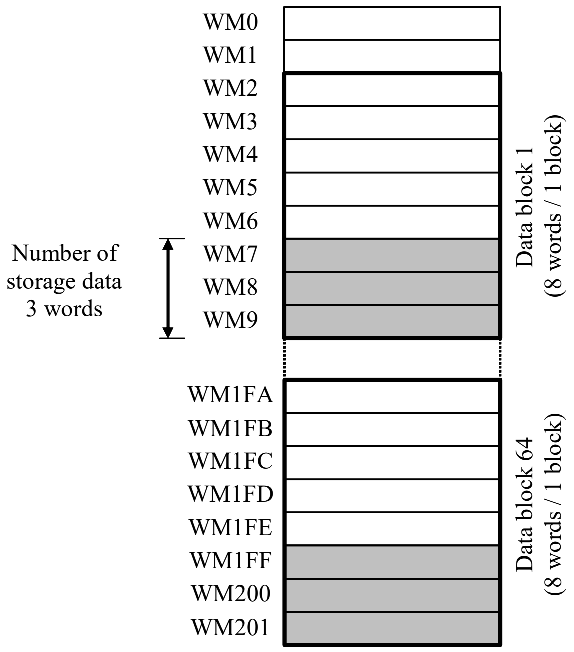
```
DR0 = ADR (WM0)
WR2 = 3
WR3 = 64
RECSET (WR0, 1)
```

```
WR10 = WX101
WR11 = WX102
WR12 = WX103
RECEXE (WR10, 1)
```

[Program description]

- Registers the data storage area of the control No.1 at the 1st scan after RUN.

  Such data storage area as the following is set in this program.



- Whenever X0 turns ON, the date and the time and values of WX101 to WX103 at the time are stored in the next data block.
- If data is stored until data block 64, the next data overwrites from the data block 1.

Name	[I/O Refresh] All points refresh

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
ALREF	—	2	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	30.82	34.88	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
—	(No argument)														

Remarks
Processing time for external input and output = Number of words × 1.4μsec

Function

Whole external input and output area are refreshed.

When this command is completed, the input (X) is updated to a state of executing. The output (Y) is the output with output value set by this command is executed.

Cautionary notes

If you want to perform the refresh partially, please use IOREF and SLREF.

Program example

[Program description]

Refreshes all I/O at the rising of R0.

PRN ➔ PRJ

This command is equivalent to FUN 80(s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 80 (s) into the program for MICRO-EHV is as follows.

FUN 80 (s) ➔ ALREF s parameter is not used.

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	[I/O Refresh] Refresh to specify classification

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
IOREF (s)	—	3	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	26.62	29.96	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	I/O classification to refresh									✓					✓

Remarks
Processing time for external input and output = Number of words × 1.4μsec

Function

- The classification specified by s parameter is refreshed.
- If the input (X) is specified, it is updated to a state that this command is executed.
- If the output (Y) is specified, the output value set by this command is executed is output.

Parameter

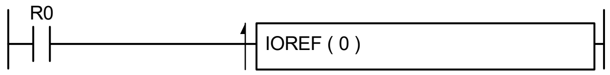
Specifies the I/O classification to refresh in the word internal output specified by s.

Parameter	Description	Details
s	Refresh I/O classification	H0000 ･･･ Input refresh H0001 ･･･ Output refresh

Cautionary notes

- Refreshes in slot units according to the I/O assignment.
- If the classification of input and output is specified of other than H0000, H0001, this command is not executed because of DER = 1.

Program example



[Program description]

・Refreshes the input area at the rising of R0.

PRN ➔ PRJ

This command is equivalent to FUN 81 (s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 81 (s) into the program for MICRO-EHV is as follows.

FUN 81 (s) ➔ IOREF (s)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	[I/O Refresh] Refresh to specify slot

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
SLREF (s)	–	3	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	8.66	9.6	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	Top I/O in table for refresh								✓					

Remarks
s is used the number of slots to refresh. Processing time for external input and output = Number of words × 1.4 μsec

Function

The unit specified by s parameter is refreshed.

Parameter

Specifies the I/O classification to refresh in the word internal output specified by s.

<table>
<tr><th>Parameter</th><th>Description</th><th>Details</th></tr>
<tr><td>s</td><td>Number of slots to refresh</td><td>The number of slots to refresh is specified.</td></tr>
<tr><td>s+1 or later</td><td>Slot position to refresh</td><td>The number of slots of the slot position to refresh is specified.
The constitution of words is shown below.
<table>
<tr><td>b15 b11</td><td>b12 b8</td><td>b7 b4</td><td>b3 b0</td></tr>
<tr><td>H0 fixed</td><td>H0 fixed</td><td>Unit No.</td><td>Slot No.</td></tr>
</table>
Unit number: 0 to 3
Slot number: 0, 1</td></tr>
</table>

Refreshing the basic unit

X0 to X39: Unit number 0, Slot number 0

Y100 to Y123: Unit number 0, slot number 1

Refresh expansion unit X48 / Y32

X*000 to X*039: Unit number *, Slot number 0

Y*100 to Y*123: Unit number *, Slot number 1

Refresh expansion unit X1Y1W (B1 / 1)

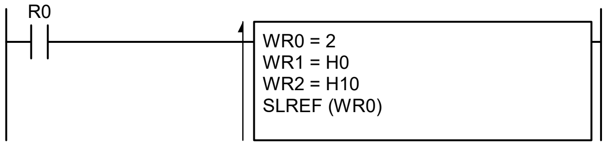
X*000 to X*015: Unit number *, Slot number 0

Y*016 to Y*031: Unit number *, Slot number 0

Cautionary notes

- Please specify the internal output used for s parameter within the I/O No.
- When specifying the inexistent position (the position without the I/O assignment) in the refresh slot position specifying, the slot is not processed because of DER = 1.

Program example



```
WR0 = 2
WR1 = H0
WR2 = H10
SLREF (WR0)
```

[Program description]

Refreshes the 0th slot in the basic unit (Unit 0) and the 0th slot in the 1st expansion unit (Unit 1) at the rising of R0. (I/O data of other units is updated at the scan END.)

PRN ➔ PRJ

This command is equivalent to FUN 82 (s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 82 (s) into the program for MICRO-EHV is as follows.

FUN 82 (s) ➔ SLREF (s)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above.

Name	PID control

■ What is PID control?

PID control is a generic control loop feedback mechanism using three separate constant parameters, P (proportional), I (integral) and D (derivative) as shown below.

P-control:	Output variable is proportional to the current error value (difference between the PV (process variable) and the SP (set point)). If the proportional gain is larger, PV will change faster. However, if Kp is too large, deviation is left and the PV could be hunting.
I-control:	Output variable is proportional to the accumulated error. This control is effective for residual steady-state error, which is not eliminated only by P-control.
D-control:	Output variable is proportional to the derivative of error. D-control is effective to reduce the magnitude of the overshoot produced by D-control or improve stability against external disturbances. However, if gain of D-control too large, the system could be unstable because MV reacts quickly for small derivation.

Manipulated variable of PID calculation is written in the formula 1.

$$MV(t) = K_P\left\{(SP-PV)+\frac{1}{T_I}\int(SP-PV)dt+T_D\frac{d}{dt}(SP-PV)\right\} \qquad \text{Formula 1}$$

This form is expressed in Laplace transform form.

$$\frac{MV(S)}{E(S)} = K_P\left(\underbrace{1}_{P}+\underbrace{\frac{1}{T_I\cdot S}}_{I}+\underbrace{T_D\cdot S}_{D}\right) \qquad \text{Formula 2}$$

MV(t), MV(S): Manipulated variable
(SP-PV), E(S): Error
K_P: Proportional gain
T_I: Integral time
T_D: Derivative time
S_P: Set point
P_V: Process variable

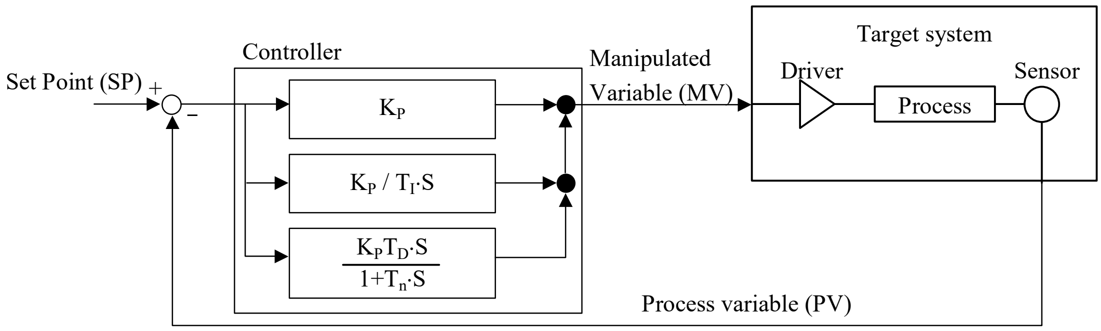
Since output for derivative term is a short pulse, which does not activate actual valves, practical PID calculation uses pseudo-derivative formula. If $T_D\cdot S$ is replaced by $\dfrac{T_D\cdot S}{1+\dfrac{T_D}{n}\cdot S}$, the formula 1 and 2 are expressed in the formula 3.

$$\frac{MV(S)}{E(S)} = K_P\left(1+\frac{1}{T_I\cdot S}+\frac{T_D\cdot S}{1+\dfrac{T_D}{n}\cdot S}\right) \qquad \text{Formula 3}$$

The parameter n is a derivative gain. Normally used value is about 10. By replacing $T_D/n = T_n$, this form is written in the formula 4. The parameter T_n is a derivative time delay.

$$\frac{MV(S)}{E(S)} = K_P\left(1+\frac{1}{T_I\cdot S}+\frac{T_D\cdot S}{1+T_n\cdot S}\right) \qquad \text{Formula 4}$$

This relation is expressed in the following block diagram.



(1) PID commands

PID command consists of three commands.

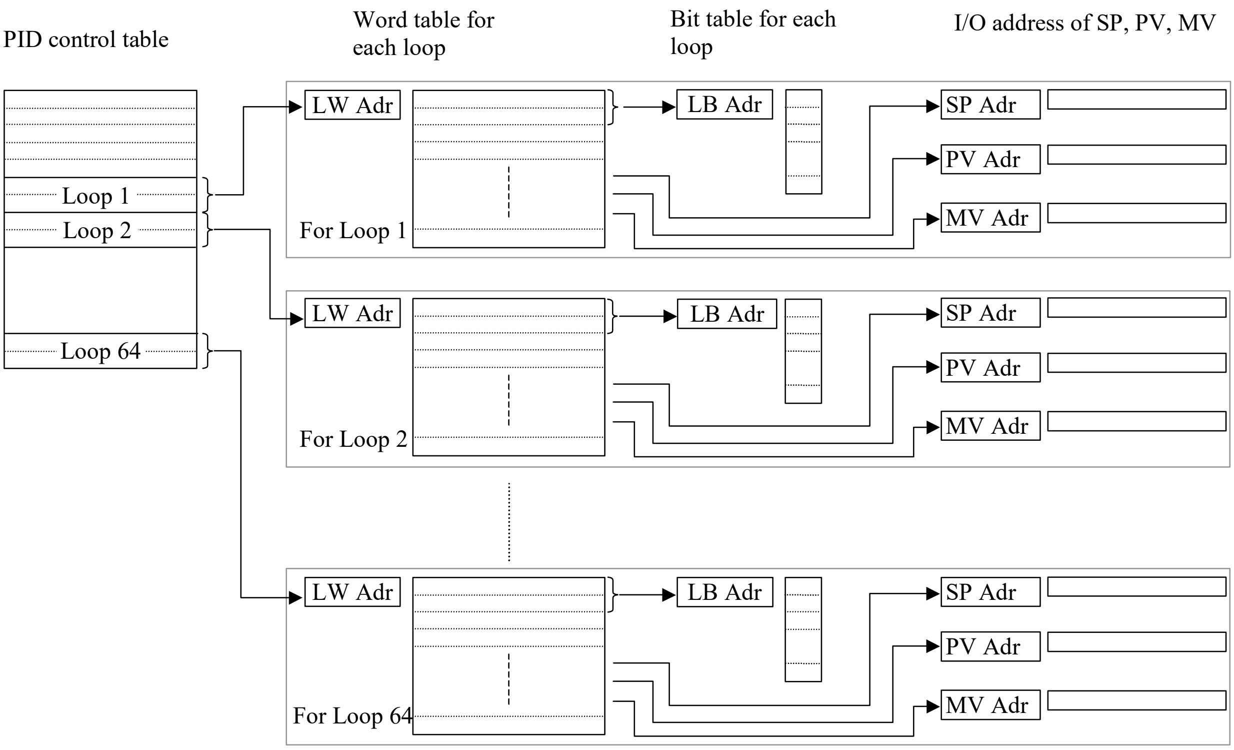
[1] PIDIT	Initializes the table used for PID command.
[2] PIDOP	Determines a loop for PID operation.
[3] PIDCL	Performs PID operation.

PIDOP and PIDCL are used together and written in cyclic scan. (Since sampling cycle of PID must be fixed time, put these commands in 20ms cyclic scan program.)

(2) Used data area for PID command

The PID command needs the following word table and bit table.

[1] PID control table	Table common for all PID loops. 5 + Number of loops × 2 words is required. (Example: Number of loops is 10; 25 words)
[2] Word table	52 words of parameter table is required for each loop.
[3] Bit table	16 bits data is required for each loop to control and monitor PID operation.



Name	[PID control] Initializing of PID calculation

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
PIDIT (s)	—	3	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	—	—	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, TDN, SS, WDT, MS, TMR, CU, RCU, CT	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	Top I/O in PID control table								✓					

Remarks

Function

All the area used for PID operation is initialized.

Parameter

s : Specifies the top I/O address in the PID control.

Cautionary notes

- If the PID control table is incomplete, this command does not work properly.
  (Error code is set to the area of "error code 0" in the PID control table.)
- If this command is executed again after the initializing completed properly (the area of "Indication of execution result of initializing" in the PID control table is H0001), this command fails with an error.

Name	[PID control] Execution control of PID calculation

Ladder format	Number of steps		Condition code				
PIDOP (s)	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
	–	3	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	–	–	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	Top I/O in PID control table								✓					

Remarks

Function

Loops of PID operation are determined.

(The loop is determined by taking in the PID execution flag and the PID constant change flag from the bit table area of each loop.)

Parameter

s : Specifies the top I/O address in the PID control table.

Cautionary notes

- Please use this command only once in 20ms cyclic scan.
- If the parameter s is specified of other than the top address in the PID control table, error occurs.
  (The error code is set to areas of “error code 0” and “error code 1” in the PID control table and this command is not executed.)

Name	[PID control] PID calculation

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
PIDCL (s)	—	3	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	—	—	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	PID loop word table								✓					

Remarks

Function

PID calculation is performed according to the sampling time set to the word table of each loop.

If PID calculation is executed, the PID calculation flag of the loop for calculating turns ON.

Parameter

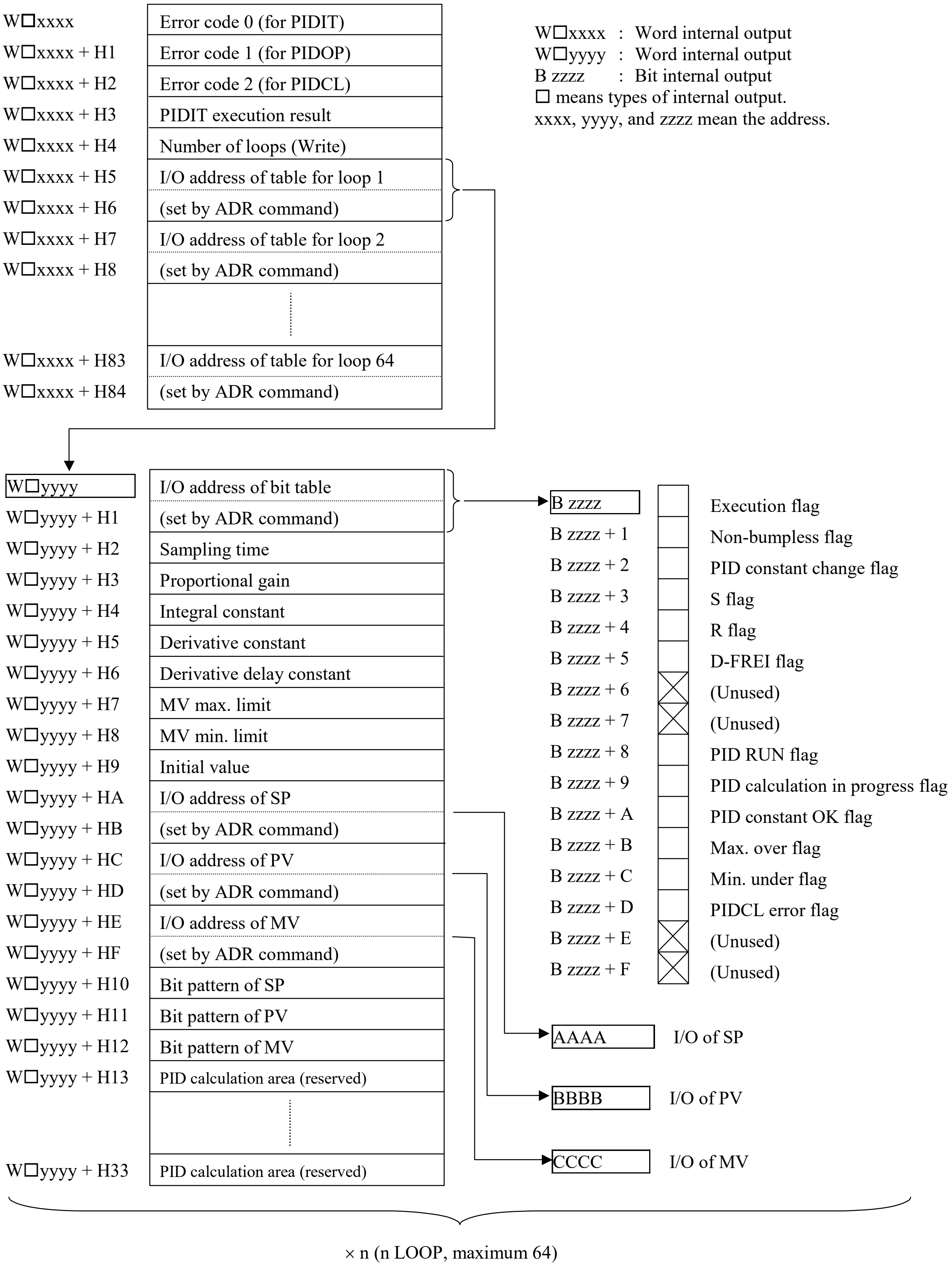
s : Specifies the top I/O address in the word table of each loop.

Cautionary notes

- Please use this command only once in 20ms cyclic scan.
- Please set all top addresses in the PID word table before this command is executed.
- This command checks the ranges of MV, bit pattern of SP/PV/MV for each loop. If error is found, the PIDCL execution flag in the bit table of each loop turns ON, and the error code is set to the area of "error code 2" in the PID control table.

  (PIDCL is executed even if error is found.)

■ Details of table for PID command



W☐xxxx : Word internal output
W☐yyyy : Word internal output
B zzzz : Bit internal output
☐ means types of internal output.
xxxx, yyyy, and zzzz mean the address.

W☐xxxx	Error code 0 (for PIDIT)
W☐xxxx + H1	Error code 1 (for PIDOP)
W☐xxxx + H2	Error code 2 (for PIDCL)
W☐xxxx + H3	PIDIT execution result
W☐xxxx + H4	Number of loops (Write)
W☐xxxx + H5	I/O address of table for loop 1
W☐xxxx + H6	(set by ADR command)
W☐xxxx + H7	I/O address of table for loop 2
W☐xxxx + H8	(set by ADR command)
	⋮
W☐xxxx + H83	I/O address of table for loop 64
W☐xxxx + H84	(set by ADR command)

W☐yyyy	I/O address of bit table
W☐yyyy + H1	(set by ADR command)
W☐yyyy + H2	Sampling time
W☐yyyy + H3	Proportional gain
W☐yyyy + H4	Integral constant
W☐yyyy + H5	Derivative constant
W☐yyyy + H6	Derivative delay constant
W☐yyyy + H7	MV max. limit
W☐yyyy + H8	MV min. limit
W☐yyyy + H9	Initial value
W☐yyyy + HA	I/O address of SP
W☐yyyy + HB	(set by ADR command)
W☐yyyy + HC	I/O address of PV
W☐yyyy + HD	(set by ADR command)
W☐yyyy + HE	I/O address of MV
W☐yyyy + HF	(set by ADR command)
W☐yyyy + H10	Bit pattern of SP
W☐yyyy + H11	Bit pattern of PV
W☐yyyy + H12	Bit pattern of MV
W☐yyyy + H13	PID calculation area (reserved)
	⋮
W☐yyyy + H33	PID calculation area (reserved)

B zzzz	☐	Execution flag
B zzzz + 1	☐	Non-bumpless flag
B zzzz + 2	☐	PID constant change flag
B zzzz + 3	☐	S flag
B zzzz + 4	☐	R flag
B zzzz + 5	☐	D-FREI flag
B zzzz + 6	☒	(Unused)
B zzzz + 7	☒	(Unused)
B zzzz + 8	☐	PID RUN flag
B zzzz + 9	☐	PID calculation in progress flag
B zzzz + A	☐	PID constant OK flag
B zzzz + B	☐	Max. over flag
B zzzz + C	☐	Min. under flag
B zzzz + D	☐	PIDCL error flag
B zzzz + E	☒	(Unused)
B zzzz + F	☒	(Unused)

AAAA	I/O of SP
BBBB	I/O of PV
CCCC	I/O of MV

× n (n LOOP, maximum 64)

(1) Composition of PID control table

The PID control table consists of [2], [3], [4], and [5]. Since the size of table depends on the number of loops [3], allocate carefully not to duplicate and exceed the maximum address of internal output. If exceeded, the error code H0004 is written to error code 0 [2].

Address	Description	Details	Remarks
xxxx	Error code 0 *1 [Read]	・Error code which occurred on PIDIT process and a part of PIDOP process is set. ・If there is no error, the last status is held.	[2]
xxxx + 1	Error code 1 *1 [Read]	・Error code which occurred on PIDOP process is set. ・If there is no error, the last status is held.	
xxxx + 2	Error code 2 *1 [Read]	・Error code which occurred on PIDCL process is set. ・If there is no error, the last status is held.	
xxxx + 3	PIDIT normal completion [Read]	・H0001 is set when PIDIT [Initialization of PID] is executed successfully. ・If there is error, it becomes H0000 and the error code is set in the error code 0	[5]
xxxx + 4	Number of loops *2 [Write]	・Set the number of loops from 1 to 64. ・If it is 0, PID process is not performed and H0002 is written to the error code 0. (PID process is not performed even if PIDOP and PIDCL have been programmed.)	[3]
xxxx + 5 xxxx + 6	Top address of word table for loop 1 *2 [Write]	52 words of internal output are used per one loop for PID parameters and calculation. If it exceeds the maximum address of the internal output, error code XX05 is written to the error code 0.	[4]
xxxx + 6 xxxx + 7	Top address of word table for loop 2 *2 [Write]	52 words of internal output are used per one loop for PID parameters and calculation. If it exceeds the maximum address of the internal output, error code XX05 is written to the error code 0.	
xxxx + 8 xxxx + 9	Top address of word table for loop 3 *2 [Write]	52 words of internal output are used per one loop for PID parameters and calculation. If it exceeds the maximum address of the internal output, error code XX05 is written to the error code 0.	
...	...	...	
xxxx + 83 xxxx + 84	Top address of word table for loop 64 *2 [Write]	52 words of internal output are used per one loop for PID parameters and calculation. If it exceeds the maximum address of the internal output, error code XX05 is written to the error code 0.	

*1 The error code is represented by 4 digits in hexadecimal system. Refer to the error code details for details.

*2 [Write] in the table means parameters are to be written by users. (Possible to read also.)

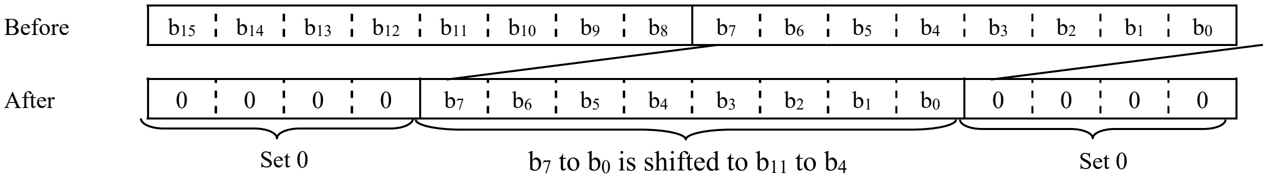
(2) Contents of Word table for each loop

Word table is specified in the area of [5] of (1) PID control table.

Address	Description	Specification	Details	Remarks
yyyy yyyy + 1	Top I/O address of bit table [Write]	Sets the top address in the bit table using the I/O address coding command.	16 bits are used per loop. Please set I/O address not to overlap the other bit tables.	[11]
yyyy + 2	Sampling time TZ [Write]	1 to 200 (× 20ms) Analog input/output module on basic or the expansion base. 4 to 200 (× 20ms) Analog input/output module on remote base.	Sets a multiple value of the number of loops [3]. (If it is same value as [3], the PID operation will be the fastest.)	[12]
yyyy + 3	Proportional gain KP [Write]	-1,000 to +1,000	Corresponds from -10.00 to +10.00.	[13]
yyyy + 4	Integral constant Ti/TZ [Write]	1 to 32,767	Sets the value of Ti/(Sampling time × Fixed cycle).	[14]
yyyy + 5	Derivative constant TD/TZ [Write]	1 to 32,767	Sets the value of TD/(Sampling time × Fixed cycle) is set.	[15]
yyyy + 6	Derivative delay constant Tn/TZ [Write]	1 to 32,767	Sets the value of Tn/(Sampling time × Periodic cycle).	[16]
yyyy + 7	MV max. limit value (UL) [Write]	-32,767 to 32,767	Set according to below. LL ≤ INIT ≤ UL	[17]
yyyy + 8	MV min. limit value (LL) [Write]	-32,767 to 32,767		[18]
yyyy + 9	Initial value (INIT) [Write]	-32,767 to 32,767		[19]
yyyy + A yyyy + B	I/O address of set point (SP) [Write]	Set the I/O address of set point (SP) by ADR command.		[20]
yyyy + C yyyy + D	I/O address of process variable (PV) [Write]	Set the I/O address of process variable (PV) by ADR command.		[21]
yyyy + E yyyy + F	I/O address of manipulated variable (MV) [Write]	Set the I/O address of manipulated variable (MV) by ADR command.		[22]
yyyy + 10	Bit pattern of set point (SP) [Write]	The set point is converted to 16-bit data according to this parameter.	Choose from 1 to 4 according to *1 in the next page.	[23]
yyyy + 11	Bit pattern of process variable (PV) [Write]	The process variable is converted to 16-bit data according to this parameter.		[24]
yyyy + 12	Bit pattern of manipulated variable (MV) [Write]	Calculated 16-bit data is converted to the manipulated variable according to this parameter.	Choose from 1 to 4 according to *2 in the next page.	[25]
yyyy + 13 to yyyy + 33	PID calculation area [Reserved]	Do not use this area because it is reserved for PIDIT, PIDOP and PIDCL.		[26]

*1 Bit pattern type for SP and PV

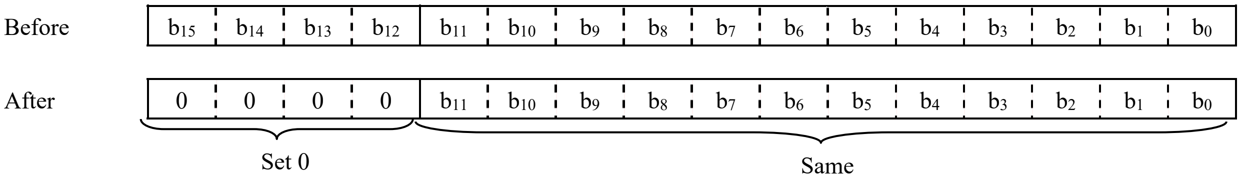
H0001: 8-bit ➔ 16-bit



Before	b_{15}	b_{14}	b_{13}	b_{12}	b_{11}	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
After	0	0	0	0	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	0	0	0	0

Set 0 — b_7 to b_0 is shifted to b_{11} to b_4 — Set 0

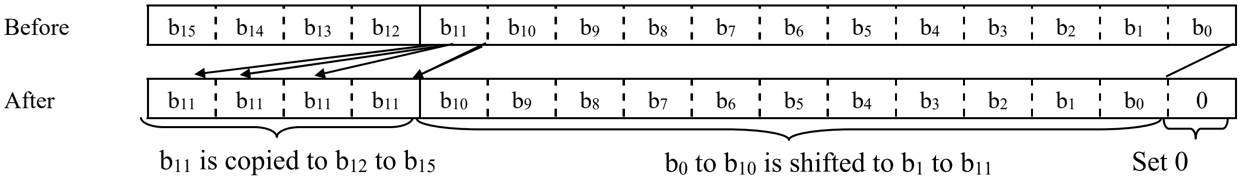
H0002: Unsigned 12-bit ➔ 16-bit



Before	b_{15}	b_{14}	b_{13}	b_{12}	b_{11}	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
After	0	0	0	0	b_{11}	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0

Set 0 — Same

H0003: Signed 12-bit ➔ Signed 16-bit



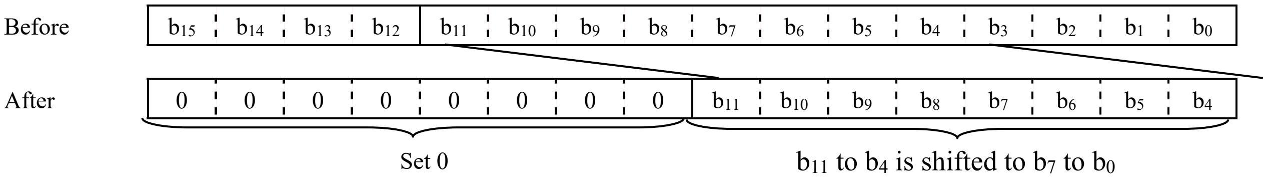
Before	b_{15}	b_{14}	b_{13}	b_{12}	b_{11}	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
After	b_{11}	b_{11}	b_{11}	b_{11}	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	0

b_{11} is copied to b_{12} to b_{15} — b_0 to b_{10} is shifted to b_1 to b_{11} — Set 0

H0004: No conversion

*2 Bit pattern type for MV

H0001: 16-bit ➔ 8-bit

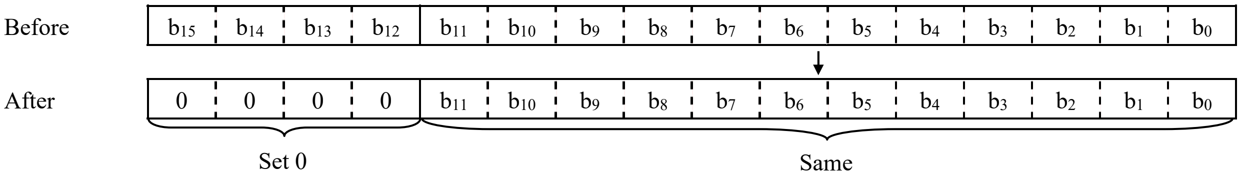


Before	b_{15}	b_{14}	b_{13}	b_{12}	b_{11}	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
After	0	0	0	0	0	0	0	0	b_{11}	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4

Set 0 — b_{11} to b_4 is shifted to b_7 to b_0

If data is from H0FFF to H7FFF, it is converted to H00FF.
If data is from H8000 to HFFFF, it is converted to H0000.

H0002: 16-bit ➔ 12-bit

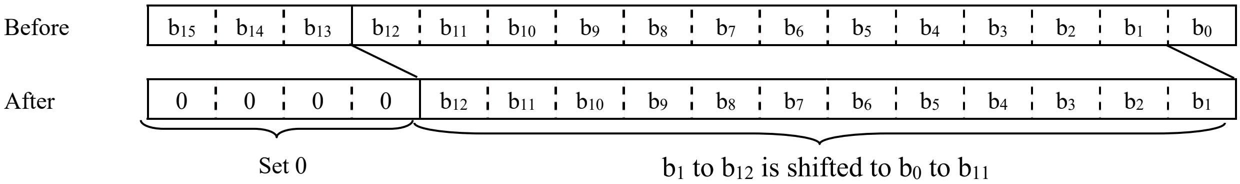


Before	b_{15}	b_{14}	b_{13}	b_{12}	b_{11}	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
After	0	0	0	0	b_{11}	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0

Set 0 — Same

If data is from H0FFF to H7FFF, it is converted to H00FF.
If data is from H8000 to HFFFF, it is converted to H0000.

H0003: Signed 16-bit ➔ Signed 12-bit



Before	b_{15}	b_{14}	b_{13}	b_{12}	b_{11}	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
After	0	0	0	0	b_{12}	b_{11}	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1

Set 0 — b_1 to b_{12} is shifted to b_0 to b_{11}

If data is from H0FFF to H7FFF, it is converted to H07FF.
If data is from H8000 to HFFFF, it is converted to H0800.

H0004: No conversion

Application

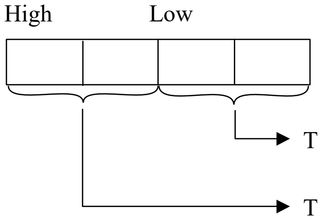
PIDCL (s)

(3) Contents of bit table

Address	PID control table	Details	Remarks
zzzz	Execution flag [Write]	·When this flag is activated, PID parameters are checked and PID data is initialized then PID calculation starts. While PID operation, PID RUN flag [58] is activated. If any error is found in PID parameters, PID calculation does not start. · When this flag is deactivated, PID calculation stops and MV becomes 0.	[50]
zzzz + 1	Non-bumpless flag [Write]	Non-bumpless means that internal PID calculation continues even S-flag or R-flag is activated. Consequently, when S or R-flag is deactivated, MV changes suddenly according to the calculation in the sleeping time. 0: Bumpless (PID halt, MV continues.) 1: Non-bumpless (PID continues, MV changes.).	[51]
zzzz + 2	PID constant change flag [Write]	· When this flag is activated, PID parameters are read and recalculated. This flag must be deactivated manually by user program. · If error is found in PID parameters (PID parameters OK = 0), the last parameters are used.	[52]
zzzz + 3	S-flag [Write]	When this flag is activated, MV is fixed as initial value. If the initial value is out of the limit values, it works as below. MV min. [18] > MV max. [17] ... No output MV min. [18] ≤ Initial [19] ≤ MV max. [17] ... MV = initial [19] MV min. [18] ≤ MV max. [17] ≤ Initial [19] ... MV = MV max. [17] Initial [19] ≤ MV min. [18] ≤ MV max. [17] ... MV = MV min. [18] S-flag has higher priority than R-flag.	[53]
zzzz + 4	R flag [Write]	When this flag is activated, MV is reset as 0.	[54]
zzzz + 5	D-FREI flag [Write]	0: Derivative calculation is disabled. (PI) 1: Derivative calculation is enabled. (PID)	[55]
zzzz + 6	Unused		
zzzz + 7	Unused		
zzzz + 8	PID RUN flag [Read]	When PID execution flag [50] is activated, parameters from [12] to [16] and [20] to [22] are checked and the result is set in this flag. 1: Correct 0: Wrong	[58]
zzzz + 9	PID calculation in progress flag [Read]	This bit is activated while PID calculation of corresponding loop. It is deactivated while other loops' calculation.	[59]
zzzz + A	PID constant OK flag [Read]	When PID execution flag [50] is activated, parameters from [12] to [16] are checked and the result is set in this flag. 1: Correct 0: Wrong	[60]
zzzz + B	Max. limit over flag [Read]	If MV is higher than MV max. [17], this flag is set.	[61]
zzzz + C	Min. limit under flag [Read]	If MV is lower than MV min. [18], this flag is set.	[62]
zzzz + D	PIDCL error flag [Read]	If error is found in the MV max. [17], the MV min. [18] and bit patterns from [23] to [25], this flag is set. The causes of error is set in the error code 2 [2]. PID calculation is performed even if error is found. If no error is found, this flag is reset. No data is stored in the error code 2 [2].	[63]

■ Error code list

Error code is represented by 4 digits in hexadecimal format.



The lower 2 digits represent causes of error.

The higher 2 digits represents the loop number.
If it is H00, this error is not related to loop numbers.
If it is from H01 to H40, check this loop number.

(1) Error code 0

Error code 0 is related to PIDIT process and a part of PIDOP process.

If there is no error, the last error code is held.

Error code	Description and cause	Measurement	Remarks
0001	Executed PIDIT again after PIDIT has already executed properly.	Do not execute PIDIT after executed properly.	"Normal completion [5] of PIDIT" holds the preceding value.
0002	The number of loops [3] is 0.	Set the number of loops [3] within a range from 1 to 64.	
0003	The number of loops [3] is 65 or more.	Set the number of loops [3] within a range from 1 to 64.	
0004	PID control table exceeds the maximum address of the internal output.	Do not exceed the maximum address of the internal output by changing the top in the PID control table or the number of loops [3].	The size of PID control table can be changed. If the number of loops [3] exceeds the end of the I/O, "PIDIT Normal End [5]" holds the preceding value.
xx05	Word table for loop xx exceeds the maximum address of internal output.	Set word table address [4] correctly.	Size of the word table is 52 words per loop.
xx06	Bit table for loop xx exceeds the maximum address of internal output.	Set bit table address [11] correctly.	Size of the bit table is 16 bits per loop.
xx07	MV max. [17] for loop xx is out of the range.	Set MV max. value [17] within the range from –32767 to 32767.	
xx08	MV min. [18] for loop xx is out of the range.	Set MV min. value [18] within the range from –32767 to 32767.	
xx09	Initial value [19] for loop xx is out of the range.	Set the initial value [19] within the range from –32767 to 32767.	
xx0A	The relation of size between Output upper limit value [17], Output lower limit value [18], and Initial value[19] for loop xx is wrong.	Set as MV min. [18] ≤ Initial value [19] ≤ MV max. [17]	
xx0B	Bit pattern of SP [23] for loop xx is outside the range.	Set the bit pattern of SP [23] within a range from 1 to 4.	
xx0C	Bit pattern of PV [24] for loop xx is out of the range.	Set the bit pattern of PV [24] within a range from 1 to 4.	
xx0D	Bit pattern of MV [25] for loop xx is outside the range.	Set the bit pattern of MV [25] within a range from 1 to 4.	
0020 (Note)	PIDOP is executed before PIDIT completed successfully.	Execute PIDOP after PIDIT is completed normally.	It is set to the error code 0 specified by s of PIDOP(s).
0021 (Note)	s-parameter of PIDOP (s) is different from that of PIDIT (s).	Set the same internal output as s of PIDIT(s) to s of PIDOP(s).	It is set to the error code 0 specified by s of PIDOP(s).

Note) The previous error code (0001 to xx0D) is overwritten by the error code 0020 and 0021.
Use PIDOP after confirming PIDIT is executed successfully.

(2) Error code 1

Error code which occurred on PIDOP process is set to the error code 1. If no error is detected, the last error is held.

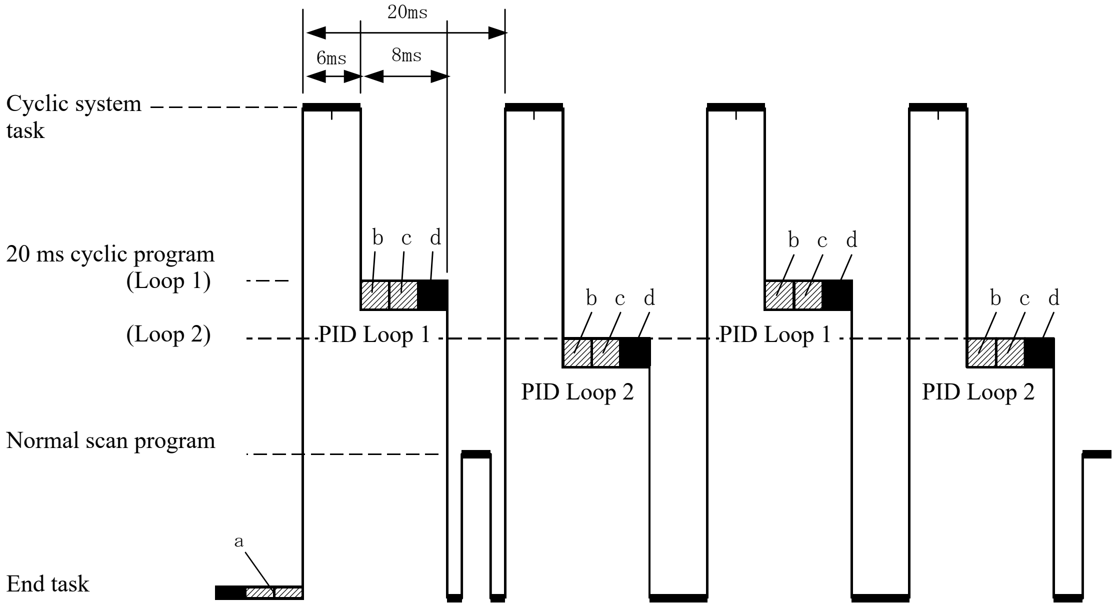
Error code	Description and cause	Measurement	Remarks
0020	PIDOP is executed before PIDIT completed successfully.	Execute PIDOP after PIDIT has been executed successfully.	It is set to the error code 0 specified by s of PIDOP(s).
0021	s of PIDOP(s) is different from s of [1]PIDIT(s) in PID control table.	Set the same internal output address as s of PIDIT(s) to s of PIDOP(s).	It is set to the error code 0 specified by s of PIDOP(s).
xx22	I/O address of SP [20] for loop xx is wrong.	Set I/O address of SP [20] by ADR command.	These errors are detected at the rising edge of the execution flag.
xx23	I/O address of PV [21] for loop xx is wrong.	Set I/O address of PV [21] by ADR command.	
xx24	I/O address of MV [22] for loop xx is wrong.	Set I/O address of MV [22] by ADR command.	
xx25	Sampling time [12] for loop xx is out of the range.	Set the sampling time [12] within the range from 1 to 200.	These errors are detected at the rising edge of the execution flag or PID constant change.
xx26	Sampling time [12] for loop xx is not multiples of the number of loops[3].	Set the sampling time [12] with multiples of the number of loops [3].	
xx27	Proportional gain [13] for loop xx is out of the range.	Set the proportional gain [13] within the range from −1000 to 1000.	
xx28	Integral constant [14] for loop xx is out of the range.	Set the integral constant [14] within the range from 1 to 32767.	
xx29	Derivative constant [15] for loop xx is out of the range.	Set the derivative constant [15] within the range from 1 to 32767.	
xx2A	Derivative delay constant [16] for loop xx is out of the range.	Set the derivative delay constant [16] within the range from 1 to 32767.	
xx30	The relation of size between MV min. [18] and MV max. [17] is wrong.	Set as MV min. [18] ≤ MV max. [17].	If S flag [53] is turned ON when PID RUN flag [58] is OFF, this error may occur.
xx31	I/O address of MV [22] for loop xx is wrong.	Set I/O address of MV [22] by ADR command.	If S flag or R flag is turned ON when PID RUN flag [58] is OFF, this error may occur.
xx32	Bit pattern of MV for loop xx is outside the range.	Set bit pattern of MV [25] within the range from 1 to 4.	

(3) Error code 2

Error code	Description and cause	Measurement	Remarks
0040			(Reserve)
xx41	Bit pattern of SP [23] for loop xx is out of the range.	Set bit pattern of SP [23] within the range from 1 to 4.	When the bit pattern is outside the range, the process is continued as “4. No conversion”.
xx42	Bit pattern of PV [24] for loop xx is out of the range.	Set bit pattern of PV [24] within the range from 1 to 4.	
xx43	Bit pattern of MV [25] for loop xx is out of the range.	Set bit pattern of MV [25] within the range from 1 to 4.	
xx44	The relation of MV min. [18] and MV max. [17] is wrong.	Set as MV min. [18] ≤ MV max [17].	When the relation is wrong, the process is continued but MV is not output.

■ Sequence of PID control

Example 1) 2 loops with TZ = 2 (x 20ms) for the both loops.



a: Execution of PIDIT
b: Execution of PIDOP
c: Execution of PIDCL
d: Other cyclic task

Figure 5.1 PID execution sequence (2 loops)

Example 2) 3 loops with
Loop 1: TZ = 3 (x20ms)
Loop 2: TZ = 6 (x20ms)
Loop 3: TZ = 12 (x20ms)

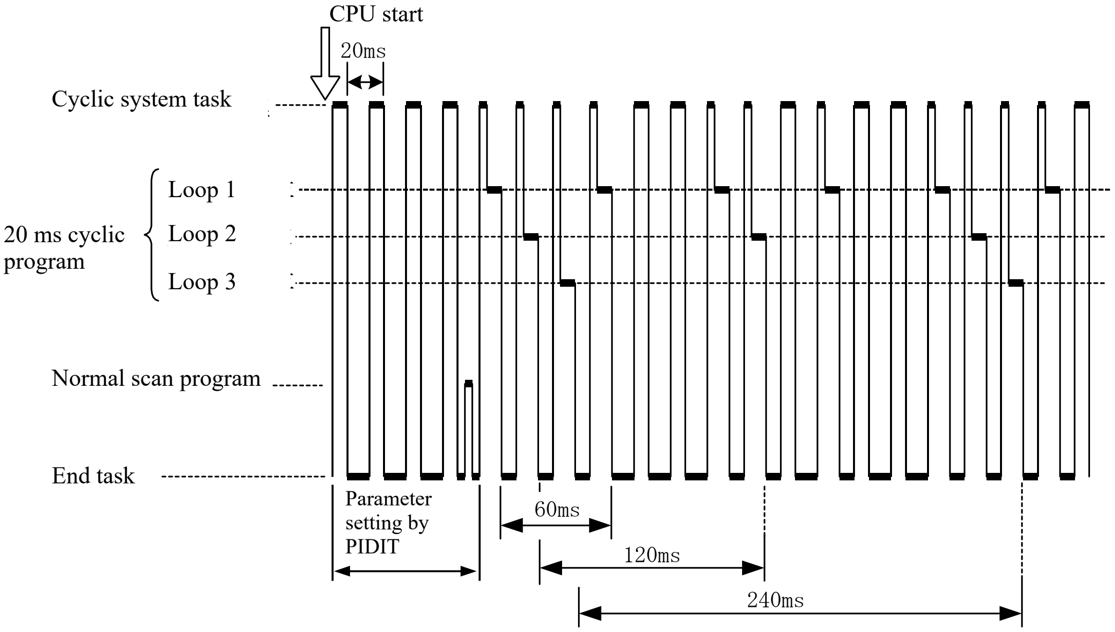


Figure 5.2 PID execution sequence (3 loops)

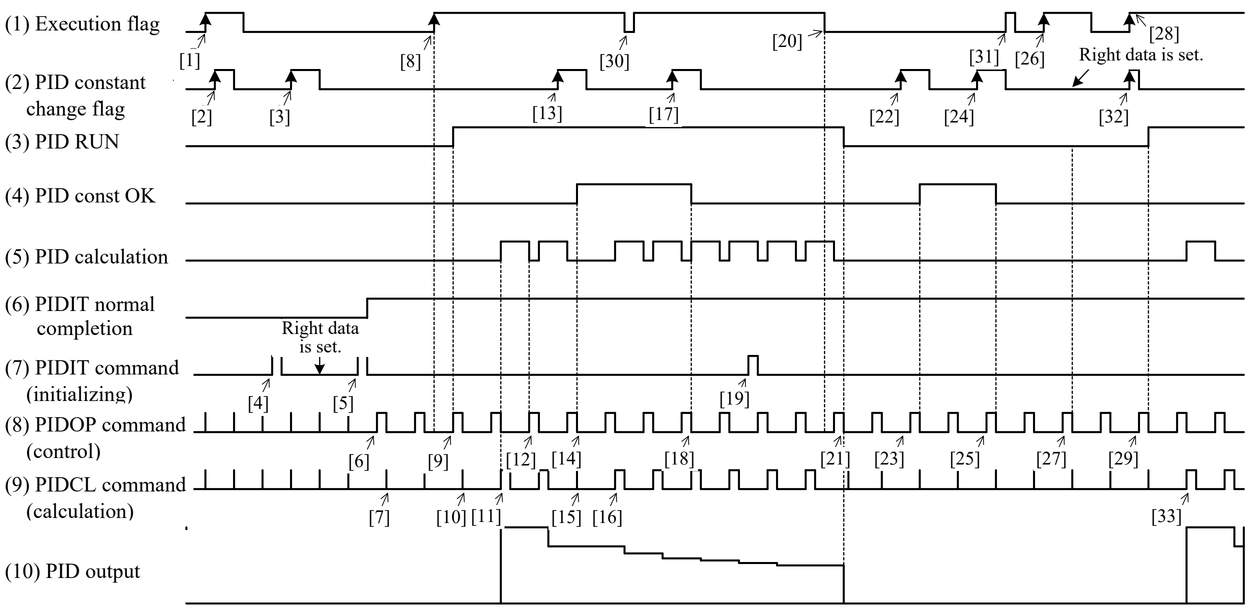
Caution

In Figure 5.1 and Figure 5.2, for the sake of clarity, the system interrupt processing every 10ms is expressed as two batches.

Time chart of PID Control

(a) Example 1

When PID execution flag is activated, the status of PID RUN flag, PID constant OK flag1, PID calculation flag, PIDIT, PIDOP and PIDCL are shown below.



Description of the time chart

[1] Execution flag and [2], [3] PID constant change flag are not counted since PIDIT has not been completed.

[4] If PID table is wrong, PIDOP is not exeuted.

[5] When PIDIT is completed successfully, [6] PIDOP command starts. [7] PIDCL does not work since execution flag is not activated.

[8] When PIDOP detects the rising edge of execution flag, PID parameters are checked and PID calculation starts with PID RUN flag set.

[10] PID calculation of PIDCL is not executed in the first cycle but calculated from [11].

[11] PID calculation flag is set before calculation and reset by [12] PIDOP.

[13] At the rising edge of PID constant change flag, [14] PIDOP checks PID constants, PID constant OK flag is set and PID constants are changed.

[15] PID calculation is not executed at [15] and executed from [16] with new PID constants.

[17] At the rising edge of PID constant change flag, [18] PIDOP checks PID constants, PID constant OK flag is reset since error is found in PID constants. PID constants are not updated.

[19] If PIDIT is executed while PID calculation, it is not effective.

[20] If [21] PIDOP detects falling edge of execution flag [20], PID RUN flag and output are reset.

[22] If [21] PIDOP detects rising edge of PID constant change flag [22], while execution flag is off, PID constants will be checked then PID constant OK flag is set since PID constants are OK.

[24] If [21] PIDOP detects rising edge of PID constant change flag [24] while execution flag is off, PID constants will be checked then PID constant OK flag is reset since PID constants are wrong.

[26] If [27] PIDOP detects rising edge of PID execution flag [26], PID RUN flag is reset since PID constants are wrong.

[28] If [29] PIDOP detects rising edge of PID execution flag [28] and PID constant change flag [32] both, PID constant change flag [32] is not counted.

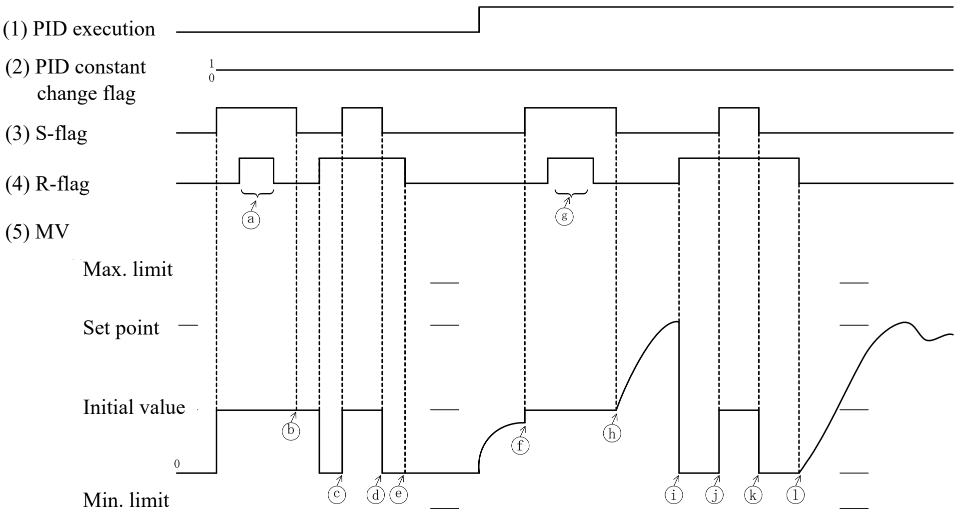
[29] PIDOP checks PID constants and if OK, then PID RUN flag is set. PID calculation starts from the next PIDCL [33].

[30][31] If execution flag is set or reset shorter timing than 20ms, it is not effective.

(b) Example 2

Time chart of S-flag and R-flag is shown below. (bumpless)

S-flag .... MV is set as the initial value.
R-flag.. MV is reset as 0.



ⓐ ⓖ MV is initial value because S-flag has higher priority than R-flag.

ⓑ ⓔ MV is hold because execution flag is off.

ⓒ ⓙ MV is initial value because S-flag has higher priority than R-flag.

ⓓ ⓚ When R-flag is on and S-flag is off, MV is reset as 0.

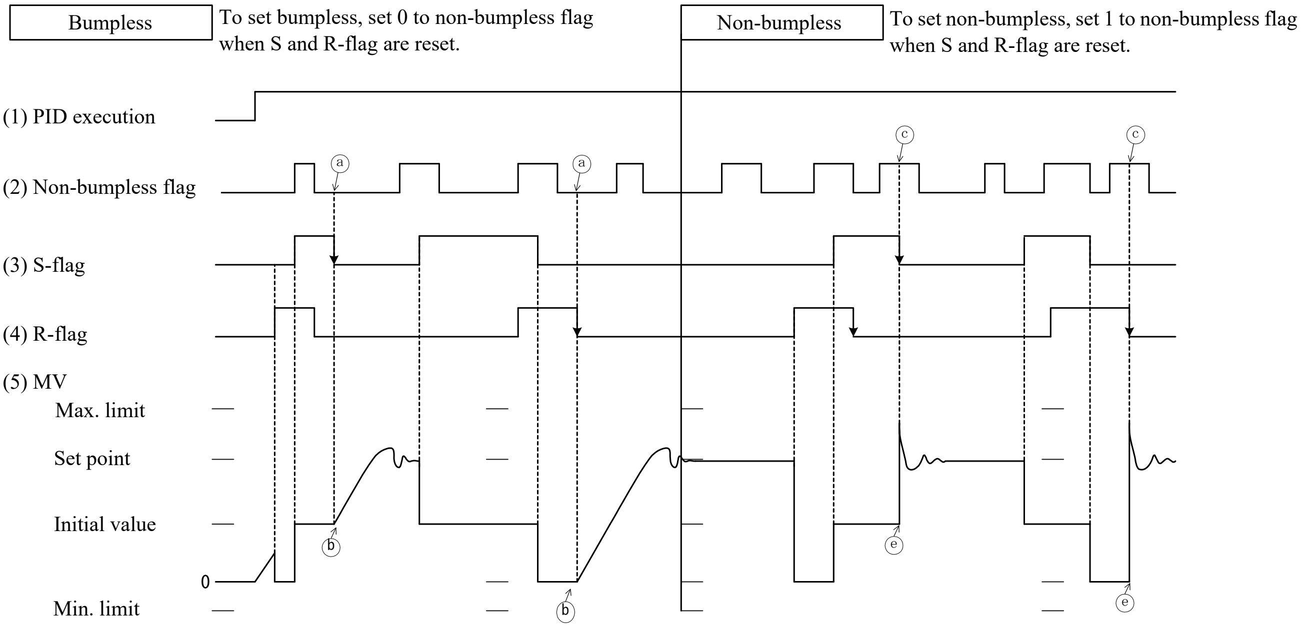
ⓕ MV is set as initial value.

ⓗ ⓛ Since execution flag is on with bumpless, MV changes continuously toward SP.

ⓘ MV is reset as 0.

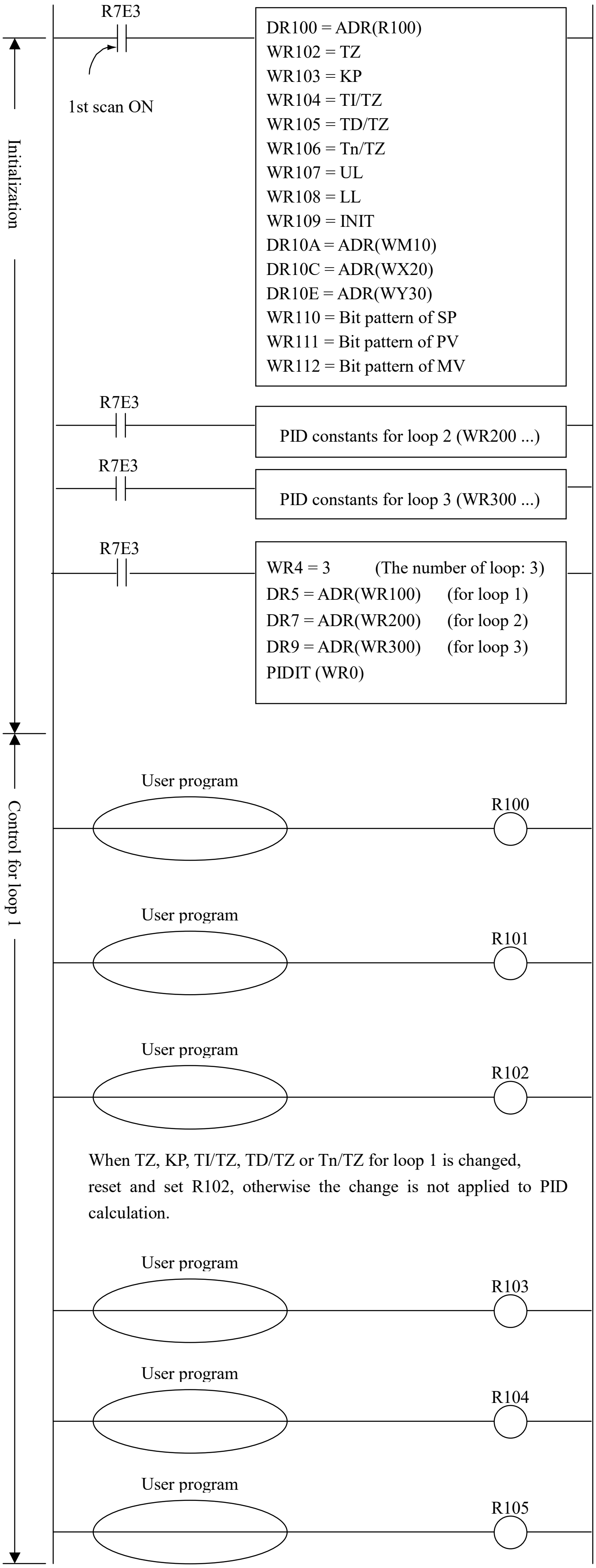
(c) Example 3

Bumpless and non-bumpless



ⓑ When S and R-flag are reset, MV changes continuously toward SP.

ⓔ When S and R-flag are reset, MV changes suddenly toward SP.



```
DR100 = ADR(R100)
WR102 = TZ
WR103 = KP
WR104 = TI/TZ
WR105 = TD/TZ
WR106 = Tn/TZ
WR107 = UL
WR108 = LL
WR109 = INIT
DR10A = ADR(WM10)
DR10C = ADR(WX20)
DR10E = ADR(WY30)
WR110 = Bit pattern of SP
WR111 = Bit pattern of PV
WR112 = Bit pattern of MV
```

PID constants for loop 1

```
PID constants for loop 2 (WR200 ...)
```

PID constants for loop 2

```
PID constants for loop 3 (WR300 ...)
```

PID constants for loop 3

```
WR4 = 3          (The number of loop: 3)
DR5 = ADR(WR100)     (for loop 1)
DR7 = ADR(WR200)     (for loop 2)
DR9 = ADR(WR300)     (for loop 3)
PIDIT (WR0)
```

Table for loop configuration

- PID constant for loop 1 to 3 are checked.
- Set all constants before PIDIT executed.

Execution flag
1: Start PID calculation for loop 1
0: Stop PID calculation for loop 1
(Output value will be 0.)

Non-bumpless flag
1: Non-bumpless
0: Bumpless

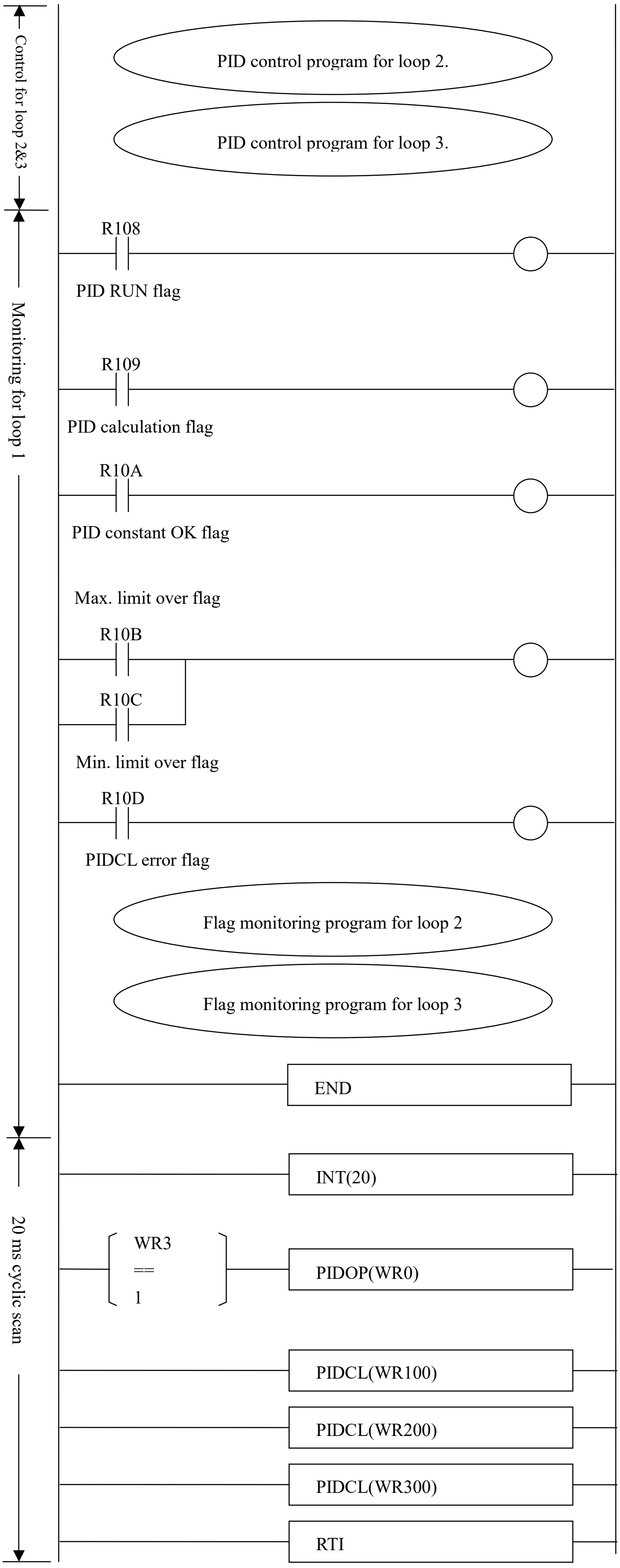
PID constant change flag
At rising edge, new PID constants are applied.

When TZ, KP, TI/TZ, TD/TZ or Tn/TZ for loop 1 is changed, reset and set R102, otherwise the change is not applied to PID calculation.

S-flag
1: MV becomes [19] Initial value.
S-flag has higher priority than R-flag.

R-flag
1: MV becomes 0.

D-FREI flag
1: Normal PID calculation
0: PI calculation (Without D control.)



PID constants are checked at the rising edge of PID execution flag (R100) and the result is set in this PID RUN flag (R108).
1: OK 0: Error

When loop 1 is calculated, this flag is set.

At rising edge of PID constant change flag (R120), new PID constants are checked and the result is set in this flag.
1: PID constants OK
0: PID constants wrong

When MV is out of the range, these flags are set.

If PIDCL command is not executed successfully, this error flag is set. Error code is set in [2] error code 2 area.

WR0: Error code 0
WR1: Error code 1
WR2: Error code 2

20ms cyclic scan program

If PIDIT command is executed successfully (WR3=1), PIDOP(WR0) is executed.
If error is detected in PIDIT, error information is stored in error code 0.

PID calculation for loop 1

PID calculation for loop 2

PID calculation for loop 3

End of cyclic scan.
Return to normal scan program.

Name	[FIFO] Initial						
Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
FIFIT (p, n)	–	4	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	17.4	19.56	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
p	Top I/O of FIFO								✓					
n	Size of FIFO													✓

Remarks
n is from 0 to 256.

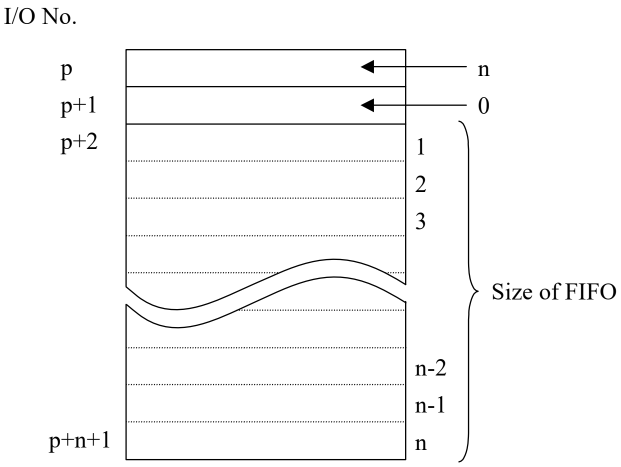
Function

- FIFO stands for First-In First-Out. Data is stored in the buffer and data which comes in first is taken out first, then the second is taken out at the second.

