HITACHI PROGRAMMABLE CONTROLLER

HIDIC EH-150

APPLICATION MANUAL

NJI-281H (X)

○ Warranty period and coverage

The warranty period is the shorter period either 18 months from the date 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 either 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

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.

MS-DOS®, Windows®, and Windows NT® 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 classified 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 below:

: 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 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 ⊕, ⊖ of the battery in reverse. Also, do not charge, disassemble, heat, place in fire, or short circuit the battery. There is a risk of explosion or fire.

PROHIBITED
• Do not disassemble or modify the unit. These actions may result in fire, malfunction, or malfunction.

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	Modules developed as second step were added.	1999/05	—
2	Modules developed as third step and the EH-CPU448 were added.	2000/06	NJI-281B(X)
3	The description that it apologized was modified.	2001/01	NJI-281C(X)
4	A precaution about EH-CPU448 was added.	2001/04	NJI-281D(X)
5	Description which runs short was added.	2001/10	NJI-281E(X)
6	The description that it apologized was modified	2002/06	NJI-281F(X)
7	Description of Enhanced CPU(EH-CPU***A) was added.	2002/08	NJI-281G(X)
8	Description of new modules (EH-CPU5**, EH-BS*A, EH-IOCH, and 3 I/O) were added.	2003/09	NJI-281H(X)

Table of contents

MEMO

Chapter 0 New CPU (EH-CPU516/548)

New CPU (EH-CPU516/548) had been released since Aug. 2003.
In this chapter, newly expanded and changed functions for the new CPU are described below. Other functions not described in this chapter are the same as old CPU.

0.1 Expanded function

Expanded function is described below.

0.1.1 Expansion base and max. slot number

The max. number of expansion base and slot are expanded for new CPU as follows.

Table 0.1 The maximum system configuration

	Exp. base	Availability of 11 slot base	Max. slot number
CPU104(A)	0	No	8
CPU208(A), 308(A), 316(A), CPU448(A)	1	No	16 (8×2)
CPU516	2	Yes	33 (11×3)
CPU548	4	Yes	55 (11×5)

Refer to the chapter 2 for further information.

0.1.2 Communication slot expanded

Communication slot is expanded for EH-BS*A as follows.

Type	Communication slot
EH-BS3, BS5, BS8	Slot 0-2
EH-BS3A	Slot 0-2
EH-BS5A	Slot 0-4
EH-BS8A	Slot 0-7
EH-BS11A	Slot 0-7 (Not allowed on slot 8, 9, A)

Refer to chapter 2 for further information.

0.1.3 On delay timer : TM

2,048 points of new on delay timer TM is available in addition to existing timer TD. Time base can be selected from 0.01s, 0.1s and 1s for all TM0 to TM2047. Timer accumulator of TM is TV (0 to 65,535).

New timer TM is supported by LADDER EDITOR for Windows Ver.3.00 or newer. User program including TM does not work with CPU104(A)/208(A)/308(A)/316(A)/448(A) and LADDER EDITOR for Windows Ver.2.** or older.

0.1.4 Virtual remote I/O for internal output

Remote I/O area can be used as internal output like WM if empty slot is assigned as "Remote2". This is useful in case WR or WM memory spent a lot. Refer to the chapter 6 for further information.

0.1.5 Built-in termination resistor

Built-in resistor is added in port 1 of RS-422/485. This can be enabled by setting special internal output. Please refer to chapter 10 and 13 for further information.

0.1.6 Additional commands

The following commands are added.

(1) Inverter control command (FUN 190)
Several Hitachi inverters SJ300 and L300P series are controlled by serial communication from CPU port 1 via RS-485.

(2) Check code generating, verifying (FUN 22, 23)
New command of check code generating for data sending, check code verifying for data receiving. Useful with serial communication command (TRNS0/RECV0).

(3) DeviceNet Explicit message command (FUN 162, 163)
Special communication to DeviceNet slave modules

(4) Data search and table search command (FUN 20, 21)
Same commands supported by Big-H series.

Chapter 1 Features

1. Compact and space saving

   The EH-150 has realized a compact size; 372.5 mm (W) × 100 mm (H) × 109 mm (D) with 1,024 I/Os. The standard I/O module accommodates both a terminal block and LED display as standard, the same as our previous series of programmable controllers. It can also be installed on a DIN rail.
   Each module is enclosed in a capsule-type case for easier handling. By introducing a curved line on the front cover, the system can be neatly consolidated into an ultra high-tech image.

2. 2 communication ports provided as standard

   The EH-150 comes standard with 2 communication ports (serial port 1 and serial port 2) supported by the same type communication method as that for connecting a programming device (personal computer). The display and programming device sold in the market that have been developed for the H series can both be used simultaneously to seamlessly construct a system.
   The setting for the port 1 can be changed, which allows it to function as a general-purpose port that can be controlled by the user program. A communication program can be created for the specific device connected (printer, bar code reader, etc.).

3. Built-in modem connection interface function

   The EH-CPU208(A)/308(A)/316(A)/448(A)/516/548, with a modem connected to the port 1, can communicate with a distant location via commercial phone lines. A system located some distance away can be monitored from an office or a monitor room.

4. Built-in RS-422/485 interface

   By setting the special internal output, serial port 1 can perform communication as an RS-422/485 interface. By using the RS-422/485 interface, a small data link system can be formed between CPU modules or with personal computers via 1:N connection. (Supported by EH-CPU308(A)/316(A)/448(A)/516/548)

5. State-of-art technology and functions have been packed in a compact size.

   The EH-150 contains 32-bit RISC processor (Super H series made by Hitachi, Ltd.) that allows high-speed operations. The user program is stored in FLASH memory so that the user program can be retained in case the battery goes dead. However, batteries are necessary when backing up the data memory.

6. Compatibility with H-series protects software properties

   The EH-150 was developed as part of the H-series family. Those who are currently using the H-series can create and debug programs using the same concepts as they are now. Those who use the EH-150 for the first time, will find that using the H-series programmable controller will greatly broaden the applications of the software properties when they seek to expand their system in the future.

7. Memory board function is supported

   The EH-CPU308(A)/316(A)/448(A)/516/548 can use the new memory boards (EH-MEMP/MEMD) that have been made available. The EH-MEMP is capable of storing up to 48 k steps of user programs. Also, by using the program transfer function, the programs can be copied without the use of peripheral devices. Furthermore, the EH-MEMD is capable of storing a maximum of 384 k words of data.
   When using a memory board, the contents of the memory board are not changed during transfer of programs from peripheral devices or when RUN is in progress. (Only the program within the CPU will be changed.) If changes are necessary, copy the program within the CPU using the program transfer function of the memory board.

8. PID operation function is supported

   PID control can be performed without the use of additional modules. A variety of process inputs and outputs can be supported by directly controlling the analog I/O module. By using this function, continuous control targets such as temperature and flow rate can be controlled in a prompt and smooth manner. (Supported by EH-CPU308(A)/316(A)/448(A)/516/548)

9. Ease of use to facilitate incorporation into other devices

The EH-150 is designed so that it can be installed on easy-to-use DIN rails.

The user programs can be retained even if he or she does not wish to use a battery with the system.

The EH-150 supports a “Online change in RUN” function that allows the user to change the program while it is running.

The standard I/O module uses a removable terminal block, which reduces wiring working drastically. LEDs are provided as standard to check operation status and wiring. Further, the battery can be replaced without having to remove the CPU module from the base unit. The battery is replaced from the front simply by opening the cover of the CPU module.

Chapter 2 System Overview

2.1 Stand alone system

The EH-150 is a module-type programmable controller with the basic configuration shown in Figure 2.1.

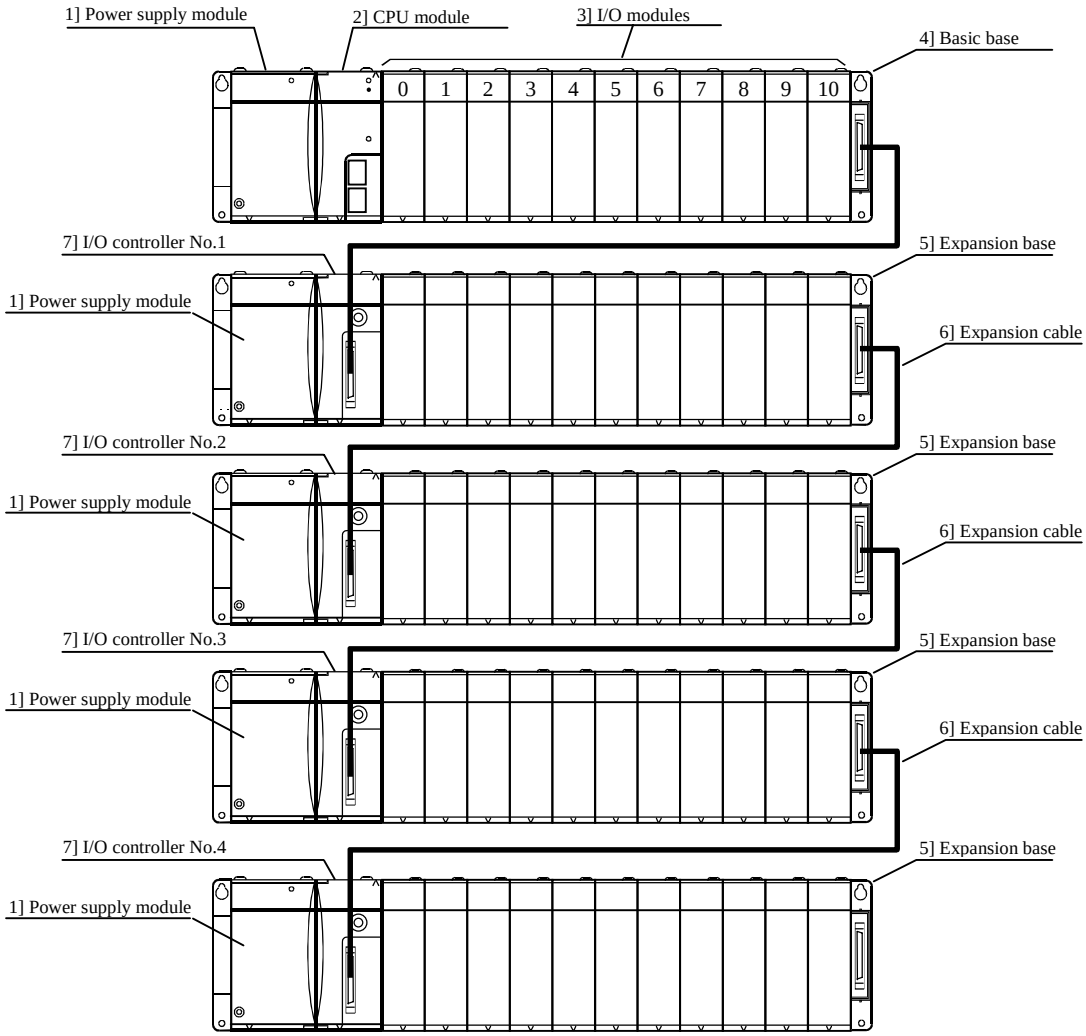


Figure 2.1 EH-150 System configuration diagram

No.	Device name	Description of function
1]	Power module	Converts power supply to the power to be used within the EH-150.
2]	CPU module	Performs operations based on the contents of the user program, receives input and controls output.
3]	I/O module	Input module, output module, analog module, etc.
4]	Basic base	Base in which the power module, CPU module, I/O module, etc. are loaded.
5]	Expansion base	Base in which the power module, I/O controller, I/O module, etc. are loaded.
6]	Expansion cable	Cable that connects the I/O controllers for the basic base and expansion base.
7]	I/O controller	Interface with expansion base and CPU module.

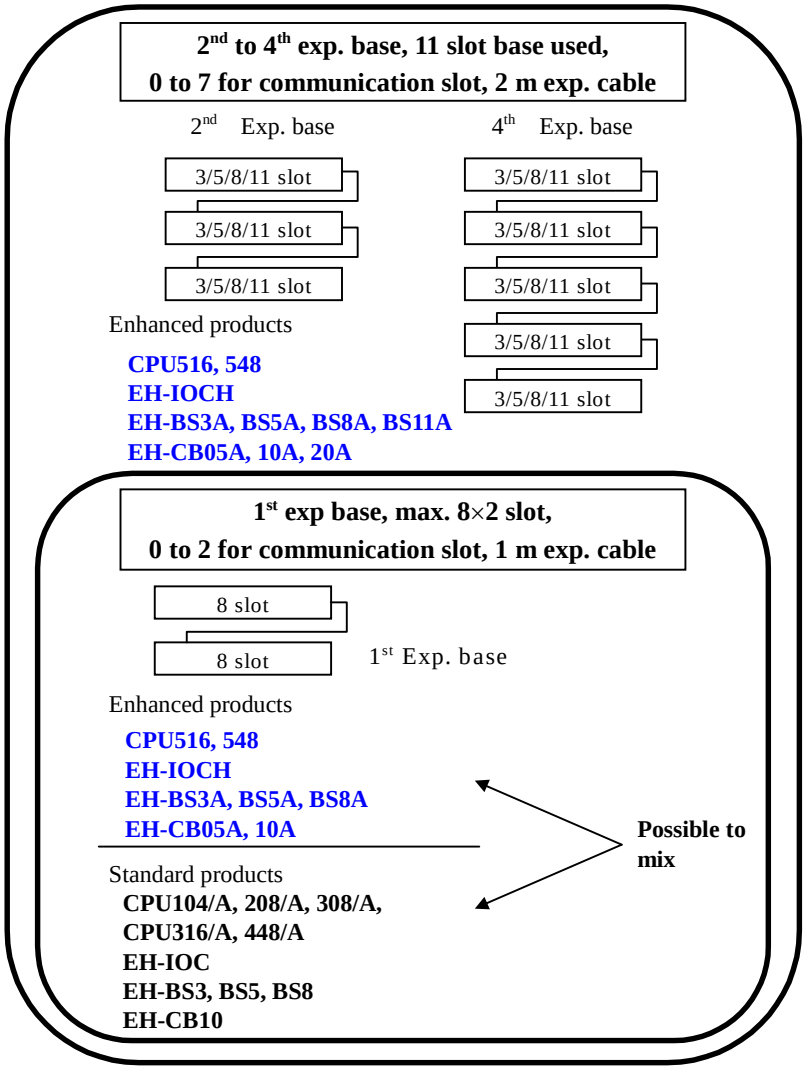
* The basic base 4] and expansion base 5] are the same product.

Maximum system configuration

	Expansion base	11 slot base	Max. slot number
CPU104(A)	0	Not supported	8
CPU208(A), 308(A), 316(A), CPU448(A)	1	Not supported	16 (8×2)
CPU516	2	Supported	33 (11×3)
CPU548	4	Supported	55 (11×5)

2.2 Compatibility

Since the enhanced version products, EH-IOCH, EH-BS3A, 5A, 8A, has upper compatibility with standard products (previous products: EH-IOC, EH-BS3, 5, 8), the both can be mixed within the previous range of expansion, such as one expansion base, 16 slots total, slot 0-2 for communication slot.
If system configuration is beyond this range, such as 2nd to 4th expansion base, 11 slot base used and slot 0-7 for communication slot, be sure to use CPU5xx, EH-IOCH, EH-BS3A, 5A, 8A, 11A only.
(* Communication slot : Special slot for communication modules.)



"Standard products" and "Enhanced products" specified as follows.

	Standard products	Enhanced products
CPU	CPU104(A) / CPU208(A) / CPU308(A) / CPU316(A) / CPU448(A)	CPU516 / CPU548
Base	EH-BS3 / BS5 / BS8	EH-BS3A / BS5A / BS8A / BS11A
I/O controller	EH-IOC	EH-IOCH
Expansion cable	EH-CB10	EH-CB05A / CB10A / CB20A

* If both CPU and basic base are enhanced products (incl. EH-BS11A), communication slot on basic base is effective even if expansion bases are standard products.

2.3 Network system

The EH-150 enables various network systems shown in Figure 2.2.

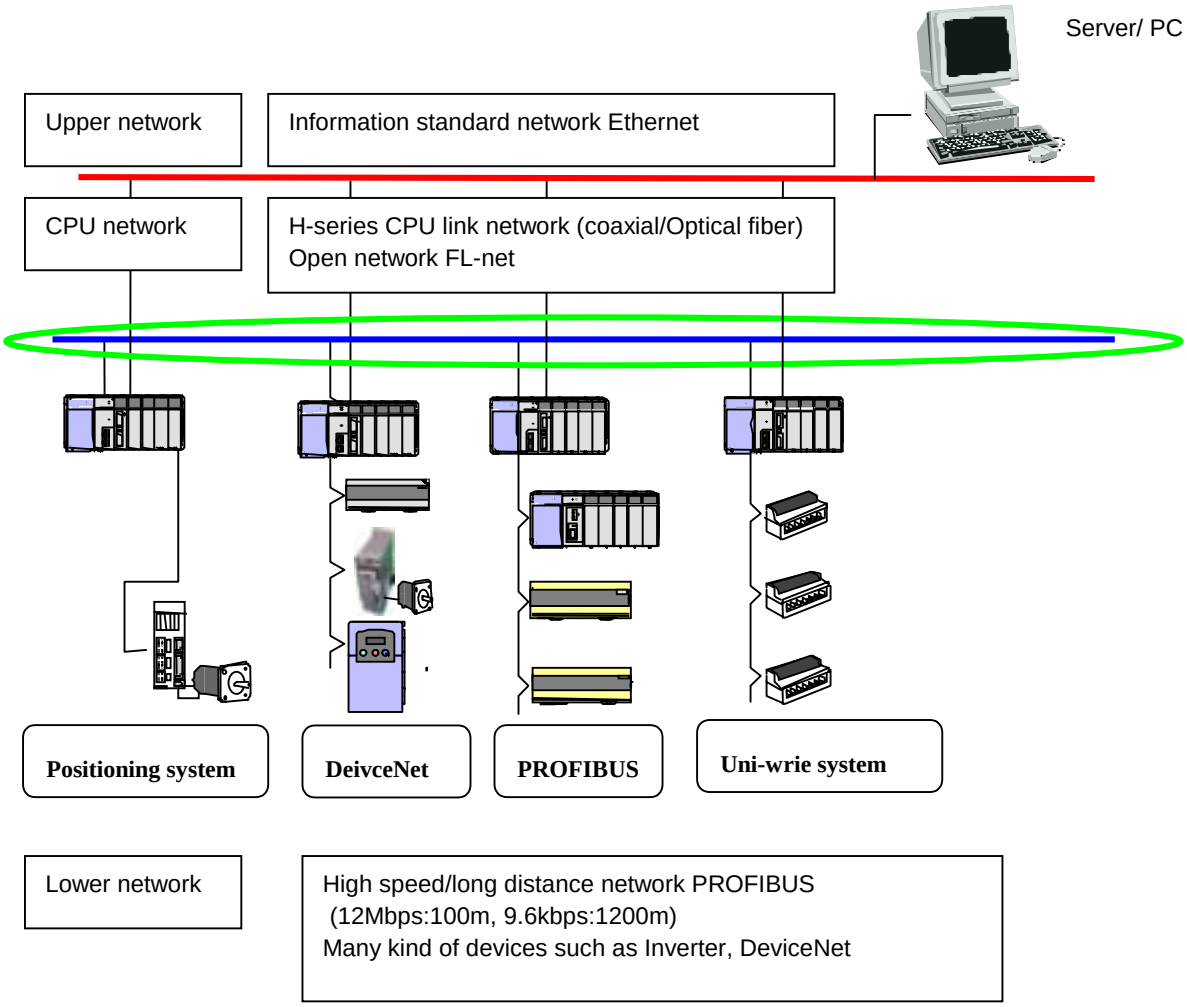


Figure 2.2 EH-150 Network diagram

MEMO

Chapter 3 Specifications

3.1 General Specifications

Item		Specification
Power voltage	AC receiving power	100/110/120 V AC (50/60 Hz), 200/220/240 V AC (50/60 Hz)
	DC receiving power	24 V DC
Power voltage fluctuation range		85 to 264 V AC wide range
		21.6 to 26.4 V DC
Allowable instantaneous power failure		85 to 100 V AC: for a momentary power failure of less than 10 ms, operation continues 100 to 264 V AC: for a momentary power failure of less than 20 ms, operation continues
Operating ambient temperature		0 to 55 °C (Storage ambient temperature -10 to 75 °C)
Operating ambient humidity		20 to 90 % RH (no condensation) (Storage ambient humidity 10 to 90 % RH (no condensation))
Vibration resistance		Conforms to JIS C 0911 (16.7 Hz double amplitude 3 mm X, Y and Z each direction)
Noise resistance		○ Noise voltage 1,500 Vpp Noise pulse width 100 ns, 1 μs (Noise created by the noise simulator is applied across the power supply module's input terminals. This is determined by this company's measuring methods.) ○ Based on NEMA ICS 3-304 (with the exception of input module) ○ Static noise: 3,000 V at metal exposed area
Insulation resistance		20 M Ω or more between the AC external terminal and case ground (FE) terminal (based on 500 V DC mega)
Dielectric withstand voltage		1,500 V AC for 1 minute between the AC external terminal and case ground (FE) terminal
Grounding		Class D grounding (ground with power supply module)
Usage environment		No corrosive gases, no excessive dust
Structure		Open, wall-mounted type
Cooling		Natural air cooling

3.2 Function Specifications

The functions available in the EH-150 are described in the table below.

No.	Item	Description of function
1	Basic functions	The following functions can be achieved when constructing a system using the EH-150. 1] An input signal is received from the control object, operations are performed according to the contents of the program created by the user and the results are output as an output signal. Also, operation results and information during the process can be retained in the internal output area. 2] After power is supplied to the main module and system begins to run, the operation described above is performed continuously until the power is shut off or the system stops running. 3] The information retained internally can be extracted by a device connected externally or can be set in other information. Also, this information is initialized at the time the system begins running, but it can also be retained according to user settings. 4] Operating status can be confirmed with the LED display of the CPU module or I/O module or with an external device that is connected.
2	Setting and display	The following have been provided for the user to set or confirm various types of operation status: 1] Setting switch (CPU module) This sets the CPU communication function setting and operation mode, etc. It can also instruct run and stop. 2] LED display (Power module, CPU module, I/O module) Indicates the power system status, operating status and I/O operation status. 3] Connector (CPU module, basic base, I/O controller) This can connect external devices using RS-232C, etc. It can also add I/O modules. 4] Terminal block (Power module, I/O module) This performs the connections for supplying power, and for exchanging signals with the control object.
3	Number of I/O points	The number of points that can be controlled with respect to the control object is as follows. 1] External I/O If 64-point module is used, EH-CPU104 is able to handle 512 points, and EH-CPU208(A) /308(A)/316(A)/448(A) is 1,024 points, CPU516 is 2,112 points, and CPU548 is 3,520 points. Input is indicated by X, WX, DX and output is indicated by Y, WY, DY. 2] Internal outputs These are areas for temporarily storing information. The I/O numbers are M, WM, DM, R, WR, DR, etc. 3] A timer and a counter are provided internally. 4] Array (only corresponding to substitution statement) Array of I/O numbers can be indicated by placing parentheses (( )) around them.
4	User program memory	The program where the control contents have been described can be stored. The memory for this is in the CPU module and the capacity differs for each CPU module. 1] The memory has a function whereby its contents are retained even if the battery dies. Because of this, it is necessary to initialize the memory since it may have uncertainties right after the unit is purchased. 2] Programming is performed using the programming software (LADDER EDITOR) and peripheral device, etc. for the H-series programmable controllers. 3] The commands that can be used are those designated by the H-series ladder. Refer to the list of EH-150 commands for details.
5	Back-up memory	The contents of a user program can be maintained without a battery. 1] A battery is not required to retain the contents of the user program. Always store the created programs to a floppy disk, etc. just in case something unexpected occurs. 2] When a user program is transferred from a peripheral device or the contents of a user program are changed in the RUN state, the EH-CPU448 writes the user program to the back-up memory without disturbing CPU running. It takes approximately two minutes to write the user program to the back-up memory. * To protect a user program while writing it to the back-up memory (while the special internal output R7EF stays “1”), do not turn off the power of the system.

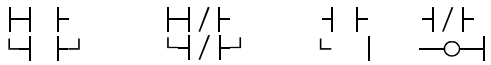
No.	Item	Description of function
5	Back-up memory	3] Several setting, such as communication port (WRF036, WRF037), system processing time (WRF038) and clear mode for link area (WRF07E), are automatically saved to the backup memory if they are changed. * WRF037 is valid only when just after power ON. ⚠ Note FLASH memory is used for the backup memory. If such special internal output values in FLASH memory are changed frequently by user program, the lifetime of FLASH memory will be shorter. Please refer to the chapter 13 in details.
6	Control method	With the EH-150, the user programs are batch converted when operation begins, and the programs after conversion will be executed in order as they are read one by one. 1] The method used for I/O data is that after the I/O data (information) is scanned (execution from the head of the program to the end), it is updated in group (refresh method). If refresh of external I/O is required in the middle of scanning, use the refresh command. 2] Apart from the program that will be normally executed, a periodic scan program which interrupts the normal program at set time intervals and is executed, can be created. The time intervals are 10 ms, 20 ms, and 40 ms. (The EH-CPU448(A)/516/548 supports 5 ms.) 3] The user program is executed from the head of the program to the end, and is once again repeated after the system processing that updates the lapsed timer value, refreshes I/O, and performs communication with peripheral devices, has been performed.
7	Run/stop control	Running and stopping of the CPU module is normally performed by the user. 1] Turn the RUN switch on to begin operation. Turn the switch off to stop operation. 2] The start and stop operation can be performed via communication from peripheral devices (PC) by changing to the REMOTE mode using the setting switch. 3] The start and stop operation can be performed with designated external inputs or internal outputs by designating operation control inputs with a programming device. 4] Apart from the above described operation, if a malfunction is detected in the system while it is running, operation stops and the outputs are shut down (OFF). 5] If the power is shut off and then turned back on while the system is running, operation starts. When the power shuts off, turn off the power to the EH-150, then shut off the external input power. When turning the power back on, turn on the external input power before turning on the power to the EH-150. 6] When starting operation, do so after clearing internal information which is not designated for storage during power failure. When stopping operation, leave the internal information as is, turn off the outputs and then stop. 7] When the power has been cut off for longer than the time allowed for instantaneous power failure, then depending on the system load status, either operation continues or the system perceives that a power shut off has occurred and restarts operation. To ensure operation resumes correctly, have the power remain off for 1 minute or longer.
8	Operation parameters	Each type of condition for operating the EH-150 can be set. The possible settings for operation when an error occurs are given below. 1] Operation can be continued when I/O information does not match. 2] Overload check time can be set. The initial value is 100 ms and operation stops when the time for one scan takes longer than the set overload check time. (overload error) 3] Operation can be made to continue when an overload error occurs. 4] When a power failure (power shutoff) occurs, the internal output area for retaining information and the timer and the counter range can be designated. And, the setting below is possible. 1] The name of the user program can be registered. 2] A password can be set up so that a third party cannot reference the program. 3] It is necessary to register the type of I/O module used as an I/O assignment table. In order to create this I/O assignment table, the type of I/O module that is connected can be read.

No.	Item	Description of function
9	Online change in RUN	A section of a program can be revised while it is running. 1] If revision is done with a programming device and the online change is performed in RUN, the user program in the CPU is changed and the altered program is switched internally at the end of scanning, and operation continues with the new program. 2] When a control command is to be included in the revisions to the program, make the changes after first performing the control command change procedure in the programming device and checking for safety. 3] Until operation starts to continue with the new program, a 'halt period' occurs when the module does not run. External input information is not being received during this time, so leave plenty of allowance in the timing for executing a online change in RUN.
10	Forced set/reset	Forced set and forced reset of the designated I/O can be performed from the programming device connected to the CPU module.
11	Forced output	Output can be forced with respect to the designated I/O number from the programming device connected to the CPU module. For I/O that are not designated, outputs are shut off.
12	Calendar clock function (Not supported by EH-CPU104(A))	EH-CPU208(A)/308(A)/316(A)/448(A)/516/548 has a calendar clock function. 1] Year, month, date, day of the week, hour, minute and second can be set. 2] There is a function for making adjustments in 30 second units. 3] When a battery is not installed, the calendar clock information is not retained when power goes off. The calendar clock must be set again.
13	Dedicated port	This is a communication port with dedicated protocol for the H-series. The communication command called the task code is defined in the port. 1] A programming device can be connected. 2] The ports that can be used as dedicated ports are Port 1 and Port 2. Transmission speed, etc. can be switched using the setting switch.
14	General-purpose port	This is a serial port that can be controlled by the user program. The various settings for communication, actual transmit and receive processing are carried out by the user program. Port 1 can be assigned for this function by switching the setting switch.
15	Modem control (The EH-CPU 104/ 104A are not supported)	A modem can be connected externally for use. It becomes operable when something arrives from outside, and after that, task code communication can be performed. When transmitting from the EH-150, set the port as a general-use port and have it controlled independently with the user program. Port 1 can be assigned for this function by switching the setting switch.
16	Self-diagnosis	Self-diagnostic tests for the following items are performed: 1] Microcomputer check 2] System program area check 3] Memory check 4] User program check 5] Internal output area check 6] Mounted I/O check
17	Abnormal handling	When an abnormal occurs, the error code that shows the error contents are output to special internal output WRF000 as a hexadecimal value. Also, the error is indicated to the outside by the ERR lamp, etc. If the error level is high, the CPU stops operation, but depending on the error, the operation may be continued using user settings. If multiple errors occur, the error code with higher error severity is set. The detailed information is also set to the special internal output. Also, this information is always recorded in the power failure memory, so the information can be referenced even after the power is cut off. (However, a battery is required.) The clearing of the error information can be conducted by turning on R7EC.

No.	Item	Description of function
18	Task code	By combining individual task codes, the following functions can be realized by the program in the host computer: 1] CPU control (RUN/STOP control of CPU, occupy/release, CPU status read, etc.) 2] I/O control (various types of monitoring) 3] Memory write (all clear, batch transfer, etc.) 4] Memory read (reading of programs, etc.) 5] Response (various responses from CPU)
19	Command	Programming can be performed for various purposes and uses by combining ladders and the command language.
20	Setting of System execution time (only EH-CPU104A/ 208A/308A/316A/448(A)/516/548)	EH-150 series CPU executes by periodic 10ms. Conventionally CPU executes system processing for peripheral equipment communication in former 5ms and user program operation in after 5 ms. Enhanced CPU can set any system processing time from 1 to 8 ms. The default is 2ms, in which calculation process becomes higher priority, thus scan time becomes faster and system processing time becomes slow. Set the value according to your application system. See chapter 0.2.4 Setting System Processing Time / 8.1.2 Setting System Processing Time.

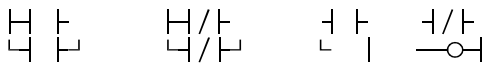
Note: The EH-150 does not support some functions that are supported by the rest of the H series (debugging, tracing, forcing and simulation functions).
Don't select the tracing function among them, because an occupy error will occur when peripheral devices are used and the trace function is selected. If an occupy error should occur, recover by first going off-line and then going back on-line and reconnecting.

3.3 Performance Specifications

Model	Type			EH-CPU104/104A	EH-CPU208/208A
Control specifications	CPU			32-bit RISC processor	
	Processing method			Stored program cyclic method	
	Processing speed	Basic commands		1.0 μs/command	
		Application commands		Several 10 μs/command	
	User program memory			3.5 k steps	7.6 k steps
Operation processing specifications	Command language	Basic commands		39 types such as LD, LDI, AND, ANI, OR, ORI, ANB, ORB, OUT, MPS, MRD, MPP	
		Arithmetic command Application commands		59 types such as arithmetic (+ – × ÷ =, etc.), jump, subroutine, division, extraction, etc.	
	Ladder	Basic commands		39 types, such as 	
		Arithmetic command Application commands		59 types such as arithmetic (+ – × ÷ =, etc.), jump, subroutine, division, extraction, etc.	
I/O processing specifications	External I/O	I/O processing method		Refresh processing	
		64 points I/O module		512 points max.	1,024 points max.
		Expansion unit		Not supported	1
	Internal output	Bit		1,984 points (R0 to R7BF)	
		Word		4,096 words (WR0 to WRFFF)	8,192 words (WR0 to WR1FFF)
		Special	Bit	64 points (R7C0 to R7FF)	
			Word	512 words (WRF000 to WRF1FF)	
		CPU link		16,384 points 1,024 words × 2 loops (L0 to L3FFF/L10000 to L13FFF, WL0 to WL3FF/WL1000 to WL13FF)	
		Remote I/O		-	
		Bit/word shared		16,384 points 1,024 words (M0 to M3FFF, WM0 to WM3FF)	
	Timer and counter	Number of points		512 points (TD + CU) (TD is up to 256 points *1)	
		Timer set value		0 to 65,535, time base 0.01s, 0.1s, 1s (64 points are maximum for 0.01 s. *2)	
		Counter set value		1 to 65,535	
	Edge detection			512 points (DIF0 to DIF511: decimal) + 512 points (DFN0 to DFN511: decimal)	
Peripheral devices	Program method			Command language, ladder diagram	
	Peripheral device			Programming software (LADDER EDITOR DOS version/Windows® version) Command language programmer, Portable graphic programmer, Graphic input device	
Maintenance functions	Self-diagnosis			PC abnormal (LED display): microcomputer error, watchdog timer error, memory error, program error, system ROM/RAM error, scan time monitoring, battery under-voltage detection, and others	

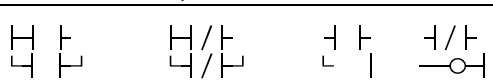
*1: The same numbers cannot be shared by the time and the counter. TD is 0 to 255.

*2: Only timers numbered 0 to 63 can use 0.01s for their time base.

Model	Type			EH-CPU308/308A	EH-CPU316/316A
Control specifications	CPU			32-bit RISC processor	
	Processing method			Stored program cyclic method	
	Processing speed	Basic commands		1.0 μs/command	
		Application commands		Several 10 μs/command	
	User program memory			7.6 k steps	15.7 k steps
Operation processing specifications	Command language	Basic commands		39 types such as LD, LDI, AND, ANI, OR, ORI, ANB, ORB, OUT, MPS, MRD, MPP	
		Arithmetic command Application commands		59 types such as arithmetic (+ – × ÷ =, etc.), jump, subroutine, division, extraction, etc.	
	Ladder	Basic commands		39 types, such as 	
		Arithmetic command Application commands		59 types such as arithmetic (+ – × ÷ =, etc.), jump, subroutine, division, extraction, etc.	
I/O processing specifications	External I/O	I/O processing method		Refresh processing	
		64 points I/O module		1,024 points maximum	1,024 points maximum
		Expansion unit		1	
	Internal output	Bit		1,984 points (R0 to R7BF)	
		Word		17,408 words (WR0 to WR43FF)	22,528 words (WR0 to WR57FF)
		Special	Bit	64 points (R7C0 to R7FF)	
			Word	512 words (WRF000 to WRF1FF)	
		CPU link		16,384 points 1,024 words × 2 loops (L0 to L3FFF/L10000 to L13FFF, WL0 to WL3FF/WL1000 to WL13FF)	
		Remote I/O		-	
		Bit/word shared		16,384 points 1,024 words (M0 to M3FFF, WM0 to WM3FF)	
	Timer and counter	Number of points		512 points (TD + CU) (TD is up to 256 points *1)	
		Timer set value		0 to 65,535, time base 0.01s, 0.1s, 1s (64 points are maximum for 0.01 s. *2)	
		Counter set value		1 to 65,535	
	Edge detection			512 points (DIF0 to DIF511: decimal) + 512 points (DFN0 to DFN511: decimal)	
Peripheral devices	Program method			Command language, ladder diagram	
	Peripheral device			Programming software (LADDER EDITOR DOS version/Windows® version) Command language programmer, Portable graphic programmer, Graphic input device	
Maintenance functions	Self-diagnosis			PC abnormal (LED display): microcomputer error, watchdog timer error, memory error, program error, system ROM/RAM error, scan time monitoring, battery under-voltage detection, and others	

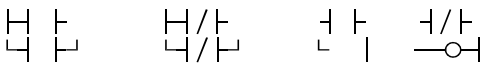
*1: The same numbers cannot be shared by the time and the counter. TD is 0 to 255.

*2: Only timers numbered 0 to 63 can use 0.01s for their time base.

Model	Type			EH-CPU448/448A
Control specifications	CPU			32-bit RISC processor
	Processing method			Stored program cyclic method
	Processing speed	Basic commands		0.1 μs/command
		Application commands		Several 10 μs/command
	User program memory			48.5 k steps
Operation processing specifications	Command language	Basic commands		39 types such as LD, LDI, AND, ANI, OR, ORI, ANB, ORB, OUT, MPS, MRD, MPP
		Arithmetic command Application commands		113 types such as arithmetic (+ – × ÷ =, etc.), jump, subroutine, division, extraction, etc.
	Ladder	Basic commands		39 types, such as 
		Arithmetic command Application commands		113 types such as arithmetic (+ – × ÷ =, etc.), jump, subroutine, division, extraction, etc.
I/O processing specifications	External I/O	I/O processing method		Refresh processing
		64 points I/O module		1,024 points maximum
		Expansion unit		1
	Internal output	Bit		1,984 points (R0 to R7BF)
		Word		50,176 words (WR0 to WRC3FF)
		Special	Bit	64 points (R7C0 to R7FF)
			Word	512 words (WRF000 to WRF1FF)
		CPU link		16,384 points 1,024 words × 2 loops (L0 to L3FFF/L10000 to L13FFF, WL0 to WL3FF/WL1000 to WL13FF)
		Remote I/O		-
		Bit/word shared		16,384 points 1,024 words (M0 to M3FFF, WM0 to WM3FF)
	Timer and counter	Number of points		512 points (TD + CU) (TD is up to 256 points *1)
		Timer set value		0 to 65,535, timer base 0.01s, 0.1s, 1s (64 points are maximum for 0.01 s. *2)
		Counter set value		1 to 65,535 times
	Edge detection			512 points (DIF0 to DIF511: decimal) + 512 points (DFN0 to DFN511: decimal)
Peripheral devices	Program method			Command language, ladder diagram
	Peripheral device			Programming software (LADDER EDITOR DOS version/Windows® version) Command language programmer, Portable graphic programmer, Graphic input device
Maintenance functions	Self-diagnosis			PC abnormal (LED display): microcomputer error, watchdog timer error, memory error, program error, system ROM/RAM error, scan time monitoring, battery under-voltage detection, and others

*1: The same numbers cannot be shared by the timer and the counter. TD is 0 to 255.

*2: Only timers numbered 0 to 63 can use 0.01s for their time base.

Model	Type			EH-CPU516	EH-CPU548
Control specifications	CPU			32-bit RISC processor	
	Processing method			Stored program cyclic method	
	Processing speed	Basic commands		0.1 μs/command	
		Application commands		Several 10 μs/command	
	User program memory			15.7 k steps	48.5 k steps
Operation processing specifications	Command language	Basic commands		39 types such as LD, LDI, AND, ANI, OR, ORI, ANB, ORB, OUT, MPS, MRD, MPP	
		Arithmetic command Application commands		59 types such as arithmetic (+ – × ÷ =, etc.), jump, subroutine, division, extraction, etc.	
	Ladder	Basic commands		39 types, such as 	
		Arithmetic command Application commands		59 types such as arithmetic (+ – × ÷ =, etc.), jump, subroutine, division, extraction, etc.	
I/O processing specifications	External I/O	I/O processing method		Refresh processing	
		64 points I/O module		2,112 points maximum	3,520 points maximum
		Expansion unit		2	4
	Internal output	Bit		1,984 points (R0 to R7BF)	
		Word		22,528 words (WR0 to WR57FF)	50,176 words (WR0 to WRC3FF)
		Special	Bit	64 points (R7C0 to R7FF)	
			Word	512 words (WRF000 to WRF1FF)	
		CPU link		16,384 points 1,024 words × 2 loops (L0 to L3FFF/L10000 to L13FFF, WL0 to WL3FF/WL1000 to WL13FF)	
		Remote I/O		512 points × 4 master stations	
		Bit/word shared		16,384 points 1,024 words (M0 to M3FFF, WM0 to WM3FF)	
	Timer and counter	Number of points		512 points (TD + CU) (TD is up to 256 points *1) 2,048 points (TM)	
		Timer set value		0 to 65,535, time base 0.01s, 0.1s, 1s (64 points are maximum for 0.01 s. *2)	
		Counter set value		1 to 65,535	
	Edge detection			512 points (DIF0 to DIF511: decimal) + 512 points (DFN0 to DFN511: decimal)	
Peripheral devices	Program method			Command language, ladder diagram	
	Peripheral device			Programming software (LADDER EDITOR DOS version/Windows® version) Command language programmer, Portable graphic programmer, Graphic input device	
Maintenance functions	Self-diagnosis			PC abnormal (LED display): microcomputer error, watchdog timer error, memory error, program error, system ROM/RAM error, scan time monitoring, battery under-voltage detection, and others	

*1: The same numbers cannot be shared by the time and the counter. TD is 0 to 255.

*2: Only timers numbered 0 to 63 can use 0.01s for their time base.

MEMO

Chapter 4 Product lineup

4.1 Product lineup List

(1) Basic devices

Table 4.1 List of system equipment (1/2)

Product	Type	Specification	I/O assignment	Remarks
CPU module	EH-CPU104(A)	512 I/O points max.*1, 4 k steps, 1.0μs/basic command (No exp. base possible.)	—	Fixed position *2
	EH-CPU208(A)	1,024 I/O points max.*1, 8 k steps, 1.0μs/basic command, clock function, 1 exp. base, modem control function	—	
	EH-CPU308(A)	1,024 I/O points max.*1, 8 k steps, 1.0μs/basic command, clock function, 1 exp. base, modem control function, Memory board function, RS-422/485 port, PID command, floating point operation		
	EH-CPU316(A)	1,024 I/O points max.*1, 16 k steps, 1.0μs/basic command, clock function, 1 exp. base, modem control function, Memory board function, RS-422/485 port, PID command, floating point operation	—	
	EH-CPU448(A)	1,024 I/O points max.*1, 48 k steps, 0.1μs/basic command, clock function, 1 exp. base, modem control function, Memory board function, RS-422/485 port, PID command, floating point operation		
	EH-CPU516	2,112 I/O points max.*1, 16 k steps, 0.1μs/basic command, clock function, 2 exp. base, modem control function, Memory board function, RS-422/485 port, PID command, floating point operation	—	
	EH-CPU548	3,520 I/O points max.*1, 48 k steps, 0.1μs/basic command, clock function, 4 exp. base, modem control function, Memory board function, RS-422/485 port, PID command, floating point operation	—	
Memory board	EH-MEMP *4	Program capacity 48 k steps	—	Mounted on optional slot *2
	EH-MEMD *4	Program capacity 16 k steps, Data capacity 384 k words	—	
I/O controller	EH-IOCH	I/O control module, 1 unit/exp. base	—	Fixed position on CPU slot *2
Power module	EH-PSA	Input 100 to 240 V AC Output 5 V DC 3.8 A, 24 V DC 0.4 A	—	Fixed position *2
	EH-PSD	Input 21.6 to 26.4 V DC Output 5 V DC 3.8 A	—	
Base unit	EH-BS3A	3 I/O modules mounted	—	Common for both basic and expansion base
	EH-BS5A	5 I/O modules mounted	—	
	EH-BS8A	8 I/O modules mounted	—	
	EH-BS11A	11 I/O modules mounted	—	
Input module	EH-XD8	8 points, 24 V DC input	X16	
	EH-XD16	16 points, 24 V DC input	X16	
	EH-XDL16	16 points, 24 V DC input, High cut filter	X16	
	EH-XD32	32 points, 24 V DC input	X32	
	EH-XD32E	32 points, 24 V DC input, Euro-terminal	X32	
	EH-XDL32E	32 points, 24 V DC input, Euro-terminal, High cut filter	X32	
	EH-XD64	64 points, 24 V DC input	X64	
	EH-XA16	16 points, 100 to 120 V AC input	X16	
	EH-XAH16	16 points, 200 to 240 V AC input	X16	
Output module	EH-YT8	8 points, transistor output (sink type)	Y16	
	EH-YR8B	8 points, relay output, Separated type, Built-in varistor		
	EH-YR12	12 points, relay output	Y16	
	EH-YR16	16 points, relay output		
	EH-YTP8	8 points, transistor output (source type)	Y16	
	EH-YT16	16 points, transistor output (sink type)	Y16	
	EH-YTP16	16 points, transistor output (source type)	Y16	
	EH-YTP16S	16 points, transistor output (source type), short-circuit protection	Y16	Electric short circuit protection
	EH-YT32	32 points, transistor output (sink type), short-circuit protection *3	Y32	
	EH-YTP32	32 points, transistor output (source type), short-circuit protection *3	Y32	

Table 2 Basic components (2/2)

Product	Type	Specification	I/O assignment	Remarks
Output module	EH-YT32E	32 points, transistor output (sink type), short-circuit protection, Euro-terminal	Y32	Electric short circuit protection
	EH-YTP32E	32 points, transistor output (source type), short-circuit protection Euro-terminal	Y32	
	EH-YT64	64 points, transistor output (sink type), short-circuit protection	Y64	
	EH-YTP64	64 points, transistor output (source type), short-circuit protection	Y64	
	EH-YS4	4 points, triac output	Y16	
	EH-YS16	16 points, triac output	Y16	
Analog input module	EH-AX44	4 ch. Current input (0 - 3 ch.), 4 to 20 mA 4 ch. Voltage input (4 - 7 ch.), 0 to 10 VDC	X8W	
	EH-AX8V	8 ch. Current output (0 - 7ch.), 0 to 10 VDC	X8W	
	EH-AX8H	8 ch. Current output (0 - 7ch.), -10 to 10 VDC	X8W	
	EH-AX8I	8 ch. Current input (0 - 7ch.), 4 to 20 mA	X8W	
	EH-AX8IO	8 ch. Current input (0 - 7ch.), 0 to 22 mA	X8W	
Analog output module	EH-AY22	2 ch. Voltage output (0 - 1ch.), 0 to 10 VDC 2 ch. Current output (2 - 3 ch.), 4 to 20 mA	Y8W	
	EH-AY2H	2 ch. Voltage output (0 - 1ch.), -10 to 10 VDC	Y8W	
	EH-AY4V	4 ch. Voltage output (0 - 3 ch.), 0 to 10 VDC	Y8W	
	EH-AY4H	4 ch. Voltage output (0 - 3 ch.), -10 to 10 VDC	Y8W	
	EH-AY4I	4 ch. Current output (0 - 3 ch.), 4 to 20 mA	Y8W	
RTD Input Module	EH-PT4	4 ch. RTD input, Signed 15 bits, Pt 100 Ω /Pt 1000 Ω	X4W	
High function module	EH-CU	2 ch. High speed counter input, maximum frequency of 100 kHz, 1/2-phase switchable, 4-point open collector output	FUN0	
	EH-CUE	1 ch. High speed counter input, maximum frequency of 100 kHz, 1/2-phase switchable, 2-point open collector output	FUN0	
	EH-POS	Positioning module (Single Axis)	4W/4W	
	EH-POS4 *5	Positioning module (4 Axes)	4W/4W	
	EH-ETH *5	Ethernet module IEEE802.3 standard, 10BASE-T, 2 units/CPU	COMM	On slot 0-2 of EH-BS3/5/8. On slot 0-7 of EH-BS3A/5A/ 8A/11A.
	EH-LNK *5	CPU link module (coaxial), Up to 2 units/CPU	LINK	
	EH-OLNK *5	CPU link module (optical fiber), 2 units/CPU	LINK	
	EH-RMD *6	Device Net master module 256/256 words in/output (LINK), Up to 2 units/CPU (LINK), 64 words in/output total (Remote2), Up to 4 units/CPU (Remote2)	LINK / Remote2	
	EH-RMP *4	PROFIBUS-DP master module 256/256 words in/output, Up to 2 units/CPU	LINK	
	EH-IOCD	Device Net slave module 256/256 words in/output	—	Fixed position on CPU slot *2
	EH-IOCP	PROFIBUS-DP slave controller, 208 words in/output total	—	Fixed position on CPU slot *2
Dummy module	EH-DUM	Module for open slots	Empty 16	

*1: I/O points in case of 64 points module used.