  This command initializes FIFO.
- The top I/O No. p and the size of FIFO n are specified.

  Case of 0 ≤ n ≤ 256 : set n to p.

  Case of 257 ≤ n : set 256 to p.
- p+1 is set to the initial set value 0 as the number of uses for FIFO.
- FIFO sets n+2 words from p to p+n+1.



Cautionary notes

- Please use p+n+1 within the I/O range. If exceeded, DER = 1 and [[Maximum value in the range (the end)]-[p-1]] is set to p.
- Please set n between 0 and 256. If n > 256, DER = 1 and it is set to 256.

Name	[FIFO] Write

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
FIFWR (p, s)	–	4	↕	●	●	●	●

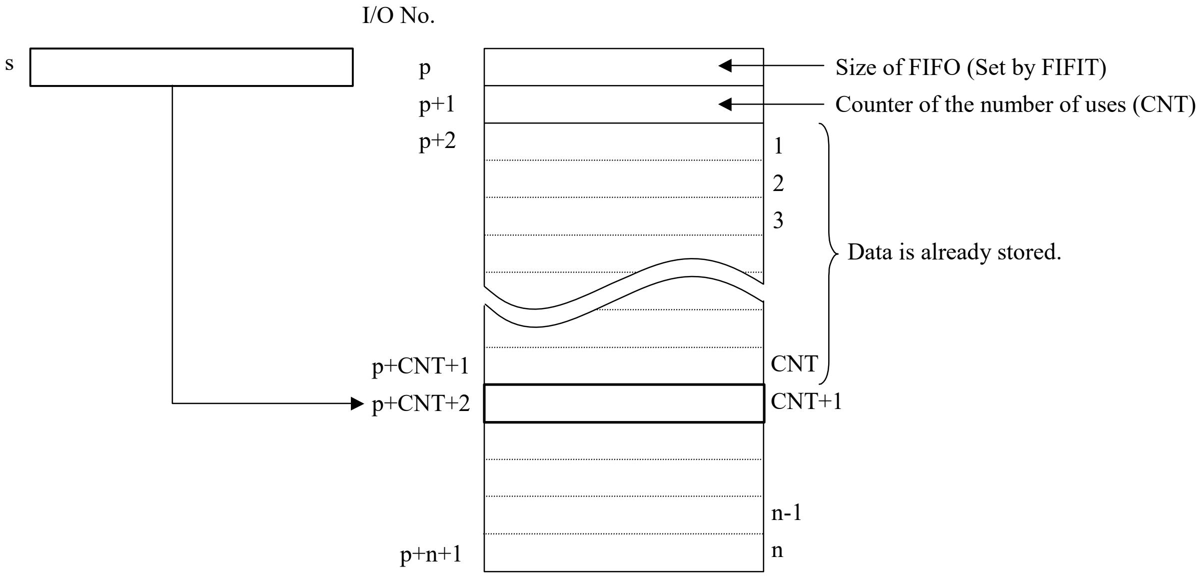
Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	9.64	17.32	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
p	Top I/O of FIFO									✓					
s	Content to write to FIFO							✓	✓	✓	✓				✓

Remarks
n is from 0 to 256.

Function

- Data is written to FIFO buffer of the top I/O No. p.
- Case of Counter of the number of uses (CNT) < Size n: the content of s is written to p+CNT+2, and 1 is added to the counter of the number of uses.
- Case of Counter of the number of uses (CNT) ≥ Size n: DER is 1 and it is not written.



Cautionary notes

- Please used p+n+1 within the I/O range. If exceeded, DER = 1 and it is not written.

Name	[FIFO] Read

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
FIFRD (p, d)	–	4	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	9	16.84	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
p	Top I/O of FIFO								✓					
d	I/O to store the read data							✓	✓	✓				

Remarks
n is from 0 to 256.

Function

- Data is read to FIFO buffer of the top I/O No. p.

  Case of 1 ≤ Counter of the number of uses (CNT) < Size n:

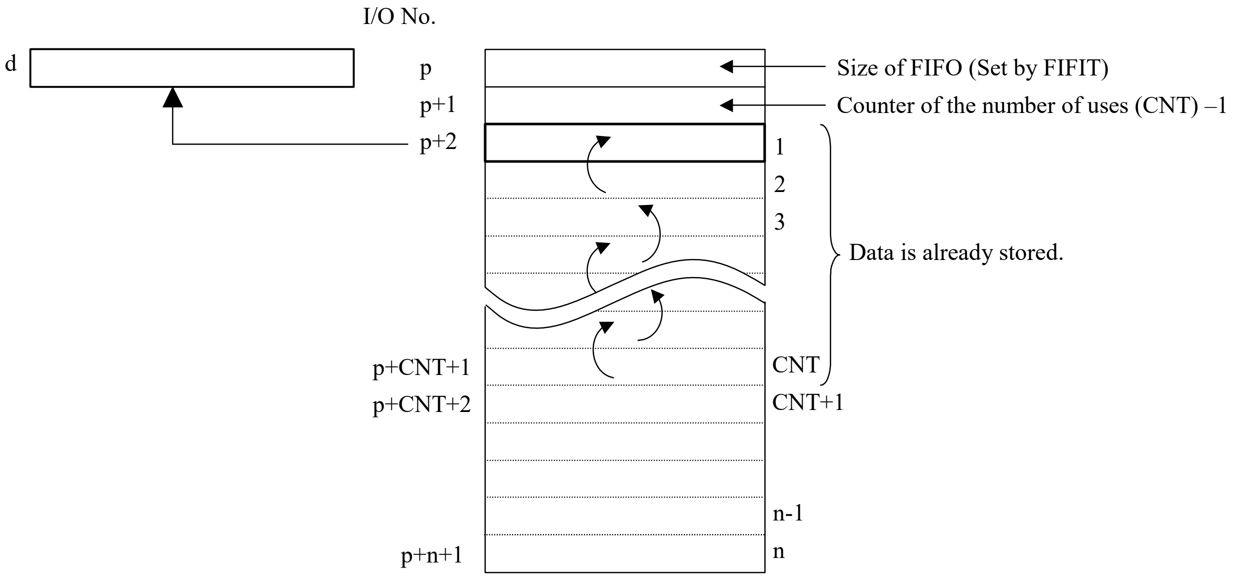
  the content of p+2 is read and the result is stored in d.

  the content from p+3 to p+CNT+2 is move to the preceding I/O respectively.

  1 is subtracted from the content of CNT.

  Case of Counter of the number of uses (CNT) > Size n, or CNT = 0:

  DER is 1 and it is not read.

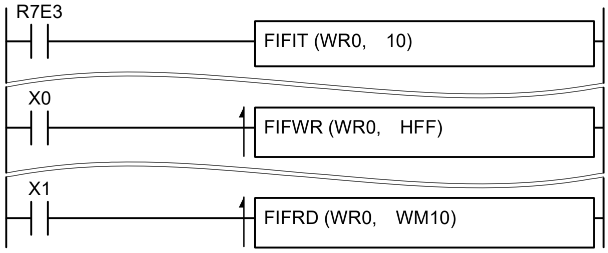


- If data is read, p+CNT+2 stores 0.

Cautionary notes

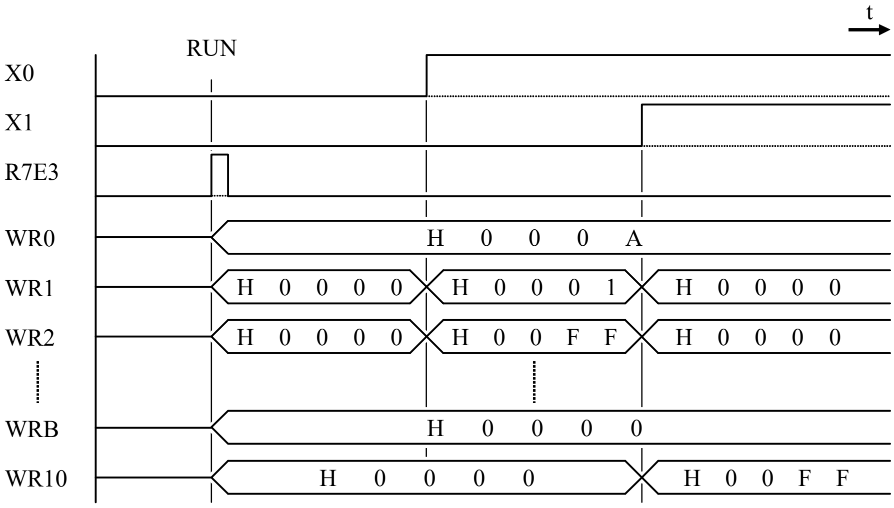
- Please use p+n+1 within the I/O range. If exceeded, DER = 1 and it is not read.

Program example



[Program description]

- Sets the FIFO buffer from WR2 to WRB at the 1st scan after RUN
- Stores HFF at the rising of X0.
- Reads HFF to WM10 at the rising of X1.



Name	[Communication support] Creation of check code

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
CCCL (s)	–	3	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	211+2.4*n	246.4+2.6*n	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	Top I/O in parameter table									✓					

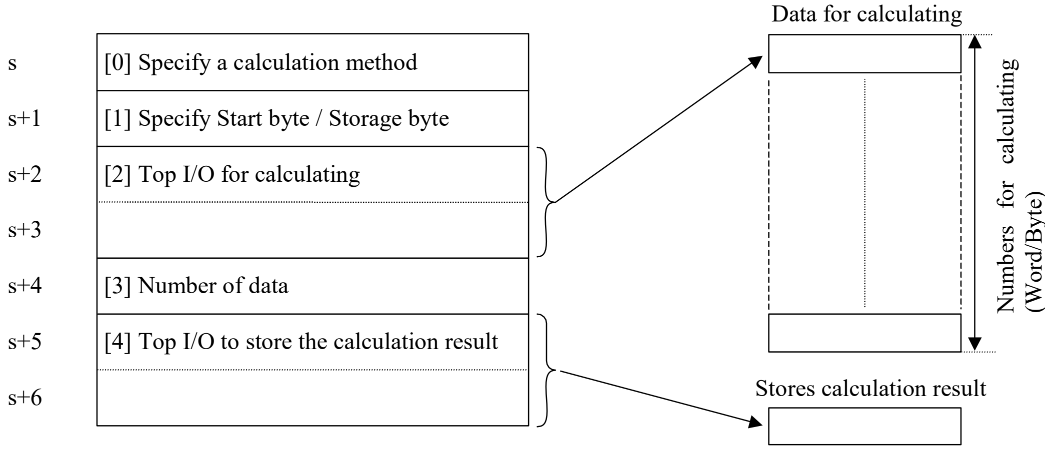
Remarks
s parameters are occupied up to s+6. n is the number of the data for calculating.

Function

In a general-purpose communication by TRNS 0 and the like, the check code to add to the data frame is created.

Parameter

- Uses 7 words from the word address specified by s.



s	[0] Specify a calculation method
s+1	[1] Specify Start byte / Storage byte
s+2	[2] Top I/O for calculating
s+3	
s+4	[3] Number of data
s+5	[4] Top I/O to store the calculation result
s+6	

Please set the address of WR and WM to [s+2, s+3] and [s+5, s+6] using the I/O address coding command.

[s+0] Specify a calculation method:

The calculation method for the check code can be specified from the following 7 kinds.

Set value	Calculation expression	Result of calculation	
H0000	(B1)+(B2)+ ··· +(Bn)	Byte	
H0001	(B1)+(B2)+ ··· +(Bn)	Word	Stores a found value in the order, upper part then lower part
H0002	(B1)+(B2)+ ··· +(Bn)	Word	Stores a found value in the order, lower part then upper part
H0003	(B1)+(B2)+ ··· +(Bn)	Byte	Stores a found value in word (in the order, upper then lower) after converted it to ASCII
H0004	(B1)+(B2)+ ··· +(Bn)	Byte	Stores a found value in word (in the order, lower then upper) after converted it to ASCII
H0005	(W1)+(W2)+ ··· +(Wn)	Word	Stores a found value in the order, upper part then lower part
H0006	(W1)+(W2)+ ··· +(Wn)	Word	Stores a found value in the order, lower part then upper part
H0010	{(B1)×or(B2)}×or···×or(Bn)	Byte	
H0011	{(B1)×or(B2)}×or···×or(Bn)	Byte	Stores a found value in word (in the order, upper then lower) after converted it to ASCII
H0012	{(B1)×or(B2)}×or···×or(Bn)	Byte	Stores a found value in word (in the order, lower then upper) after converted it to ASCII
H0013	{(W1)×or(W2)}×or···×or(Wn)	Word	Stores a found value in the order, upper part then lower part
H0014	{(W1)×or(W2)}×or···×or(Wn)	Word	Stores a found value in the order, upper part then lower part
H0020	LRC	Byte	
H0021	CRC16	Word	
Others	DATA Error (DER ON)		

[s+1] Specify Start byte /Storage byte:

Only when the check code is calculated in the byte units, the start byte can be specified of either the upper byte or the power byte. Also the calculation result storage byte can be specified of the upper byte or the lower byte.

<Upper byte>
Specify the start byte for calculation
H00xx: Start calculation from the upper byte
H01xx: Start calculation from the lower byte
Others: DATA Error (DER ON)

<Lower byte>
Specify the byte to store result of calculation
Hxx00: Sets the calculation result to the upper byte
Hxx01: Sets the calculation result to the lower byte*
Others: DATA Error (DER ON)
* When value to store is word, the lower byte is stored in the upper byte of next word.

Set value: H00xx

B	B1	B2
	B3	B4
	···	

W	W1
	W2
	···

Set value: H01xx

B	—	B1
	B2	B3
	···	

W	—	W1_u
	W1_l	W2_u
	W2_l	···

Set value: Hxx00

B	▨(1)▨	—
W	▨(1)▨	

Set value: Hxx01

B	—	▨(1)▨
W	—	▨(1)▨
	▨(1)▨	—

—: Data stored until then
(1): Store after overwriting the calculation result

[s+2, s+3] Top I/O No. to be calculated:

Please set the addresses of WR and WM using the I/O address coding command.

[s+4] Number of data:

Case of byte setting ··· Sets the number of bytes of data. (H0001～HFFFF)

Case of word setting ··· Sets the number of words of data. (H0001～HFFFF)

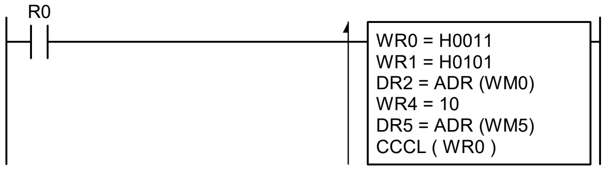
[s+5, s+6] I/O No. to store calculation result:

Please set the addresses of WR and WM using the I/O address coding command.

Cautionary notes

- Please specify the internal output used for s parameter table and the area for calculating within the I/O No.
- The check code for calculating is only the internal output of word. If other than the internal output of word is specified, the command is not executed because of DER = 1.
- Please pay attention so that the area for calculating does not overlap with s parameter. If overlapped, the command is not executed because of DER = 1.
- If the I/O other than usable I/O is specified to the area to store the calculation result, the command is not executed because of DER = 1.
- If the area to store the calculation result overlaps with s parameter table, the command is executed because of DER = 1.

Program example



```
WR0 = H0011
WR1 = H0101
DR2 = ADR (WM0)
WR4 = 10
DR5 = ADR (WM5)
CCCL ( WR0 )
```

[Program description]

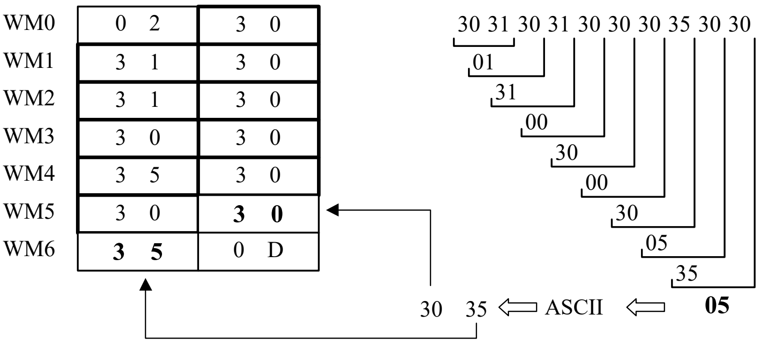
- Executes the CCCL command by setting the parameter for check code calculation at the rising edge of R0.
- Constitution of sending frame The check code is converted into ASCII after calculating XOR at every byte.

STX	Data	C.C.	CR
(02)	(30313031303030353030)	(?)	(0D)

- When the sending data area is the data composition as follows, suppose that R0 was turned ON.

WM0	0 2	3 0
WM1	3 1	3 0
WM2	3 1	3 0
WM3	3 0	3 0
WM4	3 5	3 0
WM5	3 0	? ?
WM6	? ?	0 D

- If the sample program is executed, the results is as follows.



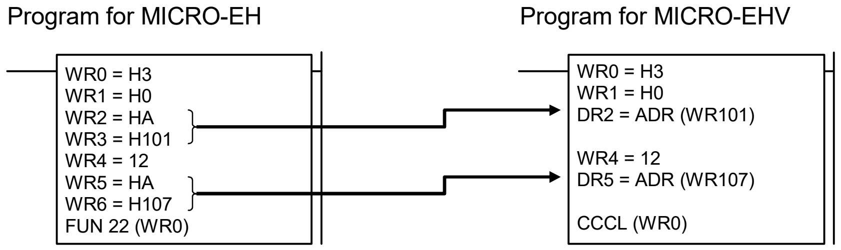
PRN ➔ PRJ

This command is equivalent to FUN 22 (s) in the program (PRN file) of MICRO-EH.

How to convert the program which has used FUN 22 (s) into the program for MICRO-EHV is as follows.

FUN 22 (s) ➔ CCCL (s)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above. However you need to modify a part of s parameter.



Program for MICRO-EH

```
WR0 = H3
WR1 = H0
WR2 = HA
WR3 = H101
WR4 = 12
WR5 = HA
WR6 = H107
FUN 22 (WR0)
```

Program for MICRO-EHV

```
WR0 = H3
WR1 = H0
DR2 = ADR (WR101)

WR4 = 12
DR5 = ADR (WR107)

CCCL (WR0)
```

Name	[Communication support] Collation of check code

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
CCCMP (d, s)	—	4	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	251.6+2.44*n	290.4+2.7*n	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	I/O to store collation result									✓					
s	Top I/O in parameter table									✓					

Remarks
s parameters are occupied up to s+6. d parameters are occupied up to d+2. n is the number of the data for calculating.

Function

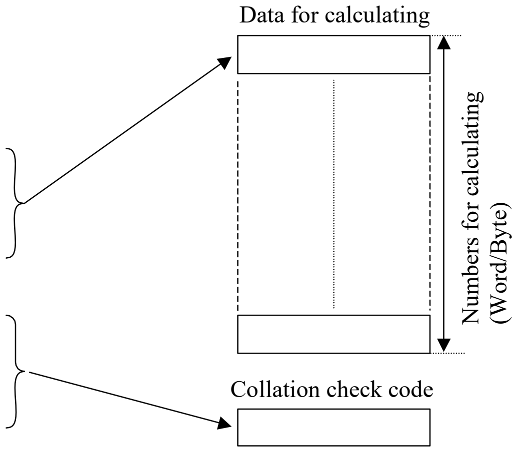
In a general-purpose communication by TRNS 0 and the like, a check code calculated from the data frame is collated with a check code added to the data frame.

Parameter

d : Stores the calculation result. Uses 3 words from the specified word I/O.

s : Table for the parameter to calculate the check code. Uses 7 words from the specified word I/O.

s	[0] Specify a calculation method
s+1	[1] Specify Start byte / Storage byte
s+2	[2] Top I/O for calculating
s+3	
s+4	[3] Number of data
s+5	[4] I/O to store collation check code
s+6	



d	[0] Collation result
d+1	[1] Calculation result
d+2	[2] Calculation result

Please set the address of WR and WM to [s+2, s+3] and [s+5, s+6] using the I/O address coding command.

[s+0] Specify a calculation method:

The calculation method of the check code can be specified form the following 7 kinds.

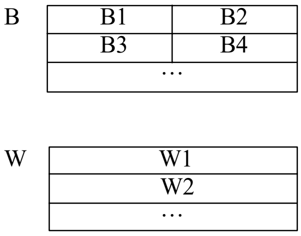
Set value	Calculation expression	Result of calculation	
H0000	(B1)+(B2)+ ⋯ +(Bn)	Byte	
H0001	(B1)+(B2)+ ⋯ +(Bn)	Word	Stores the found value in the order, upper part then lower part
H0002	(B1)+(B2)+ ⋯ +(Bn)	Word	Stores the found value in the order, lower part then upper part
H0003	(B1)+(B2)+ ⋯ +(Bn)	Byte	Stores the found value in word (in the order, upper then lower) after converted it to ASCII
H0004	(B1)+(B2)+ ⋯ +(Bn)	Byte	Stores the found value in word (in the order, lower then upper) after converted it to ASCII
H0005	(W1)+(W2)+ ⋯ +(Wn)	Word	Stores the found value in the order, upper part then lower part
H0006	(W1)+(W2)+ ⋯ +(Wn)	Word	Stores the found value in the order, lower part then upper part
H0010	{(B1)×or(B2)}×or⋯×or(Bn)	Byte	
H0011	{(B1)×or(B2)}×or⋯×or(Bn)	Byte	Stores the found value in word (in the order, upper then lower) after converted it to ASCII
H0012	{(B1)×or(B2)}×or⋯×or(Bn)	Byte	Stores the found value in word (in the order, lower then upper) after converted it to ASCII
H0013	{(W1)×or(W2)}×or⋯×or(Wn)	Word	Stores the found value in the order, upper part then lower part
H0014	{(W1)×or(W2)}×or⋯×or(Wn)	Word	Stores the found value in the order, upper part then lower part
H0020	LRC	Byte	
H0021	CRC16	Word	
Others	DATA Error (DER ON)		

[s+1] Specify Start byte for calculation / Start byte for collation:

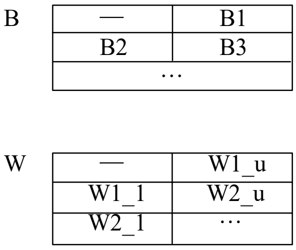
Only when the check code is calculated in the byte units, the start byte can be specified to either the upper byte or the lower byte. Also the start byte for collation can be specified to either the upper byte or the lower byte.

<Upper byte>
Specify start byte for calculation
H00xx: Start calculation from the upper byte
H01xx: Start calculation from the lower byte
Others: DATA Error (DER ON)

Set value: H00xx

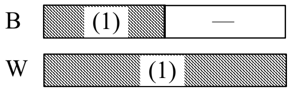


Set value: H01xx

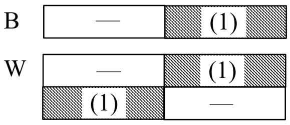


<Lower byte>
Specify start byte for collation
Hxx00: Set the calculation result to the upper byte
Hxx01: Ste the calculation result to the lower byte*
Others: DATA Error (DER ON)
* When the value to store is word, the lower byte is store in the upper byte of next word.

Set value: Hxx00



Set value: Hxx01



—: Data to be stored until then
(1): Store overwriting the calculation result

[s+2, s+3] Classification of the top I/O for calculating:

Please set the addresses of WR and WM using the I/O address coding command.

[s+4] Number of data:

Case of byte setting ⋯ Sets the number of bytes of data. (H0001 to HFFFF)

Case of word setting ⋯ Sets the number of words of data. (H0001 to HFFFF)

[s+5, s+6] Classification of the I/O to store the calculation result:

Please set the addresses of WR and WM using the I/O address coding command.

[d+0] Collation result:

OK ··· H8000, NG ··· H80FF

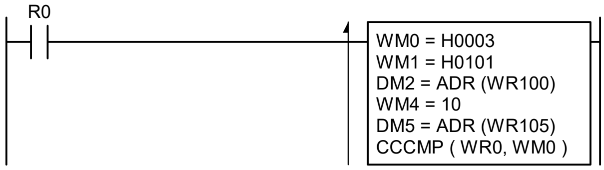
[d+1, d+2] Collation result:

The check code calculated actually is stored. When the check code to be collated extends over two words, a format of this calculation result also extends over two words.

Cautionary notes

- Please specify the internal output used for s parameter table, the internal output used for d parameter table, and the area to be calculated within the range of the I/O No.
- The check code to be calculated is only the word internal output. If other than the word internal output are specified, the command is not executed because of DER = 1.
- Please pay attention so that the area for calculating does not overlap with d and s parameters.
  If the area overlaps, the command is not executed because of DER = 1.

Program example



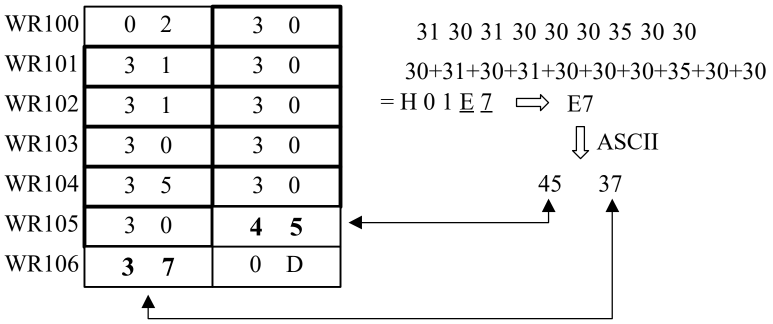
```
WM0 = H0003
WM1 = H0101
DM2 = ADR (WR100)
WM4 = 10
DM5 = ADR (WR105)
CCCMP ( WR0, WM0 )
```

[Program description]

- Executes the CCCMP command by setting the parameter for collating check code at the rising edge of R0.
- When the receiving data area is the data composition as follows, suppose that R0 was turned ON.

WR100	0 2	3 0
WR101	3 1	3 0
WR102	3 1	3 0
WR103	3 0	3 0
WR104	3 5	3 0
WR105	3 0	4 5
WR106	3 7	0 D

- If the sample program is executed, the result is as follows.



WR100	0 2	3 0
WR101	3 1	3 0
WR102	3 1	3 0
WR103	3 0	3 0
WR104	3 5	3 0
WR105	3 0	**4 5**
WR106	**3 7**	0 D

- WR0 = H8000 because the check code is matching. (If it is not matching, WR0 = H80FF.)

PRN ➔ PRJ

This command is equivalent to FUN 23 (s) in the program (PRN file) of MICRO-EH.

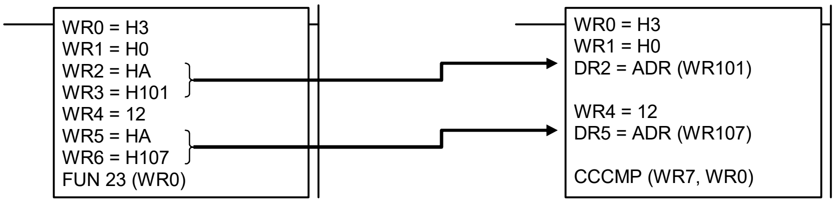
How to convert the program which has used FUN 23 (s) into the program for MICRO-EHV is as follows.

FUN 23 (s) ➔ CCCMP (s+7, s)

* If converted by Convert Tool started from Control Editor, it is converted as mentioned above. However you need to modify a part of s parameter.

Program for MICRO-EH

Program for MICRO-EHV



```
WR0 = H3
WR1 = H0
WR2 = HA
WR3 = H101
WR4 = 12
WR5 = HA
WR6 = H107
FUN 23 (WR0)
```

```
WR0 = H3
WR1 = H0
DR2 = ADR (WR101)

WR4 = 12
DR5 = ADR (WR107)

CCCMP (WR7, WR0)
```

Name	[Others] Process stepping

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
IFR (s)	–	3	↕	●	●	●	●

Command processing time (µs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	30.2	32.84	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	Top I/O in parameter table								✓					

Remarks
s parameters are occupied up to s+6.

Function

This is a process stepping command (Sequential control command).

Since a set input and a reset input can be specified to 1 point of bit I/O, the process stepping program can be realized with a regular format by combining this.

Parameter

Uses 7 words from the word I/O specified by s.

s	[0] Set input I/O
s+1	
s+2	[1] Process startup I/O
s+3	
s+4	[2] Reset input I/O
s+5	
s+6	Reserve (used by system)

[0] Set input I/O

Please specify the I/O to turn ON the process startup I/O.

[1] Process startup I/O

Please specify the I/O to be a startup condition of the process.

If [0] set input I/O turns ON, this I/O turns ON and if [2] reset input I/O turns ON, this I/O turns OFF.

[2] Reset input I/O

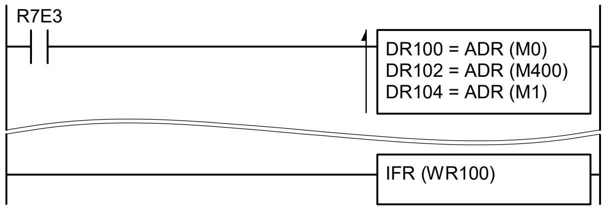
Please specify the I/O to turn OFF the process startup I/O.

Please set the address of R and M to the I/O specifying of [0] to [2] using the I/O address coding command.

Cautionary notes

- When s and s+1 (the set input), and s+4 and s+5 (the reset input) turn ON, s+4 and s+5 (the reset input) is given priority.
- When the areas specified by s to s+6 are overlapping and when the I/O specified by s to s+5 is outside the range, the process is not performed because of DER = 1.
- In each bit I/O specified by s parameter, do not specify the same I/O.

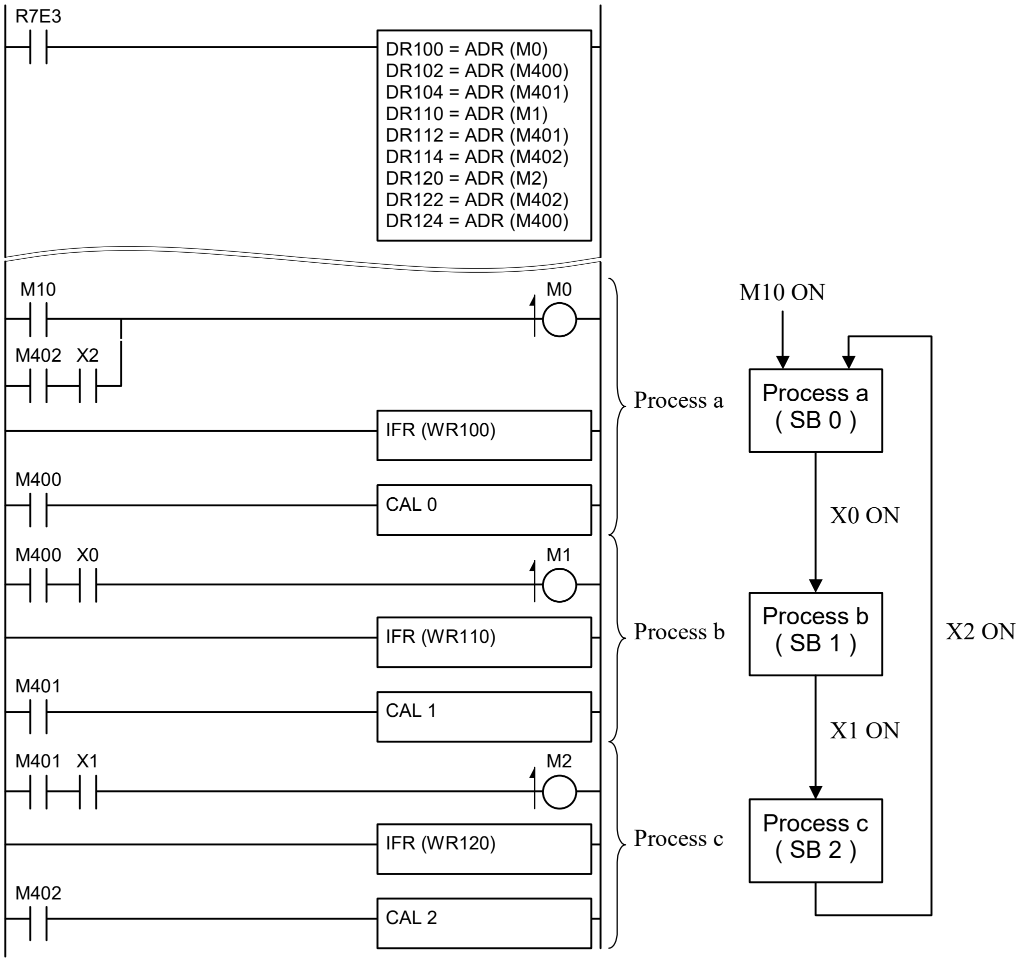
Program example



[Program description]

- Sets the set input (M0), the process startup input (M400), and the reset input (M1) at the 1st scan after RUN.

- Since IFR (s) is always running, if M0 is turned ON, M400 turns ON, and if M1 is turned ON, M400 turns OFF.



- The sequential control is possible by writing several IFR(s)s, and by the reset input of the previous process being the set input of the next process.

PRN → PRJ

This command is equivalent to FUN 4 (s) in the program (PRN file) of the MICRO-EH.

How to convert the program which has used FUN 4 (s) into the program for MICRO-EHV is as follows.

FUN 4 (s) → IFR (s)

* If converted by Convert Tool started from Control Editor, it is converted as follows. However, you need to modify a part of s parameter.

Program for MICRO-EH

```
ADRIO (WR0, M0)
ADRIO (WR1, M400)
ADRIO (WR2, M401)
FUN 4 (WR0)
```

Program for MICRO-EHV

```
DR0 = ADR (M0)
DR2 = ADR (M400)
DR4 = ADR (M401)
IFR (WR0)
```

[Note on converting for program]

Although s parameter is 3 words in the program of MICRO-EH, s parameter needs 6 words in MICRO-EHV. When converting the program, please make sure that s parameter area for incremental words is not being used for other purposes.

Name	Time range decision

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
TMRNGE (d, s1, s2)	—	6	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	—	—	—	24.2	24.4

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Result		✓	✓											
s1	Lower limit of time range									✓	✓				✓
s2	Upper limit of time range									✓	✓				✓

Remarks

Function

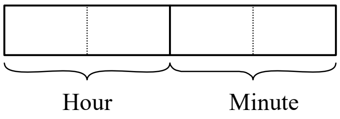
When the clock data (hour and minute data) of MICRO-EHV is s1 or more and less than s2, the I/O specified by d turns ON. In other cases, the I/O specified by d turns OFF.

Parameter

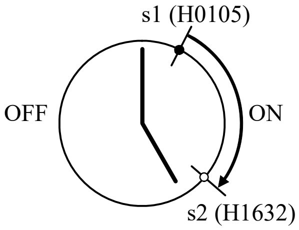
d : Specifies I/O which indicates whether the current time is in the specified range.

In the specified range … ON, Outside the specified range … OFF

s1, s2 : Specifies the ON time(s1) and the OFF time (s2) in the 24-hour system.



Example) Case of 18:32… H1832



In the left figure, the I/O specified by d turns ON when the time data of MICRO-EHV is from 1:05 to 16:31. And the I/O turns OFF when it is from 0:00 to 1:04 on that day and it is from 16:32 on that day to 1:04 on the next day.

Cautionary notes

- If it is executed on the following conditions, DER = 1 and it is not executed.

\- the hour and minute data specified by s1 and s2 is invalid. (Example: H2900, H1482, and H022B)

\- the hour and minute data specified by s1 and s2 is s1 = s2.

- Please specify the internal output used for d, s1, and s2 within the range of I/O No.

Name	User-defined function

Ladder format	Number of steps		Condition code				
	Condition	Step	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
d = UFNC (s1, s2)	—	5	↕	●	●	●	●

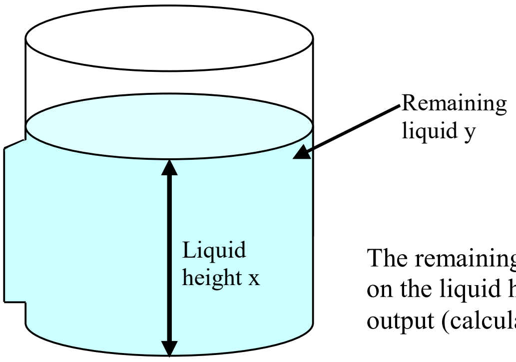
Instruction processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	—	—	—	—	—

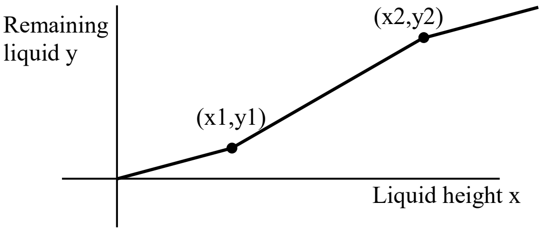
Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
d	Function output value storage I/O								✓				✓	
s1	Function input value I/O								✓				✓	
s2	Top I/O in function data								✓				✓	

Remarks

Function

- This instruction defines a user-specific function according to the table specified in s2. It is defined as a broken line that connects up to 255 points with straight lines.
- This returns the Y-coordinate value with respect to a specific X-coordinate value given in s1 based on the above function.
- This instruction is useful in calculating the remaining liquid from the liquid height in the tank of any shape.





The remaining liquid (Y coordinate) is pre-defined by the user based on the liquid height (X coordinate), and the remaining liquid is output (calculated) with respect to the current input (liquid height).

Parameter

d: Specify the first I/O of the internal output that stores the value (Y-coordinate value) output after the user-defined function operation.

s1: Specify the first I/O of the internal output that stores the value (X-coordinate value) to input the user-defined function.

s2: Specify the first I/O of the function data table.

(1) Parameter d

Specify d with the first I/O of the internal output that stores the value (Y-coordinate value) output after the user-defined function operation.

[For word data]

d+0	[1] Output value

[1] Output value: The value (Y-coordinate value) output after the user-defined function operation is set.

[For double word data]

d+0	[1] Output value (low word)
d+1	Output value (high word)

(2) Parameter s1

Specify s1 with the first I/O of the internal output that stores the value (X-coordinate value) to input the user-defined function.

[For word data]

s1+0	[1] Input value

[1] Input value: The input value (X-coordinate value) is set.

[For double word data]

s1+0	[1] Input value (low word)
s1+1	Input value (high word)

(3) Parameter s2

Specify s2 with the first I/O of the function data table.

[For word data]

s2+0	[1] Data type code
s2+1	[2] Number of data (n)
s2+2	[3] Data X0
s2+3	[4] Data Y0
s2+4	[5] Data X1
s2+5	[6] Data Y1
s2+6	[7] Data X2
s2+7	[8] Data Y2
⋮	⋮
s2+2n	Data Xn-1
s2+2n+1	Data Yn-1

[For double word data]

s2+0	[1] Data type code
s2+1	[2] Number of data (n)
s2+2	[3] Data X0 (low word)
s2+3	[3] Data X0 (high word)
s2+4	[4] Data Y0 (low word)
s2+5	[4] Data Y0 (high word)
s2+6	[5] Data X1 (low word)
s2+7	[5] Data X1 (high word)
s2+8	[6] Data Y1 (low word)
s2+9	[6] Data Y1 (high word)
⋮	⋮
s2+4n-2	Data Xn-1 (low word)
s2+4n-1	Data Xn-1 (high word)
s2+4n	Data Yn-1 (low word)
s2+4n+1	Data Yn-1 (high word)

[1] Data type code: Specify the data type to use.

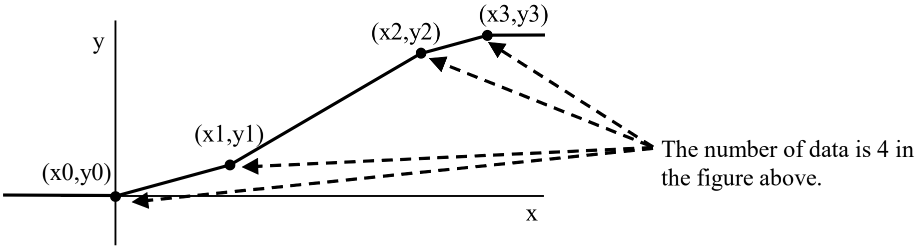
[For word data]

Data type	Setting value
Unsigned	H0001
Signed	H0002

[For double word data]

Data type	Setting value
Unsigned	H0004
Signed	H0008
Floating point	HFFFF

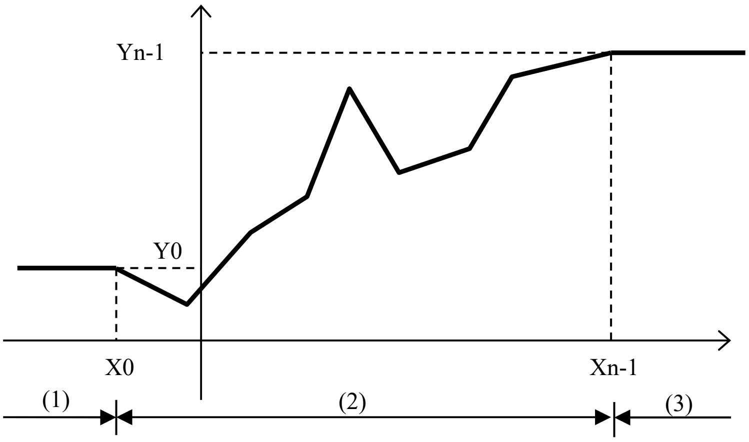
[2] Number of data: Specify the number of coordinates to define. The setting range is from 2 to 255.



[3][5][7] Data X: Specify the coordinate position of X axis.

[4][6][8] Data Y: Specify the coordinate position of Y axis.

Output value d with respect to input value s1 is calculated based on the function given in data table s2.



(1) When [s1 < X0]

If input value s1 is smaller than X0 in data table s2, Y0 in data table s2 is returned as output value d.

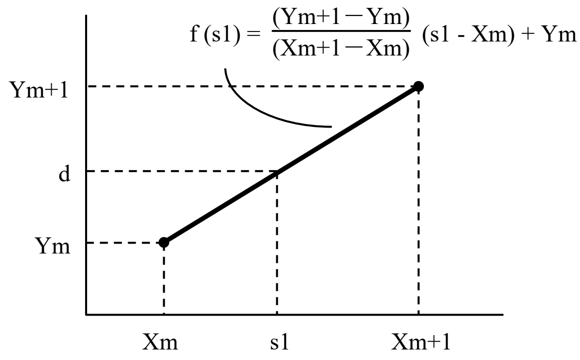
d = UFNC (s1, s2) = Y0

(2) When [X0 ≦ s1 < Xn-1]

If input value s1 is within the domain of X given in data table s2, output value d is obtained from the following formula:

When Xm ≦ s1 < Xm+1

$$d = \text{UFNC}(s1, s2) = \frac{(Ym+1 - Ym)}{(Xm+1 - Xm)}(s1 - Xm) + Ym$$



(3) When [Xn-1 ≦ s1]

If input value s1 is greater than Xn-1 in data table s2, Yn-1 in data table s2 is returned as output value d.

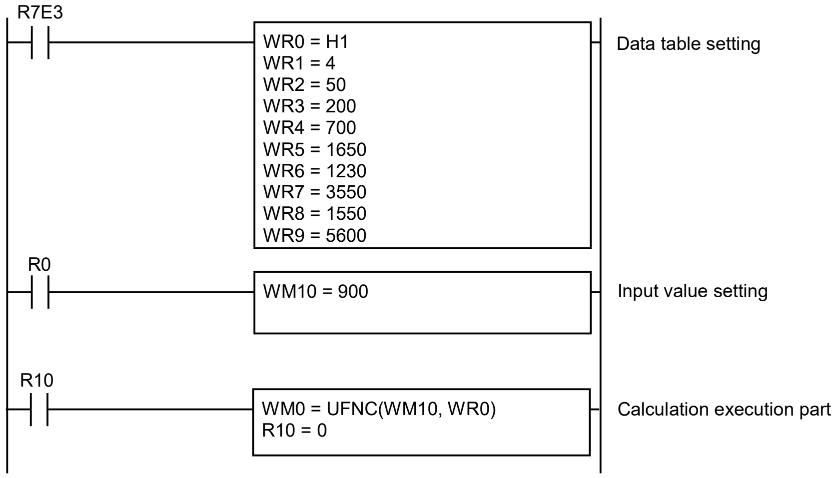
d = UFNC (s1, s2) = Yn-1

Cautionary notes

- If the specified data type value is invalid, DER is set to 1, which does not execute the operation.
- If the specified data type is an integer, the calculation result is truncated.
- The internal output and data table to be used for parameter d or s must be valid I/O Nos.
- If the area of parameter d or s overlaps the area of the data table, DER is set to 1, which does not execute the operation.
- If the number of data (n) in the data table is other than 2 to 255, DER is set to 1, which does not execute the operation.
- Xm in the data table must be placed in ascending order (X0 < X1 < ... < Xn-1). If data is Xm ≧ Xm+1, DER is set to 1, which does not execute the operation.

Example

[For word data]



```
WR0 = H1
WR1 = 4
WR2 = 50
WR3 = 200
WR4 = 700
WR5 = 1650
WR6 = 1230
WR7 = 3550
WR8 = 1550
WR9 = 5600
```

```
WM10 = 900
```

```
WM0 = UFNC(WM10, WR0)
R10 = 0
```

[Program description]

The table data (WR0 to WR9 in this example) are defined in one scan after RUN.

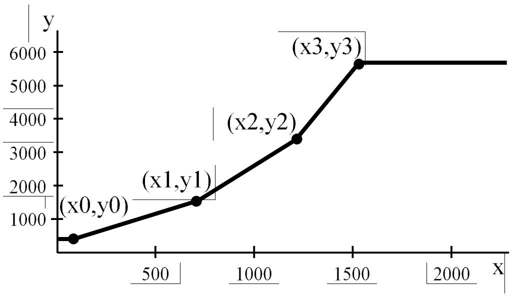
When R0 is turned ON, the input value is set.

When R10 is turned ON, the user-defined function is executed with the input value of 900, and then R10 is turned OFF.

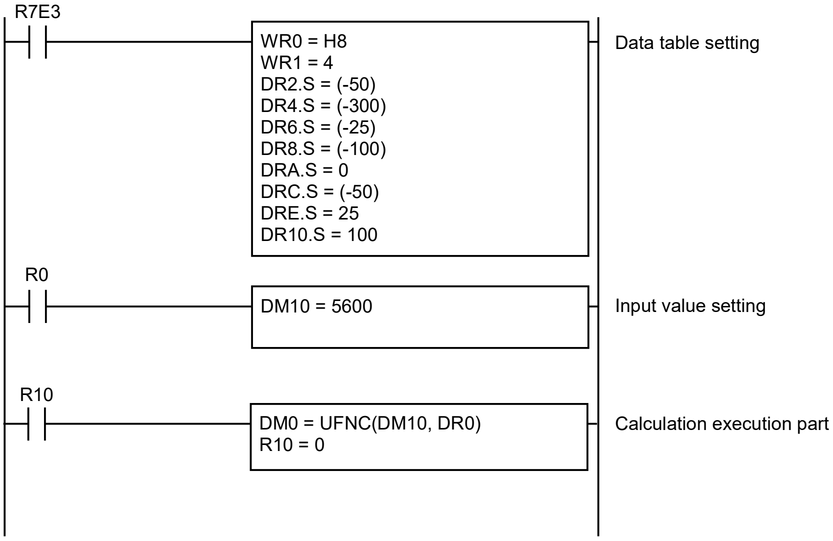
Table setting values specified in s2

Coordinate position	Setting value
(x0,y0)	(50,200)
(x1,y1)	(700,1650)
(x2,y2)	(1230,3550)
(x3,y3)	(1550,5600)

Table coordinate graph specified in s2



[For double word data]



```
WR0 = H8
WR1 = 4
DR2.S = (-50)
DR4.S = (-300)
DR6.S = (-25)
DR8.S = (-100)
DRA.S = 0
DRC.S = (-50)
DRE.S = 25
DR10.S = 100
```

```
DM10 = 5600
```

```
DM0 = UFNC(DM10, DR0)
R10 = 0
```

[Program description]

The table data (signed DR0 to DR10 in this example) are defined in one scan after RUN. At this time, word data values are set in s2+0 and s2+1.

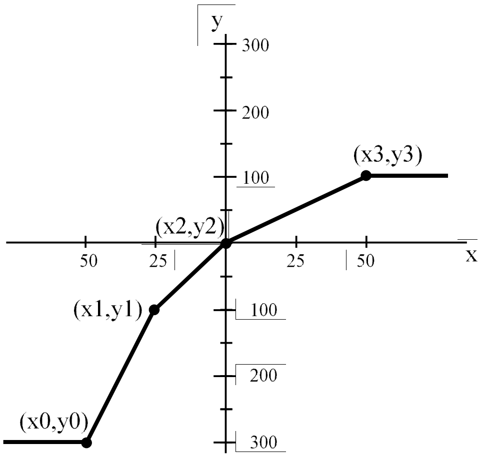
When R0 is turned ON, the input value is set.

When R10 is turned ON, the user-defined function is executed with the input value, and then R10 is turned OFF.

Table setting values specified in s2

Coordinate position	Setting value
(x0,y0)	(-50,-300)
(x1,y1)	(-25,-100)
(x2,y2)	(0,-50)
(x3,y3)	(50,100)

Table coordinate graph specified in s2



Name	Counter control

Ladder format	Number of steps		Condition code				
	Condition	Step	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
CUSTA (s)	–	3	↕	●	●	●	●

Instruction processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	6.66	8.06	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	Top I/O of control parameter								✓					

Remarks
s parameters are occupied up to s+2.

Function

- This instruction starts or stops the specified single-phase/two-phase counter.

Parameter

s: Specify the top I/O of the control parameter table.

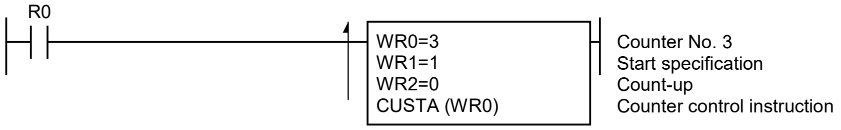
	Setting description	R/W	Setting range, setting description
s+0	Counter No. *	W	Single-phase : 1 to 5 Two-phase : 1, 3
s+1	Start/Stop	W	0: Stop, 1: Start, 2: Direction change
s+2	Direction	W	0: Upward, 1: Downward

* When a two-phase counter is used, counter No. is 1 or 3.

Cautionary notes

- s+1 must be within the I/O range. You cannot write a parameter that is outside the I/O range.
- If you try to execute this instruction when invalid values are set in s+0 and s+1, DER is set to 1, which does not execute the instruction.
- If counter input is not configured, this instruction is not executed with DER = 1.
- If you try to execute this instruction to counter No. 2 when counter No. 1 is two-phase, DER is set to 1, which does not execute the instruction. Similarly, if you try to execute this instruction to counter No. 4 when counter No. 3 is two-phase, DER is set to 1, which does not execute the instruction.
- If you stop the CPU after starting the counter, it continues to count.
- If the PLC is turned OFF when the counter is started by this instruction, the counter starts when the PLC is turned ON again. The counter can be stopped only by this stop instruction.
- When the counter is stopped, the current counter value also stops being updated. Before the counter starts, the current value is cleared to 0.

Example



[Program description]

At the rising edge of R0, counter No. 3 starts counting.

Name	Current counter value read

Ladder format	Number of steps		Condition code				
	Condition	Step	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
CURD (s)	–	3	↕	●	●	●	●

Instruction processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	6.28	7.64	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	Top I/O of control parameter								✓					

Remarks
s parameters are occupied up to s+2.

Function

- This instruction reads the current value of the specified single-phase/two-phase counter.

Parameter

s: Specify the top I/O of the control parameter table.

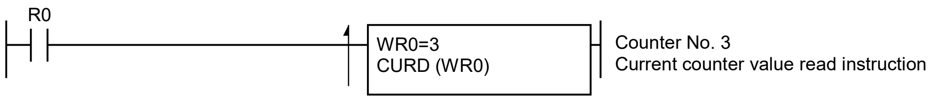
	Setting description	R/W	Setting range, setting description
s+0	Counter No. *	W	Single-phase : 1 to 5 Two-phase : 1, 3
s+1	Current counter value (low word)	R	0 to 4,294,967,295
s+2	Current counter value (high word)	R	

* When a two-phase counter is used, counter No. is 1 or 3.

Cautionary notes

- s+2 must be within the I/O range. You cannot write a parameter that is outside the I/O range.
- If you try to execute this instruction when an invalid value is set in s+0, DER is set to 1, which does not execute the instruction.
- If counter input is not configured, this instruction is not executed with DER = 1.
- If you try to execute this instruction to counter No. 2 when counter No. 1 is two-phase, DER is set to 1, which does not execute the instruction. Similarly, if you try to execute this instruction to counter No. 4 when counter No. 3 is two-phase, DER is set to 1, which does not execute the instruction.

Example



[Program description]

At the rising edge of R0, the current value of counter No. 3 is read out to internal output DR1.

Name	Current counter value write

Ladder format	Number of steps		Condition code				
	Condition	Step	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
CUWR (s)	–	3	↕	●	●	●	●

Instruction processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	6.28	7.64	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	Top I/O of control parameter									✓					

Remarks
s parameters are occupied up to s+2.

Function

- This instruction overwrites the current value of the specified single-phase/two-phase counter.
- If the counter write value is set to 0, the current counter value is cleared to 0.

Parameter

s: Specify the top I/O of the control parameter table.

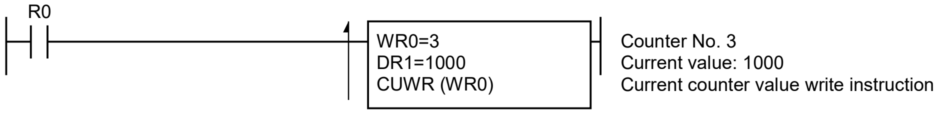
	Setting description	R/W	Setting range, setting description
s+0	Counter No. *	W	Single-phase : 1 to 5 Two-phase : 1, 3
s+1	Counter write value (low word)	W	0 to 4,294,967,295
s+2	Counter write value (high word)	W	

* When a two-phase counter is used, counter No. is 1 or 3.

Cautionary notes

- s+2 must be within the I/O range. You cannot write a parameter that is outside the I/O range.
- If you try to execute this instruction when an invalid value is set in s+0, DER is set to 1, which does not execute the instruction.
- If counter input is not configured, this instruction is not executed with DER = 1.
- If you try to execute this instruction to counter No. 2 when counter No. 1 is two-phase, DER is set to 1, which does not execute the instruction. Similarly, if you try to execute this instruction to counter No. 4 when counter No. 3 is two-phase, DER is set to 1, which does not execute the instruction.
- The current value can be overwritten by this instruction during counting.

Example



[Program description]

At the rising edge of R0, the current value of counter No. 3 is overwritten with 1,000.

Name	Counter comparison value setting

Ladder format	Number of steps		Condition code				
	Condition	Step	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
CUPRE (s)	—	3	↕	●	●	●	●

Instruction processing time (µs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	6.28	7.64	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	Top I/O of control parameter								✓					

Remarks
s parameters are occupied up to s+5.

Function

- This instruction sets the first comparison value and second comparison value of the specified single-phase/two-phase counter.
- If the first comparison value and second comparison value are the same, a match interrupt does not occur.

Parameter

s: Specify the top I/O of the control parameter table.

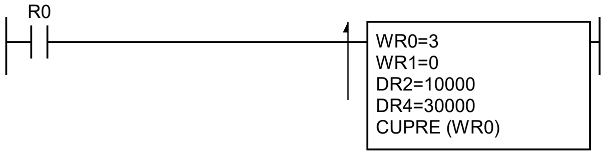
	Setting description	R/W	Setting range, setting description
s+0	Counter No. *	W	Single-phase : 1 to 5 Two-phase : 1, 3
s+1	ON/OFF specification	W	0 : Set both the first and second comparison values 1 : Set the first comparison value only 2 : Set the second comparison value only
s+2	First comparison value (low word)	W	0 to 4,294,967,295
s+3	First comparison word (high word)	W	
s+4	Second comparison value (low word)	W	0 to 4,294,967,295
s+5	Second comparison word (high word)	W	

* When a two-phase counter is used, counter No. is 1 or 3.

Cautionary notes

- s+5 must be within the I/O range. You cannot write a parameter that is outside the I/O range.
- If you try to execute this instruction when invalid values are set in s+0 to s+5, DER is set to 1, which does not execute the instruction.
- If counter input is not configured, this instruction is not executed with DER = 1.
- If you try to execute this instruction to counter No. 2 when counter No. 1 is two-phase, DER is set to 1, which does not execute the instruction. Similarly, if you try to execute this instruction to counter No. 4 when counter No. 3 is two-phase, DER is set to 1, which does not execute the instruction.
- This instruction is valid even during counting. Please note that if the instruction is executed during counting, the comparison values are overwritten.

Example



Counter No. 3
Setting of both comparison values 1 and 2
Comparison value 1: 10000
Comparison value 2: 30000
Counter comparison value setting instruction

[Program description]

At the rising edge of R0, the first comparison value of counter No. 3 is set to 10,000, the counter is set not to be cleared at the first comparison, the second comparison value is set to 30,000, and the counter is set not to be cleared at the second comparison.

Name	PWM output start / change

Ladder format	Number of steps		Condition code				
	Condition	Step	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
PWMSTA (s)	–	2	↕	●	●	●	●

Instruction processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	9.4	11.28	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	Control parameter first I/O									✓					

Remarks
s parameters are occupied up to s+3.

Function

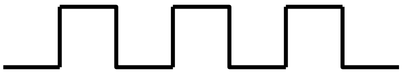
- This instruction starts PWM output with the specified output frequency and ON-duty from the specified output No.
- Executing this instruction during PWM output allows you to change the output frequency and ON-duty.

Parameter

s: Specify the first I/O of the control parameter table.

	Setting description	R/W	Setting range, setting description
s+0	PWM output No.	W	1 to 3
s+1	Output frequency (low word)	W	20 to 100,000 (Hz)
s+2	Output frequency (high word)	W	
s+3	ON-duty	W	0 to 100 (%)

<ON-duty: 50 %>

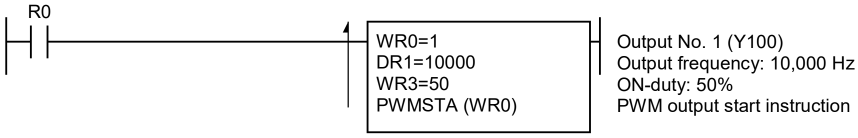


<ON-duty: 25 %>

Cautionary notes

- s+3 must be within the I/O range. You cannot write a parameter that is outside the I/O range.
- If you try to execute this instruction when invalid values are set in s+0 to s+3, DER is set to 1, which does not execute the instruction.
- If counter input is not configured, this instruction is not executed with DER = 1.
- When the CPU stops, the PWM output stops.
- The minimum output frequency is 20 Hz. Even if less than 20 Hz is set, it is treated as 20 Hz.
- The maximum output frequency is 100,000 Hz. Even if more than 100,000 Hz is set, it is treated as 100,000 Hz.

Example



[Program description]

At the rising edge of R0, the output frequency of PWM output No. 1 is set to 10,000 Hz, and the ON-duty is set to 50 %.

Name	PWM output stop

Ladder format	Number of steps		Condition code				
	Condition	Step	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
PWMSTP (s)	–	2	↕	●	●	●	●

Instruction processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	6.18	7.62	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	Control parameter first I/O								✓					

Remarks

Function

- This instruction stops the PWM output of the specified PWM output No.

Parameter

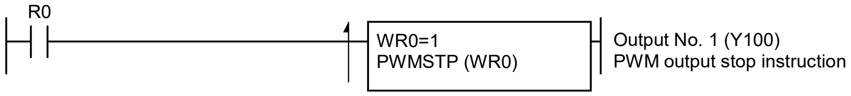
s: Specify the first I/O of the control parameter table.

	Setting description	R/W	Setting range, setting description
s+0	PWM output No.	W	1 to 3

Cautionary notes

- If counter input is not configured, this instruction is not executed with DER = 1.
- If you try to execute this instruction when an invalid value is set in s+0, DER is set to 1, which does not execute the instruction.

Example



[Program description]

At the rising edge of R0, the PWM output of PWM output No. 1 is stopped.

Name	Pulse output start

Ladder format	Number of steps		Condition code				
	Condition	Step	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
PLSTA (s)	–	3	↕	●	●	●	●

Instruction processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	228	320	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	Control parameter first I/O									✓					

Remarks
Specification of absolute position is supported from Ver.x120. s parameters are occupied up to s+8.

Function

- This instruction outputs pulse trains (without or with acceleration/deceleration) from the specified output No. and stops the output when the specified number of pulses is output.
- Executing this instruction in operation mode 0: pulse output (relative position) while outputting pulses in operation mode 0: pulse output (relative position) allows you to change the output frequency and the number of output pulses.
- If the frequency and the number of pulses are changed during pulse output, the operation continues with the new frequency and number of pulses values.

Parameter

s: Specify the first I/O of the control parameter table.

	Setting description	R/W	Setting range, setting description
s+0	Output No.	W	1 to 3
s+1	Operation mode	W	H0000: Pulse output (relative position) H0001: Pulse output with acceleration/deceleration (relative position) H0010: Pulse output (absolute position) H0011: Pulse output with acceleration/deceleration (absolute position)
s+2	Output frequency (low word)	W	20 to 100,000 (Hz)
s+3	Output frequency (high word)	W	
s+4	Number of output pulses (low word)	W	(Relative position) Represents the travel distance. 0 to 4,294,967,295 : When the direction signal is disabled -2,147,483,648 to 2,147,483,647 : When the direction signal is enabled (Absolute position) Represents the target position. 0 to 4,294,967,295 : When the direction signal is disabled -2,147,483,648 to 2,147,483,647 : When the direction signal is enabled * When the direction signal is disabled, an absolute position can be specified but moved only in + direction.
s+5	Number of output pulses (high word)	W	
s+6	Initial frequency	W	20 to 65,535 (Hz) (Valid only for pulse output with acceleration/deceleration)
s+7	Acceleration rate	W	1 to 65,535 (Hz/100 ms) (Valid only for pulse output with acceleration/deceleration)
s+8	Deceleration rate	W	1 to 65,535 (Hz/100 ms) (Valid only for pulse output with acceleration/deceleration)

- To enable/disable the direction signal, use the Control Editor.

[s+0] Output No.

The correspondence between the pulse train output signal (PLS) and the direction signal is as follows:

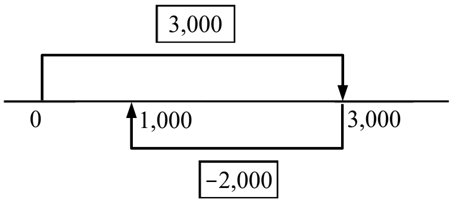
Output No.	Pulse train output (PLS)	Direction signal
1	Y100	Y103
2	Y101	Y104
3	Y102	Y105

- To enable/disable the direction signal, use the Control Editor.
- To set the polarity of the direction signal, use the Control Editor.

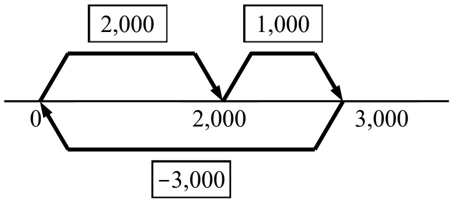
[s+1] Operation mode

Operation mode	Name	Description
H0000	Pulse output (relative position)	Performs a relative position move without acceleration/deceleration by setting the current position as the starting point. Therefore, s+6 (Initial frequency), s+7 (Acceleration rate), and s+8 (Deceleration rate) will be invalid.
H0001	Pulse output with acceleration/deceleration (relative position)	Performs a relative position move with acceleration/deceleration by setting the current position as the starting point.
H0010	Pulse output (absolute position)	Performs an absolute position move without acceleration/deceleration. Therefore, s+6 (Initial frequency), s+7 (Acceleration rate), and s+8 (Deceleration rate) will be invalid.
H0011	Pulse output with acceleration/deceleration (absolute position)	Performs an absolute position move with acceleration/deceleration.

<Pulse output (relative position)>

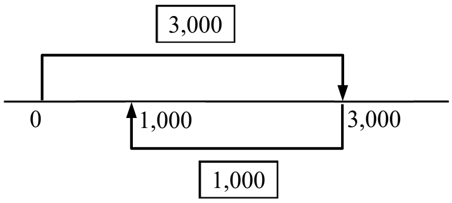


<Pulse output with acceleration/deceleration (relative position)>

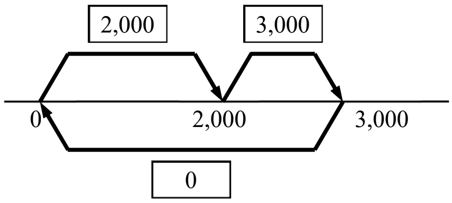


The framed value represents the number of output pulses (s+4, s+5).