*2: Mounted on special fixed position.

*3: Short circuit protection is effective from May 2001 production or later (MFG No.01E**)

*4: Supported by CPU308(A)/316(A)/448(A)/516/548

*5: Supported by CPU308A/316A/448(A)/516/548

*6: Supported by CPU308(A)/316(A)/448(A)/516/548 for CPU LINK assignment. Supported by CPU516/548 for REMOTE2 assignment.

(2) Peripheral devices

Table 4.2 List of peripheral devices

Product	Form	Specification	Remarks
Portable graphic programmer	PGM-GPH	Portable graphic programmer with a 2 m (6.56 ft.) connection cable (PGCB02H)	*7
Command language programmer	PGM-CHH	Command language programmer	
Graphic input device support software	HL-PC3	Ladder diagram/Command language editor LADDER EDITOR (for PC98 series) with CPU connection cable	
	HL-AT3E	Ladder diagram/Command language editor LADDER EDITOR (for PC/AT compatible personal computer)	
	HLW-PC3	Ladder diagram/Command language editor LADDER EDITOR (for Windows® 95/NT 4.0)	
	HLW-PC3E	Ladder diagram/Command language editor (English version) LADDER EDITOR (for Windows® 95/98/NT)	
	HLW-PCR	Ladder diagram/Command language editor LADDER EDITOR (for Windows® 95/NT 4.0), CD-ROM Version.	
	HLW-PC3L05	Ladder diagram/Command language editor LADDER EDITOR (for Windows® 95/NT 4.0), License Pack: 5 users	
	HLW-PC3L10	Ladder diagram/Command language editor LADDER EDITOR (for Windows® 95/NT 4.0), License Pack: 10 users	
	HLW-PC3L30	Ladder diagram/Command language editor Ladder Editor (for Windows® 95/NT 4.0), License Pack: 30 users	
	HLW-PC3L50	Ladder diagram/Command language editor Ladder Editor (for Windows® 95/NT 4.0), License Pack: 50 users	
	Pro-H	IEC61131-3	

Note: MS-DOS, Windows® 95, and Windows NT® are registered trademarks of Microsoft Corporation in the United States.
HI-LADDER (attached to GPCL01H) can also be used.

*7: Don't use the option box (model name: PGMIF1H) for the portable graphic programmer.
Its high current may cause the EH-150 system to break down.

(3) Connection cable

Table 4.3 List of connection cables

Product	Model name	Specification	Remarks
Cable for connecting basic base to I/O controller	EH-CB10/A	Length: 1 m (3.28 ft.) (Basic/exp. base - I/O controller)	*8
	EH-CB05A	Length: 1 m (1.64 ft.) (Basic/exp. base - I/O controller)	
	EH-CB20A	Length: 1 m (6.56 ft.) (Basic/exp. base - I/O controller)	
I/O connector cable for EH-POS	EH-POC10	Length: 1m (3.28 ft.)	
	EH-POC20	Length: 2m (6.56 ft.)	
	EH-POC50	Length: 5m (16.4 ft.)	
Conversion cable for connecting peripheral devices	EH-RS05	Length: 0.5 m (19.69 in.)	*9
For portable graphic programmer, command language programmer	PGCB02H	Length: 2 m (6.56 ft.), between CPU and programmer	
Peripheral devices	GPCB02H	Length: 2 m (6.56 ft.), between CPU and graphic input device	
	GPCB05H	Length: 5 m (16.40 ft.), between CPU and graphic input device	
	GPCB15H	Length: 15 m (49.22 ft.), between CPU and graphic input device	
	CBPGB	Length: 2 m (6.56 ft.), between graphic input device and printer	
	LP100	Length: 2 m (6.56 ft.), between graphic input device and kanji printer	
	KBADPTH	Length: 15 m (49.22 ft.), between graphic input device and JIS keyboard	
	PCCB02H	Length: 2 m (6.56 ft.), between CPU and PC98 series	
	WPCB02H	Length: 2 m (6.56 ft.), between CPU and PC98 series (25-pin)	*10
	WVCB02H	Length: 2 m (6.56 ft.), between CPU and DOS/V (9-pin)	*10
	EH-VCB02	Length: 2 m (6.56 ft.), between CPU (modular jack type) and DOS/V (9-pin)	*10

*8: EH-CB10 (without A version) can not be used with 2nd - 4th expansion bases.

*9: Required when connecting to programmer peripheral device (including PC98, DOS/V, and PC/AT compatibles)

*10: EH-VCB02, WPCB02H and WVCB02H are cables for LADDER EDITOR for Windows®

(4) . Save wiring equipment

Table 4.4 List of save wiring equipment

Product	Model name	Specification	Remarks
Distributed I/O units	RDX16D	16 points input, 24V DC, based on DeviceNet.	
	RDY16T	16 points, transistor output 0.3 A (sink type), based on DeviceNet.	
	RDY16R	16 points input, relay output 2 A DC, based on DeviceNet.	
Relay terminal block	HPX7DS-40V6	Relay terminal for 32 / 64 points I/O module	
Cable for relay terminal block	EH-CBM01W	Length: 1m, both edges connector.	
	EH-CBM03W	Length: 3m, both edges connector.	
	EH-CBM05W	Length: 5m, both edges connector.	
	EH-CBM10W	Length: 10m, both edges connector.	
Cable for I/O wiring	EH-CBM01	Length: 1m, one edge connector, (for 32 / 64 I/O module.)	
	EH-CBM03	Length: 3m, one edge connector, (for 32 / 64 I/O module.)	
	EH-CBM05	Length: 5m, one edge connector, (for 32 / 64 I/O module.)	
	EH-CBM10	Length: 10m, both ends connector, (for 32 / 64 I/O module.)	

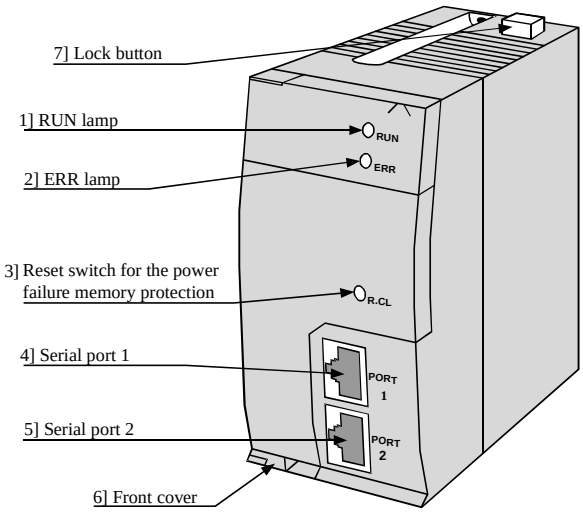
(5) Others

Table 4.5 List of others

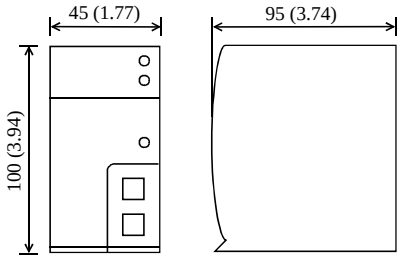
Form	Usage	Remarks
LIBAT-H	Lithium battery	The battery is used in common with the H series.
EH-LCN	L-type connector for the turn of coaxial connector. (for coaxial type CPU link module.)	

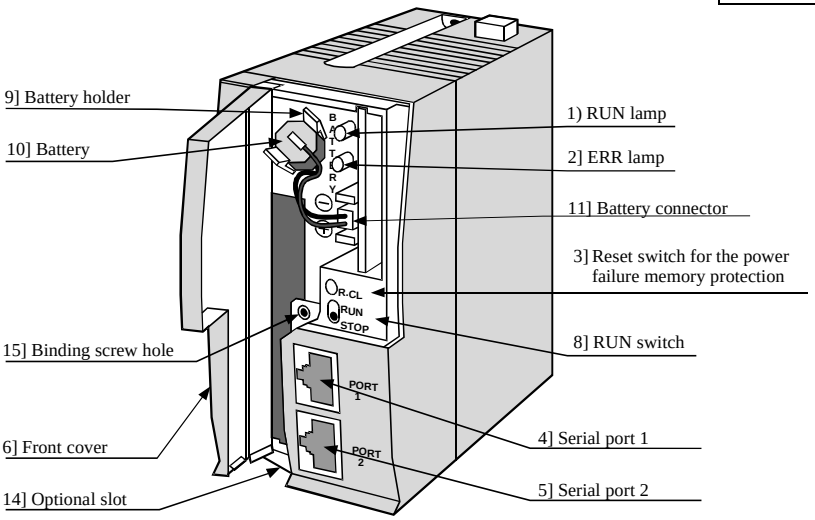
4.2 CPU Module

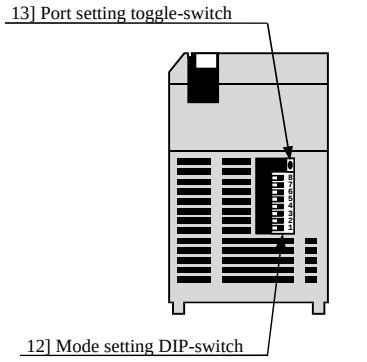
Name and function of each part



Type	EH-CPU104/A	EH-CPU516
	EH-CPU208/A	EH-CPU548
	EH-CPU308/A	
	EH-CPU316/A	
	EH-CPU448/A	
Weight	Approx. 0.18 kg (0.4 lb.) (EH-CPU448 weights approx. 0.20 kg (0.44 lb).)	
Current consumption	400 mA	
Dimensions (mm (in.))		







Drawing of CPU module bottom

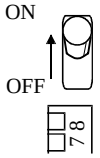
No.	Name	Function	Remark2s
1]	RUN lamp	Indicates that the CPU is running.	Green
2]	ERR lamp	Indicates that the CPU is generating an abnormal.	Red
3]	Reset switch for the power failure memory protection	Clears the area designated for power failure memory.	Enabled only during a stoppage
4]	Serial port 1	Connects a programming device, etc..	Refer to the detailed explanation.
5]	Serial port 2	Connects a programming device, etc..	Refer to the detailed explanation.
6]	Front cover	This is used when operating the RUN switch or replacing the battery.	
7]	Lock button	This is used when removing the module from the base unit.	
8]	RUN switch	Used to control the running status.	
9]	Battery holder	Used for putting a battery in.	
10]	Battery	Retains power failure memory data or clock data.	
11]	Battery connector	Used to connect a battery.	
12]	Mode setting DIP-switch	Sets the operating status for the serial port, etc.	
13]	Port setting toggle-switch	Switches the operating status for the serial port 2.	
14]	Optional slot	Memory board is loaded.	
15]	Binding screw hole	Fix the memory board with binding screw.	

<table>
<tr><th>No.</th><th>Item</th><th>Detailed explanation</th><th>Remarks</th></tr>
<tr><td colspan="2">Explanation of operation</td><td>Operations are performed according to the contents of the program created by the user.
The programming device connected to the CPU module communication port writes and reads the user program.
Memory is installed inside the CPU module in which the user program and internal output information are stored.
The internal output data and clock information can be backed up with the battery. (The clock function is not available with the EH-CPU104)</td><td></td></tr>
<tr><td>1]</td><td>RUN lamp</td><td>This indicates the CPU is running and lights up during normal operation.</td><td>The lamp is green.</td></tr>
<tr><td>2]</td><td>ERR lamp</td><td>The error contents can be determined by the combination of lighting and flashing of the ERR and RUN lamps.
For details, refer to “Chapter 12 The Error Code List”.</td><td>The lamp is red.</td></tr>
<tr><td>3]</td><td>Reset switch for the power failure memory protection</td><td>Pressing this switch when the operation is stopped, clears data from the area designated for power failure memory. The program information is retained as is.</td><td></td></tr>
<tr><td>4]</td><td>Serial port 1</td><td>Can be switched to use as a dedicated port or general-purpose port using the mode setting switch. It is also equipped with modem control functions.
(The modem control function is not available with the EH-CPU104(A))</td><td></td></tr>
<tr><td>5]</td><td>Serial port 2</td><td>Can be used as a dedicated port. A programming device can be connected and tasks such as CPU programming or monitoring can be performed.</td><td></td></tr>
<tr><td>6]</td><td>Front cover</td><td>Open and close the front cover when operating the RUN switch or replacing the battery. Keep the cover closed while the module is running. Also, when the cover is open, do not touch the printed wiring board with your hands.</td><td></td></tr>
<tr><td>7]</td><td>Lock button</td><td>When dismounting the module from a base unit, press this button and lift up the module. The module can be fixed firmly by a screw (M4, 10 mm (0.39 in.)).</td><td></td></tr>
<tr><td>8]</td><td>RUN switch</td><td>To designate run status, push this switch lever to the "RUN" side. The following conditions are necessary for the module to run correctly:
1. A user program must be written to the module.
2. When a run definition input has been set, the designated input must be on.
3. There should be no sources of errors.
4. The mode setting switch is set to the normal run status.
5. When the mode setting switch is set to "REMOTE," there should be a run start command from the personal computer, etc. that is connected.
Also, to stop operation, set this switch lever to the "STOP" side.</td><td></td></tr>
<tr><td>9]
10]
11]</td><td>Battery holder
Battery
Battery connector</td><td>The module is shipped with the battery connector disconnected so as to prevent unnecessary consumption of battery during distribution or storage. When using the CPU module, check the battery and connect the lead connector on the battery to the mating connector on the circuit board.
1. Clock data is retained by battery power and thus has not been written to the clock element initially. When using the clock function, first write the clock data to the clock element.
(The EH-CPU104/104A does not provide clock function.)
See Chapter 11 for procedures for setting the clock data.
2. Use the reference table below to determine the remaining life of battery. As a guideline, replace the battery every two years even when the total power failure time is less than the guaranteed value shown in the table.
<table>
<tr><th colspan="2">Battery life (Total power failure time) [Hr]</th></tr>
<tr><td>Guaranteed value (MIN) @55°C</td><td>Actual value (MAX) @25°C</td></tr>
<tr><td>2,000</td><td>32,000</td></tr>
</table>
</td><td></td></tr>
</table>

<table>
<tr><th>No.</th><th>Item</th><th colspan="10">Detailed explanation</th><th>Remarks</th></tr>
<tr><td>12]

13]</td><td>Mode setting switch

PHL switch</td><td colspan="10">The following operating modes are designated by setting this switch. Even if the switch setting is changed while the module is energizing, the operating mode will not change. To switch operating modes, turn the power off, then do the settings correctly. (When DR signal turns on, modifying the baud rate.)</td><td></td></tr>
<tr><td></td><td></td><td colspan="3">DIP switch</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>PHL</td><td></td></tr>
<tr><td></td><td></td><td colspan="2">RUN/STOP</td><td>Remote</td><td>ON</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td></td></tr>
<tr><td></td><td></td><td colspan="2"></td><td>RUN switch</td><td>off</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td></td></tr>
<tr><td></td><td></td><td rowspan="5">Port 1</td><td rowspan="4">Dedicated port *</td><td>4,800 bps</td><td>-</td><td>-</td><td>ON</td><td>ON</td><td>ON</td><td>-</td><td>-</td><td></td></tr>
<tr><td></td><td></td><td></td><td>9,600 bps</td><td>-</td><td>-</td><td>off</td><td>ON</td><td>ON</td><td>-</td><td>-</td><td></td></tr>
<tr><td></td><td></td><td></td><td>19,200 bps</td><td>-</td><td>-</td><td>ON</td><td>off</td><td>ON</td><td>-</td><td>-</td><td></td></tr>
<tr><td></td><td></td><td></td><td>38,400 bps</td><td>-</td><td>-</td><td>off</td><td>off</td><td>ON</td><td>-</td><td>-</td><td></td></tr>
<tr><td></td><td></td><td colspan="2">General purpose port</td><td>-</td><td>off</td><td>-</td><td>-</td><td>off</td><td>-</td><td>-</td><td></td></tr>
<tr><td></td><td></td><td colspan="2">Modem mode</td><td>-</td><td>ON</td><td>-</td><td>-</td><td>off</td><td>-</td><td>-</td><td></td></tr>
<tr><td></td><td></td><td rowspan="4">Port 2</td><td rowspan="4">Dedicated port *</td><td>4,800 bps</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>off</td><td>off</td><td></td></tr>
<tr><td></td><td></td><td></td><td>9,600 bps</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>ON</td><td>off</td><td></td></tr>
<tr><td></td><td></td><td></td><td>19,200 bps</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>off</td><td>ON</td><td></td></tr>
<tr><td></td><td></td><td></td><td>38,400 bps</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>ON</td><td>ON</td><td></td></tr>
<tr><td></td><td></td><td colspan="10">Switch 7 and 8 should be always off.

PHL switch
 

When PHL switch is on, +12V comes out from pin 4 of the port 1.

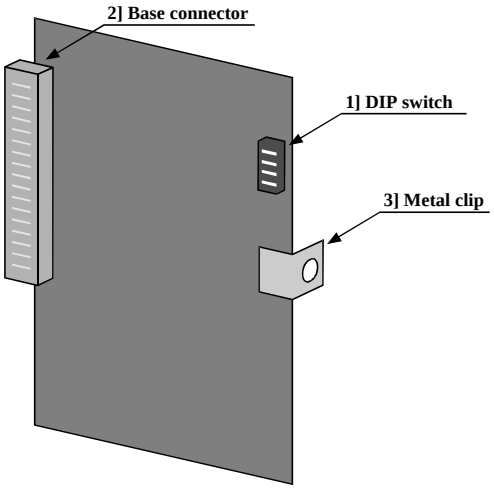
Default setting of DIP switch is all off.
(Port 1 : General purpose port, Port2 : 4800 bps dedicated port)

* For programming or HMI panel communication</td><td></td></tr>
<tr><td>14]
15]</td><td>Optional slot
Binding screw hole</td><td colspan="10">The optional memory board is inserted in this slot.
Install the memory board in the slot by sliding it in along the guide.
• When installing the memory board
First install the CPU module in the base unit, then insert the memory board in the optional slot. After installing the memory board, be sure to fix it securely using the binding screws (supplied with the memory board).
• When removing the memory board
With the memory board still attached to the CPU module, remove the entire CPU module from the base unit

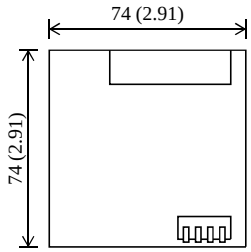
Note on installing the memory board
Do not install the CPU module in the base unit with the memory board still attached to the CPU module. Doing so may cause inappropriate connection at the base connector on the memory board.</td><td>Memory board is supported by EH-CPU308(A)/316(A) /448(A)/516/548.</td></tr>
</table>

4.3 Memory Board

Name and function of each part



Type	EH-MEMP
	EH-MEMD
Weight	Approx. 0.05 kg (0.11 lb.)
Dimensions (mm(in.))	



No.	Name	Function	Remarks
1]	DIP switch	Mode setting.	
2]	Base connector	Connector with base unit.	
3]	Metal clip	Screw bracket for CPU case.	

Memory board specifications

item	EH-MEMP	EH-MEMD
Program memory size	48 kstep max.	16 kstep max.
Data memory size	-	384 kwords
Program copy (CPU→MEMP/D) *1	✓	✓
Program copy (CPU←MEMP/D) *1	✓	✓
Program verify (CPU←→MEMP/D) *1	✓	✓
Data logging *2	-	✓
Memory type	FLASH (100,000 times overwriting max.)	

*1 Selected by dip switch. It is effective only at power on.

*2 EH-MEMD supports data logging (writing), reading and deleting.
When using the EH-MEMD for the first time, the RUN LED will rapidly flash (at 250 ms intervals) for Approx. 3 seconds for initialization of the memory board at power on.

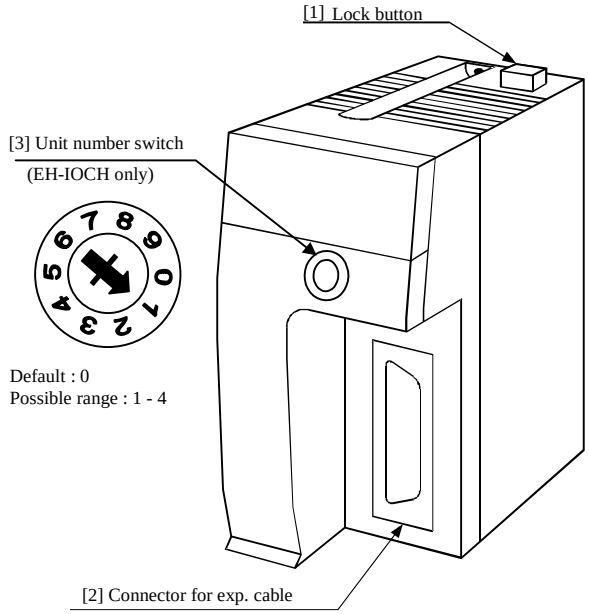
* Since user program is kept in FLASH memory in CPU module, the memory board is not necessary in normal use. The memory board is useful when program copying and data logging.

* Be sure to turn off the power of the PLC when mounting or removing the memory board from the CPU module. When mounting to the CPU module, be sure to fix to the CPU module case by binding screw (M3 × 4) as attached. Additionally, do not install the CPU module in the base unit with the memory board attached. Otherwise, this could be cause of mal function due to bad connection.
When removing, remove whole CPU module at first and then lift out the memory board.

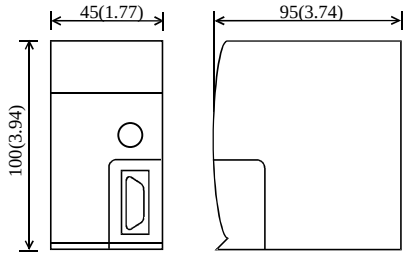
* Program and data are kept without power supplied because of flash memory. Max. data overwriting is approx. 100,000 times.

4.4 I/O controller

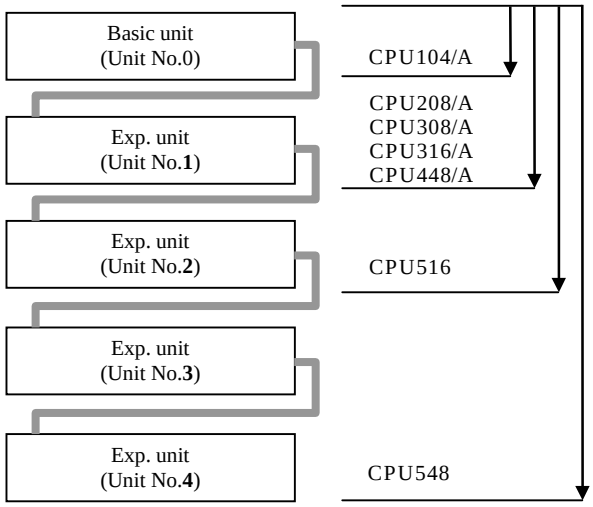
Name and function of each part



Type	EH-IOC
	EH-IOCH
Weight	Approx. 0.14 kg (0.31 lb)
Current consumption	Approx. 30 mA (IOC)
	Approx. 80 mA (IOCH)
Dimensions (mm (ic.))	



No.	Name	Function
[1]	Lock button	When dismounting the module from a base unit, press this button and lift up the module. The module can be fixed firmly by a screw (M4, 10mm)
[2]	Connector for exp. cable	Connector for expansion cable.
[3]	Unit number switch	Rotary switch for unit number. Be sure to set 1 to 4 for expansion bases from CPU side. Note that other unit number than 1 to 4 may cause mal output because of undefined address. Since CPU reads always the switch information, be sure to set after power off. Expandability depends on CPU types as below.

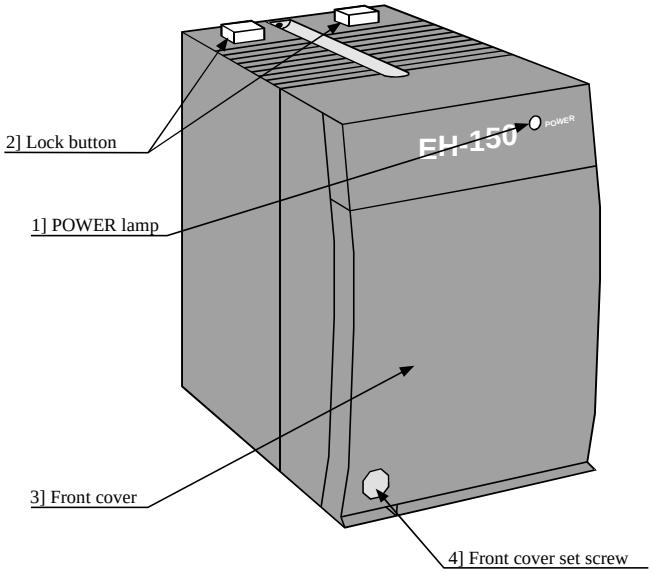


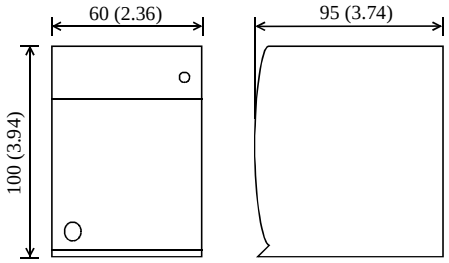
No.	Item	Details	Remarks
Function		The in/output controller sends output signal from CPU module to output modules on the exp. base, and reads input signals from input modules on the exp. base and sends back to the CPU module. This module is mounted on the next right to power supply module. EH-IOC and IOCH cannot be mixed in one system.	EH-CPU104 does not support expansion unit.

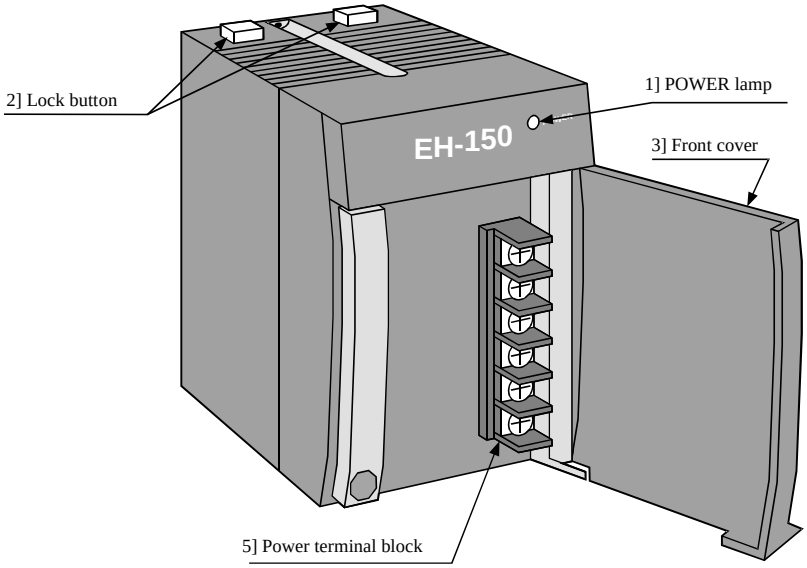
4.5 AC Power Module

Name and function of each part

Type	EH-PSA
Weight	Approx. 0.36 kg (0.79 lb.)
Dimensions (mm (in.))	







No.	Name	Function	Remarks
1]	POWER lamp	This indicates that the AC power is supplied.	Green
2]	Lock button	This is used when removing the power module from a base unit.	
3]	Front cover	This is used when wiring.	
4]	Front cover set screw	This is used to fix the front cover. When fixing is necessary, use M3 × 6 mm (0.24 in.) set screws.	
5]	Power terminal block	This is used when power is supplied externally, for 24 V output wiring, and for the grounding wiring.	

<table>
<tr><th>No.</th><th>Item</th><th>Detailed explanation</th><th>Remarks</th></tr>
<tr><td colspan="2">Operation explanation</td><td>Converts power supplied externally into the power (5 V DC) that can be used inside the EH-150. Also, it prepares power (24 V DC) for driving relays, etc.
100 V to 240 V AC can be used for the external supply voltage.
When using the 24 V DC after conversion, connect it externally.
The operating status can be confirmed with the POWER lamp on the front of the module.
Refer to the specification table for the types of protective functions.</td><td></td></tr>
<tr><td>1]</td><td>POWER lamp</td><td>Lighting : Power is supplied properly.
Off : Power is not supplied or 24VDC output is shorted.
Flashing: Output current over loaded</td><td>The lamp is green.</td></tr>
<tr><td>2]</td><td>Lock button</td><td>When dismounting the module from a base unit, press this button and lift up the module. The module can be fixed firmly by a screw (M4, 10 mm (0.39 in.)).</td><td></td></tr>
<tr><td>3]
4]</td><td>Front cover
Front cover set screws</td><td>Open and close the cover when performing cable wiring. During operation, keep the front cover closed. When opening the cover, shut the power off first to avoid getting an electric shock.
When fixing is necessary, use M3 × 6 mm (0.24 in.) screws for the set screws.</td><td></td></tr>
<tr><td>5]</td><td>Power terminal block</td><td>Upper side
<table>
<tr><td>1</td><td>24 V DC (+) output</td><td rowspan="2">Connect these when using 24 V DC.</td></tr>
<tr><td>2</td><td>24 V DC (GND) output</td></tr>
<tr><td>3</td><td>(Empty)</td><td>Do not connect anything.</td></tr>
<tr><td>4</td><td>AC input</td><td rowspan="2">Connect input power supply.</td></tr>
<tr><td>5</td><td>AC input</td></tr>
<tr><td>6</td><td>FE (functional earth)</td><td>Connect to a class D ground.</td></tr>
</table>
Lower side
The recommended crimp terminal is indicated below.
 

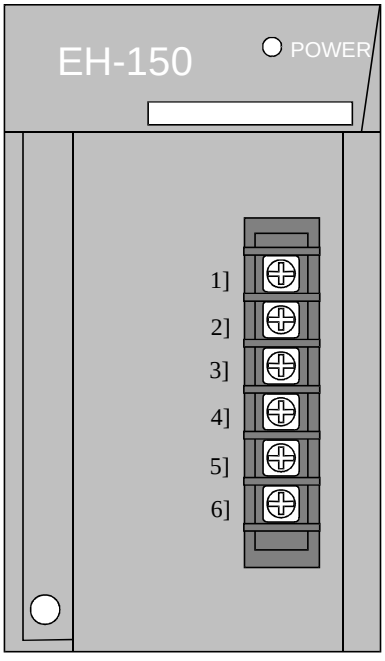
(Recommended)
 

Care must be exercised when handling the terminal since it may fall off if the screw is loose.
Unit: mm (in.)</td><td></td></tr>
</table>

Specification table

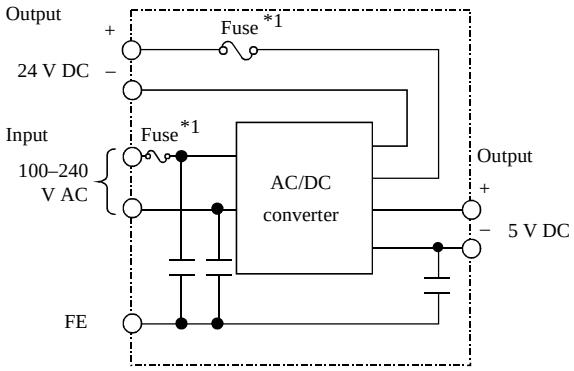
Item	Specifications	
Rated output voltage	5 V DC	24 V DC
Maximum DC output current	3.8 A	0.4 A
Efficiency	65% or more (Load of 5 V 3.8 A, 24 V 0.4 A after energizing for 5 minutes at room temperature and humidity)	
Input voltage range	85 to 264 V AC wide range	
Input current	1 A or less (85 to 264 V AC)	
Input rush current	50 A or less (Ta=25°C), 100 A or less (Ta=55°C)	
Output overcurrent protection	Output short circuit protection	
Instantaneous power failure guarantee	10 ms or more (85 to 100 V AC), 20 ms (Exceed 100 V AC to 264 V AC)	
Input leak current	3.5 mA or less (60 Hz, 264 V AC)	
Operating ambient temperature	0 to 55°C (with rated load)	
Operating ambient humidity	20% to 90% RH (Non condensing)	
Storage ambient temperature	–10 to 75°C	
Storage ambient humidity	10% to 90% RH (Non condensing)	
Dielectric withstand voltage	1 minute at 1,500 V AC between (AC input) and (DC output) 1 minute at 750 V AC between (DC output) and (FE)	
Insulation resistance	20 M Ω or more (500 V DC)	(1) Between AC input and FE (2) Between AC input and DC output
Vibration resistance	Conforms to JIS C 0911	(16.7 Hz double amplitude 3 mm (0.12 in.) X, Y and Z each direction)
	Conforms to JIS C 0040	(10 to 57 Hz single amplitude 0.075 mm) (57 to 150 Hz constant acceleration 9.8 m/s^2)
Shock resistance	Conforms to JIS C 0912	(10 G in X, Y and Z directions)
	Conforms to JIS C 0040	(15 G in X, Y and Z directions)

Terminal layout



No.	Terminal array
1]	24 V DC+
2]	24 V DC–
3]	NC
4]	100–240 V AC
5]	100–240 V AC
6]	FE

Diagram of internal circuit

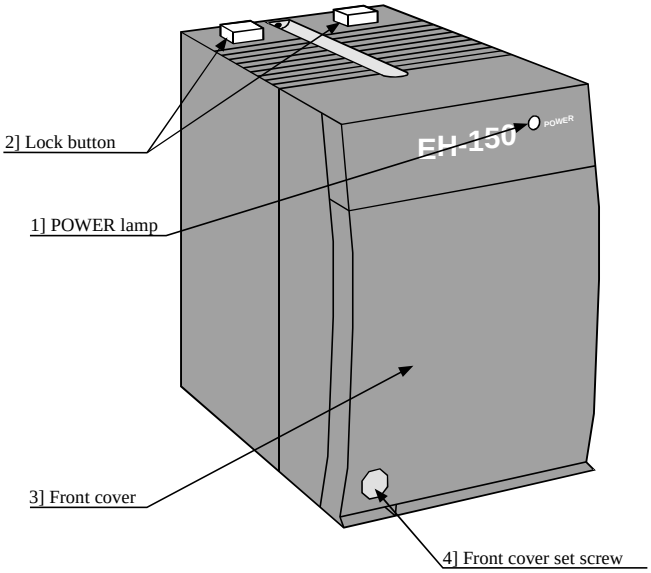


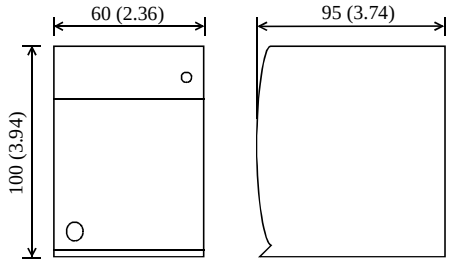
*1: The POWER lamp will not be lit when the fuse is blown. The module must be repaired. The fuse may not be replaced by the end user.

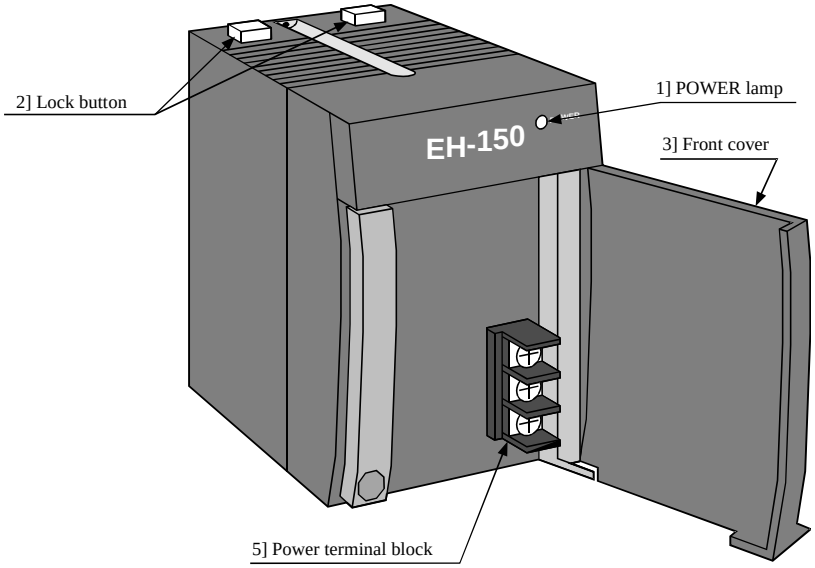
4.6 DC Power Module

Name and function of each part

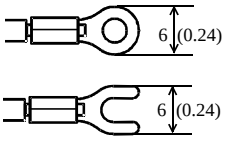
Type	EH-PSD
Weight	Approx. 0.28 kg (0.62 lb.)
Dimensions (mm (in.))	







No.	Name	Function	Remarks
1]	POWER lamp	This indicates that the DC power is supplied.	Green
2]	Lock button	This is used when removing the power module from a base unit.	
3]	Front cover	This is used when wiring.	
4]	Front cover set screw	This is used to fix the front cover. When fixing is necessary, use M3 × 6 mm (0.24 in.) set screws.	
5]	Power terminal block	This is used when power is supplied externally, and for the grounding wiring.	

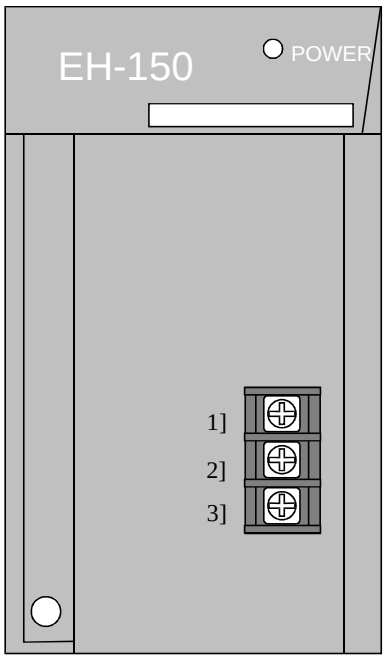
<table>
<tr><th>No.</th><th>Item</th><th>Detailed explanation</th><th>Remarks</th></tr>
<tr><td colspan="2">Operation explanation</td><td>Converts 24 V DC power supplied externally into the power (5 V DC) that can be used inside the EH-150.
21.6 V to 26.4 V DC can be used for the external power voltage.
Connect the power supply (24 V DC) to drive relays, etc., directly with the external power supply.
The operating status can be confirmed with the POWER lamp on the front of the module.</td><td></td></tr>
<tr><td>1]</td><td>POWER lamp</td><td>Lighting : Power is supplied properly.
Off : Power is not supplied or 24VDC output is shorted or over voltage.</td><td>The lamp is green.</td></tr>
<tr><td>2]</td><td>Lock button</td><td>When dismounting the module from a base unit, press this button and lift up the module. The module can be fixed firmly by a screw (M4, 10 mm (0.39 in.)).</td><td></td></tr>
<tr><td>3]
4]</td><td>Front cover
Front cover set screws</td><td>Open and close the cover when performing cable wiring. During operation, keep the front cover closed. When opening the cover, shut the power off first to avoid getting an electric shock.
When fixing is necessary, use M3 × 6 mm (0.24 in.) screws for the set screws.</td><td></td></tr>
<tr><td>5]</td><td>Power terminal block</td><td>Upper side
<table>
<tr><td>1</td><td>24 V DC (+) input</td><td rowspan="2">Connect input power supply.</td></tr>
<tr><td>2</td><td>24 V DC (–) input</td></tr>
<tr><td>3</td><td>FE (functional earth)</td><td>Connect to a class D ground
In order to comply with the CE marking, connect to 24 V DC (–). *1</td></tr>
</table>
Lower side
The recommended crimp terminal is indicated below.
 

(Recommended)
Care must be exercised when handling the terminal since it may fall off if the screw is loose.
Unit: mm (in.)</td><td>*1:When performing insulation resistance measurements or dielectric withstand voltage measurements, always remove the connection between the FE and 24 VDC (–).</td></tr>
</table>

Specification table

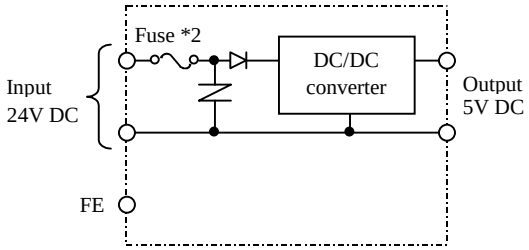
Item	Specifications
Rated output voltage	5 V DC
Maximum DC output current	3.8 A
Efficiency	70% or more (Load of 5 V DC 3.8 A)
Input voltage range	21.6 to 26.4 V DC
Input current	1.25 A or less (with 24 V DC)
Input rush current	50 A or less (Ta=25°C), 100 A or less (Ta=55°C)
Output overcurrent protection	Output short circuit protection
Instantaneous power failure guarantee	1 ms or more (21.6 to 26.4 V DC)
Operating ambient temperature	0 to 55°C (with rated load)
Operating ambient humidity	20% to 90% RH (Non condensing)
Storage ambient temperature	–10 to 75°C
Storage ambient humidity	10% to 90% RH (Non condensing)
Dielectric withstand voltage	1 minute at 1,500 V AC between DC input and FE
Insulation resistance	20 M Ω or more (500 V DC) (Between DC input and FE)
Insulation type	No insulation

Terminal layout



No.	Terminal layout
1]	24 V DC (+)
2]	24 V DC (–)
3]	FE

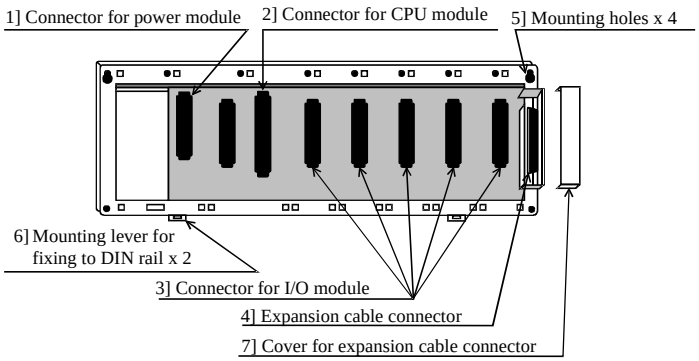
Diagram of internal circuit



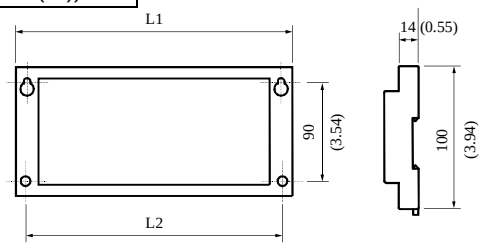
*2: The POWER lamp will not light when the fuse is blown. The module must be repaired. The fuse may not be replaced by end users.

4.7 Base Unit

Name and function of each part



Type (Weight)	EH-BS3(A) approx. 0.22kg (0.48 lb)
	EH-BS5(A) approx. 0.28 kg (0.62 lb.)
	EH-BS8(A) approx. 0.36 kg (0.79 lb.)
	EH-BS11(A) approx. 0.4 kg (0.88 lb.)
Current consumption	Approx. 200 mA
Dimensions (mm (in.))	



	L1 (Outer dimensions)	L2 (Mounted dimensions)
EH-BS3(A)	222.5 (8.76)	207 (8.15)
EH-BS5(A)	282.5 (11.12)	267 (10.51)
EH-BS8(A)	372.5 (14.67)	357 (14.06)
EH-BS11(A)	462.5 (18.21)	447 (17.60)

Communication slot (special slot for communication module)

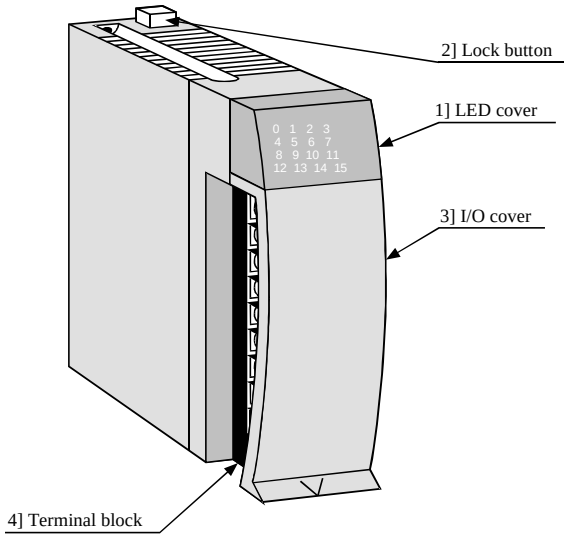
EH-BS3, 5, 8	Slot 0-2
EH-BS3A	Slot 0-2
EH-BS5A	Slot 0-4
EH-BS8A	Slot 0-7
EH-BS11A	Slot 0-7 (Not allowed on slot 8, 9, A)

No.	Name	Function		Remarks
1]	Connector for power module	This is the connector for loading the power module.		
2]	Connector for CPU module	This is the connector for loading the CPU module. When the unit is used as an expansion base, this becomes the connector for loading the I/O controller.		
3]	Connector for I/O module	This is the connector for loading the I/O module.		
		Type	Number of I/O modules to be mounted	
		EH-BS3(A)	3	
		EH-BS5(A)	5	
		EH-BS8(A)	8	
		EH-BS11A	11	
4]	Expansion cable connector	Connector for the expansion cable.		EH-CPU104(A) is not support expansion unit.
5]	Mounting holes (4 points)	These are used when the base unit is attached to a panel, etc. Use M4 × 20 mm (0.79 in.) screws.		
6]	Mounting hook for DIN rail	This is used when attaching the unit to a DIN rail.		
7]	Cover for expansion cable connector	This cover is used for protecting the expansion cable connector when it is not used.		