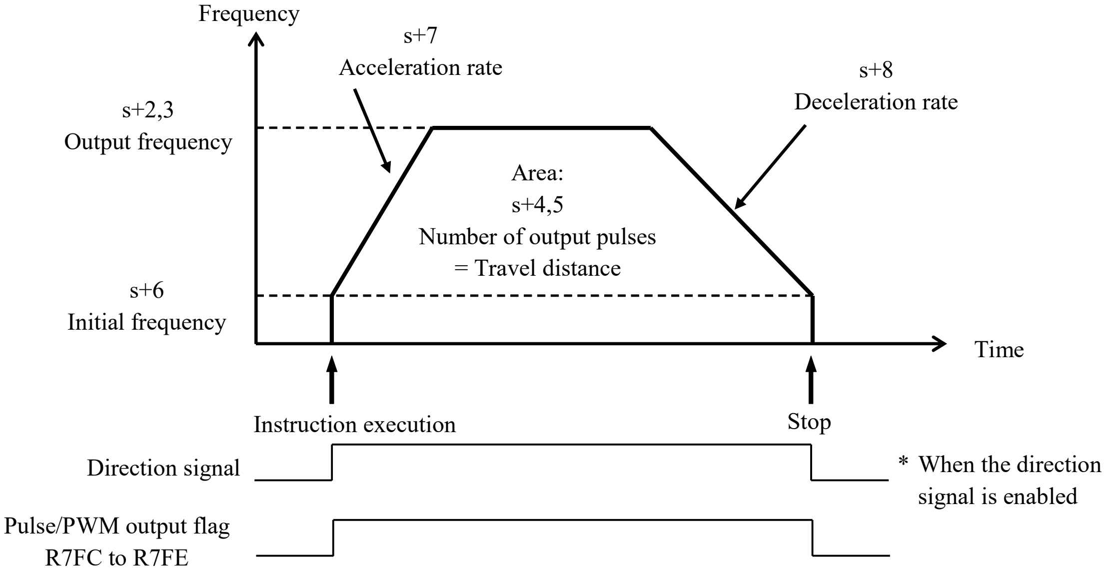
<Pulse output (absolute position)>



<Pulse output with acceleration/deceleration (absolute position)>



[s+2, 3] Output frequency, [s+4, 5] Number of output pulses, [s+6] Initial frequency, [s+7] Acceleration rate, [s+8] Deceleration rate



The unit of acceleration rate and deceleration rate is Hz/100 ms. The conversion formula from the acceleration/ deceleration time (sec) is as follows:

$$\text{Acceleration/deceleration rate} = \frac{\text{Output frequency [Hz] - Initial frequency [Hz]}}{\text{Acceleration/deceleration time [s]}} \times \frac{1}{10}$$

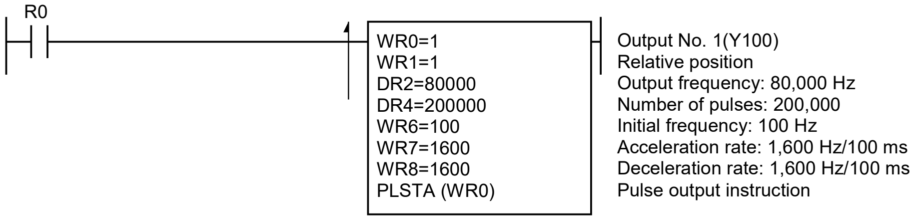
Example: To accelerate to 30,000 Hz in five seconds with the initial frequency of 200 Hz:

$$\text{Acceleration rate} = \frac{30{,}000 - 200}{5} \times \frac{1}{10} = 596 \quad \text{[Hz/100 ms]}$$

Cautionary notes

- s+8 must be within the I/O range. You cannot write a parameter that is outside the I/O range.
- If counter input is not configured, this instruction is not executed with DER = 1.
- If you try to execute this instruction during pulse output with the PLSPD instruction (pulse speed control start), DER is set to 1, which does not execute the instruction.
- If you try to execute this instruction again during pulse output in operation mode H0001 (pulse output with acceleration/deceleration), DER is set to 1, which does not execute the instruction.
- If you try to execute this instruction again in operation mode H0001 (pulse output with acceleration/deceleration) during pulse output in operation mode H0000 (pulse output (relative position)), DER is set to 1, which does not execute the instruction.
- When the CPU stops, the pulse output stops.
- The minimum output frequency is 20 Hz. Even if less than 20 Hz is set, it is treated as 20 Hz.
- The maximum output frequency is a total of 100,000 Hz.
- This instruction cannot be accepted during acceleration/deceleration.

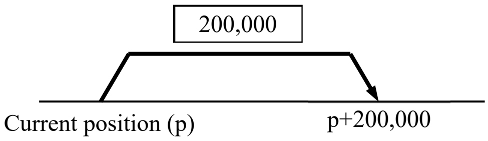
Example



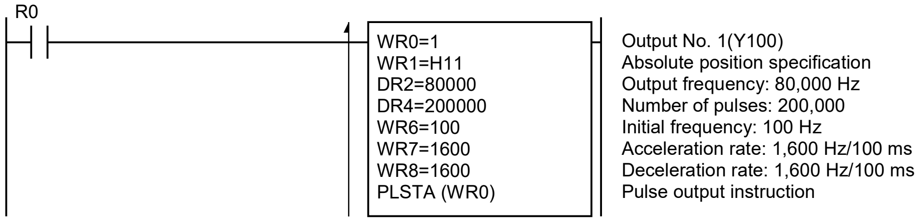
WR0=1	Output No. 1(Y100)
WR1=1	Relative position
DR2=80000	Output frequency: 80,000 Hz
DR4=200000	Number of pulses: 200,000
WR6=100	Initial frequency: 100 Hz
WR7=1600	Acceleration rate: 1,600 Hz/100 ms
WR8=1600	Deceleration rate: 1,600 Hz/100 ms
PLSTA (WR0)	Pulse output instruction

[Program description]

At the rising edge of R0, pulses with the initial frequency of 100 [Hz], acceleration rate of 1,600 [Hz/100 ms], deceleration rate of 1,600 [Hz/100 ms], output frequency of 80,000 [Hz], and the number of pulses of 200,000 are output to output No. 1.



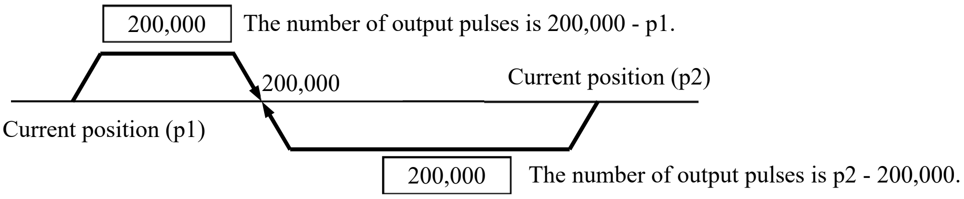
The framed value represents the number of output pulses (s+4, s+5).



WR0=1	Output No. 1(Y100)
WR1=H11	Absolute position specification
DR2=80000	Output frequency: 80,000 Hz
DR4=200000	Number of pulses: 200,000
WR6=100	Initial frequency: 100 Hz
WR7=1600	Acceleration rate: 1,600 Hz/100 ms
WR8=1600	Deceleration rate: 1,600 Hz/100 ms
PLSTA (WR0)	Pulse output instruction

[Program description]

At the rising edge of R0, output No. 1 is moved to a position with the initial frequency of 100 [Hz], acceleration rate of 1,600 [Hz/100 ms], deceleration rate of 1,600 [Hz/100 ms], output frequency of 80,000 [Hz], and the number of pulses of 200,000. If the current position is greater than 200,000 pulses, the output moves in the reverse direction, and if smaller, it moves in the forward direction. Output pulses are with acceleration/deceleration.



The framed value represents the number of output pulses (s+4, s+5).

Name	Pulse output start

Ladder format	Number of steps		Condition code				
	Condition	Step	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
PLSTAR (s)	–	3	↕	●	●	●	●

Instruction processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	228	320	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	Control parameter first I/O									✓					

Remarks
This instruction is supported in Ver. x120 or later. s parameters are occupied up to s+8.

Function

- This instruction specifies ten times the acceleration rate and deceleration rate of the PLSTA (s) (pulse output start) instruction.

  Other settings are the same as PLSTA (s).

Parameter

s: Specify the first I/O of the control parameter table.

	Setting description	R/W	Setting range, setting description
s+0	Output No.	W	1 to 3
s+1	Operation mode	W	H0000: Pulse output (relative position) H0001: Pulse output with acceleration/deceleration (relative position) H0010: Pulse output (absolute position) H0011: Pulse output with acceleration/deceleration (absolute position)
s+2	Output frequency (low word)	W	20 to 100,000 (Hz)
s+3	Output frequency (high word)	W	
s+4	Number of output pulses (low word)	W	(Relative position) Represents the travel distance. 0 to 4,294,967,295 : When the direction signal is disabled -2,147,483,648 to 2,147,483,647 : When the direction signal is enabled (Absolute position) Represents the target position. 0 to 4,294,967,295 : When the direction signal is disabled -2,147,483,648 to 2,147,483,647 : When the direction signal is enabled * When the direction signal is disabled, an absolute position can be specified but moved only in + direction.
s+5	Number of output pulses (high word)	W	

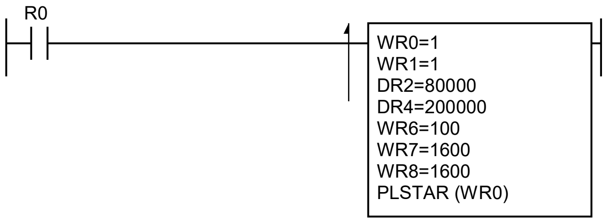
	Setting description	R/W	Setting range, setting description
s+6	Initial frequency	W	20 to 65,535 (Hz) (Valid only for pulse output with acceleration/deceleration)
s+7	Acceleration rate	W	1 to 65,535 (Hz / 10 ms) (Valid only for pulse output with acceleration/deceleration)
s+8	Deceleration rate	W	1 to 65,535 (Hz / 10 ms) (Valid only for pulse output with acceleration/deceleration)

- To enable/disable the direction signal, use the Control Editor.
- For details on any other parameter than the acceleration/deceleration rate, see the description of PLSTA (s).

Cautionary notes

This instruction has the same specifications as those of PLSTA (s) except for the acceleration/deceleration rate setting. Even the precautions are the same as PLSTA (s), so see the PLSTA (s) section for the precautions.

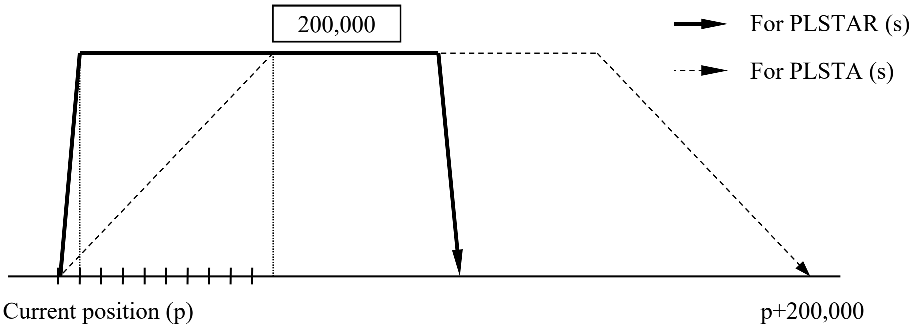
Example



```
WR0=1           Output No. 1(Y100)
WR1=1           Relative position
DR2=80000       Output frequency: 80,000 Hz
DR4=200000      Number of pulses: 200,000
WR6=100         Initial frequency: 100 Hz
WR7=1600        Acceleration rate: 1,600 Hz/10 ms
WR8=1600        Deceleration rate: 1,600 Hz/10 ms
PLSTAR (WR0)    Pulse output instruction
```

[Program description]

At the rising edge of R0, pulses with the initial frequency of 100 [Hz], acceleration rate of 1,600 [Hz/10 ms], deceleration rate of 1,600 [Hz/10 ms], output frequency of 80,000 [Hz], and the number of pulses of 200,000 are output to output No. 1.



Even if the same parameters are used, output pulses are different between the PLSTAR (s) and PLSTA (s) instructions as shown in the figure above.

Name	Pulse speed control start

Ladder format	Number of steps		Condition code				
	Condition	Step	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
PLSPD (s)	–	3	↕	●	●	●	●

Instruction processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	124	172	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	Control parameter first I/O								✓					

Remarks
s parameters are occupied up to s+5.

Function

- This instruction accelerates with the specified acceleration rate from the specified output No. and keeps outputting pulse trains with the specified frequency.
- To stop outputting pulse trains, execute the stop instruction (PLSTP). To change the frequency, execute the frequency change instruction (PLCNG).

Parameter

s: Specify the first I/O of the control parameter table.

	Setting description	R/W	Setting range, setting description
s+0	Output No.	W	1 to 3
s+1	Rotation direction	W	0: Forward, 1: Reverse
s+2	Initial frequency	W	20 to 65,535 (Hz)
s+3	Acceleration rate	W	1 to 65,535 (Hz/100ms)
s+4	Output frequency (low word)	W	20 to 100,000 (Hz)
s+5	Output frequency (high word)	W	

- To enable/disable the direction signal, use the Control Editor.
- If the direction signal is disabled, [s+1] Rotation direction is ignored.

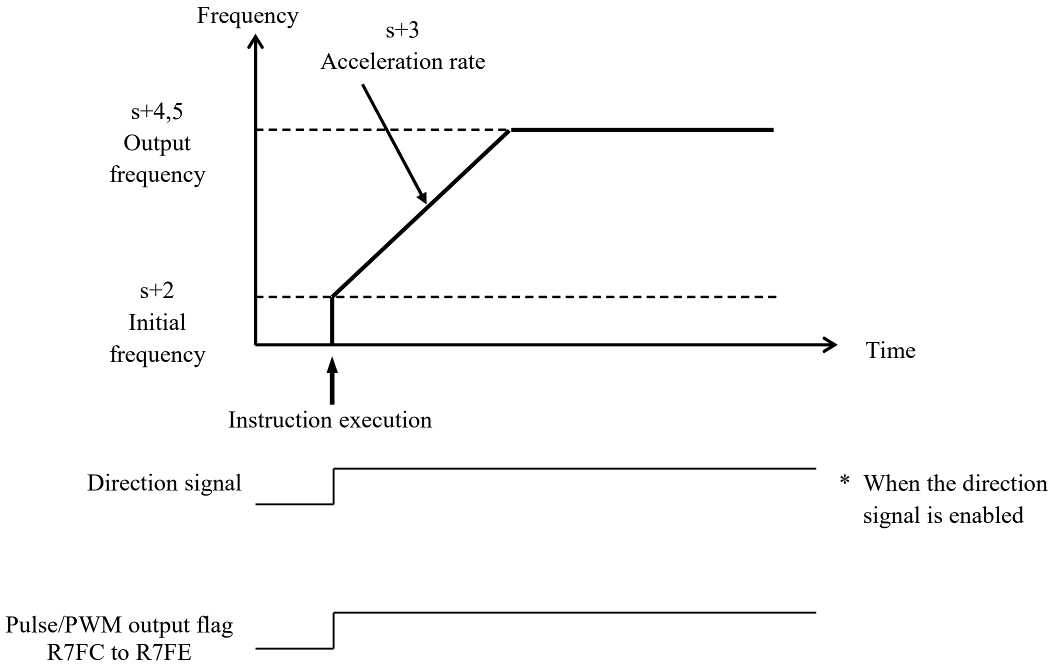
[s+0] Output No.

The correspondence between the pulse train output signal (PLS) and the direction signal is as follows:

Output No.	Pulse train output (PLS)	Direction signal
1	Y100	Y103
2	Y101	Y104
3	Y102	Y105

- To enable/disable the direction signal, use the Control Editor.
- To set the polarity of the direction signal, use the Control Editor.

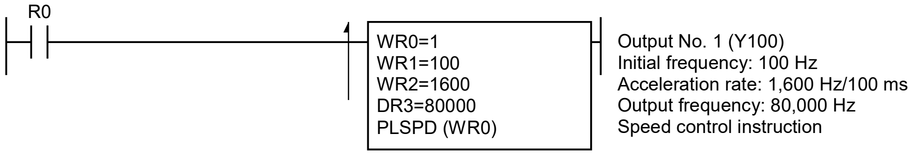
[s+1] Rotation direction, [s+2] Initial frequency, [s+3] Acceleration time, [s+4, 5] Output frequency



Cautionary notes

- s+5 must be within the I/O range. You cannot write a parameter that is outside the I/O range.
- If you try to execute this instruction when invalid values are set in s+0 and s+1 (s+4 and s+5), DER is set to 1, which does not execute the instruction.
- If counter input is not configured, this instruction is not executed with DER = 1.
- If you try to execute this instruction during pulse output with the PLSTA instruction (pulse output start), DER is set to 1, which does not execute the instruction.
- When the CPU stops, the pulse output stops.
- The minimum output frequency is 20 Hz. Even if less than 20 Hz is set, it is treated as 20 Hz.
- The maximum output frequency is a total of 100,000 Hz.

Example



[Program description]

At the rising edge of R0, a pulse train of 80,000 Hz is output.

Name	Pulse speed control start

Ladder format	Number of steps		Condition code				
	Condition	Step	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
PLSPDR (s)	–	3	↕	●	●	●	●

Instruction processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	124	172	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	Control parameter first I/O									✓					

Remarks
This instruction is supported in Ver. x120 or later. s parameters are occupied up to s+5.

Function

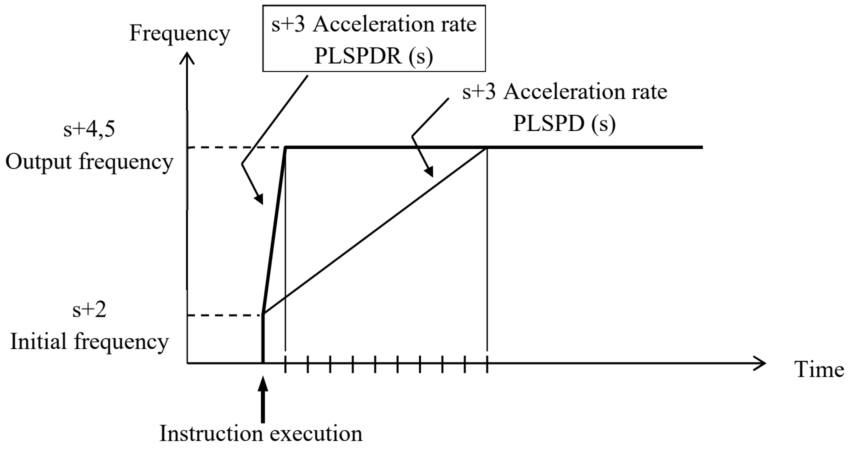
- This instruction specifies ten times the acceleration/deceleration rate of the PLSPD (s) (pulse speed control start) instruction.
  Other settings are the same as PLSPD (s).

Parameter

s: Specify the first I/O of the control parameter table.

	Setting description	R/W	Setting range, setting description
s+0	Output No.	W	1 to 3
s+1	Rotation direction	W	0: Forward, 1: Reverse
s+2	Initial frequency	W	20 to 65,535 (Hz)
s+3	Acceleration rate	W	1 to 65,535 (Hz / 10ms)
s+4	Output frequency (low word)	W	20 to 100,000 (Hz)
s+5	Output frequency (high word)	W	

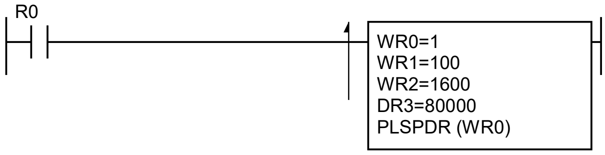
- To enable/disable the direction signal, use the Control Editor.
- If the direction signal is disabled, [s+1] Rotation direction is ignored.



Cautionary notes

This instruction has the same specifications as those of PLSPD (s) except for the acceleration rate setting. Even the precautions are the same as PLSPD (s), so see the PLSPD (s) section for the precautions.

Example



Output No. 1 (Y100)
Initial frequency: 100 Hz
Acceleration rate: 1,600 Hz/10 ms
Output frequency: 80,000 Hz
Speed control instruction

[Program description]

At the rising edge of R0, a pulse train of 80,000 Hz is output.

Name	Pulse speed change

Ladder format	Number of steps		Condition code				
	Condition	Step	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
PLCNG (s)	–	3	↕	●	●	●	●

Instruction processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	112	156	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	Control parameter first I/O									✓					

Remarks
s parameters are occupied up to s+3.

Function

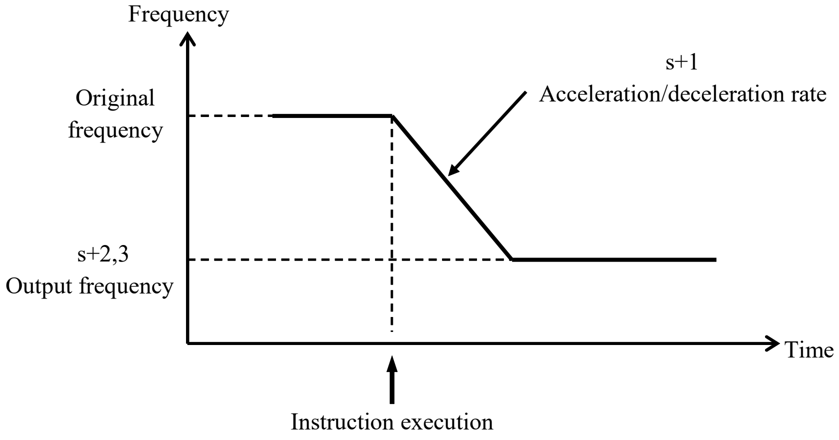
- This instruction changes the output frequency of the specified output No.
- This is valid only during pulse output with the PLSPD instruction (pulse speed control start).

Parameter

s: Specify the first I/O of the control parameter table.

	Setting description	R/W	Setting range, setting description
s+0	Output No.	W	1 to 3
s+1	Acceleration/deceleration rate	W	1 to 65,535 (Hz/100 ms)
s+2	Output frequency (low word)	W	20 to 100,000 (Hz)
s+3	Output frequency (high word)	W	

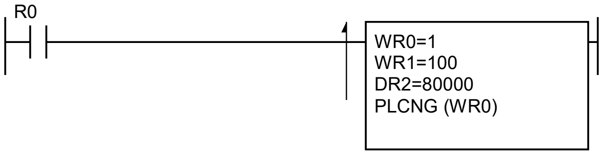
Combination with the pulse speed control start instruction (PLSPD)



Cautionary notes

- s+3 must be within the I/O range. You cannot write a parameter that is outside the I/O range.
- If you try to execute this instruction when an invalid value is set to s+0, DER is set to 1, which does not execute the instruction.
- If counter input is not configured, this instruction is not executed with DER = 1.
- If you try to execute this instruction during pulse output with the PLSTA instruction (pulse output start), DER is set to 1, which does not execute the instruction.
- The minimum output frequency is 20 Hz. Even if less than 20 Hz is set, it is treated as 20 Hz.
- The maximum output frequency is 100,000 Hz.
- This instruction cannot be accepted during acceleration/deceleration.

Example



Output No. 1 (Y100)
Acceleration/deceleration rate: 100 Hz/100 ms
Output frequency: 80,000 Hz
Frequency change instruction

[Program description]

At the rising edge of R0, the output frequency of output No. 1 is changed to 80,000 Hz.

Name	Pulse speed change

Ladder format	Number of steps		Condition code				
	Condition	Step	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
PLCNGR (s)	–	3	↕	●	●	●	●

Instruction processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	112	156	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	Control parameter first I/O									✓					

Remarks
This instruction is supported in Ver. x120 or later. s parameters are occupied up to s+3.

Function

- This instruction specifies ten times the acceleration/deceleration rate of the PLCNG (s) (pulse speed control frequency change).

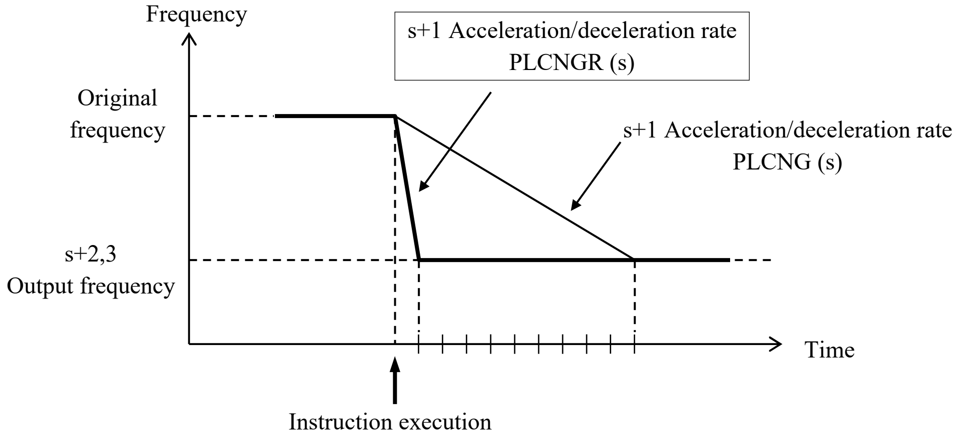
  Other settings are the same as PLCNG (s).

Parameter

s: Specify the first I/O of the control parameter table.

	Setting description	R/W	Setting range, setting description
s+0	Output No.	W	1 to 3
s+1	Acceleration/deceleration rate	W	1 to 65,535 (Hz/10 ms)
s+2	Output frequency (low word)	W	20 to 100,000 (Hz)
s+3	Output frequency (high word)	W	

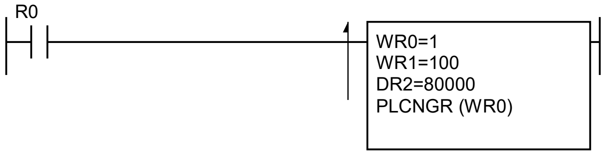
Differences from PLCNG (pulse speed control frequency change)



Cautionary notes

This instruction has the same specifications as those of PLCNG (s) except for the acceleration/deceleration rate setting. Even the precautions are the same as PLCNG (s), so see the PLCNG (s) section for the precautions.

Example



```
WR0=1
WR1=100
DR2=80000
PLCNGR (WR0)
```

Output No. 1 (Y100)
Acceleration/deceleration rate: 100 Hz/10 ms
Output frequency: 80,000 Hz
Frequency change instruction

[Program description]

At the rising edge of R0, the output frequency of output No. 1 is changed to 80,000 Hz.

Name	Pulse output stop

Ladder format	Number of steps		Condition code				
	Condition	Step	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
PLSTP (s)	–	3	↕	●	●	●	●

Instruction processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	116	160	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT, TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	Control parameter first I/O								✓					

Remarks
s parameters are occupied up to s+2.

Function

- This instruction stops the pulse output of the specified output No. (There are two stop patterns available. You can set the pattern in operation mode.)
- This is valid only during pulse output with the acceleration/deceleration pulse output instruction (PLSTA) or pulse speed control start instruction (PLSPD).
- If the pulse output is stopped in the deceleration stop mode, it decelerates and stops according to the deceleration rate specified with this instruction by ignoring the number of output pulses and the acceleration/deceleration rate specified with the acceleration/deceleration pulse output instruction (PLSTA).
- In case of the emergency stop mode, the output stops immediately after this instruction is executed.

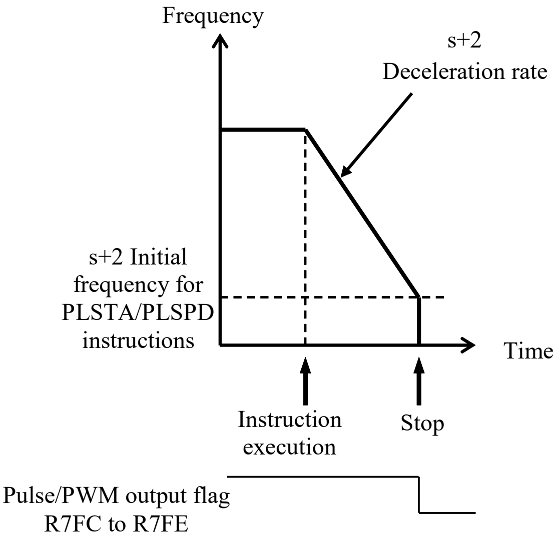
Parameter

s: Specify the first I/O of the control parameter table.

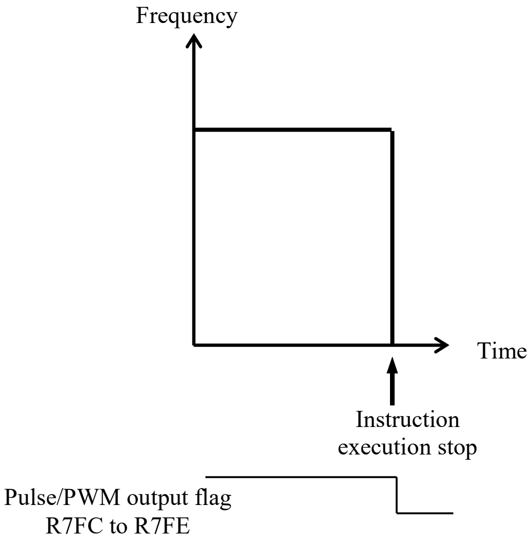
	Setting description	R/W	Setting range, setting description
s+0	Output No.	W	1 to 3
s+1	Operation mode	W	0: Deceleration and stop, 1: Emergency stop
s+2	Deceleration rate	W	0 to 65,535 (Hz/ 100ms) (Valid only for deceleration and stop)

[s+1] Operation mode

<For 0: Deceleration and stop>



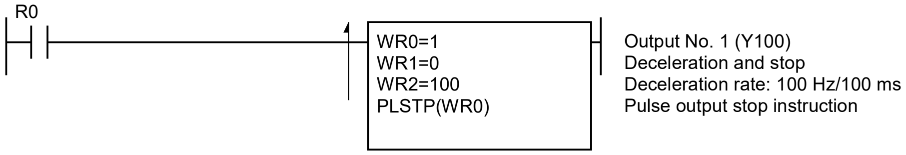
<For 1: Emergency stop>



Cautionary notes

- s+2 must be within the I/O range. You cannot write a parameter that is outside the I/O range.
- If you try to execute this instruction when invalid values are set in s+0 and s+1, DER is set to 1, which does not execute the instruction.
- If counter input is not configured, this instruction is not executed with DER = 1.
- 0: Do not accept deceleration and stop during acceleration/deceleration. 1: Accept emergency stop even during acceleration/deceleration.

Example



[Program description]

At the rising edge of R0, output No. 1, which is outputting pulses, is stopped at the deceleration rate of 100 Hz/100 ms.

Name	Pulse output stop

Ladder format	Number of steps		Condition code				
	Condition	Step	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
PLSTPR (s)	–	3	↕	●	●	●	●

Instruction processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	116	160	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	Control parameter first I/O								✓					

Remarks
This instruction is supported in Ver. x120 or later. s parameters are occupied up to s+2.

Function

- This instruction specifies ten times the deceleration rate of the PLSTP (s) (pulse output stop) instruction. Other settings are the same as PLSTP (s).

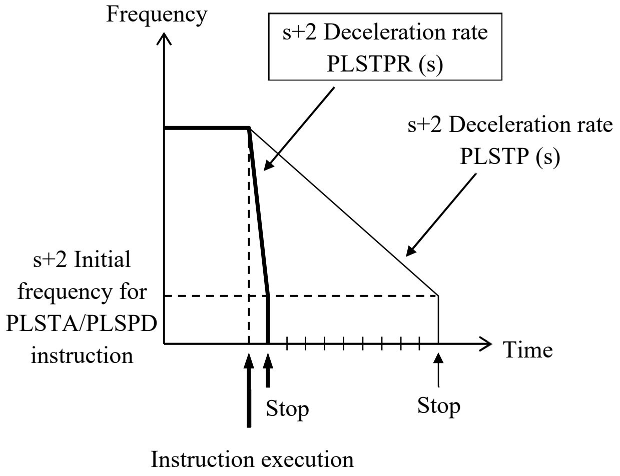
Parameter

s: Specify the first I/O of the control parameter table.

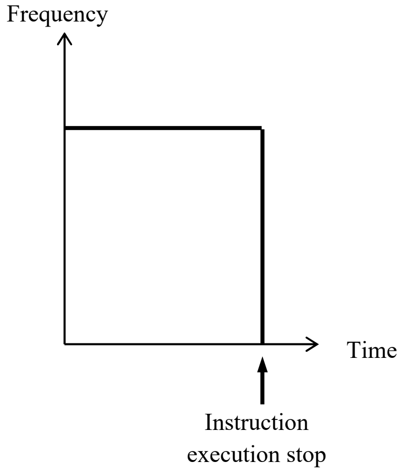
	Setting description	R/W	Setting range, setting description
s+0	Output No.	W	1 to 3
s+1	Operation mode	W	0: Deceleration and stop, 1: Emergency stop
s+2	Deceleration rate	W	0 to 65,535 (Hz/10ms) (Valid only for deceleration and stop)

[s+1] Operation mode

<For 0: Deceleration and stop>
Ten times the deceleration rate than PLSTP (s)



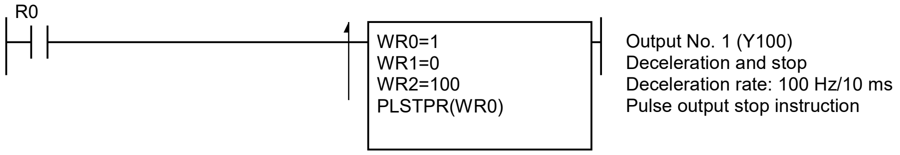
<For 1: Emergency stop>
Same as PLSTP (s)



Cautionary notes

This instruction has the same specifications as those of PLSTP (s) except for the deceleration rate setting. Even the precautions are the same as PLSTP (s), so see the PLSTP (s) section for the precautions.

Example



[Program description]

At the rising edge of R0, output No. 1, which is outputting pulses, is stopped at the deceleration rate of 100 Hz/10 ms.

Name	Homing return

Ladder format	Number of steps		Condition code				
	Condition	Step	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
PLHM (s1, s2)	–	4	↕	●	●	●	●

Instruction processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	61	95	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s1	Control channel number								✓					✓
s2	Homing direction			✓										✓

Remarks
This instruction is supported in Ver.x120 or later.

Function

- This instruction moves the axis connected to the specified channel to the home position.
- When the axis finishes moving to the home position, the position of the channel is set to 0.
- There are four homing return modes available. When this instruction is executed, the homing return mode specified in the pulse parameters is applied.

  For details on the homing return modes, see "Chapter 5 I/O Assignment and Special I/O" in the User's Manual.
- The completion of homing return can be checked with the bit special internal output.

Parameter

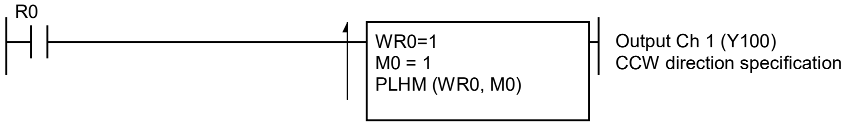
s1: Specify the channel to homing return. (1 to 3)

s2: Specify the homing return direction. (0: CW direction, 1: CCW direction)

Cautionary notes

- If counter input is not configured, this instruction is not executed with DER = 1.
- If you try to execute this instruction when the specified channel is outputting pulses, DER is set to 1, which does not execute the instruction.
- When the CPU stops, the homing return stops.
- Any other homing return mode than the arbitrary homing return outputs pulses while monitoring the states of input signals (home limit switch and marker). If the input signal does not change due to misfiring, the homing return is not finished (the pulse output is not stopped). If there is a physical upper or lower limit, prepare a fail-safe so that the pulse output is stopped before the upper or lower limit is reached.

Example



[Program description]

At the rising edge of R0, channel 1 is homing return in the CCW direction.

Name	Pulse position data read

Ladder format	Number of steps		Condition code				
	Condition	Step	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
PLSRD (s)	–	2	↕	●	●	●	●

Instruction processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	6.32	7.66	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	Control parameter first I/O									✓					

Remarks
s parameters are occupied up to s+2.

Function

- This instruction reads the current position data of the specified pulse output No. out to the specified area.

Parameter

s: Specify the first I/O of the control parameter table.

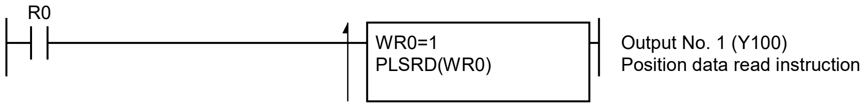
	Setting description	R/W	Setting range, setting description
s+0	Output No.	W	1 to 3
s+1	Current position data (low word)	R	0 to 4,294,967,295 : When the direction signal is disabled -2,147,483,648 to 2,147,483,647 : When the direction signal is enabled
s+2	Current position data (high word)	R	

* To enable/disable the direction signal, use the Control Editor.

Cautionary notes

- s+2 must be within the I/O range. You cannot write a parameter that is outside the I/O range.
- If you try to execute this instruction when an invalid value is set in s+0, DER is set to 1, which does not execute the instruction.
- If counter input is not configured, this instruction is not executed with DER = 1.
- The current position data is cleared at power-on. It is not cleared by RUN/STOP.
- To clear the current position data during operation, use the PLSWR (s) instruction.

Example



[Program description]

At the rising edge of R0, the current position data of output No. 1 is read and stored into DR1.

Name	Pulse position data write

Ladder format	Number of steps		Condition code				
	Condition	Step	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
PLSWR (s)	–	2	↕	●	●	●	●

Instruction processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	6.22	7.66	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	Control parameter first I/O								✓					

Remarks
s parameters are occupied up to s+2.

Function

- This instruction overwrites the current position data of the specified pulse output No. with the specified value.

Parameter

s: Specify the first I/O of the control parameter table.

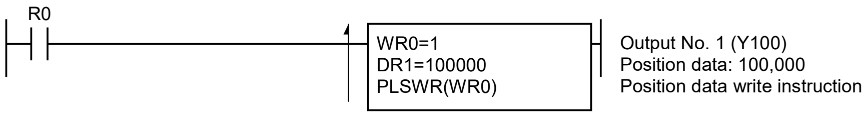
	Setting description	R/W	Setting range, setting description
s+0	Output No.	W	1 to 3
s+1	Current position data (low word)	W	0 to 4,294,967,295 : When the direction signal is disabled -2,147,483,648 to 2,147,483,647 : When the direction signal is enabled
s+2	Current position data (high word)	W	

* To enable/disable the direction signal, use the Control Editor.

Cautionary notes

- s+2 must be within the I/O range. You cannot write a parameter that is outside the I/O range.
- If you try to execute this instruction when an invalid value is set in s+0, DER is set to 1, which does not execute the instruction.
- If counter input is not configured, this instruction is not executed with DER = 1.

Example



```
WR0=1
DR1=100000
PLSWR(WR0)
```

Output No. 1 (Y100)
Position data: 100,000
Position data write instruction

[Program description]

At the rising edge of R0, the current position of output No. 1 is overwritten with 100,000.

MEMO

[1] Basic commands

[2] Arithmetic commands

[3] Application commands

[4] Control commands

[5] CPU communication commands

Name	Normal scan END

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
END	—	2	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	0.01	0.01	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
—	(No argument)													

Remarks

Function

- The termination of the normal scan program is indicated. This command execution can execute the normal scan by returning to the top of the normal scan program.
- If there are neither the subroutine program nor the interrupt scan program, this command is unnecessary.
- If there are both the subroutine program and the interrupt scam program, this command is written at the end of the normal scan.
- This command can be used on the program only once. Do not put in the startup condition.

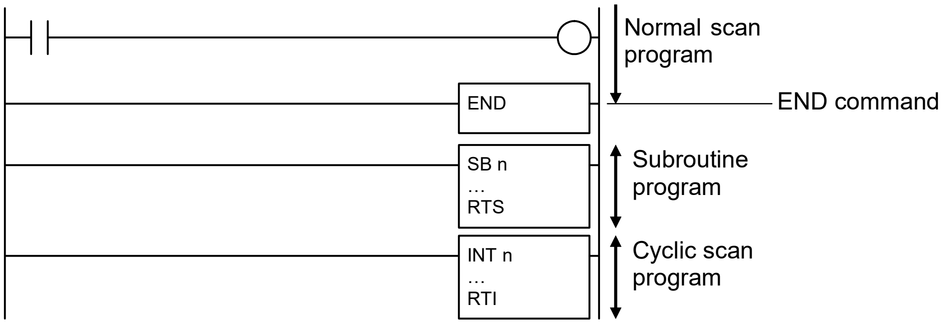
Cautionary notes

Unlike MICRO-EH currently in use, the editor detects the assembling error. Note because the program including the cause of the assembling error cannot be written into MICRO-EHV.

[Reference: Causes of the assembling error]

No.	Description of error
1	There is no END command.
2	There are two END commands or more.
3	The startup condition is in END command.

Program example



Name	Conditional END of scan

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
CEND (s)	–	3	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
Incompletion	0.6	0.6			
Completion	0.6	0.6	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	Condition for scan terminating	✓	✓	✓											

Remarks

Function

- When the condition for normal scan terminating (s) is ON, this command execution can execute the normal scan by returning to the top of the normal scan program.
- When s is OFF, the next command is executed.
- This command can be used only on the normal scan program and used many times.
- This command can set the startup condition. In this case, this command is executed when both s and the condition are ON.

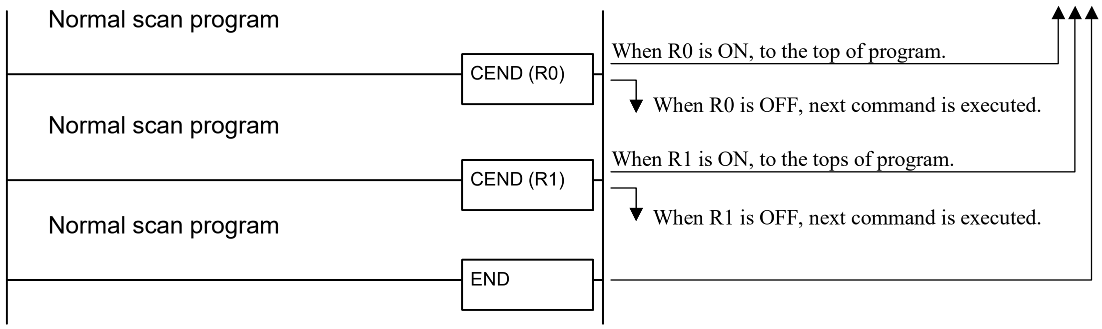
Cautionary notes

- Unlike MICRO-EH currently in use, the editor detects the assembling error. Note because the program including the cause of the assembling error is written into MICRO-EHV.

[Reference: Cause of the assembling error]

No.	Description of error
1	CEND command is behind END command.

Program example



Name	Unconditional jump

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
JMP n	—	3	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	5.88	5.88	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT, TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
n	Code No.													✓

Remarks
Code No. 0 to 511 can be used.(Decimal system)

Function

- When the startup condition of JMP n turns ON, the program jumps from this command to LBL n of the same code No. Please use JMP n and LBL n always in pairs.
- When the startup condition is incomplete, the next command is executed.
- When this command is put into the arithmetic box simultaneously with other command, please put this command at the end of the box.
- JMP n command is valid on only the same scan program. (Jumping from the normal scan to the subroutine or the interrupt scan is impossible and the opposite jumping is also impossible.)
- Although nesting of JMP n command is possible, please pay attention so that a jam error does not occur.

Cautionary notes

- If there is a timer in the jumped program, the progress value is updated but the command is not executed. Therefore, the output does not turn ON even if the conditions for ON are fulfilled.
- Unlike MICRO-EH currently in use, the editor detects the assembling error. Note because the program including the cause of the assembling error cannot be written into MICRO-EHV.

[Reference: Causes of assembling error]

No.	Description of error
1	There is no LBL n.
2	It is going to jump to other program areas.

Program example

Jump to LBL n when the startup condition turns ON.

Name	Conditional jump

Ladder format	Number of steps		Condition code				
CJMP n (s)	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
	–	4	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
Incompletion	2.29	2.29			
Completion	0.88	0.88	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
n	Code No.														✓
s	Condition for jump	✓	✓	✓											

Remarks
Code No. 0 to 511 can be used.(Decimal system)

Function

- When the condition for jump s of CJMP n (s) turns ON, the program jumps from this command to LBL n of the same code No. Please use CJMP n (s) and LBL n always in pairs.
- When the startup condition and the condition for jump are incomplete, the next command is executed.
- When this command is put in an arithmetic box simultaneously with other commands, please pay attention because the program jumps without performing remaining operations in the box if conditions are complete.
- CJMP n (s) command is valid on only the same scan program. (Jumping from the normal scan to the subroutine or the interrupt scan is impossible, and the opposite jumping is also impossible.
- Although nesting of CJMP n (s) command is possible, please pay attension so that a jam error does not occur.

Cautionary notes

- If there is a timer in the jumped program, the progress value is updated but the command is not executed. Therefore, the output does not turn ON even if the conditions for ON are fulfilled.
- Unlike MICRO-EH currently in use, the editor detects the assembling error. Note because the program including the cause of the assembling error cannot be written into MICRO-EHV.

[Reference: Causes of assembling error]

No.	Description of error
1	There is no LBL n.
2	It is going to jump to other program areas.

Program example

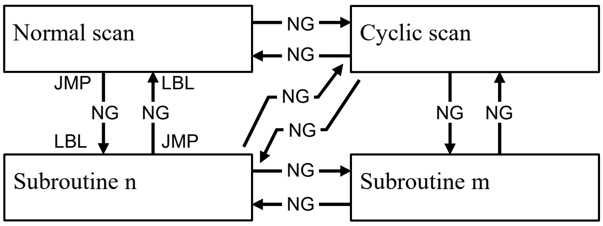
Jump to LBL n when the startup condition and the condition for jump are both ON.

In the left example, jump to LBL n when X0 and R0 are both ON.

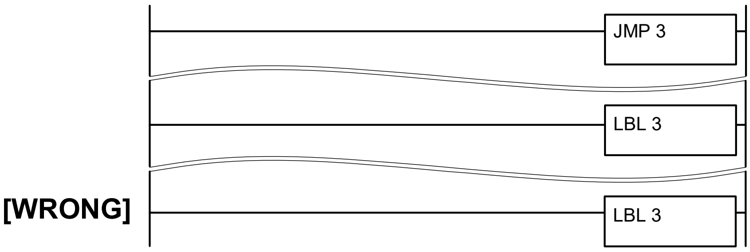
Reference Grammar of JMP and CJMP

[1] LBL n of the same code No. as the code No. of JMP is needed.

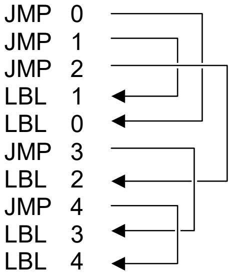
[2] Jumping to the area other than area which has JMP is impossible.



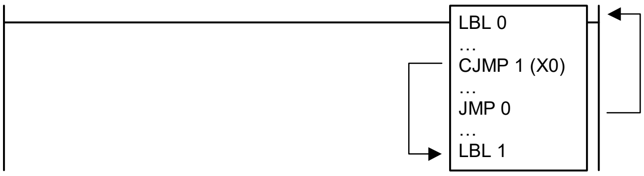
[3] LBL n of the same code No. as the code No. of JMP must not overlap.



[4] Nesting of JMP is possible.

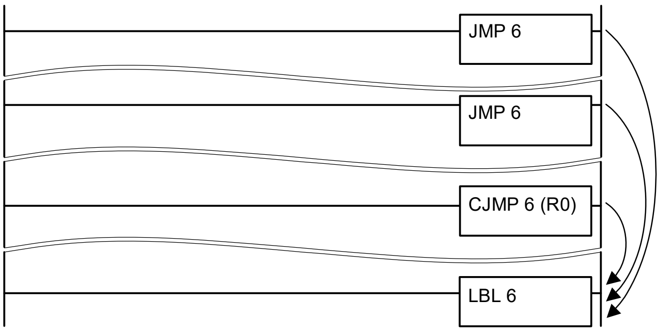


[5] JMP can jump forward of this command.



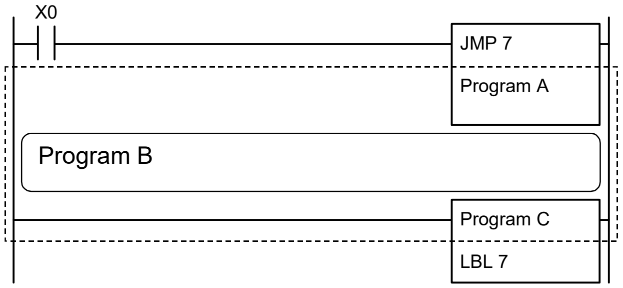
- JMP 0 jumps to the forward LBL 0.
- If the input X0 turns ON, jumping from CJMP 1 to LBL 1 can get out of the loop from JMP 0 to LBL 0.
- If there is no command to get out of the loop like CJMP 1, the loop from JMP0 to LBL 0 repeats limitlessly.

[6] JMP of the same code No. can be repeated.



[7] The startup condition can be programmed to JMP command.

Startup condition

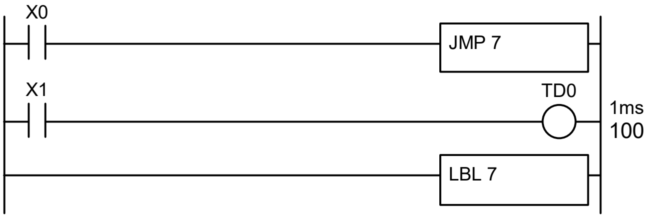


If jumping from JMP 7 to LBL 7, the program A, B, and C are not executed.

[8] CJMP also obeys the grammar from [1] to [7].

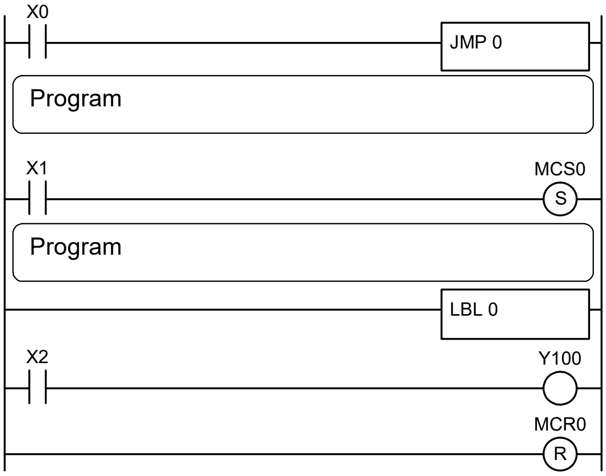
Cautionary notes

- MICRO-EHV updates the progress value at the command execution of the timer. The timer may not turn ON correctly if the program not to scan a portion to execute the timer command is created after the timer is started up.



If X0 turns ON after X1 turns ON, the progress value of TD0 is updated even if jumping from JMP7 to LBL7. If X0 is keeping ON, TD0 does not turn ON even if the progress value of TD0 exceeds 100.

- Please program with great care because the action is as follows if using by combining JMP with MCS and MCR.



When not jumping in JMP 0, Y100 turns ON when X1 is ON and X2 is ON.

When jumping in JMP 0, Y100 turns ON if X2 is ON.

- Do not create the circuit to jump out from between MCS and MCR.

Name	Label

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
LBL n	–	1	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	0.13	0.13	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
n	Code No.													✓

Remarks
Code No. 0 to 511 can be used.(Decimal system)

Function

- This command indicates where to jump when JMP n and CJMP n are executed. (n is always used in pairs.)
- n in LBL n cannot be used on the same program repeatedly.
- Nothing is performed by this command itself.
- Even if putting the startup condition in LBL n, it is ignored.

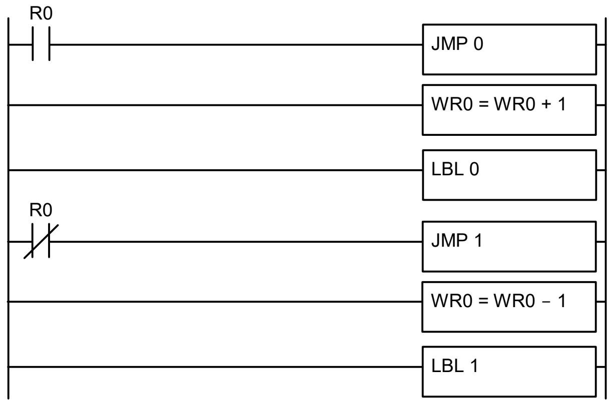
Cautionary notes

Unlike MICRO-EH currently in use, the editor detects the assembling error. Note because the program including the cause of the assembling error cannot be written into MICRO-EHV.

[Reference: Cause of assembling error]

No.	Description of error
1	There is LBL of the same No.

Program example



[Program description]

- When R0 is ON, JMP 0 is executed but JMP 1 is not executed. Therefore, the content of WR0 decreases one by one at every scan.
- When R0 is OFF, JMP 0 is not executed but JMP 1 is executed. Therefore, the content of WR0 increases one by one at every scan.

Name	FOR

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
FOR n (s)	−	4	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
−	7.33	7.33	−	−	−

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
n	Code No.													✓
s	Number of repeating times							✓	✓					

Remarks
Code No.0 to 99 can be used.(Decimal system)

Function

- It jumps from NEXT n of the same code No. to this command.
- When s>0 (s is the number of repeating times), the next command of FOR n (s) is executed.
- When s = 0 (s is the number of repeating times), it jumps to the next command of NEXT n.
- Please use FOR n (s) and NEXT n always in pairs. And please put NEXT n in the back of FOR n.
- FOR n (s) cannot be used repeatedly.
- Please use FOR n (s) and NEXT n in the same program area.
  (FOR n (s) cannot be programmed on the normal scan and NEXT n cannot be programmed on the subroutine area.)
- The nesting from FOR n (s) to NEXT n can be layered up to 5.

Cautionary notes

Unlike MICRO-EH currently in use, the editor detects the assembling error. Note because the program including causes of the assembling error cannot be written into MICRO-EHV.

[Reference: Causes of assembling error]

No.	Description of error
1	There is FOR of the same No.
2	NEXT of No. corresponding has not been defined.
3	NEXT is before FOR.
4	Area error of NEXT
5	Nesting error from FOR to NEXT
6	FOR nesting over flow

Program example

Refer to the description pages of “NEXT n”.

Name	NEXT

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
NEXT n	—	3	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	2.19	2.19	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
n	Code No.														✓

Remarks
Code No.0 to 99 can be used. (Decimal system)

Function

It subtracts 1 from the number of repeating times s of FOR n (s) of the same No., and then it jumpes to FOR n (s).

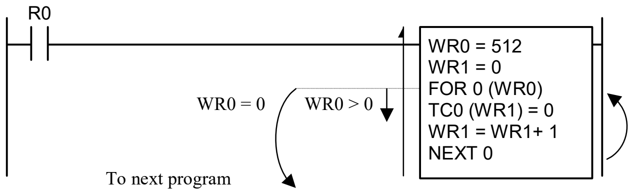
Cautionary notes

Unlike MICRO-EH currently in use, the editor detects the assembling error. Note because the program including causes of the assembling error cannot be written into MICRO-EHV.

[Reference: Causes of assembling error]

No.	Description of error
1	There is NEXT of the same No.
2	FOR of No. corresponding has not been defined.
3	FOR nesting over flow

Program example



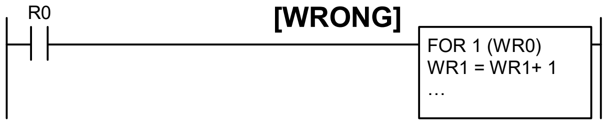
```
WR0 = 512
WR1 = 0
FOR 0 (WR0)
TC0 (WR1) = 0
WR1 = WR1+ 1
NEXT 0
```

[Program description]

- If R0 is turned ON, all data in 512 points of the progress value (TC n) of the timer counter is cleared to 0.
- FOR to NEXT keeps executing the command until s becomes 0, once it starts up.
- FOR 0 (WR0) executes commands following TC0(WR1) = 0 durign WR > 0, and it jumps to FOR 0 (WR0) after subtracting 1 from WR0 in NEXT 0.
- FOR 0 (WR0) jumps to the next command to this box when WR0 = 0.

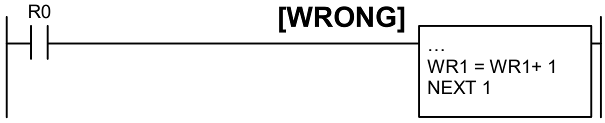
Reference Grammar of FOR and NEXT

[1] NEXT of the same code No. as the code No. of FOR is needed after FOR.



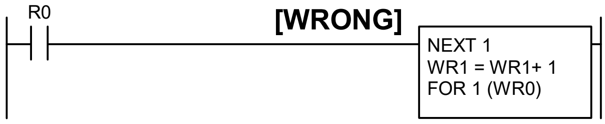
```
FOR 1 (WR0)
WR1 = WR1+ 1
…
```

There is not NEXT command to FOR command on the user program.



```
…
WR1 = WR1+ 1
NEXT 1
```

There is no FOR command before NEXT command.

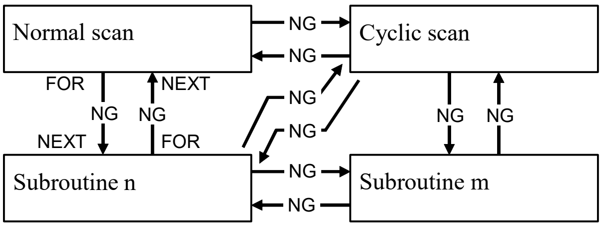


```
NEXT 1
WR1 = WR1+ 1
FOR 1 (WR0)
```

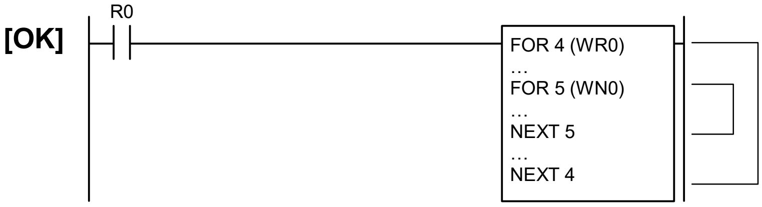
There is NEXT command before FOR command.

[2] FOR and NEXT of the same code No. cannot be repeated.

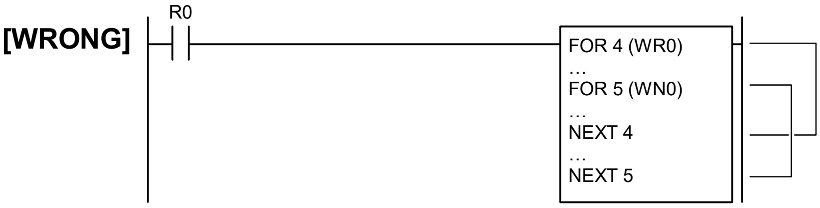
[3] FOR and NEXT have to be in the same area.



[4] Please set the nesting structure of FOR to NEXT.

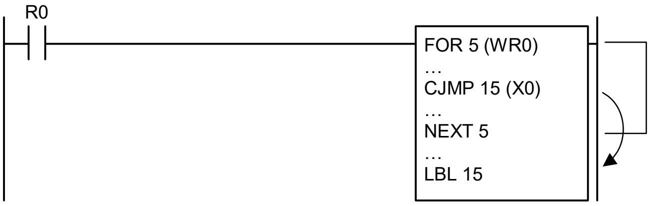


```
FOR 4 (WR0)
…
FOR 5 (WN0)
…
NEXT 5
…
NEXT 4
```



```
FOR 4 (WR0)
…
FOR 5 (WN0)
…
NEXT 4
…
NEXT 5
```

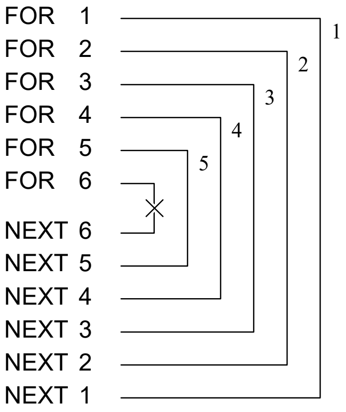
[5] It is possible to get out of a loop of FOR to NEXT by the jump command.



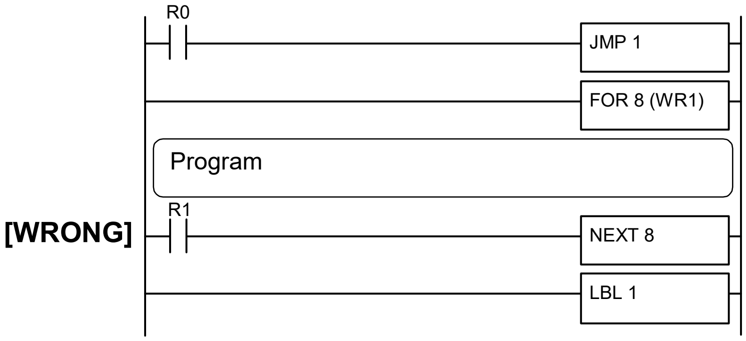
```
FOR 5 (WR0)
…
CJMP 15 (X0)
…
NEXT 5
…
LBL 15
```

If X0 turns ON without performing the loop of FOR 5 to NEXT 5 the number of times of repetition (the content of WR0), it gets out of the loop.

[6] The nesting from FOR to NEXT can be layered up to 5. If it contains the subroutine, FOR to NEXT in the subroutine is counted.



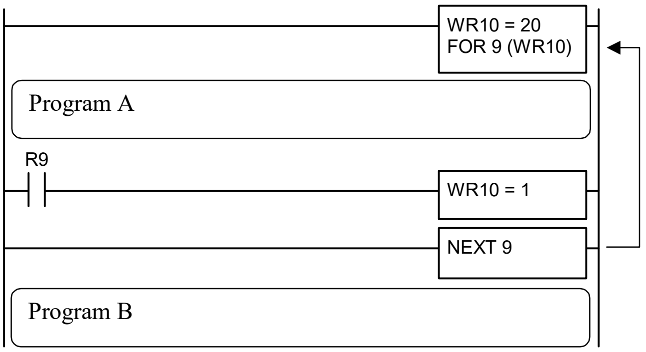
[7] Do not put the startup condition in NEXT.



When R0 is OFF, … Executes the program for the value of WR1.

When R0 is ON, … The program is not executed because of jumping from JMP 1 to LBL 1.

[9] The number of times of repetition on the program can be changed.



Jump to FOR 9 (WR10) after subtracting 1 from the content of WR10.

When R9 is OFF, … Executes the program B after repeating the program A 20 times.

When R9 is ON, … The content of WR10 becomes 0 because the number of times of repetition WR10 becomes 1 and 1 is subtracted on the processing of NEXT 9. Therefore, the program B is executed after the repetition of the program A terminates.

Name	Subroutine call

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
CAL n	–	3	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	1.4	1.4	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
n	Code No.														✓

Remarks
Code No.0 to 199 can be used.(Decimal system)

Function

- When the startup condition of CAL n is ON, the subroutine program of the same code No. in this command (the program surrounded by SB n to RTS) are executed.
- When the startup condition is OFF, the next program is executed.
- CAL (nesting) of another subroutine can be layered up to 5 in a subroutine.
- The subroutine can be called in the interrupt scan program.

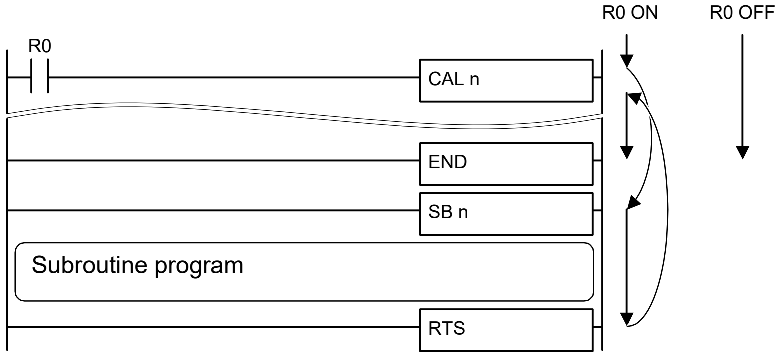
Cautionary notes

Unlike MICRO-EH currently in use, the editor detects the assembling error. Note because the program including causes of the assembling error cannot be written into MICRO-EHV.

[Reference: Causes of assembling error]

No.	Description of error
1	SB corresponding has not been defined.
2	Nesting error

Program example



[Program description]

- When R0 is ON, the subroutine program is executed at CAL n. The next program to CAL n is re-executed after execution
- When R0 is OFF, the next program is executed without executing the subroutine program.

Name	START Subroutine

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
SB n	–	1	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	0.12	0.12	–	–	–

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
n	Code No.														✓

Remarks
Code No. 0 to 199 can be used.(Decimal system)

Function

- This command means the start of the subroutine program. (No processing)
- n for SB n cannot be used repeatedly in the same program.
- It is ignored even if the startup condition is put in SB n.
- Please use SB n and RTS always in pairs.
- Please write the subroutine program from SB n to RTS to the sheet for the subroutine or behind the END command.

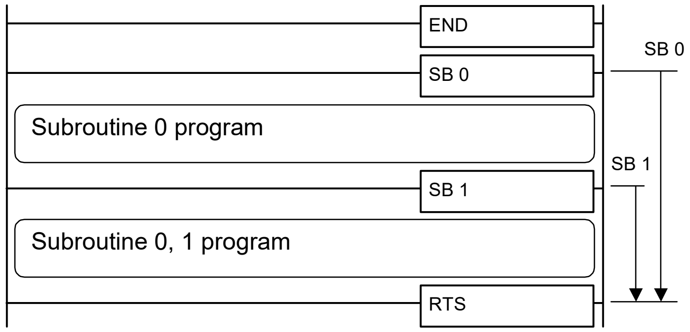
Cautionary notes

Unlike MICRO-EH currently in use, the editor detects the assembling error. Note because the program including causes of the assembling error cannot be written into MICRO-EHV.

[Reference: Causes of assembling error]

No.	Description of error
1	There is SB of the same No.
2	CAL corresponding has not been defined.

Program example



[Program description]

- If CAL 0 is executed, the program from SB 0 to RTS is executed as the subroutine.
- If CAL 1 is executed, the program from SB 1 to RTS is executed as the subroutine.

Name	RETURN from subroutine

Ladder format	Number of steps		Condition code				
RTS	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
	—	2	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	2.3	2.3	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
—	(No argument)													

Remarks

Function

- This command declares the termination of the subroutine program.
- If this command is executed, the program is executed from the next to CAL n calling the subroutine.

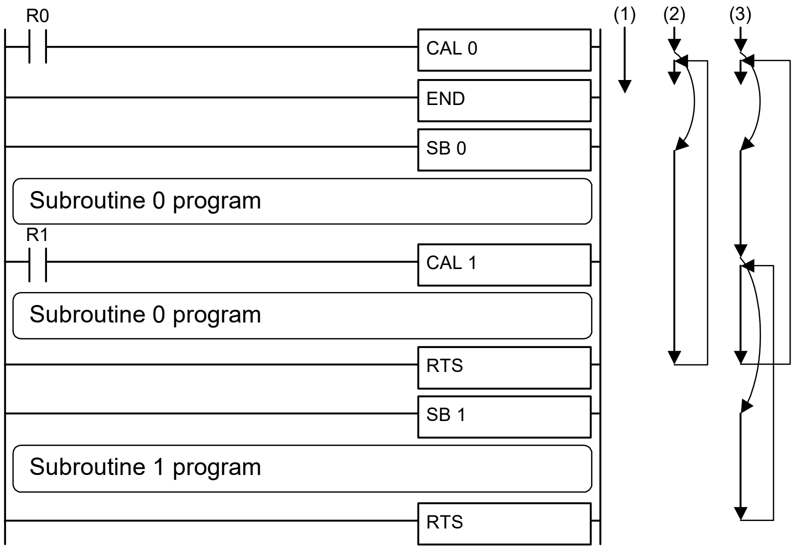
Cautionary notes

- Do not put the startup condition in this command.
- Unlike MICRO-EH currently in use, the editor detects the assembling error. Note because the program including causes of the assembling error cannot be written into MICRO-EHV.

[Reference: Causes of assembling error]

No.	Description of error
1	There are several RTS.
2	Area error of RTS
3	Startup condition error of RTS

Program example



[Program description]

If both R0 and R1 are OFF, it is executed like (1), if only R0 is ON, it is executed like (2), and if both R0 and R1 are ON, it is executed like (3).

Name	START cyclic scan

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
INT (s)	–	1	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	—	—	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	Cycle (ms)														✓

Remarks
It can be used up to 4. Cyclic can be specified from 5 to 60,000 [ms].

Function

- This command declares the start of the cyclic interrupt scan program.
- The cycle (units: ms) of the cyclic interrupt scan is specified of s.
- The shorter the cycle is, the higher the order of priority of the interrupt is.
- Please use INT (s) and RTI always in pairs.
- INT (s) is ignored even if the startup condition is put in.
- Please write the interrupt program from INT (s) to RTI to the sheet for the subroutine or behind the END command.

Cautionary notes

- The same cycle cannot be used repeatedly.
- The progress value is updated at the execution of the timer command in MICRO-EHV. Therefore, the timer may not turn ON correctly if the program not to scan a portion to execute the timer command is created using the interrupt scan. (The timer does not turn ON if the time not to scan the portion to execute the timer command exceeds time base×65,535). And note because the previous progress value is retained until the timer command is executed.
- Unlike MICRO-EH currently in use, the editor detects the assembling error. Note because the program including causes of the assembling error cannot be written into MICRO-EHV.

[Reference: Causes of assembling error]

No.	Description of error
1	There is INT of the same No. (There are several INT of the same cycle.)
2	INT has not been defined.

Program example

Refer to the description pages for ”RTI”.

PRN ➔ PRJ

This command corresponds to INT0 to INT2 in the MICRO-EH program (PRN file).

INT 0 ➔ INT (10)

INT 1 ➔ INT (20)

INT 2 ➔ INT (40)

* If converted by a conversion tool, it is converted as mentioned above.

Name	RETURN from cyclic scan

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
RTI	–	2	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	—	—	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
—	(No argument)														

Remarks

Function

- This command declares the termination of the cyclic interrupt scan program.
- This command execution returns the processing to the executing program before executing the cyclic interrupt scan.

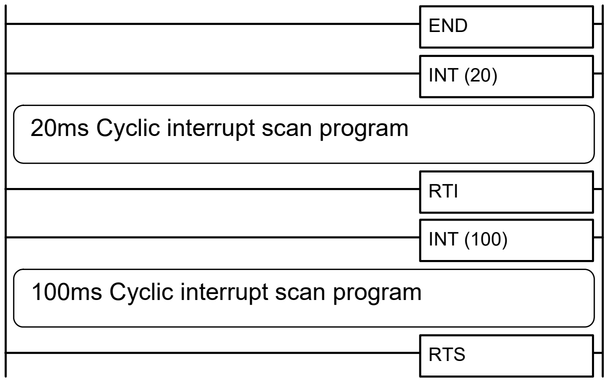
Cautionary notes

- Please put the startup condition in this command.
- Unlike MICRO-EH currently in use, the editor detects the assembling error. Note because the program including causes of the assembling error cannot be written into MICRO-EHV.

[Reference: Causes of assembling error]

No.	Description of error
1	RTI has not been defined.
2	Area error of RTI
3	Startup condition error of RTI

Program example



[Program description]

- Executes the program from INT (20) to RTI every 20 ms after RUN.
- Executes the program from INT (100) to RTI every 100 ms after RUN.

Name	START interrupt scan

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
XINT (n)	–	1	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	—	—	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
n	Input I/O number														✓

Remarks
Up to 4 can be used

Function

- This instruction declares the start of the input interrupt scan program.
- Specify the corresponding input I/O number (1, 3, 5, 7, 9) for n.

Interrupt label	Start condition
XINT (1)	Turn on the interrupt input X1
XINT (3)	Turn on the interrupt input X3
XINT (5)	Turn on the interrupt input X5
XINT (7)	Turn on the interrupt input X7
XINT (9)	Turn on the interrupt input X9

- Always use XINT (n) in combination with XRTI.
- XINT (n) is ignored even if a start condition is entered.
- If there are inputs to the input terminals assigned to interrupt inputs while the CPU is running, the corresponding interrupt scan program is started.
- Write the XINT (n) to XRTI interrupt scan program on the subroutine sheet or after the END instruction.
- The smaller the n number, the higher the interrupt priority.

Cautionary notes

- n cannot be used more than once in the same program.
- Unlike MICRO-EH currently in use, the editor detects the assembling error. Note because the program including causes of the assembling error cannot be written into MICRO-EHV.

[Reference: Causes of assembling error]

No.	Error contents
1	XINT with the same number exists
2	XINT not defined

Program example

Refer to the explanation page of “XRTI”.

Name	RETURN from interrupt scan

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
XRTI	—	2	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	—	—	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
—	(No argument)													

Remarks

Function

- This instruction declares the end of the interrupt scan program.
- When this command is executed, Processing is returned to the program that was being executed before the interrupt scan.

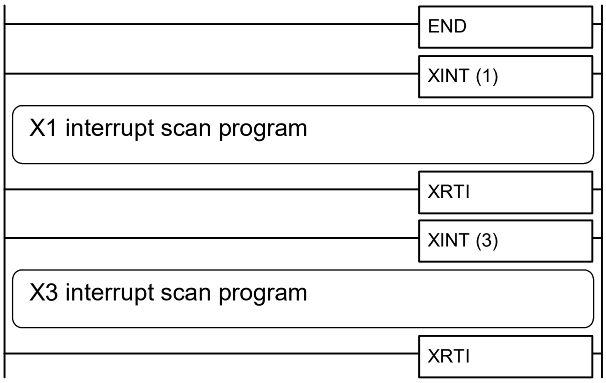
Cautionary notes

- Do not include a start condition in this command.
- Unlike MICRO-EH currently in use, the editor detects the assembling error. Note because the program including causes of the assembling error cannot be written into MICRO-EHV.

[Reference: Causes of assembling error]

No.	Error contents
1	XRTI not defined
2	XRTI area error
3	XRTI startup condition error

Program example



[Program description]

- After RUN starts, XINT (1) to XRTI programs are executed when interrupt input X1 turns on.
- After RUN starts, XINT (3) to XRTI programs are executed when interrupt input X3 turns on.

Name	START counter interrupt scan (1st comparison value)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
CINTP (n)	—	1	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	—	—	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
n	Input counter number														✓

Remarks
Up to 5 can be used

Function

- This instruction declares the start of the interrupt scan program for counter.
- For n, specify the corresponding counter number (1 to 5).

Interrupt label	Start condition
CINTP (1)	Counter 1 first comparison value match interrupt
CINTP (2)	Counter 2 first comparison value match interrupt
CINTP (3)	Counter 3 first comparison value match interrupt
CINTP (4)	Counter 4 first comparison value match interrupt
CINTP (5)	Counter 5 first comparison value match interrupt

- Always use CINTP (n) in combination with CRTIP.
- CINTP (n) is ignored even if a start condition is entered.
- When the CPU is running and there is an input at the input terminal assigned to the counter input and the curent counter value exceeds the first comparison value, the corresponding interrupt scan program is started. Interrupt scan by interrupt input calculates the interrupt scan program from CINTP (n) to CRTIP instruction.
- Write the CINTP (n) to CRTIP interrupt scan program on the subroutine sheet or after the END instruction.
- The smaller the n number, the higher the interrupt priority.

Cautionary notes

- n cannot be used more than once in the same program.
- Unlike MICRO-EH currently in use, the editor detects the assembling error. Note because the program including causes of the assembling error cannot be written into MICRO-EHV.

[Reference: Causes of assembling error]

No.	Error contents
1	CINTP with the same number exists
2	CINTP not defined

Program example

Refer to the explanation page of “CRTIP”.

Name	RETURN from counter interrupt scan (1st comparison value)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
CRTIP	—	2	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	—	—	—	—	—

Usable I/O		Bit						Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT	TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
—	(No argument)														

Remarks

Function

- This instruction declares the end of the interrupt scan program for counter.
- When this command is executed, Processing is returned to the program that was being executed before the interrupt scan.

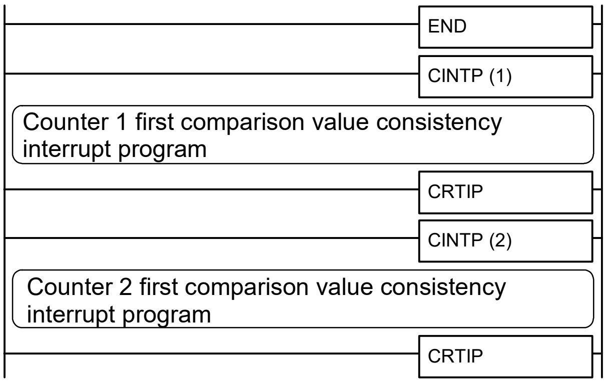
Cautionary notes

- Do not include a start condition in this command.
- Unlike MICRO-EH currently in use, the editor detects the assembling error. Note because the program including causes of the assembling error cannot be written into MICRO-EHV.

[Reference: Causes of assembling error]

No.	Error contents
1	CRTIP not defined
2	CRTIP area error
3	CRTIP startup condition error

Program example



[Program description]

- After the RUN starts, the CINTP (1) to CRTIP program is executed when the counter value of counter 1 reach the first comparison value.
- After the RUN starts, the CINTP(2) to CRTIP program is executed when the counter value of counter 2 reach the first comparison value.

Name	START counter interrupt scan (2nd comparison value)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
CINTN (n)	—	1	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	—	—	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
n	Input counter number													✓

Remarks
Up to 5 can be used

Function

- This instruction declares the start of an interrupt scan program for counter.
- For n, specify the corresponding counter number (1 to 5).

Interrupt label	Start condition
CINTN (1)	Counter 1 second comparison value match interrupt
CINTN (2)	Counter 2 second comparison value match interrupt
CINTN (3)	Counter 3 second comparison value match interrupt
CINTN (4)	Counter 4 second comparison value match interrupt
CINTN (5)	Counter 5 second comparison value match interrupt

- Always use CINTN (n) in combination with CRTIN.
- CINTN (n) is ignored even if a start condition is entered.
- When the CPU is running and there is an input at the input terminal assigned to the counter input and the curent counter value exceeds the second comparison value, the corresponding interrupt scan program is started. The interrupt scan by interrupt input calculates the interrupt scan program from CINTN (n) to the CRTIN instruction.
- Write the CINTN (n) to CRTIN interrupt scan program on the subroutine sheet or after the END instruction.
- The smaller the n number, the higher the interrupt priority.

Cautionary notes

- n cannot be used more than once in the same program.
- Unlike MICRO-EH currently in use, the editor detects the assembling error. Note because the program including causes of the assembling error cannot be written into MICRO-EHV.

[Reference: Causes of assembling error]

No.	Error contents
1	CINTN with the same number exists
2	CINTN not defined

Program example

Refer to the explanation page of “CRTIN”.

Name	RETURN from counter interrupt scan (2nd comparison value)

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
CRTIN	—	2	●	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	—	—	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
—	(No argument)													

Remarks

Function

- This instruction declares the end of the interrupt scan program for counter.
- When this command is executed, Processing is returned to the program that was being executed before the interrupt scan.

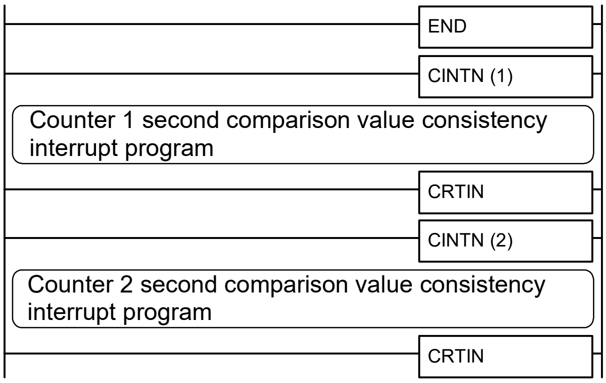
Cautionary notes

- Do not include a start condition in this command.
- Unlike MICRO-EH currently in use, the editor detects the assembling error. Note because the program including causes of the assembling error cannot be written into MICRO-EHV.

[Reference: Causes of assembling error]

No.	Error contents
1	CRTIN not defined
2	CRTIN area error
3	CRTIN startup condition error

Program example



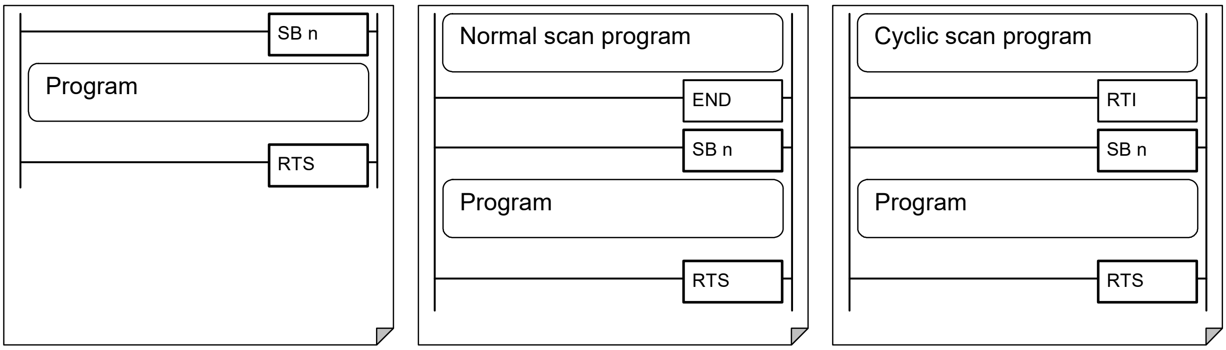
[Program description]

- After the RUN starts, the CINTN (1) to CRTIN program is executed when the counter value of counter 1 reach the second comparison value.
- After the RUN starts, the CINTN (2) to CRTIN program is executed when the counter value of counter 2 reach the second comparison value.

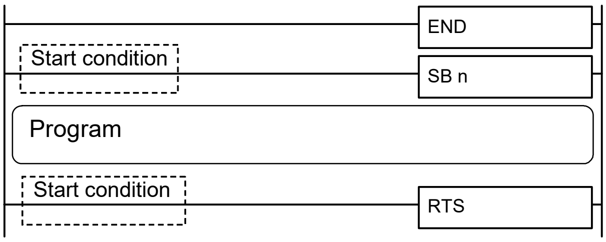
Reference Grammar of subroutine, cyclic, interrupt program

[1] The subroutine is described in the subroutine sheet.

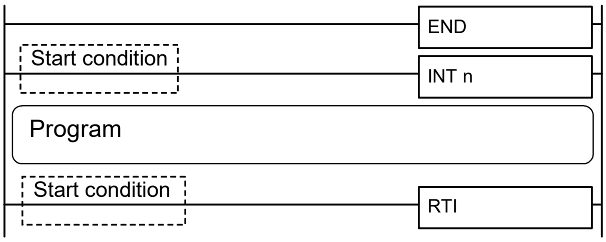
It can also be written after the normal scan END instruction and after the RTI instruction of the cyclic scan sheet.



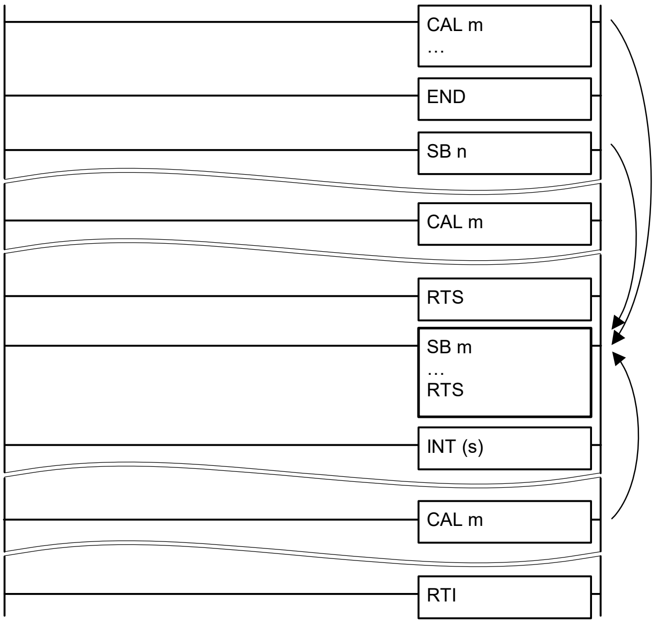
[2] Program the subroutine start command (SB n) and subroutine end command (RTS) without starting conditions.



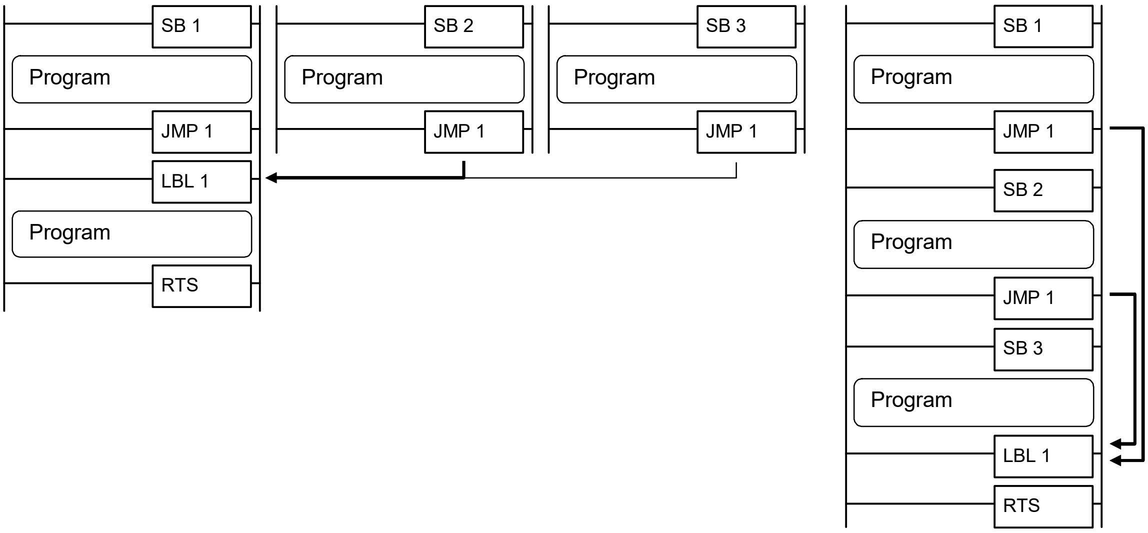
[3] Program the cyclic scan start (INT (s)) and scan end (RTI) instructions without start conditions.



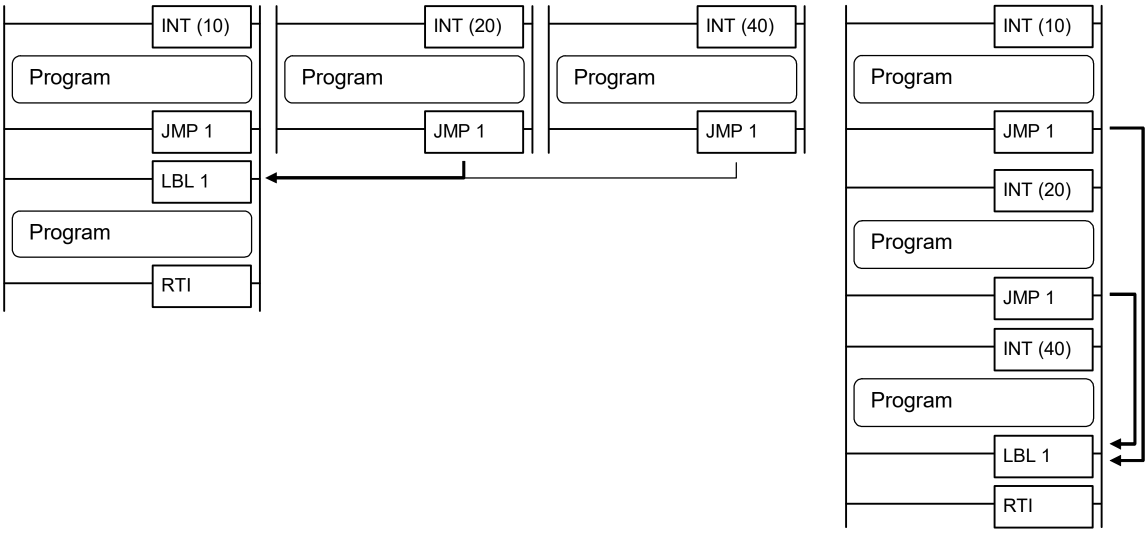
[4] The same subroutine can be called from normal scan, interrupt scan, and subroutine.



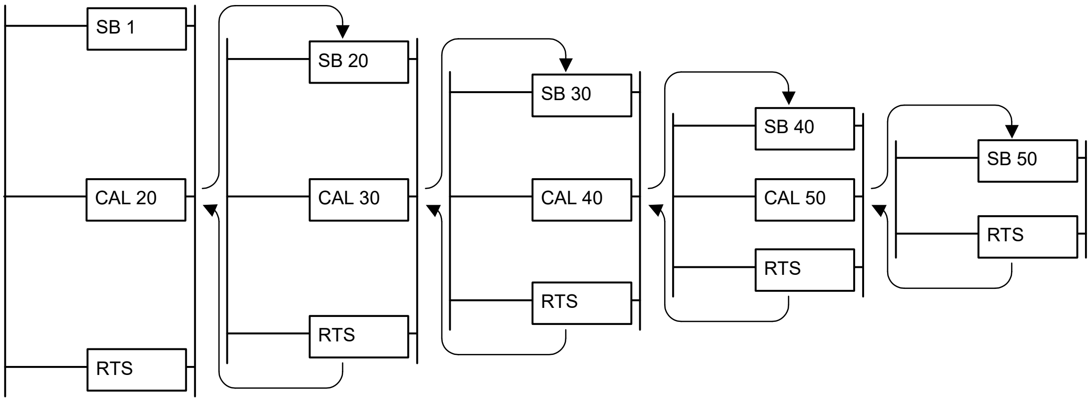
[5] Subroutines with many entrances and one exit can be programmed.



[6] Cyclic scan with many entrances and one exit can be programmed.



[7] Subroutine nesting is possible up to 5 layers.



The order of subroutine programs is not related to the order of nesting.

[1] Basic commands

[2] Arithmetic commands

[3] Application commands

[4] Control commands

[5] CPU communication commands

Name	General-purpose port transmitting command

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
TRNS0 (s, t)	–	6	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
–	400+0.858*n	400+0.858*n	–	–	–

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	Top I/O in parameter area								✓					
t	Top I/O of communication control bit			✓										

Remarks
s parameters are occupied up to s+E. t parameters are occupied up to t+B. n of processing time is the number of bytes.

Function

This is the command to communicate in a serial port

This command execution can send data from the serial port and receive the response from external devices.

Parameter

TRNS0 command uses 4 internal output areas shown below.

- Parameter for communication (s parameter area)

Area to set parameters, such as transmission speed and transmission character configuration for communiation.

- Bit for communication control (t parameter area)

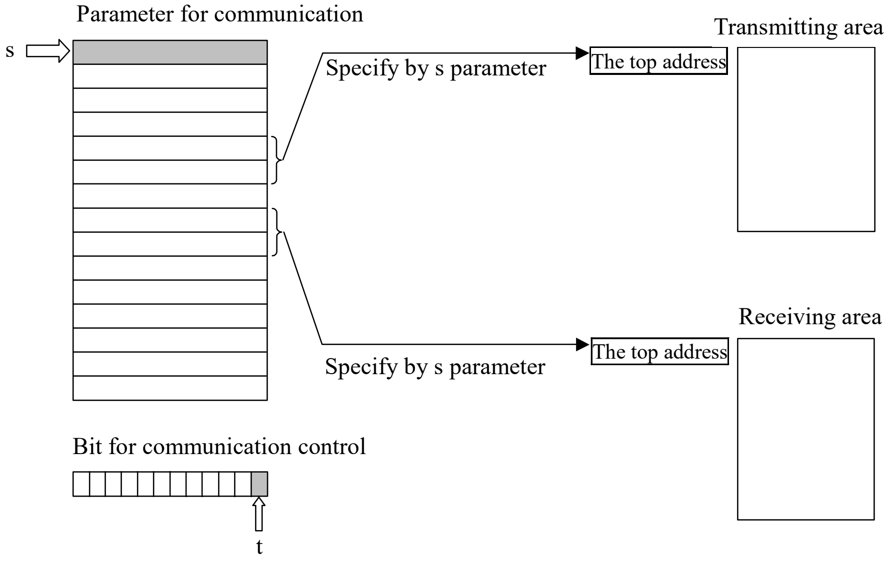
Area to start TRNS0 command and display a comamnd end and error information.

- Transmiting area

Area to set the transmitting data.

- Receiving area

Area to store the received data after transmitting.



(1) s parameter

The top I/O of "a table which stores each type of parameter for communication" is set to s.

[The details of s parameter area]

s	[1] Return code
s+1	▨ [2] System area
	▨ (No using by user)
s+3	[3] Timeout time
s+4	[4] Top I/O of transmitting data area
s+6	[5] Size of transmitting data area
s+7	[6] Top I/O of receiving data area
s+9	[7] Size of receiving data area
s+A	[8] Receiving data length
s+B	[9] Start code
s+C	[10] Termination code
s+D	[11] Transmission speed
s+E	[12] Transmission format

▨ No writing area by users

☐ Setting area by users

[1] Return code:

The executed result of TRNS 0 is set in the lower 8 bits.

Case of normal end = 0

Case of abnormal end ≠ 0 (Refer to the error code list.)

[2] System area:

This is used on the system processing for TRNS 0 when TRNS 0 is executed. Users cannot use this area.

[3] Timeout time:

The timeout time from beginning to end of execution of TRNS 0 is specified.

= 0: The timeout time is not checked.

≠ 0: The timeout of '×10ms' is checked.

(It can set up to HFFFF.)

[4] The top I/O of transmitting data area:

The top I/O of an area to store the transmitting data by TRNS 0 is specified.

The top I/O of the transmitting data area is coded by the I/O address coding command before executing TRNS 0 to store in s+4 and s+5. (Usable I/O is WR and WM)

[5] Size of transmitting data area:

The size of the transmitting data area is specified in word units.

[6] The top I/O of receiving data area:

The top I/O of an area to store the response data to the transmitting data is specified.

The top I/O in the receiving data area is coded by I/O address coding command before executing TRNS 0 to store in s+7 and s+8. (Usable I/O is WR and WM)

[7] Size of receiving data area:

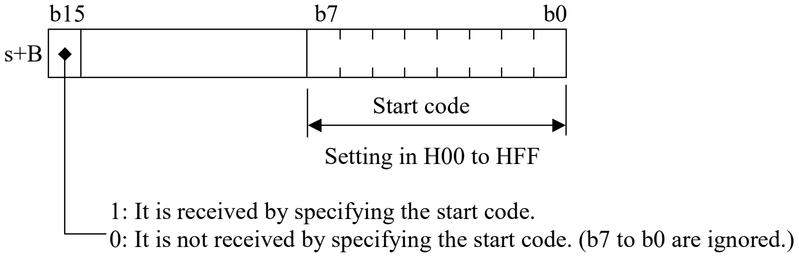
The size of the receiving data area is specified in word units.

[8] Receiving data length:

The receiving data length is specified in byte units. But the length should not exceed 1,024 bytes or the receiving data area. If exceeded, it becomes abnormal end because of DER = 1.

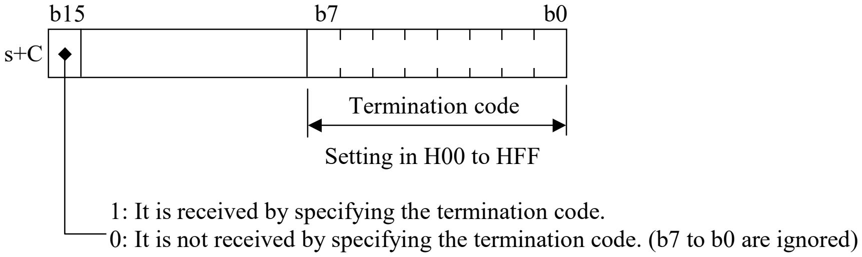
[9] Start code:

The code to start the receiving is specified.



[10] Termination code:

The code to terminate the receiving is specified.



[11] Transmission speed:

Transmission speed is specified.

Transmission speed	Set value
300 bps	H0000
600 bps	H0001
1,200 bps	H0002
2,400 bps	H0003
4,800 bps	H0004

Transmission speed	Set value
9,600 bps	H0005
19,200 bps	H0006
38,400 bps	H0007
57,600 bps	H0008
115,200 bps	H0009

[12] Transmission format:

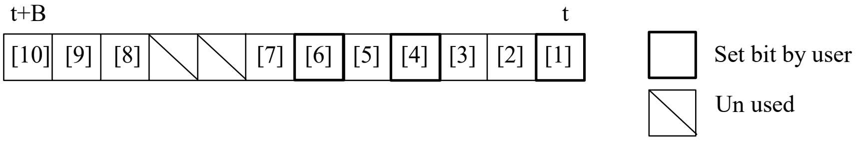
Transmission format is specified.

Transmission format			Set value
7bit	even parity	2 stops	H0000
7bit	odd parity	2 stops	H0001
7bit	even parity	1 stop	H0002
7bit	odd parity	1 stop	H0003
8bit	no parity	2 stops	H0004
8bit	no parity	1 stop	H0005
8bit	even parity	1 stop	H0006
8bit	odd parity	1 stop	H0007

(2) t parameter

The top I/O of "Bit table to control communication" is set to t.

[The details of t parameter]



[1] Execution of communication:

The user program sets 1 when TRNS 0 is executed.

TRNS 0 resets it to 0 if communication terminates.

[2] Normal end:

It is set to 1 if communication terminates normally by TRNS 0.

And when communication is started (t bit is turned ON), TRNS 0 resets it to 0.

[3] Abnormal end:

It is set to 1 if communication terminates abnormally by TRNS 0.

And when communication is started (t bit is turned ON), TRNS 0 resets it to 0.

[4] Initial requirement:

When TRNS 0 is set to the initial state, it is set to 1. The initial requirement under communicating terminates communication forcedly.

[5] Initial end:

When the initial of TRNS 0 terminates normally, it is set to 1. (In this case, [4] Initial requirement is reset to 0.)

[6] Continuation:

It sets 1 when receiving continuously after terminating the transmitting. TRNS 0 resets it to 0 after terminating communication.

[7] Parity error / Framing error / Overrun error:

If parity error, framing error, or overrun error occurs under communicating, it is set to 1.

[8] Timeout:

If a time out occurs under communicating, it is set to 1.

[9] Input buffer full:

If a receiving buffer is full, it is set to 1.

[10] Conflict error：

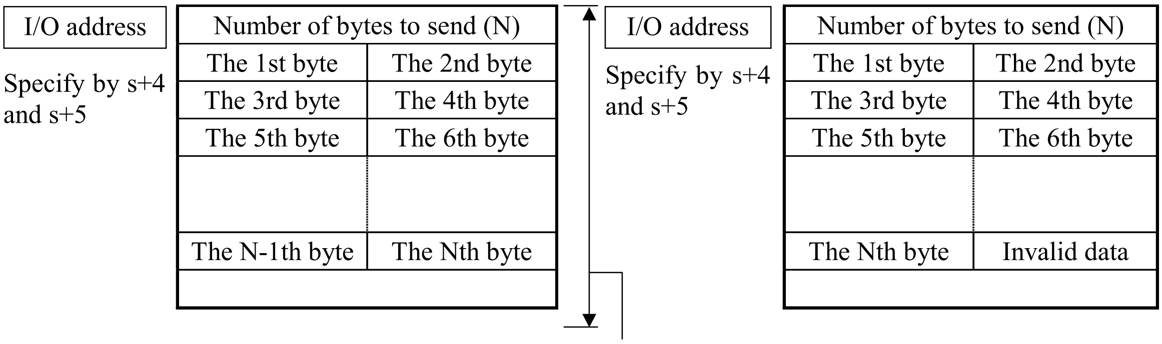
If 2 TRNS 0 or more are going to be started simultaneously on the user program, or TRNS 0 and RECV 0 are started simultaneously, it is set to 1. (In this case, communication is terminated forcedly.)

* [7] to [10] are reset to 0 by TRNS 0 initially when TRNS 0 is started.

(3) Transmitting data area

The setting of data to transmit follows the composition shown below.

The number of bytes to send is even. The number of byte to send is odd.



Number of bytes to send (N)	
The 1st byte	The 2nd byte
The 3rd byte	The 4th byte
The 5th byte	The 6th byte
The N-1th byte	The Nth byte

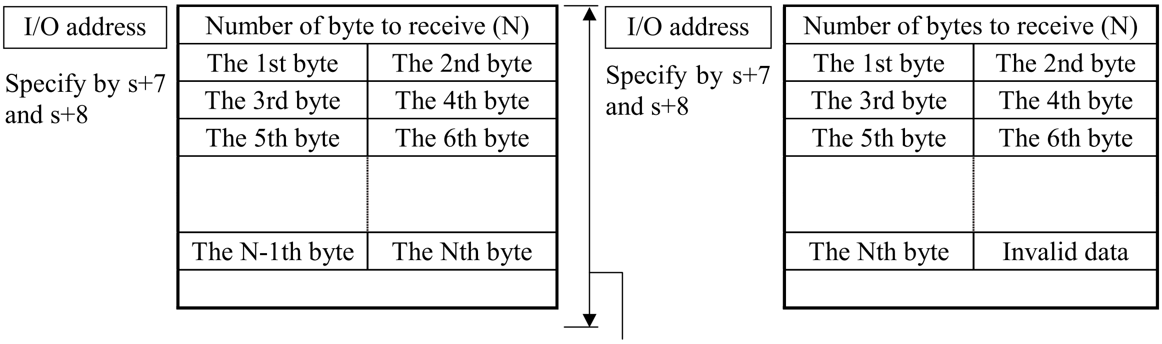
Number of bytes to send (N)	
The 1st byte	The 2nd byte
The 3rd byte	The 4th byte
The 5th byte	The 6th byte
The Nth byte	Invalid data

Size of the sending data area

(4) Receiving data area

The setting of receiving data follows the composition shown below.

The number of bytes to receive is even. The number of bytes to receive is odd.



Number of byte to receive (N)	
The 1st byte	The 2nd byte
The 3rd byte	The 4th byte
The 5th byte	The 6th byte
The N-1th byte	The Nth byte

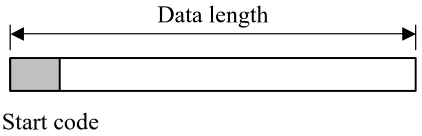
Number of bytes to receive (N)	
The 1st byte	The 2nd byte
The 3rd byte	The 4th byte
The 5th byte	The 6th byte
The Nth byte	Invalid data

Size of the receiving data area

Method of data communication

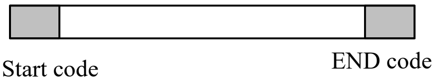
A method of data communication is specified from the following 4 ways.

(1) To specify by the start code and the receiving data length.



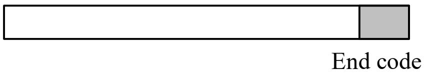
s+A : The receiving data length (byte)
s+B : H80□□ (□□ is the start code.)
s+C : H0000

(2) To specify by the start code and the end code.



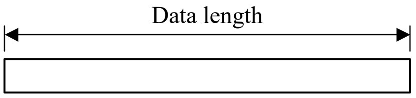
s+A : H0000
s+B : H80□□ (□□ is the start code.)
s+C : H80□□ (□□ is the end code.)

(3) To specify by the end code.



s+A : H0000
s+B : H0000
s+C : H80□□ (□□ is the end code.)

(4) To specify by the receiving data length.

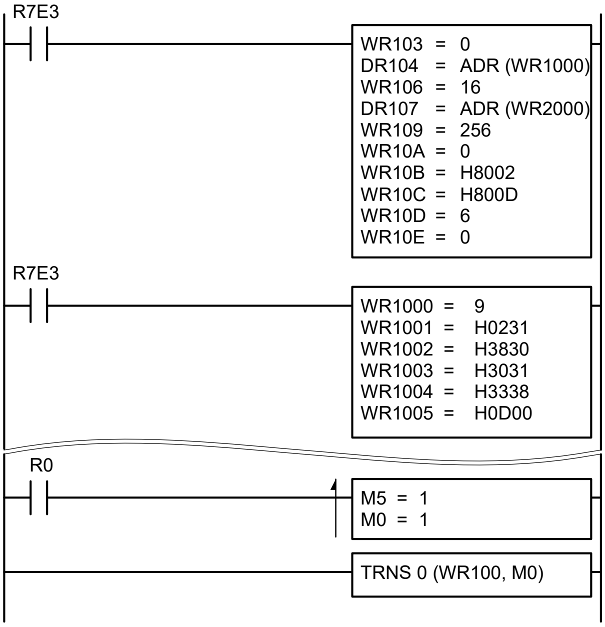


s+A : The receiving data length (byte)
s+B : H0000
s+C : H0000

Cautionary notes

- It can act when a serial communication port is set for ‘General purpose’.
- TRNS 0 initializes an internal work area at the 1st scan after RUN. Thus the set of bit (t+0) to execute communication should be executed after the 2nd scan.
- When a startup condition is before TRNS 0, the startup condition should not be specified since a system software may not be able to execute the initializing process normally.
- s+E and t+B should be used within the range of I/O. It is impossible to write a parameter outside the range of I/O.
- If timeout occurred when receiving data, data which had received by the time the timeout has occurred is stored in the receiving data area if those data are normal.
- When writing commands into the cyclic scan, the cycle of the cyclic scan should be 10 ms or more.
- If you specify a start code or end code and receive it, the start code and end code is included in the received data.

Program example



```
R7E3
WR103  =  0
DR104  =  ADR (WR1000)
WR106  =  16
DR107  =  ADR (WR2000)
WR109  =  256
WR10A  =  0
WR10B  =  H8002
WR10C  =  H800D
WR10D  =  6
WR10E  =  0

R7E3
WR1000 =   9
WR1001 =   H0231
WR1002 =   H3830
WR1003 =   H3031
WR1004 =   H3338
WR1005 =   H0D00

R0
M5  =  1
M0  =  1

TRNS 0 (WR100, M0)
```

R7E3: 1st scan turns ON after RUN
No timeout
Transmitting area 16 words from WR1000
Receiving area 256 words from WR2000
Method of receiving data:
Start code H02, End code H0D
Transmission speed 19.2k bps
Transmission format 7 bits, EVEN, 2 stops

Sending data Total 9 bytes
02 31 38 30 30 31 33 38 0D

If R0 is turned ON, the bit (M0) to execute turns ON and data is transmitted.
If M5 is turned ON, responses data from the other device can be receive after transmitting.

[Program description]

This is a sample program to transmit a forward data to our inverter SJ300/L300P. The parameter of TRNS 0 and the transmitting data are set at the first scan after RUN. The bit to execute M is started if R0 is turned ON, and data is sent. If the command is executed normally, the response from the inverter is stored in WR2000, or after.

Return code

The following table is a list of the return code to be stored in the top of s parameter area after executing TRNS 0 / RECV 0.

Return code	Name	Description	Countermeasure
H00	Normal end	Transmitting and receiving were terminated normally.	—
H22	Setting error of transmitting area	Setting of the top of the transmitting area is not correct.	Set the top of the transmitting area within correct range.
H23	Range error of transmitting area	The end of the transmitting area exceeds the range of I/O.	Set the transmitting area within correct range.
H24	Setting error of receiving area	Setting of the top of the receiving area is not correct.	Set the top of the receiving area within correct range.
H25	Range error of receiving area	The end of the receiving area exceeds the range of I/O.	Set the receiving area within correct range.
H26	Setting error of transmitting data length	Setting of the transmitting data length is more than the transmitting area length.	Set the transmitting data length within the range of the transmitting area.
H27	Setting error of receiving data length	Setting of the receiving data length is more than the receiving area length.	Set the receiving data length within the range of the receiving area.
H28	Area overlap error *1	There is an overlapped area between s parameter, t parameter, the transmitting area, and the receiving area.	Set each area without overlapping those areas.
H30	Timeout *2	Processing of transmitting and receiving was not terminated within the specified time.	Make the set value larger, or check the details of processing.
H40	Data over of receiving area *3	There is no space because the receiving area is filled with the receiving data	Make the receiving area larger.
H41	Parity error Framing error Overrun error *4	One of parity error, framing error, and overrun error occurred on the communication processing.	Check the transmission route of a general-purpose port and data format, etc.
H44	Contention error	TRNS0/RECV0 was started simultaneously at 2 places or more.	Do not start simultaneously at 2 places or more.
H45	Parameter error	The setting values for TRNS0/RECV0, such as port rate and transmission code, are not correct.	Set the correct value.
H46	Error of port specification	TRNS0/RECV0 was invoked when not a general-purpose port was specified. TRNS0/RECV0 was started when the option board was specified as a general-purpose port with no option board installed.	Check which port has been specified.

*1 Though a return code of area overlap error is H28, note that the return code may not be displayed as H28 if the return code area overlaps with a part of t parameter.

*2 Though it becomes a timeout error if a timeout occurs under receiving data, received data by the time the timeout occurs is stored in the receiving data area.

*3 The size of the receiving area is up to 1,024 bytes.

*4 The receiving data is not guaranteed at the time of receiving.

PRN ➔ PRJ

This command is equivalent to TRNS 0 (d, s, t) in the program (PRN file) of MICRO-EH.

How to convert the program which has used TRNS 0 (d, s, t) into the program for MICRO-EHV is as follows.

TRNS 0 (d, s, t) ➔ TRNS 0 (s, t)

s+4 : I/O types of the transmitting data area➔ s+4, s+5 : Transmitting data by I/O address coding command

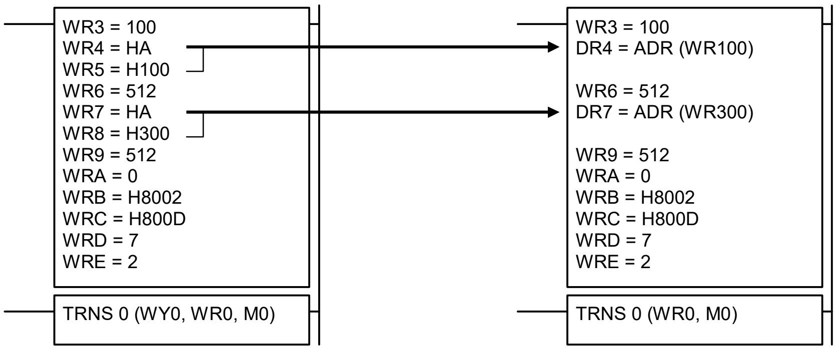
s+5 : I/O No. of the transmitting data area ➔ Specify the top address of the area

s+7 : I/O types of the receiving data area ➔ s+7, s+8 : Receiving data by I/O address coding command

s+8 : I/O No. of the receiving data area ➔ Specify the top address of the area

Example) Case of TRNS 0 (WY0, WR0, M0), 512 words from the transmitting data area, and 512 words from the transmitting data area WR300.

Program for MICRO-EH Program for MICRO-EHV



```
WR3 = 100
WR4 = HA
WR5 = H100
WR6 = 512
WR7 = HA
WR8 = H300
WR9 = 512
WRA = 0
WRB = H8002
WRC = H800D
WRD = 7
WRE = 2

TRNS 0 (WY0, WR0, M0)
```

```
WR3 = 100
DR4 = ADR (WR100)

WR6 = 512
DR7 = ADR (WR300)

WR9 = 512
WRA = 0
WRB = H8002
WRC = H800D
WRD = 7
WRE = 2

TRNS 0 (WR0, M0)
```

* Convert Tool started from Control Editor cannot convert a specific area of the top I/O of the transmitting data area and the receiving data area. Thus please convert as mentioned above.

[Note on converting a program]

The difference in action of TRNS 0 / RECV 0 between MICRO-EH and MICRO-EHV is shown below.

Item	MICRO-EH	MICRO-EHV
The receiving data at the time of occurrence of timeout	The receiving data is cancelled.	Received data by the time the timeout occurs is stored in the receiving data area.
Error of communication data	Parity error, framing error, and overrun error can be distinguished.	Parity error, framing error, and overrun error cannot be distinguished.

Name	General-purpose port receiving command

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
RECV0 (s, t)	—	6	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	500+1.107*n	500+1.107*n	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	Top I/O in parameter area								✓					
t	Top I/O of communication control bit			✓										

Remarks
s parameters are occupied up to s+E. t parameters are occupied up to t+B. n for processing time is the number of bytes.

Function

This is a command to communicate in a serial port.

This command execution can receive data from external devices on the serial port and transmit data after receiving.

Parameter

RECV 0 command uses 4 internal output areas shown below.

- Parameter for communication (s parameter area)

Area to set parameters, such as transmission speed and transmission character configuraiton for communication.

- Bit for communication control (t parameter area)

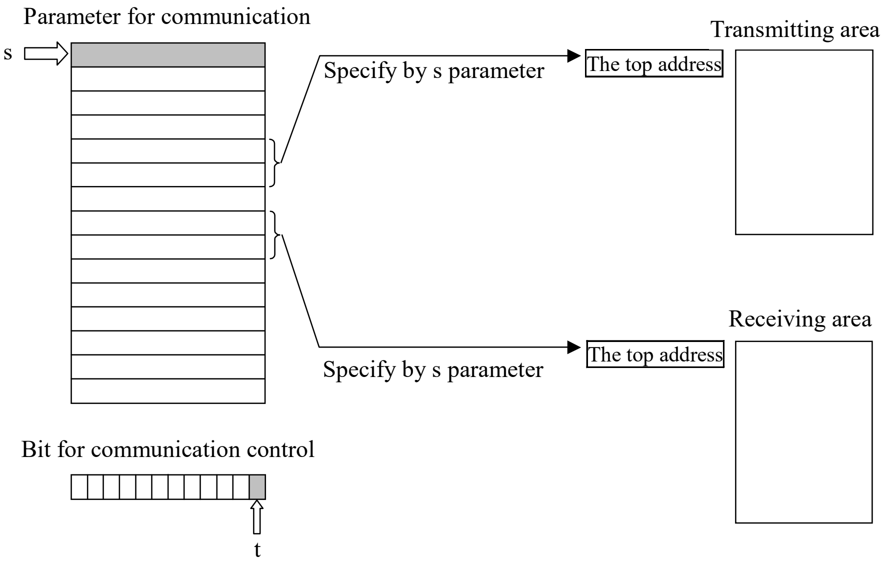
Area to start RECV 0 command and display the command end and error information.

- Transmitting area

Area to set transmitting data.

- Receiving area

Area to store received data .



(1) s parameter

The top I/O of “a table which stores each parameter for communication” is set to s.

Each parameter’s meaning is the same as TRNS 0 command. See the description of TRNS 0 for details.

(2) t parameter

The top I/O of “a bit table to control communication” is set to t.

A meaning of each bit is the same as the content of TRNS 0 command except for a continuity bit (t+5). See the description of TRNS 0 for details.

t+B											t
[10]	[9]	[8]	⧅	⧅	[7]	**[6]**	[5]	**[4]**	[3]	[2]	**[1]**

☐ Set bit by user

⧅ Un used

[6] Continuation:

It sets 1 when transmitting continuously after terminating the receiving. RECV 0 resets it to 0 after terminating communication.

(3) Transmitting data area

The composition of the transmitting data area is the same as TRNS 0. See the description of TRNS 0 for details.

(4) Receiving data area

The composition of the receiving data area is the same as TRNS 0. See the description of TRNS 0 for details.

Cautionary notes

- RECV 0 initializes an internal work area at the 1st scan after RUN. Thus the set of bit (t+0) to execute communication should be executed after the 2nd scan.
- When a startup condition is before RECV, the startup condition should not be specified since system software may not be able to execute the initializing process normally.
- s+E and t+B should be used within the range of I/O. It is impossible to write a parameter outside the range of I/O.
- If timeout occurred when receiving data, data which had received by the time the timeout has occurred is stored in the receiving data area if those data are normal.
- When writing commands into the cyclic scan, the cycle of the cyclic scan should be 10 ms or more.

Method of data communication

A method of data communication is specified from the following 4 ways.

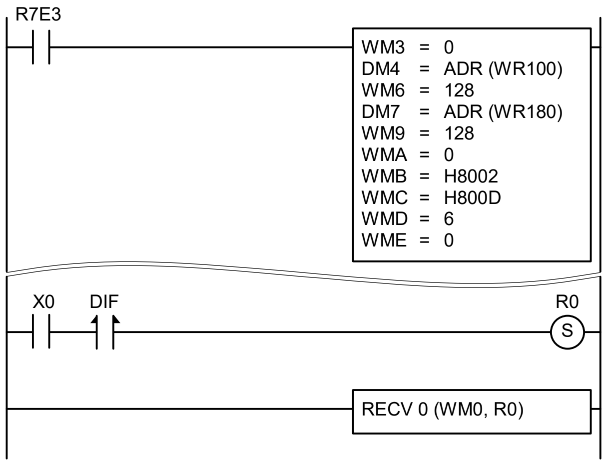
(1) To specify by the start code and the receiving data length.

(2) To specify by the start code and the end code.

(3) To specify by the end code.

(4) To specify by the receiving data length.

Program example



R7E3: 1st scan turns ON after RUN.
No timeout
Transmitting area 128 words from WR100
Receiving area 128 words from WR180
Method of data receiving:
Start code H02, End code H0D
Transmission speed 19.2k bps
Transmission format 7 bits, EVEN, 2 stops

[Program description]

The parameter of RECV 0 is set at the first scan after RUN.

When X0 turns ON, the executing bit R0 is started and it waits for data receiving. (It keeps waiting until data is received since the setting is ‘No timeout’.)

If data from external devices is received normally, the receiving data is stored in WR180 or after.

PRN ➔ PRJ

This command is equivalent to RECV 0 (d, s, t) in the program (PRN file) of MICRO-EH.

How to convert the program which has used RECV 0 (d, s, t) into the program for EHV is as follows.

RECV 0 (d, s, t)	➔	RECV 0 (s, t)
s+4 : I/O types of transmitting data area	➔	s+4, s+5 : Transmitting data by I/O address coding command
s+5 : I/O No. of transmitting data area		Specify the top address of the area
s+7 : I/O types of receiving data area	➔	s+7, s+8 : Receiving data by I/O address coding command
s+8 : I/O No. of receiving data area		Specify the top address of the area

* Convert Tool started from Control Editor cannot convert a specific part of each top I/O of the transmitting data area and the receiving data area. Please convert it referring to the description pages of TRNS 0.

Name	Modbus protocol query transmitting command

Ladder format	Number of steps		Condition code				
	Condition	Steps	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
MBMST (s, t)	—	6	↕	●	●	●	●

Command processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	51	51	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	Top I/O in parameter area								✓					
t	Top I/O of communication control bit			✓										

Remarks
s parameters are occupied up to s+9. t parameters are occupied up to t+5.

Function

This is the command to perform serial communication with Modbus protocol in option board.

This command execution can transmit query from the serial port and receive response from the external device.

Executable function codes using this command are shown in the following table.

Code	Function		Broadcast (*)
01 (0x01)	Read Coil Status	Reads the coil status.	—
02 (0x02)	Read Input Status	Reads the input status.	—
03 (0x03)	Read Holding Registers	Reads the holding register status.	—
04 (0x04)	Read Input Registers	Reads the input register status.	—
05 (0x05)	Force Single Coil	Changes the coil status to ON or OFF.	OK
06 (0x06)	Preset Single Register	Changes the holding register status.	OK
08 (0x08)	Diagnostics	Diagnoses the slave device.	—
15 (0x0F)	Force Multiple Coils	Changes the status of two or more coils to ON or OFF.	OK
16 (0x10)	Preset Multiple Registers	Changes the status of two or more holding register to ON or OFF.	OK

* Broadcast is a communication for all slave stations. Setting the slave address to [H00] makes it a broadcast communication. In broadcast communication, the slave cannot return a response command.

Reference: What is Modbus?

Modbus protocol is a communication protocol which has developed for PLC by Modiocn Inc. (AEG Schneider Automation International S.A.S.). Modbus protocol defines only communication protocol, and physical layer such as a medium of communication is not defined. Refer to Appendix 3 in this manual for Modbus protocol.

And if you need further information, refer to "Modbus Protocol Reference Guide (PI-MBUS-300)" issued by Modicon Inc.

Parameter

MBMST command uses 4 internal output areas shown below.

- Parameter for communication (s parameter area)

Area to set parameters, such as transmission speed for communication and transmission character configuration.

- Bit for communication control (t parameter area)

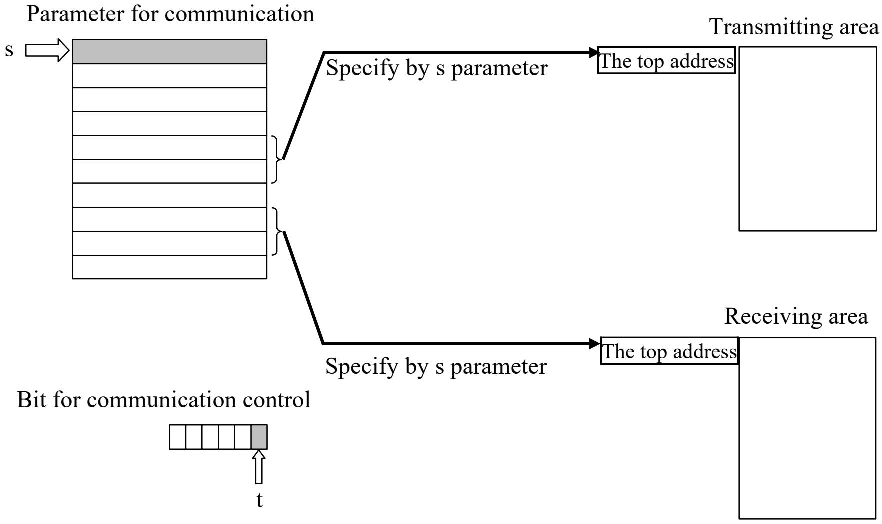
Area to start MBMST command and display the command end and error information.

- Transmitting area

Area to set transmitting data.

- Receiving area

Area to store data received after transmitting.



(1) s parameter

The top I/O of "a table which stores each parameter for communication" is set to s.

s	[1] Return code
s+1	▨ [2] System area
	▨ (No using by user)
s+3	Opened space
s+4	[3] Top I/O of transmitting data area
s+6	[4] Transmitting data area size
s+7	[5] Top I/O of receiving data area
s+9	[6] Receiving data area size

▨ No writing area by user

☐ Setting area by user

[1] Return code:

Execution result of MBMST is set.

Normal end = 0

Abnormal end ≠ 0 (See a list of error code)

[2] System area:

It is used on the system processing of MBMST when executing MBMST.

User cannot use this area.

[3] Top I/O of transmitting data area:

Specify the top I/O of the area to store the data to be sent by the MBMST command with the I/O address coversion (ADR).

(Usable I/O is WR and WM.)

[4] Transmitting data area size:

Specifies the transmitting data area size in word unit.

Set a value greater than the actual size of the data to be sent.

[5] Top I/O of receiving data area:

Specify the top I/O of the area to store the response data for the sent data with the I/O address coversion (ADR). (Usable I/O is WR and WM.)

[6] Receiving data area size:

Specify the size of the area to store received data by word. Set a value that is larger than the actual size of the received data.

(2) t parameter

The top I/O of “a bit table to control communication” is set to t.

[Details of t parameter area]

t+5					t
[6]	[5]	[4]	[3]	[2]	[1]

☐ User bit to set

[1] Execution of communication:

The user program sets 1 when executing MBMST.

MBMST resets it to 0 if communication terminates.

[2] Normal end:

It is set to 1 if communication terminates normally by MBMST.

And when starting communication (t bit is turned on), MBMST resets it to 0.

[3] Abnormal end:

It is set to 1 if communication terminates abnormally by MBMST.

And when starting communication (t bit is turned on), MBMST resets it to 0.

[4] Abnormal end (Exceptional response receiving):

When the response data received from the external device is an exceptional response, it is set to 1. (When receiving the exceptional response, both of [3] and [4] are set to 1.)

And when starting communication (t bit is turned on), MBMST resets it to 0.

But, when receiving the exceptional response, the response (function code and exceptional code) of the exceptional response is stored in the return code area and nothing is stored in the receiving area.

[5] Initial requirement:

When MBMST is set to the initial status, it is set to 1. The initial requirement under communication terminates the communication forcedly.

[6] Initial end:

When the initialization of MBMST terminates normally, it is set to 1. (In this case, [5] initial requirement is reset to0.)

(3) Transmitting data area

Transmitting data area configuration changes depending on query transmitted.

Cautionary notes

Specify the number of bytes to be sent so that the number of data to be sent is the same as the number of bytes to be sent at the beginning of the transmission data area.

If the number of data to be sent and the number of bytes to be sent do not match, the return code H60 (Error on specification of number of data to be sent) returns when the MBMST instruction is executed.

(i) When function codes are 0x01, 0x02, 0x03, and 0x04

I/O address

Specify by s+4 and s+5

Number of transmitting bytes (H0006)	
Slave address	Function code
Top address of coil and register which are read	
Number of coils and registers which are read	

Transmitting data area size

(ii) When function codes are 0x05 and 0x06

I/O address

Specify by s+4 and s+5

Number of transmitting bytes (H0006)	
Slave address	Function code
Top address of coil and register which are written	
Value of coil and register which are written	

Transmitting data area size

* When function code is 0x05, only H0000 and HFF00 of the value of coil written are effective.

(iii) When function code is 0x08

I/O address

Specify by s+4 and s+5

Number of transmitting bytes (N)	
Slave address	H 0 8
Data 1 (Note 1)	Data 2 (Note 1)
Data 3 (Note 1)	Data 4 (Note 1)
Data N-1 (Note 1)	Data N (Note 1)

Transmitting data area size

Note 1) When the transmitting data is odd bytes, data is only upper bytes.

Reference : Modbus data address

Data address must be specified for data reading and writing. According to the “Modbus Protocol Reference Guide” there are four types of data (coil, input status, input register, and holding register).

(1) Coil	Bit data from 1 to 9,999 for reading and writing.
(2) Input status	Bit data from 10,001 to 19,999 for reading only.
(3) Input register	Word data from 30,001 to 39,999 for reading only.
(4) Holding register	Word data from 40,001 to 49,999 for reading and writing.

Since data type is specified by function code, the address in message is only 4 digits. Moreover, if the address of a message format is set to "0000", then the specified address is "x0001" (x being the maximum higher rank for each data classification).

Example) Input register Data address 10789

4 low figures 789 ➔ Offset from Address 788 ➔ Specification of data address 0x0314

Hereafter, the explanation of each function code is written in accordance with this.

(iv) When function code is 0x0F

I/O address

Specify by s+4 and s+5

Number of transmitting bytes (N)	
Slave address	H 0 F
Coil top address	
Number of coils	
Data 2 (Note 1)	Data 1 (Note 1)
Data 4 (Note 1)	Data 3 (Note 1)
Data N-6 (Note 1)	Data N-7 (Note 1)

Transmitting data area size

Note 1) Set so that the top data of the coil to set is set to LSB of word data.

When tranmissint data is odd bytes, data is only lower bytes.

* Data format actually transmitted from the serial port is as follows. The element called “the number of bytes” is added with the system.

D.A.	F.C.	Top address	Coils	Bytes	D1	D2	---------	C.C.
(1)	(1)	(2)	(2)	(1)	(1)	(1)		

D.A: Slave address, F.C.: Function code, Dn: Data, C.C.: Check code

The number of bytes of ( )

Since the number of data depends on the number of bytes, the maximum of N is 252. When N is more than 252, DER = 1 and it is not executed.

(v) When function code is 0x10

I/O address

Specify by s+4 and s+5

Number of bytes (N)	
Slave address	H 1 0
Register top address	
Number of registers	
Data 1	
Data 2	
Data (N-6) / 2	

Transmitting data area size

* Data format actually transmitted from the serial port is as follows. The element called “the number of byte” is added with the system.

D.A.	F.C.	Top address	Registers	Bytes	D1(H)	DI(L)	---------	C.C.
(1)	(1)	(2)	(2)	(1)	(2)			

Since the number of data depends on the number of bytes, the maximum of N is 252. When N is more than 252, DER = 1 and it is not executed.

(4) Receiving data area

The receiving data area configuration changes depending on the response received (query to transmit).

(i) When function codes are 0x01 and 0x02

I/O address

Specify by s+7 and s+8

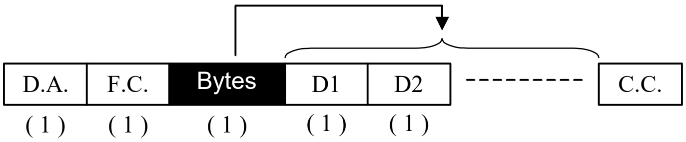
Number of receiving bytes (N)	
Slave address	Function code
Data 2 (Note 1)	Data 1 (Note 1)
Data 4 (Note 1)	Data 3 (Note 1)
Data N-2 (Note 1)	Data N-3 (Note 1)

Receiving data area size

Note 1) Data in the top coil specified is set to LSB of word data.

When the receiving data is odd bytes, data is only lower bytes.(Upper bytes stores H00.)

* Data format actually received by the serial port is as follows. The element called “the number of bytes” is deleted with the system.



(ii) When function codes are 0x03 and 0x04

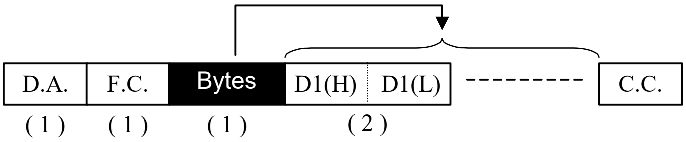
I/O address

Specify by s+7 and s+8

Number of receiving bytes (N)
Slave address / Function code
Data 1
Data 2
Data (N-2) / 2

Receiving data area size

* Data format actually received by the serial port is as follows. The element called “the number of bytes” is deleted with the system.



(iii) When function codes are 0x05, 0x06, 0x0F, and 0x10

I/O address

Specify by s+7 and s+8

Number of receiving bytes (H0002)	
Slave address	Function code

Receiving data area size

(iv) When function code is 0x08

I/O address

Specify by s+7 and s+8

Number of bytes (N)	
Slave address	H 0 8
Data 1 (Note 1)	Data 2 (Note 1)
Data 3 (Note 1)	Data 4 (Note 1)
Data N-1 (Note 1)	Data N (Note 1)

Receiving data area size

Note 1) When receiving data is odd bytes, data is only upper bytes. (Lower bytes retains the preceding value.)

Cautionary note

- This does not work and error occurs unless the serial communication port setting is “General-purpose port”.
- MBMST initializes an internal work area at the first scan after RUN. Therefore, perform the set of communication executing bit (t+0) after second scan or later.
- Do not specify the start condition because the system software cannot execute the processing for initializing properly if there is a start condition before MBMST command.
- Use s+C and t+5 within the I/O range. You cannot set a parameter which exceeds the I/O range.
- When the slave address is specified to the broadcast (H00) but the function code is not corresponding to the broadcast, the command is not executed because of DER = 1.
- When the function code is not corresponding to the number of transmitting bytes, the command is not executed because of DER = 1.

  Example) The number of transmitting bytes of function codes: 0x01, 0x02, 0x03, 0x04, 0x05, and 0x06 is other than 6.

  The number of transmitting bytes of function codes: 0x0F and 0x10 is odd number.
- When the function code is 0x05, only H0000 and HFF00 of the value written in the coil are effective.

  When other values except H0000 and HFF00 are specified, the command is not executed because of DER = 1.
- The number of receiving bytes of the receiving data area is initialized at the timing of data transmitting.
- When the response from the external device is an exceptional response, the response (function code and exceptional code) from the exceptional response is stored in the return code area and nothing is stored in the receiving data area.
- When the slave address is specified to the broadcast (H00), the normal end bit (t+1) turns on at the completion transmitting and the command is terminated.
- Communication interface depends on port type setting.
- When communication interface is specified to RS-232C, the control signal is not controlled while this command is executing.
- When setting it inside the cyclic scan, the cycle of the cyclic scan should be 10ms or more.

Return code

A list of return code stored in the top of s parameter after MBMST execution is as follows.

Return code	Name	Description	Countermeasure
H0000	Normal end	Transmitting and receiving were terminated properly.	—
H0021	Range check error	The end of parameters s and t exceeds the I/O range.	Set each parameter area within the correct range.
H0022	Setting error of transmitting area	Setting of the top of the transmitting area is not proper.	Set the top of the transmitting area within correct range.
H0023	Range error of transmitting area	The end of transmitting area exceeds the I/O range.	Set the transmitting area within correct range.
H0024	Setting error of receiving area	Setting of the top of receiving area is not proper.	Set the top of receiving area within correct range.
H0025	Range error of receiving area	The end of receiving area exceeds the I/O range.	Set the receiving area within correct range.
H0026	Setting error of transmitting data length	Setting of transmitting data length is the transmitting area length or more.	Set so that the transmitting data length is within the range of transmitting area.
H0027	Setting error of receiving data length	Setting of receiving data length is the receiving area length or more.	Set so that the receiving data length is within the range of receiving area.
H0028	Area overlap error *1	There is an overlapped area between parameters s and t, transmitting area, and receiving area.	Set each area so that there is no overlapped area.
H0030	Timeout	Transmitting and receiving processing did not terminate within the specified time.	Make the set value larger, or check the details of processing.
H0040	Data over	- Receiving data exceeded 1,028 bytes. - There is no space because receiving area was filled with receiving data.	- Verify the number of coils and registers of receiving data. - Make receiving area larger.
H0041	Parity error Framing error Overrun error	Parity error, framing error, or overrun error occurs on communication processing.	Verify the transmission route of the general-purpose port and, format and etc.
H0044	Contention error	The command using CPU serial port was started simultaneously at 2 locations or more.	Do not start the command simultaneously at 2 locations or more.
H0045	Parameter error	Set value such as transmission speed and transmission format (Modbus mode) of MBMST is not proper.	Set the correct value.
H0046	Error of port specification	MBMST was started when the serial port was not specified to the general-purpose port.	Verify the port setting.
H0060	Error of specification of the number of transmitting data	The number of transmitting bytes not corresponding to the function code was specified.	Verify the number of transmitting bytes.
H0061	Transmitting data error	Transmitting data setting is not proper.	Verify transmitting data.
H0071	Receiving data error	- Receiving data is not proper. - The number of receiving bytes except sizes of header/check code/trailer exceeds 509 bytes.	- Verify whether a device on slave side supports Modbus or not. - Verify the number of coils/registers of transmitting data.
H0072	CRC / LRC abnormal	Error occurred at CRC / LRC check.	Verify Modbus mode on slave side.