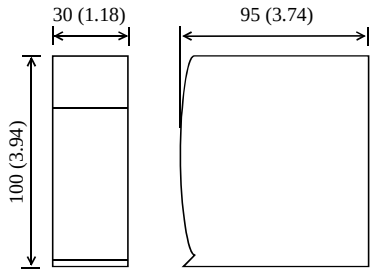
No.	Item	Details
Operation description		This is the basic unit for loading all the modules. Using the base unit, power is supplied from the power module to each of the other modules. Also, signals are sent to each module from the CPU module or the I/O controller. Select the base unit according to the number of I/O modules that are used.

4.8 Input Module

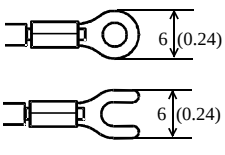
Name and function of each part



Type	EH-XD8
	EH-XD16, XDL16
	EH-XA16
	EH-XAH16
Weight	Approx. 0.16 kg (0.35 lb.) (EH-XD**) Approx. 0.18 kg (0.4 lb.) (EH-XA**)
Dimensions (mm (in.))	



No.	Name	Function	Remarks
1]	LED cover	This is the cover for the LED that displays the input status. When the input signal turns on, the LED for the relevant number lights up. The LED only lights when the module is energized.	
2]	Lock button	When dismounting the module from a base unit, press this button and lift up the module. The module can be fixed firmly by a screw (M4, 10 mm (0.39 in.)).	
3]	I/O cover	This is the cover attached to the terminal block area	
4]	Terminal block	This is the terminal block for connecting input signals. The terminal block is removable.	

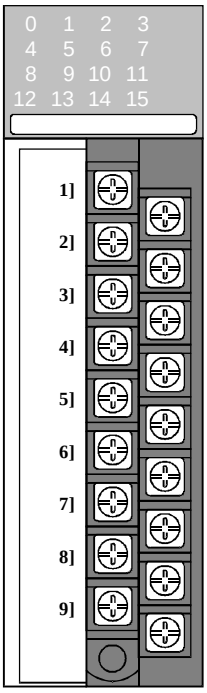
No.	Item	Detailed explanation	Remarks
Operation explanation		This module inputs digital signals. The LED's light up while the corresponding digital signal is ON. The CPU reads the status of the mounted modules. If the status accords with actual I/O configuration in the user program, the CPU inputs digital signals according to the user program.	
Terminal block		The screws for the terminal block are M3 screws. Use a crimp terminal that fits the screw diameter. The maximum thickness of the cable should be only up to 0.75 mm^2. (If you attach two crimp terminals to the same terminal, use a cable with the thickness of 0.5 mm^2.) The recommended crimp terminal is indicated below.  6 (0.24) (Recommended) 6 (0.24) Care must be exercised when handling the terminal since it may fall off if the screw is loose. Unit: mm (in.)	

Specification table

Item		Specification		
Type		EH-XD8	EH-XD16	EH-XDL16
Input specification		DC input		
Input voltage		24 V DC		
Allowable input voltage range		19.2 to 30 V DC		
Input impedance		Approx. 3.5 k Ω	Approx. 5.9 k Ω	
Input current		Approx. 6.9 mA	Approx. 4.0 mA	
Operating voltage	ON voltage	15 V or more		
	OFF voltage	5 V or less		
Input lag	OFF → ON	5 ms or less (4 ms TYP)		16 ms or less (13 ms TYP)
	ON → OFF	5 ms or less (4 ms TYP)		16 ms or less (13 ms TYP)
Number of input points / module		8 points / module	16 points / module	
Number of input points / common		8 points / common (2 common terminals *)	16 points / common (2 common terminals *)	
Polarity		None		
Insulation system		Photocoupler insulation		
Input display		LED (green)		
External connection		Removable type screw terminal block (M3)		
Internal current consumption (5 V DC)		Approx. 30 mA	Approx. 50 mA	
I/O assignment		X16		

* Common terminals are connected internally.

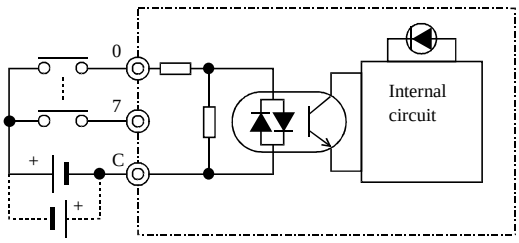
Terminal layout



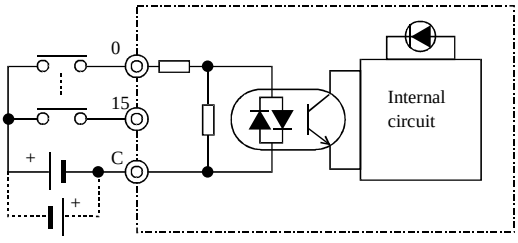
No.	Signal name	
	EH-XD8	EH-XD(L)16
1]	0	0
2]	1	1
3]	2	2
4]	3	3
5]	4	4
6]	5	5
7]	6	6
8]	7	7
9]	C	C
10]	NC	8
11]	NC	9
12]	NC	10
13]	NC	11
14]	NC	12
15]	NC	13
16]	NC	14
17]	NC	15
18]	C	C

Diagram of internal circuit

EH-XD8



EH-XD(L)16

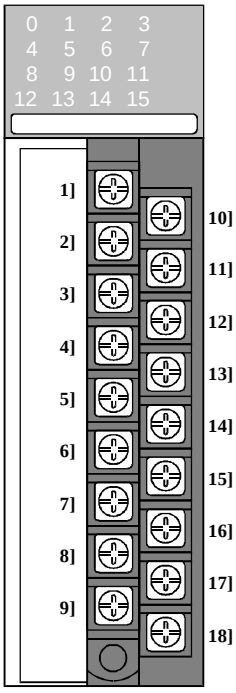


Specification table

Item		Specification	
Type		EH-XA16	EH-XAH16
Input specification		AC input	
Input voltage		100 to 120 V AC	200 to 240 V AC
Allowable input voltage range		85 to 132 V AC	170 to 264 V AC
Input impedance		Approx. 16 k Ω (50 Hz)/ Approx. 13 k Ω (60 Hz)	Approx. 32 k Ω (50 Hz)/ Approx. 27 k Ω (60 Hz)
Input current		4.8 to 7.6 mA (100 V AC/50 Hz)	4.3 to 8.0 mA (200 V AC/50 Hz)
Operating voltage	ON voltage	79 V AC	164 V AC
	OFF voltage	20 V AC	40 V AC
Input lag	OFF → ON	15 ms or less	
	ON → OFF	25 ms or less	
Number of input points / module		16 points / module	
Number of input points / common		16 points / common (2 common terminals *)	
Polarity		None	
Insulation system		Photocoupler insulation	
Input display		LED (green)	
External connection		Removable type screw terminal block (M3)	
Internal current consumption (5 V DC)		Approx. 50 mA	
I/O assignment		X16	

* Common terminals are connected internally.

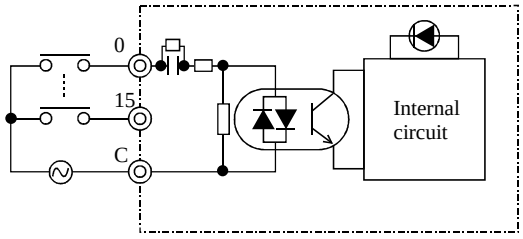
Terminal layout



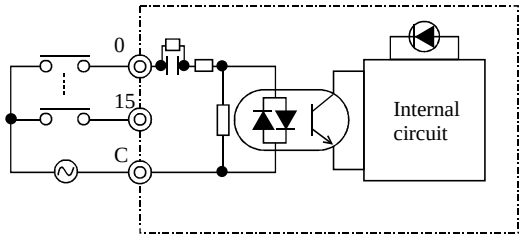
No.	Signal name	
	EH-XA16	EH-XAH16
1]	0	0
2]	1	1
3]	2	2
4]	3	3
5]	4	4
6]	5	5
7]	6	6
8]	7	7
9]	C	C
10]	8	8
11]	9	9
12]	10	10
13]	11	11
14]	12	12
15]	13	13
16]	14	14
17]	15	15
18]	C	C

Diagram of internal circuit

EH-XA16

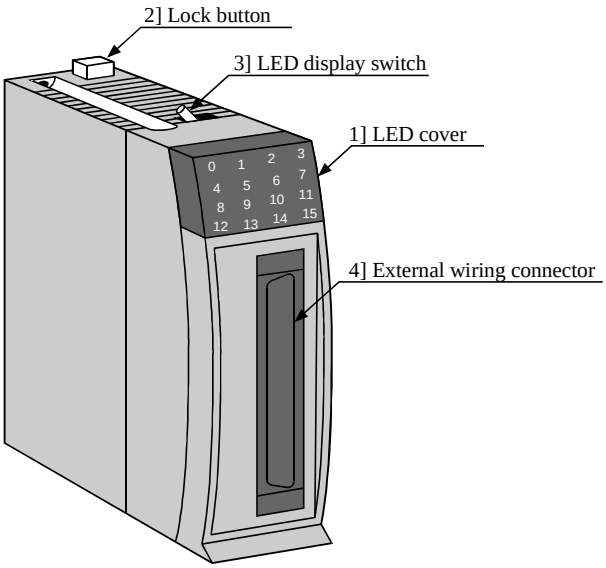


EH-XAH16

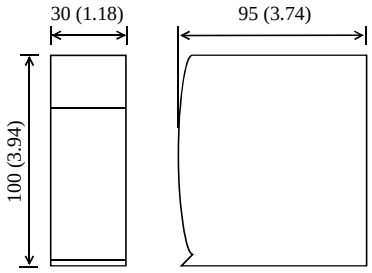


4.9 32-point Input Module

Name and function of each part



Type	EH-XD32
Weight	Approx. 0.12 kg (0.26 lb.)
Dimensions (mm (in.))	



No.	Name	Function	Remarks
1]	LED cover	This is the cover for the LED that displays the input status. When the input signal turns on, the LED for the relevant number lights up. The LED only lights when the module is energized.	
2]	Lock button	When dismounting the module from a base unit, press this button and lift up the module. The module can be fixed firmly by a screw (M4, 10 mm (0.39 in.)).	
3]	LED indication switch	Effective LED group (16 points) is selected by this switch as follows.	
4]	External wiring connector – Caution – • Approx. 120 mm (4.72 in.) of space will be required in the front of the module for the connector and cable. Please choose the installation location (space) accordingly. • Use a shielded cable and always use a Class D grounding.	Connector for input signals. Special cable with connector is optional. (Refer to chapter 4.31, 4.32.) If separate connector is needed, applicable connector type is below. Manufacturer 1 : Fujitsu Takamisawa Solder type Socket: FCN-361J040-AU Cover: FCN-360C040-E Crimp type Housing: FCN-363J040 Contact: FCN-363J-AU Cover: FCN-360C040-E Crimping tool : FCN-367J040-AU/F Manufacturer 2 : AMP Solder type : 1473381-1	•

SW	LED +16	Indication group
OFF	OFF	0 to 15
ON	ON	16 to 31

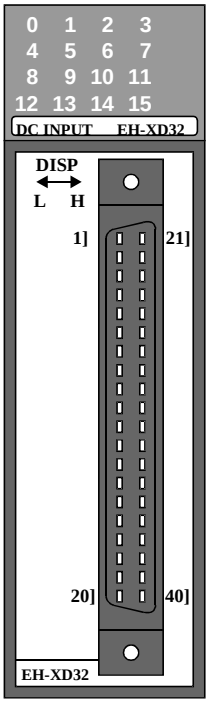
Specification table

Item		Specification
Type		EH-XD32
Input specification		DC input
Input voltage		24 V DC
Allowable input voltage range		19.2 to 30 V DC
Input impedance		Approx. 5.6 k Ω
Input current		Approx. 4.3 mA
Operating voltage	ON voltage	15 V or more
	OFF voltage	5 V or less
Input lag	OFF → ON	5 ms or less
	ON → OFF	5 ms or less
Number of input points / module		32 points / module
Number of input points / common		32 points / common (4 common terminals *1)
Polarity		None
Insulation system		Photocoupler insulation
Input display		LED (green) *2
External connection		Connector
Internal current consumption (5 V DC)		Approx. 60 mA
I/O assignment		X32

*1: Common terminals are connected internally.

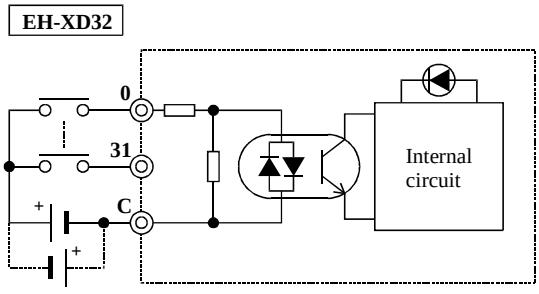
*2: There are 16 points of LED indication. The indication group is switched by toggle switch.

Terminal layout



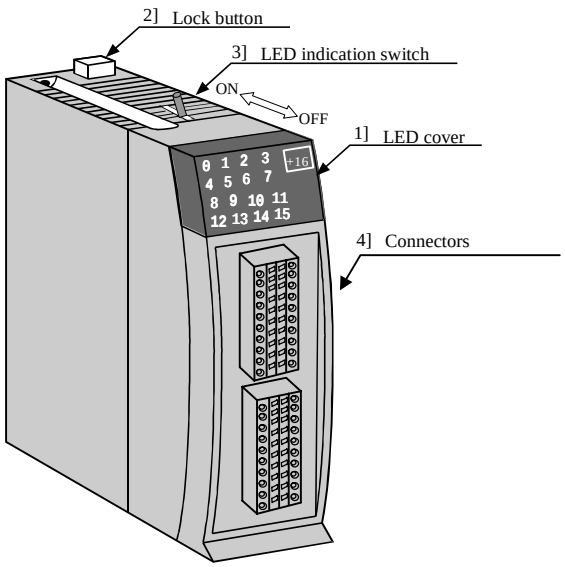
No.	Signal name	No.	Signal name
1]	0	21]	16
2]	1	22]	17
3]	2	23]	18
4]	3	24]	19
5]	4	25]	20
6]	5	26]	21
7]	6	27]	22
8]	7	28]	23
9]	C	29]	C
10]	8	30]	24
11]	9	31]	25
12]	10	32]	26
13]	11	33]	27
14]	12	34]	28
15]	13	35]	29
16]	14	36]	30
17]	15	37]	31
18]	C	38]	C
19]	NC	39]	NC
20]	NC	40]	NC

Diagram of internal circuit

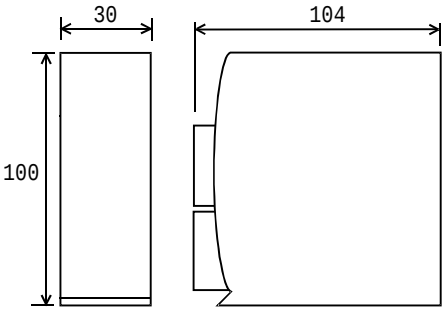


4.10 Euro-terminal 32 points input module

Name of each part



Type	EH-XD32E
	EH-XDL32E
Weight	0.15 kg (033 lb.)
Dimension (mm(in.))	



No.	Name	Function	Remarks
1]	LED cover	This is the cover for the LED that displays the input status and displayed group. When the input signal turns on, the LED for the relevant number lights up. The LED only lights when the module is energized.	
2]	Lock button	When dismounting the module from a base unit, press this button and lift up the module. The module can be fixed firmly by a screw (M4, 10 mm (0.39 in.)).	
3]	LED display switch	Effective LED group (16 points) is selected by this switch as follows.	
4]	External wiring connector	A connector for input signals. It is attached to the product. Solid wire 0.5…1.0 mm^2 Flexible wire 0.5…1.0 mm^2 Insulation stripping length 7mm Cable ferrules cannot be used	Applicable connectors (Manufacturer: Weidmuller) • Type B2L3.5/20AUOR • Pats No. 175736 • Poles 20 • Color Orange

SW	LED +16	Display group
OFF	OFF	0 to 15
ON	ON	16 to 31

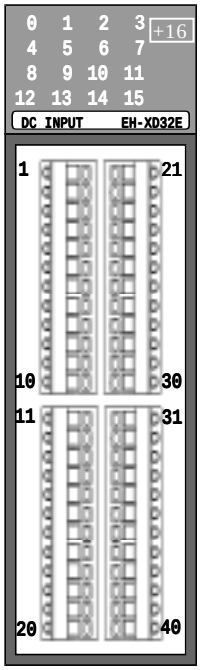
Specification table

Item		Specification	
Type		EH-XD32E	EH-XDL32E
Input specification		DC input	
Input voltage		24 V DC	
Allowable input voltage range		20.4 to 28.8 V DC	
Input impedance		Approx. 5.6 k Ω	
Input current		Approx. 4.3 mA	
Operating voltage	ON voltage	15 V or more	
	OFF voltage	5 V or less	
Input lag	OFF → ON	1 ms or less	16 ms or less
	ON → OFF	1 ms or less	16 ms or less
Number of input points / module		32 points / module	
Number of input points / common		8 points / common (2 common terminals per group *1)	
Polarity		None	
Insulation system		Photocoupler insulation	
Input display		LED (green) *2	
External connection		Spring type connector (removable)	
Internal current consumption (5 V DC)		Approx. 60 mA	
I/O assignment		X32	

*1: 4 groups (C1-C4) are separated. 2 common terminals in one group are connected internally.

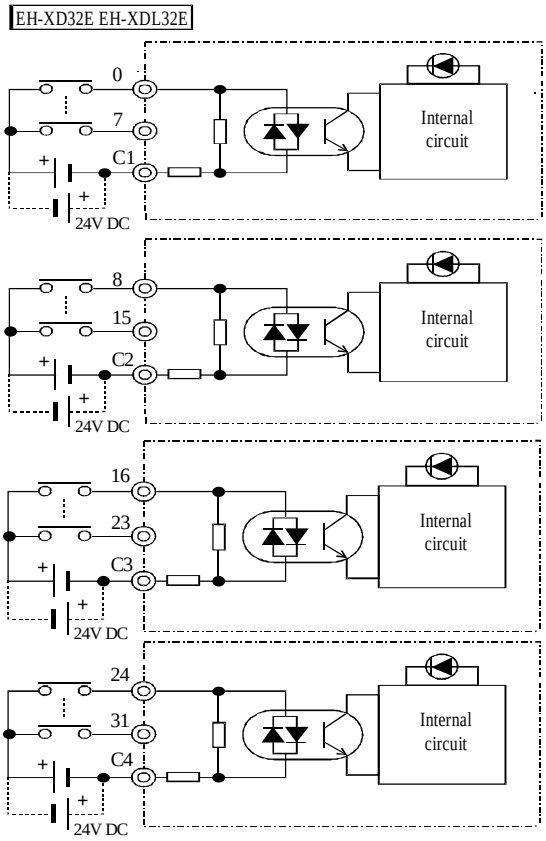
*2: There are 16 points of LED indication. The indication group is switched by toggle switch.

Terminal layout



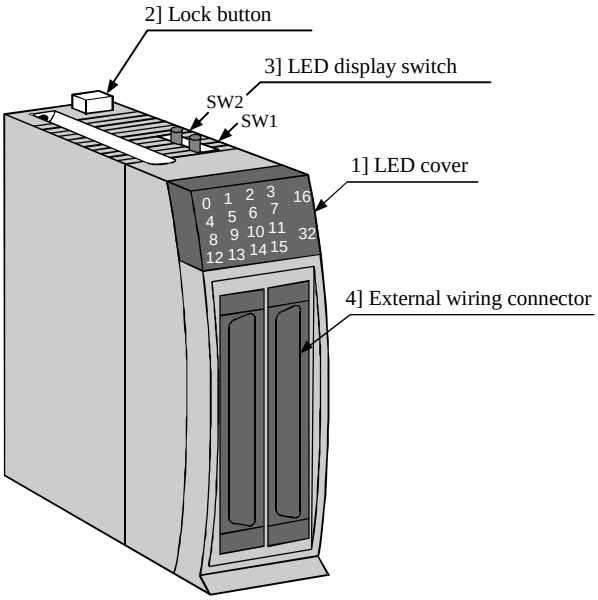
No.	Signal name	No.	Signal name
(1)	0	(21)	16
(2)	1	(22)	17
(3)	2	(23)	18
(4)	3	(24)	19
(5)	4	(25)	20
(6)	5	(26)	21
(7)	6	(27)	22
(8)	7	(28)	23
(9)	C1	(29)	C3
(10)	C1	(30)	C3
(11)	8	(31)	24
(12)	9	(32)	25
(13)	10	(33)	26
(14)	11	(34)	27
(15)	12	(35)	28
(16)	13	(36)	29
(17)	14	(37)	30
(18)	15	(38)	31
(19)	C2	(39)	C4
(20)	C2	(40)	C4

Diagram of internal circuit

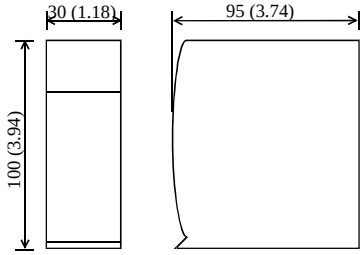


4.11 64-point Input Module

Name and function of each part



Type	EH-XD64
Weight	Approx. 0.14 kg (0.31 lb.)
Dimensions (mm (in.))	



No.	Name	Function	Remarks
1]	LED cover	This is the cover for the LED that displays the input status and displayed group. When the input signal turns on, the LED for the relevant number lights up. The LED only lights when the module is energized.	
2]	Lock button	When dismounting the module from a base unit, press this button and lift up the module. The module can be fixed firmly by a screw (M4, 10 mm (0.39 in.)).	
3]	LED display switch	Effective LED group (16 points) is selected by these switches as follows.	
4]	External wiring connector – Caution – • Approx. 120 mm (4.72 in.) of space will be required in the front of the module for the connector and cable. Please choose the installation location (space) accordingly. • Use a shielded cable and always use a Class D grounding.	Connector for input signals. Special cable with connector is optional. (Refer to chapter 4.31, 4.32.) If separate connector is needed, applicable connector type is below. Manufacturer 1 : Fujitsu Takamisawa Solder type Socket: FCN-361J040-AU Cover: FCN-360C040-E Crimp type Housing: FCN-363J040 Contact: FCN-363J-AU Cover: FCN-360C040-E Crimping tool : FCN-367J040-AU/F Manufacturer 2 : AMP Solder type : 1473381-1	

SW1	SW2	LED16	LED32	Display group
OFF	OFF	Not lit	Not lit	0 to 15
ON	OFF	Lit	Not lit	16 to 31
OFF	ON	Not lit	Lit	32 to 47
ON	ON	Lit	Lit	48 to 63

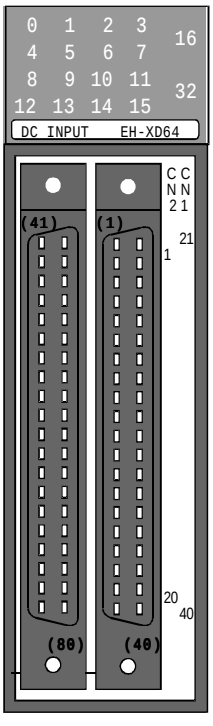
Specification table

Item		Specification
Type		EH-XD64
Input specification		DC input
Input voltage		24 V DC
Allowable input voltage range		20.4 to 28.8 V DC
Input impedance		Approx. 5.6 k Ω
Input derating		See the derating diagram.
Input current		Approx. 4.3 mA
Operating voltage	ON voltage	15 V or more
	OFF voltage	5 V or less
Input lag	OFF → ON	1 ms or less
	ON → OFF	1 ms or less
Number of input points / module		64 points / module
Number of input points / common		32 points / common (4 common terminals per group *1)
Polarity		None
Insulation system		Photocoupler insulation
Input display		LED (green) *2
External connection		Connector
Internal current consumption (5 V DC)		Approx. 80 mA
I/O assignment		X64

*1: 2 groups (C1,C2) are separated. 4 common terminals in one group are connected internally.

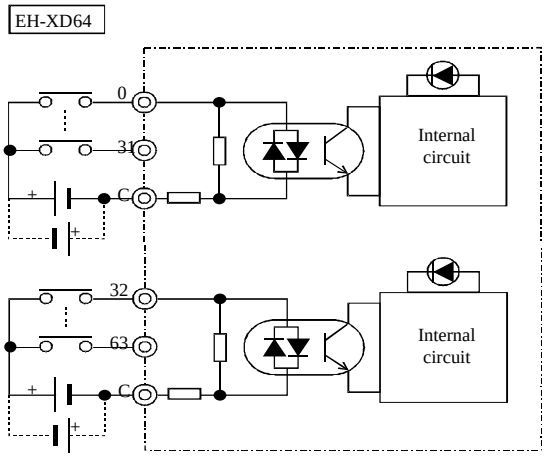
*2: There are 16 points of LED indication. The indication group is switched by toggle switch.

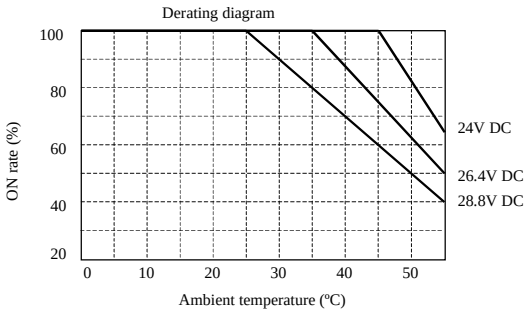
Terminal layout



Left (CN2)				Right (CN1)			
No.	Signal name	No.	Signal name	No.	Signal name	No.	Signal name
(41)	32	(61)	48	(1)	0	(21)	16
(42)	33	(62)	49	(2)	1	(22)	17
(43)	34	(63)	50	(3)	2	(23)	18
(44)	35	(64)	51	(4)	3	(24)	19
(45)	36	(65)	52	(5)	4	(25)	20
(46)	37	(66)	53	(6)	5	(26)	21
(47)	38	(67)	54	(7)	6	(27)	22
(48)	39	(68)	55	(8)	7	(28)	23
(49)	C2	(69)	C2	(9)	C1	(29)	C1
(50)	40	(70)	56	(10)	8	(30)	24
(51)	41	(71)	57	(11)	9	(31)	25
(52)	42	(72)	58	(12)	10	(32)	26
(53)	43	(73)	59	(13)	11	(33)	27
(54)	44	(74)	60	(14)	12	(34)	28
(55)	45	(75)	61	(15)	13	(35)	29
(56)	46	(76)	62	(16)	14	(36)	30
(57)	47	(77)	63	(17)	15	(37)	31
(58)	C2	(78)	C2	(18)	C1	(38)	C1
(59)	NC	(79)	NC	(19)	NC	(39)	NC
(60)	NC	(80)	NC	(20)	NC	(40)	NC

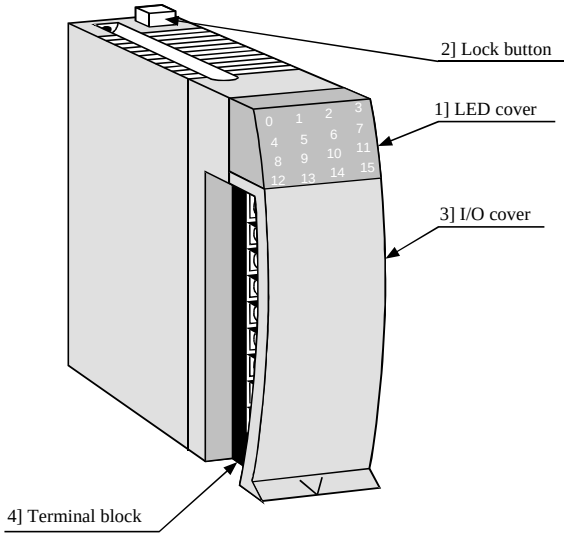
Diagram of internal circuit





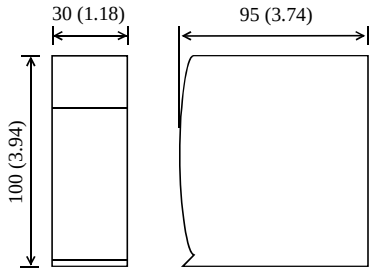
4.12 Output Module

Name and function of each part

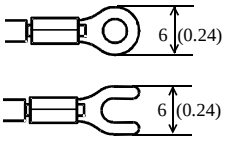


Type (Weight)	EH-YT8, EH-YTP8 (0.16 kg, 0.35 lb)
	EH-YT16, EH-YTP16 (0.16 kg, 0.35 lb)
	EH-YTP16S (0.16 kg, 0.35 lb)
	EH-YR8B (0.16 kg, 0.35 lb)
	EH-YR12 (0.20 kg, 0.44 lb)
	EH-YR16 (0.24 kg, 0.53 lb)
	EH-YS4 (0.18 kg, 0.40 lb)
	EH-YS16 (0.23 kg, 0.51 lb)

Dimensions (mm (in.))



No.	Name	Function	Remarks
1]	LED cover	This is the cover for the LED that displays the output status. When the output signal turns on, the LED for the relevant number lights up. The LED only works when the module is energized.	
2]	Lock button	When dismounting the module from a base unit, press this button and lift up the module. The module can be fixed firmly by a screw (M4, 10 mm (0.39 in.)).	
3]	I/O cover	This is the cover attached to the terminal block area.	
4]	Terminal block	This is the terminal block for connecting output signals. The terminal block is removable.	

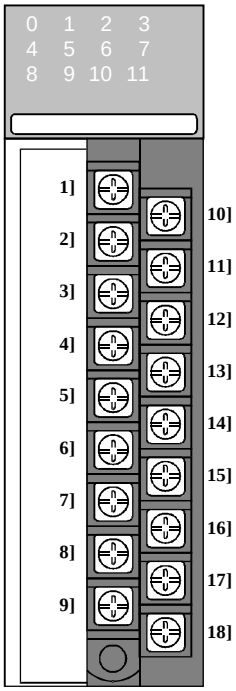
No.	Item	Detailed explanation	Remarks
Operation explanation		This module outputs digital signals. The LED's light up while the corresponding digital signal is ON. The CPU reads the status of the mounted modules. If the status accords with actual I/O configuration in the user program, the CPU outputs digital signals according to the user program.	
Terminal block		The screws for the terminal block are M3 screws. Use a crimp terminal that fits the screw diameter. The maximum thickness of the cable should be only up to 0.75 mm^2. (If you attach two crimp terminals to the same terminal, use a cable with the thickness of 0.5 mm^2.) The recommended crimp terminal is indicated below.  (Recommended) Care must be exercised when handling the terminal since it may fall off if the screw is loose. Unit: mm (in.)	

Specification table (relay output module)

Item		Specification		
Type		EH-YR8B	EH-YR12	EH-YR16
Output specification		Relay output		
Rated load voltage		100/240 V AC, 24 V DC		
Minimum switching current		1 mA		
Leak current		None		
Maximum load current	1 circuit	2 A		
	1 common	2 A	5 A	8A
Output response time	OFF → ON	10 ms or less		
	ON → OFF	10 ms or less		
Number of output points / module		8 points / module	12 points / module	16 points / module
Number of output points / common		1 point / common (channel separated)	12 points / common (2 common terminals) *1	16 points / common (2 common terminals) *1
Surge removal circuit		Varistor (voltage characteristic of varistor : 423 - 517V)	None	
Fuse		None		
Insulation system		Relay insulation	Photocoupler insulation	Relay insulation
Output display		LED (green)		
External connection		Removable type screw terminal block (M3)		
Internal current consumption (5 V DC)		Approx. 220 mA	Approx. 40 mA	Approx. 430 mA
Externally supplied power *2 (For driving relays)		Not used	24 V DC (+10%, -5%) (maximum 70 mA)	Not used
I/O assignment		Y16		

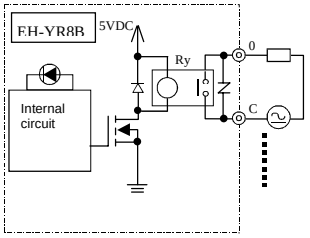
*1: The common terminals are connected internally.
*2: 24 V DC must be supplied externally to drive the relays. (The 24 V output of the power module can be used.)

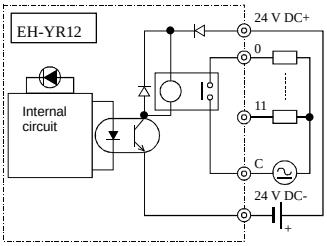
Terminal layout

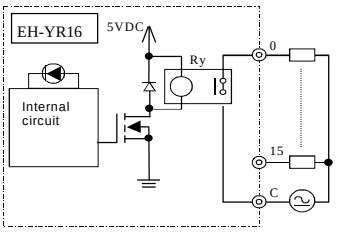


No.	Signal name		
	EH-YR8B	EH-YR12	EH-YR16
1]	0	24 V DC+	0
2]	1	NC	1
3]	2	0	2
4]	3	1	3
5]	4	2	4
6]	5	3	5
7]	6	4	6
8]	7	5	7
9]	NC	C	C
10]	C0	24V DC-	8
11]	C1	NC	9
12]	C2	6	10
13]	C3	7	11
14]	C4	8	12
15]	C5	9	13
16]	C6	10	14
17]	C7	11	15
18]	NC	C	C

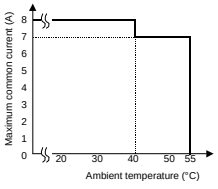
Diagram of internal circuit







EH-YR16 Derating table

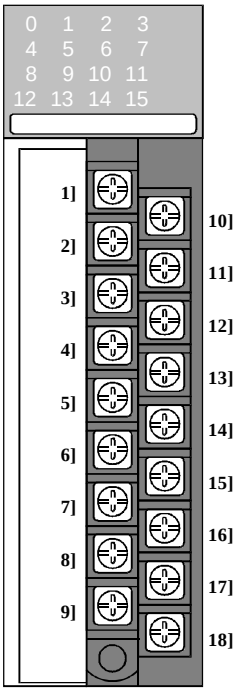


Specification table (transistor output module)

Item		Specification	
Type		EH-YT8	EH-YT16
Output specification		Transistor output (sink type)	
Rated load voltage		12/24 V DC (+10%, -15%)	
Minimum switching current		1 mA	
Leak current		0.1 mA	
Maximum load current	1 circuit	0.3 A (MFG No.02F** or older) *1 0.5 A (MFG No.02G** or newer) *1	
	1 common	2.4 A	4 A
Output response time	OFF → ON	0.3 ms or less	
	ON → OFF	1 ms or less	
Number of output points / module		8 points / module	16 points / module
Number of output points / common		8 points / common	16 points / common
Surge removal circuit		Diode	
Fuse *2		4 A/common	8 A/common
Insulation system		Photocoupler insulation	
Output display		LED (green)	
External connection		Removable type screw terminal block (M3)	
Internal current consumption (5 V DC)		Approx. 30 mA	Approx. 50 mA
External power supply *3 (For supplying power to the S terminal)		12/24 V DC (+10%, –15%) (maximum 30 mA) *4	
I/O assignment		Y16	

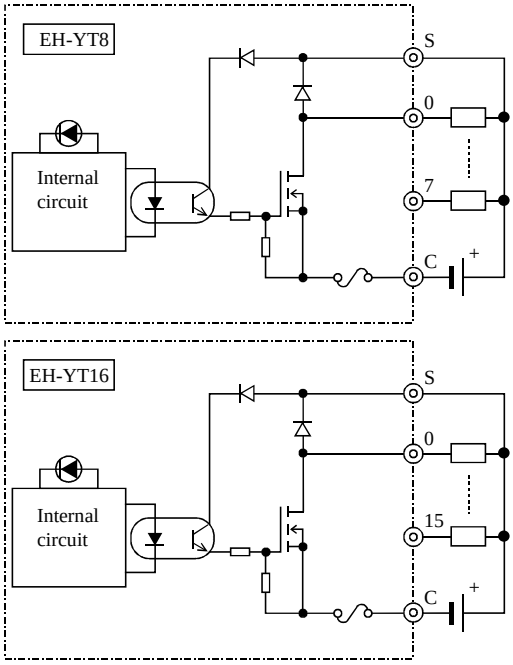
*1: MFG No.02F** : Jun.2002 production, MFG No.02G** : Jul.2002 production
*2: The module needs to be repaired in case of fuse blown. It is not allowed for uses to replace the fuse.
*3: It is necessary to supply 12/24 V DC to the S terminal.
*4: This value is internal current consumption of the module. Additional current is necessary to drive other devices.

Terminal layout



No.	Signal name	
	EH-YT8	EH-YT16
1]	0	0
2]	1	1
3]	2	2
4]	3	3
5]	4	4
6]	5	5
7]	6	6
8]	7	7
9]	C	C
10]	NC	8
11]	NC	9
12]	NC	10
13]	NC	11
14]	NC	12
15]	NC	13
16]	NC	14
17]	NC	15
18]	S	S

Diagram of internal circuit

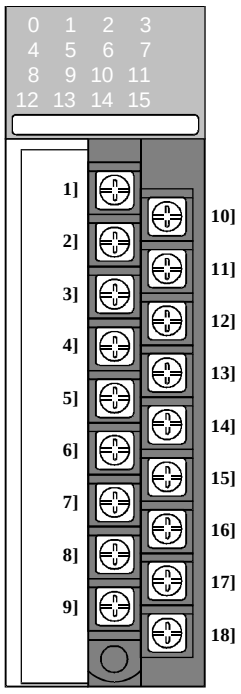


Specification table (transistor output module)

Item		Specification		
Type		EH-YTP8	EH-YTP16	EH-YTP16S
Output specification		Transistor output (source type)		
Rated load voltage		12/24 V DC (+10%, -15%)		
Minimum switching current		1 mA		
Leak current		0.1 mA		
Maximum load current	1 circuit	0.3 A (MFG No.02F** or older) *1 0.5 A (MFG No.02G** or newer) *1		0.8 A
	1 common	2.4 A	4 A	5 A
Output response time	OFF → ON	0.3 ms or less		
	ON → OFF	1 ms or less		
Number of output points / module		8 points / module	16 points / module	
Number of output points / common		8 points / common	16 points / common	
Surge removal circuit		Diode		Built in
Fuse *2		4 A/common	8 A/common	None
Insulation system		Photocoupler insulation		
Output display		LED (green)		
External connection		Removable type screw terminal block (M3)		
Internal current consumption (5 V DC)		Approx. 30 mA	Approx. 50 mA	
External power supply *3 (For supplying power to the S terminal)		12/24 V DC (+10%, -15%) (maximum 30 mA) *4		
I/O assignment		Y16		
Short-circuit protection function		Not available		Available

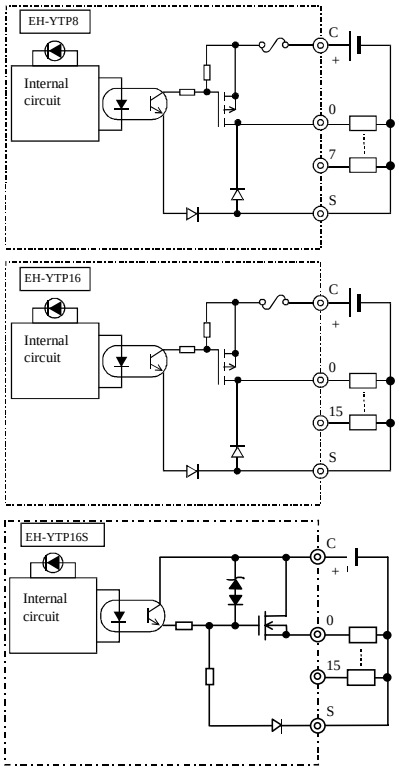
*1: MFG No.02F** : Jun.2002 production, MFG No.02G** : Jul.2002 production
*2: The module needs to be repaired in case of fuse blown. It is not allowed for uses to replace the fuse.
*3: It is necessary to supply 12/24 V DC to the S terminal.
*4: This value is internal current consumption of the module. Additional current is necessary to drive other devices.

Terminal layout



No.	Signal name	
	EH-YTP8	EH-YTP16, YTP16S
1]	0	0
2]	1	1
3]	2	2
4]	3	3
5]	4	4
6]	5	5
7]	6	6
8]	7	7
9]	C	C
10]	NC	8
11]	NC	9
12]	NC	10
13]	NC	11
14]	NC	12
15]	NC	13
16]	NC	14
17]	NC	15
18]	S	S

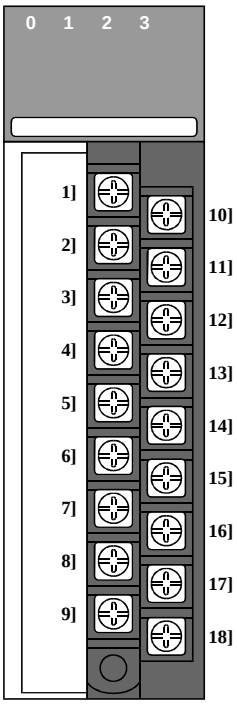
Diagram of internal circuit



Specification table (SSR output module)

Item		Specification	
Type		EH-YS4	EH-YS16
Output specification		Triac output	
Rated load voltage		100/240 V AC (85 to 250 V AC)	
Minimum switching current		100 mA	10 mA
Leak current		5 mA or less	2 mA or less
Maximum load current	1 circuit	0.5 A	0.3 A
	1 common	2 A	4 A (ambient temp. 45 °C) See below derating curve.
Output response time	OFF → ON	1 ms or less	
	ON → OFF	1 ms+1/2 cycles or less	
Number of output points / module		4 points / module	16 points / module
Number of output points / common		4 points / common	16 points / common
Surge removal circuit		Varistor	
Fuse		4 A	6.3 A (Be sure to install an external fuse)
Insulation system		Photo-triac insulation	
Output display		LED (green)	
External connection		Removable type screw terminal block (M3)	
Internal current consumption (5 V DC)		Approx. 70 mA	Approx. 250 mA
I/O assignment		Y16	

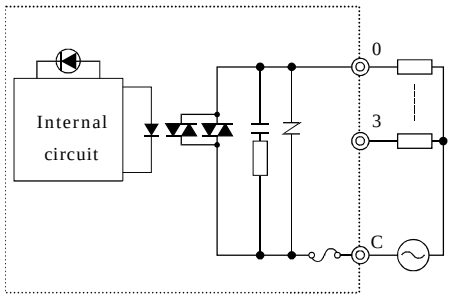
Terminal layout



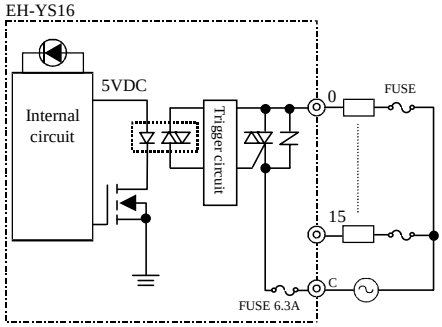
No.	Signal name	
	EH-YS4	EH-YS16
1]	0	0
2]	NC	1
3]	1	2
4]	NC	3
5]	2	4
6]	NC	5
7]	3	6
8]	NC	7
9]	C	C
10]	NC	8
11]	NC	9
12]	NC	10
13]	NC	11
14]	NC	12
15]	NC	13
16]	NC	14
17]	NC	15
18]	NC	C

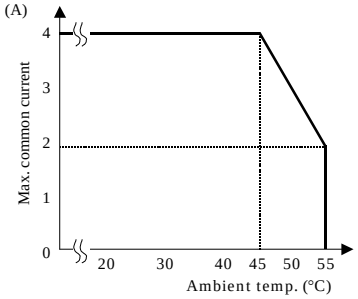
Diagram of internal circuit

EH-YS4



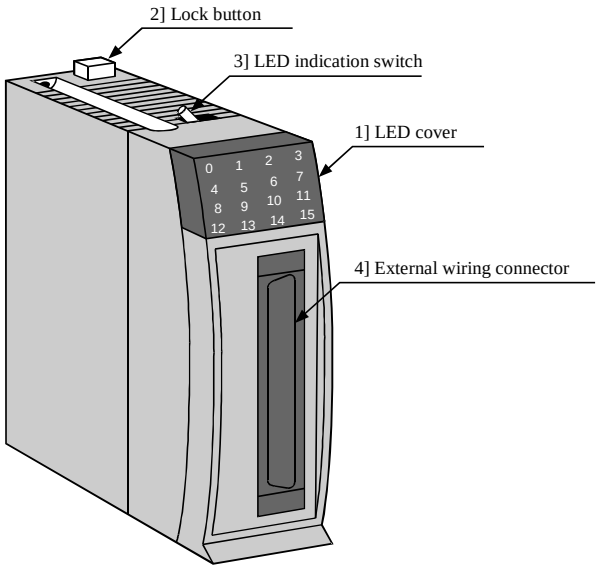
EH-YS16



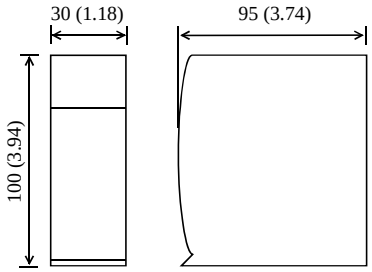


4.13 32-point Output Module

Name and function of each part



Type	EH-YT32
	EH-YTP32
Weight	Approx. 0.12 kg (0.26 lb.)
Dimensions (mm (in.))	