*1 Please note that though the return code of the area overlap error is H28, H28 as the return code may not be displayed if the return code area and a part of t parameter are used overlapping

Return code	Name	Description	Countermeasure
H81xx*2	F.C. 0x01 error	The exceptional response was received at F.C. 0x01.	Verify transmitting data.
H82xx*2	F.C. 0x02 error	The exceptional response was received at F.C. 0x02.	Verify transmitting data.
H83xx*2	F.C. 0x03 error	The exceptional response was received at F.C. 0x03.	Verify transmitting data.
H84xx*2	F.C. 0x04 error	The exceptional response was received at F.C. 0x04.	Verify transmitting data.
H85xx*2	F.C. 0x05 error	The exceptional response was received at F.C. 0x05.	Verify transmitting data.
H86xx*2	F.C. 0x06 error	The exceptional response was received at F.C. 0x06.	Verify transmitting data.
H88xx*2	F.C. 0x08 error	The exceptional response was received at F.C. 0x08.	Verify transmitting data.
H8Fxx*2	F.C. 0x0F error	The exceptional response was received at F.C. 0x0F.	Verify transmitting data.
H90xx*2	F.C. 0x 10 error	The exceptional response was received at F.C. 0x10.	Verify transmitting data.

*2 xx is the exceptional code.

Exceptional code

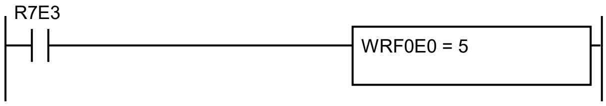
A list of exceptional codes is shown below.

Exceptional code	Name	Meaning
01	Illigal function	Slave does not support the function code received in the query..
02	Illigal data address	There is no specified data address in the slave device.
03	Illigal data value	A value contained inthe query data field is not allowd for the slave.
04	Slave device failure	Impossible to respond due to deveice filure.
05	Acknowledge	The elave has accepted the request and is processing it, but it takes time to reponse. (Prevent the timeout error of master.)
06	Slave device Busy	The slave is engaged in processing of the last command.

Refer to the manual of the device which is being connected for further information.

CAUTION

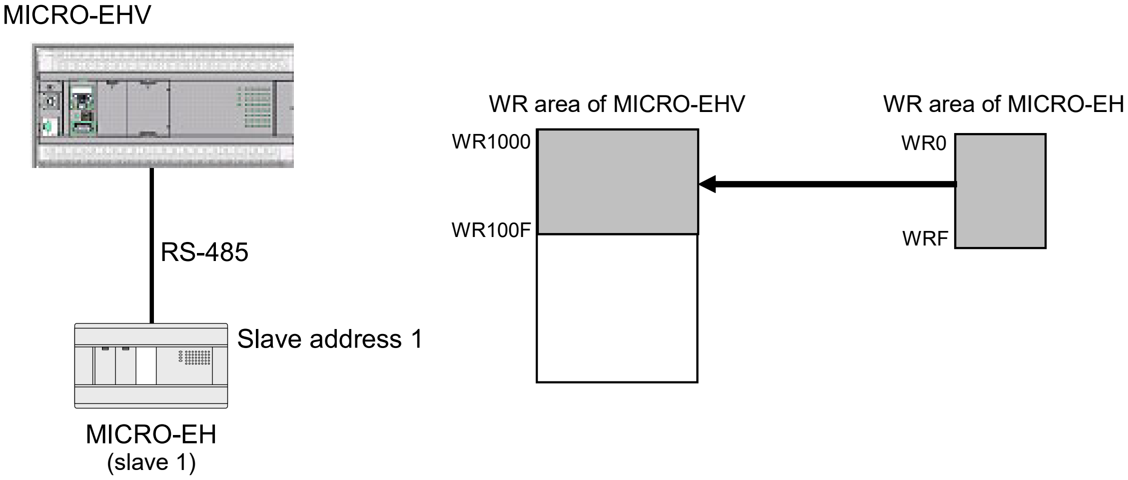
When communicating with multiple slaves continuously, commands may not be received correctly depending on the slave if the time from response reception to command transmission is short. To avoid this, the MICRO-EHV can insert the wait time until command transmission with the special internal output WRF0E0. This setting is common to all MBMST, INV1, OMST1, OCTP1, and Modbus gateway functions. Since this internal output is reset before RUN starts, set it with the ladder program as shown below.



R7E3: 1st scan turns ON after RUN.
WRF0E0: Modbus-RTU waiting time (master) 5ms

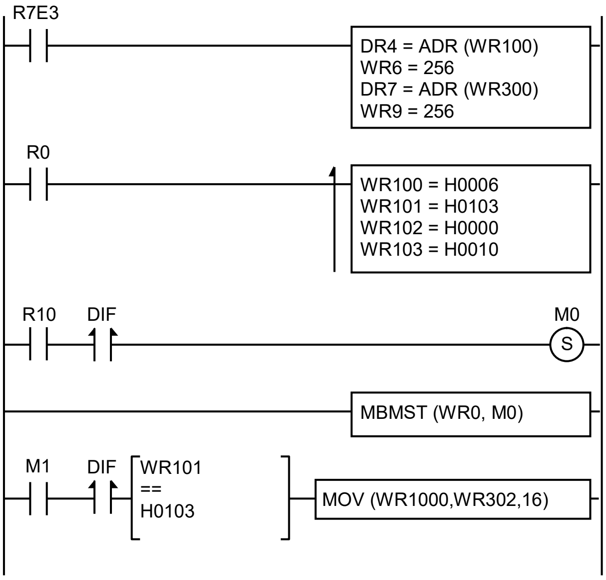
Program example

This program example is a program example with MICRO-EHV as the Modbus master and MICRO-EH as the Modbus slave. MICRO-EH (slave) collects the internal output of each MICRO-EH for one unit.



Function code used by MICRO-EHV：0x03(Read Holding Registers)

MICRO-EH Slave 1 read area：16 words from WR0 →address H0000, read size H0010(16)



R7E3: 1st scan turns ON after RUN.
No timeout
Transmitting area 256 words from WR100
Receiving area 256 words from WR300

Number of bytes sent 6
Slave address 01
Function code 03 (register read)
Read register start address H0000
Number of read registers 16

Store read data of slave 1 in WR1000

[Program description]

The MBMST command parameters are set in the first scan after RUN.

When R0 turns ON, the transmission data is set in the transmission area.

When R10 turns ON, execution BitM0 is activated and a query is sent from the master to the slave.

When data (response) from the MICRO-EH slave (address 1) is received normally, the received data is stored in WR300 and later.WR300 of the first word in the reception area is the number of received bytes, WR301 of the second word is the slave address and function code, and since the third word and later is data, the contents of WR302 to WR401 are copied to WR1000 to WR100F.

Reference : Functions in MICRO-EH when issuing each function code

Function Code	Function	Functions in MICRO-EH
01 (0x01)	Read Coil Status	Continuous n-point bit output (Y) status readout
02 (0x02)	Read Input Status	Continuous n-point bit input (X) status reading
03 (0x03)	Read Holding Registers	Continuous n-point word internal output (WR) status reading
04 (0x04)	Read Input Registers	Continuous n-point word output (WM) status reading
05 (0x05)	Force Single Coil	Bit output (Y) forced set
06 (0x06)	Preset Single Register	Word internal output (WM) set
08 (0x08)	Diagnostics	Diagnose slave devices.
15 (0x0F)	Force Multiple Coils	Continuous n-point bit output (Y) forced set
16 (0x10)	Preset Multiple Registers	Continuous n-point Word internal output (WR) set

CAUTION

Refer to the MICRO-EH 20 / 40 / 64-point basic unit application (NJI-465 *) for the specifications, address system, setting method, etc. of Modbus slave function of MICRO-EH.

PRN ➔ PRJ

This command is equivalent to FUN 191 (s) in the MICRO-EH program (PRN file).

When converting a program using FUN 191 (s) for MICRO-EHV, convert as follows.

FUN 191 (s) ➔ MBMST (s, t)

s+4 : Send by I/O address coding instruction ➔ s+4, s+5 : Send by I/O address coding instruction

Specify the start address of the data area → Specify the start address of the data area

s+5 : Dummy

s+7 : Received by I/O address coding instruction ➔ s+7, s+8 : Received by I/O address coding instruction

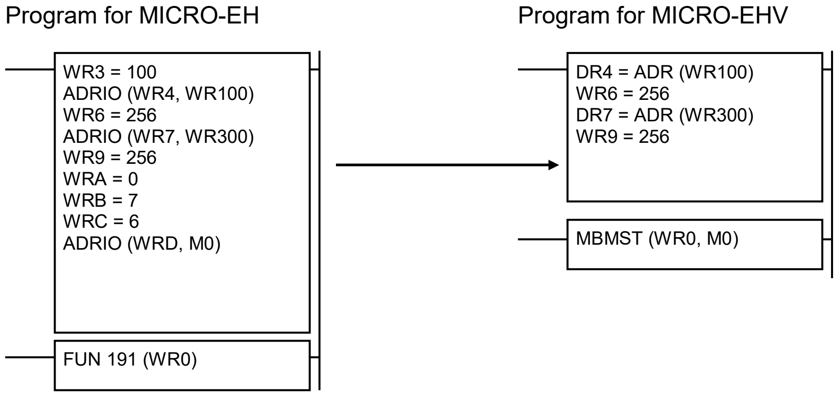
Specify the start address of the data area → Specify the start address of the data area

s+8 : Dummy

s+D : Received by I/O address coding instruction ➔ t : Top I/O of communication control bit

Specify the start address of the control bit area

Example : When FUN 191 (WR0), transmission data area WR100 to 256 word, transmission data area WR300 to 256 word, Top I/O of communication control bit is M0



Program for MICRO-EH

```
WR3 = 100
ADRIO (WR4, WR100)
WR6 = 256
ADRIO (WR7, WR300)
WR9 = 256
WRA = 0
WRB = 7
WRC = 6
ADRIO (WRD, M0)
```

```
FUN 191 (WR0)
```

Program for MICRO-EHV

```
DR4 = ADR (WR100)
WR6 = 256
DR7 = ADR (WR300)
WR9 = 256
```

```
MBMST (WR0, M0)
```

* The conversion tool does not convert FUN 191 correctly. Convert as above.

[Notes on program conversion]

There are the following differences in the operation of FUN 191 and MBMST between MICRO-EH and MICRO-EHV.

Item	MICRO-EH	MICRO-EHV
Modbus mode	RTU and ASCII can be used.	Only RTU can be used.
Timeout time setting	Specify in the S + 3 area.	Specify in advance by Modbus-TCP / RTU settings in Control Editor
Transmission format setting	Specify in the S + B area.	Specify in advance by Modbus-TCP / RTU settings in Control Editor
Transmission speed setting	Specify in the S + C area.	Specify in advance by Modbus-TCP / RTU settings in Control Editor

Name	Hitachi inverter control instruction 1

Ladder format	Number of steps		Condition code				
	Condition	Step	R7F4	R7F3	R7F2	R7F1	R7F0
			DER	ERR	SD	V	C
INV1 (s)	—	4	↕	●	●	●	●

Instruction processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	25.5	29.3	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	Data table first I/O								✓					

Note
s parameters are occupied up to s+7.

Function

- This instruction is dedicated to controlling a Hitachi-IES inverter. The target models are NE-S1, WJ200, and SJ700.
- The communication media is RS-485, and data are communicated via the Modbus-RTU protocol.
- While the instruction execution bit is ON, the PLC communicates with the inverter for data read and write.
- The inverter can be controlled by turning ON or OFF bits on the memory map image.
- The inverter status (operation state, trip, output frequency, output current) can always be monitored.

Parameter

s: Specify the first I/O of the function data table.

		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
s+0	W	EXE	SJ	IT8	IT7	IT6	IT5	IT4	IT3	IT2	IT1	-	FQL	FQE	RST	REV	FWD
s+1	W	Station number (0 to 247)															
s+2	W	Frequency setting (0.01 to 400.00 Hz) [F001]															
s+3	R	ERR	-	-	-	-	-	-	-	-	-	-	AL	ARF	RDY	DIR	RUN
s+4	R	-	-	MI8	MI7	MI6	MI5	MI4	MI3	MI2	MI1	MO6	MO5	MO4	MO3	MO2	MO1
s+5	R	Frequency monitor (0.01 to 400.00 Hz) [d001]															
s+6	R	Output current monitor (0.00 to 655.30 A) [d002]															
s+7	R	Return code (An error code is set when a communication error occurs or an exceptional response is received)															

[s+0] Control bit (write)

This flag controls the inverter. The function of each bit is as shown in the table below.

Bit	Definition	Name	Description
0	FWD	Forward operation	Set 1 to operate the inverter in the forward rotation. Set 0 to stop the inverter. If both FWD and REV are set to 1, the inverter stops.
1	REV	Reverse operation	Set 1 to operate the inverter in the reverse rotation. Set 0 to stop the inverter. If both FWD and REV are set to 1, the inverter stops.
2	RST	Reset	Set 1 to reset the inverter. When the reset request is successfully transferred, this bit is turned OFF by the system.
3	FQE	Frequency change request (edge)	When this bit is set to 1, the frequency stored in s+2 is transmitted only once to the inverter at the set timing. When the frequency setting value is successfully transferred, this bit is turned OFF by the system. Use this bit when the frequency does not need to be changed often.
4	FQL	Frequency change request (level)	When this bit is set to 1, the frequency stored in s+2 continues to be transmitted to the inverter while the bit is 1. Use this bit when the frequency needs to be changed often or the set value needs to be associated with an analog input value. Set it back to 0 if the frequency does not need to be changed.
5	—	—	Undefined.
6 to D	IT1 to 8	Intelligent input terminal 1 to 8	Set 1/0 to turn ON/OFF the corresponding intelligent input terminal. *
E	SJ	SJ700 mode	If the control target inverter is SJ700, set 1 before turning ON the instruction execution bit. For WJ200 or NE-S1, leave it as 0. Do not change this bit while the instruction execution bit is ON. Even if it is changed, it does not affect the operation.
F	EXE	Instruction execution	Set 1 to start communication with the inverter. The PLC communicates with the station number (s+1) used when this bit is set to 1. While this bit is ON, the PLC is always communicating with the inverter. Set it back to 0 to stop communication with the inverter.

* The inverter unit parameter d005 (intelligent input terminal monitor) is used to monitor the hardware input state. Therefore, please note that even if you turn ON an intelligent input terminal in terms of software using IT1 to 8 of this instruction, you cannot monitor it with the inverter unit parameter d005.

[s+1] Station number setting (write)

Set the inverter station number (Modbus slave address). The setting range is from 0 to 247. Station number 0 represents broadcast, which allows simultaneous control of all slaves, but broadcast is enabled only for write commands, so parameters s+3 to s+7 are not updated.

To use broadcast in an environment containing different inverter models, please see the notes described later.

[s+2] Output frequency setting (write)

Set the inverter output frequency (Inverter parameter: F001). The setting range is from 0.01 to 400.00 Hz, in increments of 0.01 Hz. So, in case of 50 Hz, set "5000". When you set the FQE bit or FQL bit of the s+0 area to 1 after storing the frequency setting value in this area, the frequency is transmitted.

[s+3] Status bit (read)

This flag monitors the inverter status. The definition of each bit is as shown in the table below.

Bit	Definition	Name	Description
0	RUN	Operation state	0: Stop 1: Run
1	DIR	Rotation direction	0: Forward 1: Reverse
2	RDY	Inverter ready state	0: The inverter is preparing for operation. 1: The inverter is ready for operation.
3	ARF	Frequency match	0: Stopped or during acceleration/deceleration 1: Constant speed reached
4	AL	Trip signal	0: The inverter is in a normal state. 1: The inverter is tripping.
5 to E	–	–	Undefined. Always 0.
F	ERR	Communication error	This bit is set to 1 when a communication error is detected. A detected error reason is stored as an error code into the s+6 area.

[s+4] Intelligent input/output terminal monitor (read)

This is the intelligent input/output terminal monitor area of the inverter.

This area consists of 16 bits, and the definition of each bit is as shown in the table below.

Bit	Definition	Name	Description
0 to 5	MO1 to 6	Intelligent output terminal 11 to 15, relay output monitor *	0: Corresponding intelligent output/relay terminal state OFF 1: Corresponding intelligent output/relay terminal state ON
6 to D	MI1 to 8	Intelligent input terminal 1 to 8 monitor	0: Corresponding intelligent input terminal state OFF 1: Corresponding intelligent input terminal state ON
E, F	–	–	Undefined. Always 0.

* The number of intelligent input/output terminals varies depending on the model, and the bit positions of intelligent relay terminals vary accordingly as follows:

Model	Number of intelligent input terminals	Number of intelligent output terminals	Number of intelligent relays
NE-S1	5 (MI1 to 5)	1 (MO1)	1 (MO2)
WJ200	7 (MI1 to 7)	2 (MO1 to 2)	1 (MO3)
SJ700	8 (MI1 to 8)	5 (MO1 to 5)	1 (MO6)

[s+5] Frequency monitor (Inverter parameter: d001) (read)

Stores the inverter output frequency.

The increment is 0.01 Hz. In case of 50 Hz, "5000" is stored.

[s+6] Output current monitor (Inverter parameter: d002) (read)

Stores the inverter output current value.

The increment is 0.01 A. In case of 1.23 A, "123" is stored.

[s+7] Return code (read)

Sets the instruction execution result. See the list of return codes.

Cautionary notes

- An RS-485 communication option board is required separately from this unit.
- This instruction reads and writes inverter data by sending multiple Modbus commands. Data update cycles are as shown in the table below. When the FQL bit is ON, the update cycle becomes longer as it always transmits the frequency setting command. When the number of connected units is N, the required number of instructions is N, which makes the update cycle N times longer. To run or stop multiple units simultaneously or set the frequency for them without delayed time, it is recommended that you use station number 0 (broadcast). (In the Modbus standard, station number 0 has a special meaning, and if the master sends a command by specifying station number 0, it will be sent to all slaves. This is called broadcast. However, slaves do not respond to the station number 0 command. Therefore, broadcast works only for data write commands from the master.)

Unit: [ms]

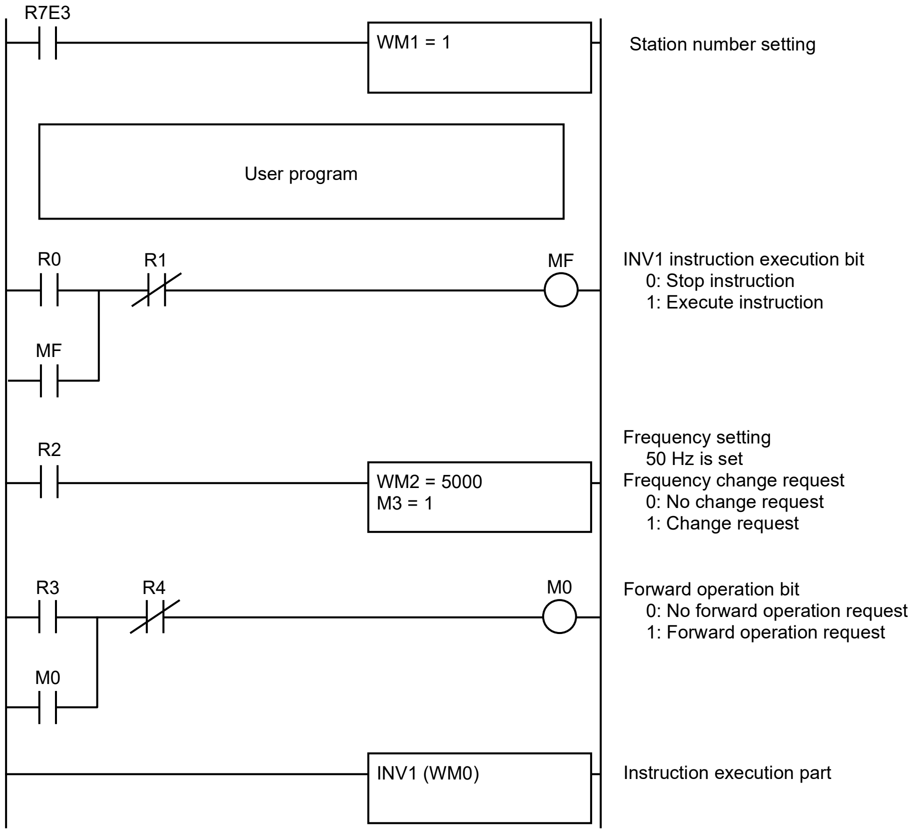
Communication speed	NE-S1		WJ200		SJ700	
	FQL: OFF	FQL: ON	FQL: OFF	FQL: ON	FQL: OFF	FQL: ON
4,800 bps	165	250	165	240	155	220
9,600 bps	95	135	90	130	95	130
19,200 bps	50	65	45	65	45	65
38,400 bps	35	45	35	45	-	-
57,600 bps	-	-	25	36	-	-
115,200 bps	-	-	20	30	-	-

- When station number 0 (broadcast) is specified, data in the s+3 to s+6 areas are not updated because monitor communication is not performed.
- If the SJ700 and WJ200/NE-S1 coexist, station number 0 (broadcast) does not function correctly. When the SJ bit is 0, only the WJ200 and NE-S1 run but not the SJ700. Conversely, when the SJ bit is 1, only the SJ700 runs but not the WJ200 and NE-S1. This is because the SJ700 has the same Modbus address as the WJ200/NE-S1 but has a different byte order when reading or writing multiple coils.
- When communication is unstable, setting the communication wait time with the special internal output WRF0E0 may improve the communication (however, the data update cycle becomes longer). For details on the setting method, see the notes on the MBMST instruction.
- The INV1 instruction and Modbus master instruction (MBMST instruction) can be used together.
- To use this instruction, set or change the following inverter parameters according to the system configuration.

Item	Function code	Data/data range	Description	Default
Frequency instruction select	A001	03	Modbus communication	02
Operation instruction select	A002	03	Modbus communication	02
OPE/Modbus select *1	C070	01	Modbus RS-485 (NE-S1 only)	
Communication speed select	C071	03 to 10	Use the same setting as the MICRO-EHV.	05
Communication station number select	C072	1 to 247	Modbus slave address	1
Communication bit length select *2	C073	8	8 bits (SJ700 only)	7
Communication parity select	C074	00 to 02	Use the same setting as the MICRO-EHV.	00
Communication stop bit select	C075	1 to 2	Use the same setting as the MICRO-EHV.	1
Communication error behavior select	C076	00 to 04	Use the system requirement value.	02
Communication timeout time	C077	0.00 to 99.99	Use the system requirement value.	0.00
Communication wait time	C078	0 to 1,000	Use the system requirement value.	0
Communication mode select *2	C079	01	Modbus-RTU method (SJ700 only)	00

*1 This setting item is available only for the NE-S1.
*2 This setting item is available only for the SJ700.

Program example

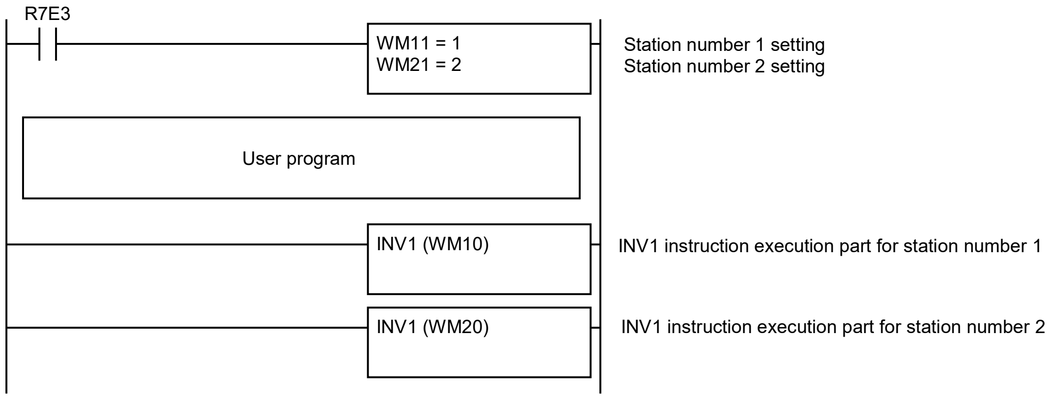


[Program description]

- The destination station number is set to 1 in the first scan.
- The instruction execution bit (MF in this example) is set to remain ON until the instruction stops (R1 is turned ON) using self-latching.
- When the frequency setting condition is met (R2 is turned ON), the setting frequency is set in the frequency setting area (WM2 in this example), and the frequency change request bit (M3 in this example) is turned ON.
- When the forward operation start condition is met (R3 is turned ON), the forward operation bit (M0 in this example) is turned ON to start the inverter in the forward operation. The forward operation bit is set to remain ON until the instruction stop (including reverse operation) (R4 is turned ON) using self-latching.

< Multiple-unit connection >

To connect multiple units, write as many instructions as the number of connected units and specify the destination of the target INV1 instruction in the s+1 station number setting. The maximum number of INVn/OMSTn/OCTPn instructions that can be executed simultaneously is 32.



Return Code

The following lists the return codes to be stored into the [s+7] area after INV1 execution.

(1) When the PLC is in a normal state or has a communication or instruction error

Return code	Name	Description	Solution
H0000	Normal	Transmission is in a normal state.	-
H0021*	Range check error	The end of parameter s is outside the I/O range.	Set the area of parameter s within the valid range.
H0030	Timeout	Transmission did not end within the specified time.	Set a larger value or change the processing.
H0040	Data capacity exceeded	• The received data has exceeded 1,028 bytes. • There is no available space for received data.	Check the number of coils/registers of received data.
H0041	Parity error Framing error Overrun error	A parity error, framing error, or overrun error occurred in communication processing.	Check the transmission line of the optional port and the data format.
H0044	Simultaneous start limit exceeded	More than the maximum number of INVn, OMSTn, and OCTPn instructions was started (EXE bit was turned ON).	The maximum number of INVn/OMSTn/OCTPn instructions that can be started simultaneously is 32.
H0045	Parameter error	• The station number setting is outside the range or duplicated. • The destination station number was changed during communication. • The OMSTn or OCTPn instruction and parameter s are duplicated.	• The station number must be set between 0 and 247. • Reset the station number to the one used when the communication started. • Change the area of parameter s.
H0046	Port specification error	• The INV1 instruction was started when the optional port was not specified as the Modbus master. • No optional board is installed.	Check the port specification.
H0071	Received data error	• The received data is invalid. • The number of received bytes, except for the size of header, check code, and trailer, has exceeded 509.	Check if the slave device is compatible with Modbus.
H0072	CRC error	A CRC check error occurred.	Check the slave Modbus mode.

* The return code of the range check error is H21, but if the return code area is outside the I/O range, the return code may not be displayed.

(2) When the PLC receives an exceptional response

The exception code from the slave device is stored into the low byte, and the write data block ID is stored into the high byte.

Return code	Name	Description	Solution
Hxx01*	Illegal function	The slave device does not support the requested function.	Check if the slave device is compatible with this instruction.
Hxx02*	Illegal data access	The specified data address does not exist in the slave device.	Check if the slave device is compatible with this instruction.
Hxx03*	Data format error	This format does not allow the slave device to accept the specified data.	Check if the slave device is compatible with this instruction.
Hxx21*	Invalid data	The specified data is outside the setting range.	Check the write data.
Hxx22*	Slave error	The inverter is in a situation that does not allow the function.	For details, see the inverter manual.
Hxx23*	Illegal write	The write was attempted to a read-only register (coil).	Check if the slave device is compatible with this instruction.

* xx represents the write data block ID.

ID	Write parameter
01	FWD, REV, RST, IT1 to IT8
02	Frequency setting

Name	Oriental Motor stepping motor control instruction 1

Ladder format	Number of steps		Condition code				
	Condition	Step	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
OMST 1 (s)	—	3	↕	●	●	●	●

Instruction processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	24.83	20.6	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT, TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	Data table first I/O								✓					

Note
s parameters are occupied up to s+1F (32 words).

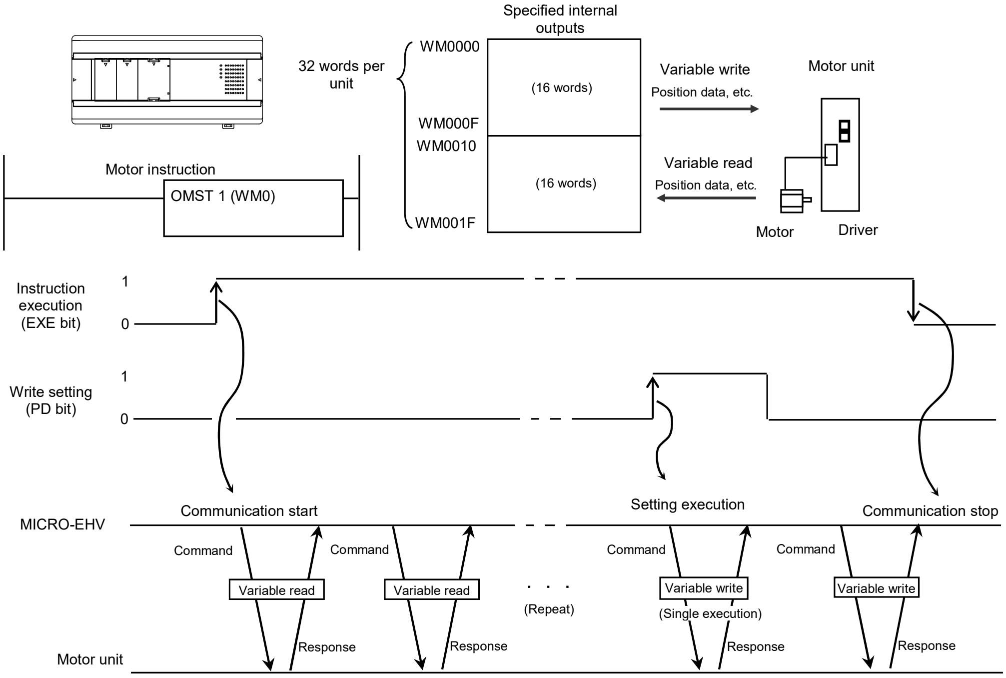
Function

- This instruction performs Modbus communication with Oriental Motor stepping motor unit (αSTEP high-efficiency AR series, etc.) using the communication port 2 of the optional board.

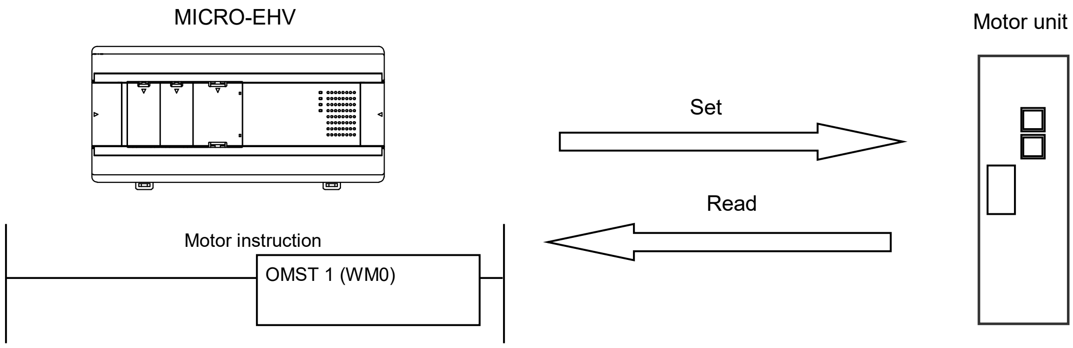
  [Compatible units] (1) AR Series FLEX AC power input built-in controller type

  (2) AR Series FLEX DC power input built-in controller type

  (3) RKII Series FLEX AC power input built-in controller type
- The topology is 1:1 or 1:N. The maximum number of connected motor units is 31. One instruction of this type is required to commutate with each motor unit.
- While the instruction execution bit is ON, Modbus communication with motor units is performed.
- Motor units can be controlled by turning ON or OFF bits on the memory map image. The internal output area, which communicates with motor units, can be specified in the argument I/O of the instruction.



Command to the motor unit issued by this instruction



(1) Motor unit setting

No.	Command type	Register name	Modbus address / setting range
1	Operation command	Driver input instruction (low) (Network I/O NET-IN0 to 15) Set the driver input instruction.	H007D / [Default] or user setting
2	Parameter R/W command	Operation data position data No. 0 to 7	H0400, H0401 to H40E, H40F / -8,388,608 to 8,388,607
3		Operation data operation speed No. 0 to 7	H0480, H0481 to H048E, H48F / 0 to 1,000,000 Hz
4	Maintenance command	Alarm reset Resets the alarm that occurred. Some alarms cannot be reset depending on the type.	H0181 / A write from 0 to 1 resets the alarm.

(2) Motor unit read

No.	Command type	Register name	Modbus address/setting range
1	Operation command	Driver output instruction (low) Network I/O NET-OUT0 to 15 Reads the driver output status.	H007F/ [Default] or user setting
2	Monitor command	Instruction position Indicates the instruction position.	H00C6, H00C7/ -2,147,483,648 to 2,147,483,647 step
3		Instruction speed Indicates the current instruction speed.	H00C8, H00C9/ -4,500 to 4,500 r/min +: Forward rotation, -: Reverse rotation, 0: Stop
4		Feedback position Indicates the feedback position.	H00CC, H00CD/ -2,147,483,648 to 2,147,483,647 step
5		Feedback speed Indicates the feedback speed.	H00CE, H00CF/ -4,500 to 4,500 r/min
6		Alarm Indicates the alarm code that occurred.	H0081/ H00 to HFF

This instruction was created according to the Oriental Motor AR Series FLEX Built-in Controller Type User's Manual (HM-60223). It is hereinafter referred to as Oriental Manual (HM-60223).

Motor unit communication status items to be monitored by this instruction

No.	Item	Description	Note
1	Communication status	Inverts the bit between 0 and 1 every time communication with the motor unit is completed.	Use it as a reference to check the communication status.
2	Communication cycle	Stores the cycle (×1 ms) at which the MICRO-EHV communicates with the motor unit.	Use it as a guide to check the communication interval. (Default: 0)

Parameter

The OMST1 instruction uses the internal output areas of parameter s shown in the table below. Specify the first I/O of the instruction data table for s.

		Data	Description
s+0	W	Control bit area	This is the control bit area of this instruction. For details, see (1) [s+0] Control bit area.
s+1	W	Station number (0 to 247)	Set the station number (Modbus slave address) of the motor unit you want to communicate with.
s+2	W	Driver input instruction write data *1 [007D]	This is the network I/O driver input signal. For details, see (3) [s+2] Driver input signal write.
s+3	W	Setting operation data No. specification	Specify operation data No. used to set values written into s+4 to s+5 and s+6 to s+7. Setting range: 0 to 7 (Any other value is treated as 0.)
s+4	W	Operation data No. n position data (low)*2 [040*]	Sets the position data (travel distance) of the operation data No. specified in s+3. Setting range: -8,388,608 to 8,388,607
s+5	W	Operation data No. n position data (high)*2 [040*]	
s+6	W	Operation data No. n operation speed (low)*2 [048*]	Sets the operation speed of the operation data No. specified in s+3. Setting range: 0 to 1,000,000 Hz
s+7	W	Operation data No. n operation speed (high)*2 [048*]	
s+8	W	Alarm reset [0181]	Resets the alarm that occurred in the motor unit. Setting range: 0, 1 (A write from 0 to 1 resets the alarm.)
s+9 -15		-	(Undefined)
s+16	R	Control response bit area	This is a response to the control bit area. For details, see (2) [s+16] Control response bit area.
s+17	R	Instruction return code	Sets the instruction execution result. See the list of return codes.
s+18	R	Communication status	Inverts the lowest bit between 0 and 1 every time communication is completed. You can check the communication status.
s+19	R	Communication cycle	Stores the communication cycle in units of milliseconds.
s+20	R	Driver output instruction read data *1 [007F]	This is the network I/O driver output signal. For details, see (4) [s+20] Driver output signal data.
s+21	R	Instruction position (low) *2 [00C7]	Reads the instruction position. -2,147,483,648 to 2,147,483,647 step
s+22	R	Instruction position (high) *2 [00C6]	
s+23	R	Instruction speed (low) *2 [00C9]	Reads the instruction speed. -4,500 to 4,500 r/min, +: Forward rotation, -: Reverse rotation, 0: Stop
s+24	R	Instruction speed (high) *2 [00C8]	
s+25	R	Feedback position (low) *2 [00CD]	Reads the feedback position. -2,147,483,648 to 2,147,483,647 step
s+26	R	Feedback position (high) *2 [00CC]	
s+27	R	Feedback speed (low) *2 [00CF]	Reads the feedback speed. -4,500 to 4,500 r/min
s+28	R	Feedback speed (high) *2 [00CE]	
s+29	R	Alarm (low) [0081]	Reads the alarm code (00 to HFF).
s+30 -31		-	(Undefined)

W: Write, R: Read [ ]: Motor unit variable area address (hexadecimal)

*1: Use bit access.

*2: Use double word access. Specify a signed integer (.S) for negative value data. If the data is accessed in word, the high and the low are reversed from those of the motor unit. Data on the Modbus line are swapped by this instruction.

(1) [s+0] Control bit area

This is the control bit manipulation area for the motor unit. This area consists of 16 bits, and the function of each bit is as shown in the table below. To set each parameter, turn ON the corresponding bit.

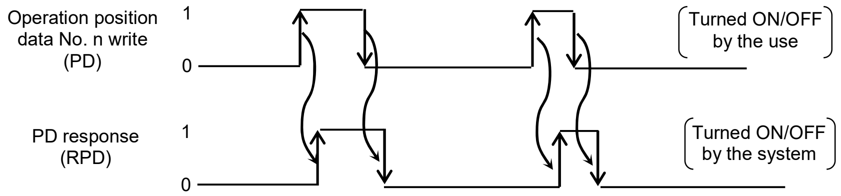
	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
[s+0]	EXE	ECR	–	–	ALM	FBV	FBP	CV	CP	–	–	–	–	ALR	PV	PD

Bit	Definition	Name	Setting data
0	PD	Operation No. n position data setting	Writes the data set in [s+4] to [s+5] into the motor unit as the position data of the operation No. specified in [s+3] at the rising edge of this bit.
1	PV	Operation No. n speed data setting	Writes the data set in [s+6] to [s+7] into the motor unit as the speed data of the operation No. specified in [s+3] at the rising edge of this bit.
2	ALR	Alarm reset	Writes the data set in [s+8] into the motor unit as the alarm reset at the rising edge of this bit.
3 to 6	-	-	(Undefined)
7	CP	Operation instruction position read	Reads the instruction position from the motor unit while this bit is ON. The read data is stored into [s+21] to [s+22]. Position data: -2,147,483,648 to 2,147,483,647 step
8	CV	Operation instruction speed read	Reads the instruction speed from the motor unit while this bit is ON. The read data is stored into [s+23] to [s+24]. Speed data: -4,500 to 4,500 r/min, +: Forward rotation, -: Reverse rotation, 0: Stop
9	FBP	Feedback position read	Continues to read the feedback position from the motor unit while this bit is ON. The read data is stored into [s+25] to [s+26]. Feedback position data: -2,147,483,648 to 2,147,483,647 step
A	FBV	Feedback speed read	Continues to read the feedback speed from the motor unit while this bit is ON. The read data is stored into [s+27] to [s+28]. Speed data: -4,500 to 4,500 r/min, +: Forward rotation, -: Reverse rotation, 0: Stop
B	ALM	Alarm read	Reads the instruction speed from the motor unit at the rising edge of this bit. Read data is stored into [s+29].
C to D	-	-	(Undefined)
E	ECR	Error clear	Sets the ERR bit of [s+16] to 1 when a communication error is detected. When this bit is set to 1, the ERR bit of [s+16] and the error code stored in [s+17] are cleared to 0. The operation relationship with the ERR bit is as shown in the figure below. 
F	EXE	Instruction execution	Set 1 to start communication with the motor unit. The PLC communicates with the station number (s+1) used when this bit is set to 1. While this bit is ON, communication with the motor unit is performed. Set it back to 0 to stop communication with the inverter.

(2) [s+16] Control response bit area

The response is stored into this area when the control request manipulated in [s+0] is successfully transmitted. The position of the control response bit is the same as that of the control bit. If a data error occurs, the ERR bit is turned ON without changing the bit responding to the request.

	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
[s+16]	REX	ERR	–	–	RAL	RFV	RFP	RCV	RCP	–	–	–	–	RAR	RPV	RPD



Bit	Definition	Name	Setting data
0	RPD	Operation No. n position data setting response	Turns ON when the write request of the position data setting of the operation No. specified in [s+3] is successfully completed. If it terminates with an error, ERR is turned ON without changing this bit. If PD is found to be turned OFF, this bit is also turned OFF.
1	RPV	Operation No. n speed data setting response	Turns ON when the write request of the speed data setting of the operation No. specified in [s+3] is successfully completed. If it terminates with an error, ERR is turned ON without changing this bit. If PV is found to be turned OFF, this bit is also turned OFF.
2	RAR	Alarm reset response	Turns ON when the write request of the alarm reset is successfully completed. If it terminates with an error, ERR is turned ON without changing this bit. If ALR is found to be turned OFF, this bit is also turned OFF.
3 to 6	-	-	(Undefined)
7	RCP	Operation instruction position read response	Turns ON while the read request of the operation instruction position is transmitted. If it terminates with an error, ERR is turned ON without changing this bit. If CP is found to be turned OFF, this bit is also turned OFF.
8	RCV	Operation instruction speed read response	Turns ON while the read request of the operation instruction speed is transmitted. If it terminates with an error, ERR is turned ON without changing this bit. If CV is found to be turned OFF, this bit is also turned OFF.
9	RFP	Feedback position read response	Turns ON while the read request of the feedback position is transmitted. If it terminates with an error, ERR is turned ON without changing this bit. If FBP is found to be turned OFF, this bit is also turned OFF.
A	RFV	Feedback speed read response	Turns ON while the read request of the feedback speed is transmitted. If it terminates with an error, ERR is turned ON without changing this bit. If FBV is found to be turned OFF, this bit is also turned OFF.
B	RAL	Alarm read response	Turns ON when the read request of the alarm is successfully completed. If it terminates with an error, ERR is turned ON without changing this bit. If ALM is found to be turned OFF, this bit is also turned OFF.
C to D	-	-	(Undefined)
E	ERR	Error	Turns ON when a communication error is detected. If ECR is found to be turned ON, this bit is turned OFF.
F	REX	Instruction execution response	Turns ON while EXE is ON. If EXE is found to be turned OFF, this bit is also turned OFF.

(3) [s+2] Driver input instruction write

This is a write to the network I/O driver input signal defined in the motor unit.

When any bit change (0 to 1 or 1 to 0) of bit 0 to 15 is detected, this register value is transmitted to the motor unit.

[AR series driver input (NET-IN0 to 15)]

Address	Address description ([ ]: Default motor unit value)															
H007D	bit15	bit14	bit13	bit12	bit11	Bit10	bit9	bit8	bit7	Bit6	bit5	bit4	bit3	bit2	bit1	bit0
	[RVS]	[FWD]	[-JOG]	[+JOG]	[SSTART]	[MS2]	[MS1]	[MS0]	[-]	[FREE]	[STOP]	[HOME]	[START]	[M2]	[M1]	[M0]

[RKII series driver input (NET-IN0 to 15)]

Address	Address description ([ ]: Default motor unit value)															
H007D	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	Bit6	bit5	bit4	bit3	bit2	bit1	bit0
	[RVS]	[FWD]	[-JOG]	[+JOG]	[SSTART]	[MS2]	[MS1]	[MS0]	[ALM-RST]	[FREE]	[STOP]	[HOME]	[START]	[M2]	[M1]	[M0]

Make sure that the network I/Os (driver inputs/outputs) are assigned beforehand to the motor unit using the motor unit's data setting software.

Assigned No.	Signal name	Function	Assigned No.	Signal name	Function
0	Unused	Not used	32	R0	General-purpose signal. Used for RS-485 communication.
1	FWD	Start the continuous operation in + direction	33	R1	
2	RVS	Start the continuous operation in - direction	34	R2	
3	HOME	Start the homing return operation	35	R3	
4	START	Start the positioning operation	36	R4	
5	SSTART	Start the progressive positioning operation	37	R5	
6	+JOG	Start the JOG operation in + direction	38	R6	
7	-JOG	Start the JOG operation in - direction	39	R7	
8	MS0	Start the direct positioning operation of the operation data No. specified in the I/O parameter	40	R8	
9	MS1		41	R9	
10	MS2		42	R10	
11	MS3		43	R11	
12	MS4		44	R12	
13	MS5		45	R13	
16	FREE		46	R14	
17	C-ON	Energize or de-energize the motor (AR series only)	47	R15	
	AWO	Energize or de-energize the motor (RKII series only)	48	M0	Select the operation data No.
18	STOP	Stop the motor	49	M1	
24	ALM-RST	Reset the current alarm (RKII series only)	50	M2	
25	P-PRESET	Perform the position preset (RKII series only)	51	M3	
27	HMI	Unlock the OPX-2A/MEXE02 functionality	52	M4	
-	–	–	53	M5	

For details, see the motor unit manual.

(4) [s+20] Driver output instruction read data

This is the network I/O driver output signal defined in the motor unit. This data is always read by turning ON the EXE bit of [s+0] Control bit area.

[AR series driver output (NET-OUT0 to 15)]

Address	Address description ([ ]: Default motor unit value)															
H007F	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	[TLC]	[END]	[MOVE]	[TIM]	[AREA3]	[AREA2]	[AREA1]	[S-BSY]	[ALM]	[WNG]	[READY]	[HOME-P]	[START_R]	[M2_R]	[M1_R]	[M0_R]

[RKII series driver output (NET-OUT0 to 15)]

Address	Address description ([ ]: Default motor unit value)															
H007F	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	[STEPOUT]	[-]	[MOVE]	[TIM]	[AREA3]	[AREA2]	[AREA1]	[S-BSY]	[ALM]	[WNG]	[READY]	[HOME-P]	[START_R]	[M2_R]	[M1_R]	[M0_R]

Make sure that the network I/Os (driver inputs/outputs) are assigned beforehand to the motor unit using the motor unit's data setting software.

[Signals that can be assigned to the network I/O driver output instruction]

Assigned No.	Signal name	Function	Assigned No.	Signal name	Function
0	Unused	Not used	45	R13	General-purpose signal. Output the R0 to R15 state
1	FWD_R	Output the response to FWD	46	R14	
2	RVS_R	Output the response to RVS	47	R15	
3	HOME_R	Output the response to HOME	48	M0_R	Output the response to M0 to M5
4	START_R	Output the response to START	49	M1_R	
5	SSTART_R	Output the response to SSTART	50	M2_R	
6	+JOG_R	Output the response to +JOG	51	M3_R	
7	-JOG_R	Output the response to -JOG	52	M4_R	
8	MS0_R	Output the response to MS0	53	M5_R	
9	MS1_R	Output the response to MS1	60	+LS_R	Output the response to +LS
10	MS2_R	Output the response to MS2	61	-LS_R	Output the response to -LS
11	MS3_R	Output the response to MS3	62	HOMES_R	Output the response to HOMES
12	MS4_R	Output the response to MS4	63	SLIT_R	Output the response to SLIT
13	MS5_R	Output the response to MS5	65	ALM	Output the driver alarm
16	FREE_R	Output the response to FREE	66	WNG	Output the driver warning
17	C-ON_R	Output the response to C-ON (AR series only)	67	READY	Output when driver operation is ready
	AWO_R	Output the response to AWO (RKII series only)	68	MOVE	Output when the motor is operating
18	STOP_R		69	END	Output when positioning operation is completed (AR series only)
32	R0	General-purpose signal. Output the R0 to R15 state	70	HOME-P	Output when the motor is at the home position
33	R1		71	TLC	The load is outside the motor torque range (AR series only)
34	R2		72	TIM	Output every time the motor output axis rotates by 7.2°
35	R3		73	AREA1	Output when the motor is within area 1
36	R4		74	AREA2	Output when the motor is within area 2
37	R5		75	AREA3	Output when the motor is within area 3
38	R6		80	S-BSY	Output when the driver is performing internal processing
39	R7		82	MPS	The main power is ON (except for the AR series DC power input)
40	R8		83	STEPOUT	Output when a deviation error occurs (RKII series only)
41	R9		84	OH	Output when an overheat warning occurs
42	R10		85	ZSG	Output the response to the encoder ENC-Z
43	R11		86	MBC	Output the control state of the electromagnetic brake
44	R12				

For details, see the motor unit manual.

Cautionary notes

- When station number 0 is specified, data in the s+16 to s+31 areas do not change because monitor communication is not performed.
- To use this instruction, set or change the following motor unit functions according to the system configuration.

[AR serial communication settings]

Item	Description
Protocol	Turn ON No. 2 of the function setting switch (SW4) (Modbus protocol).
Unit number	Set with No. 1 of the unit number setting switch (ID) and function setting switch (SW4).
Communication speed	Communication speed setting (SW2): 9600/19200/38400/57600/115,200 bps
Termination resistor	Turn ON the termination resistor setting switch for the terminating unit.
Transmission wait time	10 ms
Transmission speed	115,200 bps
Transmission format	8-bit even-parity 1 stop bit

[RKII serial communication settings]

Item	Description
Protocol	Turn ON No. 2 of the function setting switch (SW1) (Modbus protocol).
Unit number	Set with No. 1 of the unit number setting switch (ID) and function setting switch (SW1).
Communication speed	Communication speed setting (SW2): 9600/19200/38400/57600/115,200 bps
Termination resistor	Turn ON the termination resistor setting switch for the terminating unit.
Transmission wait time	10 ms
Transmission speed	115,200 bps
Transmission format	8-bit even-parity 1 stop bit

If higher response performance is required, you can shorten the communication cycle by changing the transmission wait time (Minimum: 0 ms).

(For details on the setting information, see the motor unit manual.)

- To connect multiple axes of the motor unit using communication of 57,200 bps or more, a timeout error may be detected because the motor unit does not respond to the request from the MICRO-EHV. In such a case, set an appropriate value in the special internal output WRF0E0 (wait time) to adjust the transmission cycle.
- Driver input instruction write, operation data No. n position data write, and operation data No. n operation speed data write are supported by the motor unit's group transmission function.

Program example

Instruction operation example

Used map

	I/O No.	R/W	Data	Motor side address	Setting data	Control bit
0	WM000	Write	Control bit area	-	Refer to the following figure	
1	WM001		Station number	-	Slave address (1 to 31)	-
2	WM002		Driver input instruction	H007D	Refer to the following figure	
3	WM003		Setting operation No. specification	-	0 to 7	-
4	DM004.S		Position data	H040x	Operation data No. n position data (-8,388,608 to 8,388,607 step)	PD
5						
6	DM006.S		Speed data	H048x	Operation data No. n speed data (0 to 1,000,000 Hz)	PV
7						
8	WM008		Alarm reset	H0181	When this is set to 1, the alarm is cleared.	ALR
9	WM009		-	-	-	-
10	WM00A		-	-	-	-
11	WM00B		-	-	-	-
12	WM00C		-	-	-	-
13	WM00D		-	-	-	-
14	WM00E		-	-	-	-
15	WM00F		-	-	-	-
16	WM010	Read	Control response bit area	-	Refer to the following figure	-
17	WM011		Instruction return code	-	-	-
18	WM012		Communication status	-	-	-
19	WM013		Communication cycle	-	-	-
20	WM014		Driver output instruction	H007F	Refer to the following figure	-
21	DM015.S		Position data	H00C6	Current position data (-2,147,483,648 to 2,147,483,647 step)	CP
22						
23	DM017.S		Speed data	H00C8	Current speed data (-4,500 to +4,500 r/min)	CV
24						
25	DM019.S		Feedback back position data	H00CC	Feedback position data (-2,147,483,648 to 2,147,483,647)	FBP
26						
27	DM01B.S		Feedback speed data	H00CE	Current feedback speed data (-4,500 to +4,500 r/min) AR series only (both AC input and DC input) (Not for RKII series because there is no register address of this monitor)	FBV
28						
29	WM01D		Alarm code	H0081	-	ALM
30	WM01E		-	-	-	-
31	WM01F		-	-	-	-

The assigned signal values of the driver input instruction and driver output instruction are the default. You can change the values using data setting software.

Control bit area

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
EXE	ECR	-	-	ALM	FBV	FBP	CV	CP	-	-	-	-	ALR	PV	PD

Driver input instruction

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
RVS	FWD	-JOG	+JOG	SSTART	MS2	MS1	MS0	-	FREE	STOP	HOME	START	M2	M1	M0

Control response bit area

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
REX	ERR	-	-	RAL	RFV	RFP	RCV	RCP	-	-	-	-	RAR	RPV	RPD

Driver output instruction

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
TLC	END	MOVE	TIM	AREA3	AREA2	AREA1	S-BSY	ALM	WNG	READY	HOME-P	START_R	M2_R	M1_R	M0_R

(1) Positioning operation example

A) Positioning operation

No.	Ladder program processing	Description
1	Check that (M145) [READY] is ON	Check that [READY (operation ready)] is ON
2	Turn ON any one or more of (M20/M21/M22) [M0/M1/M2] or all OFF	Select the operation data No.
3	Turn ON (M23) [READY]	**Positioning operation start** Turn ON [START (positioning operation start)]
4	Check that (M145) [READY] is OFF	Check that [READY (operation ready)] is OFF
5	Turn OFF (M23) [START]	Turn OFF [START (positioning operation start)]
6	Check that (M145) [READY] is ON	**Positioning operation completion** Check that [READY (operation ready)] is ON

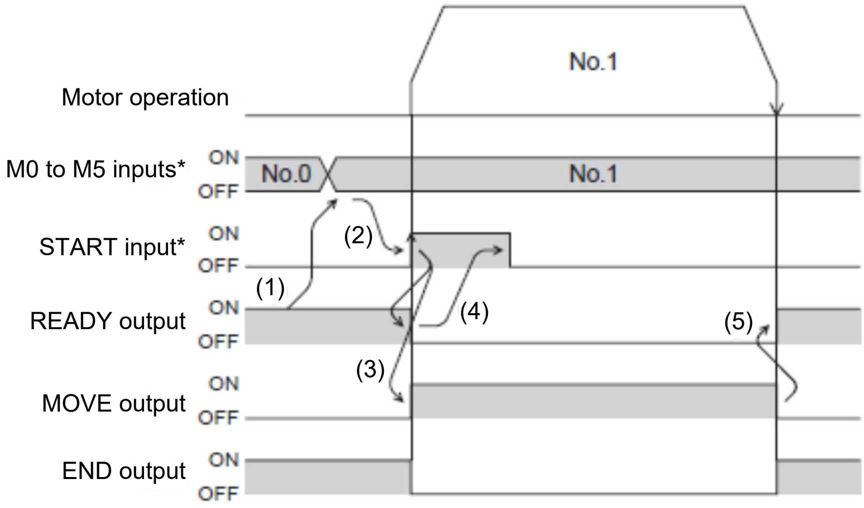
(1) Check that the READY output is ON.

(2) Select the operation data No. with the MO to M5 inputs and turn ON the START input.

(3) The motor starts the positioning operation.

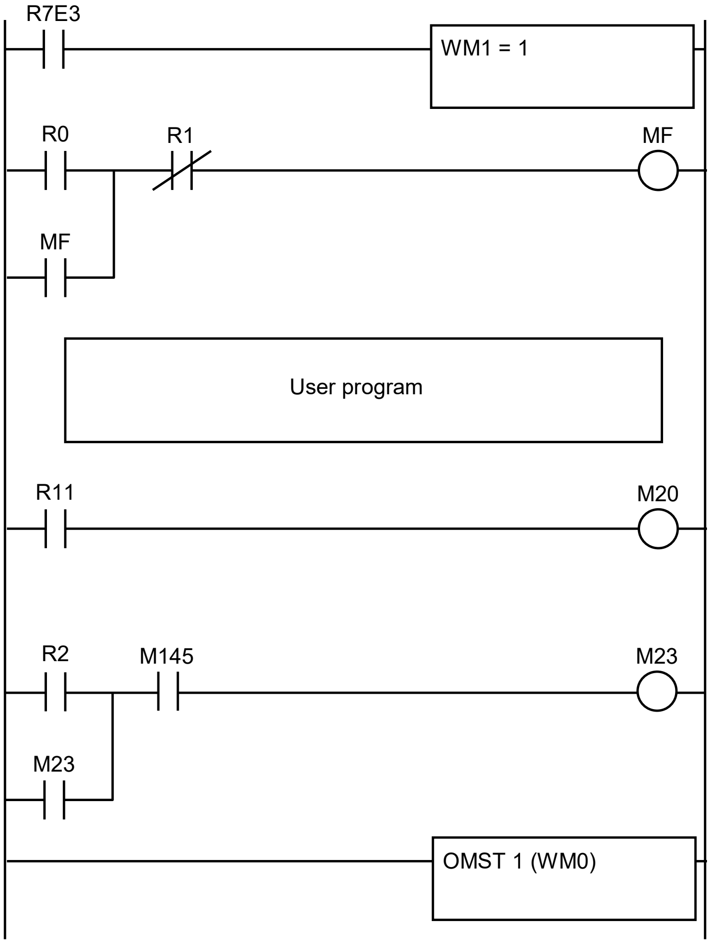
(4) Check the the READY output is turned OFF and turn OFF the START input.

(5) When the positioning operation is completed, the READY output is turned ON.



(Oriental Manual (HM-60223) P.3-9)

* The M0 to M5 inputs are used in the above example, but the M0 to M2 inputs are supported in this instruction.



Station number setting

R0: OMST1 instruction execution condition
R1: OMST1 instruction stop condition

R11: Operation No. 1 specification condition
M20: M0 input
M21: M1 input
M22: M2 input
(M0, M1, M2) = (1, 0, 0)
→ Operation No. 1

R2: Positioning operation start condition
M23: START input
M145: READY output
When the condition is met, the START input is turned ON.
When the READY output is turned OFF, the START input is turned OFF.

Instruction execution part

[Program description]

- After RUN, the station number of the motor driver to be connected in the first scan is set, and when the execution condition is met (R0 is turned ON), the OMST1 instruction is executed to communicate with the motor driver.
- The instruction execution bit (MF) is set to remain ON until the instruction stops (R1 is turned ON) using self-latching.
- The operation No. is set to 1 (R11 is turned ON) by specifying the M0 to M2 inputs (M20 to M22) to the motor driver.
- While the READY output (M145) from the motor driver is ON, if the positioning operation start condition (R2) is met, the START input (M23) to the motor driver is output, and the stepping motor starts the positioning operation.

B) Homing return

No.	Ladder program processing	Description
1	Check that (M145) [READY] is ON	Check that [READY (operation ready)] is ON
2	Turn ON (M24) [HOME]	**Homing return operation start** Turn ON [HOME (Homing return operation start)]
3	Check that (M145) [READY] is OFF	Check that [READY (operation ready)] is OFF
4	Turn OFF (M24) [HOME]	Turn OFF [HOME (Homing return operation start)]
5	Check that (M144) [HOME-P] is ON	**Homing return operation completion** Check that [HOME-P (motor home position)] is ON

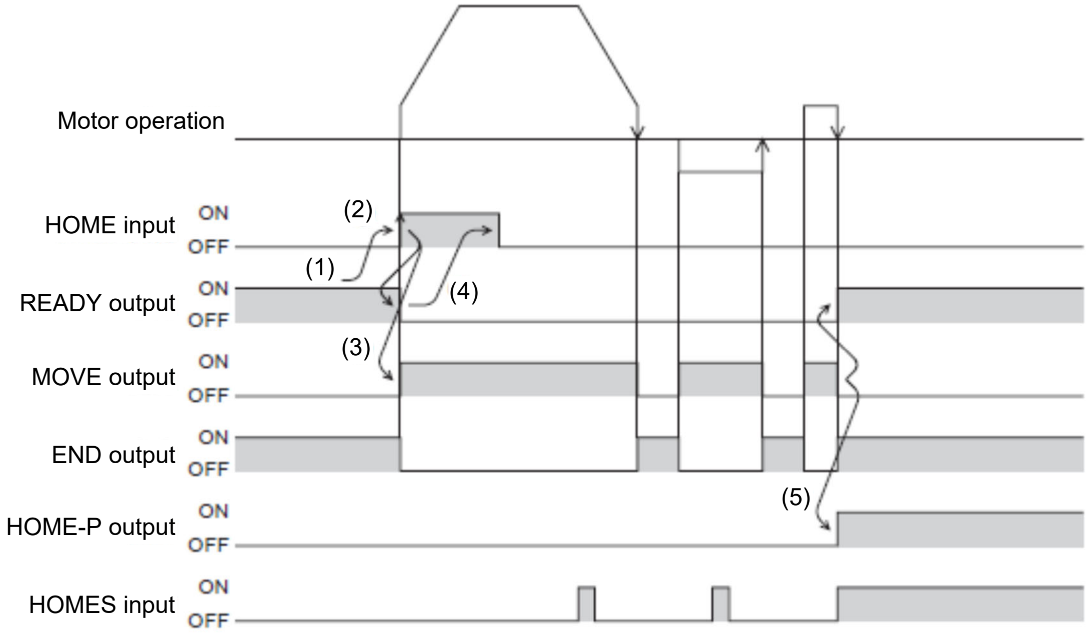
(1) Check that the READY output is ON.

(2) Turn ON the HOME input.

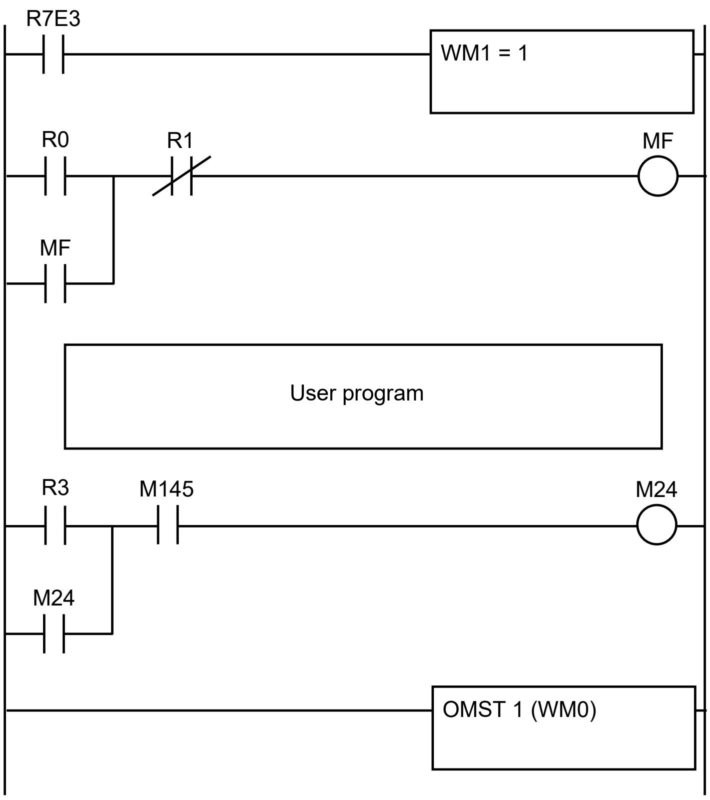
(3) The homing return operation starts.

(4) Check the the READY output is turned OFF and turn OFF the HOME input.

(5) When the homing return operation is completed, the HOME-P output is turned ON.



(Oriental Manual (HM-60223) P.3-20)



Station number setting

R0: OMST1 instruction execution condition
R1: OMST1 instruction stop condition

R3: Homing return operation start condition
M23: HOME input
M145: READY output
When the condition is met, the HOME input is turned ON.
When the READY output is turned OFF, the HOME input is turned OFF.

Instruction execution part

[Program description]

- After RUN, the station number of the motor driver to be connected in the first scan is set, and when the execution condition is met (R0 is turned ON), the OMST1 instruction is executed to communicate with the motor driver.
- The instruction execution bit (MF) is set to remain ON until the instruction stops (R1 is turned ON) using self-latching.
- While the READY output (M145) from the motor driver is ON, if the homing return operation start condition (R3) is met, the HOME input (M24) to the motor driver is output, and the stepping motor starts the homing return operation.

C) Direct positioning

No.	Ladder program processing	Description
1	Check that (M145) [READY] is ON	Check that [READY (operation ready)] is ON
2	Turn ON any of (M28/M29/M2A) [MS0/MS1/MS2]	**Positioning operation start** Start the positioning operation with operation data No. 0/1/2
3	Check that (M145) [READY] is OFF	Check that [READY (operation ready)] is OFF
4	Turn OFF (M28/M29/M2A) [MS0/MS1/MS2]	
5	Check that (M145) [READY] is ON	**Positioning operation completion** Check that [READY (operation ready)] is ON

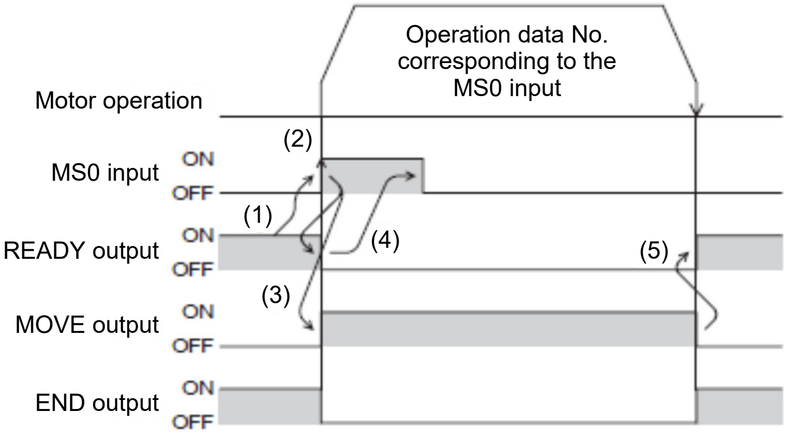
(1) Check that the READY output is ON.

(2) Turn ON the MS0 input.

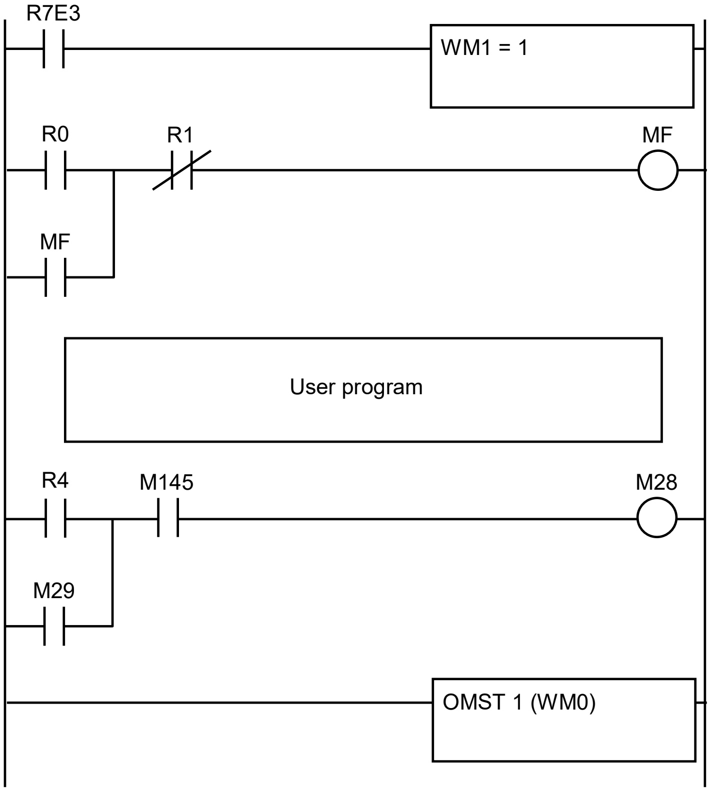
(3) The motor starts the positioning operation.

(4) Check the the READY output is turned OFF and turn OFF the MS0 input.

(5) When the positioning operation is completed, the READY output is turned ON.



(Oriental Manual (HM-60223) P.3-20)



Station number setting

R0: OMST1 instruction execution condition
R1: OMST1 instruction stop condition

R4: Operation No. 0 direct positioning operation start condition
M28: MS0 input
M145: READY output
When the condition is met, the MS1 input is turned ON.
When the READY output is turned OFF, the MS1 input is turned OFF.

Instruction execution part

[Program description]

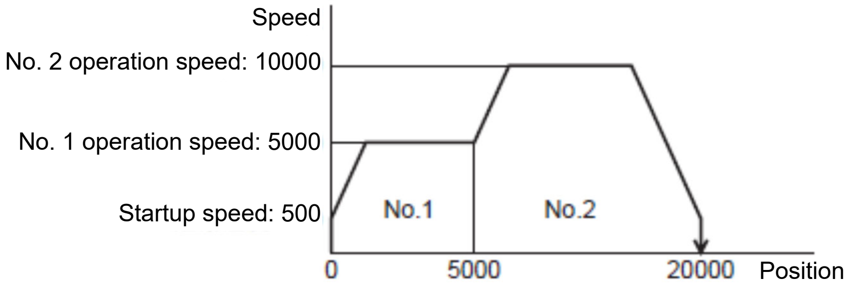
- After RUN, the station number of the motor driver to be connected in the first scan is set, and when the execution condition is met (R0 is turned ON), the OMST1 instruction is executed to communicate with the motor driver.
- The instruction execution bit (MF) is set to remain ON until the instruction stops (R1 is turned ON) using self-latching.
- While the READY output (M145) from the motor driver is ON, if operation No. 0 direct positioning operation start condition (R4) is met, the MS0 input (M28) to the motor driver is output, and the stepping motor starts operation No. 0 positioning operation.

D) Connected run

No.	Ladder program processing	Description
1	Check that (M145) [READY] is ON	Check that [READY (operation ready)] is ON
2	Turn ON (M20) [M0]	Select operation data No. 1
3	Turn ON (M23) [READY]	**Positioning operation start** Turn ON [START (positioning operation start)]
4	Check that (M145) [READY] is OFF	Check that [READY (operation ready)] is OFF
5	Turn OFF (M23) [START]	Turn OFF [START (positioning operation start)]
6	-	Complete the operation data No. 1 positioning operation
7	-	Start the operation data No. 2 positioning operation
8	Check that (M145) [READY] is ON	**Positioning operation completion** Check that [READY (operation ready)] is ON

Operation data	Position	Operation speed	Acceleration	Deceleration	Operation method	Operation function	Dwell time	Applied current	Progressive positioning
No. 1	5000	5000	1000	1000	INC	Connected	Unused	Unused	Unused
No. 2	20000	10000	Unused	Unused	INC	Independent	Unused	Unused	Unused

Operation image



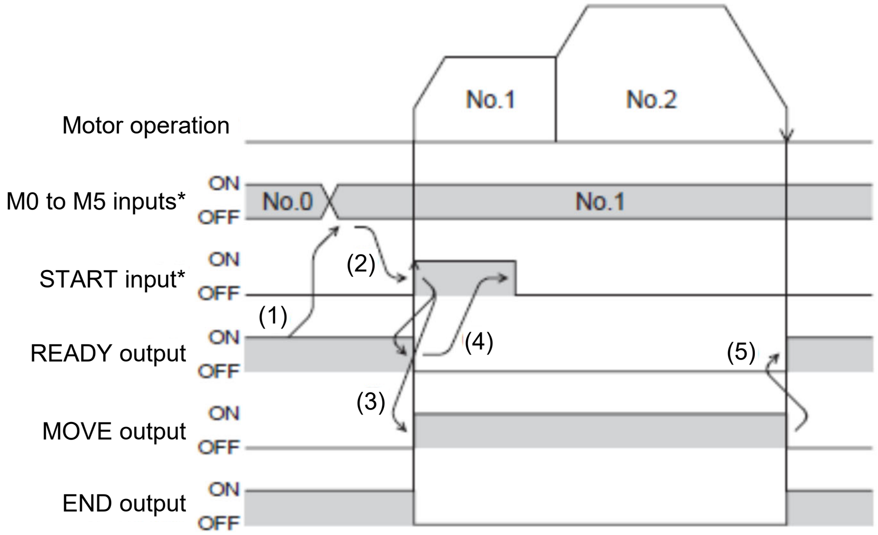
(1) Check that the READY output is ON.

(2) Turn ON the M0 input to select operation data No. 1 and turn ON the START input.

(3) The motor starts the positioning operation that connects operation data No. 1 and No. 2.

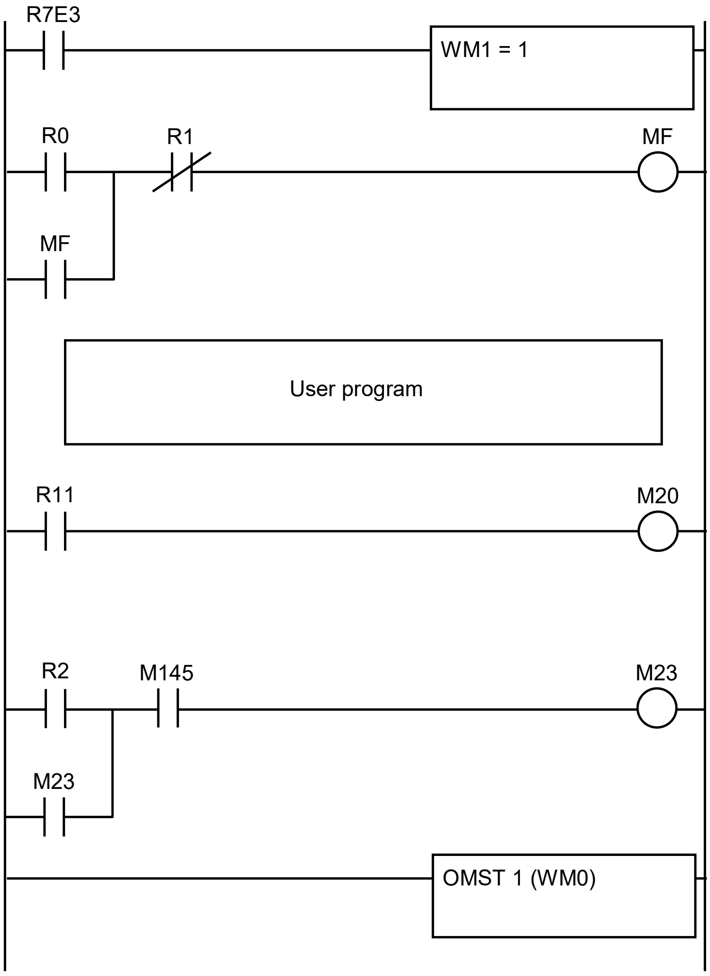
(4) Check the the READY output is turned OFF and turn OFF the START input.

(5) When the positioning operation is completed, the READY output is turned ON.



(Oriental Manual (HM-60223) P.3-13)

* The M0 to M5 inputs are used in the above example, but the M0 to M2 inputs are supported in this instruction.



Station number setting

R0: OMST1 instruction execution condition
R1: OMST1 instruction stop condition

R11: Operation No. 1 specification condition
M20: M0 input
M21: M1 input
M22: M2 input (M0, M1, M2) = (1, 0, 0)
→ Operation No. 1

R2: Positioning operation start condition
M23: START input
M145: READY output
When the condition is met, the START input is turned ON.
When the READY output is turned OFF, the START input is turned OFF.

Instruction execution part

[Program description]

If the operation function of the target operation data is set to "connected", the program runs the same as an independent operation.

- After RUN, the station number of the motor driver to be connected in the first scan is set, and when the execution condition is met (R0 is turned ON), the OMST1 instruction is executed to communicate with the motor driver.
- The instruction execution bit (MF) is set to remain ON until the instruction stops (R1 is turned ON) using self-latching.
- The operation No. is set to 1 (R11 is turned ON) by specifying the M0 to M2 inputs (M20 to M22) to the motor driver.
- While the READY output (M145) from the motor driver is ON, if the positioning operation start condition (R2) is met, the START input (M23) to the motor driver is output, and the stepping motor starts the positioning operation.

(2) Speed control operation example

A) Continuous operation (+ direction only)

No.	Ladder program processing	Description
1	Check that (M145) [READY] is ON	Check that [READY (operation ready)] is ON
2	Turn OFF all of (M20/M21/M22) [M0/M1/M2]	Select operation data No. 0
3	Turn ON (M2E) [FWD]	**+ direction continuous operation start** Turn ON [FWD (+ direction continuous operation start)]
4	Check that (M145) [READY] is OFF	Check that [READY (operation ready)] is OFF
5	-	Accelerate to the speed of operation data No. 0
6	The condition of transition from operation data No. 0 to No. 1 is met	
7	Turn ON (M20) [M0]	Accelerate to the speed of operation data No. 1
8	The condition of transition from operation data No. 1 to No. 0 is met	
9	Turn OFF (M20) [M0]	Decelerate to the speed of operation data No. 0
10	The stop condition is met	
11	Turn OFF (M2E) [FWD]	Turn OFF [FWD (+ direction continuous operation start)]
12	Check that (M145) [READY] is ON	**Positioning operation completion** Check that [READY (operation ready)] is ON

(1) Check that the READY output is ON.

(2) Select operation data No. with the M0 to M5 inputs and turn ON the FWD input.

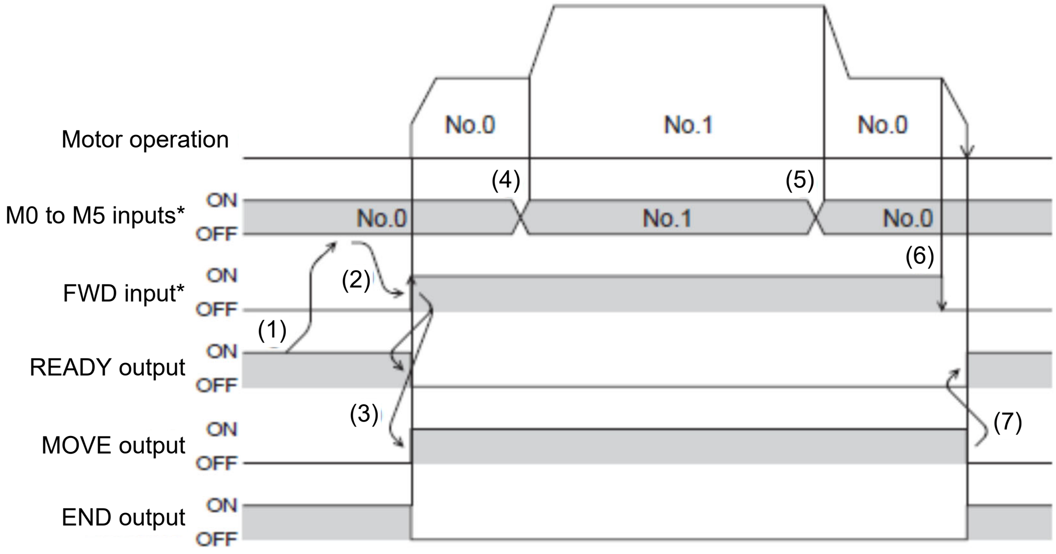
(3) The motor starts the continuous operation. The READY output is turned OFF.

(4) Turn ON the M0 input to select operation data No. 1. The motor accelerates to the operation speed of operation data No. 1.

(5) Turn OFF the M0 input to select operation data No. 0. The motor decelerates to the operation speed of operation data No. 0.

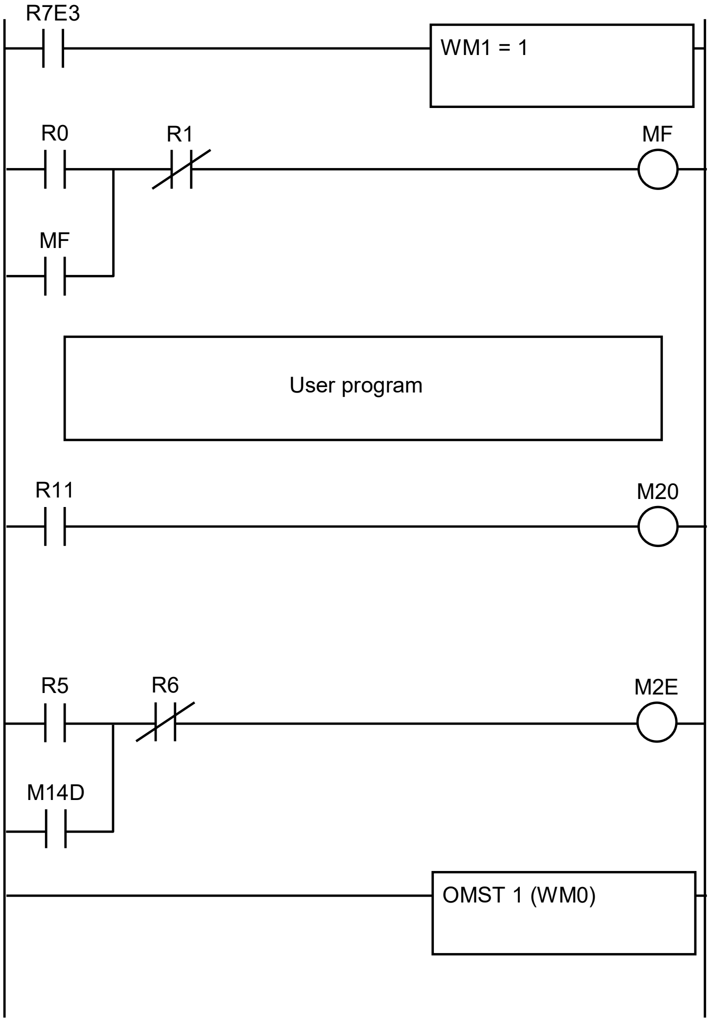
(6) Turn OFF the FWD input.

(7) The motor decelerates and stops, and the READY output is turned ON.



(Oriental Manual (HM-60223) P.3-25)

* The M0 to M5 inputs are used in the above example, but the M0 to M2 inputs are supported in this instruction.



Station number setting

R0: OMST1 instruction execution condition
R1: OMST1 instruction stop condition

R10: Operation No. 0 specification condition
R11: Operation No. 1 specification condition
M20: M0 input
M21: M1 input
M22: M2 input
(M0, M1, M2) = (0, 0, 0)
→ Operation No. 0
(M0, M1, M2) = (1, 0, 0)
→ Operation No. 1

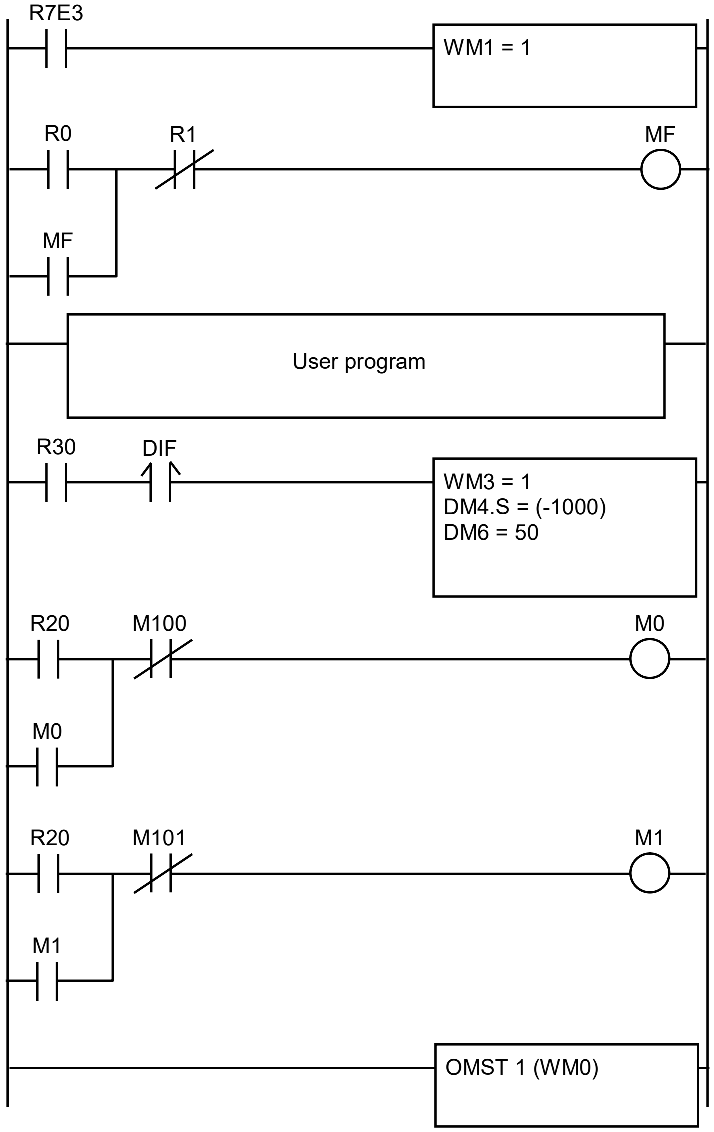
R5: Forward continuous operation start condition
R6: Operation stop condition
M2E: FWD input
M14D: MOVE output
When the condition is met, the FWD input is turned ON.
When the stop condition is met, the FWD input is turned OFF.

Instruction execution part

[Program description]

- After RUN, the station number of the motor driver to be connected in the first scan is set, and when the execution condition is met (R0 is turned ON), the OMST1 instruction is executed to communicate with the motor driver.
- The instruction execution bit (MF) is set to remain ON until the instruction stops (R1 is turned ON) using self-latching.
- When the forward continuous operation start condition (R5) is met, the FWD input (M2E) to the motor driver is output, and the stepping motor starts the forward continuous operation.
- When the condition of transition from operation data No. 0 to No. 1 is met (R11 is turned ON), the operation No. is set to 1 by specifying the M0 to M2 inputs (M20 to M22) to the motor driver.
- When the condition of transition from operation data No. 1 to No. 0 is met (R11 is turned OFF), the operation No. is set to 0 by specifying the M0 to M2 inputs (M20 to M22) to the motor driver.
- When the stop condition (R6) is met, the FWD input (M2E) to the motor driver stops being output.

(3) Operation data settings



Station number setting

R0: OMST1 instruction execution condition
R1: OMST1 instruction stop condition

R30: Operation No. 1 ready condition
WM3: Setting operation data No. specification condition
DM4: Position data
DM6: Operation speed
When the condition is set, the operation data No. is set to 1.

R20: Operation data No. 1 setting condition
M0: Position data setting request (PD)
M100: Position data setting response (RPD)

R20: Operation data No. 1 setting condition
M1: Speed data setting request (PV)
M101: Speed data setting response (RPV)

Instruction execution part

[Program description]

- After RUN, the station number of the motor driver to be connected in the first scan is set, and when the execution condition is met (R0 is turned ON), the OMST1 instruction is executed to communicate with the motor driver.
- The instruction execution bit (MF) is set to remain ON until the instruction stops (R1 is turned ON) using self-latching.
- When the condition (R30) is met, the position data is set to -1000, and the operation speed is set to 50 Hz in operation data No. 1.
- When the operation data is set, the position data setting request bit (M0) and the speed data setting request bit (M1) are turned ON.
- The position data setting request bit (M0) and the speed data setting request bit (M1) are turned OFF when the corresponding response bits (M100, M101) are found to be turned ON.

Return Code

The following lists the return codes to be stored into the [s+17] area after OMST1 execution.

(1) When the PLC is in a normal state or has a communication or instruction error

Code	Name	Description	Solution
H0000[*1]	Normal termination	Transmission is in a normal state.	-
H0021[*2]	Range check error	The end of parameter s is outside the I/O range.	Set the area of parameter s within the valid range.
H0030	Timeout	Transmission did not end within the specified time.	Set a larger value or change the processing.
H0040	Data capacity exceeded	• The received data has exceeded 1,028 bytes. • There is no available space for received data.	Check the number of coils/registers of received data.
H0041	Parity error Framing error Overrun error	A parity error, framing error, or overrun error occurred in communication processing.	Check the transmission line of the optional port and the data format.
H0044	Simultaneous start limit exceeded	More than the maximum number of INVn, OMSTn, and OCTPn instructions was started (EXE bit was turned ON).	The maximum number of INVn/OMSTn/OCTPn instructions that can be started simultaneously is 32.
H0045	Parameter error	• The station number setting is outside the range or duplicated. • The destination station number was changed during communication. • The INVn or OCTPn instruction and parameter s are duplicated.	• The station number must be set between 0 and 247. • Reset the station number to the one used when the communication started. • Change the area of parameter s.
H0046	Port specification error	• The OMST1 instruction was started when the optional port was not specified as the Modbus master. • No optional board is installed.	Check the port specification.
H0071	Received data error	• The received data is invalid. • The number of received bytes, except for the size of header, check code, and trailer, has exceeded 509.	Check if the slave device is compatible with Modbus.
H0072	CRC error	A CRC check error occurred.	Check the slave Modbus mode.

*1 To always continue communication during operation, this instruction does not update the error code when the communication is successfully completed. If a communication error occurs, clear the error after removing the cause.

*2 If the return code area is outside the I/O range, the return code may not be displayed.

(2) When the PLC receives an exceptional response

The exception code from the slave device is stored into the low byte, and the write request ID is stored into the high byte.

Code	Name	Description	Solution
Hxx01*	Illegal function	The slave device does not support the requested function.	Check if the slave device is compatible with this instruction.
Hxx02*	Illegal data access	The specified data address does not exist in the slave device.	Check if the slave device is compatible with this instruction.
Hxx03*	Illegal data	The specified data is invalid.	Check if the slave device is compatible with this instruction.
Hxx04*	Slave error	An error occurred on the slave device, preventing query processing from running.	For details, see the motor unit manual.

* The ID shown in the table below corresponding to the write request that resulted in the error is set in the high byte xx.

ID	Transmission request
H80	Driver input instruction write
H81	Operation No. n position data setting
H82	Operation No. n speed data setting
H83	Alarm reset

Name	Omron temperature controller control instruction 1

Ladder format	Number of steps		Condition code				
	Condition	Step	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C
OCTP 1 (s)	—	3	↕	●	●	●	●

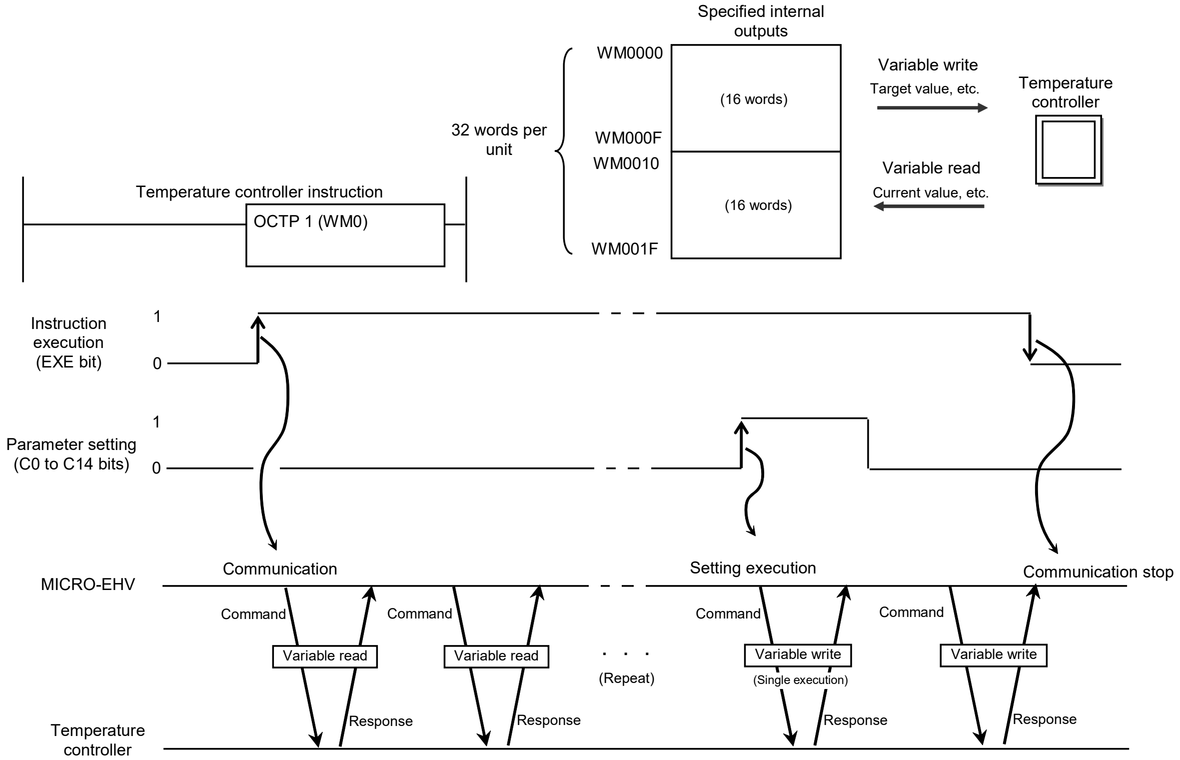
Instruction processing time (μs)					
Average			Maximum		
Condition	Time		Condition	Time	
	MVH (High function)	MVL (Standard)		MVH (High function)	MVL (Standard)
—	24.6	20.16	—	—	—

Usable I/O		Bit					Word				Double word			Constant
		X	Y	R,M	TD, SS, MS, CU, CT TDN, WDT, TMR, RCU,	WR, (.m)	WX	WY	WR, WM	TC	DX	DY	DR,DM	
s	Data table first I/O								✓					

Note
s parameters are occupied up to s+1B (28 words).

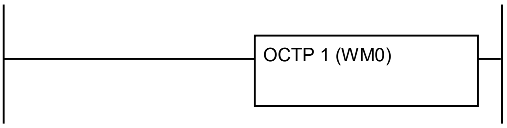
Function

- This instruction performs Modbus communication with OMRON temperature controller (digital controller model: E5CC/E5EC) using the communication port 2 of the optional board.
- The topology is 1:1 or 1:N. The maximum number of connected temperature controllers is 31. One instruction of this type is required to commutate with each temperature controller.
- While the instruction execution bit is ON, Modbus communication with temperature controllers (digital controllers) is performed.
- Temperature controllers can be controlled by turning ON or OFF bits on the memory map image. The internal output area, which communicates with temperature controllers, can be specified in the argument I/O of the instruction.



Program example

OCTP1 (s)



s: Specify the first I/O of the function data table.

		Data	Description
s+0	W	Control bit area	This is the control bit area of this instruction. For details, see (1) [s+0] Control bit area.
s+1	W	Station number (0 to 247)	Set the station number (Modbus slave address) of the temperature controller you want to communicate with. (0 to 247)
s+2	W	Target value [2103]	These parameters are used to write into the temperature controller. For details, see the temperature controller manual.
s+3	W	Proportional band [2701]	
s+4	W	Integral time [2702]	
s+5	W	Derivative time [2703]	
s+6	W	Warning value 1 [2104]	
s+7	W	Warning upper limit 1 [2105]	
s+8	W	Warning lower limit 1 [2106]	
s+9	W	Warning value 2 [2107]	
s+10	W	Warning upper limit 2 [2108]	
s+11	W	Warning lower limit 2 [2109]	
s+12	W	Heater burnout detection 1 [271B]	
s+13	W	PV input correction value [2723]	
s+14	W	SP lamp setting value [270D]	
s+15		-	(Undefined)
s+16	R	Return code	Sets the instruction execution result. See the list of return codes.
s+17	R	Communication status	This is the bit area for communication status check. For details, see (2) [s+17] Communication status.
s+18	R	Communication cycle	Stores the cycle (x 1 msec) at which the MICRO-EHV communicates with the temperature controller. Use it as a guide to check the communication interval of with target temperature controller. (Default: 0)
s+19	R	Status (low)* [2407]	Stores the parameters read from the temperature controller. These data are always read by turning ON the EXE bit of [s+0] Control bit area. For details on each parameter, see the temperature controller manual.
s+20	R	Status (high)* [2406]	
s+21	R	Status 2 (high)* [2409]	
s+22	R	Decimal point monitor [2410]	
s+23	R	Current value [2000]	
s+24	R	Internal target value [2002]	
s+25	R	Heater current value 1 monitor [2003]	
s+26	R	Manipulated variable monitor (heating) [2004]	
s+27		-	(Undefined)

W : Write parameter, R: Read parameter
[ ] : Temperature controller variable area address (hexadecimal, 2-byte mode)
* : Use double word access.

(1) [s+0] Control bit area

This is the area for write to the temperature controller. This area consists of 16 bits, and the function of each bit is as shown in the table below.

	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
[s+0]	EXE	ECR	—	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0

To set each parameter, turn ON the corresponding bit. When a write is completed, the system turns it OFF. If it does not successfully terminate due to a communication error, the error code is stored into the return code area with the corresponding bit remaining ON, and the ERR bit of the communication status area is turned ON.

Bit	Definition	Name	Setting data
0	C0	Target temperature setting	[s+2] Target value
1	C1	Proportional band setting	[s+3] Proportional band
2	C2	Integral time setting	[s+4] Integral time
3	C3	Derivative time setting	[s+5] Derivative time
4	C4	Warning value 1 setting	[s+6] Warning value 1
5	C5	Warning upper limit 1 setting	[s+7] Warning upper limit 1
6	C6	Warning lower limit 1 setting	[s+8] Warning lower limit 1
7	C7	Warning value 2 setting	[s+9] Warning value 2
8	C8	Warning upper limit 2 setting	[s+10] Warning upper limit 2
9	C9	Warning lower limit 2 setting	[s+11] Warning lower limit 2
A	C10	Heater burnout detection 1	[s+12] Heater burnout detection 1
B	C11	PV input correction value	[s+13] PV input correction value
C	C12	SP lamp setting value	[s+14] SP lamp setting value
D	-	-	(Undefined)
E	ECR	Communication error clear	Sets the ERR bit of [s+17] to 1 when a communication error is detected. When this bit is set to 1, the ERR bit of [s+17] and the error code stored in [s+16] are cleared to 0. Set it back to 0 after making sure that the error has been cleared.
F	EXE	Instruction execution	Set 1 to start communication with the temperature controller. The PLC communicates with the station number [s+1] used when this bit is set to 1. While this bit is ON, the PLC is always communicating with the temperature controller. Set it back to 0 to stop communication with the temperature controller.

(2) [s+17] Communication status

This is the area for write to the temperature controller. This area consists of 16 bits, and the function of each bit is as shown in the table below.

	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
[s+17]	REX	ERR	—	—	—	—	—	—	—	—	—	—	—	—	—	STS

Bit	Definition	Name	Setting data
0	STS	Status	Inverts the lowest bit between 0 and 1 every time communication with the temperature control is completed. Use it as a reference to check the communication status.
1 to D	-	-	(Undefined)
E	ERR	Communication error	Turns ON when a communication error is detected. If the ECR bit of [s+0] is found to be turned ON, this bit is turned OFF.
F	REX	Instruction execution	Turns ON while the EXE bit of [s+0] is ON. If the EXE bit is found to be turned OFF, this bit is also turned OFF.

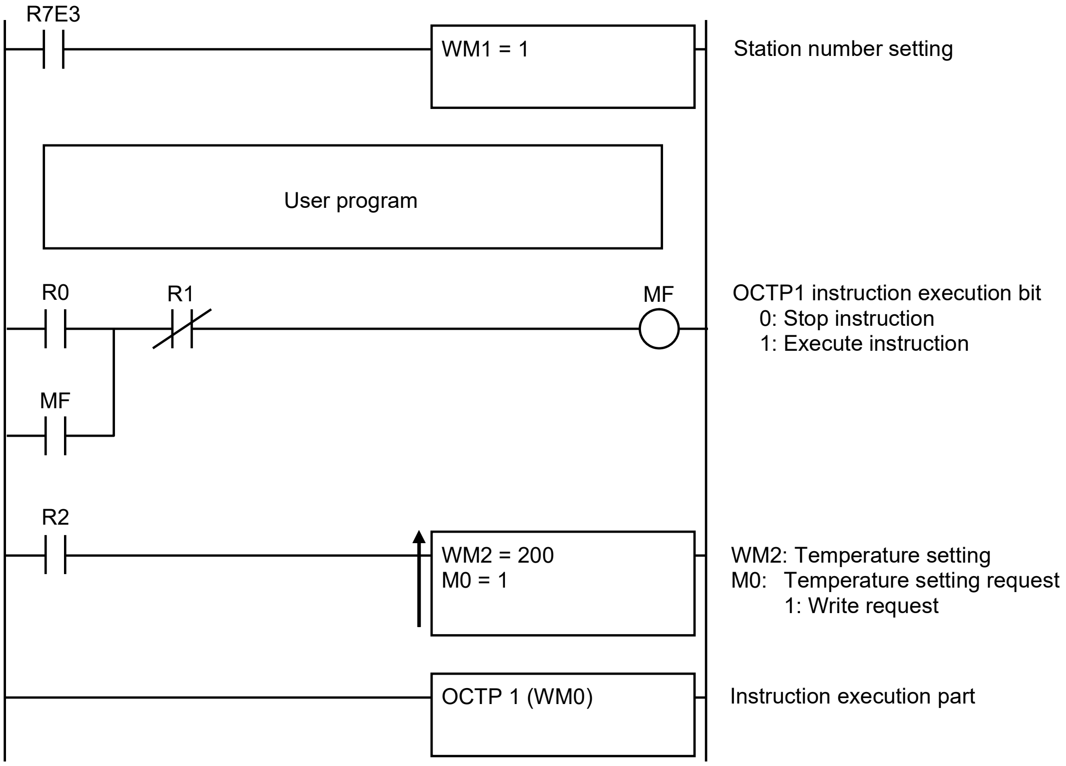
Cautionary notes

- When station number 0 is specified, data in the s+16 to s+31 areas do not change because monitor communication is not performed.
- To use this instruction, set or change the following temperature controller parameters according to the system configuration.

Item	Description
Control method	ON/OFF control heating system
Adjusting sensibility	1.0 ℃
Warning setting	None
Advanced function setting level protect	Reset Modbus slave address
Communication write setting	Allowed
Communication protocol	Modbus
Transmission wait time	20 ms
Transmission speed	38,400 bps
Transmission format	8-bit even-parity 1 stop bit

(For details on the setting information, see the temperature controller manual.)

Program example

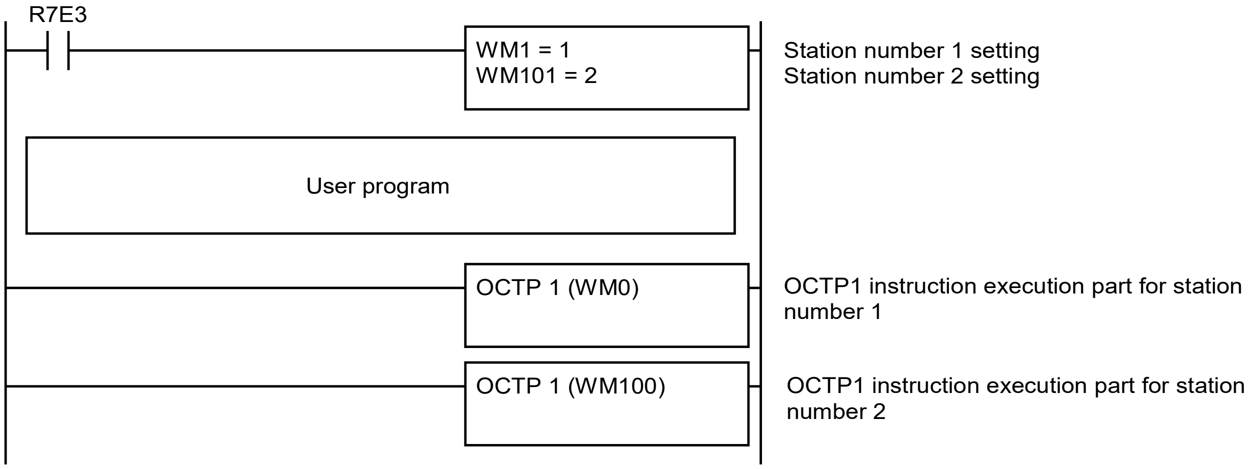


[Program description]

- The destination station number is set to 1 in the first scan.
- The instruction execution bit (MF in this example) is set to remain ON until the instruction stops (R1 is turned ON) using self-latching.
- When the temperature setting condition is met (R2 is turned ON), the setting temperature is set in the frequency setting area (WM2 in this example), and the temperature change request bit (M3 in this example) is turned ON.

<Multiple-unit connection>

To connect multiple temperature controllers, write as many instructions as the number of connected controllers and specify the destination of the target OCTP1 instruction in the s+1 station number setting. The maximum number of INVn / OMSTn / OCTPn instructions that can be executed simultaneously is 32.



Return Code

The following lists the return codes to be stored into the [s+16] area after OCTP1 execution.

(1) When the PLC is in a normal state or has a communication or instruction error

Code	Name	Description	Solution
H0000[*1]	Normal termination	Transmission is in a normal state.	-
H0021[*2]	Range check error	The end of parameter s is outside the I/O range.	Set the area of parameter s within the valid range.
H0030	Timeout	Transmission did not end within the specified time.	Set a larger value or change the processing.
H0040	Data capacity exceeded	• The received data has exceeded 1,028 bytes. • There is no available space for received data.	Check the number of coils/registers of received data.
H0041	Parity error Framing error Overrun error	A parity error, framing error, or overrun error occurred in communication processing.	Check the transmission line of the optional port and the data format.
H0044	Simultaneous start limit exceeded	More than the maximum number of INVn, OMSTn, and OCTPn instructions was started (EXE bit was turned ON).	The maximum number of INVn/OMSTn/OCTPn instructions that can be started simultaneously is 32.
H0045	Parameter error	• The station number setting is outside the range or duplicated. • The destination station number was changed during communication. • The INVn or OMSTn instruction and parameter s are duplicated.	• The station number must be set between 0 and 247. • Reset the station number to the one used when the communication started. • Change the area of parameter s.
H0046	Port specification error	• The OCTP1 instruction was started when the optional port was not specified as the Modbus master. • No optional board is installed.	Check the port specification.
H0071	Received data error	• The received data is invalid. • The number of received bytes, except for the size of header, check code, and trailer, has exceeded 509.	Check if the slave device is compatible with Modbus.
H0072	CRC error	A CRC check error occurred.	Check the slave Modbus mode.

*1: To always continue communication during operation, this instruction does not update the error code when the communication is successfully completed. If a communication error occurs, clear the error after removing the cause.

*2: If the return code area is outside the I/O range, the return code may not be displayed.

(2) When the PLC receives an exceptional response

The exception code from the slave device is stored into the low byte, and the write request ID is stored into the high byte.

Code	Name	Description	Solution
Hxx01*	Function code error	The slave device does not support the requested function.	Check if the slave device is compatible with this instruction.
Hxx02*	Variable address error	The specified data address does not exist in the slave device.	Check if the slave device is compatible with this instruction.
Hxx03*	Variable data error	The specified data is outside the setting range.	Check the write data.
Hxx04*	Operation error	An error occurred on the slave device, preventing query processing from running.	For details, see the temperature controller manual.

* The ID shown in the table below corresponding to the write request that resulted in the error is set in the high byte xx.

ID	Transmission request
H80	Target value
H81	Proportional band
H82	Integral time
H83	Derivative time
H84	Warning value 1

ID	Transmission request
H85	Warning upper limit 1
H86	Warning lower limit 1
H87	Warning value 2
H88	Warning upper limit 2

ID	Transmission request
H89	Warning lower limit 2
H8A	Heater burnout detection 1
H8B	PV input correction value
H8C	SP lamp setting value

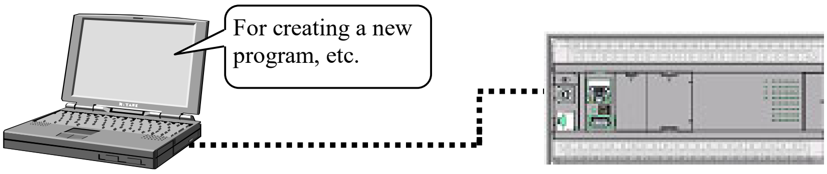
Chapter 6 Online Connection

6.1 User program up/downloading

Use the programming tool to write and read the user program. The programming tool has the following three connection modes.

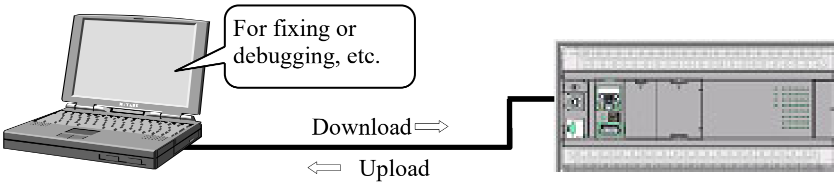
■ Offline mode

This mode does not connect the tool to the MICRO-EHV. With this mode, you cannot write and read the program to/from the MICRO-EHV. This is used to create a new program or modify a program without connecting to the MICRO-EHV.



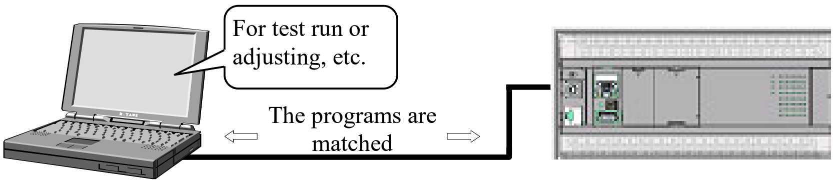
■ Online mode

This mode connects the tool to the MICRO-EHV. You can monitor the states of circuits and I/Os and transfer a program using Download / Upload functions. The program can be read but not be changed during operation. This is mostly used to debug or perform a test run while connecting the actual PLC to the MICRO-EHV.



■ On-direct mode

This mode connects the tool to the MICRO-EHV. This mode is available only when the program on the PC matches the program on the MICRO-EHV. Unlike the online mode, the on-direct mode allows you to change the program during operation (online change in RUN). After the program is edited, the modification is transferred to the MICRO-EHV by build operation. This is mostly used to fix the program during a test run or system operation.



(1) User program write (Download)

This is the procedure for writing a user program created in the programming tool to the MICRO-EHV.

The steps of turning ON the PLC and connecting the programming tool to the MICRO-EHV with the communication cable are omitted.

i) Select the [Online] connection in the programming tool.
   Click the online icon at the top of the editor or select [Online] - [Mode] - [Online] on the menu.

ii) Select [Download].
   Click Download to PLC icon at the top of the editor or select [Online] - [PLC Transfer] - [Download] from the menu.

(2) User program read (Upload)

This is the procedure for reading the user program stored in the MICRO-EHV out to the programming tool.

i) Select the [Online] connection in the programming tool.

ii) Select [Upload].
   Click Upload from PLC icon at the top of the editor or select [Online] - [PLC Transfer] - [Upload] from the menu.

Caution

Please note that if you read the program, any unsaved project being edited will be lost.

(3) User program verification (PLC verify)

This is the procedure for checking the user program stored in the MICRO-EHV against the user program opened in the programming tool.

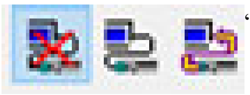
i) Select the [Online] connection in the programming tool.

ii) Select [Verify With PLC].
   Click PLC Verify icon at the top of the editor or select [Online] - [PLC Transfer] - [Verify With PLC] from the menu.

Reference

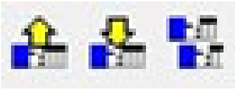
The following shows the icons to perform the above operations in the Control Editor:

Connection mode switch icon



‘Offline”, “Online”, and “On-direct” in order from the left

PLC transfer icon



“Upload”, “Download”, and “PLC verify” in order from the left

6.2 Online change in RUN

Modifying a part of the user program during operation (RUN) is called online change in RUN.

Online change in RUN retains the states of outputs and data memory, allowing you to change the user program without affecting the network.

Normally online change in RUN is executed immediately after END instruction. If [Enable Manual Online Change] is enabled, this timing can be determined manually with the programming tool. This method allows you to change the program while avoiding the time when the system should never stop.

(1) Procedure for online change in RUN

To perform online change in RUN, the user program in the programming tool must match the user program stored in the MICRO-EHV. If they are unmatched, read the program from the MICRO-EHV with the programming tool or open the same program (project file) as the one stored in the MICRO-EHV.

The flow of online change in RUN is as follows:

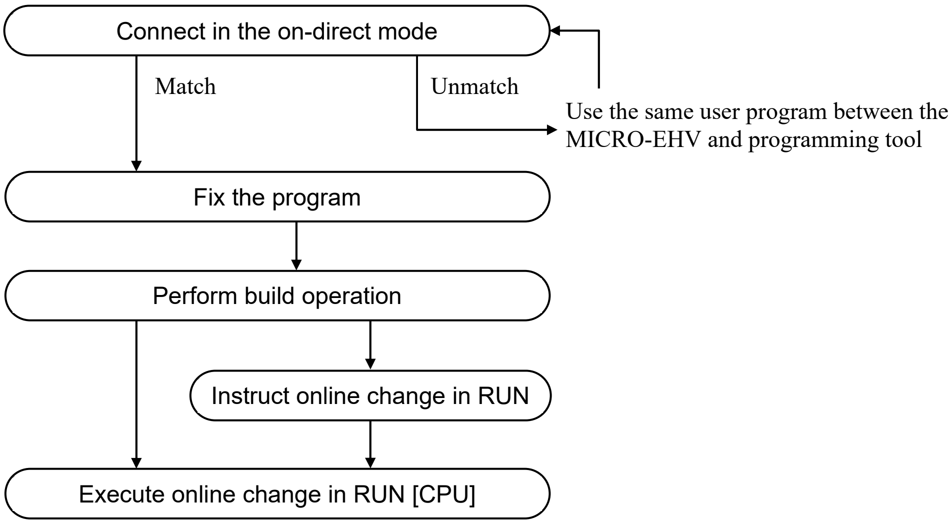
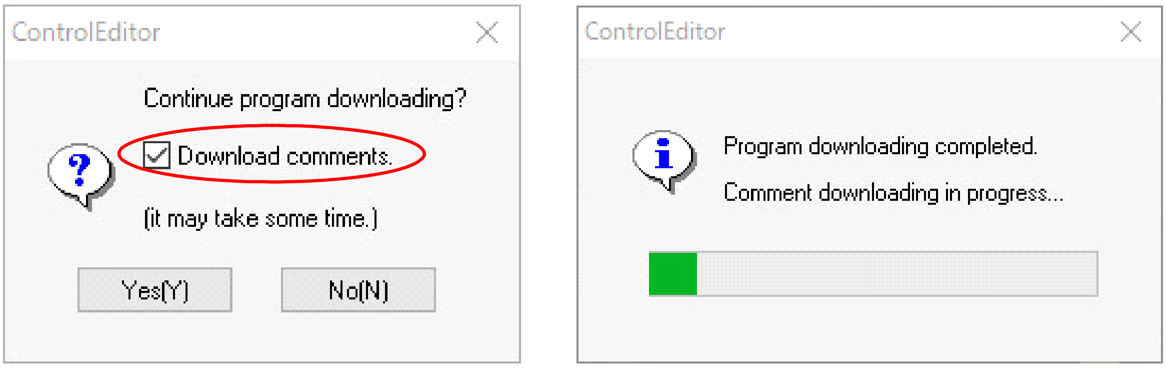


Figure 6.1 Flow of online change in RUN

During on-direct connection, even if the MICRO-EHV is in the STOP state, the circuits written by build operation are reflected on the user program in the MICRO-EHV.

(2) Comment transfer in RUN

The comment transfer in RUN function transfers all comment data at once to the CPU after program change.



ControlEditor

Continue program downloading?

☑ Download comments.

(it may take some time.)

ControlEditor

Program downloading completed.

Comment downloading in progress...

Figure 6.2 Online change in RUN message and comment transfer window

To transfer comments as well after program change, select the check box on the confirmation message on the left in Figure 6.2. During comment transfer, the transfer screen on the right in Figure 6.2 appears. Since the program change is already completed during the comment transfer, the CPU continues the operation.

All of I/O comments, box comments, and circuit comments are transferred at once. You can leave the check box unselected normally and select it only when transferring comments at the last program change. However, there is no function available to transfer only comments. If you forget to select the check box at the last program change or need to transfer only comments without changing the program, double-click any program symbol and click the [OK] button without changing anything. When a part of the program is put in the edit mode, perform online change in RUN.

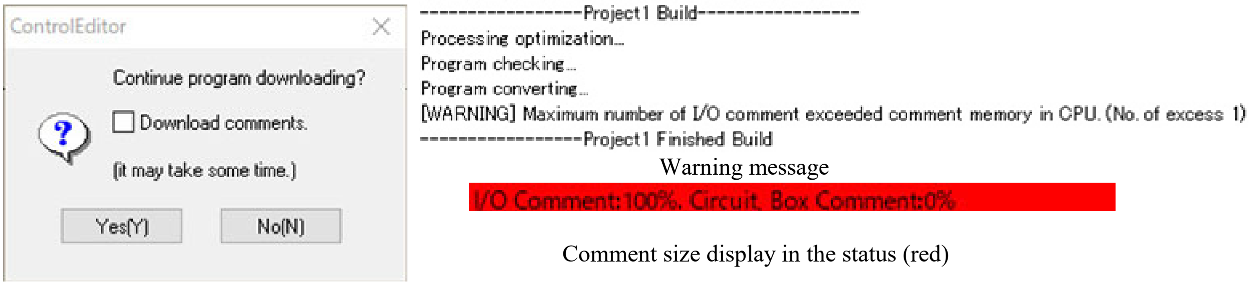
Caution

The Control Editor checks the programs before the PLC enters the on-direct mode or monitor mode, and if the programs are matched between the CPU and Control Editor, the PLC enters the corresponding mode, but comment match check is not performed. For this reason, the PLC can enter the on-direct mode even if comments are unmatched. So, please pay extra attention when reading the program and comments simultaneously from the CPU after online change in RUN. Meanwhile, the PLC verify function checks comments as well. If you are not sure whether the comments on the CPU side are the latest, unselect the comment check box before performing upload so that only the program is read.

(3) Comment size exceeded during online change in RUN

If the comment size is exceeded during online change in RUN in the on-direct mode, the behavior varies depending on whether the [Download comments] check box is selected or not.

i) When the [Download comments] check box is not selected (when not transferring comments)



Continue program downloading?

☐ Download comments.

(It may take some time.)

```
-----------------Project1 Build-----------------
Processing optimization…
Program checking…
Program converting…
[WARNING] Maximum number of I/O comment exceeded comment memory in CPU. (No. of excess 1)
-----------------Project1 Finished Build
```

Warning message

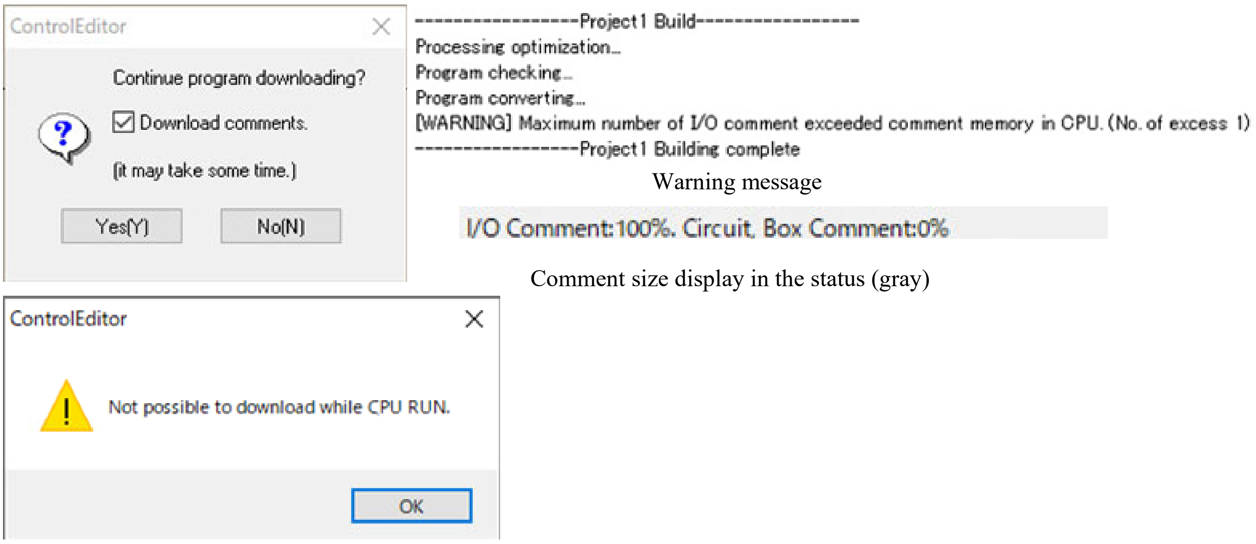
I/O Comment:100%. Circuit. Box Comment:0%

Comment size display in the status (red)

Figure 6.3 Online change in RUN when comment size is exceeded (1)

In this case, even if the comment size exceeds 100 %, the comments are not transferred to the CPU, so online change in RUN is possible although the warning shown in the figure appears.

ii) When the [Download comments] check box is selected (when transferring comments)



Continue program downloading?

☑ Download comments.

(It may take some time.)

```
-----------------Project1 Build-----------------
Processing optimization…
Program checking…
Program converting…
[WARNING] Maximum number of I/O comment exceeded comment memory in CPU. (No. of excess 1)
-----------------Project1 Building complete
```

Warning message

I/O Comment:100%. Circuit. Box Comment:0%

Comment size display in the status (gray)

Not possible to download while CPU RUN.

Figure 6.4 Online change in RUN when comment size is exceeded (2)

Since the comment size is exceeded, the comments are not transferred, and neither is the program. Therefore, online change in RUN operation is canceled. Please note that the comment size display area at the lower part of the window remains gray.

(4) Unacceptable conditions for online change in RUN

You cannot perform online change in RUN under the following conditions. Change the conditions before the operation.

Table 6.1 Unacceptable conditions for online change in RUN

No	Online change in RUN unacceptable condition	Specific situation	Solution
1	During READ occupation	Another programming tool is connected.	Take the programming tool offline.
2		A PC or touch panel is connected for monitoring.	Take the PC or touch panel offline.*
3	The END instruction cannot be executed.	The executed program loops infinitely.	Fix the program so that it does not loop infinitely.
4	The password is set.	The executed program is password locked.	Contact the system administrator to unlock the password before retry.
5	The user is logged in with an ID that has no authority to change the program.	The security function prevents the program from being changed.	Log in again with an ID that has authority to change the program.

* In the Hi-Protocol communication, which is the protocol used for the MICRO-EHV, there are commands that need to be exclusively locked or not even for the same request. When monitoring or setting/resetting using a command that requires no exclusive lock, you do not need to take it offline.

(5) HALT time

When you perform online change in RUN, the scan is stopped only for a while during the change. This scan stop time is called "HALT time".

The HALT time can be calculated from the following formula, except for online change in RUN including the conditions shown below:

High Function model: HALT time [ms] = Program size [k steps] × 2.25 + 1.2 [ms]

Standard model: HALT time [ms] = Program size [k steps] × 2.30 + 1.2 [ms]

- Number of used timers

  The number of timers used in the entire program × 0.25 µs is added.

- Edge instruction deletion / change

  If a circuit to be changed in RUN contains edge instructions, 100 µs is added regardless of the number of edge instructions.

  Edge instruction: DIF, DFN, edge coil, and edge processing box

(6) Online change in RUN for control commands

The MICRO-EHV allows online change in RUN for control commands. In the programming tool, however, you cannot build into a program that ends up in a syntax error. (Online change in RUN is not possible.)

Table 6.2 Reasons for control instruction syntax errors

Instruction	Syntax error reason
END	There are two END instructions. The END instruction contains the start condition.
CEND (s)	CEND (s) follows the END instruction.
JMP n / CJMP n	There is no corresponding LBL n. The program is trying to jump to another program area.
LBL	LBL is duplicated.
FOR n (s) / NEXT	FOR is duplicated. NEXT is not defined. FOR nesting overflow. FOR and NEXT are not specified in the same area.
CAL / SB n / RTS	SB is not defined. Nesting over. SB is duplicated. RTS is not defined. SB and RTS are not specified in the same area. RTS contains the start condition.
INT (s) / RTI	INT is not defined. The same cycle is duplicated. RTI is not defined. INT and RTI are not specified in the same area. RTI contains the start condition.
XINT (s) / XRTI	XINT is not defined. The same cycle is duplicated. XRTI is not defined. XINT and XRTI are not specified in the same area. XRTI contains the start condition.
CINTP (s) / CRTIP	CINTP is not defined. The same cycle is duplicated. CRTIP is not defined. CINTP and CRTIP are not specified in the same area. CRTIP contains the start condition.
CINTN (s) / CRTIN	CINTN is not defined. The same cycle is duplicated. CRTIN is not defined. CINTN and CRTIN are not specified in the same area. CRTIN contains the start condition.

(7) Online change in RUN including cyclic scan

When you perform online change in RUN including a cyclic scan, if online change in RUN timing overlaps a cyclic scan timing, the cyclic scan is skipped once. (This occurs only when the timings overlap, so it does not mean that online change in RUN always skips a cyclic scan once.)

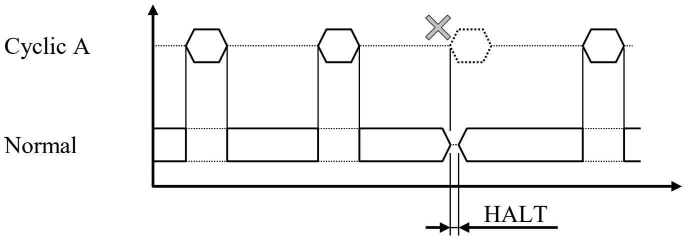


Figure 6.5 Online change in RUN for program including cyclic scan

Reference

The MICRO-EHV allows you to change a cyclic scan cycle and add or delete a cyclic scan using online change in RUN. When multiple cyclic scans exist in the program, if you change one or more cyclic scan cycle time, the cycle time of other cyclic scan may be changed once at online change timing because all the cyclic scans are restarted.

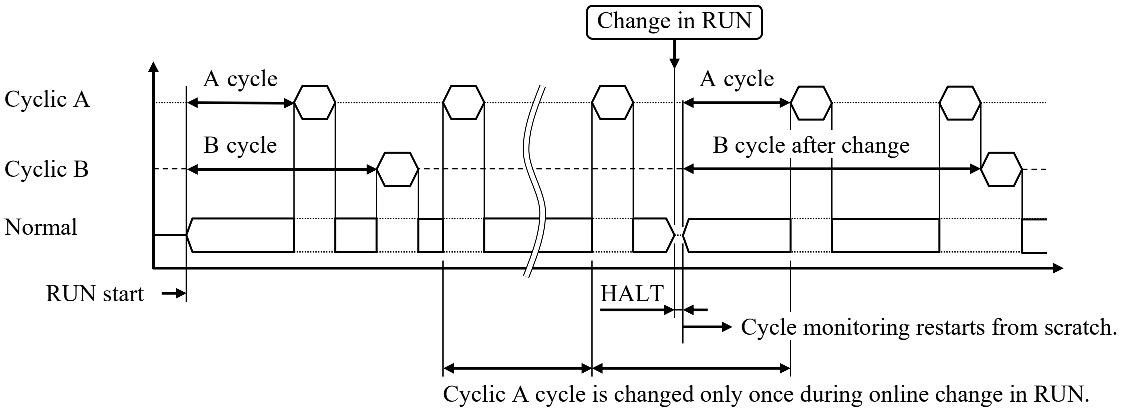


Figure 6.6 Adding cyclic scan in online change in RUN

6.3 Forced output

Menu: [Online] - [Force Set]

The forced output function outputs only a specified output. Even if multiple outputs are ON, any other output than the specified one is turned OFF, which helps you check the wiring of the output.

Reference

There is a similar function called I/O "set/reset". The wiring can be checked using the set/reset function, but if you check the output one by one, you need to turn each output OFF after the check. Forced output is useful because any other output than the specified one is automatically turned OFF.

6.4 Monitor

Menu: [Online] - [Circuit Monitor]

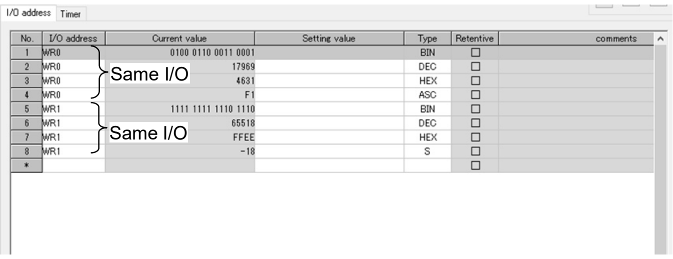
You can monitor the states and values of I/Os in the programming tool.

■ I/O monitor

The I/O monitor window allows you to check the states and values of specified I/Os.

You can also check whether the bit I/O state is ON (1) or OFF (0). For word data or double word data, the data type can be specified for each I/O, so you can monitor data of the type specified in the program.

(Even when you are monitoring the same I/O, if you change the display type, it is represented by another value.)



No.	I/O address	Current value	Setting value	Type	Retentive	comments
1	WR0	0100 0110 0011 0001		BIN	☐	
2	WR0	17969		DEC	☐	
3	WR0	4631		HEX	☐	
4	WR0	F1		ASC	☐	
5	WR1	1111 1111 1110 1110		BIN	☐	
6	WR1	65518		DEC	☐	
7	WR1	FFEE		HEX	☐	
8	WR1	-18		S	☐	
*					☐	

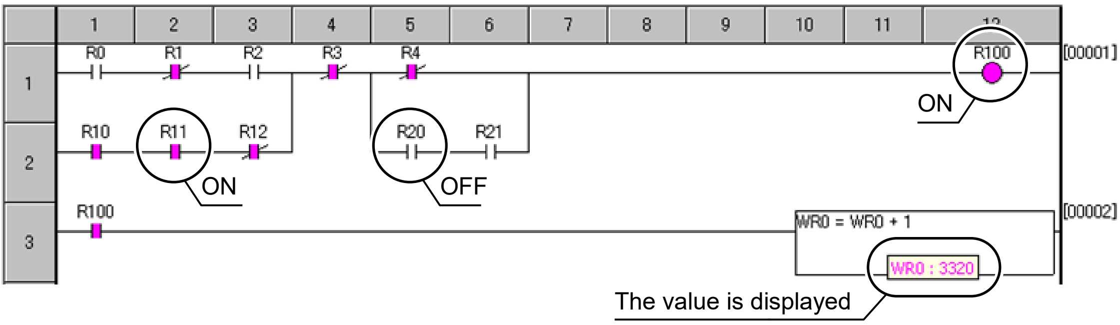
Figure 6.7 Differences in monitor values between data types

■ Circuit monitor

Menu: [Online] - [Circuit Monitor]

This window allows you to monitor the user program status.

A contact changes its color in the center when turned ON and remains white when turned OFF. A coil is filled when turned ON and remains white when turned OFF. When you place the mouse cursor over a processing box, the I/O value in the box is displayed.



The value is displayed

Figure 6.8 Circuit monitor

6.5 Prohibiting input and output refresh

Menu: [Online] - [Operate CPU] - [Input refresh disabled]

Menu: [Online] - [Operate CPU] - [Output refresh disabled]

You can temporarily prohibit input refresh and output refresh.

Since actual external inputs are not imported into the PLC while the input refresh is prohibited, you can turn ON external input data with the set/reset function using the touch panel or Control Editor regardless of external input signals (X, WX) are ON or OFF. This function is useful when external inputs are not wired or during debugging.

When you turn ON external input data with the set/reset function using the touch panel or Control Editor, the input LED on the unit does not light up. When actual external inputs are turned ON, the input LED on the unit lights up.