No.	Name	Function	Remarks
1]	LED cover	This is the cover for the LED that displays the output status. When the output signal turns on, the LED for the relevant number lights up. The LED only works when the module is energized.	
2]	Lock button	When dismounting the module from a base unit, press this button and lift up the module. The module can be fixed firmly by a screw (M4, 10 mm (0.39 in.)).	
3]	LED indication switch	Effective LED group (16 points) is selected by this switch as follows.	
4]	External wiring connector – Caution – • Approx. 120 mm (4.72 in.) of space will be required in the front of the module for the connector and cable. Please choose the installation location (space) accordingly. • Use a shielded cable and always use a class D grounding.	Connector for input signals. Special cable with connector is optional. (Refer to chapter 4.31, 4.32.) If separate connector is needed, applicable connector type is below. Manufacturer 1 : Fujitsu Takamisawa Solder type Socket: FCN-361J040-AU Cover: FCN-360C040-E Crimp type Housing: FCN-363J040 Contact: FCN-363J-AU Cover: FCN-360C040-E Crimping tool : FCN-367J040-AU/F Manufacturer 2 : AMP Solder type : 1473381-1	

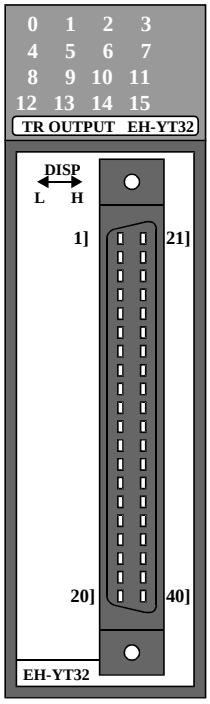
SW	LED +16	Indication group
OFF	OFF	0 to 15
ON	ON	16 to 31

Specification table (transistor output module)

Item		Specification	
Type		EH-YT32	EH-YTP32
Output specification		Transistor output (sink type)	Transistor output (source type)
Rated load voltage		12/24 V DC (+10%, -15%)	
Minimum switching current		1 mA	
Leak current		0.1 mA or less	
Maximum load current	1 circuit	0.2 A	
	1 common	6.4 A *1	
Output response time	OFF → ON	0.3 ms or less	
	ON → OFF	1 ms or less	
Number of output points / module		32 points / module	
Number of output points / common		32 points / common (4 common terminals)	
Surge removal circuit		Diode	
Fuse *2		10 A/1 common	
Insulation system		Photocoupler insulation	
Output display		LED (green) *3	
Short-circuit protection		Electronic	
External connection		Connector	
Internal current consumption (5 V DC)		Approx. 90 mA	
External power supply *4 (For supplying power to the S terminal)		12/24 V DC (+10%, -15%) (maximum 100 mA)	
I/O assignment		Y32	

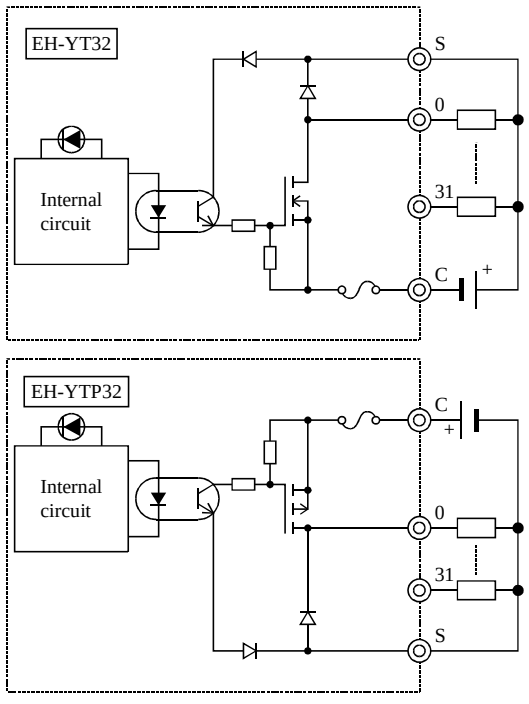
*1: Total current for 4 common terminals. Do not use more than 3A with one common terminal pin. 4 common terminals are connected internally
*2: The module needs to be repaired in case of fuse blown. It is not allowed for uses to replace the fuse.
*3: There are 16 points of LED indication. The indication group is switched by toggle switch.
*4: It is necessary to supply 12/24 V DC to the S terminal.

Terminal layout



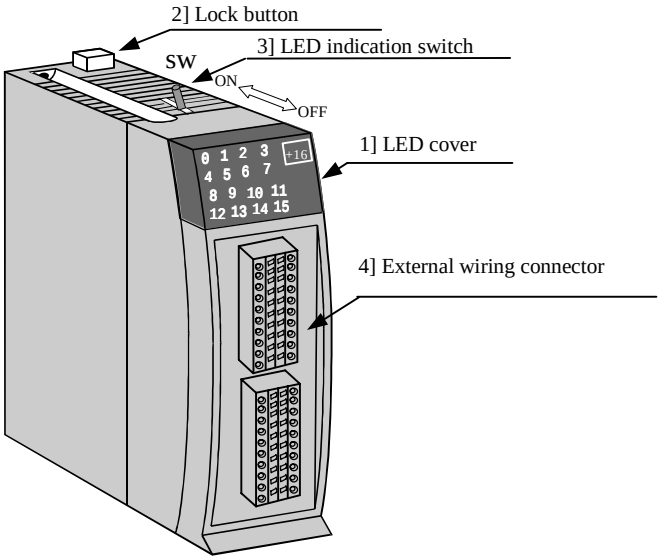
No.	Signal name	No.	Signal name
1]	0	21]	16
2]	1	22]	17
3]	2	23]	18
4]	3	24]	19
5]	4	25]	20
6]	5	26]	21
7]	6	27]	22
8]	7	28]	23
9]	C	29]	C
10]	S	30]	S
11]	8	31]	24
12]	9	32]	25
13]	10	33]	26
14]	11	34]	27
15]	12	35]	28
16]	13	36]	29
17]	14	37]	30
18]	15	38]	31
19]	C	39]	C
20]	S	40]	S

Diagram of internal circuit

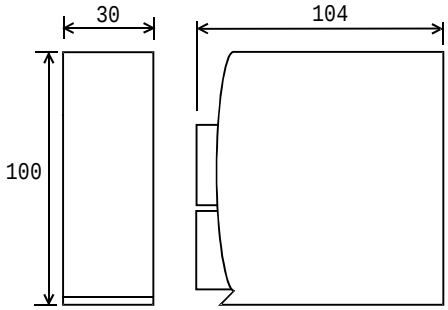


4.14 Euro-terminal 32-point Output Module

Name and function of each part



Type	EH-YT32E
	EH-YTP32E
Weight	Approx. 0.15 kg (0.33 lb.)
Dimensions (mm (in.))	



No.	Name	Function	Remarks
1]	LED cover	This is the cover for the LED that displays the input status and displayed group. When the input signal turns on, the LED for the relevant number lights up. The LED only lights when the module is energized.	
2]	Lock button	When dismounting the module from a base unit, press this button and lift up the module. The module can be fixed firmly by a screw (M4, 10 mm (0.39 in.)).	
3]	LED indication switch	Effective LED group (16 points) is selected by this switch as follows.	
4]	External wiring connector	A connector for input signals. It is attached to the product. Solid wire 0.5…1.0 mm^2 Flexible wire 0.5…1.0 mm^2 Insulation stripping length 7mm Cable ferrules cannot be used	Applicable connectors (Manufacturer: Weidmuller) • Type B2L3.5/20AUOR • Pats No. 175736 • Poles 20 • Color Orange

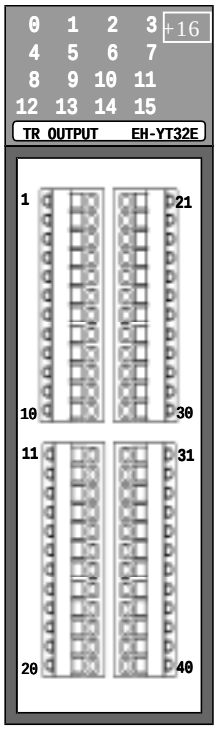
SW	LED +16	Indication group
OFF	Not lit	0 to 15
ON	Lit	16 to 31

Specification table (transistor output module)

Item		Specification	
Type		EH-YT32E	EH-YTP32E
Output specification		Transistor output (sink type)	Transistor output (source type)
Rated load voltage		12/24 V DC (+10%, -15%)	
Minimum switching current		1 mA	
Leak current		0.1 mA or less	
Maximum load current	1 circuit	0.2 A	
	1 common	4.0 A *1	
Output response time	OFF → ON	0.3 ms or less	
	ON → OFF	1 ms or less	
Number of output points / module		32 points / module	
Number of output points / common		8 points / common	
Surge removal circuit		Diode	
Fuse *2		10 A/1 common	
Insulation system		Photocoupler insulation	
Output display		LED (green) *3	
Short-circuit protection		Electronic	
External connection		Spring type connector (removable)	
Internal current consumption (5 V DC)		Approx. 90 mA	
External power supply*4 (For supplying power to the S terminal)		12/24 V DC (+10%, -15%) (max. 100 mA)	
I/O assignment		Y32	

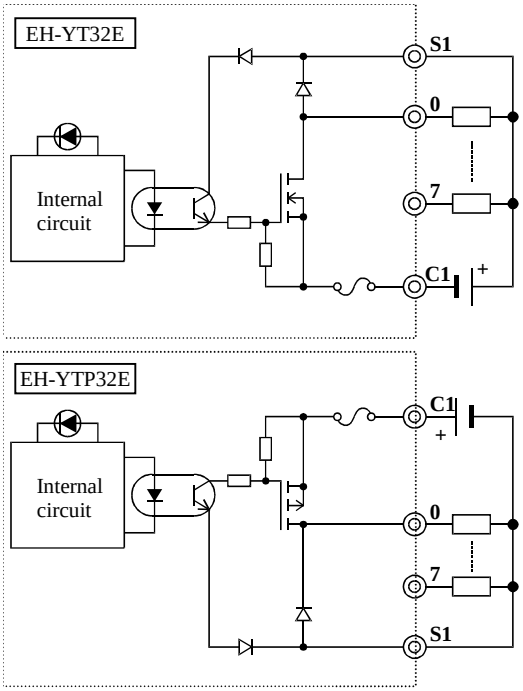
*1: Total current for 4 common terminals. Do not use more than 3A with one common terminal pin.
*2: The module needs to be repaired in case of fuse blown. It is not allowed for uses to replace the fuse.
*3: There are 16 points of LED indication. The indication group is switched by toggle switch.
*4: It is necessary to supply 12/24 V DC to the S terminal.

Terminal layout



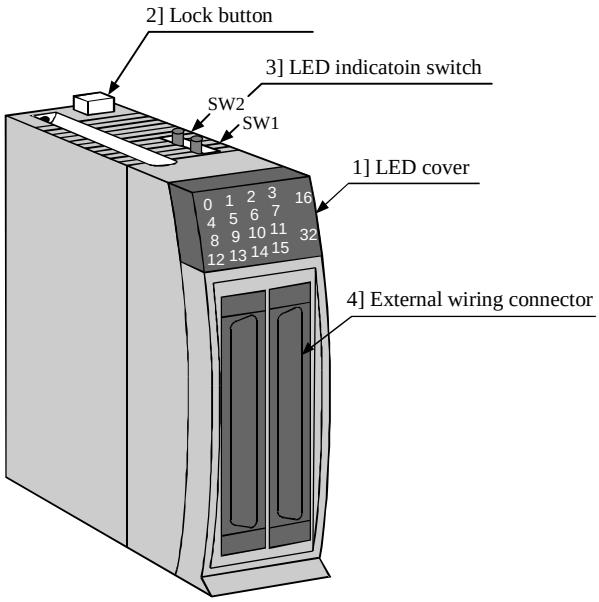
No.	Signal name	No.	Signal name
(1)	0	(21)	16
(2)	1	(22)	17
(3)	2	(23)	18
(4)	3	(24)	19
(5)	4	(25)	20
(6)	5	(26)	21
(7)	6	(27)	22
(8)	7	(28)	23
(9)	C1	(29)	C3
(10)	S1	(30)	S3
(11)	8	(31)	24
(12)	9	(32)	25
(13)	10	(33)	26
(14)	11	(34)	27
(15)	12	(35)	28
(16)	13	(36)	29
(17)	14	(37)	30
(18)	15	(38)	31
(19)	C2	(39)	C4
(20)	S2	(40)	S4

Diagram of internal circuit

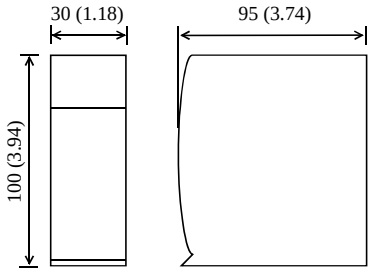


4.15 64-point Output Module

Name and function of each part



Type	EH-YT64
	EH-YTP64
Weight	Approx. 0.13 kg (0.29 lb.)
Dimensions (mm (in.))	



No.	Name	Function	Remarks
1]	LED cover	This is the cover for the LED that displays the output status and displayed group. When the output signal turns on, the LED for the relevant number lights up. The LED only works when the module is energized.	
2]	Lock button	When dismounting the module from a base unit, press this button and lift up the module. The module can be fixed firmly by a screw (M4, 10 mm (0.39 in.)).	
3]	LED indication switch	Effective LED group (16 points) is selected by these switches as follows.	
4]	External wiring connector – Caution – • Approx. 120 mm (4.72 in.) of space will be required in the front of the module for the connector and cable. Please choose the installation location (space) accordingly. • Use a shielded cable and always use a class D grounding.	Connector for input signals. Special cable with connector is optional. (Refer to chapter 4.31, 4.32.) If separate connector is needed, applicable connector type is below. Manufacturer 1 : Fujitsu Takamisawa Solder type Socket: FCN-361J040-AU Cover: FCN-360C040-E Crimp type Housing: FCN-363J040 Contact: FCN-363J-AU Cover: FCN-360C040-E Crimping tool : FCN-367J040-AU/F Manufacturer 2 : AMP Solder type : 1473381-1	

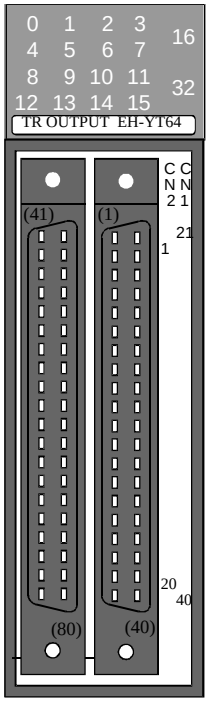
SW1	SW2	LED16	LED32	Display group
OFF	OFF	Not lit	Not lit	0 to 15
ON	OFF	Lit	Not lit	16 to 31
OFF	ON	Not lit	Lit	32 to 47
ON	ON	Lit	Lit	48 to 63

Specification table (transistor output module)

Item		Specification	
Type		EH-YT64	EH-YTP64
Output specification		Transistor output (sink type)	Transistor output (source type)
Rated load voltage		12/24 V DC (+10%, -15%)	
Minimum switching current		1 mA	
Leak current		0.1 mA or less	
Maximum load current	1 circuit	0.1 A	
	1 common	3.2 A	
Output response time	OFF → ON	0.3 ms or less	
	ON → OFF	1 ms or less	
Number of output points / module		64 points / module	
Number of output points / common		32 points / common (4 common terminals per group *2)	
Surge removal circuit		Diode	
Fuse *1		5 A/1 common	
Insulation system		Photocoupler insulation	
Output display		LED (green) *3	
Short-circuit protection		Electronic	
External connection		Connector	
Internal current consumption (5 V DC)		Approx. 120 mA	
External power supply *4 (For supplying power to the S terminal)		12/24 V DC (+10%, -15%) (maximum 100 mA)	
I/O assignment		Y64	

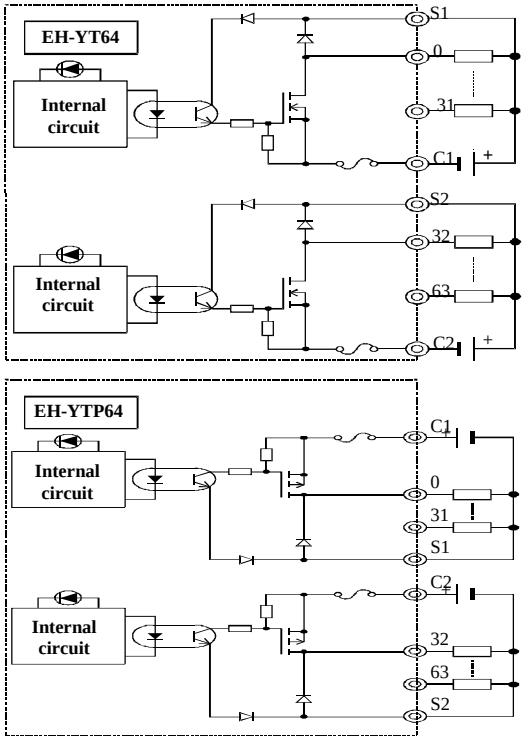
*1: The module needs to be repaired in case of fuse blown. It is not allowed for uses to replace the fuse.
*2: 2 groups (C1,C2) are separated. 4 common terminals in one group are connected internally
*3: There are 16 points of LED indication. The indication group is switched by toggle switch.
*4: It is necessary to supply 12/24 V DC to the S terminal.

Terminal layout



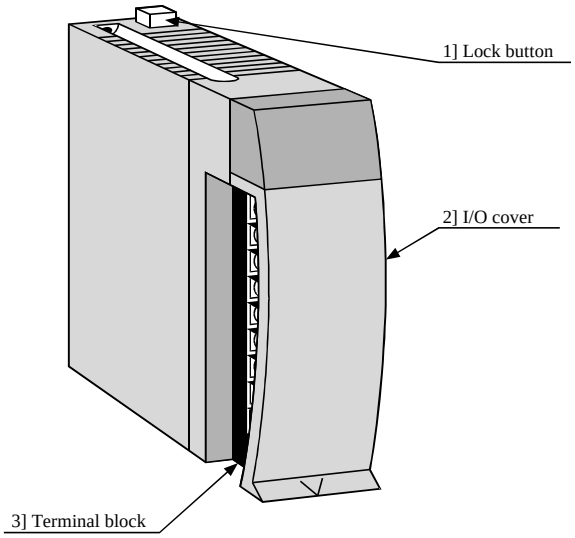
Left (CN2)				Right (CN1)			
No.	Signal name	No.	Signal name	No.	Signal name	No.	Signal name
(41)	32	(61)	48	(1)	0	(21)	16
(42)	33	(62)	49	(2)	1	(22)	17
(43)	34	(63)	50	(3)	2	(23)	18
(44)	35	(64)	51	(4)	3	(24)	19
(45)	36	(65)	52	(5)	4	(25)	20
(46)	37	(66)	53	(6)	5	(26)	21
(47)	38	(67)	54	(7)	6	(27)	22
(48)	39	(68)	55	(8)	7	(28)	23
(49)	C2	(69)	C2	(9)	C1	(29)	C1
(50)	S2	(70)	S2	(10)	S1	(30)	S1
(51)	40	(71)	56	(11)	8	(31)	24
(52)	41	(72)	57	(12)	9	(32)	25
(53)	42	(73)	58	(13)	10	(33)	26
(54)	43	(74)	59	(14)	11	(34)	27
(55)	44	(75)	60	(15)	12	(35)	28
(56)	45	(76)	61	(16)	13	(36)	29
(57)	46	(77)	62	(17)	14	(37)	30
(58)	47	(78)	63	(18)	15	(38)	31
(59)	C2	(79)	C2	(19)	C1	(39)	C1
(60)	S2	(80)	S2	(20)	S1	(40)	S1

Diagram of internal circuit

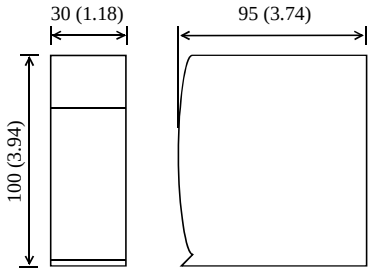


4.16 Analog I/O modules

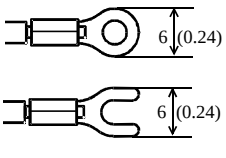
Name and function of each part



Type	EH-AX44, EH-AX8*
	EH-AY22, EH-AY2H, EH-AY4*
Weight	Approx. 0.18 kg (0.41 lb.)
Dimensions (mm (in.))	



No.	Name	Function	Remarks
1]	Lock button	When dismounting the module from a base unit, press this button and lift up the module. The module can be fixed firmly by a screw (M4, 10 mm (0.39 in.)).	
2]	I/O cover	This is the cover attached to the terminal block area.	
3]	Terminal block	This is the terminal block for connecting output signals. The terminal block can be connected or disconnected.	

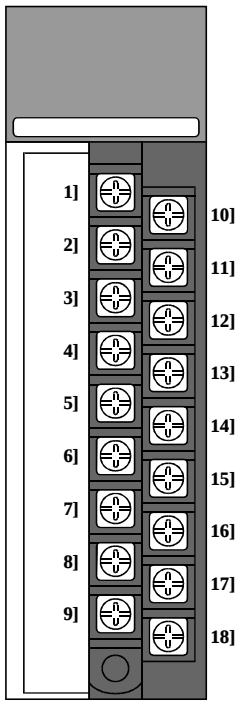
No.	Item	Detailed explanation	Comments
	Explanation of operation	Analog Input Module The CPU module verifies the status of the installed module and if the I/O assignment information matches that contained in the user program, the input information is received according to the contents of the user program. EH-AX**: Receives analog output signal (voltage or current). Analog Output Module The CPU module verifies the status of the installed module and if the I/O assignment information matches with that contained in the user program, the output information is output according to the contents of the user program. EH-AY** : Outputs analog output signal (voltage or current).	CPU104/208 of HARD REV.00 does not support analog modules.
	Terminal block	The screws for the terminal block are M3 screws. Use a crimp terminal that fits the screw diameter. The maximum thickness of the cable should be only up to 0.75 mm^2. (Use 0.5 mm^2 cable when two crimp terminals are attached to the same terminal.) The recommended crimp terminal is indicated below.  6 (0.24) (Recommended) 6 (0.24) Care must be exercised when handling the terminal since it may fall off if the screw is loose. Unit: mm (in.)	

*1: The hardware revision of the CPU module was updated to REV.01 in March of 1998. The revision of the CPU module is noted on the specifications plate attached on the side of the unit.

Specification table (analog input module)

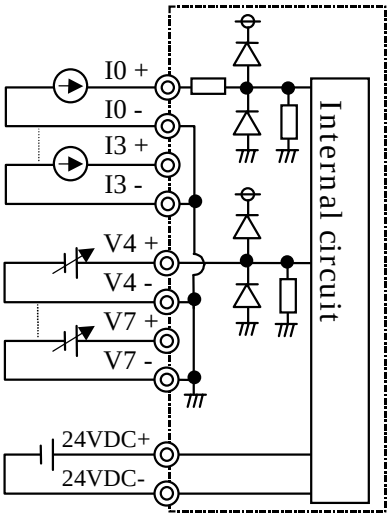
Item		Specification
Type		EH-AX44
Input current range (0 to 3 channels)		4 to 20 mA
Input voltage range (4 to 7 channels)		0 to 10 V DC
Resolution		12 bits
Conversion time		5 ms or less
Overall accuracy		± 1% or less (of full-scale value)
Input impedance	Current input (0 to 3 channels)	Approx. 100 Ω
	Voltage input (4 to 7 channels)	Approx. 100 k Ω
Insulation	Channel · Internal circuit	Photocoupler insulation
	Between channels	No insulation
Number of channels	Input current range	4 channels/module (0 to 3 channels)
	Input voltage range	4 channels/module (4 to 7 channels)
External connection		Removable type screw terminal block (M3)
Internal current consumption (5 V DC)		Approx. 100 mA
External power supply		24 V DC (+20%, -15%) Approx. 0.15 A (Approx. 0.4 A at power On)
External wiring		2-core shield wire (20 m (65.62 ft.) or less)
I/O assignment		WX8W

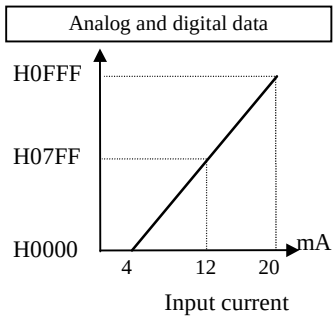
Terminal layout

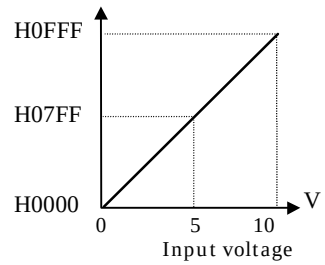


No.	Signal name
1]	I0 +
2]	I1 +
3]	I2 +
4]	I3 +
5]	V4 +
6]	V5 +
7]	V6 +
8]	V7 +
9]	24VDC+
10]	I0 –
11]	I1 –
12]	I2 –
13]	I3 –
14]	V4 –
15]	V5 –
16]	V6 –
17]	V7 –
18]	24 VDC-

Diagram of internal circuit



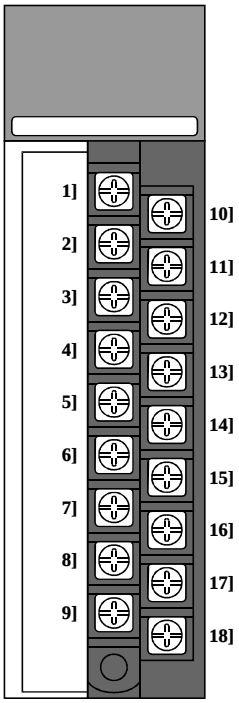




Specification table (analog voltage input module)

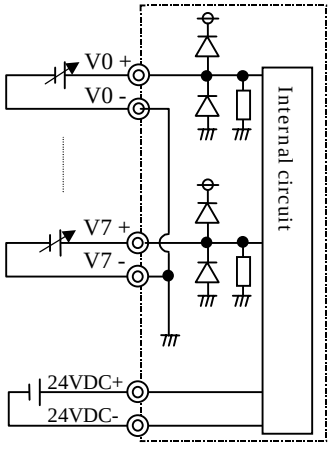
Item		Specification	
Type		EH-AX8V	EH-AX8H
Input voltage range		0 to 10 V DC	-10 to 10 V DC
Resolution		12 bits	
Conversion time		5 ms or less	
Overall accuracy		± 1% or less (of full-scale value)	
Input impedance		Approx. 100 k Ω	
Insulation	Channel · Internal circuit	Photocoupler insulation	
	Between channels	No insulation	
Number of channels		8 channels/module (0 to 7 channel)	
External connection		Removable type screw terminal block (M3)	
Internal current consumption (5 V DC)		Approx. 100 mA	
External power supply		24 V DC (+20%, -15%) Approx. 0.15 A (Approx. 0.4 A at power On)	
External wiring		2-core shield wire (20 m (65.62 ft.) or less)	
I/O assignment		WX8W	

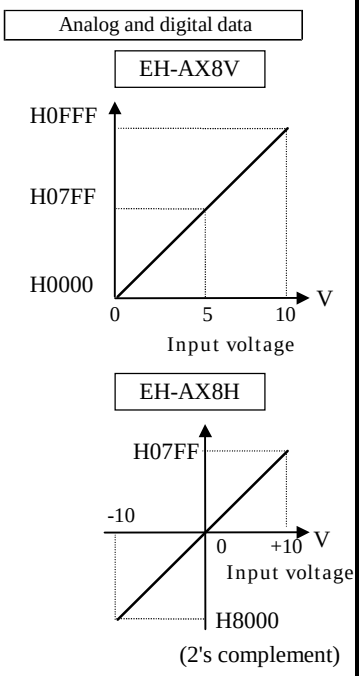
Terminal layout



No.	Signal name
1]	V0 +
2]	V1 +
3]	V2 +
4]	V3 +
5]	V4 +
6]	V5 +
7]	V6 +
8]	V7 +
9]	24 VDC+
10]	V0 –
11]	V1 –
12]	V2 –
13]	V3 –
14]	V4 –
15]	V5 –
16]	V6 –
17]	V7 –
18]	24 VDC–

Diagram of internal circuit

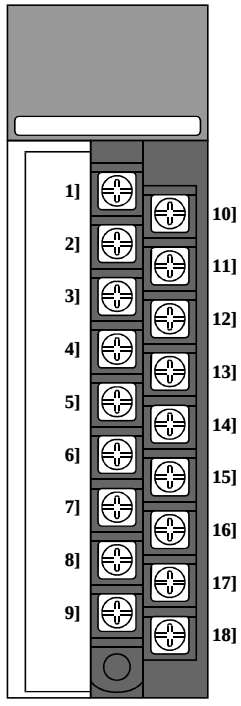




Specification table (analog current input module)

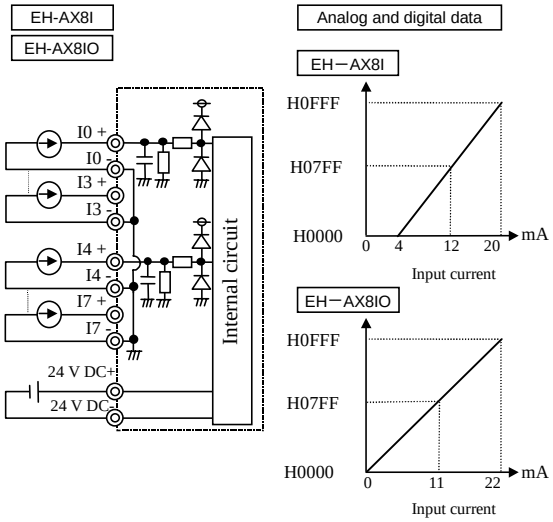
Item		Specification	
Type		EH-AX8I	EH-AX8IO
Input current range		4 to 20 mA	0 to 22 mA
Resolution		12 bits	
Conversion time		5 ms or less	
Overall accuracy		± 1% or less (of full-scale value)	
Input impedance		Approx. 100 Ω	
Insulation	Channel · Internal circuit	Photocoupler insulation	
	Between channels	No insulation	
Number of channels		8 channels/module (0 to 7 channel)	
Weight		Approx. 0.18 kg (0.4 lb.)	
External connection		Removable type screw terminal block (M3)	
Internal current consumption (5 V DC)		Approx. 100 mA	
External power supply		24 V DC (+20%, -15%) Approx. 0.15 A (Approx. 0.4 A at power On)	
External wiring		2-core shield wire (20 m (65.62 ft.) or less)	
I/O assignment		WX8W	

Terminal layout



No.	Signal name
1]	I0 +
2]	I1 +
3]	I2 +
4]	I3 +
5]	I4 +
6]	I5 +
7]	I6 +
8]	I7 +
9]	24 V DC +
10]	I0 –
11]	I1 –
12]	I2 –
13]	I3 –
14]	I4 –
15]	I5 –
16]	I6 –
17]	I7 –
18]	24 V DC –

Diagram of internal circuit

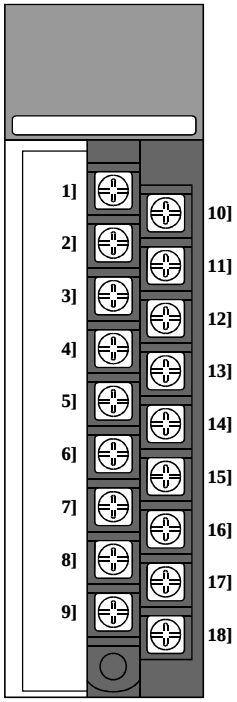


Specification table (analog output module)

Item		Specification	
Type		EH-AY22	EH-AY2H
Output voltage range (0 to 1 channel)		0 to 10 V DC	-10 to 10 V DC
Output current range (2 to 3 channels)		4 to 20 mA	—
Resolution		12 bits	
Conversion time		5 ms or less	
Overall accuracy		± 1 % or less (of full-scale value)	
External load resister	Voltage output (0 to 1 channel)	10 k Ω or more	
	Current output (2 to 3 channels)	0 to 500 Ω	—
Insulation	Channel · Internal circuit	Photocoupler insulation	
	Between channels	No insulation	
Number of channels	Output voltage range *3	2 channels/module (0 to 1 channel)	
	Output current range *3	2 channels/module (2 to 3 channels)	—
Weight		Approx. 0.18 kg (0.4 lb.)	
External connection		Removable type screw terminal block (M3)	
Internal current consumption (5 V DC)		Approx. 100 mA	
External power supply		24 V DC (+20 %, -15 %) Approx. 0.15 A (Approx. 0.5 A at power On)	
External wiring		2-core shield wire (20 m (65.62 ft.) or less)	
I/O assignment		WY8W	

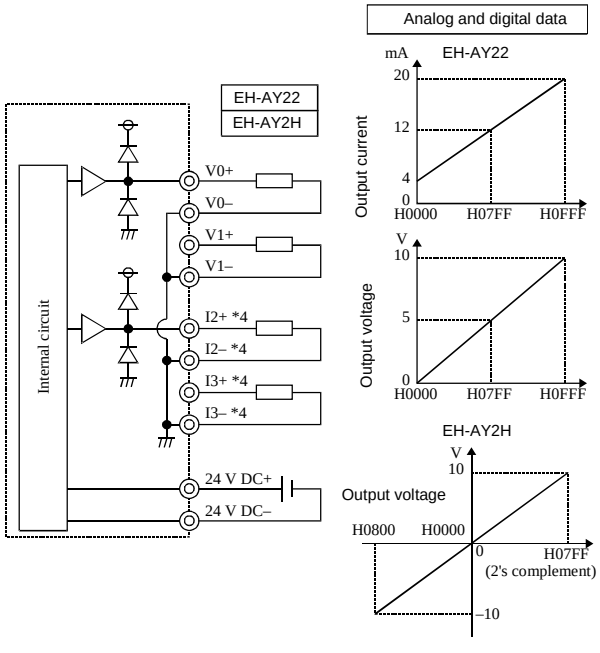
*3: With the EH-AY22, voltage output and current output can be used simultaneously.

Terminal layout



No.	Signal name
1]	V0 +
2]	V1 +
3]	I2 + *4
4]	I3 + *4
5]	NC
6]	NC
7]	NC
8]	NC
9]	24 V DC +
10]	V0 –
11]	V1 –
12]	I2 – *4
13]	I3 – *4
14]	NC
15]	NC
16]	NC
17]	NC
18]	24 V DC –

Diagram of internal circuit

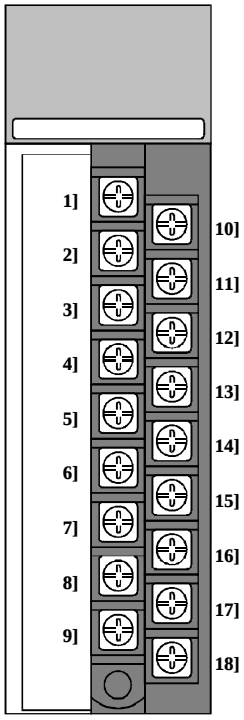


*4: Only for the EH-AY22. "NC" for the EH-AY2H.

Specification table (analog voltage output module)

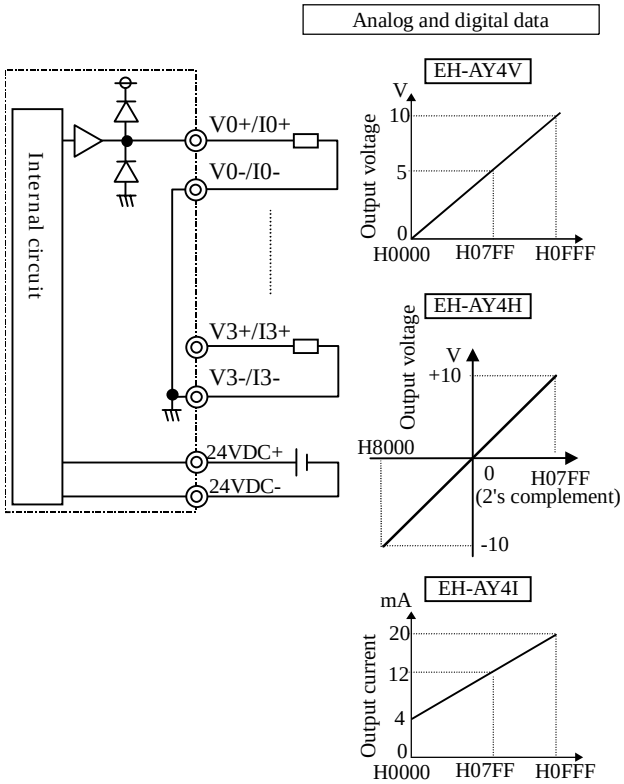
Item		Specification		
Type		EH-AY4V	EH-AY4H	EH-AY4I
Input voltage range		0 to 10 V DC	-10 to 10 V DC	4 to 20 mA
Resolution		12 bits		
Conversion time		5 ms or less		
Overall accuracy		± 1% or less (of full-scale value)		
External load resister		10 k Ω or more		0 to 350 Ω
Insulation	Channel · Internal circuit	Photocoupler insulation		
	Between channels	No insulation		
Number of channels		4 channels/module (0 to 3 channel)		
Weight		Approx. 0.18 kg (0.4 lb.)		
External connection		Removable type screw terminal block (M3)		
Internal current consumption (5 V DC)		Approx. 100 mA		Approx. 130 mA
External power supply		24 V DC (+20%, -15%) Approx. 0.15 A (Approx. 0.5 A at power On)		
External wiring		2-core shield wire (20 m (65.62 ft.) or less)		
I/O assignment		WY8W		

Terminal layout



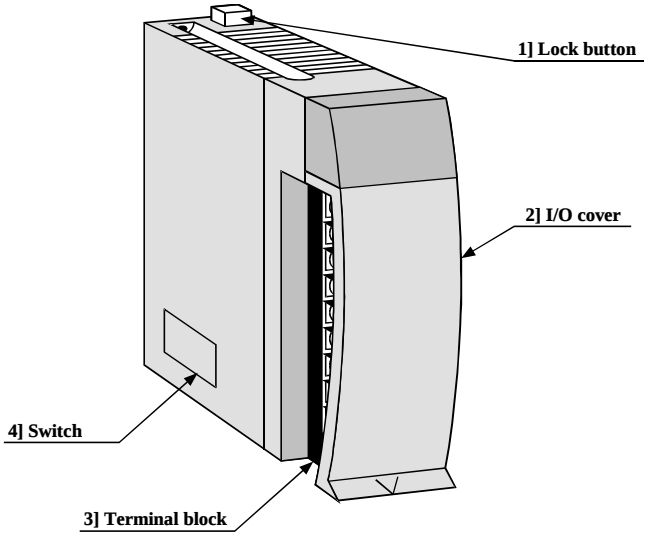
No.	Signal name	
	AY4V AY4H	AY4I
1]	V0 +	I0 +
2]	V1 +	I1 +
3]	V2 +	I2 +
4]	V3 +	I3 +
5]	NC	NC
6]	NC	NC
7]	NC	NC
8]	NC	NC
9]	24 V DC +	
10]	V0 –	I0 –
11]	V1 –	I1 –
12]	V2 –	I2 –
13]	V3 –	I3 –
14]	NC	NC
15]	NC	NC
16]	NC	NC
17]	NC	NC
18]	24 V DC –	

Diagram of internal circuit

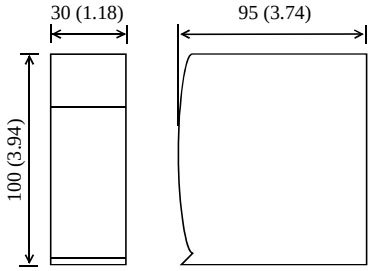


4.17 Resistance Temperature Detective Input Module

Name and function of each part



Type	EH-PT4
Weight	Approx. 0.18 kg (0.4 lb.)
Dimensions (mm(in.))	



No.	Name	Function	Remarks
1]	Lock button	When dismounting the module from a base unit, press this button and lift up the module. The module can be fixed firmly by a screw (M4, 10 mm (0.39 in.)).	
2]	I/O cover	This is the cover attached to the terminal block area	
3]	Terminal block	This is the terminal block for connecting input signals. The terminal block can be connected or disconnected.	
4]	Switch	This is to set the temperature range.	

No.	Item	Detailed explanation	Remarks
Operation explanation		The module receives input signals from RTD element. The CPU module recognizes the status of the loaded module and when it matches the I/O assignment information included in the user program, input information is received according to the contents of the user program.	
Terminal block		The screws for the terminal block are M3 screws. Use a crimp terminal that fits the screw diameter. The maximum thickness of the cable should be only up to 0.75 mm^2. (Use 0.5 mm^2 cable when two crimp terminals are attached to the same terminal.) The recommended crimp terminal is indicated below.  6 (0.24) (Recommended)  6 (0.24) Care must be exercised when handling the terminal since it may fall off if the screw is loose. Unit: mm (in.)	

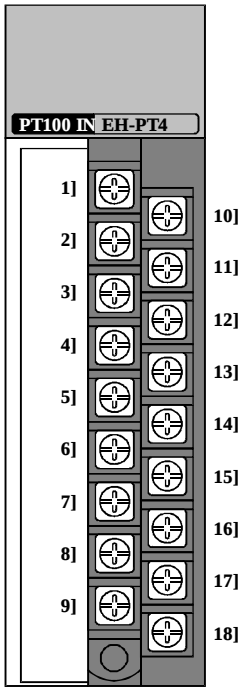
Specification table

Item		Specifications
Type		EH-PT4
Applicable temperature-measuring		Platinum temperature-measuring resistor Pt 100 (JIS C 1604-1989)/Pt 1000
Temperature conversion data		Signed 15 bits
Accuracy *1	-20 to 40°C (Pt 100)	± 0.1°C @25°C (± 0.5°C @ 0 to 55°C)
	-50 to 400°C (Pt 100)	± 0.6°C @25°C (± 3°C @ 0 to 55°C)
	-50 to 400°C (Pt 1000)	± 0.8°C @25°C (± 6°C @ 0 to 55°C)
Temperature measuring range		-20 to + 40°C/-50 to + 400°C (2 mA constant current system)
Number of input points		4
Conversion time		Approx. 1 second per 4 values
Insulation	Between value and internal circuit	Photocoupler insulation
	Between values	No insulation
Internal current consumption (5 V DC)		Approx. 160 mA
Externally supplied power		24 V DC
Unused terminal processing		The temperature conversion data for one of the four values is H7FFF.
External wiring register		The maximum total resistance of 4 values is 400 Ω
External wiring		Shielded cable
Additional function		Linearization
Error detection		The temperature conversion data at or below -51°C or at or above 410°C is H7FFF.
Wire breakage processing		The temperature conversion data for one of the four values is H7FFF.
I/O assignment		X4W

*1: The accuracy indicates the value after 10 minutes from the power-up. The value may become slightly higher immediately after the power-up. Also check the temperature-measuring resistor beforehand because it is also subject to error.

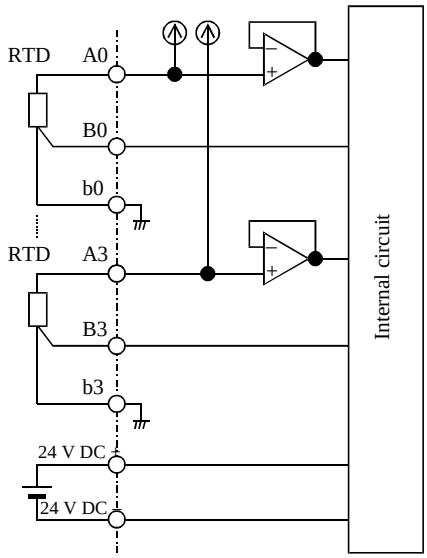
*2: Indicates the current terminal wiring in open state. When an open error occurs in the voltage terminal wiring, the data becomes inconsistent.7

Terminal layout



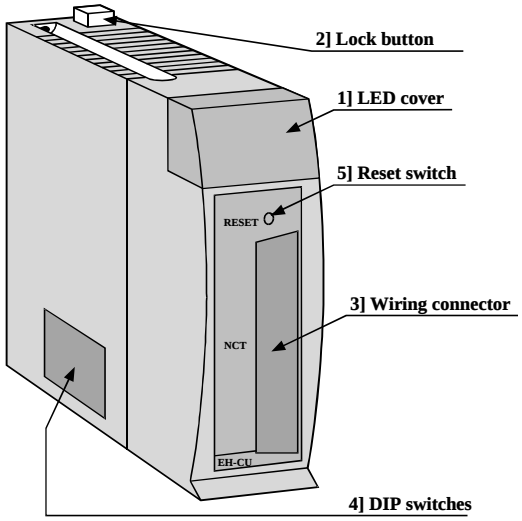
No.	Signal name
	EH-PT4
1]	b0
2]	B0
3]	b1
4]	B1
5]	b2
6]	B2
7]	b3
8]	B3
9]	24 V DC +
10]	A0
11]	NC
12]	A1
13]	NC
14]	A2
15]	NC
16]	A3
17]	NC
18]	24 V DC –

Diagram of internal circuit

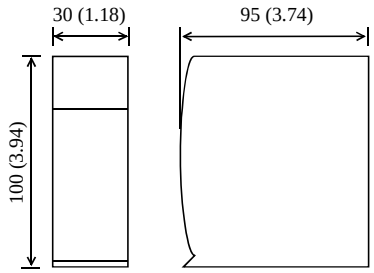


4.18 Counter Module

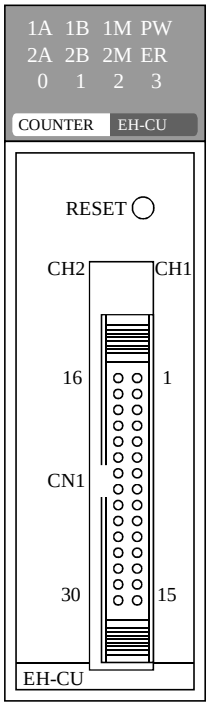
Name and function of each part



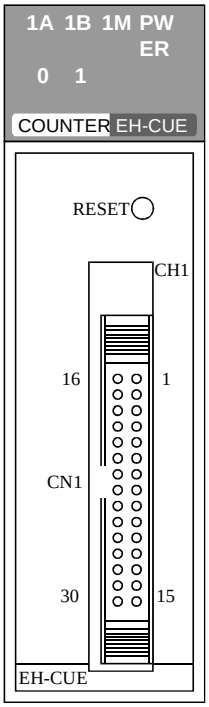
Type	EH-CU (2ch)
	EH-CUE (1ch)
Weight	Approx. 0.16 kg (0.352 lb.)
Dimensions (mm(in.))	



No.	Name	Function	Remarks
1]	LED cover	This is the cover for the LED that displays the I/O status. When the I/O signal turns on, the LED for the relevant number lights up.	
2]	Lock button	When dismounting the module from a base unit, press this button and lift up the module. The module can be fixed firmly by a screw (M4, 10 mm (0.39 in.)).	
3]	Wiring connector	15 × 2 lines 30-pin connector (Note) Common for EH-CU and EH-CUE. Connector (socket) type : HIF3BA-30PA-2.54DS (30 pin, male, HIROSE) Applicable connector (plug) (HIROSE) HIF3BA-30D-2.54C (30 pin, female) HIF3-2226SCC (connector pin) HIF3-TB2226HC (crimping tool) HIF3--30CV (connector cover)	
4]	DIP switches	Performs various initialization settings of the EH-CU and EH-CUE. Turn off the power and remove the module to perform settings.	
5]	Reset switch	This is used when the module generates a hardware error. (Note) After the power is turned on, pressing down the reset switch turns on the "ER" LED. This is normal.	

EH-CU Terminal Layout	No.	CH2	No.	CH1	Meaning of signal	
	16	Vin A	1	Vin A	Phase A	Connect to a 12 - 24 V DC power supply when using voltage input.
	17	A (+)	2	A (+)		Connect to (+) polarity when using differential input.
	18	A (–)	3	A (–)		Connect to the open collector signal when using voltage input Connect to (–) polarity when using differential input.
	19	Vin B	4	Vin B	Phase B	Connect to a 12 - 24 V DC power supply when using voltage input.
	20	B (+)	5	B (+)		Connect to (+) polarity when using differential input.
	21	B (–)	6	B (–)		Connect to the open collector signal when using voltage input Connect to (–) polarity when using differential input.
	22	Vin M	7	Vin M	Marker	Connect to a 12 - 24 V DC power supply when using voltage input.
	23	M (+)	8	M (+)		Connect to (+) polarity when using differential input.
	24	M (–)	9	M (–)		Connect to the open collector signal when using voltage input Connect to (–) polarity when using differential input.
	25 to 27 N.C.		10 to 12 N.C.			Do not connect anything.
	28	Y2	13	Y0	Output	Coincidence output. Connect to the other input.
	29	Y3	14	Y1		Coincidence output. Connect to the other input.
	30	Com2	15	Com1		(–) common for coincidence output. Commons 1 and 2 are independent.

Note: The pin number defined in the EH-CU does not correspond to the pin number defined by the connector manufacturer.

EH-CUE Terminal Layout	No.	CH2	No.	CH1	Meaning of signal	
	16	N.C.	1	Vin A	Phase A	Connect to a 12 - 24 V DC power supply when using voltage input.
	17	N.C.	2	A (+)		Connect to (+) polarity when using differential input.
	18	N.C.	3	A (–)		Connect to the open collector signal when using voltage input Connect to (–) polarity when using differential input.
	19	N.C.	4	Vin B	Phase B	Connect to a 12 - 24 V DC power supply when using voltage input.
	20	N.C.	5	B (+)		Connect to (+) polarity when using differential input.
	21	N.C.	6	B (–)		Connect to the open collector signal when using voltage input Connect to (–) polarity when using differential input.
	22	N.C.	7	Vin M	Marker	Connect to a 12 - 24 V DC power supply when using voltage input.
	23	N.C.	8	M (+)		Connect to (+) polarity when using differential input.
	24	N.C.	9	M (–)		Connect to the open collector signal when using voltage input Connect to (–) polarity when using differential input.
	25 to 27 N.C.		10 to 12 N.C.			Do not connect anything.
	28	N.C.	13	Y0	Output	Coincidence output. Connect to the other input.
	29	N.C.	14	Y1		Coincidence output. Connect to the other input.
	30	N.C.	15	Com1		(–) common for coincidence output.

Note: The pin number defined in the EH-CUE does not correspond to the pin number defined by the connector manufacturer.

General specifications

Item		Specification	
Type		EH-CU	EH-CUE
Insulation withstand voltage		250 V DC between I/O signal and FE	
Internal current consumption		5 V 200 mA	
Operating ambient temperature/humidity		0 to 55 degrees, 20 to 90% RH (Non condensing)	
Storage ambient temperature/humidity		-10 to 75 degrees, 10 to 90% RH (Non condensing)	
I/O assignment		FUN0	

Counter specifications

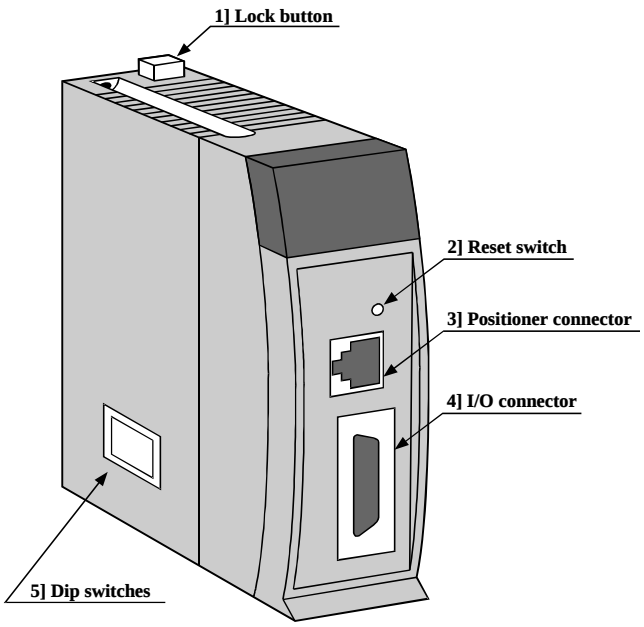
Item		Specification	
Type		EH-CU	EH-CUE
Maximum number of count		32 bit (0 to 4, 294, 967, 295)	
Maximum frequency		100 k Hz (25 k Hz when multiple of 4)	
Count mode		Select via dip switch settings. (Common to both channels for the EH-CU.) 2 phases; 1 phase (cw/ccw, ck, U/D); 2 phases, multiplication by 4	
Number of channels		2 channels	1 channel
Differential input current		4 mA or higher	
Differential input voltage		12 to 24 V DC	
	Minimum ON voltage	10 V DC	
	Maximum OFF voltage	4 V DC	
Insulation method		Photocoupler	
Number of input points 3 points x 2 channels	A: A, CW, CK	Phase difference of each channel (A - B) during 2-phase counting +45° to +125° when up, -45° to -125° when down	
	B: B, CCW, U/D		
	M: Marker (z)		
Minimum counter pulse width		ON: 4 μ s or higher, OFF: 4 μ s or higher	
Minimum marker pulse width		10 μ s or higher (Detected via ON edge)	
External wiring method		30-Pin batch connector for both channels	30-pin connector
External wiring		Wired with twisted pair wires and batch shielded wires	

Output specifications

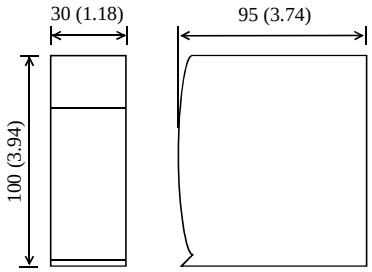
Item		Specification	
Type		EH-CU	EH-CUE
Output voltage		12/24 V DC (30 V DC maximum)	
Load current		20 mA/point maximum	
Output method		Open collector output	
Minimum load current		1 mA	
Output delay time	ON → OFF	1 ms or less	
	OFF → ON	1 ms or less	
Voltage drop when ON		1.5 V maximum	
Number of external output points		4 points/module External terminal of output destination may be specified for each channel.	2 points/module
	Normal counter	Current value = Set Value 1 or current value > Set Value 1	
	Ring counter	Current value = Set Value 2	
Leak current		0.5 mA maximum	
Polarity		(–) common within the module	
External power supply		12/24 V DC (30 V DC maximum)	
Insulation method		Photocoupler	

4.19 Single-Axis Pulse Positioning Module

Name and function of each part



Type	EH-POS
Weight	Approx. 0.17 kg (0.37 lb.)
Dimensions (mm(in.))	



No.	Name	Function	Remarks
1]	Lock button	When dismounting the module from a base unit, press this button and lift up the module. The module can be fixed firmly by a screw (M4, 10 mm (0.39 in.)).	
2]	Reset switch	Resets the unit when the module is malfunctioning.	
3]	Positioner connector	Used to connect the positioner	
4]	I/O connector	Pulse output, external control input connector (20-pin)	Connector on cable: Sumitomo 3M Solder type: 10120-3000VE Case: 10320-52F0-008 (or equivalent)
5]	Dip switches	Perform initialization settings for the pulse output method (CW/CWW, CK/Direction switching) and output logic (positive/negative logic). Also used to set whether or not external input is used. Turn off the power and remove the module to perform settings.	(Note) Setting dip switches 5 and 6 to ON causes the external RUN input to become invalid (± 0).

Switch number	ON	OFF	Additional information
1, 2	1	2	
	ON	ON	CW/CCW pulse output (Negative logic)
	ON	OFF	CW/CCW pulse output (Positive logic)
	OFF	ON	Clock pulse/Directional signal output (Negative logic)
	OFF	OFF	Clock pulse/Directional signal output (Positive logic)
3	Unused	Unused	
4	COIN not input	COIN input	Sets whether or not the positioning complete signal (COIN) is externally input.
5	+ 0. RUN not input	+ 0. input	Sets whether or not the + direction over-run (+O.RUN) is externally input.
6	- 0. RUN not input	– 0. input	Sets whether or not the – direction over-run (–O.RUN) is externally input.

General specifications

Item	Specification
Current consumption	5 V DC, 300 mA, 600 mA (When the positioner is connected) Supply power from the power supply module.
Operating ambient temperature	0 to 55°C (Storage ambient temperature -10 to 75°C)
Operating ambient humidity	20 to 90% RH (no condensation) (Storage ambient humidity 10 to 90% RH (no condensation))
Usage environment	No corrosive gases, no excessive dirt
Cooling method	Natural air cooling
Weight	Approx. 0.17 kg (0.37 lb.)
External dimensions (mm)	30 (1.18 in.) (W) × 100 (3.94 in.) (H) × 95 (37.4 in.) (D) (170 (6.7 in.) (D) with I/O connector)
External power supply	5 V DC ± 5%, 100 mA (For pulse chain output) 24 V DC, 10 mA/point (For external control input)

Function specifications

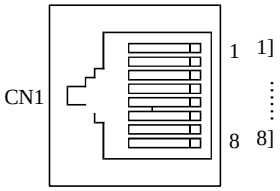
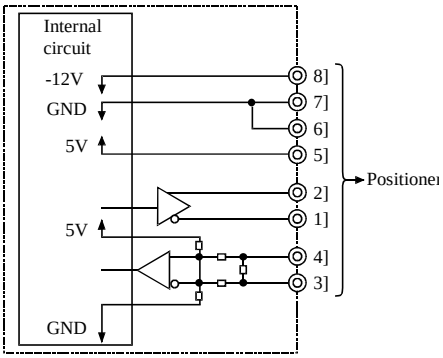
Item		Specification
Number of control axes		1 axis
Highest frequency		400 k pulse/s
Positioning data	Capacity	256 points
	Setting procedures	1. Sequence program 2. Positioner (Note that the positioner is optional.)
Positioning	Method	1. Absolute system 2. Absolute system + increment system 3. Increment system
	Positioning command	1. Pulse specification 2. μ m specification 3. inch specification 4. degree specification
	Speed command	Automatic, manual, home position return 6.25 pulse/s to 400 k pulse/s μ m/s, inch/s, degree/s input function
	Speed stage	10 stages
	Acceleration/deceleration system	Trapezoid acceleration/deceleration S-curve acceleration/deceleration (3-stage acceleration/deceleration)
	Acceleration/deceleration time	1 to 65,535 ms
	Backlash	0 to 255 pulse
	High/low limit setting	+2,147,483,647 to -2,147,483,648 pulse
	Pulse output method	1. Pulse chain (CW/CCW) 2. Clock + direction signal (CK/direction) (Use dip switches 1 and 2 to select the pulse output method and to switch between positive and negative logic for the selected method.)
	Pulse output procedures	1. Open collector output (Photocoupler insulation) 2. Line driver output (Photocoupler insulation)
Home position return function		1. Arbitrary origin 2. Low speed origin return 3. High speed origin return 1 4. High speed origin return 2 5. Absolute value encoder home position return
Teaching		Possible
Manual (JOG) operation		Pulse output by manual input signal
Operation when the CPU has stopped		Operation may be performed via I/O setting or using the positioner.
I/O allocation		Word 4W/4W
Absolute value encoder input		Supports the Σ series and ΣII series by Yasukawa Denki and the P series by Sanyo Denki.

(Notes)
- Stopping the CPU during operation causes the motor to decelerate and come to a stop.
- The maximum travel per single movement is 2,147,483,647 pulses. When an operation was attempted to move beyond the maximum travel, the motor decelerates and stops at the maximum travel position.

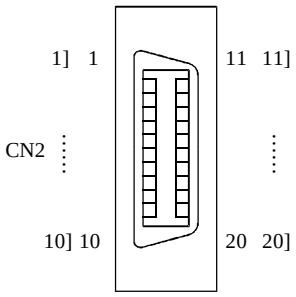
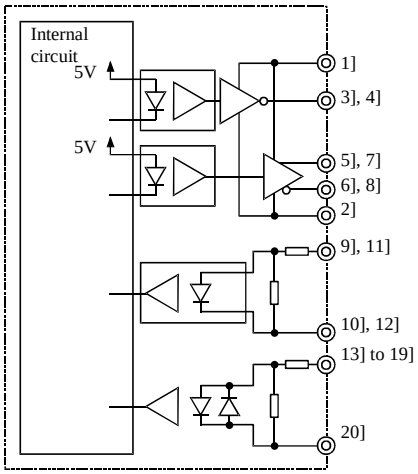
I/O interface specifications

Item			Specification
Output	Pulse chain (CW/CCW) output Clock + direction signal (CK/direction) Pulse output		1. Open collector output Photocoupler insulation (30 V DC maximum, 30 mA resistive load) 2. Line driver output Photocoupler insulation (5 V DC)
	Maximum leakage current		100 μ A or less
	Maximum voltage drop at ON		0.8 V maximum (at output current 30 mA)
Input	Input voltage		10.8 to 30 V DC
	Input impedance		Approx. 2.2 k Ω
	Input current		Approx. 10 mA (24 V DC)
	Operation voltage	Minimum ON voltage	9 V
		Maximum OFF voltage	3.6 V
	Input lag	ON → OFF	1 ms or less
		OFF → ON	1 ms or less
	Polarity		Only the encoder signal input uses the plus common inside the module. Other inputs do not specify polarity.
	Insulation method		Photocoupler

A) Positioner connector (CN1): Conforms to RS-422.

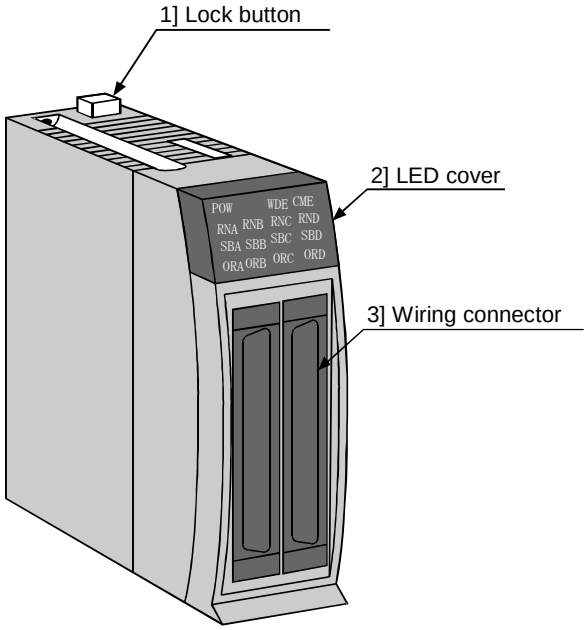
Terminal layout	No.	Signal	Signal name	Internal ladder diagram
	1]	Do –	Driver output –	
	2]	Do +	Driver output +	
	3]	Ri –	Receiver input –	
	4]	Ri +	Receiver input +	
	5]	5 V DC +	+ 5 V	
	6]	0 V	GND	
	7]	0 V	GND	
	8]	12 V DC –	-12 V	

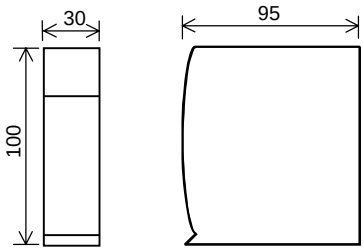
B) I/O connector (CN2)

Terminal layout	No.	Signal	Signal name	Internal ladder diagram
	1]	5 V DC +	Pulse output power supply	
	2]	0 V		
	3]	CW	Open collector pulse output	
	4]	CCW		
	5]	CW +	Line driver pulse output	
	6]	CW –		
	7]	CCW +		
	8]	CCW –		
	9]	C +	Encoder C phase	
	10]	C –		
	11]	PS –	Encoder location signal	
	12]	PS +		
	13]	COIN	Positioning complete	
	14]	PROG	Home position LS	
	15]	+ 0. RUN	+ Over run	
	16]	– 0. RUN	– Over run	
	17]	MODE-SEL	Control mode switch	
	18]	M-CW	Manual CW	
	19]	M-CCW	Manual CCW	
	20]	24 V DC +	Control power supply	

4.20 4-Axes Pulse Positioning Module

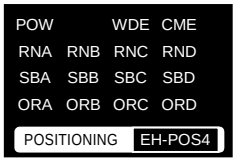
Name of Parts		Model	EH-POS4
		Weight	Approx. 0.13 kg (0.29 lb.)
		Dimension(mm)	





No.	Name	Function	Remarks
1]	Lock button	When dismounting the module from a base unit, press this button and lift up the module. The module can be fixed firmly by a screw (M4, 10 mm (0.39 in.)).	
2]	LED cover	This indicates the status of this module.	
3]	Wiring connector	This is connector for pulse output and external control input. Connector for cable: Manufacturer 1 : Fujitsu Takamisawa Solder type Socket: FCN-361J040-AU Cover: FCN-360C040-E Crimp type Housing: FCN-363J040 Contact: FCN-363J-AU Cover: FCN-360C040-E Crimping tool : FCN-367J040-AU/F Manufacturer 2 : AMP Solder type : 1473381-1	

LED display

Outlook	Name	Signal	Contents	Coler
	POW	Power source	Lighted when the module is valid	Yellow green
	RN*	RUN	Lighted in positioning	Yellow green
	SB*	Stand by	Lighted in stand by mode	Yellow green
	OR*	Overrun error	Lighted when overrun is occurred	Red
	CME	Command error	Lighted when command error is occurred	Red
	WDE	Watchdog error	Lighted when Watchdog error is occurred	Red

Note 1: *=A (Axis A), B (Axis B),C (Axis C),D (Axis D)
Note 2: All of the LED are lighted when power on.

General Specification

Item	Specification
Consumption current	5 V DC , 850 mA (supplied from Power module)
Operation temperature	0 to 55 °C
Storage temperature	–10 to 75 °C
Operation humidity	20 to 90 % RH (without condensation)
Storage humidity	10 to 90 % RH (without condensation)
Operating atmosphere	Free from corrosive gas and dust
Cooling	Natural air cooling
Weight	Approx. 0.13 k g
Dimension	30(W) x 100(H) x 95(D) (mm)
External power source	24 V DC, approx. 4.3 mA /point (for external input)

Specification

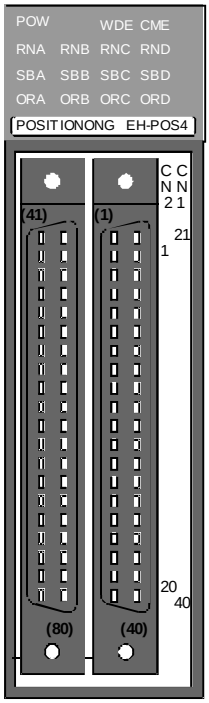
Item		Specification
Number of controlled axes		4
Number of interpolation axes		Linear interpolation : up to 4 axes Circular interpolation : 2 axes
Maximum speed		1 M pulse/ s
Positioning data	Number of positioning points	Maximum 256 points/ axis (storage in the module)
	Setting method	1) Ladder Program 2) Positioning Data Setting tool
Positioning	Positioning mode	1) Absolute mode 2) Absolute and Incremental 3) Incremental
	Positioning Unit	1) Pulse 2) μm 3) inch 4) degree
	Speed unit	1 pulse/ s - 1M pulse/ s (Auto, Manual, Homing) μm /s , inch/s , degree/s (selectable by common parameter)
	Number of speed stage	Maximum 256 stages (in continuous operation)
	Acceleration and Deceleration	Linear S-curve (3 types)
	Acceleration and Deceleration time	1 up to 65 535 ms
	Backlash	0 - 65 535 pulses
	Operation range	- 2,147,483,648 up to + 2,147,483,647 pulses - 214,748,364.8 up to + 214,748,364.7 μm - 21,474.83648 up to +21,474.83647 inch - 21,474.83648 up to + 21,474.83647 degree
	Pulse train signal	1) 2 Pulse signal (CW pulse and CCW pulse) 2) Pulse and Direction signal (PLS and SIG) (Selectable by common parameter)
	Output method	Line driver
Homing		1) Free home position 2) Low speed homing 3) High speed homing 1 (Off edge stop) 4) High speed homing 2 (Phase Z input stop) 5) Absolute encoder homing
Applied servo amp in absolute homing		Hitachi AD series
Manual operation		Manual command
Teaching function		Teaching command
I/O assignment		Word 4W/4W
Operation on CPU stopping		Available

Note: When CPU is turned "RUN" to "STOP" or "STOP" to "RUN", the servo motor stops.

Input / Output Interface

Item			Specification
Output	Pulse & Sign		Line driver (SN75158(TI))
	"High" voltage		Minimum 2.4 V
	"Low" voltage		Maximum 0.4 V
Phase input	Phase Z input and Absolute encoder serial signal		Line driver (input impedance: 220 ohm)
Input	Input voltage		20.4 up to 28.8 V DC
	Input impedance		Approx. 5.6 k ohm
	Input current		Approx. 4.3 mA (24 V DC)
	Operation voltage	"ON" voltage	Minimum 15 V DC
		"OFF" voltage	Maximum 5 V DC
	Delay	"ON" to "OFF"	Maximum 1 ms
		"OFF" to "ON"	Maximum 1 ms
	Polarity		No
	isolation		Photo-coupler

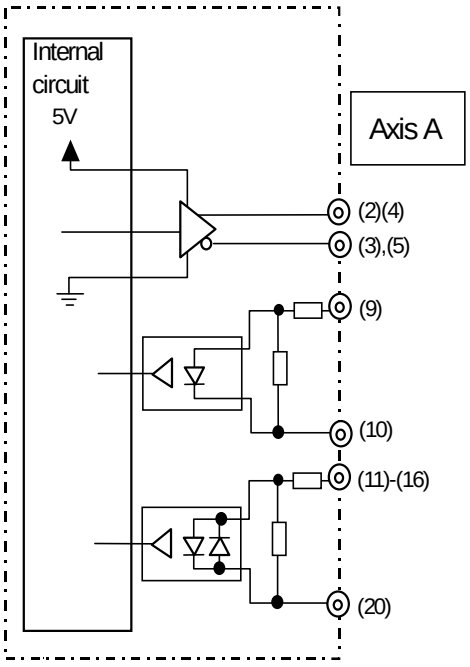
assignment



Pin assignment and signal

left(CN2)		right(CN1)		
Axis C	Axis D	Axis A	Axis B	signal
No.	No.	No.	No.	
(41)	(61)	(1)	(21)	N.C.
(42)	(62)	(2)	(22)	CW+
(43)	(63)	(3)	(23)	CW-
(44)	(64)	(4)	(24)	CCW+
(45)	(65)	(5)	(25)	CCW-
(46)	(66)	(6)	(26)	N.C.
(47)	(67)	(7)	(27)	N.C.
(48)	(68)	(8)	(28)	N.C.
(49)	(69)	(9)	(29)	Z-(PS-)
(50)	(70)	(10)	(30)	Z+(PS+)
(51)	(71)	(11)	(31)	SRDY
(52)	(72)	(12)	(32)	COIN
(53)	(73)	(13)	(33)	PORG
(54)	(74)	(14)	(34)	+ORUN
(55)	(75)	(15)	(35)	-ORUN
(56)	(76)	(16)	(36)	MODSEL
(57)	(77)	(17)	(37)	N.C.
(58)	(78)	(18)	(38)	N.C.
(59)	(79)	(19)	(39)	N.C.
(60)	(80)	(20)	(40)	COM(+24V)

Internal circuit



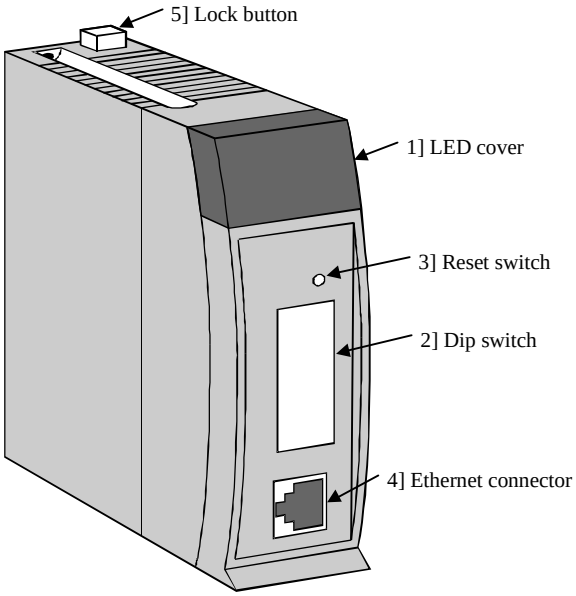
Same circuit about Axis B, C, D

I/O Signal

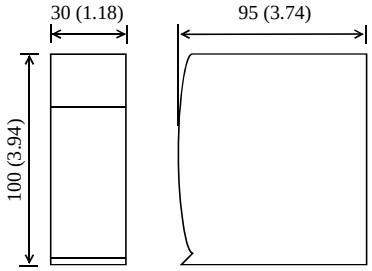
Axis C No.	Axis D No.	Axis A No.	Axis B No.	I/O	Symbol	Name	Remark
41	61	1	21				
42	62	2	22	Output	CW+	Pulse train output (CW pulse/ PLS)	Line driver
43	63	3	23		CW-		
44	64	4	24		CCW+	Pulse train output Or direction signal (CCW pulse/ SIG)	
45	65	5	25		CCW-		
46	66	6	26				
47	67	7	27				
48	68	8	28				
49	69	9	29	Input	Z-(PS-)	Phase Z input (Absolute encoder signal)	Phase Z output from servo Amp (Absolute encoder serial signal)
50	70	10	30		Z+(PS+)		
51	71	11	31		SRDY	Servo ready	Ready output from servo AMP
52	72	12	32		COIN	In position	Positioning complete signal
53	73	13	33		PORG	Origin signal	Origin limit switch
54	74	14	34		+ORUN	+ Overrun	Overrun for normal direction
55	75	15	35		-ORUN	- Overrun	Overrun for reversal direction
56	76	16	36		MODSEL	Control mode select	Control mode selective switch In speed and positioning mode
57	77	17	37				
58	78	18	38				
59	79	19	39				
60	80	20	40	Common	COM(+24V)	Input common	24 V DC or GND of external power source

4.21 Ethernet Module

Name and function of each part

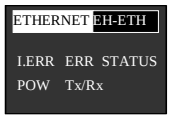


Type	EH-ETH
Weight	Approx. 0.15 kg (0.33 lb.)
Dimensions (mm(in.))	



No.	Name	Function	Remarks
1]	LED cover	This is the cover for the LED that displays the status.	
2]	Dip switch	Sets the operation mode.	
3]	Reset switch	Resets when the module is abnormal.	
4]	Ethernet connector	RJ45 type connector	
5]	Lock button	When dismounting the module from a base unit, press this button and lift up the module. The module can be fixed firmly by a screw (M4, 10 mm (0.39 in.)).	

Explanation of LED display

Front diagram	Display	Description of display	ON status	OFF status
	POW	Indicates the Ethernet line connection status.	Connected	Not connected
	Tx/Rx	Indicates the data transmission and reception status.	Sending or receiving	No data transmission
	I. ERR	Indicates Ethernet information setting error.	Error present	No error
	ERR	Indicates whether or not a transmission or reception error has occurred.	Error present	No error
	STATUS	Indicates whether or not a hardware error has occurred.	Error present (flashes)*1	No error

*1: Displays the type of a hardware error by the illumination color and flash count. For details, refer to the Ethernet Module Application Manual (NJI-361(X)).

Operation mode setting

External view	Bit								Description
	1	2	3	4	5	6	7	8	
	ON	ON	Least significant byte of IP address						Sets Ethernet information from the user program
		OFF							Sets Ethernet information using general Web browser.
	OFF	*	*	*	*	*	ON	ON	Transmission/reception test mode
								OFF	External loop-back check
							OFF	ON	Internal loop-back check
								OFF	Normal operation

Note: (1) "*" indicates that it is not dependent of the bit status.
(2) When the least significant byte of the IP address is to be set, the switch OFF becomes "0," and ON becomes "1."
(3) Do not set all of the least significant bits of the IP address to OFF.

General specifications

Item	Specification
Internal current consumption	5 V DC, 260 mA
Operating ambient temperature, humidity	0 to 55°C, 20 to 90% RH (no condensation)
Storage ambient temperature, humidity	-10 to 75°C, 10 to 90% RH (no condensation)
Noise proof feature	• Noise voltage 1,500 Vpp, noise pulse width 100 nsec, 1 μsec • 3,000 V at the electrostatic noise metal exposing section
Operating environment	No corrosive gases, no excessive dust
Structure	Open wall-mount type
Cooling method	Natural air cooling
Mounted slot position	Slots 0 to 2 on the basic base: up to two units can be mounted at the same time
I/O assignment	COMM

Performance specifications

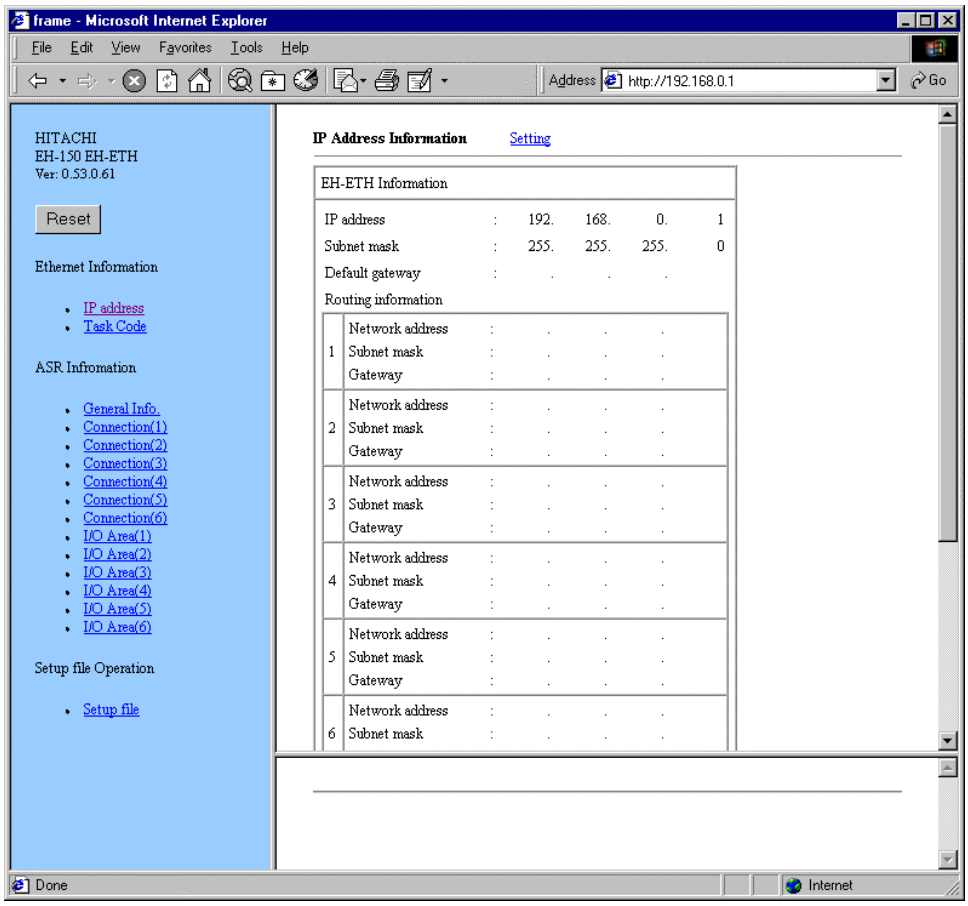
Item		Specification
Communication specification	Ethernet standard	IEEE802.3 standard
	Transfer modulation method	Base band
	Medium access method	CSMA/CD
	Transfer speed	10 Mbps
	Max. length Hub - EH-ETH	100 (m)
Number of ASR connection		Maximum connection is 6 at once. Maximum data is 1454 bytes / each sending or receiving
H-protocol (Taskcode communication)		4 connections (max.)

Functional specifications

Item	Specification
Setup function	• Select the setup mode using a DIP switch, and perform initial settings such as the IP address, transmission operation specification, transmission/reception area specification using a general-purpose Web browser. • The IP address can also be set by programming with a ladder program.
Auto Sending/Receiving function, event transmission function	• Data can be transmitted and received periodically by specifying an internal output signal in a table format. • Data can be transmitted and received by signal variation (event) in a ladder program.
Task code communication	• Either TCP/IP or UDP/IP can be specified. • H series task code communication can be performed.
Test function	• Internal loop and external loop check functions are supported. • One-to-one transmission/reception test function is supported.

Example of setup function

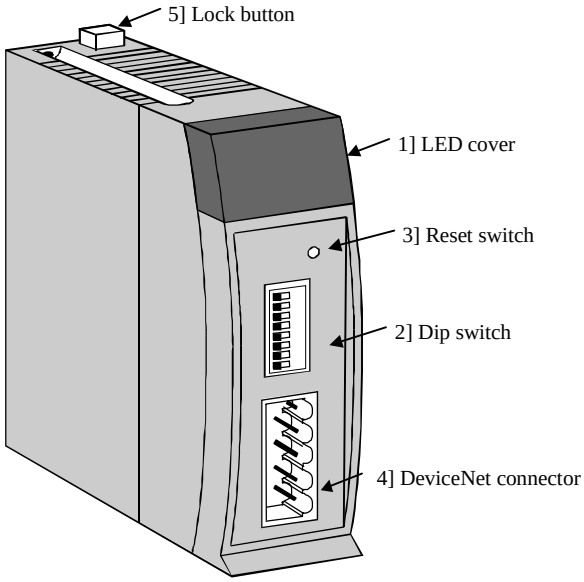
To create Ethernet information and an auto communication table, the setup page of the general Web browser is used. The following shows the setup tool screen.



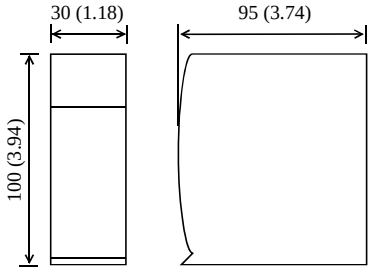
Ethernet information setup page using a Web browser

4.22 DeviceNet Master Module

Name and function of each part



Type	EH-RMD
Weight	Approx. 0.15 kg (0.33 lb.)
Dimensions (mm(in.))	



No.	Name	Function	Remarks
1]	LED cover	This is the cover for the LED that displays the network status and error information.	
2]	Dip switch	Sets the node address and baud rate.	
3]	Reset switch	Resets the module.	
4]	DeviceNet connector	Connects to the network.	
5]	Lock button	When dismounting the module from a base unit, press this button and lift up the module. The module can be fixed firmly by a screw (M4, 10 mm (0.39 in.)).	

General specifications

Item	Specification
Current consumption	5 V DC, 280 mA
Operating ambient temperature	0 to 55°C (Storage ambient temperature –10 to 75°C)
Operating ambient humidity	20 to 90 % RH (no condensation) (Storage ambient humidity 10 to 90 % RH (no condensation))
Usage environment	No corrosive gases, no excessive dirt
Cooling method	Natural air cooling
External power supply	24 V DC ± 10 % (supplied from communication connector)

Performance specifications

No.	Item	Performance specification	
		LINK mode	REMOTE mode
1	No. of installed units	2 units (only on communication slot *)	4 units (only on communication slot *)
2	No. of slave-connected units	63 units	
3	I/O assignment	LINK	REMOTE2
4	Output data	256 words (WL0-)	64 words (WX1000-, WY1000-)
5	Input data	256 words (WL200-)	
6	Communication protocol	DeviceNet 2.0 standard	
7	Supported connections	1] Poll I/O connection 2] Bit strobe I/O connection 3] Cyclic I/O connection 4] Change of state (COS) I/O connection 5] Explicit message connection	
8	Connection mode	1] Multi-drop connection 2] Multi-branch connection using T branch	
9	Communication speed	500k/250k/125kbps (set by DIP switches)	
10	Cable	Dedicated DeviceNet cable	
11	Communication distance	(see table below)	

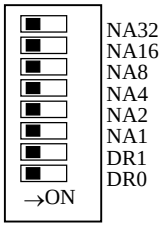
Communication	Maximum	Each sub-line	Total sub-line
500 kbps	100 m or less	6 m or less	39 m or less
250 kbps	250 m or less	6 m or less	78 m or less
125 kbps	500 m or less	6 m or less	156 m or less

The maximum network length shows the value when a thick trunk cable is used.

* Refer to chapter 4.7 Base unit.

Note: The followings are recommended communication cables and crimp type terminals for the cables:
Mfg'd by Showa Densen TDN18-**G Trunk cable (thick cable)
TDN24-**G Drop cable (thin cable)
(** indicates the number of m's. However, available lengths are 10/30/50/100/300/500m.)
Mfg'd by Nichifu Trunk cable crimp type terminal TME TC-2-11 (power supply line)
TME TC-1.25-11 (communication line)
Drop cable crimp type terminal TME TC-0.5 (common power supply line/communication line)
Crimp tool NH-32

Node address and communication speed settings

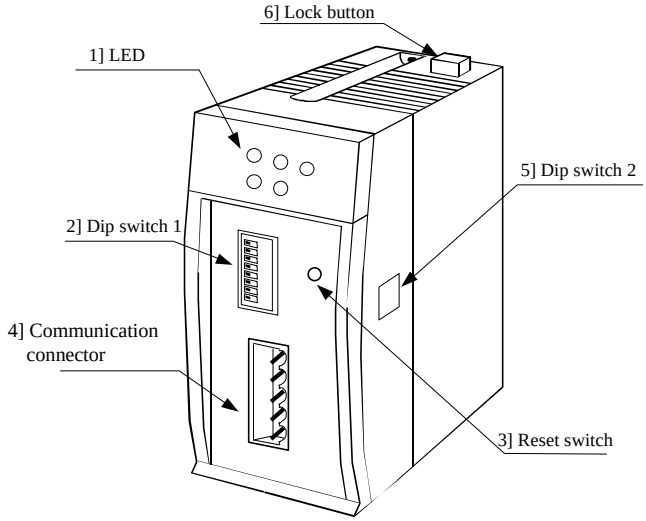


Node address	NA1	NA2	NA4	NA8	NA16	NA32
0	OFF	OFF	OFF	OFF	OFF	OFF
1	ON	OFF	OFF	OFF	OFF	OFF
2	OFF	ON	OFF	OFF	OFF	OFF
. .						
62	OFF	ON	ON	ON	ON	ON
63	ON	ON	ON	ON	ON	ON

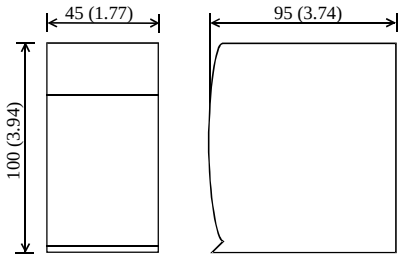
Band rate	DR0	DR1
125	OFF	OFF
250	ON	OFF
500	OFF	ON
	ON	ON

4.23 DeviceNet Slave Module

Name and function of each part



Type	EH-IOCD
Weight	Approx. 0.17 kg (0.37 lb.)
Dimensions (mm(in.))	



No.	Name	Function	Remarks
1]	LED	This is the cover for the LED that displays the network status and error information.	
2]	Dip switch 1	Sets the node address and baud rate.	
3]	Reset switch	Resets the module.	
4]	Communication connector	Connects to the network.	
5]	Dip switch 2	Sets to retain/clear output for the output module on the EH-10CD.	
6]	Lock button	When dismounting the module from a base unit, press this button and lift up the module. The module can be fixed firmly by a screw (M4, 10 mm (0.39 in.)).	

General specifications

Item	Specification
Current consumption	5 V DC, 320 mA
Operating ambient temperature	0 to 55°C (Storage ambient temperature -10 to 75°C)
Operating ambient humidity	20 to 90% RH (no condensation) (Storage ambient humidity 10 to 90% RH (no condensation))
Usage environment	No corrosive gases, no excessive dirt
Cooling method	Natural air cooling
External power supply	24 V DC ± 10 % (supplied from communication connector)

Performance specifications

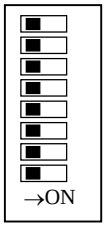
No.	Item	Specification
1]	No. of installed I/O modules	16 units/EH-IOCD (Use EH-IOC/H to install 9 or more units.)
2]	Output data	256 words
3]	Input data	256 words
4]	Communication protocol	DeviceNet 2.0 standard
5]	Supported connections	1] Poll I/O connection 2] Bit Strobe I/O connection 3] Cyclic I/O connection 4] Change of state (COS) I/O connection 5] Explicit message connection
6]	Connection mode	1] Multi-drop connection 2] Multi-drop connection using T branch
7]	Baud rate	500k/250k/125kbps (switched by DIP switches)
8]	Cable	Dedicated DeviceNet cable (see Note below)
9]	Communication distance	(see table below)

Communication speed	Maximum network length	Each sub-line length	Total sub-line length
500 k bits/s	100 m or less	6 m or less	39 m or less
250 k bits/s	250 m or less	6 m or less	78 m or less
125 k bits/s	500 m or less	6 m or less	156 m or less

The maximum network length shows the value when a thick trunk cable is used.

Note: The following are recommended communication cables and crimp type terminals for the cables:
Mfg'd by Showa Densen TDN18-**G Trunk cable (thick cable)
TDN24-**G Drop cable (thin cable)
(** indicates the number of m's. However, available lengths are 10/30/50/100/300/500m.)
Mfg'd by Nichifu Trunk cable crimp type terminal TME TC-2-11 (power supply line)
TME TC-1.25-11 (communication line)
Drop cable crimp type terminal TME TC-0.5 (common power supply line/communication line)
Crimp tool NH-32

Node address and communication speed settings



Node address	NA1	NA2	NA4	NA8	NA16	NA32
0	OFF	OFF	OFF	OFF	OFF	OFF
1	ON	OFF	OFF	OFF	OFF	OFF
2	OFF	ON	OFF	OFF	OFF	OFF
.						
62	OFF	ON	ON	ON	ON	ON
63	ON	ON	ON	ON	ON	ON

Band rate	DR0	DR1
125	OFF	OFF
250	ON	OFF
500	OFF	ON
	ON	ON

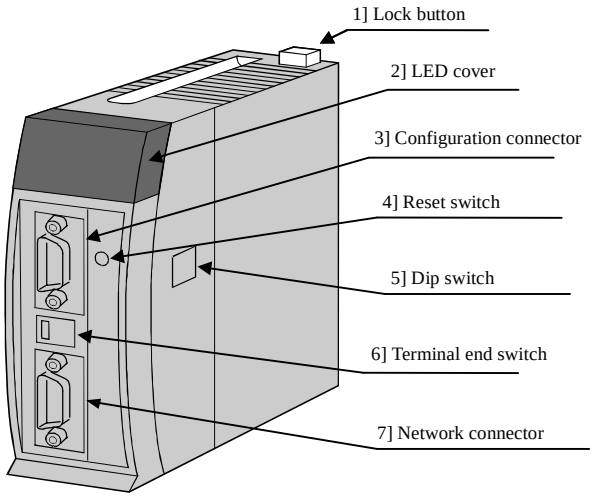
Supported I/O modules.

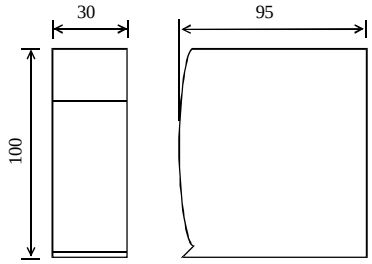
Type	Input size (Word)	Output (word)	Type	Input size (Word)	Output (word)
EH-XD8/XD16/XDL16	1	0	EH-YT64/YTP64	0	4
EH-XA16/XAH16	1	0	EH-AX44/8V/8H/8I/8IO	8	0
EH-XD32(E)/ XDL32(E)	2	0	EH-AY22/2H/4V/4I	0	8
EH-XD64	4	0	EH-PT4	4	0
EH-YT8/YTP8/YT16/YTP16	0	1	EH-POS/4	4	4
EH-YS4/YR8B/YR12/YR16	0	1	EH-CU/CUE	5	3
EH-YT32(E)/ YTP32(E)	0	2			

4.24 PROFIBUS Master Module

Name and function of each part

Type	EH-RMP
Weight	Approx. 0.13 kg (0.286 lb.)
Dimensions (mm(in.))	

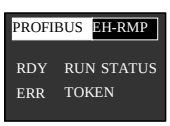




No.	Name	Function	Remarks
1]	Lock button	When dismounting the module from a base unit, press this button and lift up the module. The module can be fixed firmly by a screw (M4, 10 mm (0.39 in.)).	
2]	LED cover	Displays the communication and other statuses of the module.	
3]	Configuration connector	Used to transfer the configuration data. D-sub 9-pin male	The temperature gets high when connected. Exercise caution when handling the connector.
4]	Reset switch	Resets the unit when the module is malfunctioning.	
5]	Dip switch *1	Specifies the data transfer mode for the slave when the CPU is in the stop state.	
6]	Terminal end switch	Turns on when this module is at the end of a network.	
7]	Network connector	A connector for connecting to a network. D-sub 9-pin female	The temperature gets high when connected. Exercise caution when handling the connector.

*1: Prior to setting the DIP switches, remove the module.
When the setting of the DIP switches is complete, paste the attached protective sheet

Explanation of LED display

Front diagram	Display	Description of display
	RDY	Indicates the hardware abnormality of this module.
	RUN	Indicates the communicating status with the slave.
	ERR	Indicates an error on the network.
	TOKEN	Always lit.
	STATUS	Indicates a hardware abnormality, parameter abnormality and other status of this module.