While the output refresh is prohibited, external output data (Y, WY) are not reflected on the actual external output signals. This is not only for ladder program instruction outputs but also for the set/reset function using the touch panel or Control Editor. Use this function when you do not want any actual external device to run during debugging.

The output LED on the unit does not light up.

This setting also applies to pulse/PWM outputs.

Caution

The OK LED on the PLC flashes while input/output refresh is prohibited.

Chapter 7 Troubleshooting

7.1 Error code and countermeasure procedures

(1) Error indication

The MICRO-EHV basic unit displays an error by OK LED.

And an error code is set in the special internal output (such as WRF000).

The smaller the error code number, the more serious the error is.

When two or more errors occur, the smaller number is set. For example, if “71” (battery error) and “31” (user memory error) occur simultaneously, “31” is set. If the levels are same, the cause code which occurs later than others will be displayed.

The clearing of the error special internal output is performed by setting the special internal output R7EC to 1. R7EC can be set to 1 either by connecting the programming device or by incorporating a subprogram to set R7EC using external input. (Turn R7EC on always after checking the error factor when turning it on by the program. However, if R7EC is turned on by the program in which a watchdog error occurs, the system may clear the error factor and run again after the system detects the watchdog error.)

* Error codes are set in a hexadecimal number. Verify the error code by setting the monitor to hexadecimal display.

(2) OK LED

MICRO-EHV detects an error that is serious than the middle failure, the OK LED turns off.

When detected error is not serious but error which should be recognized by customer, the OK LED blinking.

(3) Setting of error indication level

In MICRO-EHV, error indication level can be set by customer. If error display level is set, when error which is low level than setting level occurs, error code is not stored in the special internal output (excepts bit special internal output) and OK LED does not indicate the error information. (All error information is stored to error history regardless of error indication level.)

If the error display level is changed from high to low, OK LED may not turn on depending on the error factor because the error information is reflected after changing error indication level.

Reference

Error level which can set Display / Non-display

Level	Details	Display	Remarks
—	No setting	Displays all error codes	Factory Setting
1	Error codes of warning level (communication error) are not displayed.	Error code 6x or higher are not displayed.	
2	Error codes of warning level (high-function module error) are not displayed.	Error code 5x or higher are not displayed.	
3	Error codes of slight failure level are not displayed.	Error code 4x or higher are not displayed.	

The battery error (H71), Retentive data area is undefined status (H76), and the backup memory error (H77) are set independent of the error indication level. Even if the slight failure level is set to the non-display on the error indication level, the battery error is displayed if it is set to the display.

(4) Operation / Error history

MICRO-EHV can memorize the history of errors that occurred in MICRO-EHV and the operations performed by the user. The time information is added to the history, and it is possible to memorize up to 128 pieces. Since the history storage area is a ring buffer, if it exceeds 128, the oldest history is overwritten.

This history information can be checked in [CPU Log] in the [Tool] menu of Control Editor.

Caution

If the error is detected during the initial processing that is executed in power-on, time-stamp of history data may be recorded as the initial value.

(5) Self-diagnosis error codes

The list of error codes is shown below.

Table 7.1 List of error codes

Error code	Error name [Detecting timing]	Classification	Details of error	OK LED	Operation	Related special internal output	
						Bit	Word
–	Power off, Power error [Always check]	Serious failure	No power supply from the power module.	●	Stop	—	—
11	System ROM error [At power ON]	Serious failure	FLASH which is a copy source of the system has a sum error or cannot be read.	●	Stop	R7C8 R7DB	—
12	System RAM error [At power ON]	Serious failure	RAM used in the system program cannot be read and written properly.	●	Stop	R7C8 R7DB	—
13	Microcomputer error [Always check]	Serious failure	Address error interrupt and undefined command interrupt occurred in the microcomputer.	●	Stop	R7C8 R7DB	—
16	FLASH system error [Always check]	Serious failure	FLASH CRC anomaly used in system programs.	●	Stop	R7C8 R7DB	—
18	Ethernet MAC Address Anomaly [At power ON]	Serious failure	MAC address has a sum error.	⊕	Stop	R7C8 R7DB	—
23	Undefined instruction [Checking during operation]	Medium failure	The microcomputer code which cannot be decoded was performed and as a result, error was detected.	◔	Stop	R7C9 R7DB	—
27	Data memory error [At power On, at initializing]	Medium failure	Data memory cannot be read and written properly.	◔	Stop	R7DB	—
31	User memory error [At power On, during RUN, at initializing]	Medium failure	A sun error is detected in the user memory or the RUNNING memory.	◔	Stop	R7CA R7DB	—
41	I/O information verify error [Always check]	Minor failure	I/O assignment information and actual mounting of module do not match.	◐	Stop*1	R7CD R7DB	WRF002
42 *2	Option board verification error [Always check]	Minor failure	Option board information and actual mounting option board do not match. (OBV-AIG, OBV-AIOG, OBV-RTD)	◐	Stop*3	—	—
44	Overload error (Normal scan/constant scan) [During operation]	Minor failure	Performance time for the normal scan exceeded the overload check time set by the parameter. Constant scan did not end at specified Time.	◐	Stop*1	R7D1 R7DB	—
45	Overload error (Cyclic scan) [Cyclic processing]	Minor failure	Performance time for the cyclic scan exceeded the performance cycle.	◐	Stop*1	R7D2 R7DB	—
46	Overload error (Interrupt scan) [During operation]	Minor failure	Performance time for the interrupt scan exceeded the performance cycle.	◐	Stop*1	R7D3 R7DB	—
4F	Overload error (Cyclic scan) [During operation]	Minor failure	The different scan, of which priority is lower, from the executing scan is started during execution of the cyclic scan.	◐	Stop*1	R7D2 R7DB	—
5E	Option board error [Always check]	Warning	Detected a hardware error of the option board.	◐	Run	—	—
5F	Backup memory error [When writing settings and programs to flash]	Warning	FLASH memory erase failed.	◐	Run	—	—

🟢 : ON ⊕ : 250 ms OFF/250 ms ON ◔ : 500 ms OFF/500 ms ON ◐ : 1 s OFF/1 s ON ◕ : 2 s OFF/2 s ON ● : OFF

*1 Depending on the run parameter setting, operation can continue even when error occurs.
*2 Error code is added from software Ver.x126.
*3 Depending on the parameter for option board in Control Editor, operation can continue even when error occurs.

Table 7.1 List of error codes (continued from the preceding page)

Error code	Error name [Detecting timing]	Classification	Details of error	OK LED	Operation	Related special internal output	
						Bit	Word
61	RS-232C port transmission error (parity) [At transmission]	Warning	Parity error was detected during transmission.	🟢	Run	—	—
62	RS-232C port transmission error (Framing / Overrun) [At transmission]	Warning	Framing error, or overrun error was detected during transmission.	🟢	Run	—	—
63	RS-232C port transmission error (timeout) [At transmission]	Warning	Timeout error was detected during transmission.	🟢	Run	—	—
64	RS-232C port transmission error (protocol error) [At transmission]	Warning	Protocol (transmission procedure) error was detected during transmission.	🟢	Run	—	—
65	RS-232C port transmission error (BCC error) [At transmission]	Warning	Sum error was detected during transmission.	🟢	Run	—	—
67	RS-485 port transmission error (parity) [At transmission]	Warning	Parity error was detected during transmission.	🟢	Run	—	—
68	RS-485 port transmission error (Framing / Overrun) [At transmission]	Warning	Framing error, or overrun error was detected during transmission.	🟢	Run	—	—
69	RS-485 port transmission error (timeout) [At transmission]	Warning	Timeout error was detected during transmission.	🟢	Run	—	—
6A	RS-485 port transmission error (protocol error) [At transmission]	Warning	Protocol (transmission procedure) error was detected during transmission.	🟢	Run	—	—
6B	RS-485 port transmission error (BCC error) [At transmission]	Warning	Sum error was detected during transmission.	🟢	Run	—	—
6C	Ethernet port (Con 1) Timeout error [At transmission]	Warning	Timeout error of Ethernet communication port (Con 1) was detected during transmission.	🟢	Run	—	—
6D	Ethernet port (Con 2) Timeout error [At transmission]	Warning	Timeout error of Ethernet communication port (Con 2) was detected during transmission.	🟢	Run	—	—
6E	Ethernet port (Con 3) Timeout error [At transmission]	Warning	Timeout error of Ethernet communication port (Con 3) was detected during transmission.	🟢	Run	—	—
6F	Ethernet port (Con 4) Timeout error [At transmission]	Warning	Timeout error of Ethernet communication port (Con 4) was detected during transmission.	🟢	Run	—	—
71	Battery error [Always check]	Warning	Battery voltage dropped below prescribed value. Battery is not installed.	◕	Run	R7D9	—
76	Retentive data area is undefined status [At power ON]	Warning	Area specified to retentive data area is undefined because the battery is dead.	◕	Run	—	—
77	Backup memory error (FLASH) [in writing into FLASH memory]	Warning	Data cannot be written into the flash memory.	◕	Run	R7D9	—
78	Setting parameter data is undefined status.	Warning	Serial communication stored in CPU, Ethernet communication, and parameter set by security function have sum error.	◕	Run	—	—
7B	Data memory backup error [Power ON]	Warning	Checksum value is wrong in backup data.	◕	Run	—	—
7C	Logging setting error [Power ON]	Warning	Checksum value is wrong in the logging setting stored in MICRO-EHV.	◕	Run	—	—
7D	Analog option board conversion processing delay [Always check]	Warning	Conversion of analog values had not completed within the set time.	🟢	Run	—	—

🟢 : ON ⊕ : 250 ms OFF/250 ms ON ◔ : 500 ms OFF/500 ms ON ◐ : 1 s OFF/1 s ON ◕ : 2 s OFF/2 s ON ○ : OFF

Table 7.1 List of error codes (continued from the preceding page)

Error code	Error name [Detecting timing]	Classification	Details of error	OK LED	Operation	Related special internal output	
						Bit	Word
81	Modbus Gateway port transmission error (parity) [At transmission]	Warning	A parity error was detected during transmission.	☀	Run	—	—
82	Modbus Gateway port transmission error (framing/ overrun) [At transmission]	Warning	A framing error or overrun error was detected during transmission.	☀	Run	—	—
83	Modbus Gateway port transmission error (timeout) [At transmission]	Warning	A timeout error was detected during transmission	☀	Run	—	—
85	Modbus Gateway port transmission error (CRC error) [At transmission]	Warning	A sum error was detected during transmission.	☀	Run	—	—
88	WDT error [Always check]	Serious error	Watchdog timer detects microprocessor abnormality because the microprocessor no longer works according to the system program.	●	Stop	R7C8[*5] R7DB[*5]	—
8C	Modbus-TCP port (Con 1) Timeout error [At transmission]	Warning	Timeout error of Modbus-TCP communication port (Con 1) was detected during transmission.	☀	Run	—	—
8D	Modbus-TCP port (Con 2) Timeout error [At transmission]	Warning	Timeout error of Modbus-TCP communication port (Con 2) was detected during transmission.	☀	Run	—	—
8E	Modbus-TCP port (Con 3) Timeout error [At transmission]	Warning	Timeout error of Modbus-TCP communication port (Con 3) was detected during transmission.	☀	Run	—	—
8F	Modbus-TCP port (Con 4) Timeout error [At transmission]	Warning	Timeout error of Modbus-TCP communication port (Con 4) was detected during transmission.	☀	Run	—	—
A0	USB memory device error [When USB memory is inserted]	Warning	The USB device could not be detected correctly.	☀	Run	—	—
A1	USB memory file open error [When opening a file]	Warning	File in USB memory could not open.	☀	Run	—	—
A2	USB memory file write error [When writing to file]	Warning	File in USB memory could not be written.	☀	Run	—	—
A3	USB memory file read error [When reading from file]	Warning	File in USB memory could not be read.	☀	Run	—	—
A4	Work memory allocation error [When accessing memory]	Warning	Work memory for file edit could not be allocated.	☀	Run	—	—
A7 *4	Too many files opened	Warning	Number of files that can be opened at the same time has exceeded the upper limit. (12 files)	☀	Run	—	—
A8 *4	Downloaded program includes not supported function	Warning	Program downloaded from USB memory includes the parameter which is not correspond with this unit.	☀	Run	—	—
A9 *4	Downloaded program has not sufficient information	Warning	Program downloaded from USB memory does not contain the parameters required to use new features.	☀	Run	—	—

☀: ON ◍: 250 ms OFF/250 ms ON ◔: 500 ms OFF/500 ms ON ◐: 1 s OFF/1 s ON ◕: 2 s OFF/2 s ON ●: OFF

*4 Error code is added from software Ver.x126.

*5 Related special internal output are added from Ver.x126.

Note

The OK LED display pattern has been changed from MICRO-EHV basic unit software Ver.x104.

The MICRO-EHV basic unit software Ver.x101 and Ver.x102 before the change are indicated by the OK LED below.

Error Code [18] [23] [27] [31] [41] [44] [45] [46] [4F] : ● OFF

Error Code [5E] [5F] [78] : ☀ ON

Error Code [71] [76] [77] : ◐ 1s OFF/1s ON

The range of the special internal output that is cleared when R7EC is set to 1 is shown below.

No.	Bit special internal output
R7C8	Serious failure flag
R7C9	Operation microcomputer abnormal
R7CA	User memory error
R7CD	I/O verify mismatch
R7D0	Overload error (normal scan)
R7D1	Overload error (cyclic scan)
R7D2	Overload error (interrupt scan)
R7D6	I/O assignment points over
R7D8	Clock error
R7D9	Battery error
R7DB	Self-diagnostic error
R800	Undefined IP address
R801	Undefined serial communication setup
R802	Undefined Ethernet task code function setup
R803	Undefined ASR function setup
R804	Undefined NTP function setup
R805	Undefined security function parameter
R806	Undefined ASR mode setup
R807	Undefined Modbus-TCP parameter

No.	Word special internal output
WRF000	Self-diagnostic error code
WRF002	I/O verify mismatch details

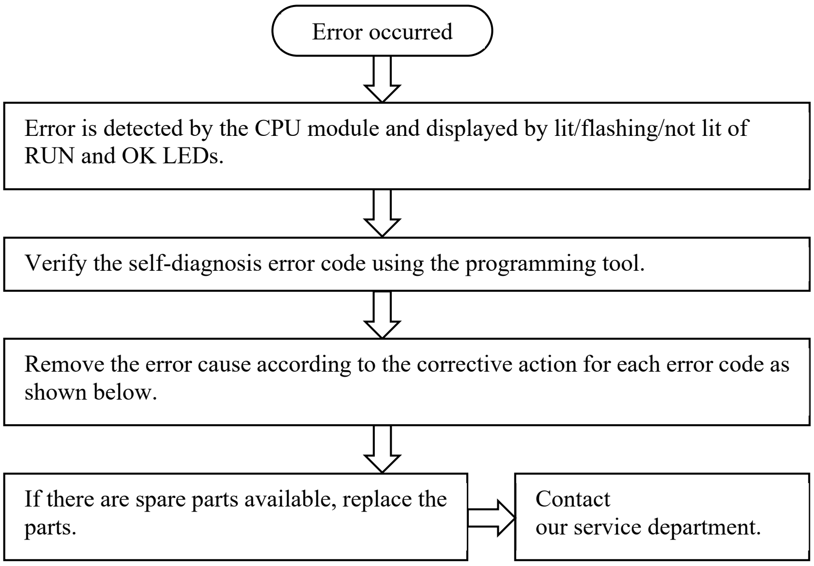
When all of the special internal output data cannot be cleared during program execution, refer to the self-diagnostic error code list and clear only the corresponding error flags by using forced set of the programmer or peripheral unit.

Note

If the internal output for a self-diagnostic error R7DB (WRF000) is used as a system error for the stop condition of CPU RUN, the R7DB may be turned on even with an error of the warning level (battery error, etc.), causing the CPU to stop. Therefore, do not use the internal output of the self-diagnostic error as a condition for stopping the CPU.

(6) Corrective action when an error occurred

The process flow when an error occurred is shown below.



Error occurred

Error is detected by the CPU module and displayed by lit/flashing/not lit of RUN and OK LEDs.

Verify the self-diagnosis error code using the programming tool.

Remove the error cause according to the corrective action for each error code as shown below.

If there are spare parts available, replace the parts.

Contact our service department.

Error code	Error name	Corrective action
88	Microcomputer overload error	Restart the power. If the same error occurs, it is a hardware error in the basic unit. Replace the CPU module with a spare. Make sure that there are no machines, etc. that generate excessive noise near MICRO-EHV.
11	System ROM error	
12	System RAM error	
13	Microcomputer error	
16	FLASH system error	
18	Ethernet MAC Address Anomaly	
23	Undefined instruction	
27	Data memory error	
–	Power shut-off, power supply error	Check the basic and expansion power supply voltage.
31	User memory error	The contents of the user program are destroyed. Transfer the program again after initialization. This is displayed if the battery is left removed or the battery is left dead for a long period.
41	I/O information verify error	Check the I/O assignment once more. Recheck the fixation of each basic unit and expansion unit, and the connection of the expansion cable.
42	Option board verification error	Check the parameter setting for the option board once more. Recheck the connection of the option board and basic unit.
44	Overload error (Normal scan)	Change the program so that the scan time of the user program is less or change the congestion check time.
45	Overload error (Cyclic scan)	Change the program so that the cyclic interrupt program execution time is less.
46	Overload error (Interrupt scan)	Change the program so that the interrupt program execution time is less.
4F	Overload error (Cyclic scan)	Set a cycle of all cyclic scan to a multiple of integer.

Error code	Error name	Corrective action
5E	Option board error	Check for abnormal in option board and replace the malfunctioning module.
5F	Backup memory error	Transfer the program again after CPU initialization.
61	RS-232C port transmission error (parity)	Check the connection of the communication cable. Check the settings such as the transmission speed. Check there are no sources of noise near the communication cable.
62	RS-232C port transmission error (framing / overrun)	
63	RS-232C port transmission error (timeout)	Check the connection of the communication cable. Check there are no sources of noise near the communication cable.
64	RS-232C port transmission error (protocol error)	Verify the protocol specification, examine the host computer processing, and correct any error.
65	RS-232C port transmission error (BCC error)	
67	RS-485 port transmission error (parity)	Check the connection of the communication cable. Check the settings such as the transmission speed. Check there are no sources of noise near the communication cable.
68	RS-485 port transmission error (framing / overrun)	
69	RS-485 port transmission error (timeout)	Check the connection of the communication cable. Check there are no sources of noise near the communication cable.
6A	RS-485 port transmission error (protocol error)	Verify the protocol specification, examine the host computer processing, and correct any error.
6B	RS-485 port transmission error (BCC error)	
6C	Ethernet port (Con 1) timeout error	Check the connection of the communication cable. Verify if the other device is operating normally. Check there are no sources of noise near the communication cable.
6D	Ethernet port (Con 2) timeout error	
6E	Ethernet port (Con 3) timeout error	
6F	Ethernet port (Con 4) timeout error	
71	Battery error	Replace the battery with a new one. Check the connection of the battery connector.
76	Retentive data area is undefined status	Clear the retentive data area
77	Backup memory error	Read the programming tool from the user program and back up it. Though the basic unit can operate unless the program is changed, replace with a spare part because of a hardware error of the CPU module.
78	Each setting parameter undefined	Set using the programming tool again.
7B	Data memory backup error	The backup data is abnormal. After executing data backup again, restart the power supply and check whether the data backup is performed normally.
7C	Logging setting error	The logging setting is abnormal. After setting the parameter for logging again, and check whether the logging is performed normally.
7D	Analog option board conversion processing delay	
81	Modbus Gateway port transmission error (Parity)	Check the connection of the communication cable. Verify if the other device is operating normally. Check there are no sources of noise near the communication cable.
82	Modbus Gateway port transmission error (Framing/Overrun)	
83	Modbus Gateway port transmission error (Timeout)	
85	Modbus Gateway port transmission error (CRC error)	

Error code	Error name	Corrective action
8C	Modbus-TCP port (Con1) Timeout error	Check the connection of the communication cable. Check there are no sources of noise near the communication cable. Verify the protocol specification, examine the host computer processing, and correct any error.
8D	Modbus-TCP port (Con2) Timeout error	
8E	Modbus-TCP port (Con3) Timeout error	
8F	Modbus-TCP port (Con4) Timeout error	
A0	USB memory device error	Check the connection of the USB memory Check there are no sources of noise in USB memory.
A1	USB memory file Open error	
A2	USB memory file Write error	
A3	USB memory file Read error	
A4	Work memory allocation error	
A7	Too many files opened	Check the ladder program for data logging. Under the logging condition is "Specify file number in internal output" and "Add to same file", if the file number in the internal output is updated by the program, a new file will be generated with the previous file open. If the number of files opened at the same time exceeds 12, A7 error will occur.
A8	Downloaded program includes not supported function	The basic unit does not support the additional functions stored in the OBC file. Replace with the basic unit that supports the additional functions. This error occurs when the software version of MICRO-EHV is old, but if you do not use the additional functions, you can use it as it is even if this error occurs.
A9	Downloaded program has not sufficient information	The downloaded OBC file lacks information about additional functions. Create the OBC file with the Control Editor that supports the additional function, or upload the OBC file again from MICRO-EHV that supports the additional function. This error occurs when the software version of the MICRO-EHV is new, but if you do not use the additional functions, you can use it as it is even if this error occurs.

⬇

Perform the following procedure to erase the error display.

(a) When CPU is stopped

Turn the CPU RUN switch to “STOP”, then to “RUN” again.

The OK LED turns on if the error has been corrected.

However, the error information remains in the error special internal output, which stored the CPU error types and details.

(It is possible to analyze error after recovery.)

To reset the error information, perform the procedures shown in (b).

(b) When the CPU is running (RUN)

Set the special internal output R7EC to 1.

If the error has been corrected, the OK LED will be lit and the error information set in the error special internal output, which stores the type and details of the CPU error, will be reset.

7.2 Operation error code and countermeasure procedures

When an error occurs in an execution of a control command, "1" will set to the internal special output (R7F3) for the operation error (ERR), and an error code that indicates the error description will be set to WRF015.

Please perform "R7F3 = 0" by the forced sets from a program or a peripheral device to clear the operation error. Please perform "WRF015 = 0" by the forced sets from a program or a peripheral device to clear the error code.

Table 7.2 Operation error code

Error code	Name of the error	Description	The command that an error occurs
H0041	CAL nesting overflow	Number of nesting layers exceeds six in a subroutine.	CAL
H0046	FOR - NEXT nesting overflow	Number of nesting layers exceeds six in a "FOR - NEXT".	FOR NEXT

7.3 Check list when abnormality occurred

Check the following items if an error is generated in the MICRO-EHV system. Please contact our service department if there are no problems in the following items.

(1) Power supply related items

- Is the power voltage correct? (85 to 264 V AC)
- Are there any warps in the power supply waveform?
- Are there any excessive noises in the power supply?
- Is power supplied for all basic and expansion modules?
- Is the power supply supplied from the power supply for the sensor within the specification (430 mA)?
- Is the POW LED lit? Are the turn off or blinking?

(2) The basic unit related items

- Are the initial settings (CPU initialization, I/O assignment, parameter settings, etc.) proper?
- Is the OK LED turn off or blinking?
- Is the RUN switch in the proper location?
- Are batteries mounted properly? Is the battery life still remaining?
- Are the expansion connector and communication cable properly connected?

(3) Input related items

- Is the input voltage within the specifications?
- Is there any noise or chattering in the input?
- Do the I/O assignment numbers in the program match?
- Is the wiring done properly?

(4) Output related items

- Do the output specifications of the unit and the load power type (DC/AC) match?
- Do the load voltage and current match the specification of the output section?
- Is there any noise or chattering in the output waveform?
- Is the wiring done properly?
- Do the I/O assignment numbers in the program match?
- Are there any unintentional overlaps in the output numbers?

(5) Wiring related items

- Is the FE terminal of the power supply wiring grounded using Class D dedicated grounding?
- Is the wiring between the expansions mixed up with other wires?
- Are the power supply wiring and I/O cables separated?
- Are there any foreign substances in the connector of each unit?

Note

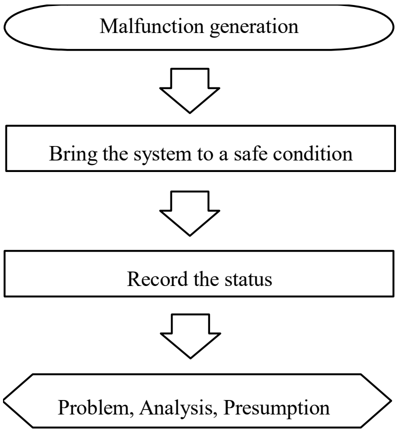
(1) Always replace the unit with the power off.

(2) Please notify us of the malfunctioning effect in as much detail as possible when returning the unit for repair. (including error codes, malfunctioning I/O bit No., will not turn on or off, etc.)

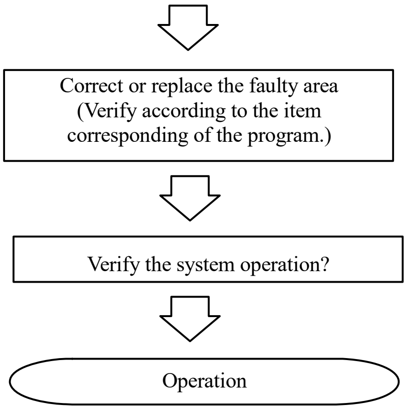
(3) The tools and devices necessary for troubleshooting are briefly as follows:
Phillips/flathead drivers, digital multimeter, tester, oscilloscope (necessary depending on the case) etc.

7.4 Procedure to solve abnormality

The following shows the processing flow when a problem has occurred:

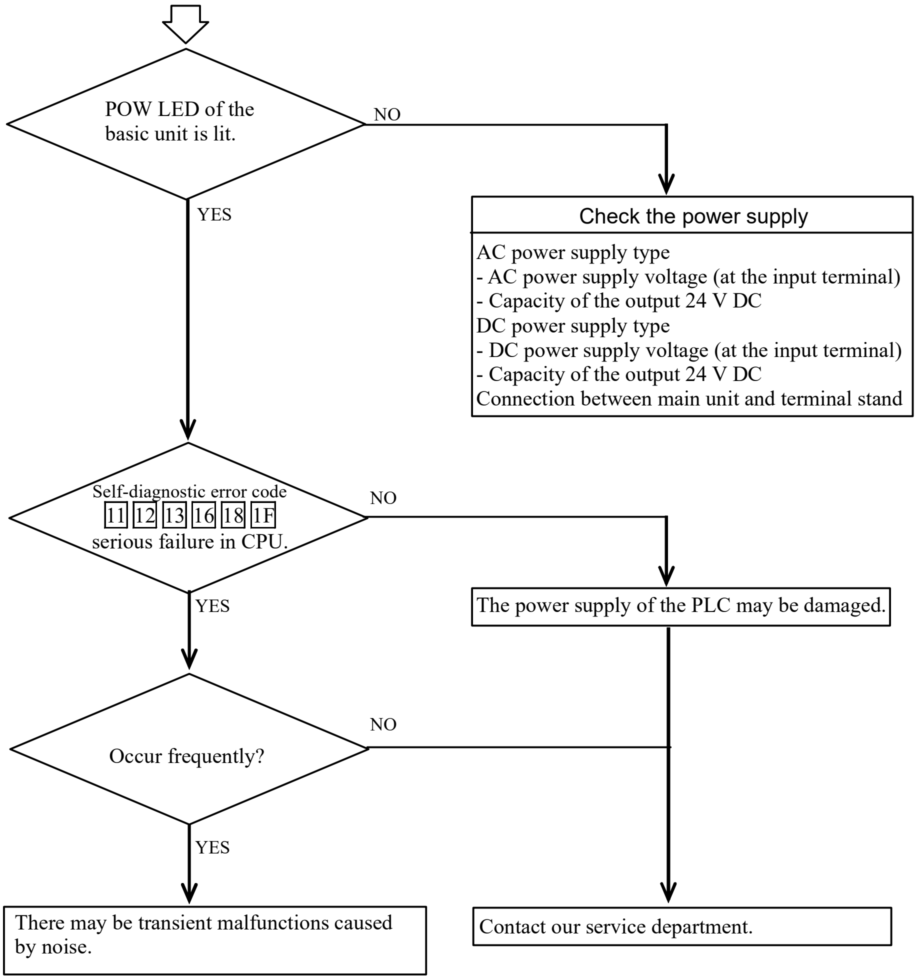


Major problem	Check point	Typical causes of the problem	Reference item
PLC will not start	POW LED, OK LED	Power supply abnormal, power off, power supply capacity shortage, serious CPU failure	(a)
Will not operate (will not RUN)	POW LED, OK LED, RUN LED, Internal output of error	I/O assignment error, incorrect parameter setting, incorrect user program, Syntax error, RUN conditions not met, write-occupied status	(b)
Operation stopped (RUN stopped)	POW LED, OK LED, RUN LED, CPU error code	Power supply abnormal, expansion power supply abnormal/off, CPU abnormal, memory error.	(c)
Wrong input, or will not input. (abnormal operation)	OK LED, RUN LED, I/O LED Monitoring by peripheral devices	User program timing, input power supply, bad connection, I/O external input circuit error, I/O inductive noise.	(d)
Wrong output, or will not output. (abnormal operation)	OK LED, RUN LED, I/O LED, Monitoring by peripheral devices Forced setting	User program timing, bad connection, I / O output circuit error, I/O inductive noise.	(e)
Peripheral devices abnormal	CPU error code, Peripheral device	Serious CPU failure, peripheral devices abnormal, peripheral devices setting error, cable abnormal	(f)



(a) PLC will not start

[The OK LED on basic unit does not light up when the power is turned on, and peripherals cannot be connected online.]

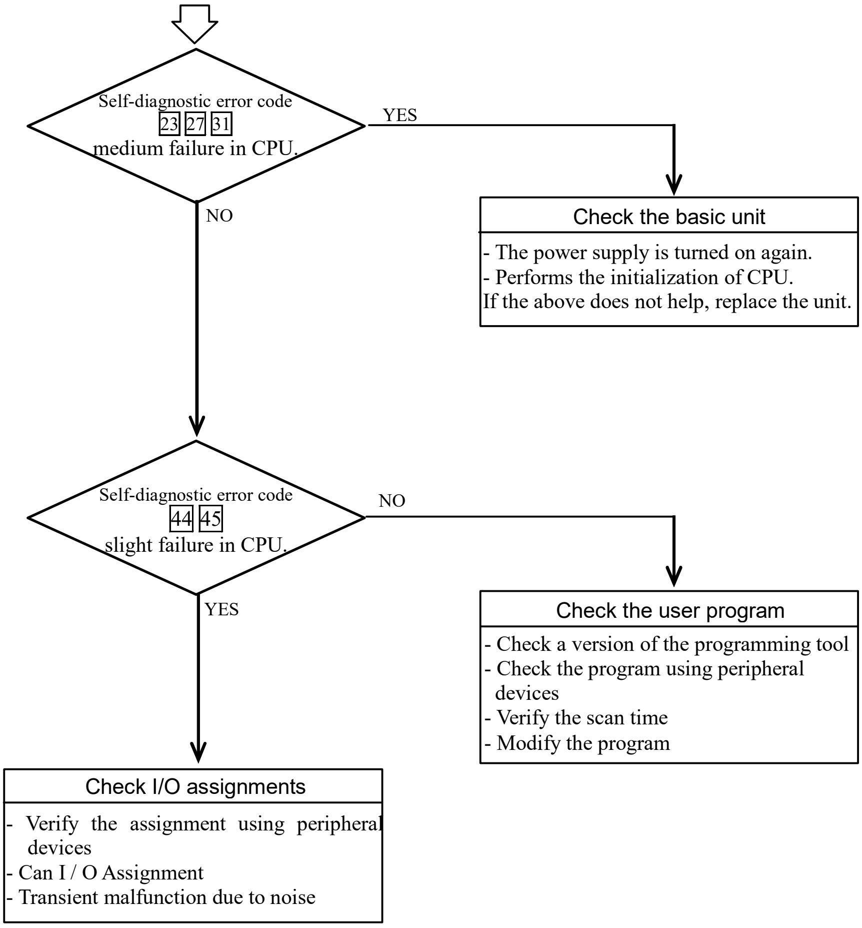


(b) Will not operate (will not RUN)

[Even if the operation conditions of PLC are met, CPU does not operate (RUN LED does not light up) and remains stopped. However, peripheral devices go on-line.]

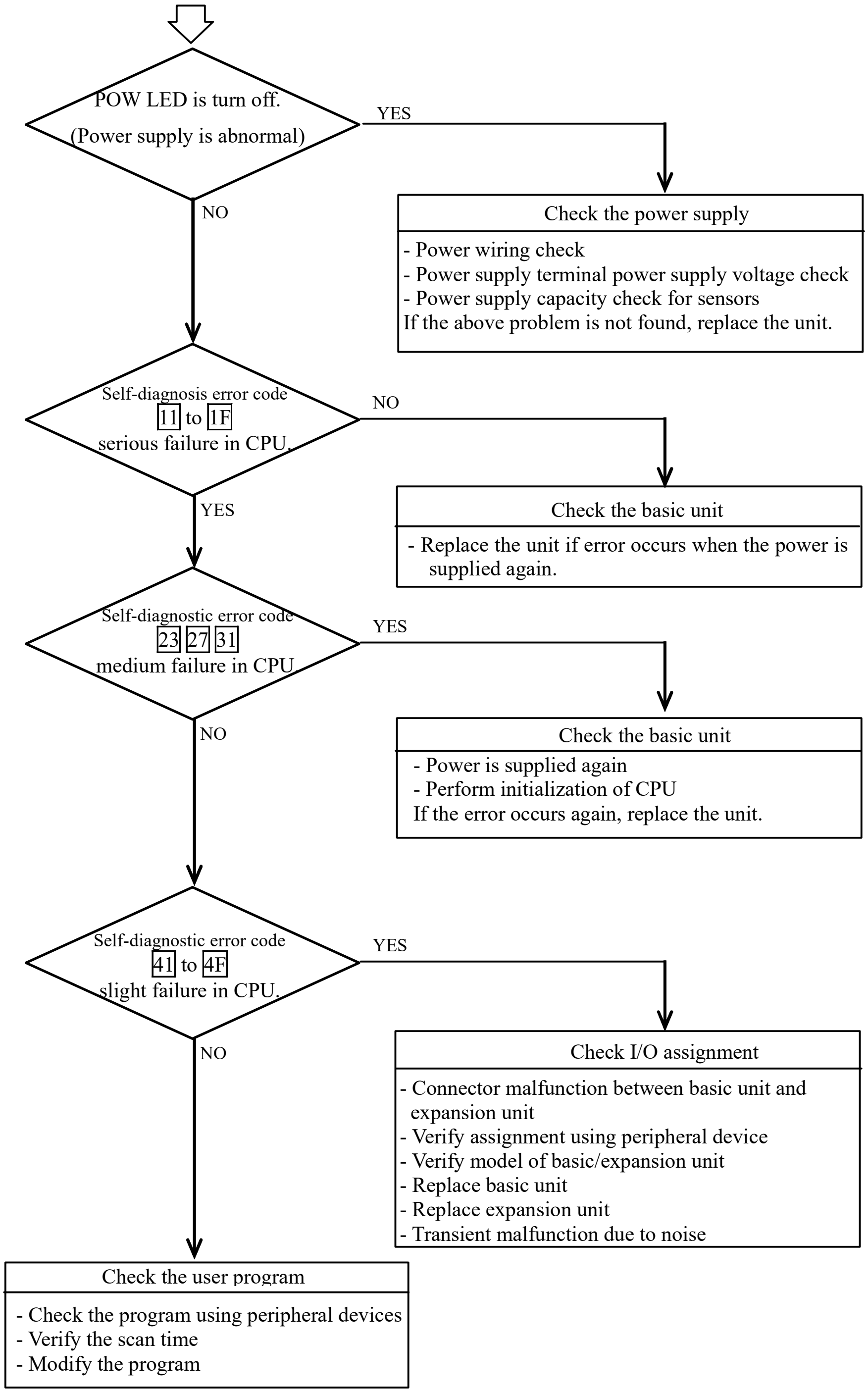
Caution

If the CPU is in the WRITE occupied state, switching the RUN switch from "STOP" to "RUN" will not cause the CPU to RUN. Connect the device and perform the occupancy operation.



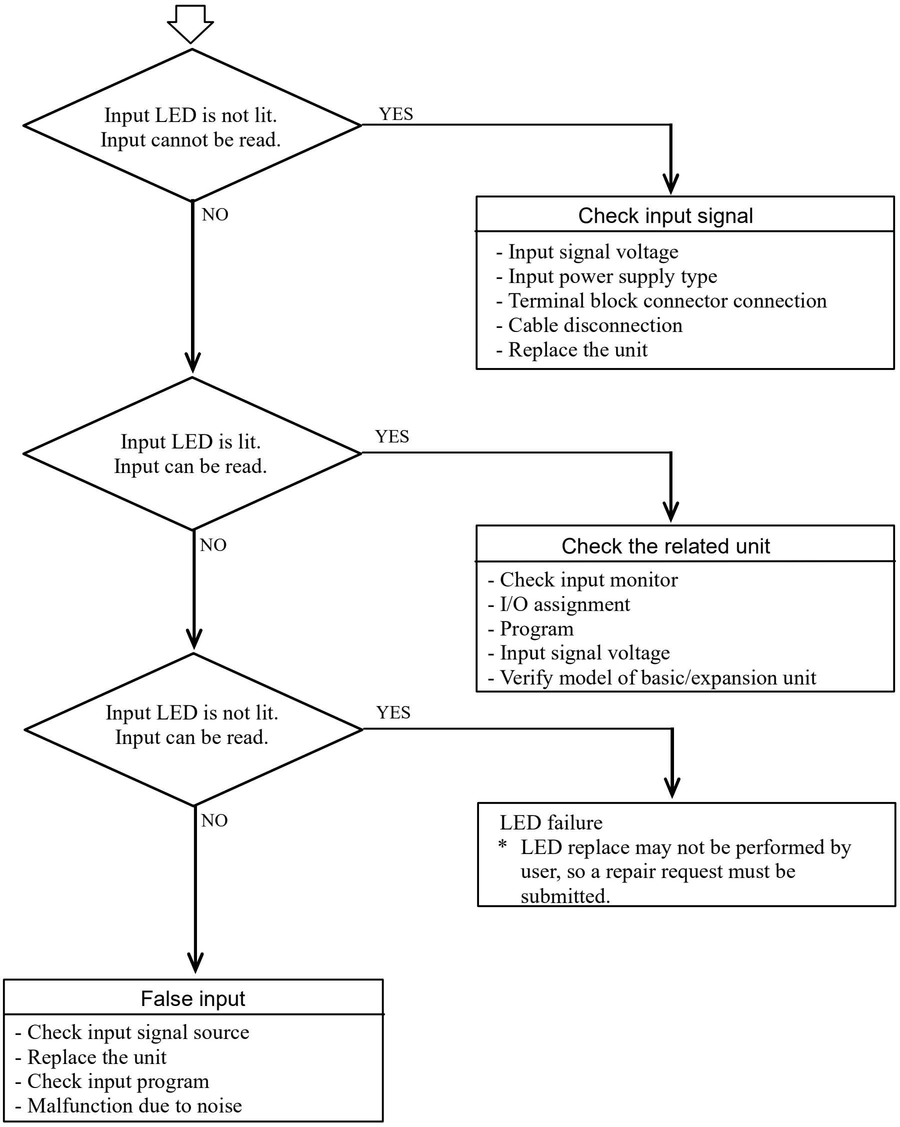
(c) Operation stopped (RUN stopped)

[CPU stops suddenly (the RUN LED goes out) during normal operation]

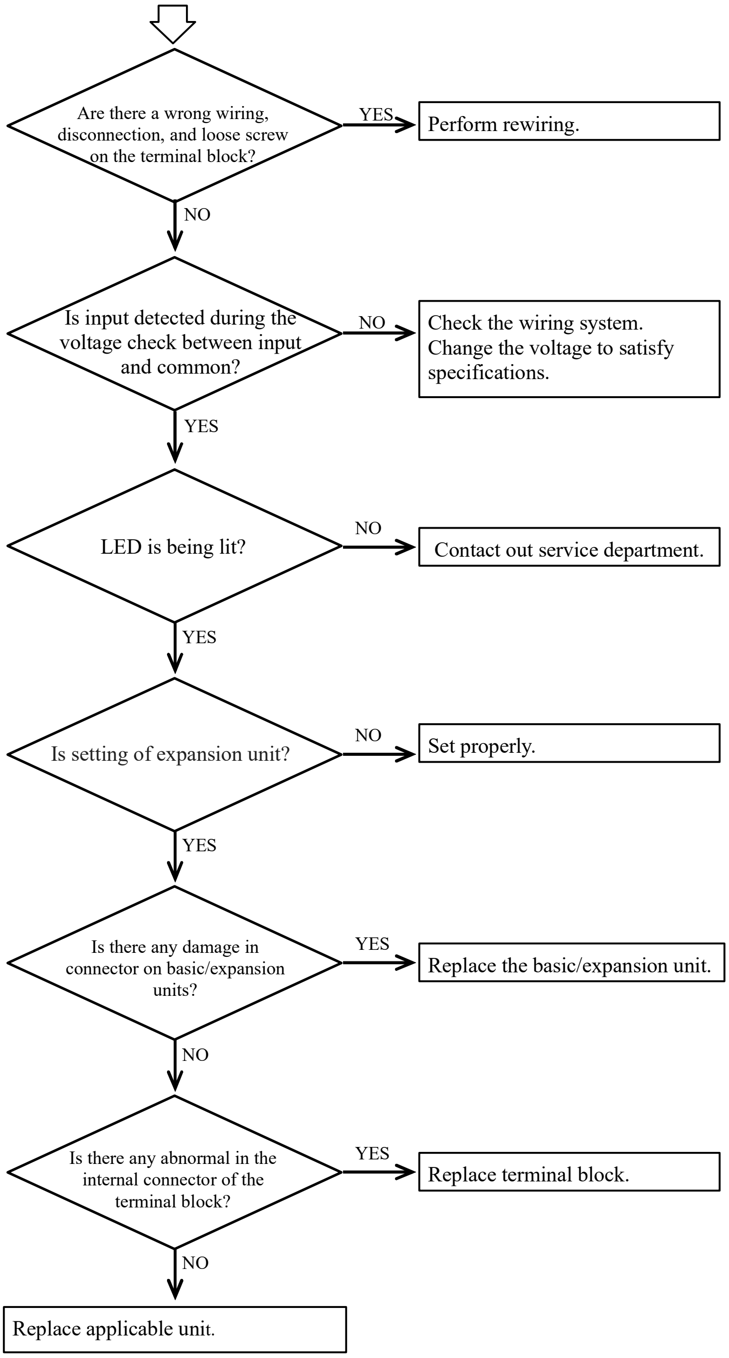


(d) Wrong input, or will not input. (Operation error)

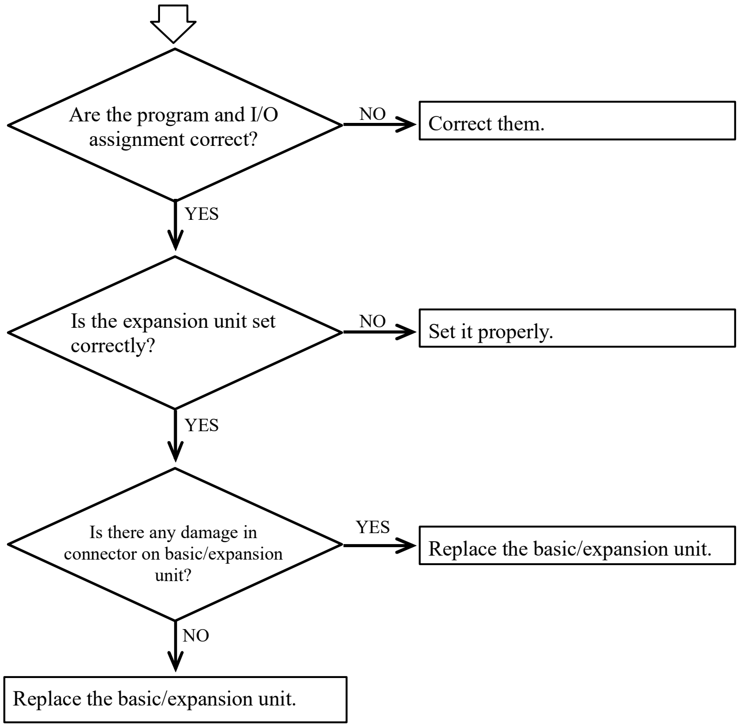
[CPU runs, but the input data is not correct.]



[Data cannot be input.]

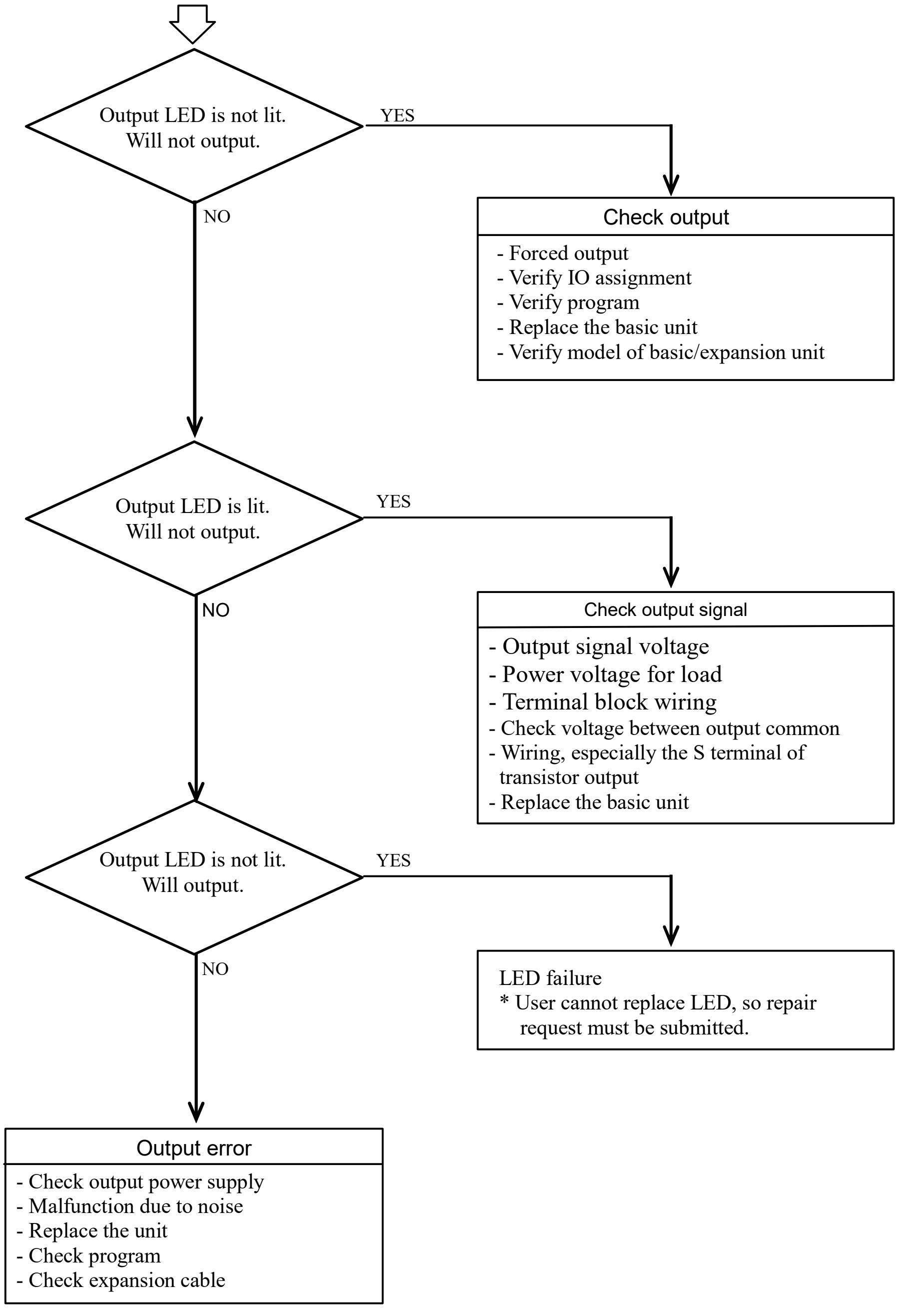


[I/O assignment error occurs, but data is read.]

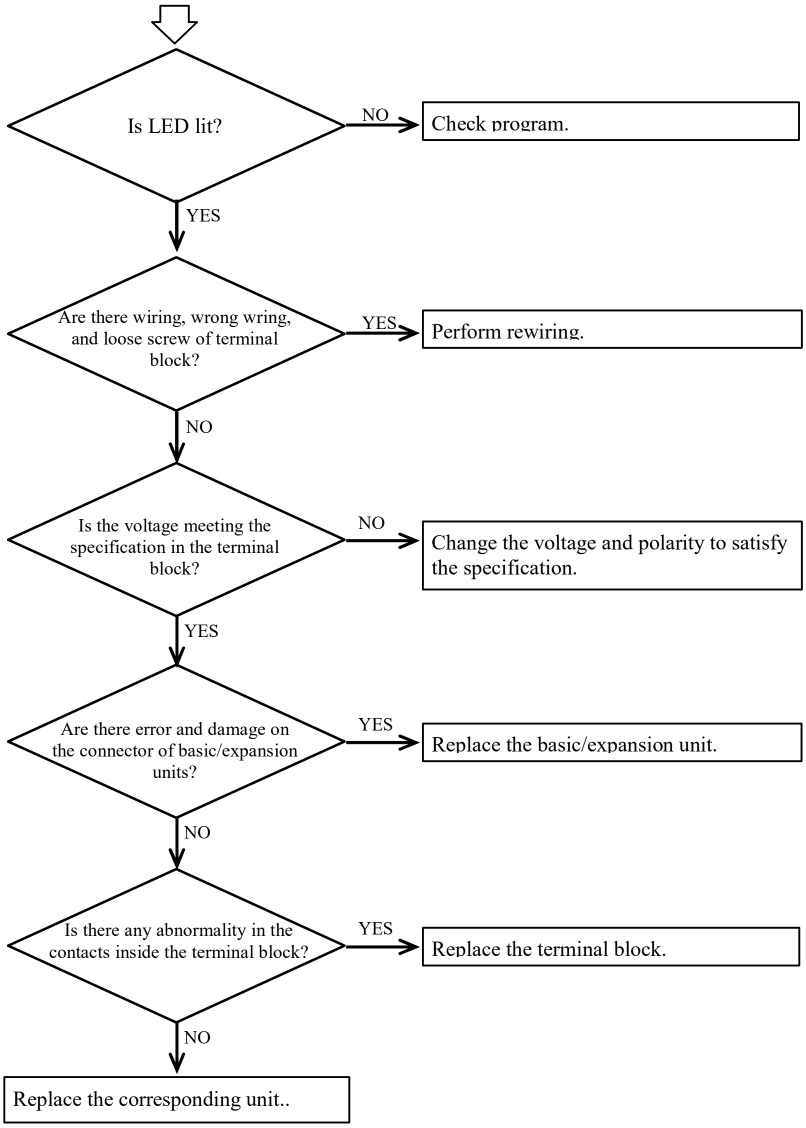


(e) Output error, no output (abnormal operation)

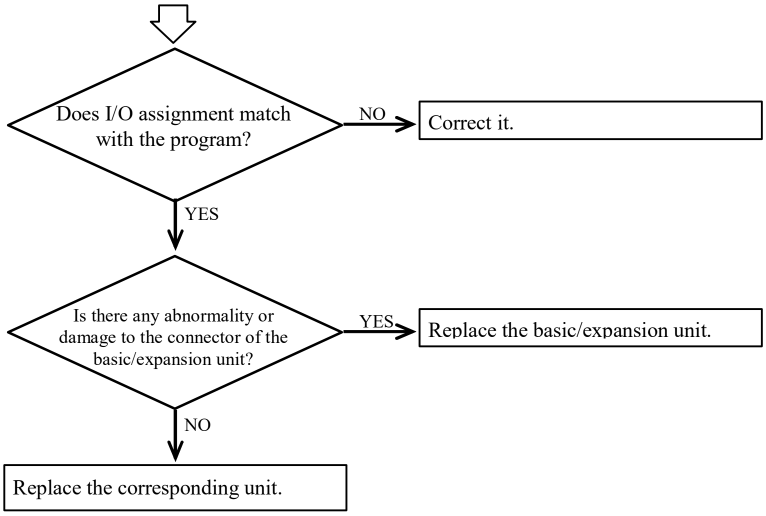
[CPU runs, but output signals are not correct.]



[CPU runs, but output signal are not detected.]

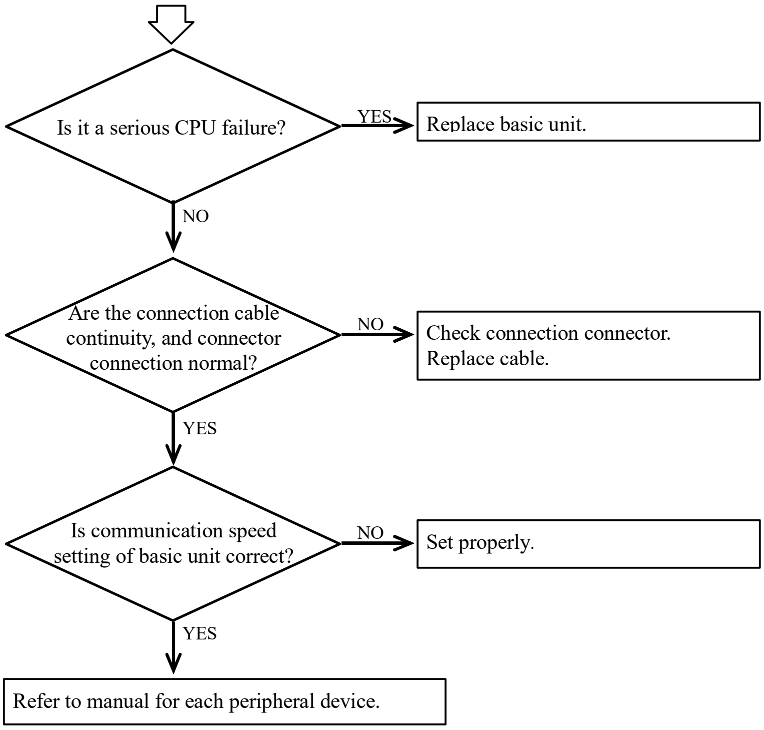


I/O assignment error occurs, but output is normal.

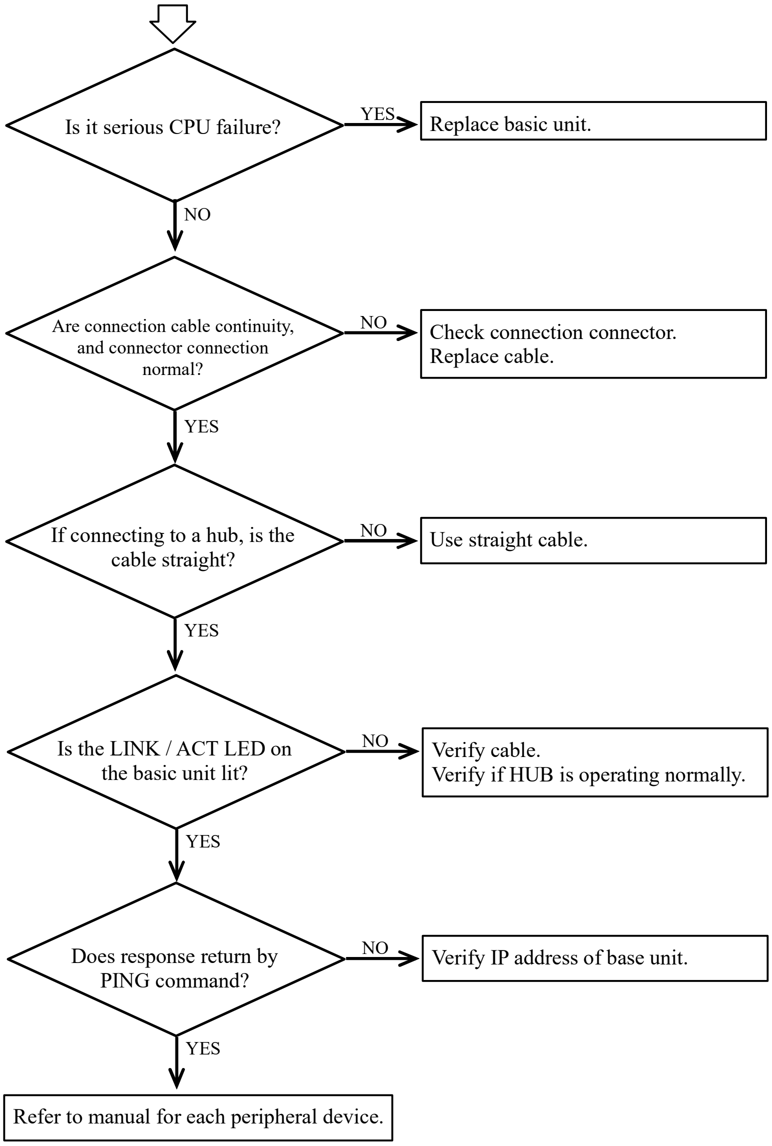


(f) Peripheral devices abnormal

[Peripheral devices cannot be connected with serial communication port.]



[Cannot connect with Ethernet port]



MEMO

Appendix1 List of Special Internal Output

The special internal output is an internal output of bit or word that special functions are assigned.

The special internal output area is retentive area, the value is retained even if the power supply is OFF if the battery is installed.

A1.1 Bit special internal output

The bit special internal output area is from R7C0 to RFFF. The functions of each bit are shown in the following table.

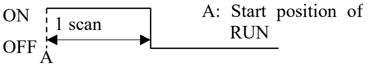
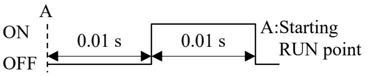
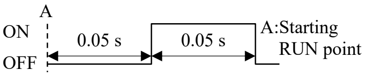
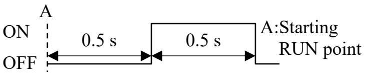
No	Name [Main use]	Meaning	Description	Setting condition	Resetting condition
R7C0 to R7C7	Undefined	—	—	—	—
R7C8	Serious failure flag	0: No serious failure 1: Serious failure	Indicates whether there is an abnormal in system.	S	U, R7EB, R7EC
R7C9	Microcomputer error	0: Normal 1: Error	Indicates whether there is an abnormal in the microcomputer.	S	U, R7EB, R7EC
R7CA	User memory error	0: Normal 1: Error	Indicates whether there is an abnormal in the user memory.	S	U, R7EB, R7EC
R7CB	Undefined	—	—	—	—
R7CC	Undefined	—	—	—	—
R7CD	I/O verify mismatch	0: Normal 1: Unmatched	Indicates whether I/O assignment and loading are matched. (Mismatched information output to WRF002)	S	U, R7EB, R7EC
R7CE	Undefined	—	—	—	—
R7CF	Undefined	—	—	—	—
R7D0	Undefined	—	—	—	—
R7D1	Overload error (normal scan)	0: Normal 1: Scan time over	Indicates whether the normal scan execution time has exceeded the designated time.	S	U, R7EB, R7EC
R7D2	Overload error (cyclic scan)	0: Normal 1: Scan time over	Indicates whether the cyclic scan was completed within cycle time.	S	U, R7EB, R7EC
R7D3	Overload error (interrupt scan)	0: Normal 1: Scan time over	Indicates whether an interrupt of the same factor occurred during interrupt scan execution.	S	U, R7EB, R7EC
R7D4	Undefined	—	—	—	—
R7D5	Undefined	—	—	—	—
R7D6	IO assignment points over	0: Normal 1: I/O assignment points over	Indicates whether the number of I/O assigned points has exceeded the maximum points.	S	U, R7EB, R7EC
R7D7	Undefined	—	—	—	—
R7D8	Clock error	0: Normal 1: Error	When clock IC is in error, this bit is activated.	S	U, R7EC
R7D9	Battery error	0: Normal 1: Error	Indicates the battery voltage drop or the backup memory abnormal.	S	U *1 R7EB, R7EC
R7DA	Undefined	—	—	—	—
R7DB	Self-diagnostic error	0: Normal 1: Error	Indicates whether there is a self-diagnostic error. (Detailed information output to WRF000)	S *2	U, R7EB, R7EC
R7DC to R7DF	Undefined	—	—	—	—

Set / Reset Condition :

S…ON / OFF by system, U…ON / OFF by user, R7EB…Set 1 to R7EB, R7EC…Set 1 to R7EC,
X…Always display

*1 Battery error (R7D9) will be turned off if the cause of the error is removed by replacing the battery.

*2 Self-diagnostic error (R7DB) turns on only when there is the serious, medium, or minor failure. Unlike EH-CPU, it does not turn on with the warning level error.

No	Name [Main use]	Meaning	Description	Setting condition	Resetting condition
R7E0	RUN switch position (STOP)	0: RUN/STOP Sw; RUN 1: RUN/STOP Sw; STOP	Turn on when RUN/STOP Sw position is STOP.	S	S
R7E1	Undefined	—	—	—	—
R7E2	RUN switch position (RUN)	0: RUN/STOP Sw; STOP 1: RUN/STOP Sw; RUN	Turn on when RUN/STOP Sw position is RUN.	S	S
R7E3	The first scan is ON after RUN	0: The 2nd scan or later after RUN 1: The 1st scan after RUN		S	S
R7E4	Always ON	0: Non-status of 0 1: Always	Always outputs 1 regardless of CPU status.	S	Cannot be turned OFF.
R7E5	0.02 second clock	0: 0.01 seconds 1: 0.01 seconds		S	S
R7E6	0.1 second clock	0: 0.05 seconds 1: 0.05 seconds		S	S
R7E7	1.0 second clock	0: 0.5 seconds 1: 0.5 seconds		S	S
R7E8	Occupied flag	0: Unoccupied 1: Occupied	Indicates occupancy status from the peripheral unit.	S	S
R7E9	RUN prohibited	0: Operation allowed 1: Operation prohibited	Indicates whether it is operation prohibited status.	S	S
R7EA	Executing a online change in RUN	0: Not being executed 1: Being executed	Indicates whether operation is temporarily stopped (output hold) due to online change in RUN.	S	S
R7EB	Clear the retentive area	0: - 1: Clear the retentive area and error	Clear the retentive area and error special internal output. (WRF000, WRF002, R7C8 to R7CA, R7CD, R7D1 to R7D3, R7D6, R7D9, R7DB, R800 to R804, R806, R807, R9DC to R9DF)	U	S
R7EC	Clear error special internal output	0: - 1: Clear error information	Clear error special internal output (WRF000, WRF002, R7C8 to R7CA, R7CD, R7D1 to R7D3, R7D6, R7D8, R7D9, R7DB, R800 to R804, R806, R807, R9DC to R9DF), ERR Display, CPU status	U	S
R7ED	Undefined	—	—	—	—
R7EE	Undefined	—	—	—	—
R7EF	Backup memory writing execution flag	0: Write complete 1: Write in progress	Indicates whether or not data is being written to the backup memory.	S	S
R7F0	Carry flag (CY)	0: No carry 1: Carry	Indicates whether there is a carryover from the operation result.	S	S
R7F1	Overflow flag (V)	0: No overflow 1: Overflow	Indicates whether there is overflow in the operation result.	S	S
R7F2	Shift data (SD)	0: Shift data "0" 1: Shift data "1"	Designates the shift data used in shift instructions, etc.	U	U
R7F3	Operation error (ERR)	0: Normal 1: Error	Indicates whether there is an operation error when operation is executed.	S	S
R7F4	Data error (DER)	0: Normal 1: Error	Indicates whether there is a data error when operation is being executed.	S	S
R7F5 to R7F7	Undefined	—	—	—	—

Set / Reset Condition :

S…ON / OFF by system, U…ON / OFF by user, R7EB…Set 1 to R7EB, R7EC…Set 1 to R7EC, X…Always display

No	Name [Main use]	Meaning	Description	Setting condition	Resetting condition
R7F8	Calendar, clock read request	1: Read	Read the present values of calendar, clock and set in WRF01B to WRF01F	U	S
R7F9	Calendar, clock setting request	1: Set	Set the data set in WRF01B to WRF01F in the calendar and clock.	U	S
R7FA	Clock ±30 second adjustment request	1: Request adjustment	When second data (WRF00F) is 0 to 29, it becomes 0 seconds and when it is 30 to 59, +1 minute is added and second data becomes 0.	U	S
R7FB	Calendar and clock set data error	0: Normal 1: Error	Indicates whether there is an error in calendar and clock set data.	S	S
R7FC	Pulse/PWM running flag 1 (Y100)	0: Pulse/PWM output stopped 1: Pulse/PWM output	ON during pulse / PWM output at Y100.	S	S
R7FD	Pulse/PWM running flag 2 (Y101)	0: Pulse/PWM output stopped 1: Pulse/PWM output	ON during pulse / PWM output at Y101.	S	S
R7FE	Pulse/PWM running flag 3 (Y102)	0: Pulse/PWM output stopped 1: Pulse/PWM output	ON during pulse / PWM output at Y102.	S	S
R7FF	Undefined	—	—	—	—
R800 *3	IP address undefined	0: Normal 1: Error	IP address of Ethernet port is unfixed.	S	U, R7EB, R7EC
R801 *3	Serial communication setting data undefined	0: Normal 1: Error	Setup about serial communication port is unfixed.	S	U, R7EB, R7EC
R802 *3	Ethernet task code communication setting data undefined	0: Normal 1: Error	Parameter about task code function of Ethernet port is not fixed.	S	U, R7EB, R7EC
R803 *3	ASR function setting data undefined	0: Normal 1: Error	Parameter about ASR function of Ethernet port is not fixed.	S	U, R7EB, R7EC
R804 *3	NTP setting data undefined	0: Normal 1: Error	Parameter about NTP function of Ethernet port is not fixed.	S	U, R7EB, R7EC
R805	Undefined	—	—	—	—
R806 *3	ASR mode setting data undefined	0: Normal 1: Error	Parameter about ASR mode of Ethernet port is not fixed.	S	U, R7EB, R7EC
R807 *3	Modbus-TCP parameter undefined	0: Normal 1: Error	The parameter about a Modbus-TCP function is undefined.	S	U, R7EB, R7EC
R808 to R80F	Undefined	—	—	—	—
R810	Backup request bit	1: Backup request	When data memory backup function enables and this bit is turned on, data from WR7F00 to WR7FFF is written to FLASH memory.	U	S
R811	Backup result bit	0: Normal 1: Error	The result of data memory backup is shown.	S	S
R812 to R81F	Undefined	—	—	—	—
R820	HSDL Run / Stop	0: HSDL Stop 1: HSDL Run	Serial data link (HSDL) can be controlled by this bit (Only for master).	U	U
R821	HSDL link data update time (Max.) initialization	1: Initialization	HSDL link data update time (Max.) in WRF0ED will be update to zero (Only for master).	U	S
R822	HSDL link data update time (Min.) initialization	1: Initialization	HSDL link data update time (Min.) in WRF0EF will be update to 65,535 (Only for master).	U	S
R823 to R8FF	Undefined	—	—	—	—

Set / Reset Condition :

S…ON / OFF by system, U…ON / OFF by user, R7EB…Set 1 to R7EB, R7EC…Set 1 to R7EC,
X…Always display

*3 Even if there were multiple error factors, only one bit is turned on.