General specifications

Item	Specification
Operating ambient temperature	0 to 55°C (Storage ambient temperature –10 to 75°C)
Operating ambient humidity	20 to 90 % RH (no condensation), storage ambient humidity 10 to 90 % RH (no condensation)
Current consumption	5 V DC 600 mA
Usage environment	No corrosive gases, no excessive dirt
Cooling method	Natural air cooling

Functional specifications

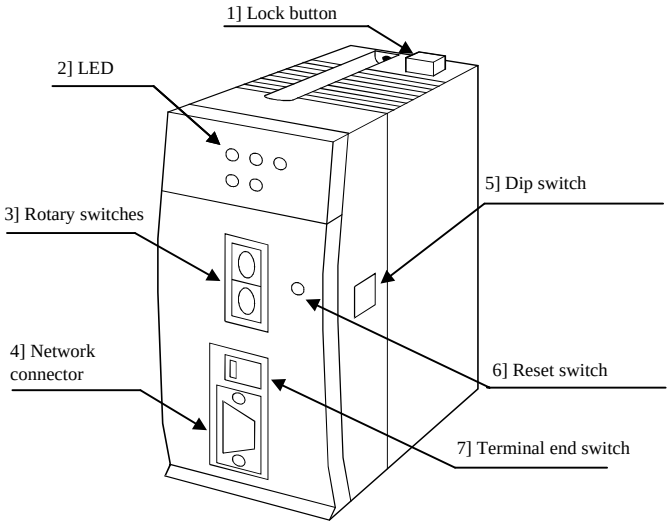
Item	Specification
No. of installed units	2 units/CPU (only on communication slot *1)
No. of supported slave units	Maximum of 124 units. However, a repeater is required to connect 32 or more units.
No. of output words	256 words
No. of input words	256 words
I/O assignment	LINK *2
Band rate: Segment length	9.6 kbps : 1,200 m 19.2 kbps : 1,200 m 45.45 kbps : 1,200 m 93.75 kbps : 1,200 m 187.5 kbps : 1,000 m 500 kbps : 400 m 1,500 kbps : 200 m 3 Mbps : 100 m 6 Mbps : 100 m 12 Mbps : 100 m
Self-diagnostics	System ROM/RAM check Watchdog timer
GSD file	File name: Hita1004.gsd Please contact our sales department.
Configurator	HMS Fieldbus, configuration software made by AB, is used. Please contact our sales department regarding the purchase.

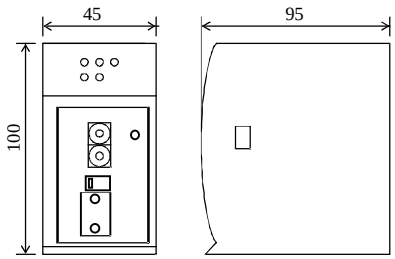
*1: Refer to chapter 4.7 Base unit.
*2: Among 1024 words, only 512 words are used. Do not use the remaining area.

4.25 PROFIBUS Slave Module

Name and function of each part

Type	EH-IOCP
Weight	Approx. 0.16 kg (0.35 lb.)
Dimensions (mm(in.))	





No.	Name	Function	Remarks
1]	Lock button	When dismounting the module from a base unit, press this button and lift up the module. The module can be fixed firmly by a screw (M4, 10 mm (0.39 in.)).	
2]	LED	Displays the communication and other statuses of the module.	
3]	Rotary switches	Sets the node address.	
4]	Network connector	A connector for connecting to a network. D-sub 9-pin female	The temperature gets high when connected. Exercise caution when handling the connector.
5]	Dip switch	Specifies the output data for the output module when the network is abnormal.	
6]	Reset switch	Resets the unit when the module is malfunctioning.	
7]	Terminal end switch	Turns on when this module is at the end of a network.	

General specifications

Item	Specification
Operating ambient temperature	0 to 55°C (Storage ambient temperature –10 to 75°C)
Operating ambient humidity	20 to 90 % RH (no condensation), storage ambient humidity 10 to 90 % RH (no condensation)
Current consumption	5 V DC 600 mA
Usage environment	No corrosive gases, no excessive dirt
Cooling method	Natural air cooling

Functional specifications

Item	Specification
No. of installed I/O modules	16 units/EH-IOCP (use EH-IOC to install 9 and more units.)
Node address setting range	1 to 99
Input/output capacity	208 words
Data update time	5 ms
Transmission speed: Segment length	9.6 kbps : 1,200 m 19.2 kbps : 1,200 m 93.75 kbps : 1,200 m 187.5 kbps : 1,000 m 500 kbps : 400 m 1,500 kbps : 200 m 3 Mbps : 100 m 6 Mbps : 100 m 12 Mbps : 100 m
Self-diagnostics	System ROM/RAM check Watchdog timer
GSD file	File name: Hita 049.gsd Please contact our sales department.

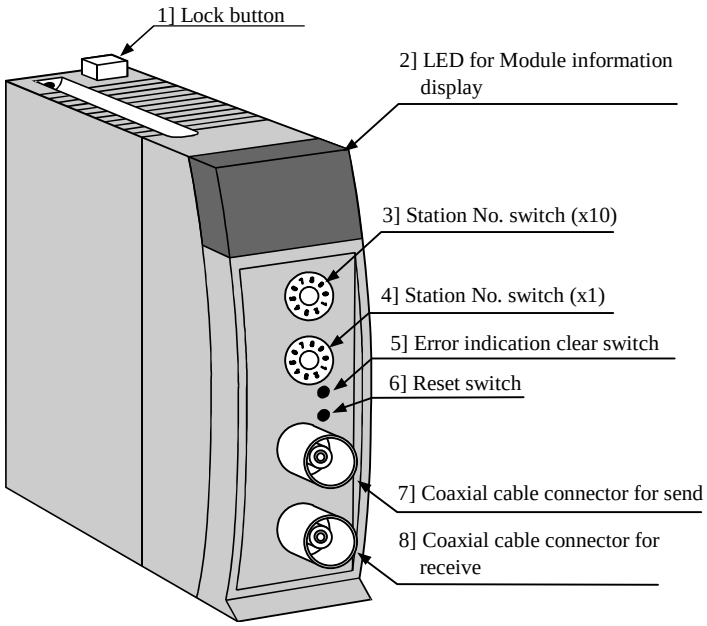
Supported I/O modules.

Type	Input size (Word)	Output (word)	Type	Input size (Word)	Output (word)
EH-XD8/XD16/XDL16	1	0	EH-YT64/YTP64	0	4
EH-XA16/XAH16	1	0	EH-AX44/8V/8H/8I/8IO	8	0
EH-XD32(E)/ XDL32(E)	2	0	EH-AY22/2H/4V/4I	0	8
EH-XD64	4	0	EH-PT4	4	0
EH-YT8/YTP8/YT16/YTP16	0	1	EH-POS/4	4	4
EH-YS4/YR8B/YR12/YR16	0	1	EH-CU/CUE	5	3
EH-YT32(E)/ YTP32(E)	0	2			

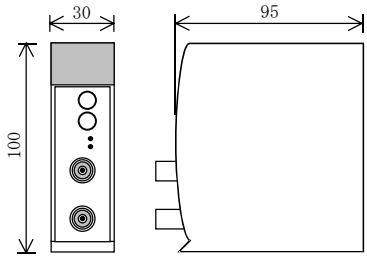
4.26 CPU Link Module (Coaxial cable)

4.26 CPU Link Module (Coaxial type)

Name and function of each part



Type	EH-LNK
Weight	Approx. 0.15 kg
Dimensions (mm)	



No.	Name	Function	Remark
1]	Lock button	When dismounting the module from a base unit, press this button and lift up the module. The module can be fixed firmly by a screw (M4, 10 mm (0.39 in.)).	
2]	LED for Module information display	It is shown that data transmission is carried out. And error information are indicated.	
3]	Station No. switch (x10)	The Link station number is determined by this switch. Setting of this switch becomes valid when the power is ON or reset switch is pushed down on. (Example)  x10 -- 1  x1-- 8 } Station No. 18	- Address range is from No.00 to 63. - When station no. is duplicated, duplicate station No. error occurs. - If No.00 station does not exist in link system, it will not work.
4]	Station No. switch (x1)		
5]	Error indication clear switch	Contents of display on ERR LED are cleared by this switch. However, the error is indicated again if it has not yet been cancelled.	The ERR LED will be extinguished. The CPU error information will not be cleared.
6]	Reset switch	Hardware reset is made by this switch. Data in WL area is also cleared.	*
7]	Coaxial cable connector for send (TXD)	It sends data to other station. Connect to RXD connector on next station.	
8]	Coaxial cable connector for receive (RXD)	It receives data from other station. Connect to TXD connector on next station.	

* The CPU module will detect a "Link Module Error (error code: 59H)" when the reset switch is pressed.
Resolve the CPU module error after the link module resumes normal operations when the reset switch has been pressed.

General Specification

Item		Specification
General Specification	Operating temperature	0 to 55℃
	Preserving temperature	－10 to 75℃
	Operating humidity	No condensation 20 to 90% RH
	Preserving humidity	No condensation 10 to 90% RH
	Current consumpion	5V DC Approx. 550 mA

Functional Specification

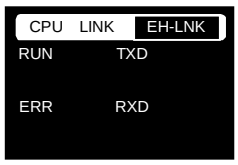
Item			Specification
Functional Specification	No. of connected Link module		Max. 64 units per Link system
	No. of mounted units		Max. 2 units per CPU (Only on communication slot *1)
	No. of Link points		1,024 words per Link system (2,048 words per 2 Link systems) *2
	Data delivery system		Common data area system
	Send / Receive distinction on data area allocation		Parameter setting from peripheral device
	Designation of Station No.		0 to 63 ; designated by rotary switch
	Communication speed		1.0 Mbps
	Transfer method		Half – duplex serial transfer, frame synchronization
	Communication method		Token passing
	Modulation method		Base band
	Refresh time		Approx. 390ms in case of 1,024 words communication with 64 stations connected *3
	Error check		CRC, overrun check, time out, open circuit, parameter error (dual designation of station No., overlapped Link area, etc.)
	Self – diagnosis		System ROM / RAM check, watchdog timer check, transfer loop back check
Transfer path Specification	Transfer path form		Loop type
	Cable length	Between stations	Max. 500m
		Total extension	Max. 1,000m
	Error station processing		Bypass system
	Recommended cable		5D2V with shield or equivalent
	Recommended connector		413631-1 (made by AMP) or equivalent

*1 Refer to chapter 4.7 Base unit.

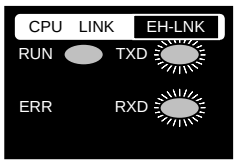
*2 It is not possible to assign retentive area.

*3 This could be longer if other peripheral devices access via Link network.

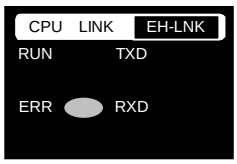
Display of the LED

Appearance	Name	Color	Display
	TXD	Green	Indicates the status of sending data by blinking.
	RXD	Green	Indicates the status of receiving data by blinking.
	RUN	Green	Indicates the normal status by turn on.
	ERR	Red	Normal state : Turn off Error (data link is possible) : Blinking (blinking in 1 sec intervals) Error (data link is impossible) : Blinking (blinking in 0.5 sec intervals) or Turn on

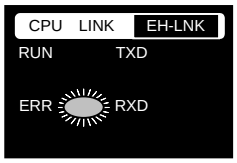
(1) In normal state

	Status of LED display / Status of Link module
	RUN LED is turned on. LED of TXD / RXD is blinking when data are transmitted or received. Link module do communication with other station. * It becomes the same indication when the warning that a data link is possible occurs.

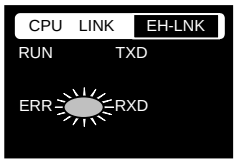
(2) In hardware failure (Watchdog timer error, ROM / RAM error)

	Status of LED display / Status of Link module
	ERR LED is turned on. Link module do not communication with other station.

(3) In serious failure (Overlap of station No, Outside of range of station No. etc.)

	Status of LED display / Status of Link module
	ERR LED blinking in 0.5 sec intervals. Link module do not communication with other station.

(4) In Warning (CRC error, Time out error in receiving peripheral data, etc.)

	Status of LED display / Status of Link module
	ERR LED blinking in 1 sec intervals. Link module do not communication temporarily with other station. (When the same error frequently occurs, Link module doesn't communicate with other station.)

Caution

1. The CPU module which can't use EH-LNK.

The list of the CPU which can't use EH-LNK is shown in the following.

CPU Type	Version
EH-CPU104 / 208	Not supported
EH-CPU104A / 208A	
EH-CPU308 / 316	
EH-CPU308A / 316A	Supported by all the versions
EH-CPU448 / 448A	Ver. C3.22 / C4.00 or later
EH-CPU516 / 548	Supported by all the versions

2. Mount

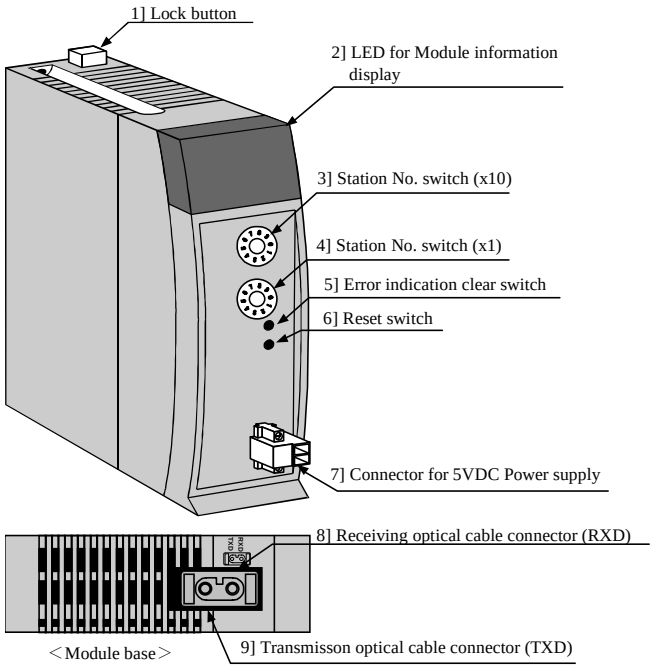
Mount a CPU link module in either of 0 slot from 2 slot of the basic base. It can't be mounted on the slot except for the above of the basic base and the expansion base.

3. Data link with large H series.

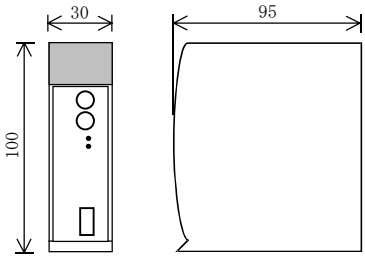
The timing when data are renewed in the large H series and EH-150 is different. EH-150 faces though it is renewed at the timing of I/O Refresh, and a large H series is renewed at the timing when a CPU referred to it. Be careful of the preparation of the program when a large H series is mixed in the link system with EH-150.

4.27 CPU Link Module (Optical type)

Name and function of each part



Type	EH-OLNK
Weight	Approx. 0.15 kg
Dimensions (mm)	



No.	Name	Function	Remark
1]	Lock button	When dismounting the module from a base unit, press this button and lift up the module. The module can be fixed firmly by a screw (M4, 10 mm (0.39 in.)).	
2]	Module information display LED	Displays data transmission and receipt, error information and other details. Refer to the section on LED display for further details.	
3]	Station No. setting switch (10-digits)	This switch determines the number of the link station. The setting for this switch is validated when the power supply is switched on or when the reset switch is pressed. The setting range is between 00 and 63. (Ex.) Station No. 18  10-digits … 1  1-digits … 8 } Station No. 18	- Address range is from No.00 to 63. - When station no. is duplicated, duplicate station No. error occurs. - If No.00 station does not exist in link system, it will not work.
4]	Station No. setting switch (1-digits)		
5]	Error display clearance switch	To clear the contents of the ERR LED display (the error will be displayed again if the cause is not resolved.)	The ERR LED will be extinguished. The CPU error information will not be cleared.
6]	Reset switch	Performs the hardware reset procedure.	*
7]	Connector for 5VDC power supply	If the link system needs to work while this module is not powered, supply 5VDC from another power source.	
8]	Receiving optical cable connector (RXD)	To Receive data from other stations. Connect to the next station's TXD by optical cable.	
9]	Transmission optical cable connector (TXD)	To send data to other stations. Connect to the next station's RXD by optical cable.	

* The CPU module will detect a "Link Module Error (error code: 59H)" when the reset switch is pressed. Resolve the CPU module error after the link module resumes normal operations when the reset switch has been pressed.

General Specification

Item		Specification
General Specification	Operating temperature	0 to 55°C
	Preserving temperature	-10 to 75°C
	Operating humidity	No condensation 20 to 90% RH
	Preserving humidity	No condensation 10 to 90% RH
	Current consumpion	5V DC Approx. 550 mA

Functional Specification

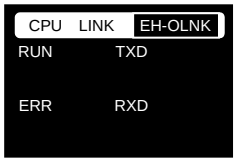
Item			Specification
Functional Specification	No. of connected Link module		Max. 64 units per Link system
	No. of mounted units		Max. 2 units per CPU (Only on communication slot *1)
	No. of Link points		1,024 words per Link system (2,048 words per 2 Link systems) *2
	Data delivery system		Common data area system
	Send / Receive distinction on data area allocation		Parameter setting from peripheral device
	Designation of Station No.		0 to 63 ; designated by rotary switch
	Communication speed		1.0 Mbps
	Transfer method		Half – duplex serial transfer, frame synchronization
	Communication method		Token passing
	Modulation method		Base band
	Refresh time		Approx. 390ms in case of 1,024 words communication with 64 stations connected *3
	Error check		CRC, overrun check, time out, open circuit, parameter error (dual designation of station No., overlapped Link area, etc.)
	Self – diagnosis		System ROM / RAM check, watchdog timer check, transfer loop back check
Transfer path Specification	Transfer path form		Loop type
	Cable length	Between stations	Max. 1,000m
		Total extension	Max. 15,000m
	Error station processing		Bypass system (In case of supply a 5VDC from the outside.)
	Recommended Cable and connector		CA7103-(1)M-(2)L(3)1 JAPAN OPNEXT product (1) Cable length, (2)Cable type, (3) Core number

*1 Refer to chapter 4.7 Base unit.

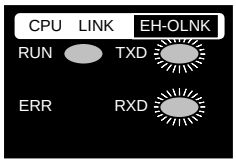
*2 It is not possible to assign retentive area.

*3 This could be longer if other peripheral devices access via Link network.

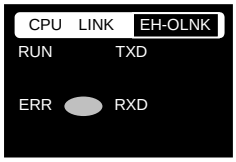
LED indication

Appearance	Name	Color	Description
	TXD	Green	Blinking when data is being transmitted.
	RXD	Green	Blinking when data is being received.
	RUN	Green	Lighting when the link module is operating normally.
	ERR	Red	Normal operations: Off "data link possible" errors: Blinks (1s cycle) "data link not possible" errors: Blinks (0.5s cycle) or lighting

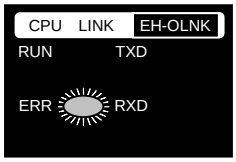
(1) Normal operations (LED display status and module status)

	Status of LED indication / Status of Link module
	The RUN LED lights. The TXD and RXD LEDs blink while link data is being transmitted to/from other link modules. The same display will apply when a [data link possible] error occurs.

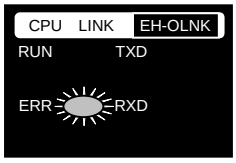
(2) Hardware error (Watchdog timer error and ROM/RAM error)

	Status of LED indication / Status of Link module
	The ERR LED lights. Communications with other link modules will not be performed.

(3) Configuration error (Station number duplicated or out of the range, etc.)

	Status of LED indication / Status of Link module
	The ERR LED blinks every 0.5 second. (0.5s ON / 0.5s OFF) Communications with other link modules will not be performed.

(4) Communication error (transmission errors, peripheral data receiving errors, etc.)

	Status of LED indication / Status of Link module
	The ERR LED blinks every 1 second. (1s ON / 1s OFF) Communications with other link modules temporarily suspended. (Communications with other link modules will not be performed if the same error occurs frequently.)

Caution

1. Supported CPU modules

The EH-OLNK can be used together with the following types of CPU module.

CPU Type	Version
EH-CPU104 / 208	Not supported
EH-CPU104A / 208A	
EH-CPU308 / 316	
EH-CPU308A / 316A	Supported by all the versions
EH-CPU448 / 448A	Ver. C3.22 / C4.00 or later
EH-CPU516 / 548	Supported by all the versions

2. Available slot number

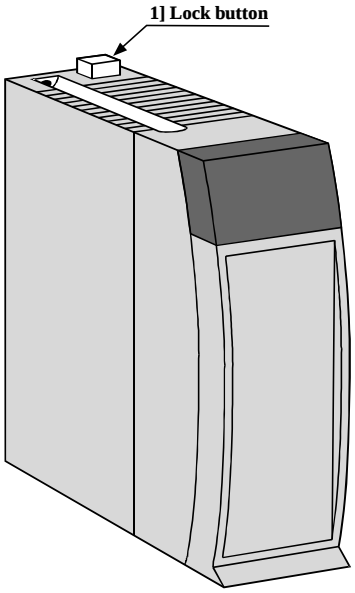
Mount the EH-OLNK on slot 0 to 2 of the basic base. The other slots of basic base and all slots of expansion base are not available.

3. Link data with large H series.

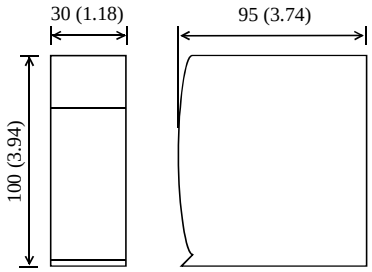
Timing of Link data refresh is different between large H series (CPU-xxHa, CPU2-xxH) and EH-150 series. In case of EH-150 series, link data is refreshed at scan end. But in case of H series, when a command is executed, the link data is refreshed directly. Make your program carefully when large H and EH-150 series are mixed in the link system.

4.28 Dummy Module

Name and function of each part



Type	EH-DUM
Weight	Approx. 0.06 kg (0.132 lb.)
Dimensions (mm(in.))	



No.	Name	Function	Remarks
1]	Lock button	Use when dismounting the unit from base unit.	

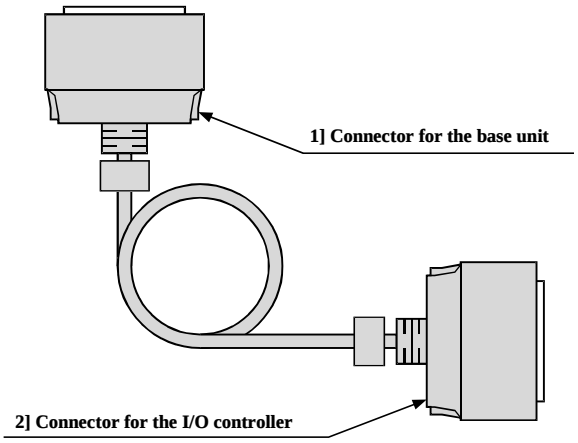
Electronic parts are not mounted on the dummy module.

Use "Empty16" for I/O assignment.

4.29 Expansion Cable

Name and function of each part

Type	EH-CB10
Weight	Approx. 0.24 kg (0.53 lb.)
Length	1 m (3.281 ft.)

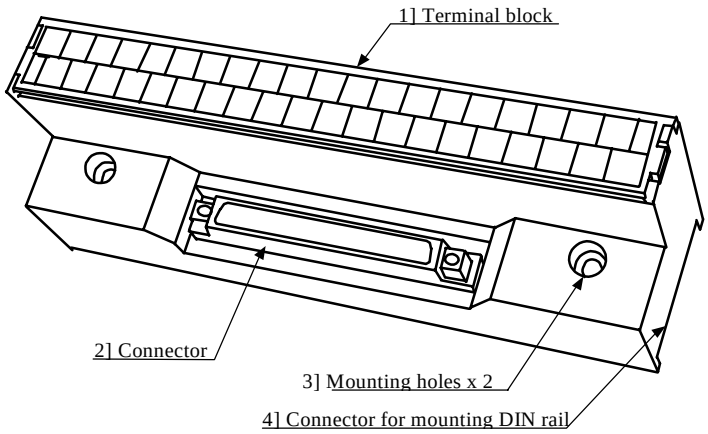


No.	Name	Function	Remarks
1]	Connector for the base unit *	Connect to the connector of the basic base unit.	
2]	Connector for the I/O controller *	Connect to the connector of the I/O controller.	

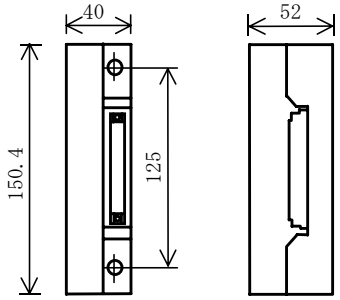
* The connectors are represented as the base unit side and I/O controller side for presentation purposes, but either one can be connected to either side.

4.30 Terminal Unit for 32 / 64 points I/O module

Name and function of each part

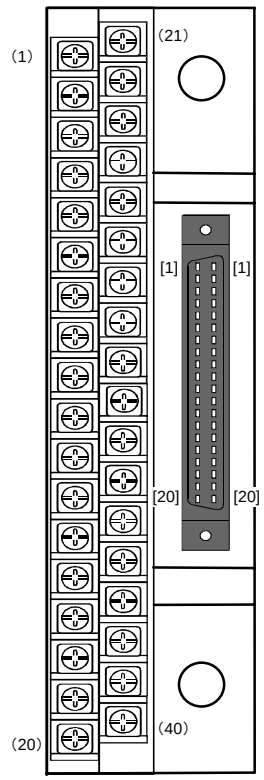


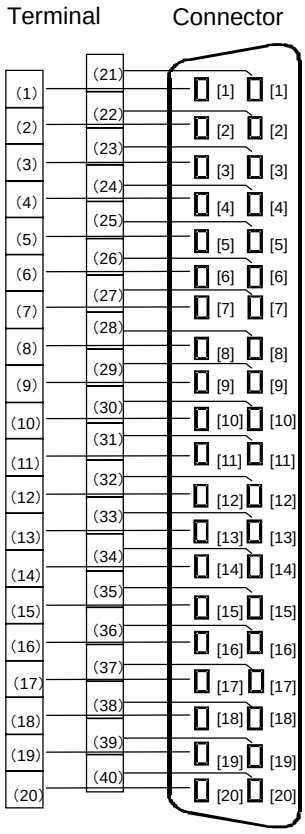
Type	HPX7DS-40V6
Weight	Approx. 0.22 kg
Dimensions (mm)	



No.	Name	Function	Remark
1]	Terminal block	This is the terminal block for connecting input / output signals.	
2]	Connector	A Connector for 32 / 64 I/O modules. 20 x 2 line, 40 Pin	
3]	Mounting holes	These are used when the terminal unit is attached to a panel. Use M4 x 25mm screws.	
4]	Connector for mounting DIN rail	This is used when the terminal unit is attached to the DIN rail.	

Terminal configuration

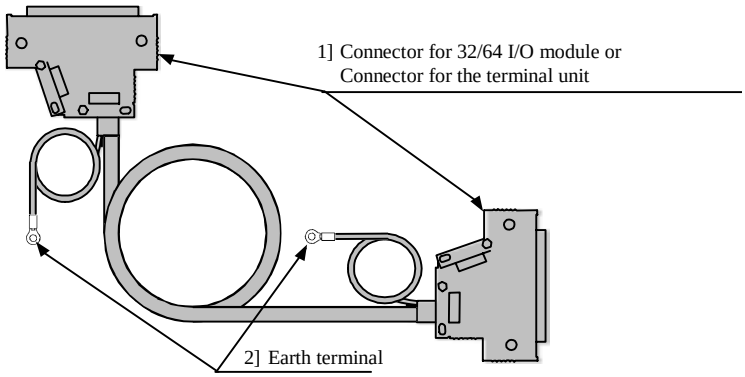




4.31 Cable for 32 / 64 points module (connector type)

Name and function of each part

Type	EH-CBM01W / EH-CBM03W EH-CBM05W / EH-CBM10W
Length	1 m / 3 m / 5 m / 10 m
Diameter	AWG# 28

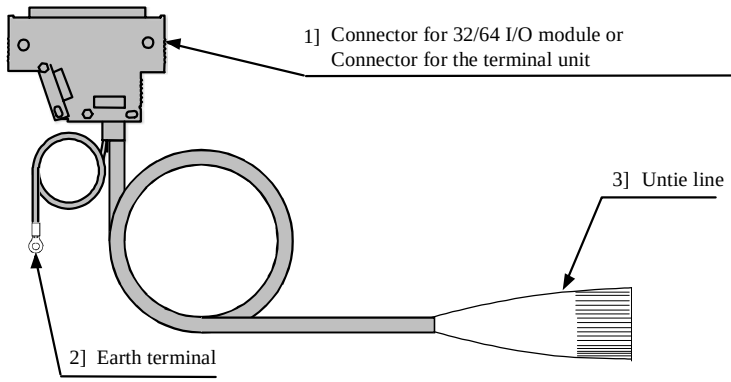


No.	Name	Function	Remark
1]	Connector for I/O module Connector for the terminal unit	Connector to the 32 / 64 I/O module. Connector to the terminal unit.	
2]	Earth terminal	Use a terminal for class D grounding	

4.32 Cable for 32 / 64 points module (open type)

Name and function of each part

Type	EH-CBM01 / EH-CBM03 EH-CBM05 / EH-CBM10
Length	1 m / 3 m / 5 m / 10 m
Diameter	AWG# 28



No.	Name	Function	Remark
1]	Connector for I/O module Connector for the terminal unit	Connector used to connect the 32 / 64 I/O module. Connector used to connect the terminal unit.	
2]	Earth terminal	Use a terminal for class D grounding	
3]	Untie line	Use a wiring for I/O signal from 32 / 64 I/O module or the terminal unit.	

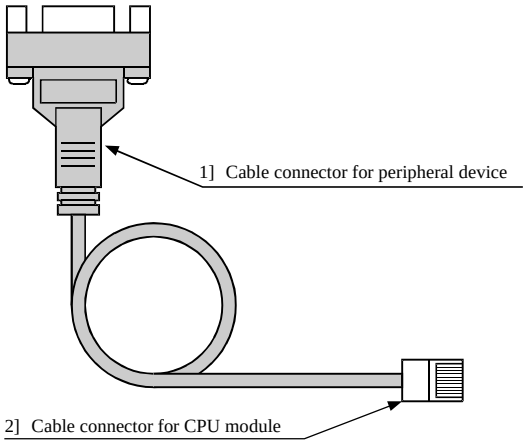
Cable code for wiring

Connector Pin No.	Color	Dot (Color)	Connector Pin No.	Color	Dot (Color)
1	Orange	■ (Black)	21	Orange	■■■ (Black)
2	Orange	□ (Red)	22	Orange	□□□ (Red)
3	Gray	■ (Black)	23	Gray	■■■ (Black)
4	Gray	□ (Red)	24	Gray	□□□ (Red)
5	White	■ (Black)	25	White	■■■ (Black)
6	White	□ (Red)	26	White	□□□ (Red)
7	Yellow	■ (Black)	27	Yellow	■■■ (Black)
8	Yellow	□ (Red)	28	Yellow	□□□ (Red)
9	Pink	■ (Black)	29	Pink	■■■ (Black)
10	Pink	□ (Red)	30	Pink	□□□ (Red)
11	Orange	■■ (Black)	31	Orange	■■■■ (Black)
12	Orange	□□ (Red)	32	Orange	□□□□ (Red)
13	Gray	■■ (Black)	33	Gray	■■■■ (Black)
14	Gray	□□ (Red)	34	Gray	□□□□ (Red)
15	White	■■ (Black)	35	White	■■■■ (Black)
16	White	□□ (Red)	36	White	□□□□ (Red)
17	Yellow	■■ (Black)	37	Yellow	■■■■ (Black)
18	Yellow	□□ (Red)	38	Yellow	□□□□ (Red)
19	Pink	■■ (Black)	39	Pink	■■■■ (Black)
20	Pink	□□ (Red)	40	Pink	□□□□ (Red)
			Earth	Shielded Cable	

4.33 Conversion Cable for Connecting between CPU and Programmer, etc.

Name and function of each part

Type	EH-RS05
Weight	Approx. 0.07 kg (0.154 lb.)
Length	0.5 m (1.64 ft.)



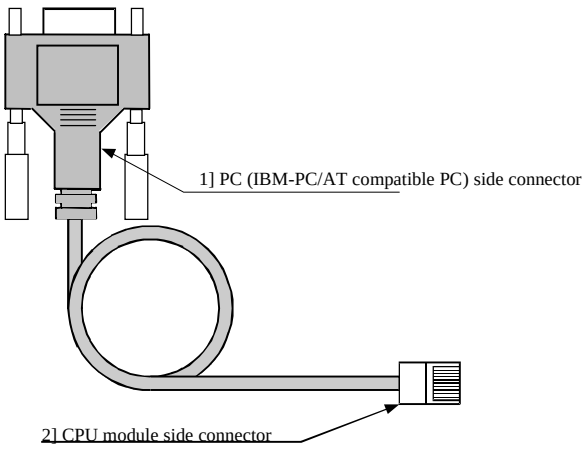
No.	Name	Function	Remarks
1]	Cable connector for peripheral device	To be connected to the cable leading to a peripheral device.	
2]	Cable connector for CPU module	To be connected to CPU module port 1 or port 2.	

When EH-RS05 is used as the cable, connection to peripheral devices may be made similarly to the connector of the H-series CPU for peripheral devices. For connection to peripheral units, it is necessary to set the mode switches. For details, see Chapter 10, “Communication Specifications.” (If the settings differ, communication cannot be performed.)

4.34 Cable for Connecting between CPU and PC (IBM-PC/AT Compatible Personal Computer)

Name and function of each part

Type	EH-VCB02
Weight	Approx. 0.13 kg (0.29 lb.)
Length	2 m (6.56 ft.)



No.	Name	Function	Remarks
1]	Connector for peripheral device	To be connected to PC (IBM-PC/AT compatible PC)	
2]	Cable connector for CPU module	To be connected to CPU module port 1 or port 2.	

If the EH-VCB02 cable is used, the CPU and PC (IBM-PC/AT compatible PC) can directly be connected without a conversion cable. For connection to peripheral units, it is necessary to set the mode switches. For details, see Chapter 10, “Communication Specifications.” (If the settings differ, communication cannot be performed.)

4.35 Current consumption

Name	Type	Current consumption [mA]	Name	Type	Current consumption [mA]
CPU module	EH-CPU104(A)	400	Analog module	EH-AX44	100
	EH-CPU208(A)	400		EH-AX8V	100
	EH-CPU308(A)	400		EH-AX8H	100
	EH-CPU316(A)	400		EH-AX8I	100
	EH-CPU448(A)	400		EH-AX8IO	100
	EH-CPU516	400		EH-AY22	100
	EH-CPU548	400		EH-AY2H	100
Memory board	EH-MEMP	30		EH-AY4V	100
	EH-MEMD	30		EH-AY4H	100
I/O controller	EH-IOC	30		EH-AY4I	130
	EH-IOCH	80		EH-PT4	160
Power supply module	EH-PSA	-	High function module	EH-CU	310
	EH-PSD	-		EH-CUE	310
Base unit	EH-BS3A	200		EH-POS	300 (600)*1
	EH-BS5A	200		EH-POS4	850
	EH-BS8A	200	Communication module	EH-LNK	550
	EH-BS11A	200		EH-OLNK	550
Input module	EH-XD8	30		EH-ETH	260
	EH-XD16	50		EH-RMD	280
	EH-XDL16	50		EH-RMP	600
	EH-XD32	60		EH-IOCD	320
	EH-XD32E	60		EH-IOCP	600
	EH-XDL32E	60	Dummy module	EH-DUM	0
	EX-XD64	80			
	EH-XA16	50			
	EH-XAH16	50			
Output module	EH-YR8B	220			
	EH-YR12	40			
	EH-YR16	430			
	EH-YT8	30			
	EH-YTP8	30			
	EH-YT16	50			
	EH-YTP16	50			
	EH-YTP16S	50			
	EH-YT32	90			
	EH-YTP32	90			
	EH-YT32E	90			
	EH-YTP32E	90			
	EH-YT64	120			
	EH-YTP64	120			
	EH-YS4	70			
	EH-YS16	250			

*1 : Value in brackets is in case positioner connected

Chapter 5 Command Specifications

5.1 Command Classifications

The commands used with the EH-150 are classified as shown in the following table.

Table 5.1 Command classification table

	Command classification	Description	Type
1	Basic commands	Sequence	21
		Timer/counter	11*
		Relational box	8
2	Arithmetic commands	Substitution (array variable)	1
		Mathematical operations	10
		Logical operations	3
		Relational expression	8
3	Application commands	Bit operation	3
		Shift/rotate	8
		Transfer	6
		Negation/Two's complement/Sign	6
		Conversion	5
		Application: Square root, bit count, swap, FIFO, unit, distribute, I/O conversion	9
4	Control commands	END, JMP, CAL, FOR, NEXT, RTS, RTI, LBL, SB, INT, CEND, CJMP	12
5	High-function module transfer commands	TRNS0, RECV0, TRNS7, RECV7, TRNS8	5*
6	FUN commands	Process stepping, trigonometry, comment, PID and others	71*

* The number of commands that can be used is different depending on the CPU module type.

Caution:

WR internal output area is different depending on the type of CPU module.

EH-CPU104(A) : WR000 to WRFFF
EH-CPU208(A) : WR000 to WR1FFF
EH-CPU308(A) : WR000 to WR43FF
EH-CPU316(A) : WR000 to WR57FF
EH-CPU448(A) : WR000 to WRC3FF
EH-CPU516 : WR000 to WR57FF
EH-CPU548 : WR000 to WRC3FF

5.2 List of Commands

[Legend]

Condition code

DER Data error (special internal output R7F4)
Set to “1” as an error when the I/O number is exceeded or when the BCD was abnormally, etc. When there is no data error, it is set to “0.”

ERR Error (special internal output R7F3)
Set to “1” when an error is generated when a control command and a special command are executed. The error code is set in WRF015. When there are no errors, the prior condition is maintained.

SD Shift data (special internal output R7F2)
Performs shift-in of the contents of SD by SHR or SHL command.

V Over flow (special internal output R7F1)
Indicates that a digit overflow has occurred and the signed data range is exceeded as a result of signed data operations.

C Carry (special internal output R7F0)
Indicates the contents of digit increase due to addition, digit decrease due to subtraction, and shift-out due to shifting.

● Maintains the prior status.

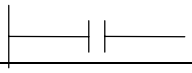
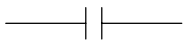
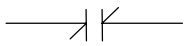
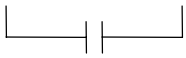
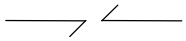
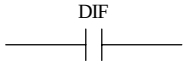
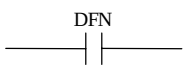
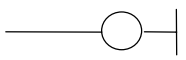
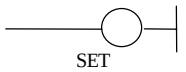
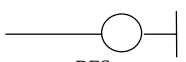
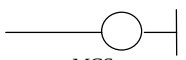
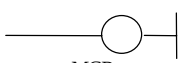
1] Set to “1” when there is an error in operation results. Prior status is maintained if there is no error.

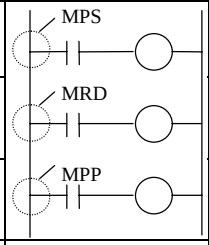
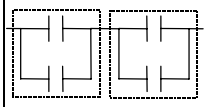
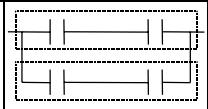
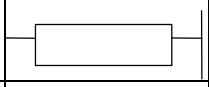
↕ Changes according to the operation result.

Processing time

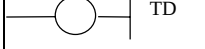
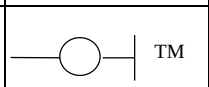
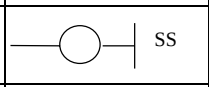
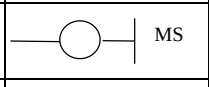
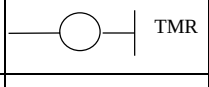
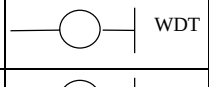
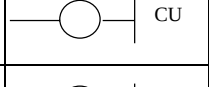
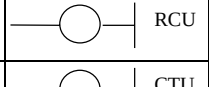
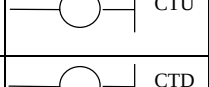
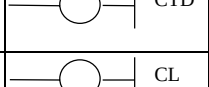
This indicates the command processing time (average value) of the EH-CPU448. Since the processing time of some commands changes depending on the parameter setting and the number of data, refer to the details of the command specifications.

1. Basic commands (sequence commands)

Classification	Item number	Ladder symbol	Command symbol	Command name	Process descriptions	I/O types used	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C	Process time (μ s) CPU5** CPU4** / CPU***A	Step	Remarks
Sequence command	1		LD	Logical operation start	Indicates the commencement of a-contact operation.	X, Y R000 to R7FF L0000 to L3FFF L10000 to L13FFF M0000 to M3FFF TD, SS, WDT, MS, TMR, CU, RCU, CT Timer: 0 to 255 Counter: 0 to 511 DIF0 to DIF511 DFN0 to DFN511	●	●	●	●	●	0.1 1.0	1	
	2		LDI	Logical negation operation start	Indicates the commencement of b-contact operation.							0.1 1.0		
	3		AND	Logical AND	Indicates a-contact series connection.							0.1 1.0		
	4		ANI	Logical NAND	Indicates b-contact series connection.							0.1 1.0		
	5		OR	Logical OR	Indicates a-contact parallel connection.		●	●	●	●	●	7.1 2.1	2	
	6		ORI	Logical NOR	Indicates b-contact parallel connection.							7.1 2.32		
	7		NOT	Logical NOT	Reverses all operation results up to that point.	None	●	●	●	●	●	0.2 2.14	2	
	8	DIF 	AND DIF	Rising edge detection	Indicates detection of the input rise.	DIF0 to DIF511 (Decimal)	●	●	●	●	●	0.3 3.05	3	Number overlap not allowed
		DIF 	OR DIF									7.3 4.05	4	
	9	DFN 	AND DFN	Falling edge detection	Indicates detection of the input fall.	DFN0 to DFN511 (Decimal)	●	●	●	●	●	0.3 3.05	3	Number overlap not allowed
		DFN 	OR DFN									7.3 4.05	4	
	10		OUT	Coil output	Indicates an output coil.	X, Y R000 to R7FF L0000 to L3FFF L10000 to L13FFF M0000 to M3FFF TD, SS, WDT, MS, TMR, CU, RCU, CTU, CTD, CL Timer: 0 to 255 Counter: 0 to 511	●	●	●	●	●	0.1 1.04	1	
	11	 SET	SET	Set coil output	Indicates a set output.	X, Y R000 to R7FF L0000 to L3FFF L10000 to L13FFF M0000 to M3FFF	●	●	●	●	●	0.1 2.07	1	
	12	 RES	RES	Reset coil output	Indicates a reset output.							0.1 2.12		
	13	 MCS	MCS	Set master control	Indicates a master control set operation.	MCS0 to MCS49	●	●	●	●	●	0.3 3.05	3	Number overlap allowed
	14	 MCR	MCR	Reset master control	Indicates a master control reset operation.	MCR0 to MCR49	●	●	●	●	●	0.2 2.15	2	Number overlap allowed

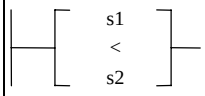
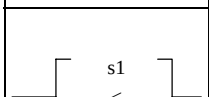
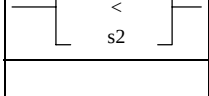
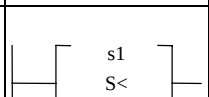
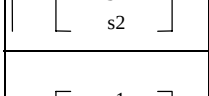
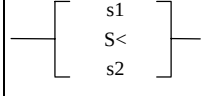
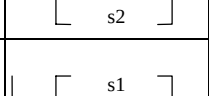
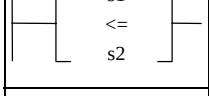
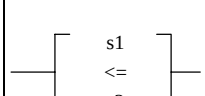
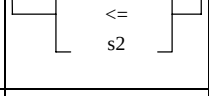
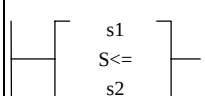
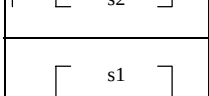
Classification	Item number	Ladder symbol	Command symbol	Command name	Process descriptions	I/O types used	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C	Process time (μ s) CPU5** CPU4** / CPU***A	Step	Remarks
Sequence command	15		MPS	Operation result push	Saves the operation result immediately prior.	None	●	●	●	●	●	—	0	
	16		MRD	Operation result read	Reads the saved operation result and continues operation.									
	17		MPP	Operation result pull	Reads the saved operation result, continues operation and clears the saved result.									
	18		ANB	Logical block series connection	Indicates series connection between two logical blocks.	None	●	●	●	●	●	—	0	
	19		ORB	Logical block parallel connection	Indicates parallel connection between two logical blocks.	None						7.0 / 0.95	1	
	20		[ ]	Processing box start and end	Indicates start and end of a process box.	None	●	●	●	●	●	0.3 / 2.15	3	
	21		( )	Relational box start and end	Indicates start and end of a comparison box.	None	●	●	●	●	●	—	0	

2. Basic commands (timer, counter)

Classification	Item number	Ladder symbol	Command symbol	Command name	Process descriptions	I/O types used	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C	Process time (μ s) CPU5** CPU4** / CPU***A	Step	Remarks
Timer	22		OUT TD	On delay timer	Indicates on delay timer operation.	TD0 to TD255 When 0.01 s, it is possible to use from 0 to 63.	●	●	●	●	●	26.4 / 28.05	5	Number overlap not allowed
	23		OUT SS	Expanded on delay timer	Indicates expanded on delay timer operation.	TM0 to TD2,047 0.01s time base (Fixed)	●	●	●	●	●	15.2 / -	5	
	24		OUT SS	Single shot	Indicates a single shot operation.	SS0 to SS255 0.01s time base for No.0 to 63.	●	●	●	●	●	21.42 / 23.05	5	
	25		OUT MS	Mono stable timer	Indicates a mono stable timer operation.	MS0 to MS255 0.01s time base for No.0 to 63.	●	●	●	●	●	21.4 / 23.05	5	
	26		OUT TMR	Integral timer	Indicates a integral timer operation.	TMR0 to TMR255 0.01s time base for No.0 to 63.	●	●	●	●	●	21.4 / 23.05	5	
	27		OUT WDT	Watchdog timer	Indicates a watchdog timer operation.	WDT0 to WDT255 0.01s time base for No.0 to 63.	●	●	●	●	●	27.6 / 32.3	7	
	28		OUT CU	Counter	Indicates a counter operation.	CU0 to CU511	●	●	●	●	●	13.4 / 15.05	5	
Counter	29		OUT RCU	Ring counter	Indicates a ring counter operation.	RCU0 to RCU511	●	●	●	●	●	13.4 / 15.05	5	
	30		OUT CTU	Up of up/down counter	Indicates an up operation of up-down counter.	CTU0 to CTU511	●	●	●	●	●	13.4 / 15.05	5	
	31		OUT CTD	Down of up/down counter	Indicates a down operation of up-down counter.	CTD0 to CTD511	●	●	●	●	●	12.2 / 14.05	3	
	32		OUT CL	Counter clear	Indicates a clear operation for CU, RCU, CTU, CTD and WDT.	CL0 to CL511	●	●	●	●	●	0.1 / 0.1	1	

3. Basic commands (relational box)

Classification	Item number	Ladder symbol	Command symbol	Command name	Process descriptions	I/O types used	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C	Process time (μ s) CPU5** CPU4** CPU***A	Step	Remarks
Relational box	32	s1 == s2	LD (s1== s2)	= Relational box	When s1 = s2: Continuity When s1 ≠ s2: Noncontinuity	[Word] WX, WY, WR, WL, WM, Timer · Counter [Double word] DX, DY, DR, DL, DM Constant	●	●	●	●	●	0.5 23.45 18.4 26.45	5 6 7 8	*1 *2 Upper case: W Lower case: DW
		s1 == s2	AND (s1== s2)									0.5 23.45 18.4 26.45		
		s1 == s2	OR (s1== s2)									7.5 23.45 25.4 26.45		
	33	s1 S== s2	LD (s1 S== s2)	Signed = Relational box	When s1 = s2: Continuity When s1 ≠ s2: Noncontinuity s1 and s2 are compared as signed 32-bit binary.	DX, DY, DR, DL, DM Constant	●	●	●	●	●	18.4 26.45	5 6 7 8	*2
		s1 S== s2	AND (s1 S== s2)									18.4 26.45		
		s1 S== s2	OR (s1 S== s2)									25.4 26.45		
	34	s1 <> s2	LD (s1< >s2)	<> Relational box	When s1 = s2: Noncontinuity When s1 ≠ s2: Continuity	[Word] WX, WY, WR, WL, WM, Timer · Counter [Double word] DX, DY, DR, DL, DM Constant	●	●	●	●	●	0.5 23.45 18.4 27.75	5 6 7 8	*1 *2 Upper case: W Lower case: DW
		s1 <> s2	AND (s1< >s2)									0.5 23.45 18.4 27.75		
		s1 <> s2	OR (s1< >s2)									7.5 23.45 25.4 27.75		
	35	s1 S<> s2	LD (s1 S<> s2)	Signed <> Relational box	When s1 = s2: Noncontinuity When s1 ≠ s2: Continuity s1 and s2 are compared as signed 32-bit binary.	DX, DY, DR, DL, DM Constant	●	●	●	●	●	18.4 27.75	5 6 7 8	*2
		s1 S<> s2	AND (s1 S<> s2)									18.4 27.75		
		s1 S<> s2	OR (s1 S<> s2)									25.4 27.75		

Classification	Item number	Ladder symbol	Command symbol	Command name	Process descriptions	I/O types used	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C	Process time (μ s) CPU5** CPU4** CPU***A	Step	Remarks
Relational box	36		LD (s1< s2)	< Relational box	When s1 < s2: Continuity When s1 ≥ s2: Noncontinuity	[Word] WX, WY, WR, WL, WM, Timer · Counter [Double word] DX, DY, DR, DL, DM Constant	●	●	●	●	●	0.5 23.45 19.4 28.75	5 6 7 8	*1 *2 Upper case: W Lower case: DW
			AND (s1< s2)									0.5 23.45 19.4 28.75		
			OR (s1< s2)									7.5 23.45 26.5 28.75		
	37		LD (s1 S< s2)	Signed < Relational box	When s1 < s2: Continuity When s1 ≥ s2: Noncontinuity s1 and s2 are compared as signed 32-bit binary.	DX, DY, DR, DL, DM Constant	●	●	●	●	●	19.4 24.75	5 6 7 8	*2
			AND (s1 S< s2)									19.4 28.75		
			OR (s1 S< s2)									26.4 28.75		
	38		LD (s1 <= s2)	<= Relational box	When s1 ≤ s2: Continuity When s1 > s2: Noncontinuity	[Word] WX, WY, WR, WL, WM, Timer · Counter [Double word] DX, DY, DR, DL, DM Constant	●	●	●	●	●	0.5 23.45 19.4 28.75	5 6 7 8	*1 *2 Upper case: W Lower case: DW
			AND (s1 <= s2)									0.5 23.45 19.4 28.75		
			OR (s1 <= s2)									7.5 23.45 26.5 28.75		
	39		LD (s1 S<= s2)	Signed <= Relational box	When s1 ≤ s2: Continuity When s1 > s2: Noncontinuity s1 and s2 are compared as signed 32-bit binary.	DX, DY, DR, DL, DM Constant	●	●	●	●	●	19.4 28.75	5 6 7 8	*2
			AND (s1 S<= s2)									19.4 28.75		
			OR (s1 S<= s2)									26.4 28.75		

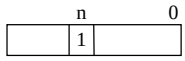
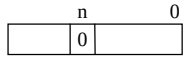
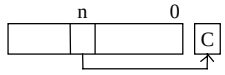
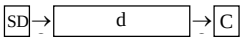
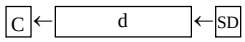
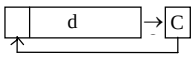
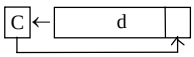
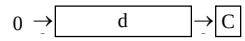
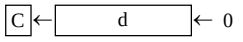
4. Arithmetic command

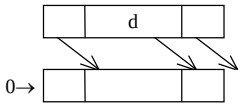
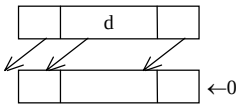
Classification	Item number	Ladder symbol	Command symbol	Command name	Process descriptions	I/O types used	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C	Process time (μ s) CPU5** CPU4** CPU***A	Step	Remarks
Substitution statement	1	d=s		Substitution statement	d ← s	[Bit] d: Y, R, L, M s: X, Y, R, L, M, Constant	↕	●	●	●	●	0.3 15.6	3	I/O: I/O
												31 31.9	4	I/O: Array
												27 27.9	4	Array: I/O
												43 40.05	5	Array: Array
						[Word] d: WY, WR, WL, WM, Timer · Counter s: WX, WY, WR, WL, WM, Timer · Counter, Constant	↕	●	●	●	●	0.3 14.6	3	I/O: I/O
												27 29.9	4	I/O: Array
												25 26.9	4	Array: I/O
												39 37.9	5	Array: Array
						[Double word] d: DY, DR, DL, DM s: DX, DY, DR, DL, DM, Constant * Array variables can be used.	↕	●	●	●	●	15 19.9	4	I/O: I/O
												29 36.55	4	I/O: Array
												27 32.55	5	Array: I/O
												42 46.55	5	Array: Array
Mathematical operations	2	d=s1+s2		Binary addition	d ← s1+s2	[Word] d: WY, WR, WL, WM s1, s2: WX, WY, WR, WL, WM, Timer · Counter, Constant	●	●	●	↕	↕	0.4 19.9	4	Upper case: W Lower case: DW
												22 30.5	6	
	3	d=s1 B+ s2		BCD addition	d ← s1+s2		↕	●	●	●	↕	36 40.9	4	Upper case: W Lower case: DW
												59 68.5	6	
	4	d=s1−s2		Binary subtraction	d ← s1−s2	[Double word] d: DY, DR, DL, DM s1, s2: DX, DY, DR, DL, DM, Constant	●	●	●	↕	↕	0.4 19.9	4	Upper case: W Lower case: DW
												20 30.5	6	
	5	d=s1 B− s2		BCD subtraction	d ← s1−s2		↕	●	●	●	↕	34 38.9	4	Upper case: W Lower case: DW
												54 62.5	6	

Classification	Item number	Ladder symbol	Command symbol	Command name	Process descriptions	I/O types used	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C	Process time (μ s) CPU5** CPU4** CPU***A	Step	Remarks
Mathematical operations	6	d=s1×s2		Binary multiplication	d ← s1×s2	[Word] d: WY, WR, WL, WM s1, s2: WX, WY, WR, WL, WM, Timer · Counter, Constant [Double word] d: DY, DR, DL, DM s1, s2: DX, DY, DR, DL, DM, Constant	↕	●	●	●	●	30 30.9 39 44.5	4 6	Upper case: W Lower case: DW
	7	d=s1 B× s2		BCD multiplication	d ← s1×s2		↕	●	●	●	●	63 63.9 132 136.5	4 6	Upper case: W Lower case: DW
	8	d=s1 S× s2		Signed binary multiplication	d ← s1×s2	[Double word] d: DY, DR, DL, DM s1, s2: DX, DY, DR, DL, DM, Constant	↕	●	●	●	●	42 47.5	6	
	9	d=s1 / s2		Binary division	[Word] d ← s1 / s2 WRF016 ← s1 mod s2 [Double word] d ← s1 / s2 DRF016 ← s1 mod s2	[Word] d: WY, WR, WL, WM s1, s2: WX, WY, WR, WL, WM, Timer · Counter, Constant [Double word] d: DY, DR, DL, DM s1, s2: DX, DY, DR, DL, DM, Constant	↕	●	●	●	●	28 32.9 32 41.5	4 6	Upper case: W Lower case: DW
	10	d=s1 B/ s2		BCD division								50 54.9 69 77.5	4 6	Upper case: W Lower case: DW
	11	d=s1 S/ s2		Signed binary division		[Double word] d: DY, DR, DL, DM s1, s2: DX, DY, DR, DL, DM, Constant	↕	●	●	↕	●	35 43.5	6	
Logical operation	12	d=s1 OR s2		Logical OR	d ← s1+s2	[Bit] d: Y, R, L, M s1, s2: X, Y, R, L, M [Word] d: WY, WR, WL, WM, Timer · Counter s1, s2: WX, WY, WR, WL, WM, Timer · Counter, Constant [Double word] d: DY, DR, DL, DM s1, s2: DX, DY, DR, DL, DM, Constant	●	●	●	●	●	0.4 18.9 0.4 21.2 18 24.2	4 4 6	Upper case: B Middle case: W Lower case: DW
	13	d=s1 AND s2		Logical AND	d ← s1 · s2		●	●	●	●	●	0.4 18.9 0.4 17.9 18 27.5	4 4 6	Upper case: B Middle case: W Lower case: DW
	14	d=s1 XOR s2		Exclusive OR	d ← s1 + s2		●	●	●	●	●	0.4 18.9 0.4 17.9 18 27.5	4 4 6	Upper case: B Middle case: W Lower case: DW

Classification	Item number	Ladder symbol	Command symbol	Command name	Process descriptions	I/O types used	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C	Process time (μ s) CPU5** CPU4** / CPU***A	Step	Remarks
Relational expression	15	d=s1 == s2		= Relational expression	When s1 = s2, d ← 1 When s1 ≠ s2, d ← 0	[Word] d: Y, R, L, M s1, s2: WX, WY, WR, WL, WM, Timer · Counter, Constant [Double word] d: Y, R, L, M s1, s2: DX, DY, DR, DL, DM, Constant	●	●	●	●	●	0.4 18.9 18 26.5	4 6	Upper case: W Lower case: DW
	16	d=s1 S== s2		Signed = Relational expression	When s1 = s2, d ← 1 When s1 ≠ s2, d ← 0 s1 and s2 are compared as signed 32-bit binary.	[Double word] d: Y, R, L, M s1, s2: DX, DY, DR, DL, DM, Constant						18 26.5	6	
	17	d=s1<>s2		<> Relational expression	When s1 = s2, d ← 0 When s1 ≠ s2, d ← 1	[Word] d: Y, R, L, M s1, s2: WX, WY, WR, WL, WM, Timer · Counter, Constant [Double word] d: Y, R, L, M s1, s2: DX, DY, DR, DL, DM, Constant	●	●	●	●	●	0.4 18.9 18 26.5	4 6	Upper case: W Lower case: DW
	18	d=s1 S<> s2		Signed <> Relational expression	When s1 = s2, d ← 0 When s1 ≠ s2, d ← 1 s1 and s2 are compared as signed 32-bit binary.	[Double word] d: Y, R, L, M s1, s2: DX, DY, DR, DL, DM, Constant						18 26.5	6	
	19	d=s1<s2		< Relational expression	When s1 < s2, d ← 1 When s1 ≥ s2, d ← 0	[Word] d: Y, R, L, M s1, s2: WX, WY, WR, WL, WM, Timer · Counter, Constant [Double word] d: Y, R, L, M s1, s2: DX, DY, DR, DL, DM, Constant	●	●	●	●	●	0.4 18.9 19 26.5	4 6	Upper case: W Lower case: DW
	20	d=s1 S< s2		Signed < Relational expression	When s1 = s2, d ← 1 When s1 ≠ s2, d ← 0 s1 and s2 are compared as signed 32-bit binary.	[Double word] d: Y, R, L, M s1, s2: DX, DY, DR, DL, DM, Constant						19 26.5	6	
	21	d=s1 <= s2		≤ Relational expression	When s1 ≤ s2, d ← 1 When s1 > s2, d ← 0	[Word] d: Y, R, L, M s1, s2: WX, WY, WR, WL, WM, Timer · Counter, Constant [Double word] d: Y, R, L, M s1, s2: DX, DY, DR, DL, DM, Constant	●	●	●	●	●	0.4 18.9 19 26.5	4 6	Upper case: W Lower case: DW
	22	d=s1 S<= s2		Signed ≤ Relational expression	When s1 ≤ s2, d ← 1 When s1 > s2, d ← 0 s1 and s2 are compared as signed 32-bit binary.	[Double word] d: Y, R, L, M s1, s2: DX, DY, DR, DL, DM, Constant						19 26.5	6	

5. Application commands

Classification	Item number	Ladder symbol	Command symbol	Command name	Process descriptions	I/O types used	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C	Process time (μ s) CPU5** CPU4** CPU***A	Step	Remarks
Bit operations	1	BSET(d, n)		Bit set	 Sets 1 to bit n.	[Word] d: WY, WR, WL, WM, TC n(0-15): WX, WY, WR, WL, WM, TC, Constant [Double word] d: DY, DR, DL, DM n(0-31): WX, WY, WR, WL, WM, TC, Constant	●	●	●	●	●	13 14.6 15 16.6	3	Upper case: W Lower case: DW
	2	BRES(d, n)		Bit reset	 Sets 0 to bit n.		●	●	●	●	●	13 14.6 15 16.6	3	Upper case: W Lower case: DW
	3	BTS(d, n)		Bit test	 Acquires the value in bit n to C (R7F0).		●	●	●	●	↕	13 15.6 15 17.6	3	Upper case: W Lower case: DW
Shift/rotate	4	SHR(d, n)		Shift right	 Shifts right by n bits.	[Word] d: WY, WR, WL, WM, TC n: WX, WY, WR, WL, WM, TC, Constant [Double word] d: DY, DR, DL, DM n: WX, WY, WR, WL, WM, TC, Constant *C: R7F0 SD: R7F2	●	●	●	●	↕	17 18.6 18 20.6	3	Upper case: W Lower case: DW
	5	SHL(d, n)		Shift left	 Shifts left by n bits.		●	●	●	●	↕	16 18.6 18 20.6	3	Upper case: W Lower case: DW
	6	ROR(d, n)		Rotate right	 Rotates right by n bits.		●	●	●	●	↕	21 22.6 22 24.6	3	Upper case: W Lower case: DW
	7	ROL(d, n)		Rotate left	 Rotates left by n bits.		●	●	●	●	↕	21 22.6 22 24.6	3	Upper case: W Lower case: DW
	8	LSR(d, n)		Logical shift right	 Shifts right by n bits.		●	●	●	●	↕	16 17.6 18 19.6	3	Upper case: W Lower case: DW
	9	LSL(d, n)		Logical shift left	 Shifts left by n bits.		●	●	●	●	↕	16 17.6 18 19.6	3	Upper case: W Lower case: DW

Classification	Item number	Ladder symbol	Command symbol	Command name	Process descriptions	I/O types used	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C	Process time (μ s) CPU5** CPU4** CPU***A	Step	Remarks
Shift/rotate	10	BSR(d, n)		BCD shift right	 Shifts BCD to right by n digits.	[Word] d: WY, WR, WL, WM, TC n: WX, WY, WR, WL, WM, TC, Constant	●	●	●	●	●	15 16.6 17 19.6	3	Upper case: W Lower case: DW
	11	BSL(d, n)		BCD shift left	 Shifts BCD to left by n digits.	[Double word] d: DY, DR, DL, DM n: WX, WY, WR, WL, WM, TC, constant	●	●	●	●	●	15 16.6 17 19.6	3	Upper case: W Lower case: DW
Transfer	12	WSHR(d, n)		Batch shift right	Shifts n bits (or words) starting with I/O number d to the right by 1 bit (or word).	[Bit] d: R, L, M n(0-255): WX, WY, WR, WL, WM, TC, Constant	↕	●	●	●	●	31 28.9 24 23.26	3	*3 Upper case: B Lower case: W
	13	WSHL(d, n)		Batch shift left	Shifts n bits (or words) starting with I/O number d to the left by 1 bit (or word).	[Word] d: WR, WL, WM n(0-255):WX, WY, WR, WL, WM, TC, Constant	↕	●	●	●	●	31 28.7 25 24.35	3	*3 Upper case: B Lower case: W
	14	WBSR(d, n)		Batch BCD shift right	Shifts n digits of BCD starting with I/O number d to the right by 1 digit.	[Word] d: WR, WL, WM n(0-255):WX, WY, WR, WL, WM, TC, Constant	↕	●	●	●	●	25 24.1	3	*3
	15	WBSL(d, n)		Batch BCD shift left	Shifts n digits of BCD starting with I/O number d to the left by 1 digit.		↕	●	●	●	●	25 24.55	3	*3
	16	MOV(d, s, n)		Block transfer	Transfers (copies) n bits (or words) of data starting with I/O number s, to the n bit (or word) range starting with I/O number s.	[Bit] d, s: R, L, M n(0-255): WX, WY, WR, WL, WM, TC, Constant [Word] d, s: WR, WL, WM n(0-255):WX, WY, WR, WL, WM, TC, Constant	↕	●	●	●	●	87 59.9 83 62.2	4	*3 Upper case: B Lower case: W
	17	COPY(d, s, n)		Copy	Copies the bit (or word) data of I/O number s to the n bit (or word) range starting with I/O number d.	[Bit] d: R, L, M s: X, Y, R, L, M, Constant n(0-255): WX, WY, WR, WL, WM, TC, Constant [Word] d: WR, WL, WM s, n(0-255): WX, WY, WR, WL, WM, TC, Constant	↕	●	●	●	●	43 47.48 93 70.28	4	*3 Upper case: B Lower case: W
Negation / Two's complement / Sign	18	XCG(d1, d2, n)		Block exchange	Exchanges the n bit (or word) range starting with I/O number d1 and the n bit (or word) range starting with I/O number d2.	[Bit] d1, d2: R, L, M n(0-255): WX, WY, WR, WL, WM, TC, Constant [Word] d: WR, WL, WM n(0-255): WX, WY, WR, WL, WM, TC, Constant	↕	●	●	●	●	47 44.73 42 41.3	4	*3 Upper case: B Lower case: W

Classification	Item number	Ladder symbol	Command symbol	Command name	Process descriptions	I/O types used	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C	Process time (μ s) CPU5** CPU4** CPU*** A	Step	Remarks
Negation / Two's complement / Sign	19	NOT(d)		Reverse	Reverses the bit for the I/O number d value.	[Bit] Y, R, L, M [Word] WY, WR, WL, WM [Double word] DY, DR, DL, DM	●	●	●	●	●	13 11.3 11 10.3 13 12.3	2	Upper case: B Middle case: W Lower case: DW
	20	NEG(d)		Two's complement	Stores two's complement of the value stored in I/O number d, in d.	[Word] WY, WR, WL, WM [Double word] DY, DR, DL, DM	●	●	●	●	●	12 11.3 14 13.3	2	Upper case: W Lower case: DW
	21	ABS(d, s)		Absolute value	Stores the absolute value of s in d, and the sign value of s in carry (R7F0). (0: Positive, 1: Negative)	[Word] d: WY, WR, WL, WM s: WX, WY, WR, WL, WM, TC, Constant [Double word] d: DY, DR, DL, DM s: DX, DY, DR, DL, DM, Constant	●	●	●	●	↕	13 14.75 16 20.55	3 4	Upper case: W Lower case: DW
	22	SGET(d, s)		Sign addition	If the value of carry (R7F0) is 0, the value is stored in d, and if it is 1, two's complement of the s is stored in d.		●	●	●	●	●	13 14.75 16 19.55	3 4	Upper case: W Lower case: DW
	23	EXT(d, s)		Sign expansion	Copies the sign bit value of s to all bits in the upper word of d, and stores the value of s in the lower word of d.	d: DY, DR, DL, DM s: WX, WY, WR, WL, WM, TC, Constant	●	●	●	●	●	15 16.75	3	
Conversion	24	BCD(d, s)		Binary → BCD conversion	Converts the value of s into BCD and stores it in I/O number d. If there the value of s is erroneous, DER (R7F4) = 1 is set.	[Word] d: WY, WR, WL, WM s: WX, WY, WR, WL, WM, TC, Constant	↕	●	●	●	●	28 29.75 31 35.2	3 4	Upper case: W Lower case: DW
	25	BIN(d, s)		BCD → Binary conversion	Converts the value of s into binary and stores it in I/O number d. If there the value of s is erroneous, DER (R7F4) = 1 is set.	[Double word] d: DY, DR, DL, DM s: DX, DY, DR, DL, DM, Constant	↕	●	●	●	●	19 20.75 27 32.2	3 4	Upper case: W Lower case: DW
	26	DECO(d, s, n)		Decode	Decodes the value indicated by the least significant n bits of s, and sets the bit that corresponds to the decoding result of the bit row starting from I/O number d, to 1.	d: R, L, M s: WX, WY, WR, WL, WM, TC, Constant n: Constant(1-8)	↕	●	●	●	●	38 37.55	4	n=1
	27	ENCO(d, s, n)		Encode	Encodes the bit location in which 1 is set within the bit row, which starts with I/O number s and lasts for the amount of nth power of 2, and stores it in I/O number d. If multiple bits that contain 1 exist, the one with the upper bit locations will be encoded.	d: WY, WR, WL, WM s: R, L, M n: Constant(1-8)	↕	●	●	●	↕	38 41.3	4	n=1
	28	SEG(d, s)		7 segment decode	Coverts the value of s as a 1-digit 4-bit value to a 4-digit 7-segment display code and stores it in d.	d: DY, DR, DL, DM s: WX, WY, WR, WL, WM, TC, Constant	●	●	●	●	●	16 17.75	3	

Classification	Item number	Ladder symbol	Command symbol	Command name	Process descriptions	I/O types used	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C	Process time (μ s) CPU5** CPU4** / CPU**A	Step	Remarks
Application commands	29	SQR(d, s)		Square root	Calculates the square root of the value of s (8-digit BCD) and stores it in d (4-digit BCD).	d: WY, WR, WL, WM s: DX, DY, DR, DL, DM, Constant	↕	●	●	●	●	86 90.2	4	
	30	BCU(d, s)		Bit count	Among the contents of s (word, double-word), stores the number of bits that are set to 1 in I/O number d.	[Word] d: WY, WR, WL, WM s: WX, WY, WR, WL, WM, TC, Constant [Double word] d: WY, WR, WL, WM s: DX, DY, DR, DL, DM, Constant	●	●	●	●	●	14 15.75 17 21.2	3 4	Upper case: W Lower case: DW
	31	SWAP(d)		Swap	Swaps the upper 8 bits and the lower 8 bits of the value (word) for I/O number d.	d: WY, WR, WL, WM	●	●	●	●	●	12 11.3	2	
	32	FIFIT(P, n)		FIFO initialization	Stores the value of n in the FIFO size area (P) and stores 0 in the FIFO used number area (P+1).	P: WR, WL, WM n: Constant(0-255)	↕	●	●	●	●	24 23.6	3	
	33	FIFWR(P, s)		FIFO write	Stores the value of I/O number s in the FIFO write location and adds 1 to the FIFO used number area (P+1).	P: WR, WL, WM s: WX, WY, WR, WL, WM, TC, Constant	↕	●	●	●	●	25 22.6	3	
	34	FIFRD(P, d)		FIFO read	Reads data from the FIFO read location and stores it in d, then shifts the data in the FIFO by one data item and subtracts 1 from the value of the FIFO usage number area (P+1).	P: WR, WL, WM d: WY, WR, WL, WM, TC	↕	●	●	●	●	157 156.6	3	*4
	35	UNIT(d, s, n)		Unit	Stores the lower 4 bit values of the n words starting with s in the lower 4 bits each of d (word).	d:WY, WR, WL, WM s: WR, WL, WM n: Constant(0-4)	↕	●	●	●	●	30 31.9	4	
	36	DIST(d, s, n)		Distribute	Extracts the value of s (word) in 4 bit units starting with the least significant bits, and sets them in the lower 4 bits of each word starting with I/O number d (word). The upper bits are set to 0.	d: WR, WL, WM s: WX, WY, WR, WL, WM, TC, Constant n: Constant(0-4)	↕	●	●	●	●	27 29.05	4	
	37	ADRIO(d, s)		I/O address conversion	Stores the actual address of the I/O designated by s in d.	[Bit] s: X, Y, R, L, M d: WY, WR, WL, WM [Word] s: WX, WY, WR, WL, WM d: WY, WR, WL, WM	●	●	●	●	●	12 14.6	3	*5

6. Control commands

Classification	Item number	Ladder symbol	Command symbol	Command name	Process descriptions	I/O types used	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C	Process time (μ s) CPU5** CPU4** / CPU***A	Step	Remarks
Cntrol commands	1	END		Normal scan end	Indicates the end of a normal scan and re-executes normal scan from the head of the normal scan.	None	●	●	●	●	●	194 / 194.95	1	
	2	CEND(s)		Scan conditional end	Re-executes normal scan from the head of the normal scan when s=1, while the next command is executed when s=0.	s: X, Y, R, L, M	●	●	●	●	●	6 / 196 / 202.5	2	*6 / *7
	3	JMP n		Unconditio-nal jump	Jumps to LBL n of the same No. n.	n: Constant(0-255)	●	1]	●	●	●	21 / 22	2	
	4	CJMP n (s)		Conditional jump	When s=1, jumps to the LBL n of the same No. n when s=0, executes the next command.	n: Constant(0-255) s: X, Y, R, L, M	●	1]	●	●	●	6 / 22 / 26.45	3	*6 / *7
	5	LBL n		Label	Indicates the jump destination of JMP or CJMP of the same No. n.	n: Constant(0-255)	●	●	●	●	●	0.1 / 0.65	1	
	6	FOR n (s)		FOR	When s=0, jumps to the location after the NEXT n of the same No. n when s is not 0, executes the next command.	n: Constant(0-49) s: WY, WR, WL, WM	●	1]	●	●	●	29 / 37	3	
	7	NEXT n		NEXT	Subtracts 1 from the s value of the FOR n of the same No. and jumps to FOR n.	n: Constant(0-49)	●	1]	●	●	●	23 / 25	2	
	8	CAL n		Call subroutine	Executes the SB n subroutine of the same No. n.	n: Constant(0-99)	●	1]	●	●	●	19 / 20	2	
	9	SB n		Start subroutine	Indicates the start of No. n subroutine.	n: Constant(0-99)	●	●	●	●	●	0.1 / 0.65	1	
	10	RTS		RETURN SUBROUTIN	Returns from subroutine.	None	●	●	●	●	●	17 / 16.7	1	
	11	INT n		Start interrupt scan	Indicates the start of No. n interrupt scan.	n: Constant(0-2)	●	●	●	●	●	0.1 / 0.65	1	
	12	RTI		RETURN INTERRUPT	Returns from interrupt scan.	None	●	●	●	●	●	8 / 5	1	

7. High-function module transfer commands

Classification	Item number	Ladder symbol	Command symbol	Command name	Process descriptions	I/O types used	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C	Process time (μ s) CPU5** CPU4** / CPU***A	Step	Remarks
High-function module transfer commands	1	TRNS 0 (d, s, t)		General-purpose port transmission command	Transmits data from the CPU's general-purpose port.	t: R, L, M d: WY(Dummy I/O) s: WR, WL, WM	↕	●	●	●	●	61 / 57	5	
	2	RECV 0 (d, s, t)		General-purpose port receiving command	Receives data at the CPU's general-purpose port.	t: R, L, M d: WX(Dummy I/O) s: WR, WL, WM	↕	●	●	●	●	61 / 57	5	
		TRNS 8 (d, s, t) Note.1) Not supported by EH-CPU104(A)		Telecommunication command	Controls the modem with the CPU's ladder program.	t: R, L, M d: WY(Dummy I/O) s: WR, WL, WM	↕	●	●	●	●	128 / 68	5	

8. FUN commands

Classification	Item number	Ladder symbol	Command symbol	Command name	Process descriptions	I/O types used	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C	Process time (μ s) CPU5** / CPU4** / CPU**A	Step	Remarks
FUN commands	1	FUN 0 (s) (PIDIT (s))		PID operation initialization	Initializes the area for PID operation.	s: WR	●	●	●	●	●	1,661 / 2979.9	3	Number of loops = 1 *8
	2	FUN 1 (s) (PIDOP (s))		PID operation execution control	Performs control for PID operation execution.		●	●	●	●	●	83 / 85.4	3	*8
	3	FUN 2 (s) (PIDCL (s))		PID operation calculation	Executes PID operation.		●	●	●	●	●	87 / 91.4	3	Number of loops = 1 *8
	4	FUN 4 (s) (IFR (s))		Process stepping	Performs the process stepping processing.	s: WR,WL,WM	↕	●	●	●	●	208 / 191.4	3	
	5	FUN 10 (s) (SIN (s))		SIN function	Calculates the SIN of the value designated by s and stores the result in s+1, s+2.		↕	●	●	●	●	39 / 40.4	3	
	6	FUN 11 (s) (COS (s))		COS function	Calculates the COS of the value designated by s and stores the result in s+1, s+2.		↕	●	●	●	●	40 / 41.4	3	
	7	FUN 12 (s) (TAN (s))		TAN function	Calculates the TAN of the value designated by s and stores the result in s+1, s+2.		↕	●	●	●	●	40 / 41.4	3	
	8	FUN 13 (s) (ASIN (s))		ARC SIN function	Calculates the ARC SIN of the value designated by s (fractional portion) and s+1 (integer portion), and stores the result in s+2.		↕	●	●	●	●	62 / 63.4	3	
	9	FUN 14 (s) (ACOS (s))		ARC COS function	Calculates the ARC COS of the value designated by s (fractional portion) and s+1 (integer portion), and stores the results in s+2.		↕	●	●	●	●	63 / 64.4	3	
	10	FUN 15 (s) (ATAN (s))		ARC TAN function	Calculates the ARC TAN of the value designated by s (fractional portion) and s+1 (integer portion), and stores the results in s+2.		↕	●	●	●	●	41 / 42.4	3	
	11	FUN 20 (s)		Data search	Requested data is searched in assigned I/O table.		↕	●	●	●	●	86.2 / -		
	12	FUN 21 (s)		Table search	Requested data block is taken from assigned I/O block table, and stored assigned address.		↕	●	●	●	●	104.2 / -		
	13	FUN 22 (s)		Check code calculation	Check code for sending serial communication message is calculated and created.		↕	●	●	●	●	783.2 / -		
	14	FUN 23 (s)		Check code verifying	Check code for receiving serial communication message is verified.		↕	●	●	●	●	790.2 / -		
	15	FUN 30 (s) (BINDA (s))		BIN → ASCII conversion (16 bits)	Converts 16-bit unsigned binary data to a decimal ASCII code, then stores it.		↕	●	●	●	●	140 / 141.4	3	
	16	FUN 31 (s) (DBINDA (s))		BIN → ASCII conversion (32 bits)	Converts 32-bit unsigned binary data to a decimal ASCII code, then stores it.		↕	●	●	●	●	199 / 200.4	3	
	17	FUN 32 (s) (BINHA (s))		BIN → ASCII conversion (16 bits)	Converts 16-bit unsigned binary data to an ASCII code, then stores it.		↕	●	●	●	●	124 / 125.4	3	
	18	FUN 33 (s) (DBINHA (s))		BIN → ASCII conversion (32 bits)	Converts 32-bit unsigned binary data to an ASCII code, then stores it.		↕	●	●	●	●	168	3	
	19	FUN 34 (s) (BCDDA (s))		BIN → ASCII conversion (16 bits)	Converts 16-bit BCD (BCD 4-digit) data to an ASCII code, then stores it.		↕	●	●	●	●	112 / 112.4	3	

Classification	Item number	Ladder symbol	Command symbol	Command name	Process descriptions	I/O types used	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C	Process time (μ s) CPU5** CPU4** CPU**A	Step	Remarks
	20	FUN 35 (s) (DBCDDA (s))		BIN → ASCII conversion (32 bits)	Converts 32-bit BCD (BCD 8-digit) data to an ASCII code, then stores it.	s: WR,WL,WM	↕	●	●	●	●	171 / 171.4	3	
	21	FUN 36 (s) (DABIN (s))		ASCII → BIN conversion (16 bits)	Converts unsigned BCD 5-digit data to an ASCII code, then stores it.		↕	●	●	●	●	60 / 61.4	3	
	22	FUN 37 (s) (DDABIN (s))		ASCII → BIN conversion (32 bits)	Converts signed BCD 10-digit data to an ASCII code, then stores it.		↕	●	●	●	●	97 / 98.4	3	
	23	FUN 38 (s) (HABIN (s))		ASCII → BIN conversion (16 bits)	Converts a 4-digit hexadecimal ASCII code to 16-bit binary data, then stores it.		↕	●	●	●	●	63 / 64.4	3	
	24	FUN 39 (s) (DHABIN (s))		ASCII → BIN conversion (32 bits)	Converts a 8-digit hexadecimal ASCII code to 32-bit binary data, then stores it.		↕	●	●	●	●	94 / 94.4	3	
	25	FUN 40 (s) (DABCD (s))		ASCII → BIN conversion (16 bits)	Converts a 4-digit ASCII code to 4-digit BCD data, then stores it.		↕	●	●	●	●	64 / 65.4	3	
	26	FUN 41 (s) (DDABCD (s))		ASCII → BIN conversion (32 bits)	Converts a 8-digit ASCII code to 8-digit BCD data, then stores it.		↕	●	●	●	●	93 / 94.4	3	
	27	FUN 42 (s) (ASC (s))		BIN → ASCII conversion (designated)	Converts binary data to an ASCII code of the designated number of characters, then stores it.		↕	●	●	●	●	119 / 116.4	3	
	28	FUN 43 (s) (HEX (s))		ASCII → BIN conversion (designated)	Converts an ASCII code of the designated number of characters to binary data, then stores it.		↕	●	●	●	●	191 / 188.4	3	
	29	FUN 44 (s) (SADD (s))		Merge character strings	Merges the designated character stings (up to NULL), then stores it in the I/O at the designated position.		↕	●	●	●	●	196 / 184.4	3	
	30	FUN 45 (s) (SCMP (s))		Compare character strings	Compares the designated character stings (up to NULL), then stores the comparison result.		↕	●	●	●	●	157 / 147.4	3	
	31	FUN 46 (s) (WTOB (s))		Word → byte conversion	Divides 16-bit word data, converts it to 8-bit byte data, then stores it.		↕	●	●	●	●	115 / 111.4	3	
	32	FUN 47 (s) (BTOW (s))		Byte → word conversion	Divides 8-bit byte data, merges it into 16-bit word data, then stores it.		↕	●	●	●	●	109 / 104.4	3	
FUN commands	33	FUN 48 (s) (BSHR (s))		Right-shift byte unit	Shifts the designated data string to the right for the number of the designated bytes (8 bits*n).		↕	●	●	●	●	81 / 79.4	3	
	34	FUN 49 (s) (BSHL (s))		Left-shift byte unit	Shifts the designated data string to the left for the number of the designated bytes (8 bits*n).		↕	●	●	●	●	85 / 84.4	3	
	35	FUN 60 (s)		Binary square root	Calculates the square root of a 32-byte binary value.		↕	●	●	●	●	453 / 454.4	3	
	36	FUN 61 (s)		Dynamic scan pulse	Repeats on and off operations for the number of the designated scans.		↕	●	●	●	●	73 / 70.4	3	
	37	FUN 80 (s) (ALREF (s))		I/O refresh (all points)	Refreshes all external I/O ranges.		↕	●	●	●	●	69 / 41.4	3	

Classification	Item number	Ladder symbol	Command symbol	Command name	Process descriptions	I/O types used	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C	Process time (μ s) CPU5** CPU4** CPU**A	Step	Remarks
FUN commands	38	FUN 81 (s) (IOREF (s))		I/O refresh (I/O /link designation)	Refreshes only the input range, output range or link range.	s: WR,WL,WM	↕	●	●	●	●	44 45.4	3	
	39	FUN 82 (s) (SLREF (s))		I/O refresh (any slot)	Refreshes the I/O at the designated slot.		↕	●	●	●	●	80 79.4	3	
	40	FUN 100 (s) (INTW (s))		Floating point operation (Real number to integer)	Real number to integer (Word) conversion.		↕	●	●	●	●	57	3	*8
	41	FUN 101 (s) (INTD (s))		Floating point operation (Real number to integer)	Real number to integer (Double word) conversion.		↕	●	●	●	●	69 69.4	3	*8
	42	FUN 102 (s) (FLOAT (s))		Floating point operation (integer to Real number)	Integer (word) to real number conversion.		↕	●	●	●	●	43 43.4	3	*8
	43	FUN 103 (s) (FLOATD (s))		Floating point operation (Integer to real number)	Integer (Double word) to real number conversion.		↕	●	●	●	●	47 47.4	3	*8
	44	FUN 104 (s) (FADD (s))		Floating point operation (addition)	The addition of the real number.		↕	●	●	●	●	70 70.4	3	*8
	45	FUN 105 (s) (FSUB (s))		Floating point operation (subtraction)	The subtraction of the real number.		↕	●	●	●	●	70 70.4	3	*8
	46	FUN 106 (s) (FMUL (s))		Floating point operation (multiplication)	The multiplication of the real number.		↕	●	●	●	●	69 70.4	3	*8
	47	FUN 107 (s) (FDIV (s))		Floating point operation (division)	The division of the real number.		↕	●	●	●	●	119 119.4	3	*8
	48	FUN 108 (s) (FRAD (s))		Floating point operation (radian conversion)	Angle to radian conversion.		↕	●	●	●	●	63 64.4	3	*8
	49	FUN 109 (s) (FDEG (s))		Floating point operation (angle conversion)	Radian to angle conversion.		↕	●	●	●	●	63 64.4	3	*8
	50	FUN 110 (s) (FSIN (s))		Floating point operation (calculate the SIN)	Calculates the SIN of the floating point number.		↕	●	●	●	●	282 282.4	3	*8
	51	FUN 111 (s) (FCOS (s))		Floating point operation (calculate the COS)	Calculates the COS of the floating point number.		↕	●	●	●	●	303 304.4	3	*8
	52	FUN 112 (s) (FTAN (s))		Floating point operation (calculate the TAN)	Calculates the TAN of the floating point number.		↕	●	●	●	●	343 341.4	3	*8
	53	FUN 113 (s) (FASIN (s))		Floating point operation (calculate the ARC SIN)	Calculates the ARC SIN of the floating point number.		↕	●	●	●	●	200 200.4	3	*8
	54	FUN 114 (s) (FACOS (s))		Floating point operation (calculate the ARC COS)	Calculates the ARC COS of the floating point number.		↕	●	●	●	●	192 191.4	3	*8
	55	FUN 115 (s) (FATAN (s))		Floating point operation (calculate the ARC TAN)	Calculates the ARC TAN of the floating point number.		↕	●	●	●	●	394 395.4	3	*8
	56	FUN 116 (s) (FSQR (s))		Floating point operation (calculate the square root)	Calculates the square root of the floating point number.		↕	●	●	●	●	481 482.4	3	*8
	57	FUN 117 (s) (FEXP (s))		Floating point operation (calculate the exponent)	Calculates the exponent of the floating point number.		↕	●	●	●	●	329 330.4	3	*8
	58	FUN 118 (s) (FLOG (s))		Floating point operation (logarithm)	Calculates the logarithm of the floating point number.		↕	●	●	●	●	187 187.4	3	*8

Classification	Item number	Ladder symbol	Command symbol	Command name	Process descriptions	I/O types used	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C	Process time (μ s) CPU5** CPU4** CPU**A	Step	Remarks
FUN commands	59	FUN 120 (s) (INDXD (s))		Index setting (argument d)	Sets the argument used as an index in respect to argument d of the MOV and COPY commands.	s: WR,WL,WM	●	●	●	●	●	18 22.4	3	
	60	FUN 121 (s) (INDXS (s))		Index setting (argument s)	Sets the argument used as an index in respect to argument s of the MOV and COPY commands.		●	●	●	●	●	18 22.4	3	
	61	FUN 122 (s) (INDXC (s))		Index cancel	Cancels the index specification that is set for argument d or s of the MOV and COPY commands.		↕	●	●	●	●	18	3	
	62	FUN 123 (s) (INC (s))		Increment (Word I/O)	Add 1 to the number specified with word I/O.		●	●	●	●	●	29 30.4	3	*9
	63	FUN 124 (s) (INCD (s))		Increment (Double word I/O)	Add 1 to the number specified with double word I/O.		↕	●	●	●	●	42 42.4	3	*9
	64	FUN 125 (s) (DEC (s))		Decrement (Word I/O)	Subtract 1 from the number specified with word I/O.		●	●	●	●	●	30 30.4	3	*9
	65	FUN 126 (s) (DECD (s))		Decrement (Double word I/O)	Subtract 1 from the number specified with double word I/O.		↕	●	●	●	●	42 42.4	3	*9
	65	FUN 127 (s) (BITTOW (s))		Expand bit data to word data	Sets the data equivalent to the number of bits starting from the specified I/O number in the word I/O number.		↕	●	●	●	●	139 133.4	3	
	66	FUN 128 (s) (WTOBIT)		Expand word data to bit data	Sets the data equivalent to the number of words starting from the specified I/O number in the bit I/O number.		↕	●	●	●	●	145 139.4	3	
	66	FUN 162 (s)		Explicit message execution	Explicit message sent from EH-RMD to slave units.		↕	●	●	●	●	127.2 -	3	
	67	FUN 163 (s)		Explicit message initial setting	Initial setting for explicit message sending.		↕	●	●	●	●	37.2 -	3	
	68	FUN 190 (s)		Inverter control command	Hitachi inverter SJ300/L300P series controlled via RS-485.		↕	●	●	●	●			
	69	FUN 200 (s) (XYR/W (s))		X/Y area read/write command	Command for reading and writing data using the X and Y areas.		↕	●	●	●	●	65 57.4	3	*10
	70	FUN 201 (s) (SCR/W (s))		Status control read/write command	Command for reading and writing data in the status control area.		↕	●	●	●	●	66	3	*10
	71	FUN 210 (s) (LOGIT (s))		Data logging initialization setting	Initializes the data logging function.		↕	●	●	●	●	1,728 2906.4	3	*8
	72	FUN 211 (s) (LOGWRT (s))		Log data write	Writes the data to the memory board.		↕	●	●	●	●	645 526.9	3	*8

Classification	Item number	Ladder symbol	Command symbol	Command name	Process descriptions	I/O types used	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C	Process time (μ s) CPU5** CPU4** CPU**A	Step	Remarks
FUN commands	73	FUN 212 (s) (LOGCLR (s))		Log data clear	Clears the log data in the memory board.	s: WR,WL,WM	↕	●	●	●	●	164 / 79.4	3	*8
	74	FUN 213 (s) (LOGRED (s))		Log data read	Reads the log data to the specified I/O area.		↕	●	●	●	●	268 / 210.4	3	*8
	75	FUN 254 (s) (BOXC (s))		BOX comment	Nothing is processed in the CPU.		●	●	●	●	●	0.3 / 14.4	3	
	76	FUN 255 (s) (MEMC (s))		Memo comment	Nothing is processed in the CPU.		●	●	●	●	●	0.3 / 14.4	3	

*1 Value in case of word, it is 5 steps for LD (s1□s2) and AND (s1□s2), and 6 steps for OR (s1□s2).

*2 Value in case of D-word for LD (s1□s2) and AND (s1□s2), it is 5 steps when s1 and s2 is I/O and I/O, 6 steps when s1 and s2 is I/O and const. or const. and I/O, and 7 steps when s1 and s2 is const. and const.. For OR (s1□s2), 1 step is added each.

*3 Processing time when n=1.

*4 Value in case of the size of FIFO is n=1.

*5 When condition not fulfilled.

*6 When condition not fulfilled.

*7 When condition fulfilled.

*8: Supported by the EH-CPU308(A)/316(A)/448(A)/516/548.

*9: Supported by the EH-CPU308/316/***A/448/516/548.

*10: Supported by the EH-CPU308A/316A/448(A)/516/548.

*11: Supported by the EH-CPU***A/448/516/548.

*12: Supported by the EH-CPU516/548.

5.3 Command Specification Details

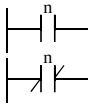
(1) Basic commands	
(2) Arithmetic commands	
(3) Application commands	
(4) Control commands	
(5) High-function module transfer commands	
(6) FUN commands	

The classification of the processing time column in each command table is shown in the table below:

Description in the table	Classification
EH-CPU4**/ EH-CPU5**	EH-CPU448/448A/516/548
EH-CPU***A	EH-CPU104A/208A/308A/316A
EH-CPU3**	EH-CPU308/316 of ROM Ver. 04 or later
Other than left	EH-CPU104/208, and EH-CPU308/316 of ROM Ver. 03 or earlier

Depending on the CPU model, all of the commands may not be supported. For the commands that are supported by each CPU, see the explanation of each command or Appendix 2, “H-series Command Support Comparison Chart”.