No	Name [Main use]	Meaning	Description	Setting condition	Resetting condition
R900	NTP time retrieval user program control valid/invalid	0: Program setting cycle 1: Control by R901 ➔For details, refer to User's Manual chapter 3, 3.5 setting a clock with NTP Communication.	Specifies whether to perform the time data retrieval from NTP server with the cycle set in programmer, or to control it by R901.	U	U
R901	NTP time retrieval request	1: Retrieval start ➔For details, refer to User's Manual chapter 3, 3.5 setting a clock with NTP Communication.	Retrieves the time data from NTP server.	U	S
R902	NTP time retrieval result	0: Retrieval success 1: Retrieval failure ➔For details, refer to User's Manual chapter 3, 3.5 setting a clock with NTP Communication.	Indicates failure of the time data retrieval from NTP server.	S	U
R903 to R90E	Undefined	–	–	–	–
R90F	Modbus-TCP server access target switching	0: Write to WM 1: Write to WR	Type of internal output which is wrote by Modbus protocol function code 06 can be changed.	U	U
R910	Ethernet port (Task code port 1) Reset request	1: Reset request	Reset task code port 1. Clear the task code send / receive counter (WRF1F0 to WRF1F3) to 0. (Setting data by Control Editor remains.)	U	S
R911	Ethernet port (Task code port 2) Reset request	1: Reset request	Reset task code port 2. Clear the task code send / receive counter (WRF1F4 to WRF1F7) to 0. (Setting data by Control Editor remains.)	U	S
R912	Ethernet port (Task code port 3) Reset request	1: Reset request	Reset task code port 3. Clear the task code send / receive counter (WRF1F8 to WRF1FB) to 0. (Setting data by Control Editor remains.)	U	S
R913	Ethernet port (Task code port 4) Reset request	1: Reset request	Reset task code port 4. Clear the task code send / receive counter (WRF1FC to WRF1FF) to 0. (Setting data by Control Editor remains.)	U	S
R914	Ethernet port (ASR port 1) Reset request	1: Reset request	Reset ASR port 1. (Setting data by Control Editor remains.)	U	S
R915	Ethernet port (ASR port 2) Reset request	1: Reset request	Reset ASR port 2. (Setting data by Control Editor remains.)	U	S
R916	Ethernet port (ASR port 3) Reset request	1: Reset request	Reset ASR port 3. (Setting data by Control Editor remains.)	U	S
R917	Ethernet port (ASR port 4) Reset request	1: Reset request	Reset ASR port 4. (Setting data by Control Editor remains.)	U	S
R918	Ethernet port (ASR port 5) Reset request	1: Reset request	Reset ASR port 5. (Setting data by Control Editor remains.)	U	S
R919	Ethernet port (ASR port 6) Reset request	1: Reset request	Reset ASR port 6. (Setting data by Control Editor remains.)	U	S
R91A	Ethernet port (Modbus-TCP) Reset request	1: Reset request	Reset Modbus-TCP port. (Setting data by Control Editor remains.)	U	S
R91B to R91F	Undefined	–	–	–	–
R920	Ethernet communication initialization completed	0: Initializing 1: Initialization complete	This bit turns ON when the power is turned on and initialization of the Ethernet port is completed.	S	S
R921	Task code send / receive counter clear	1: Counter clear request	Clear the task code send / receive counter (WRF1F0 to WRF1FF) to 0.	U	S

Set / Reset Condition :

S…ON / OFF by system, U…ON / OFF by user, R7EB…Set 1 to R7EB, R7EC…Set 1 to R7EC,
X…Always display

No	Name [Main use]	Meaning	Description	Setting condition	Resetting condition
R922 to R92F	Undefined	—	—	—	—
R930 to R9CF	Data logging status / control bit	For details, refer to User's Manual Chapter 7, 7.7 Special internal output for Logging and Trace function.		U, S	U, S
R9D0	Ethernet port (Task code port 1) Reset complete	1: Reset complete	Turns ON when reset of task code port 1 is completed.	S	U
R9D1	Ethernet port (Task code port 2) Reset complete	1: Reset complete	Turns ON when reset of task code port 2 is completed.	S	U
R9D2	Ethernet port (Task code port 3) Reset complete	1: Reset complete	Turns ON when reset of task code port 3 is completed.	S	U
R9D3	Ethernet port (Task code port 4) Reset complete	1: Reset complete	Turns ON when reset of task code port 4 is completed.	S	U
R9D4	Ethernet port (ASR port 1) Reset complete	1: Reset complete	Turns ON when ASR port 1 reset is complete.	S	U
R9D5	Ethernet port (ASR port 2) Reset complete	1: Reset complete	Turns ON when ASR port 2 reset is complete.	S	U
R9D6	Ethernet port (ASR port 3) Reset complete	1: Reset complete	Turns ON when ASR port 3 reset is complete.	S	U
R9D7	Ethernet port (ASR port 4) Reset complete	1: Reset complete	Turns ON when ASR port 4 reset is complete.	S	U
R9D8	Ethernet port (ASR port 5) Reset complete	1: Reset complete	Turns ON when ASR port 5 reset is complete.	S	U
R9D9	Ethernet port (ASR port 6) Reset complete	1: Reset complete	Turns ON when ASR port 6 reset is complete.	S	U
R9DA	Ethernet port (Modbus-TCP) Reset complete	1: Reset complete	Turns ON when reset of Modbus-TCP port is completed.	S	U
R9DB	Undefined	—	—	—	—
R9DC	Ethernet port (Task code port 1) Error occurred	0: Normal 1: Error	Turns ON when a communication error occurs on the Ethernet port (task code port 1).	S	U, R7EB, R7EC
R9DD	Ethernet port (Task code port 2) Error occurred	0: Normal 1: Error	Turns ON when a communication error occurs on the Ethernet port (task code port 2).	S	U, R7EB, R7EC
R9DE	Ethernet port (Task code port 3) Error occurred	0: Normal 1: Error	Turns ON when a communication error occurs on the Ethernet port (task code port 3).	S	U, R7EB, R7EC
R9DF	Ethernet port (Task code port 4) Error occurred	0: Normal 1: Error	Turns ON when a communication error occurs on the Ethernet port (task code port 4).	S	U, R7EB, R7EC
R9E0 to R9FF	Undefined	—	—	—	—

Set / Reset Condition :

S…ON / OFF by system, U…ON / OFF by user, R7EB…Set 1 to R7EB, R7EC…Set 1 to R7EC,
X…Always display

No	Name [Main use]	Meaning	Description	Setting condition	Resetting condition
RA00 to RA0A	System use area	—	(Area which is used by system.)	—	—
RA0B	Ch1 Homing in progress	0: Before homing or executing other operation. 1: Under Homing.	Indicates that ch1 is performing the homing operation.	S	S
RA0C	System use area	—	(Area which is used by system.)	—	—
RA0D	Ch1 Homing complete	0: Before homing or executing other operation. 1: Under Homing.	If homing operation for Ch1 completed it will be turned on, and after this, when pulse outputs this bit will be turned off.	S	S
RA0E to RA1A	System use area	—	(Area which is used by system.)	—	—
RA1B	Ch2 Homing in progress	0: Before homing or executing other operation. 1: Under Homing.	Indicates that ch2 is performing the homing operation.	S	S
RA1C	System use area	—	(Area which is used by system.)	—	—
RA1D	Ch2 Homing complete	0: Before homing or during pulse output. 1: Homing complete.	If homing operation for Ch2 completed it will be turned on, and after this, when pulse outputs this bit will be turned off.	S	S
RA1E to RA2A	System use area	—	(Area which is used by system.)	—	—
RA2B	Ch3 Homing in progress	0: Before homing or executing other operation. 1: Under Homing.	Indicates that ch3 is performing the homing operation.	S	S
RA2C	System use area	—	(Area which is used by system.)	—	—
RA2D	Ch3 Homing complete	0: Before homing or during pulse output. 1: Homing complete.	If homing operation for Ch3 completed it will be turned on, and after this, when pulse outputs this bit will be turned off.	S	S
RA2E to RA3F	System use area	—	(Area which is used by system.)	—	—
RA40 to RFFF	Undefined	—	—	—	—

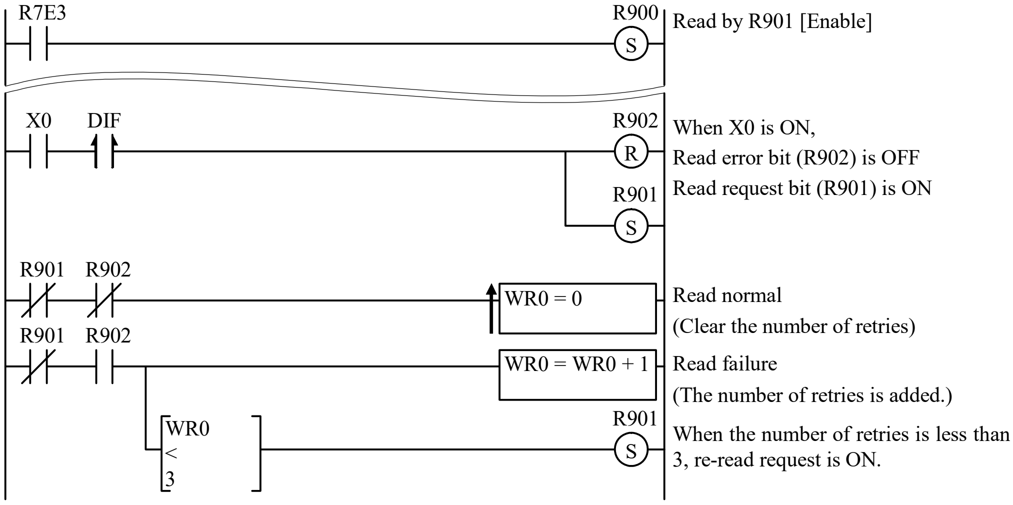
Set / Reset Condition：

S…ON / OFF by system, U…ON / OFF by user, R7EB…Set 1 to R7EB, R7EC…Set 1 to R7EC,
X…Always display

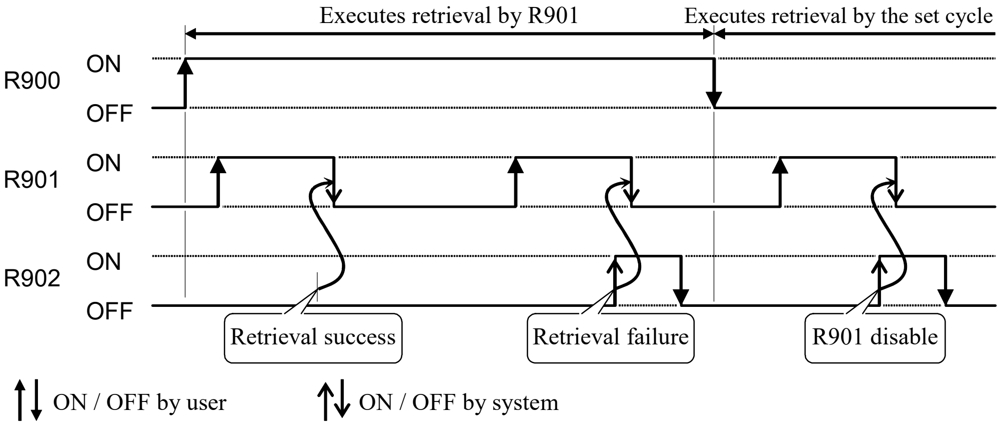
(1) Special internal output for NTP client function

The clock data can be acquired from NTP server automatically by setting NTP client function using Control Editor. There are two methods to acquisition the clock data, one is a method by a constant cycle and the other is a method by the user program. In the case of the former, the cycle is specified using Control Editor. In the case of the latter, the bit special internal output is used. (In both cases, it needs to validate the NTP client function beforehand using Control Editor.)

■ Clock data read using NTP client function Sample program



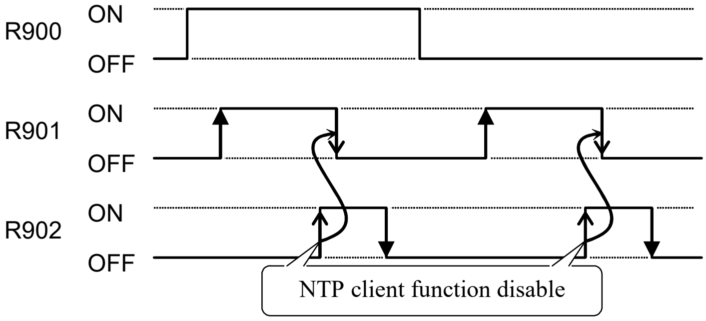
■ Control of special internal output



Note

In order to enable NTP client function, the power supply needs to be turned ON again after setting the parameter using Control Editor.

When NTP client function is disable, the clock data cannot be read from NTP server using the special internal output.

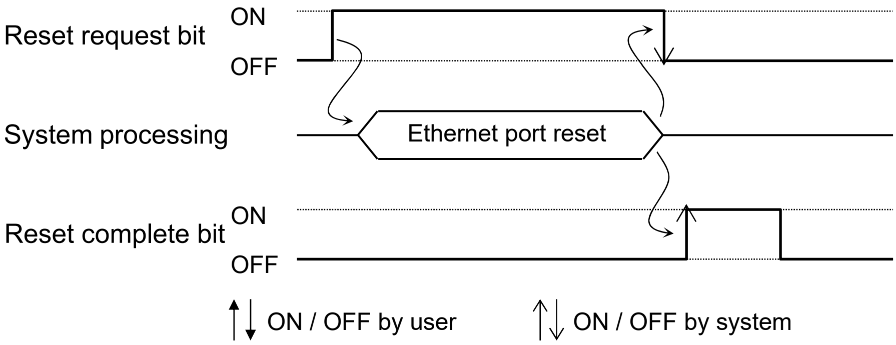


(2) Reset function for Ethernet port

If the Ethernet port (task code port, ASR port, Modbus-TCP port) becomes unable to communicate for some reason, the Ethernet port can be returned to the initial state using a programming tool. (In the case of TCP/IP, it returns to the state before connection establishment.)

The Ethernet port is reset by bit special internal output.

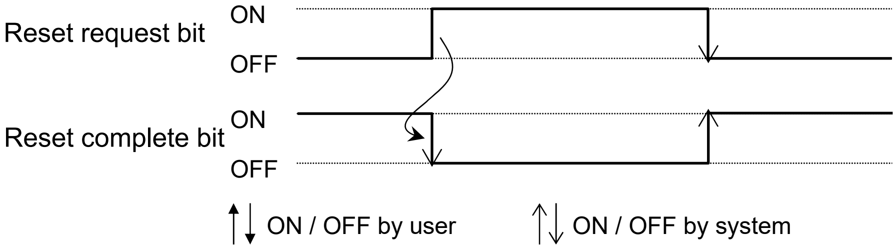
Ethernet port reset request bits (R910 to R91A) and reset completion bits (R9D0 to R9DA) are provided for bit special internal output. When the reset request bit of the corresponding Ethernet port is turned ON, the Ethernet port reset starts. When the reset of the Ethernet port is completed, the reset request bit is automatically turned OFF and the reset completion bit is turned ON.



Reference

The reset completion bit indicates the result, so even if the reset completion bit is ON, the Ethernet port can be reset by turning ON the reset request bit.

(When the reset request flag is turned ON, the reset completion flag is automatically turned OFF.)



A1.2 Word special internal output

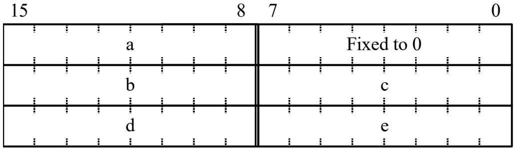
The word special internal output area is WRF000 to WRFFFF. Functions of each word are shown below.

No	Name [Main use]	Meaning	Description	Setting condition	Resetting condition
WRF000	Self-diagnosis error code	Error code (2 digit hexadecimal, upper 2 digits are 00)	Stores the error No. in CPU as a binary code.	S	U, R7EB, R7EC
WRF001	Undefined	—	—	—	—
WRF002 (R7CD)	I/O verify mismatch details	Mismatched slot number *	15 1211 87 43 0 0 a b 0 a: Unit number (0 to 5) b: Slot number (0 to F)	S	U, R7EB, R7EC
WRF003 to WRF00A	Undefined	—	—	—	—
WRF00B	Calendar and clock current value (4 digit BCD)	Year	Always displays the 4 digit year.	S	X
WRF00C		Month/day	Always displays month/day data.	S	X
WRF00D		Day of the week	Always displays day of the week data.	S	X
WRF00E		Hour/minute	(Sunday: 0000 to Saturday: 0006)	S	X
WRF00F		Seconds	Always displays hour/minute data.	S	X
WRF010	Scan time (maximum value)	Maximum execution time for a normal scan	Maximum execution time for a normal scan is stored in 1 ms units.	S	S
WRF011	Scan time (current value)	Present value of execution time for a normal scan	Present value of execution time for a normal scan is stored in 1 ms units.	S	S
WRF012	Scan time (minimum value)	Minimum execution time for a normal scan	Minimum execution time for a normal scan is stored in 1 ms units. (the first scan after RUN is HFFFF)	S	S
WRF013	CPU status	15 14 13 8 7 6 5 4 3 2 1 0 ▨ a b ▨ c ▨ d e b15,14 : Unused, b13 to b8 : (a)CPU type MVH-*64* : 100101 MVH-*40* : 100100 MVL-*64* : 100010 MVL-*40* : 100001 MVL-*20* : 100000 b7 : (b) Battery error 0 : No Error, 1 : Error b6 to b4 : Unused b3 : (c) Error 0 : No Error, 1 : Error b2 : Unused b1 : (d) HALT 0 : No Executing, 1 : Executing b0 : (e)CPU operation 0 : STOP, 1 : RUN		S	S
WRF014	Word internal output capacity	Number of words for word internal output (WR)	MICRO-EHV: H8000	S	X
WRF015	Operation error code	Operation error code	Operation error code is stored. (4 digit hexadecimal)	S	U
WRF016	Division remainder register (lower)	Remainder data when division instruction executed	For a word operation: WRF016 only	S	S
WRF017	Division remainder register (upper)		For a double word operation: WRF017 (upper), WRF016 (lower)	S	S
WRF018 to WRF01A	Undefined	—	—	—	—

Set / Reset Condition :

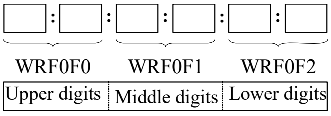
S…ON / OFF by system, U…ON / OFF by user, R7EB…Set 1 to R7EB, R7EC…Set 1 to R7EC,
X…Always display

* You must turn the applicable bit special internal output (inside ( ) of No) OFF once if you want to check which slot the error is occurring currently. And you must check it after turning R7EC ON once.

No.	Name [Main use]	Meaning	Description	Setting condition	Resetting condition
WRF01B	Reading (latch) or writing data area for the calendar clock (BCD 4 digits)	Year	Stores the 4-digit year read or sets the set value.	S, U	U
WRF01C		Month/day	Stores month/day data read or sets the set value.	S, U	U
WRF01D		Day of the week data (Sunday: 0000 to Saturday: 0006)	Stores day data of the week read or sets the set value.	S, U	U
WRF01E		Hour/minute (24-hour system)	Stores hour/minute data read or sets the set value.	S, U	U
WRF01F		Seconds	Stores second data read or sets the set value.	S, U	U
WRF020	NTP client time zone	Time zone setup	Time zone of NTP client function is specified. ➔ refer to appendix A1-16 * Current time zone setting value is set on the system only when the power supply is ON.	U	S
WRF021	Reading (latch) or writing data area for the calendar clock (BCD 4 digits)	Year	4 digits for year [yyyy]	S	S
WRF022		Month/day	Month and day [mmdd]	S	S
WRF023		Day of the week data (Sunday : 0000 to Saturday : 0006)	Sunday : 0000 to Saturday : 0006	S	S
WRF024		Hour/minute(24-hour system)	Hour and minute [hhmm] (24-hour)	S	S
WRF025		Seconds	Second [00ss] (Upper 2 digits are 00.)	S	S
WRF026 to WRF03B	Undefined	—	—	—	—
WRF03C	Option board analog output 1	Output value of analog channel 1 on option board	Stores the analog value output from option board analog channel 1.	U	S, U
WRF03D	Option board analog output 2	Output value of analog channel 2 on option board	Stores the analog value output from option board analog channel 2.	U	S, U
WRF03E	Option board analog input 1	Input value of analog channel 1 on option board	Analog input value of option board analog channel 1 is stored.	S	S
WRF03F	Option board analog input 2	Input value of analog channel 2 on option board	Analog input value of option board analog channel 2 is stored.	S	S
WRF040 to WRF042	Occupied member registration area 1	Occupied port number  a: 0=Not occupied, 1=Read-occupied, 2=Write-occupied b: Loop number c: Unit number d: Module number e: Port number (serial :H01 , Ethernet :H03 to H06, USB :H07)		S	S
WRF043 to WRF045	Occupied member registration area 2			S	S
WRF046 to WRF048	Occupied member registration area 3			S	S
WRF049 to WRF04B	Occupied member registration area 4			S	S
WRF04C to WRF04D	Undefined	—	—	—	—
WRF04E	Option board analog input 3	Input value of analog channel 3 on option board	Analog input value of option board analog channel 3 is stored.	S	S
WRF04F	Option board analog input 4	Input value of analog channel 4 on option board	Analog input value of option board analog channel 4 is stored.	S	S
WRF050	System software version [DISP]	System software version	The most significant digit in the hexadecimal display indicates the model. 0 ***: High function model (MVH) 1 ***: Standard model (MVL)	S	S
WRF051 to WRF053	System use area	—	(Area which is used by system.)	X	X
WRF054 WRF055	Total power-on time	Displays cumulative energization time in seconds	Accumulated energization time is stored in double word.(DRF054)	S	X

Set / Reset Condition :

S…ON / OFF by system, U…ON / OFF by user, R7EB…Set 1 to R7EB, R7EC…Set 1 to R7EC, X…Always display

No	Name [Main use]	Meaning	Description	Setting condition	Resetting condition
WRF056 to WRF05F	System use area	—	(Area which is used by system.)	X	X
WRF060 to WRF077	Ethernet port ASR communication control	ASR communication control and status display ➔ Refer to Appendix 1-16	WRF060 to WRF063 ASR port 1 WRF064 to WRF067 ASR port 2 WRF068 to WRF06B ASR port 3 WRF06C to WRF06F ASR port 4 WRF070 to WRF073 ASR port 5 WRF074 to WRF077 ASR port 6	Refer to Appendix 1-16	Refer to Appendix 1-16
WRF078 to WRF07F	System use area	—	(Area which is used by system.)	X	X
WRF080 to WRF0DF	Ethernet port ASR communication control	ASR communication mode, transmission size setting, reception size display ➔ Refer to Appendix 1-17	WRF080 to WRF08F ASR port 1 WRF090 to WRF09F ASR port 2 WRF0A0 to WRF0AF ASR port 3 WRF0B0 to WRF0BF ASR port 4 WRF0C0 to WRF0CF ASR port 5 WRF0D0 to WRF0DF ASR port 6	Refer to Appendix 1-17	Refer to Appendix 1-17
WRF0E0	Modbus-RTU waiting time (master)	Modbus-RTU waiting time (master)	Wait time can be set before query transmission in MBMST command, INV command, OMST command, OCTP command, Modbus-TCP gateway. Unit of time : ms	U	U
WRF0E1	Modbus-RTU waiting time (slave)	Modbus-RTU waiting time (slave)	The time from query reception to response transmission can be set in the Modbus slave function. Unit of time : ms	U	U
WRF0E2 to WRF0EC	Undefined	—	—	—	—
WRF0ED	HSDL link data update time (max)	Maximum update time for HSDL	Maximum update time for HSDL is stored in 1 ms units.	S	U
WRF0EE	HSDL link data update time (current)	Current update time for HSDL	Current update time for HSDL is stored in 1 ms units.	S	X
WRF0EF	HSDL link data update time (min)	Minimum update time for HSDL	Minimum update time for HSDL is stored in 1 ms units.	S	U
WRF0F0 to WRF0F2	MAC address	Displays Ethernet peculiar address of MICRO-EHV (MAC address)	 Example) Case of A4:97:BB:05:12:00 WRF0F0 … HA497 WRF0F1 … HBB05 WRF0F2 … H1200	S	X
WRF0F3	Ethernet port communication speed [DISP]	Displays communication speed of Ethernet port which is set to MICRO-EHV	0: Auto-negotiation 1: 100 Mbps / Full duplex 2: 100 Mbps / Half duplex 3: 10 Mbps / Full duplex 4: 10 Mbps / Half duplex	S	X
WRF0F4	USB memory size (L)	The capacity of USB memory attached on MICRO-EHV	Displays the capacity of the USB memory in kilobytes.	S	X
WRF0F5	USB memory size (H)				
WRF0F6	USB free space (L)	Free space of USB memory attached on MICRO-EHV	Displays the free space of the USB memory in kilobytes.	S	X
WRF0F7	USB free space (H)				
WRF0F8 to WRF0FF	Undefined	—	—	—	—

Set / Reset Condition :

S…ON / OFF by system, U…ON / OFF by user, R7EB…Set 1 to R7EB, R7EC…Set 1 to R7EC, X…Always display

No	Name [Main use]	Meaning	Description	Setting condition	Resetting condition
WRF100 WRF101	Counter 1 Current value	Current value of counter 1	The current value of counter 1 is stored as a double word. (DRF100)	S	U
WRF102 WRF103	Counter 2 Current value	Current value of counter 2	The current value of counter 2 is stored as a double word. (DRF102)	S	U
WRF104 WRF105	Counter 3 Current value	Current value of counter 3	The current value of counter 3 is stored as a double word. (DRF104)	S	U
WRF106 WRF107	Counter 4 Current value	Current value of counter 4	The current value of counter 4 is stored as a double word. (DRF106)	S	U
WRF108 WRF109	Counter 5 Current value	Current value of counter 5	The current value of counter 5 is stored as a double word. (DRF108)	S	U
WRF10A WRF10B	Pulse 1 Position data	Pulse output 1 position data	The position data of pulse output 1 is stored as a double word. (DRF10A)	S	U
WRF10C WRF10D	Pulse 2 Position data	Pulse output 2 position data	The position data of pulse output 2 is stored as a double word. （DRF10C）	S	U
WRF10E WRF10F	Pulse 3 Position data	Pulse output 3 position data	The position data of pulse output 3 is stored as a double word. （DRF10E）	S	U
WRF110	Counter 1 system use area	—	(Area which is used by system.)	X	X
WRF111	Undefined	—	—	—	—
WRF112 WRF113	Counter 1 ON-preset	Counter 1 ON-preset	Counter 1 ON-preset value is stored in double word. (DRF112)	S	S
WRF114 WRF115	Counter 1 OFF-preset	Counter 1 OFF-preset	Counter 1-OFF preset value is stored in double word. (DRF114)	S	S
WRF116 to WRF119	System use area	—	(Area which is used by system.)	X	X
WRF11A WRF11B	Counter 2 ON-preset	Counter 2 ON-preset	Counter 2 ON-preset value is stored in double word. (DRF11A)	S	S
WRF11C WRF11D	Counter 2 OFF-preset	Counter 2 OFF-preset	Counter 2 OFF-preset value is stored in double word. (DRF11C)	S	S
WRF11E to WRF121	System use area	—	(Area which is used by system.)	X	X
WRF122 WRF123	Counter 3 ON-preset	Counter 3 ON-preset	Counter 3 ON-preset value is stored in double word. (DRF122)	S	S
WRF124 WRF125	Counter 3 OFF-preset	Counter 3 OFF-preset	Counter 3 OFF-preset value is stored in double word. (DRF124)	S	S
WRF126 to WRF129	System use area	—	(Area which is used by system.)	X	X
WRF12A WRF12B	Counter 4 ON-preset	Counter 4 ON-preset	Counter 4 ON-preset value is stored in double word. (DRF12A)	S	S
WRF12C WRF12D	Counter 4 OFF-preset	Counter 4 OFF-preset	Counter 4 OFF-preset value is stored in double word. (DRF12C)	S	S
WRF12E WRF131	System use area	—	(Area which is used by system.)	X	X
WRF132 WRF133	Counter 5 ON-preset	Counter 5 ON-preset	Counter 5 ON-preset value is stored in double word. (DRF132)	S	S
WRF134 WRF135	Counter 5 OFF-preset	Counter 5 OFF-preset	Counter 5 OFF-preset value is stored in double word. (DRF134)	S	S
WRF136 to WRF137	System use area	—	(Area which is used by system.)	X	X

Set / Reset Condition：

S…ON / OFF by system, U…ON / OFF by user, R7EB…Set 1 to R7EB, R7EC…Set 1 to R7EC,
X…Always display

No	Name [Main use]	Meaning	Description	Setting condition	Resetting condition
WRF138 WRF139	Pulse/PWM 1 Output frequency	Pulse/PWM 1 Output frequency	Pulse / PWM 1 output frequency is stored in double word. (DRF138)	S	U
WRF13A WRF13B	Pulse 1 Number of pulses	Pulse 1 Number of pulses	The number of output pulses of pulse 1 is stored as a double word. (DRF13A)	S	U
	PWM 1 ON-Duty	PWM 1 ON-Duty	PWM 1 ON-Duty is stored in double word. (DRF13A)		
WRF13C to WRF13F	System use area	—	(Area which is used by system.)	X	X
WRF140 WRF141	Pulse/PWM 2 Output frequency	Pulse/PWM 1 Output frequency	Pulse / PWM 2 output frequency is stored in double word. (DRF140)	S	U
WRF142 WRF143	Pulse 2 Number of pulses	Pulse 2 Number of pulses	The number of output pulses of pulse 2 is stored as a double word. (DRF142)	S	U
	PWM 2 ON-Duty	PWM 2 ON-Duty	PWM 2 ON-Duty is stored in double word. (DRF142)		
WRF144 to WRF147	System use area	—	(Area which is used by system.)	X	X
WRF148 WRF149	Pulse/PWM 3 Output frequency	Pulse/PWM 3 Output frequency	Pulse / PWM 3 output frequency is stored in double word. (DRF148)	S	U
WRF14A WRF14B	Pulse 3 Number of pulses	Pulse 3 Number of pulses	The number of output pulses of pulse 3 is stored as a double word. (DRF14A)	S	U
	PWM 3 ON-Duty	PWM 3 ON-Duty	PWM 3 ON-Duty is stored in double word. (DRF14A)		
WRF14C to WRF14F	System use area	—	(Area which is used by system.)	X	X
WRF150 to WRF15F	HSDL status area	Status display for HSDL master and slave	As to the details, refer to “3.6 Serial data link communication”. WRF150 … St. No.1, Master WRF151 … St. No.3, St. No.2 WRF152 … St. No.5, St. No.4 WRF153 … St. No.7, St. No.6 WRF154 … St. No.9, St. No.8 WRF155 … St. No.11, St. No.10 WRF156 … St. No.13, St. No.12 WRF157 … St. No.15, St. No.14 WRF158 … St. No.17, St. No.16 WRF159 … St. No.19, St. No.18 WRF15A … St. No.21, St. No.20 WRF15B … St. No.23, St. No.22 WRF15C … St. No.25, St. No.24 WRF15D … St. No.27, St. No.26 WRF15E … St. No.29, St. No.28 WRF15F … St. No.31, St. No.30	S	X
WRF160 to WRF1DF	System use area	—	(Area which is used by system.)	X	X
WRF1E0 WRF1EF	Undefined	—	—	—	—

Set / Reset Condition :

S…ON / OFF by system, U…ON / OFF by user, R7EB…Set 1 to R7EB, R7EC…Set 1 to R7EC,
X…Always display

No	Name [Main use]	Meaning	Description	Setting condition	Resetting condition
WRF1F0	Ethernet port (task code port 1) sending counter	Task code port 1 response sending counter	In Hitachi protocol communication, the number of times is added each time a response is sent to a request from a host.	S	U
WRF1F1	Ethernet port (task code port 1) receiving counter 1	Task code port 1 request Receiving counter	In Hitachi protocol communication, the number of times is added when a packet that exceeds 7 bytes and the start code is the specified value is received.	S	U
WRF1F2	Ethernet port (task code port 1) receiving counter 2	Task code port 1 error request (error command) reception count	In Hitachi protocol communication, if a packet of less than 7 bytes is received or a packet of 7 bytes or more whose leading code is not the specified value is received, the number of times is added.	S	U
WRF1F3	Ethernet port (task code port 1) receiving counter 3	Task code port 1 error request (error format) reception count	In Hitachi protocol communication, when a request task code that returns an abnormal response is received or a request that returns a non-executable response is received, the number is added.	S	U
WRF1F4	Ethernet port (task code port 2) sending counter	Task code port 2 response transmission count	In Hitachi protocol communication, the number of times is added each time a response is sent to a request from a host.	S	U
WRF1F5	Ethernet port (task code port 2) receiving counter 1	Task code port 2 request reception count	In Hitachi protocol communication, the number of times is added when a packet that exceeds 7 bytes and the start code is the specified value is received.	S	U
WRF1F6	Ethernet port (task code port 2) receiving counter 2	Task code port 2 error request (error command) reception count	In Hitachi protocol communication, if a packet of less than 7 bytes is received or a packet of 7 bytes or more whose leading code is not the specified value is received, the number of times is added.	S	U
WRF1F7	Ethernet port (task code port 2) receiving counter 3	Task code port 2 error request (error format) reception count	In Hitachi protocol communication, when a request task code that returns an abnormal response is received or a request that returns a non-executable response is received, the number is added.	S	U
WRF1F8	Ethernet port (task code port 3) sending counter	Task code port 3 response transmission count	In Hitachi protocol communication, the number of times is added each time a response is sent to a request from a host.	S	U
WRF1F9	Ethernet port (task code port 3) receiving counter 1	Task code port 3 request reception count	In Hitachi protocol communication, the number of times is added when a packet that exceeds 7 bytes and the start code is the specified value is received.	S	U
WRF1FA	Ethernet port (task code port 3) receiving counter 2	Task code port 3 error request (error command) reception count	In Hitachi protocol communication, if a packet of less than 7 bytes is received or a packet of 7 bytes or more whose leading code is not the specified value is received, the number of times is added.	S	U
WRF1FB	Ethernet port (task code port 3) receiving counter 3	Task code port 3 error request (error format) reception count	In Hitachi protocol communication, when a request task code that returns an abnormal response is received or a request that returns a non-executable response is received, the number is added.	S	U

Set / Reset Condition：

S…ON / OFF by system, U…ON / OFF by user, R7EB…Set 1 to R7EB, R7EC…Set 1 to R7EC,
X…Always display

No	Name [Main use]	Meaning	Description	Setting condition	Resetting condition
WRF1FC	Ethernet port (task code port 4) transmission counter	Task code port 4 response transmission count.	In Hitachi protocol communication, the number of times is added each time a response is sent to a request from a host.	S	U
WRF1FD	Ethernet port (task code port 4) reception counter 1	Task code port 4 request reception count.	In Hitachi protocol communication, the number of times is added when a packet that exceeds 7 bytes and the start code is the specified value is received.	S	U
WRF1FE	Ethernet port (task code port 4) reception counter 2	Task code port 4 error request (error command) reception count.	In Hitachi protocol communication, if a packet of less than 7 bytes is received or a packet of 7 bytes or more whose leading code is not the specified value is received, the number of times is added.	S	U
WRF1FF	Ethernet port (task code port 4) reception counter 3	Task code port 4 error request (error format) reception count.	In Hitachi protocol communication, when a request task code that returns an abnormal response is received or a request that returns a non-executable response is received, the number is added.	S	U

Set / Reset Condition：

S…ON / OFF by system, U…ON / OFF by user, R7EB…Set 1 to R7EB, R7EC…Set 1 to R7EC,
X…Always display

(1) NTP client function time zone

The relation between the value set to WRF020 and time zone is shown below.

Set value	Time zone
H0000	GMT – 12:00
H0001	GMT – 11:00
H0002	GMT – 10:00
H0003	GMT – 9:00
H0004	GMT – 8:00
H0005	GMT – 7:00
H0006	GMT – 6:00
H0007	GMT – 5:00
H0008	GMT – 4:00
H0009	GMT – 3:30
H000A	GMT – 3:00
H000B	GMT – 2:00
H000C	GMT – 1:00

Set value	Time zone
H000D	GMT
H000E	GMT + 1:00
H000F	GMT + 2:00
H0010	GMT + 3:00
H0011	GMT + 3:30
H0012	GMT + 4:00
H0013	GMT + 4:30
H0014	GMT + 5:00
H0015	GMT + 5:30
H0016	GMT + 5:45
H0017	GMT + 6:00
H0018	GMT + 6:30
H0019	GMT + 7:00

Set value	Time zone
H001A	GMT + 8:00
H001B	GMT + 9:00
H001C	GMT + 9:30
H001D	GMT + 10:00
H001E	GMT + 11:00
H001F	GMT + 12:00
H0020	GMT + 13:00
Out of range	GMT

Note

Setting the parameter to WRF020. Makes the time zone valid. However, even if a time zone is changed while time data is read from the NTP server, the time zone is not reflected.

(2) Ethernet communication port ASR function

■Status register, Control register and Sending/Receiving counter

ASR communication registers are assigned to the special internal outputs WRF060 to WRF077 (24 words). As shown in Figure A1.1, it consists of 4 words per port and 24 words for 6 ports. Refer to the status register information and control the ASR function with the control register.

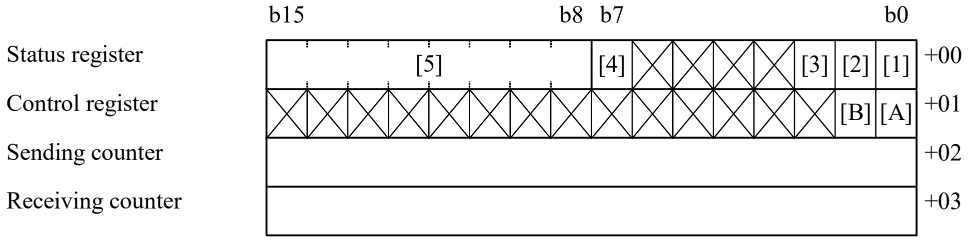


Figure A1.1 Status register and Control register

[Status register]

[1] Port status flag	1: Open
[2] Even transmitted completed flag	1: Transmitting completed
[3] Receiving completed flag	1: Receiving completed
[4] Error flag	1: Error occurred

[5] Error code 0x01: Despite not opening the port, the event transmitted request flag was turned ON.
0x02: The event transmitted request flag was re-turned ON while the transmitting has not completed.

[Control register]

[A] Port opened request flag	1: Open request	0: Close request
[B] Event transmitted request flag	1: Start transmitting	

■ Sending mode, Sending data size and Receiving data size

-The sending data size is dynamically changed of bytes in the ladder program.

-The receiving data size is referred to in the ladder program.

-Special internal outputs about this function are assigned after WRF080.

Special internal outputs	Meanings
WRF080 to WRF08F	Mode, Sending data size and Receiving data size about ASR port 1.
WRF090 to WRF09F	Mode, Sending data size and Receiving data size about ASR port 2.
WRF0A0 to WRF0AF	Mode, Sending data size and Receiving data size about ASR port 3.
WRF0B0 to WRF0BF	Mode, Sending data size and Receiving data size about ASR port 4.
WRF0C0 to WRF0CF	Mode, Sending data size and Receiving data size about ASR port 5.
WRF0D0 to WRF0DF	Mode, Sending data size and Receiving data size about ASR port 6.

No.	Name	Meanings
WRF0x0	Setup of mode 0: Control Editor setting operation mode 1: Special Internal Output setting operation mode	Control Editor setting operation mode transmits data (word unit) for the number of I/O points set from Control Editor. Special Internal Output setting operation mode transmits data (byte unit) of the size set for special internal output.
WRF0*1	Setup of sending data size for send area 1	Set the send data size in bytes.
WRF0*2	Setup of sending data size for send area 2	Set the send data size in bytes.
WRF0*3	Setup of sending data size for send area 3	Set the send data size in bytes.
WRF0*4	Setup of sending data size for send area 4	Set the send data size in bytes.
WRF0*5	Setup of sending data size for send area 5	Set the send data size in bytes.
WRF0*6	Setup of sending data size for send area 6	Set the send data size in bytes.
WRF0*7	Setup of sending data size for send area 7	Set the send data size in bytes.
WRF0*8	Setup of sending data size for send area 8	Set the send data size in bytes.
WRF0*9	Setup of sending data size for send area 9	Set the send data size in bytes.
WRF0*A	Setup of sending data size for send area 10	Set the send data size in bytes.
WRF0*B	Display of receiving data size	Receive data size is stored in bytes.
WRF0*C to WRF0*F	Reserve	

* It means 8 to D corresponding to ASR ports 1 to 6.

MEMO

Appendix 2 I/O Address

A2.1 External I/O

The external input is represented by X and the external output is represented by Y.

List of external I/O classification and data type

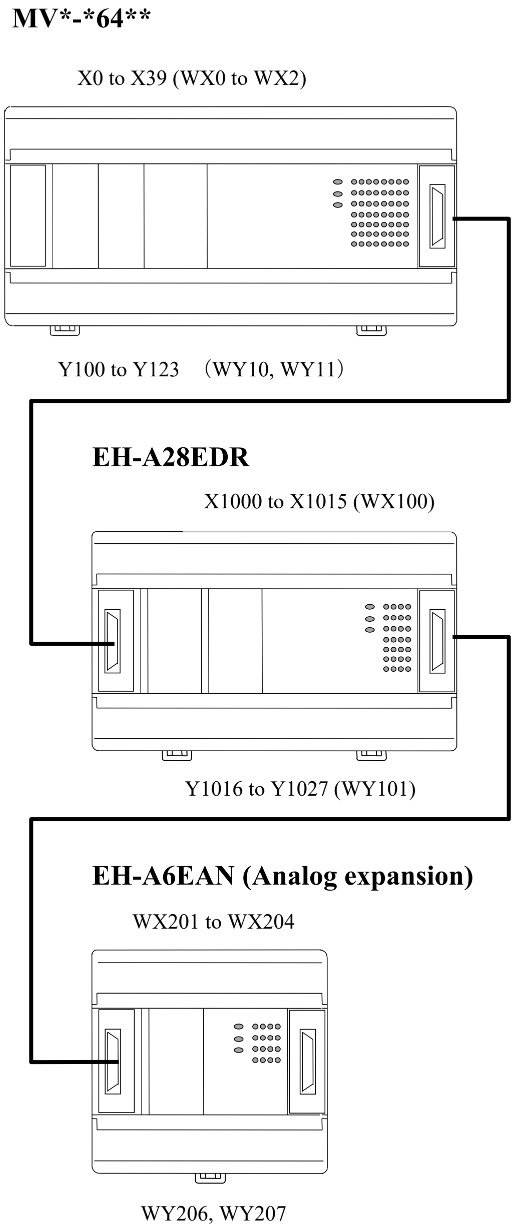
I/O symbols	Input or output	Data types	Remarks
X	External input	Bit (Bool) (1 bit)	Decimal number (X0, 1, 2, .., 9, 10, .., 15, 16, 17, .., 39)
WX		Word (16 bits)	Data in 0 to 15 are batch-processed. 16-point synchronicity is guaranteed.
DX		Double word (32 bits)	Address is hexadecimal number. Simultaneity of 32 bits data is not ensured.
Y	External output	Bit (Bool) (1 bit)	Decimal number (Y100, 101, 102, .., 109, 110, .., 115, 116, 117, .., 123)
WY		Word (16 bits)	Data in 0 to 15 are batch-processed. 16-point synchronicity is guaranteed.
DY		Double word (32 bits)	Two word data are batch-expressed. 32-point synchronicity is not guaranteed

The I/O configuration and I/O address of each unit are shown below.

I/O configuration and I/O address of each unit

Type			I/O configuration	Input/ output	20-point	40-point	64-point
Basic	Digital		(Fixed)	Input	X0 to 11 (WX0)	X0 to 23 (WX0 to 1)	X0 to 39 (WX0 to 2)
				Output	Y100 to 107 (WY10)	Y100 to 115 (WY10)	Y100 to 123 (WY10 to 11)
Expansion 1	Digital	8/14/16/28-point	B1/1	Input	X1000 to 1015 (WX100)		
				Output	Y1016 to 1027 (WY101)		
		64-point expansion	X48/Y32	Input	X1000 to 1039 (WX100 to 102)		
				Output	Y1100 to 1123 (WY110 to 111)		
	Analog		FUN0	Input	WX100 to 104		
				Output	WY105 to 107		
Expansion 2	Digital	8/14/16/28-point	B1/1	Input	X2000 to 2015 (WX200)		
				Output	Y2016 to 2027 (WY201)		
		64-point expansion	X48/Y32	Input	X2000 to 2039 (WX200 to 202)		
				Output	Y2100 to 2123 (WY210 to 211)		
	Analog		FUN0	Input	WX200 to 204		
				Output	WY205 to 207		
Expansion 3	Digital	8/14/16/28-point	B1/1	Input	X3000 to 3015 (WX300)		
				Output	Y3016 to 3027 (WY301)		
		64-point expansion	X48/Y32	Input	X3000 to 3039 (WX300 to 302)		
				Output	Y3100 to 3123 (WY310 to 311)		
	Analog		FUN0	Input	WX300 to 304		
				Output	WY305 to 307		
Expansion 4	Digital	8/14/16/28-point	B1/1	Input	X4000 to 4015 (WX400)		
				Output	Y4016 to 4027 (WY401)		
		64-point expansion	X48/Y32	Input	X4000 to 4039 (WX400 to 402)		
				Output	Y4100 to 4123 (WY410 to 411)		
	Analog		FUN0	Input	WX400 to 404		
				Output	WY405 to 407		

I/O number example



A2.2 Internal output

The internal output is a register that the user program can use.

The EHV series internal output has a bit dedicated area (R), a word dedicated area (WR, WN) and a bit / word shared area (M / WM) in the internal output. And there is a link area (L / WL) for exchanging data with other CPU using the link module. However, the MICRO-EHV series does not have a dedicated word area (WN) or link area (L / WL).

The internal output includes an area that can be freely accessed by the user and an area that is used for a specific purpose. The area that is used for a specific purpose is called "special internal output". The special internal output is used for setting the system and indicating the status. Refer to “Appendix 1 List of Special Internal Output” for the details.

List of internal outputs

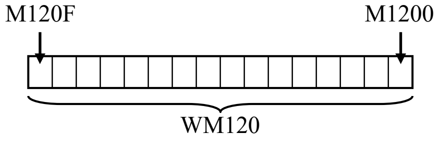
CPU model I/O type		Number of points	
		MVL-*20/40/64** (Standard model)	MVH-*40/64** (High Function model)
Bit		1,984 bits (R0 to R7BF)	
Word (WR)		32,768 words (WR0 to WR7FFF)	
Word (WN)		-	
Bit/word shared (WM)		32,768 bits, 2,048 words (M0 to M7FFF, WM0 to WM7FF)	
Special internal output	Bit	2,112 bits (R7C0 to RFFF)	
	Word	4,096 words (WRF000 to WRF1FF)	
CPU link		-	

Internal output I/O numbers are represented based on the following rules.

List of internal output I/O numbering rules (1/2)

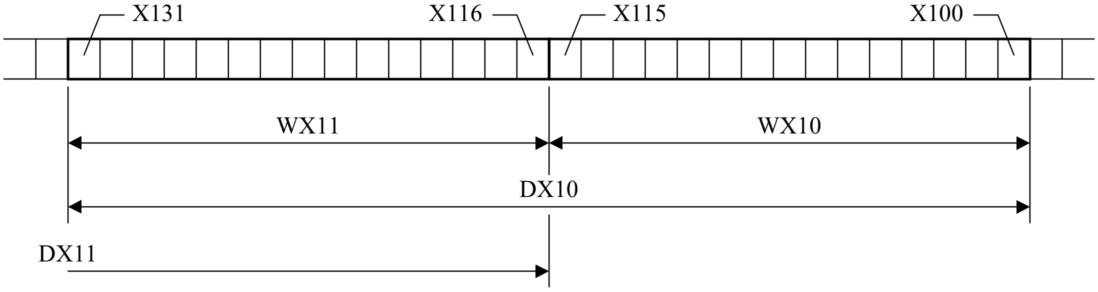
Data type	Numbering rule	
Bit-dedicated type		R □□□ Normal area H000 to H7BF Special area H7C0 to H7FF Both hexadecimal
Word-dedicated type	<For word>	W R □□□□ Normal area H0000 or above Special area HF000 or above Both hexadecimal
	[Bit specification]	W R □□□□.□ Specify ".n". (n: Bit No., 0 to F)
	[Signed integer]	W R □□□□. S Specify ".S".
	[String specification]	W R □□□□. A S C . n Specify ".n". (n: Number of bytes, 1 to 32 [decimal]) Specify ".ASC".
	<For double word>	D R □□□□ Normal area H0000 or above Special area HF000 or above Sequential 2-word WR representation Both hexadecimal
	[Signed integer]	D R □□□□. S Specify ".S".
	[Real number (floating point)]	D R □□□□. FL Specify ".FL".

List of internal output I/O numbering rules (2/2)

Data type	Numbering rule
Bit/word shared type	<For bit> M ☐☐☐☐☐ H00000 or above / H0000 or above Hexadecimal
	<For word> W M ☐☐☐☐ H0000 or above Hexadecimal  * No bit specification is available for the bit/word shared type.
	[Signed integer] W M ☐☐☐☐. S Specify ".S".
	[String specification] W M ☐☐☐☐. A S C . n Specify ".n". (n: Number of bytes, 1 to 32 [decimal]) Specify ".ASC".
	<For double word> D M ☐☐☐☐ H0000～ Both hexadecimal Sequential 2-word WR representation
	[Signed integer] D M ☐☐☐☐. S Specify ".S".
	[Real number (floating point)] D M ☐☐☐☐. FL Specify ".FL".

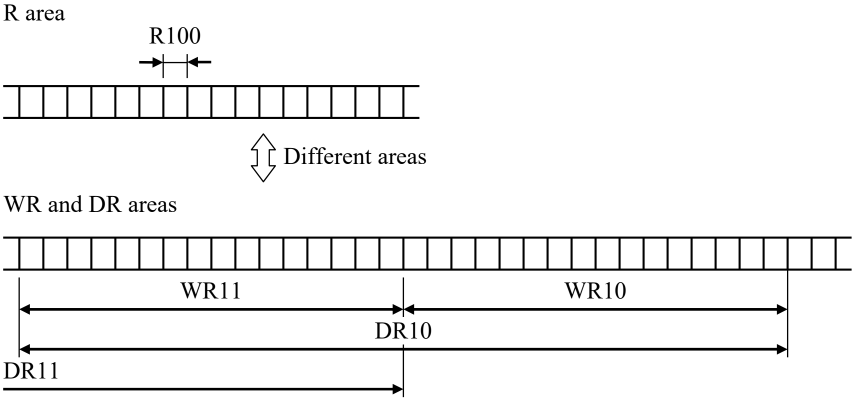
The word type data for external input / output is data that puts together 16 points bit data, and the double word type data is data that puts together 32 points.

Example: Relationship between DX10, WX10 and X100-X115



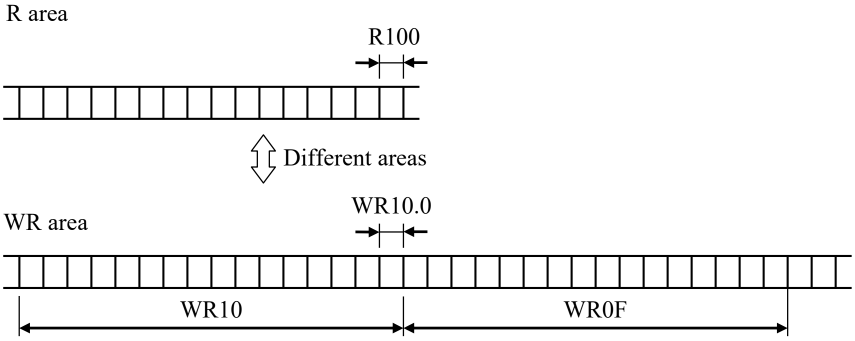
The internal output R area is different from the WR and DR areas.

Example: Correspondence between R100 and WR10/DR10



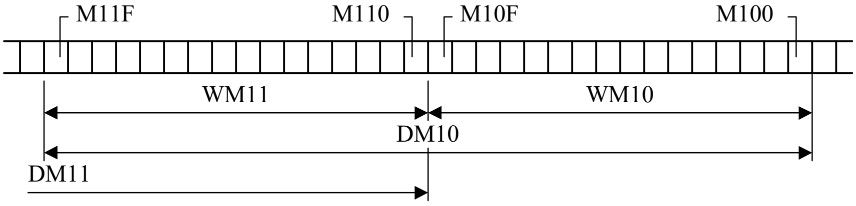
Caution

MICRO-EHV can select a specific bit from word data to access.



The internal outputs M, WM, and DM use the same area. (Each bit can be manipulated via word I/O.)

Example: Correspondence between M100 and WM10/DM10



Appendix 3 List of Supported Functions

MICRO-EHV has different functions depending on the software version. The software version can be monitored with the special internal output WRF050 (hexadecimal display).

Function			Software version		
		High Function (MVH)	0102 or older	0104	0105
		Standard (MVL)	1102 or older	1104	1105
Command	Basic command		✔	✔	✔
	Arithmetic command (all except the following)		✔	✔	✔
	Arithmetic command (SGET, EXT, SQR, BSQR, POW, EXP, FLOG, FLOG10 Radian trigonometric function)		-	✔	✔
	Application command (all except the following)		✔	✔	✔
	Application command (PID)		-	✔	✔
	Application command (High speed counter)		✔	✔	✔
	Application command (Pulse・PWM output) *1		-	✔	✔
	Application command (BMOV / BCOPY)		-	-	-
	Control command		✔	✔	✔
	Transfer command (TRNS0 / RECV 0)		✔	✔	✔
	Transfer command (MBMST / INV1)		-	✔	✔
	Transfer command (OMST1 / OCTP1)		-	-	-
I/O	Expansion	Expansion unit (Digital)	✔	✔	✔
		Expansion unit (Analog)	-	✔	✔
		Expansion unit (RTD, Thermocouple)	-	✔	✔
		Expansion unit (Positioning)	-	-	-
	Special internal output	NTP client function *2	-	✔	✔
		Task code port reset *2	-	-	✔
		Option board analog	-	-	✔
		communication speed setting *2	-	-	✔
	Special IO	High-speed counter input	✔	✔	✔
		Interrupt input	✔	✔	✔
		Pulse / PWM output *1	-	✔	✔
Communication function	Serial communication port (RS-232C)	Programming / Display	✔	✔	✔
		General-purpose communication (TRNS0 / RECEIVE 0)	✔	✔	✔
	Ethernet port *2	Programming / Display (TCP / IP)	✔	✔	✔
		Display etc. (UDP / IP)	-	-	-
		Modbus-TCP server	✔	✔	✔
		ASR communication function (TCP / IP)	-	-	✔
		ASR communication function (UDP / IP)	-	-	-
		Variable communication speed	-	-	✔
	USB port	Programming	✔	✔	✔
USB memory (host) *2	Uploading program (USB memory ← PLC)		✔	✔	✔
	Downloading program(USB memory → PLC)		✔	✔	✔
	Data logging		-	-	-
	Multiple programs of USB upload (in the file name MAC added)		-	-	-
Other	Data memory backup function		-	-	-
Option board	Communication	Programming / display (serial)	-	✔	✔
		Modbus-RTU master	-	✔	✔
		Modbus-RTU slave	-	✔	✔
		General-purpose communication (TRNS0 / RECEIVE 0)	-	✔	✔
		Modbus gateway *2	-	✔	✔
	Analog input		-	-	✔

✔ : Supported　　- : Not supported

* 1 This function is supported only for the transistor output type.

* 2 This function is supported only by the high function version (MVH).

Function			Software version			
		High Function (MVH)	0106	0107	0108	0109
		Standard (MVL)	1106	1107	1108	1109
Command	Basic command		✔	✔	✔	✔
	Arithmetic command (all except the following)		✔	✔	✔	✔
	Arithmetic command (SGET, EXT, SQR, BSQR, POW, EXP, FLOG, FLOG10 Radian trigonometric function)		✔	✔	✔	✔
	Application command (all except the following)		✔	✔	✔	✔
	Application command (PID)		✔	✔	✔	✔
	Application command (High speed counter)		✔	✔	✔	✔
	Application command (Pulse・PWM output) *1		✔	✔	✔	✔
	Application command (BMOV / BCOPY)		✔	✔	✔	✔
	Control command		✔	✔	✔	✔
	Transfer command (TRNS0 / RECV 0)		✔	✔	✔	✔
	Transfer command (MBMST / INV1)		✔	✔	✔	✔
	Transfer command (OMST1 / OCTP1)		✔	✔	✔	✔
I/O	Expansion	Expansion unit (Digital)	✔	✔	✔	✔
		Expansion unit (Analog)	✔	✔	✔	✔
		Expansion unit (RTD, Thermocouple)	✔	✔	✔	✔
		Expansion unit (Positioning)	-	-	-	-
	Special internal output	NTP client function *2	✔	✔	✔	✔
		Task code port reset *2	✔	✔	✔	✔
		Option board analog	✔	✔	✔	✔
		communication speed setting *2	✔	✔	✔	✔
	Special IO	High-speed counter input	✔	✔	✔	✔
		Interrupt input	✔	✔	✔	✔
		Pulse / PWM output *1	✔	✔	✔	✔
		Programming / Display	-	-	-	✔
Communication function	Serial communication port （RS-232C）	General-purpose communication (TRNS0 / RECV 0)	✔	✔	✔	✔
		Programming / Display (TCP / IP)	✔	✔	✔	✔
	Ethernet port *2	Display etc. (UDP / IP)	✔	✔	✔	✔
		Modbus-TCP server	-	-	✔	✔
		ASR communication function (TCP / IP)	✔	✔	✔	✔
		ASR communication function (UDP / IP)	✔	✔	✔	✔
		Variable communication speed	-	✔	✔	✔
		Programming	✔	✔	✔	✔
	USB port	Expansion unit (Digital)	✔	✔	✔	✔
USB memory (host) *2	Uploading program (USB memory ← PLC)		✔	✔	✔	✔
	Downloading program (USB memory → PLC)		✔	✔	✔	✔
	Data logging		-	-	✔	✔
	Multiple programs of USB upload (in the file name MAC added)		-	-	-	✔
	Data memory backup function		-	-	-	✔
Other	Uploading program (USB memory ← PLC)		-	-	✔	✔
Option board	Communication	Programming / display (serial)	✔	✔	✔	✔
		Modbus-RTU master	✔	✔	✔	✔
		Modbus-RTU slave	✔	✔	✔	✔
		General-purpose communication (TRNS0 / RECV 0)	✔	✔	✔	✔
		Modbus gateway *2	✔	✔	✔	✔
	Analog input		✔	✔	✔	✔
	Analog output		-	-	-	✔

✔ : Supported - : Not supported

*1 This function is supported only for the transistor output type.

*2 This function is supported only by the high function version (MVH).

*3 The counter ON / OFF preset value, pulse / PWM output frequency, pulse output pulse count, and PWM ON duty are displayed on the special internal output.

Function			Software version	
		High Function (MVH)	0109 or older	0110
		Standard (MVL)	1109 or older	1110
Special internal Output (Bit)	Reset Ethernet Modbus-TCP	[R91A]	-	✔
	Clear Task code transmit counter	[R921]	-	✔
	Task code P1 resetting completed	[R9D0]	-	✔
	Task code P2 resetting completed	[R9D1]	-	✔
	Task code P3 resetting completed	[R9D2]	-	✔
	Task code P4 resetting completed	[R9D3]	-	✔
	ASR port1 resetting completed	[R9D4]	-	✔
	ASR port2 resetting completed	[R9D5]	-	✔
	ASR port3 resetting completed	[R9D6]	-	✔
	ASR port4 resetting completed	[R9D7]	-	✔
	ASR port5 resetting completed	[R9D8]	-	✔
	ASR port6 resetting completed	[R9D9]	-	✔
	Ethernet port (Modbus-TCP port) Reset completed	[R9DA]	-	✔
	Task code port1 error	[R9DC]	-	✔
	Task code port2 error	[R9DD]	-	✔
	Task code port3 error	[R9DE]	-	✔
	Task code port4 error	[R9DF]	-	✔
Special internal Output(Word)	Task code P1 send count	[WRF1F0]	-	✔
	Task code P1 correctly receive count	[WRF1F1]	-	✔
	Task code P1 receive error (command)	[WRF1F2]	-	✔
	Task code P1 receive error (format)	[WRF1F3]	-	✔
	Task code P2 send count	[WRF1F4]	-	✔
	Task code P2 correctly receive count	[WRF1F5]	-	✔
	Task code P2 receive error (command)	[WRF1F6]	-	✔
	Task code P2 receive error (format)	[WRF1F7]	-	✔
	Task code P3 send count	[WRF1F8]	-	✔
	Task code P3 correctly receive count	[WRF1F9]	-	✔
	Task code P3 receive error (command)	[WRF1FA]	-	✔
	Task code P3 receive error (format)	[WRF1FB]	-	✔
	Task code P4 send count	[WRF1FC]	-	✔
	Task code P4 correctly receive count	[WRF1FD]	-	✔
	Task code P4 receive error (command)	[WRF1FE]	-	✔
	Task code P4 receive error (format)	[WRF1FF]	-	✔

✔ : Supported - : Not supported

Function			Software version	
		High Function (MVH)	0110 or older	0120
		Standard (MVL)	1110 or older	1120
I/O	Special I/O	I/O mixed setting	-	✔
		Operation from Control Editor (Jog operation)	-	✔
		Operation from Control Editor (Inching operation)	-	✔
		Operation from Control Editor (Return to origin)	-	✔
Command	Application command (PLSTA) Absolute position coordinate specification		-	✔
	Application command (PLSTAR)		-	✔
	Application command (PLSPDR)		-	✔
	Application command (PLSCNGR)		-	✔
	Application command (PLSTPR)		-	✔
Special internal Output (Bit)	Ch1 Homing returning in progress	[RA0B]	-	✔
	Ch1 Homing returned	[RA0D]	-	✔
	Ch2 Homing returning in progress	[RA1B]	-	✔
	Ch2 Homing returned	[RA1D]	-	✔
	Ch3 Homing returning in progress	[RA2B]	-	✔
	Ch3 Homing returned	[RA2D]	-	✔

✔ : Supported - : Not supported

Function			Software version	
		High Function (MVH)	0125 or older	0126
		Standard (MVL)	1125 or older	1126
Communication	Option board port	Hitachi Serial Data Link (HSDL) protocol	-	✔
Option board	Analog input (OBV-AIG)		-	✔
	Analog input / output (OBV-AIOG)		-	✔
	RTD input (OBV-RTD)		-	✔
Special internal Output (Bit)	R7C8 (Serious failure), R7DB (Self-diagnostic error) turns on in WDT error.		-	✔
	HSDL Run / Stop	[R820]	-	✔
	HSDL Link data update time (max) initialization	[R821]	-	✔
	HSDL Link data update time (min) initialization	[R822]	-	✔
	Modbus F.C.	[R90F]	-	✔
Special internal Output (Word)	Option board analog input 3	[WRF04E]	-	✔
	Option board analog input 4	[WRF04F]	-	✔
	HSDL Link data update time (max)	[WRF0ED]	-	✔
	HSDL Link data update time (current)	[WRF0EE]	-	✔
	HSDL Link data update time (min)	[WRF0EF]	-	✔
	HSDL Status (Station 1, Master)	[WRF150]	-	✔
	HSDL Status (Station 3, Station 2)	[WRF151]	-	✔
	HSDL Status (Station 5, Station 4)	[WRF152]	-	✔
	HSDL Status (Station 7, Station 6)	[WRF153]	-	✔
	HSDL Status (Station 9, Station 8)	[WRF154]	-	✔
	HSDL Status (Station 11, Station 10)	[WRF155]	-	✔
	HSDL Status (Station 13, Station 12)	[WRF156]	-	✔
	HSDL Status (Station 15, Station 14)	[WRF157]	-	✔
	HSDL Status (Station 17, Station 16)	[WRF158]	-	✔
	HSDL Status (Station 19, Station 18)	[WRF159]	-	✔
	HSDL Status (Station 21, Station 20)	[WRF15A]	-	✔
	HSDL Status (Station 23, Station 22)	[WRF15B]	-	✔
	HSDL Status (Station 25, Station 24)	[WRF15C]	-	✔
	HSDL Status (Station 27, Station 26)	[WRF15D]	-	✔
	HSDL Status (Station 29, Station 28)	[WRF15E]	-	✔
	HSDL Status (Station 31, Station 30)	[WRF15F]	-	✔
Error code	H42 Option board verification error		-	✔
	H7D Analog option board *4 conversion processing delay		-	✔
	HA7 Too many files opened in USB memory		-	✔
	HA8 USB download program not supported		-	✔
	HA9 Insufficient USB download program information		-	✔

✔ : Supported - : Not supported

*4 This error will be detected only in OBV-AIG, OBV-AIOG, OBV-RTD.

MEMO