Item number	Basic commands-1, 2	Name	Logical operation start (LD, LDI)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
	DER	ERR	SD	V	C	Ave	Max	Ave	Max	Ave	Max	
	●	●	●	●	●	0.1	←	1.0 1.1 1.0 1.5	←	1.3	1.5	Upper case: LD Lower case: LDI

Command format	Number of steps	
	Condition	Steps
LD n		
LDI n	—	1

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
n	I/O number	○	○	○	○									

Function

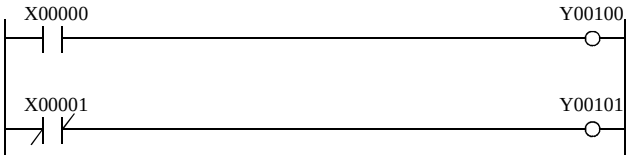
Starts the a-contact logical operation. Enters the continuity state when input is on.

Starts the b-contact logical operation. Enters the continuity state when input is off.

Cautionary notes

- L and WL become the internal output when link modules are not used.

Program example



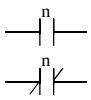

```
LD    X00000
OUT   Y00100

LDI   X00001
OUT   Y00101
```

Program description

- When input X00000 is on, output Y00100 is on; when off, the output is off.
- When input X00001 is off, output Y00101 is on; when on, the output is off.

Item number	Basic commands-3, 4	Name	Contact series connection (AND, ANI)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
	DER	ERR	SD	V	C	Ave	Max	Ave	Max	Ave	Max	
	●	●	●	●	●	0.1	←	1.0	←	1.0	1.2	Upper case: AND Lower case: ANI
Command format	Number of steps							1.1				
AND n	Condition		Steps									
ANI n	—		1					1.0				
								1.3				

Usable I/O	Bit				Word				Double word			Constant	Other
	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
n I/O number	○	○	○	○									

Function

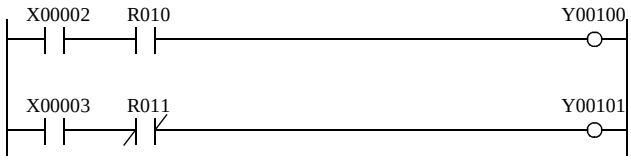
Obtains AND of the previous operation result and the a-contact operation.

Obtains AND of the previous operation result and the b-contact operation.

Cautionary notes

- L and WL become the internal output when link modules are not used.

Program example




```
LD    X00002
AND   R010
OUT   Y00100

LD    X00003
ANI   R011
OUT   Y00101
```

Program description

- When input X00002 and R010 are both on, output Y00100 is on and all others are off.
- When input X00003 is on and R011 is off, output Y00101 is on and all others are off.

Item number	Basic commands-5, 6	Name	Contact parallel connection (OR, ORI)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
	DER	ERR	SD	V	C	Ave	Max	Ave	Max	Ave	Max	
	●	●	●	●	●	7.1	←	2.1	←	1.3	1.5	Upper case: OR Lower case: ORI
Command format	Number of steps							1.2				
OR n	Condition			Steps				2.3				
ORI n	—			2				1.4				

Usable I/O	Bit				Word				Double word			Constant	Other
	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
n I/O number	○	○	○	○									

Function

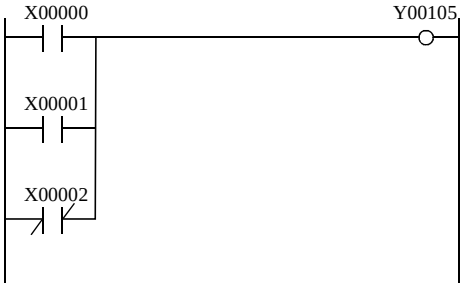
Obtains OR of the previous operation result and the a-contact operation.

Obtains OR of the previous operation result and the b-contact operation.

Cautionary notes

- L and WL become the internal output when link modules are not used.

Program example




```
LD    X00000
OR    X00001
ORI   X00002
OUT   Y00105
```

Program description

- When X00000 is on, X00001 is on, or X00002 is off, the operation is 1 and Y00105 turns on.

Item number	Basic commands-7	Name	Negation (NOT)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
	DER	ERR	SD	V	C	Ave	Max	Ave	Max	Ave	Max	
	●	●	●	●	●	0.2	←	2.1	←	1.7	←	

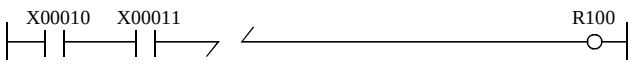
Command format	Number of steps	
	Condition	Steps
NOT	—	2

Usable I/O	Bit				Word				Double word			Constant	Other
	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		

Function

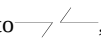
- Reverses the operation result obtained up to that point.

Program example

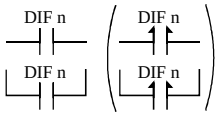



```
LD      X00010
AND     X00011
NOT
OUT     R100
```

Program description

- When input X00010 and input X00011 are both on, the operation is 1, but due to  , the calculation turns into “0” and R100 turns off.
- In all other cases, R100 turns on.

Item number	Basic commands-8	Name	Rising edge detection (AND DIF, OR DIF)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
	DER	ERR	SD	V	C	Ave	Max	Ave	Max	Ave	Max	
	●	●	●	●	●	0.3	←	3.1	←	2.7	2.9	Upper case: AND DIF Lower case: OR DIF

Command format	Number of steps	
	Condition	Steps
AND DIF n	AND DIF n	3
OR DIF n	OR DIF n	4

	Bit				Word				Double word				
Usable I/O	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM	Constant	Other
n Number												○	0 to 511 (Decimal)

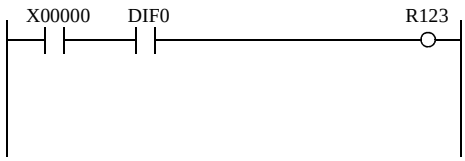
Function

- Detects the rise of an input signal and retains the operation result for one scan.
  ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

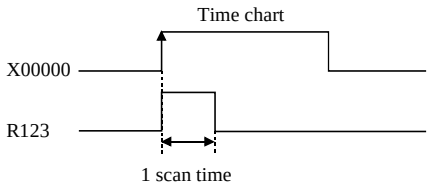
- DIF number may not be overlapped. (However, no error is generated even if overlapped numbers are used.)
- DIF cannot use the b contact.

Program example



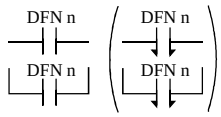

```
LD    X00000
AND   DIF0
OUT   R123
```

Program description



- Upon rising of X00000 on, R123 turns on for one scan.
- If b-contact is used for X00000, operation will be the same as the a-contact DFN operation.

Item number	Basic commands-9	Name	Falling edge detection (AND DFN, OR DFN)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		Upper case: AND DFN Lower case: OR DFN
	DER	ERR	SD	V	C	Ave	Max	Ave	Max	Ave	Max	
	●	●	●	●	●	0.3	←	3.1	←	2.7	2.9	
Command format	Number of steps							3.1				
AND DFN n	Condition			Steps				4.1				
OR DFN n	AND DFN n			3				3.1				
	OR DFN n			4								

Usable I/O	Bit				Word				Double word			Constant	Other
	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
n Number												○	0 to 511 (Decimal)

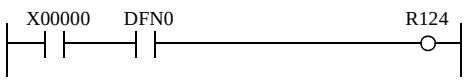
Function

- Detects the fall of an input signal and retains the operation result for one scan.
  ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

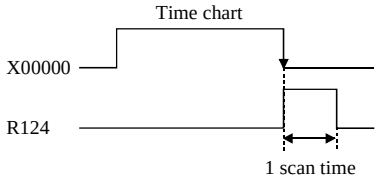
- DFN number may not be overlapped. (However, no error is generated even if overlapped numbers are used.)
- DFN cannot use the b contact.

Program example



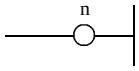

```
LD    X00000
AND   DFN0
OUT   R124
```

Program description



- Upon a fall of X00000, R124 turns on for one scan.
- If b-contact is used for X00000, operation will be the same as the a-contact DIF operation.

Item number	Basic commands-10	Name	Coil output (OUT)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
	DER	ERR	SD	V	C	Ave	Max	Ave	Max	Ave	Max	
	●	●	●	●	●	0.1	←	1.0	←	1.0	1.7	

Command format	Number of steps	
	Condition	Steps
OUT n	—	1

Usable I/O	Bit				Word				Double word			Constant	Other
	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
n I/O number		○	○	○									

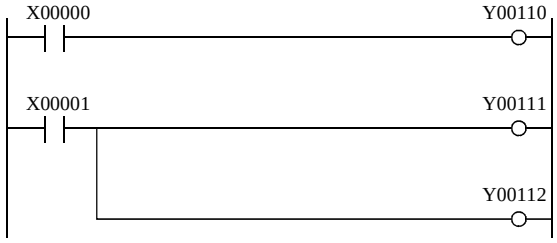
Function

- Switches on the coil when the operation result obtained up to that point is “1.”
- Switches off the coil when the operation result obtained up to that point is “0.”

Cautionary notes

- L and WL become the internal output when link modules are not used.

Program example




```
LD    X00000
OUT   Y00110

LD    X00001
OUT   Y00111
OUT   Y00112
```

Program description

- When input X00000 is on, the operation is “1” and Y00110 turns on.
- When input X00001 is on, the operation is “1,” and Y00111 and Y00112 turn on.

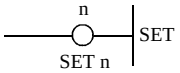
Item number	Basic commands-11, 12	Name	Set/reset coil output (SET, RES)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
	DER	ERR	SD	V	C	Ave	Max	Ave	Max	Ave	Max	
	●	●	●	●	●	0.1	←	2.1	←	0.9	1.7	Upper case: SET Lower case: RES

Command format	Number of steps										
SET n RES n	Condition	Steps	0.1	←	2.1	←	0.9	1.7			
	—	1			2.1						

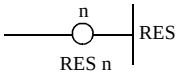
Usable I/O	Bit				Word				Double word			Constant	Other
	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
n I/O number		○	○										

Function



Switches on the device when the operation result obtained up to that point is "1."

The device that is switched on will not be switched off even if the operation result is "0."



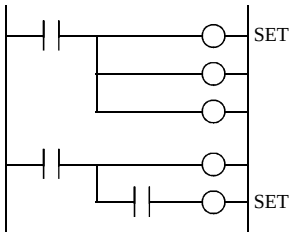
Switches off the device when the operation result obtained up to that point is "1."

( ) indicates the display when the LADDER EDITOR is used.

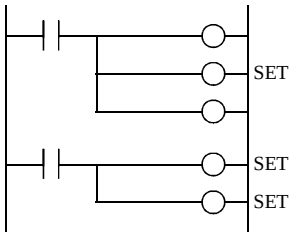
Cautionary notes

- When a set/reset coil is used on a multi layer coil, it must be set to the highest level or an arbitrary coil must be entered immediately before the set/reset coil.

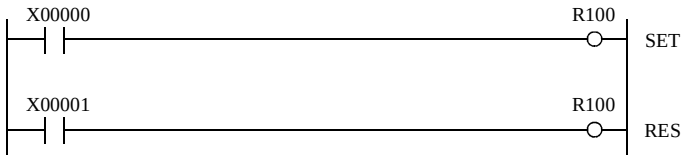
Example of OK



Example of NG



Program example



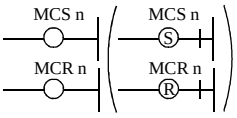

```
LD    X00000
SET   R100
LD    X00001
RES   R100
```

Program description

- When input X00000 turns on, output R100 turns on. Even if X00000 turns off, R100 remains on.
- When input X00001 turns on, output R100 turns off.

When input X00000 and X00001 both turn on, the one executed later than the other in the program takes priority.

Item number	Basic commands-13, 14	Name	Set (start)/reset (cancel) master control (MCS, MCR)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		Upper case: MCS Lower case: MCR
	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps										
MCS n MCR n	Condition	Steps	0.3	←	3.1 3.1	←	2.3	←			
	MCS n	3	0.2	←	2.2 2.2	←	1.6	←			
	MCR n	2									

	Bit				Word				Double word				
Usable I/O	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM	Constant	Other
n Number												○	0 to 49 (Decimal)

Function

- Controls the input to the ladder sandwiched by the master control set (MCS n) and reset (MCR n). (An AND operation is performed with respect to each input and MCS.)

The master control can be used up to eight layers.

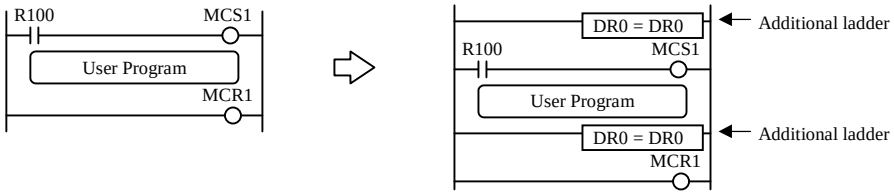
( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

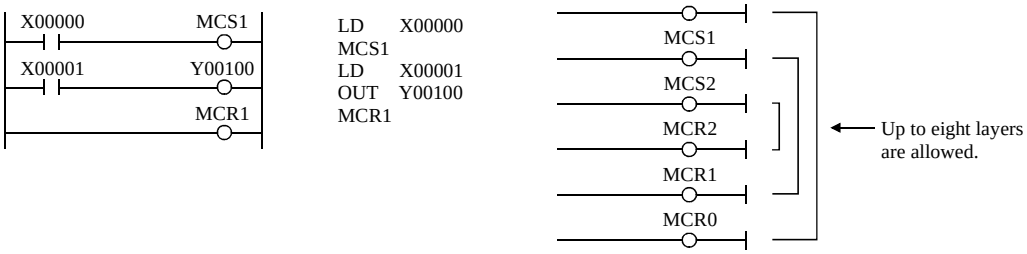
- Always use the master control MCS and MCR in pairs.
- When a EH-CPU448* (ROM VER. C413 / C331 or older) is used. Insert a double word operation just before a master control instruction.

* As for the EH-CPU448 (ROM VER. C414 / C332 or later) , dummy operation doesn't need to add to user program.

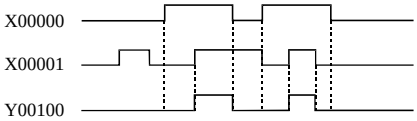
EX.)



Program example



Program description



- When input X00000 is on, the ladders surrounded by MCS and MCR obeys input X00001, and output Y00100 turns on/off.
- When input X00000 is off, the ladders surrounded by MCS and MCR are independent of input X00001, and output Y00100 turns off.

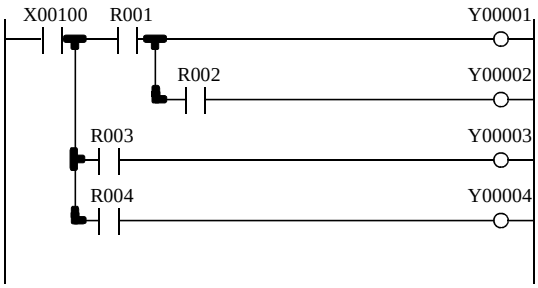
Item number	Basic commands-15, 16, 17	Name	Save/read/clear operation result (Branching of ladder)

Ladder format	Condition code					Processing time (μs)						Remark
Save Read Clear	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps											
MPS Save MRD Read MPP Clear	Condition	Steps	—	—	—	—	—	—				
	—	0										

Usable I/O	Bit				Word				Double word			Constant	Other
	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		

Function




```
LD    X00100
MPS
AND   R001
MPS
OUT   Y00001
MPP
AND   R002
OUT   Y00002
MRD
AND   R003
OUT   Y00003
MPP
AND   R004
OUT   Y00004
```

- MPS stores the immediately prior operation results. (Push)
- MRD reads the results stored by the MPS and continues operation.
- MPP reads the results stored immediately prior by the MPS and continues operation, then clears the results after operation. (Pull)

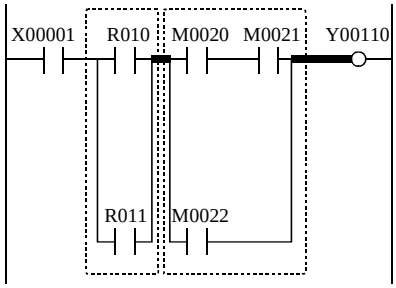
Item number	Basic commands-18	Name	Logical block series connection (ANB)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
(See Function column)	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps	
	Condition	Steps
ANB	—	0

Usable I/O	Bit				Word				Double word			Constant	Other
	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		

Function




```
LD    X00001
LD    R010
OR    R011
ANB
LD    M0020
AND   M0021
OR    M0022
ANB
OUT   Y00110
```

This command is used to perform AND operation with respect to the logical operation blocks (dotted line areas).

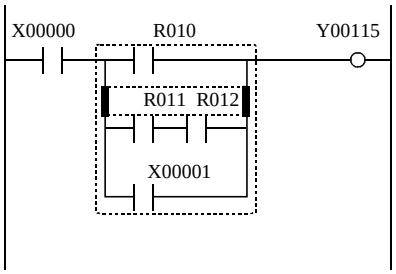
Item number	Basic commands-19	Name	Logical block parallel connection (ORB)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
(See Function column)	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps										
	Condition	Steps									
ORB	—	1	7.0	←	1.0 ..... 1.0	←	0.7	—			

Usable I/O	Bit				Word				Double word			Constant	Other
	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		

Function




```
LD    X00000
LD    R010
LD    R011
AND   R012
ORB
OR    X00001
ANB
OUT   Y00115
```

This command is used to perform OR operation with respect to the logical operation blocks (dotted line areas).

Item number	Basic commands-20	Name	Processing box start and end (PROCESSING BOX)

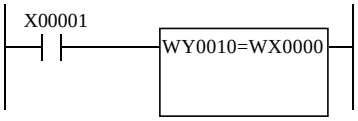
Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps										
	Condition	Steps	0.3	←	2.2 ..... 2.2	←	1.8	←			
[ ]	—	3									

Usable I/O	Bit				Word				Double word			Constant	Other
	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		

Function

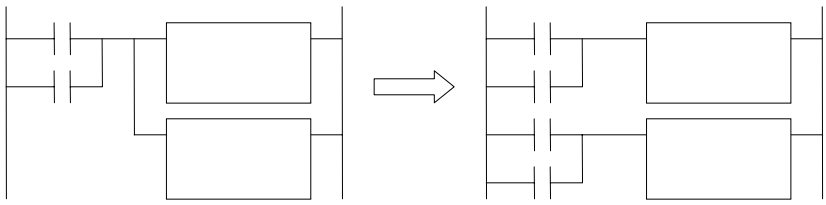
- Indicates the start and end of the processing box.



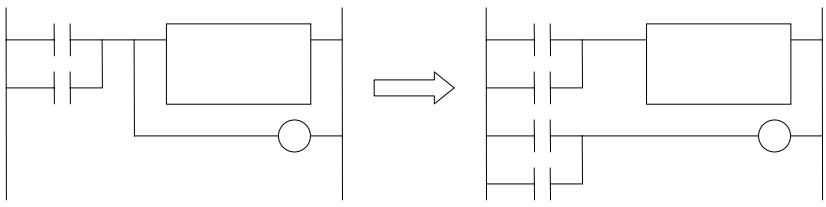

```
LD      X00001
[
WY0010=WX0000
]
```

In the above example, the operation of inside the processing box will be executed when input X00001 is on.

Parallel connection of processing box or coil is not allowed.



Not allowed Allowed



Not allowed Allowed

Item number	Basic commands-21	Name	Relational box start and end (RELATIONAL BOX)

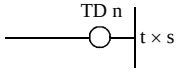
Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					—	—	—	—	—	—	
	Condition			Steps								
( )	—			0								

Usable I/O	Bit				Word				Double word			Constant	Other
	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		

Function

- Indicates the start and end of the relational box.

Item number	Basic commands-22	Name	On delay timer

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					26.4	←	28.1	←	11.5	12.0	
	Condition		Steps					9.0				
OUT TD n t s	—		5									

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
n	Timer number												○	0 to 255 (Decimal)
t	Time base													.01s, .1s, 1s
s	Set value					○	○	○					○	1 to 65,535 (Decimal)

Function

- The progress value is updated when the startup condition is on, and the coil switches on when the progress value is greater than or equal to the set value.
- If the startup condition switches off, the progress value is cleared and the coil switches off.
- The progress value is set in TC n and does not exceed 65,535 (decimal).
- If the progress value is updated while the system is running, the operation will be performed using the new progress value at that point.
- If an I/O is set for the set value, the set value can be changed during operation by changing the I/O value, since the set values are updated during each scan.

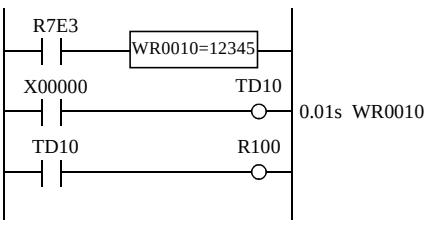
Cautionary notes

- The .01s time base can only be used for timer numbers 0 to 63 (64 points).
- The .1s and 1s time bases can be used for all timer numbers (0 to 255).
- A maximum of 256 points can be used for the timers TD, SS, MS, TMR and WDT in total.
  However, the same area as the counter is used. Timer number and counter number may not be overlapped.
- The progress value is updated when the timer command is expected in the EH-CPU***A/448/516/548. Therefore, if a program that does not scan the timer command execution section after the timer is activated is created using the JMP command or master control (MCS), the timer may not turn on correctly. (The timer does not turn on correctly when the time that does not scan the timer command execution section exceeds the time calculated by multiplying the time base by 65,535.) Note that the previous progress value is also retained until the timer command is executed.

Program example


```
LD    X00000
OUT   TD10  0.01s  12345
LD    TD10
OUT   R100
```

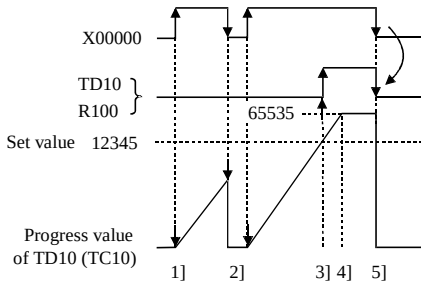
- An example of a word I/O being used as the set value for the ladder shown above.




```
LD    R7E3
[
WR0010=12345
]
LD    X00000
OUT   TD10  0.01s  WR0010
LD    TD10
OUT   R100
```

Program description

[Time chart]



1] When input X00000 turns on, TD progress value is updated.

2] When input X00000 turns off, the TD progress value is cleared.

3] TD10 turns on when progress value ≧ set value.

4] While X00000 is on, the progress value increases, but will not increase exceeding 65,535.

5] When X00000 turns off, TD10 also turns off and the progress value is cleared.

- Example using word I/O as the set value
  When RUN is commenced, the set value is set to word I/O.
  Or, designate the word I/O for the set value to store in the power failure memory beforehand.

Item number	Basic commands-22	Name	On delay timer (ON DELAY TIMER)

Ladder format	Condition code					Processing time (μs)						Remark
TM n — t × s	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps									
	Condition	Steps	26.4	←	28.1	←	11.5	12.0		
OUT TD n t s	—	5			9.0					

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
n	Timer number												○	0 to 255 (Decimal)
t	Time base													.01s, .1s, 1s
s	Set value					○	○	○					○	1 to 65,535 (Decimal)

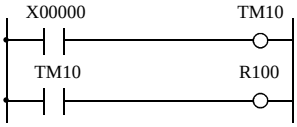
Function

- The progress value is updated when the startup condition is on, and the coil switches on when the progress value is greater than or equal to the set value.
- If the startup condition switches off, the progress value is cleared and the coil switches off.
- The progress value is set in TV n and does not exceed 65,535 (decimal).
- If the progress value is updated while the system is running, the operation will be performed using the new progress value at that point.
- If an I/O is set for the set value, the set value can be changed during operation by changing the I/O value, since the set values are updated during each scan.

Cautionary notes

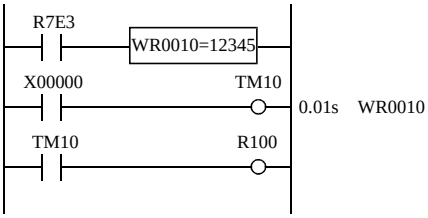
- Time base is fixed as 0.01s.
- Timer TM is supported by LADDER EDITOR for Windows Ver.3.00 or newer. User program including TM does not work with CPU104(A)/208(A)/308(A)/316(A)/448(A) and LADDER EDITOR for Windows Ver.2.** or older.
- The progress value is updated when the timer command is expected. Therefore, if a program that does not scan the timer command execution section after the timer is activated is created using the JMP command or master control (MCS), the timer may not turn on correctly. (The timer does not turn on correctly when the time that does not scan the timer command execution section exceeds the time calculated by multiplying the time base by 65,535.) Note that the previous progress value is also retained until the timer command is executed.

Program example




```
LD      X00000
OUT     TM10   0.01s   12345
LD      TM10
OUT     R100
```

- An example of a word I/O being used as the set value for the ladder shown above.




```
LD      R7E3
[
WR0010=12345
]
LD      X00000
OUT     TM10   0.01s   WR0010
LD      TM10
OUT     R100
```

Please refer to the “TD” for the timing chart.

Item number	Basic commands-23	Name	Single shot timer

Ladder format	Condition code					Processing time (μs)						Remark
SS n t × s	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					21.4	←	23.1 9.9	←	12.2	←	
OUT SS n t s	Condition		Steps									
	—		5									

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
n	Timer number												○	0 to 255 (Decimal)
t	Time base													.01s, .1s, 1s
s	Set value					○	○	○					○	1 to 65,535 (Decimal)

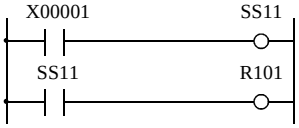
Function

- Detects the rising edge of the startup condition, starts updating progress values, and switches on the coil.
- The coils switches off when the progress value is greater than or equal to the set. If a rising edge is detected while the progress value is less than the set value, the progress value is set to 0 and the counter is reset.
- The progress value is set in TC n and does not exceed 65535 (decimal).
- If the progress value is updated while the system is running, the operation will be performed using the new progress value at that point.
- If an I/O is set for the set value, the set value can be changed during operation by changing the I/O value, since the set values are updated during each scan.

Cautionary notes

- The .01s time base can only be used for timer numbers 0 to 63 (64 points).
- The .1s and 1s time bases can be used for all timer numbers (0 to 255).
- A maximum of 256 points can be used for the timers TD, SS, MS, TMR and WDT in total.
  However, the same area as the counter is used. Timer number and counter number may not be overlapped.
- Since the startup condition of a single shot is edge detection, the condition cannot be detected during the first scan after the running is started.
- The progress value is updated when the timer command is expected in the EH-CPU***A/448/516/548. Therefore, if a program that does not scan the timer command execution section after the timer is activated is created using the JMP command or master control (MCS), the timer may not turn on correctly. (The timer does not turn on correctly when the time that does not scan the timer command execution section exceeds the time calculated by multiplying the time base by 65,535.) Note that the previous progress value is also retained until the timer command is executed.

Program example



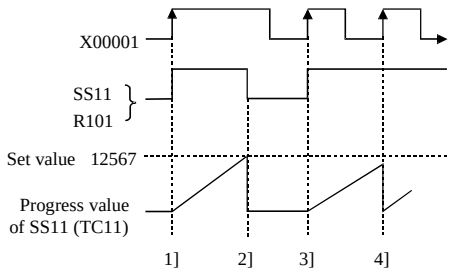

```
LD    X00001
OUT   SS11  0.01s  12567
LD    SS11
OUT   R101
```

- An example of a word I/O being used as the set value for the ladder shown above.


```
LD    R7E3
[
WR0011=12567
]
LD    X00001
OUT   SS11  0.01s  WR0011
LD    SS11
OUT   R101
```

Program description

[Time chart]



1] The progress value is updated and SS11 turns on at the rising edge of X00001.

2] SS11 turns off when set value ≧ progress value.
X00001 is turned on at this time, but the single shot startup conditions are ignored because it uses edge trigger.

3] SS11 is turned on at the rising edge of X00001 again, and the progress value is updated.

4] When the rising edge of X00001 is detected while the progress value does not reach the set value, the single shot timer is triggered again and the progress value returns to 0, then starts increasing. The SS11 remains on.

- Example using word I/O as the set value
  When RUN is commenced, the set value is set to word I/O.
  Or, designate the word I/O for the set value to store in the power failure memory beforehand.

Item number	Basic commands-24	Name	Mono stable timer

Ladder format	Condition code					Processing time (μs)						Remark
MS n t × s	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					21.4	←	23.1 9.8	←	12.2	←	
OUT MS n t s	Condition: —		Steps: 5									

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
n	Timer number												○	0 to 255 (Decimal)
t	Time base													.01s, .1s, 1s
s	Set value					○	○	○					○	1 to 65,535 (Decimal)

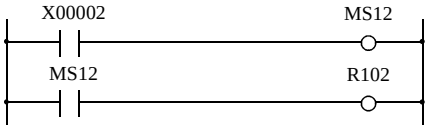
Function

- Detects the rising edge of the startup condition, starts updating progress values, and switches on the coil.
- The coil switches off when the progress value is greater than or equal to the set value. The rise of the startup condition is ignored while the MS is on.
- The progress value is set in TC n and does not exceed 65,535 (decimal).
- If the progress value is updated while the system is running, the operation will be performed using the new progress value at that point.
- If an I/O is set for the set value, the set value can be changed during operation by changing the I/O value, since the set values are updated during each scan.

Cautionary notes

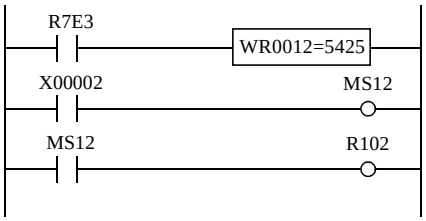
- The .01s time base can only be used for timer numbers 0 to 63 (64 points).
- The .1s and 1s time bases can be used for all timer numbers (0 to 255).
- A maximum of 256 points can be used for the timers TD, SS, MS, TMR and WDT in total.
  However, the same area as the counter is used. Timer number and counter number may not be overlapped.
- Since the startup condition of the mono stable timer is edge detection, the condition may not be detected during the first scan after the running is started.
- The progress value is updated when the timer command is expected in the EH-CPU***A/448/516/548. Therefore, if a program that does not scan the timer command execution section after the timer is activated is created using the JMP command or master control (MCS), the timer may not turn on correctly. (The timer does not turn on correctly when the time that does not scan the timer command execution section exceeds the time calculated by multiplying the time base by 65,535.) Note that the previous progress value is also retained until the timer command is executed.

Program example




```
LD    X00002
OUT   MS12  0.1s  5425
LD    MS12
OUT   R102
```

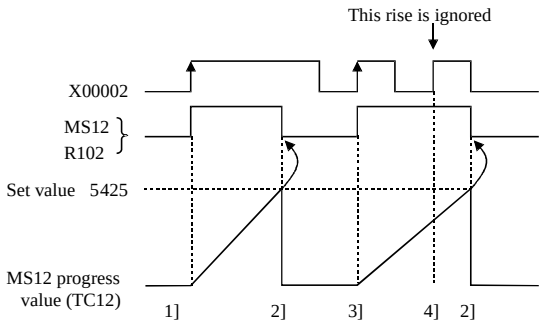
- An example of a word I/O being used as the set value for the ladder shown above.




```
LD    R7E3
[
WR0012=5425
]
LD    X00002
OUT   MS12  0.1s  WR0012
LD    MS12
OUT   R102
```

Program description

[Time chart]



1] The progress value is updated and MS12 turns on at the rising edge of X00002.

2] MS12 turns off when set value ≧ progress value. X00002 is on at this time, but it is ignored since the startup condition for a mono-stable timer is edge trigger.

3] MS12 is turned on at the rising edge of X00002 again, and progress value is updated.

4] Even if the rising edge of X00002 is detected while the progress value does not reach the set value, the mono-stable timer ignores the rise.

- Example using word I/O as the set value
  When RUN is commenced, the set value is set to word I/O.
  Or, designate the word I/O for the set value to store in the power failure memory beforehand.

Item number	Basic commands-25	Name	Integral timer

Ladder format	Condition code					Processing time (μs)						Remark
TMR n t × s	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					21.4	←	23.1 9.4	←	12.2	←	
OUT TMR n t s	Condition			Steps								
	—			5								

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
n	Timer number												○	0 to 255 (Decimal)
t	Time base													.01s, .1s, 1s
s	Set value					○	○	○					○	1 to 65,535 (Decimal)

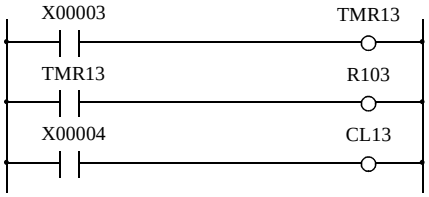
Function

- Updates the progress value while the startup condition is on. The progress value is not cleared even if the startup condition switches off, and the update resumes when the condition switches on again.
- The coil switches on when the progress value is greater than or equal to the set value, and will not turn off until the clear input CL n switches on.
- The progress value is set in TC n and does not exceed 65,535 (decimal).
- If the progress value is updated while the system is running, the operation will be performed using the new progress value at that point.
- If an I/O is set for the set value, the set value can be changed during operation by changing the I/O value, since the set values are updated during each scan.

Cautionary notes

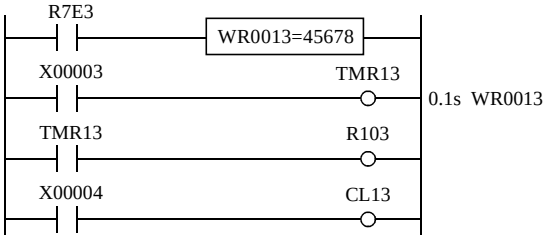
- The .01s time base can only be used for timer numbers 0 to 63 (64 points).
- The .1s and 1s time bases can be used for all timer numbers (0 to 255).
- A maximum of 256 points can be used for the timers TD, SS, MS, TMR and WDT in total.
  However, the same area as the counter is used. Timer number and counter number may not be overlapped.
- While the clear input CL n is on, the turning on of the startup condition is ignored.
- The progress value is updated when the timer command is expected in the EH-CPU***A/448/516/548. Therefore, if a program that does not scan the timer command execution section after the timer is activated is created using the JMP command or master control (MCS), the timer may not turn on correctly. (The timer does not turn on correctly when the time that does not scan the timer command execution section exceeds the time calculated by multiplying the time base by 65,535.) Note that the previous progress value is also retained until the timer command is executed.

Program example




```
LD    X00003
OUT   TMR13  0.1s  45678
LD    TMR13
OUT   R103
LD    X00004
OUT   CL13
```

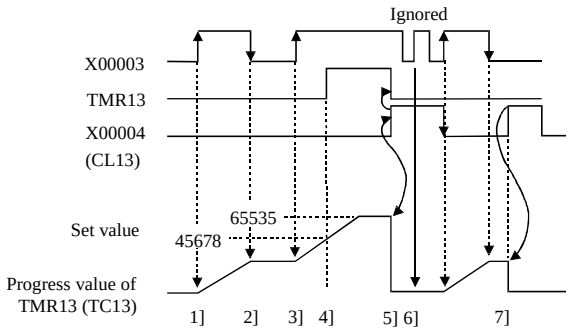
- An example of a word I/O being used as the set value for the ladder shown above.




```
LD    R7E3
[
WR0013=45678
]
LD    X00003
OUT   TMR13  0.1s  WR0013
LD    TMR13
OUT   R103
LD    X00004
OUT   CL13
```

Program description

[Time chart]



1] The progress value is updated while X00003 is on.

2] When X00003 turns off, the update of the progress value is stopped, but the progress value is maintained.

3] When X00003 is turned on again, the progress value is re-updated.

4] Timer coil TMR13 is turned on while the progress value ≥ set value. This state is maintained until the timer clear is turned on.

5] When timer clear (CL13) is turned on, the timer coil and progress value are both cleared.

6] While timer clear (CL13) is on, the startup condition is ignored.

7] The progress value is cleared to 0 when the timer is cleared.

- Example using word I/O as the set value
  When RUN is commenced, the set value is set to word I/O.
  Or, designate the word I/O for the set value to store in the power failure memory beforehand.
- Timer is cleared by the conditions immediately prior to the execution of the timer coil command.

Item number	Basic commands-26	Name	Watchdog timer

Ladder format	Condition code					Processing time (μs)						Remark
WDT n t × s1 s2	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					27.6	←	32.3 13.6	←	14.4	15.1	
	Condition			Steps								
OUT WDT n t s1 s2	—			7								

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
n	Timer number												○	0 to 255 (Decimal)
t	Time base													.01s, .1s, 1s
s1	Set value 1					○	○	○					○	1 to 65,535 (Decimal)
s2	Set value 2					○	○	○					○	1 to 65,535 (Decimal)

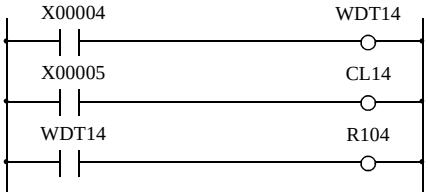
Function

- Updates the progress value while the startup condition is switched on. The coil will not turn on if the clear input CL n is accessed while set value 1 ≤ progress value < set value 2. The coil switches on if the clear input CL n is accessed while the progress value is less than set value 1 or if set value 2 is less than or equal to the progress value. If the startup condition switches off, everything is cleared.
- The progress value is set in TC n and does not exceed 65,535 (decimal).
- If the progress value is updated while the system is running, the operation will be performed using the new progress value at that point.
- If an I/O is set for the set value, the set value can be changed during operation by changing the I/O value, since the set values are updated during each scan.

Cautionary notes

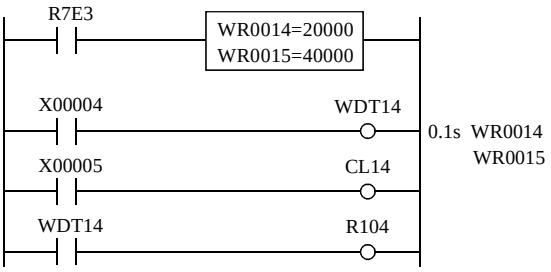
- The .01s time base can only be used for timer numbers 0 to 63 (64 points).
- The .1s and 1s time bases can be used for all timer numbers (0 to 255).
- A maximum of 256 points can be used for the timers TD, SS, MS, TMR and WDT in total.
  However, the same area as the counter is used. Timer number and counter number may not be overlapped.
- The s1 of the set value must always be less than s2. If s1 is greater than or equal to s2, the coil turns on when the progress value reaches s2.
- The progress value is updated when the timer command is expected in the EH-CPU***A/448/516/548. Therefore, if a program that does not scan the timer command execution section after the timer is activated is created using the JMP command or master control (MCS), the timer may not turn on correctly. (The timer does not turn on correctly when the time that does not scan the timer command execution section exceeds the time calculated by multiplying the time base by 65,535.) Note that the previous progress value is also retained until the timer command is executed.

Program example




```
LD    X00004
OUT   WDT14  0.1s  20000  40000
LD    X00005
OUT   CL14
LD    WDT14
OUT   R104
```

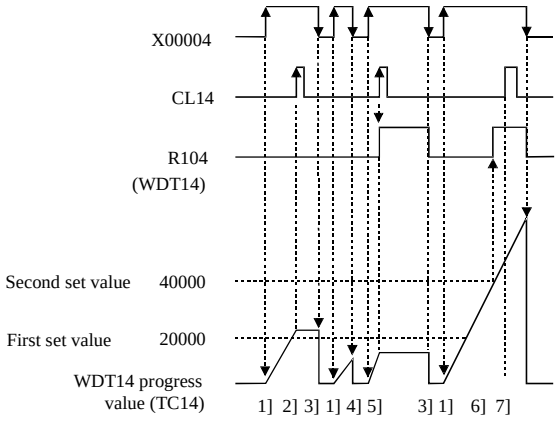
- An example of a word I/O being used as the set value for the ladder shown above.




```
LD     R7E3
[
WR0014=20000
WR0015=40000
]
LD     X00004
OUT    WDT14  0.1s  WR0014  WR0015
LD     X00005
OUT    CL14
LD     WDT14
OUT    R104
```

Program description

[Time chart]



1] The progress value is updated while X00004 is on.

2] Since watchdog clear (CL14) turned on after the first set value is exceeded and before the second set value is exceeded, it is considered a normal operation, so R104 (WDT14) does not turn on.

3] When X00004 turns off, the progress value and WDT coil output are cleared.

4] Since the startup condition turned off before the progress value exceeded the first set value, the WDT coil does not turn on, and the progress value is cleared to 0.

5] Since the watchdog clear (CL14) turned on before the progress value exceeded the first set value, it is considered an abnormal operation and R104 (WDT14) turns on. The progress value is retained as is.

6] Even if the progress value exceeded the second set value, since the watchdog clear (CL14) did not turn on, it is considered an abnormal operation and R104 (SDT14) turns on. The progress value keeps being updated.

7] Even if the watchdog clear (CL14) turns on after the WDT coil is turned on after the progress value exceeds the second set value, it is ignored.

- The clear is performed under the conditions set immediately prior to the execution of the WDT coil command.
- Example using word I/O as the set value

  When RUN is commenced, the set value is set to word I/O.

  Or, designate the word I/O for the set value to store in the power failure memory beforehand.

Item number	Basic commands-27	Name	Counter

Ladder format	Condition code					Processing time (μs)						Remark
CU n s	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps		Processing time (μs)					
	Condition	Steps	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left	
			Ave	Max	Ave	Max	Ave	Max
OUT CU n s	—	5	13.4	←	15.1 ..... 9.5	←	15.2	18.4

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
n	Counter number												○	0 to 511 (Decimal)
s	Set value					○	○	○					○	1 to 65,535 (Decimal)

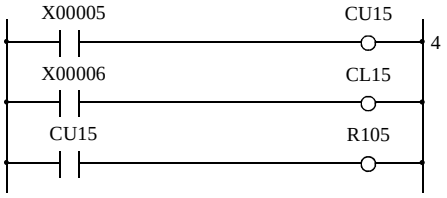
Function

- Increments the progress value by 1 each time the rising edge of the startup condition is detected, and switches on the coil when the progress value is greater than or equal to the set value. The coil that is switched on turns off when the counter clear CL n is switched on, and the progress value is cleared to 0.
- The progress value is set in TC n and does not exceed 65,535 (decimal).
- If the progress value is updated while the system is running, the operation will be performed using the new progress value at that point.
- If an I/O is set for the set value, the set value can be changed during operation by changing the I/O value, since the set values are updated during each scan.

Cautionary notes

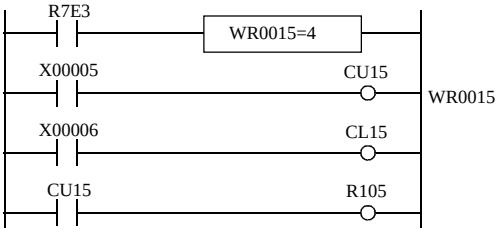
- The counter can be used up to 512 points (No. 0 to 511). However, the first 256 points (No. 0 to 255) will use the same area as the timer.
- The timer numbers and counter numbers can not overlap.
- While the counter clear CL n is on, the rise of startup condition is ignored.
- Since the startup condition of the counter is edge detection, the condition can not be detected during the first scan after the operation is started.
- If the set value is set to 0, it is regarded as a coil that is always on and controlled by the CL n.

Program example




```
LD    X00005
OUT   CU15 4
LD    X00006
OUT   CL15
LD    CU15
OUT   R105
```

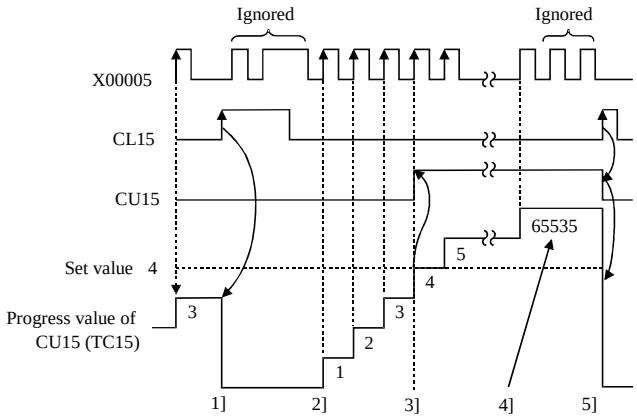
- An example of a word I/O being used as the set value for the ladder shown above.




```
LD    R7E3
[
WR0015=4
]
LD    X00005
OUT   CU15  WR0015
LD    X00006
OUT   CL15
LD    CU15
OUT   R105
```

Program description

[Time chart]



1] The progress value (count) is cleared to 0 by the counter clear (CL15). While the counter clear is on, the progress value will not be updated.

2] The progress value is updated at the rising edge of X00005.

3] Counter coil (CU15) is turned on since set value ≧ progress value.

4] The count value will not exceed 65,535 (decimal).

5] The progress value and counter coil are cleared by counter clear (CL15).

- The clear is performed under the conditions set immediately prior to the execution of the counter coil command.

- Example using word I/O as the set value
  When RUN is commenced, the set value is set to word I/O.
  Or, designate the word I/O for the set value to store in the power failure memory beforehand.

Item number	Basic commands-28	Name	Ring counter

Ladder format	Condition code					Processing time (μs)						Remark
RCU n s	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					13.4	←	15.1 / 9.9	←	14.3	16.9	
	Condition		Steps									
OUT RCU n s	—		5									

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
n	Counter number												○	0 to 511 (Decimal)
s	Set value					○	○	○					○	1 to 65,535 (Decimal)

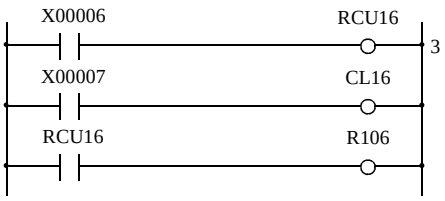
Function

- Increments the progress value by 1 each time the rising edge of the startup condition is detected, and clears the progress value to 0 when the progress value is greater than or equal to the set value. The progress value becomes 0 when the counter clear CL n is switched on, and the coil switches off.
- The progress value is set in TC n and does not exceed the set value.
- If the progress value is updated while the system is running, the operation will be performed using the new progress value at that point.
- If an I/O is set for the set value, the set value can be changed during operation by changing the I/O value, since the set values are updated during each scan.

Cautionary notes

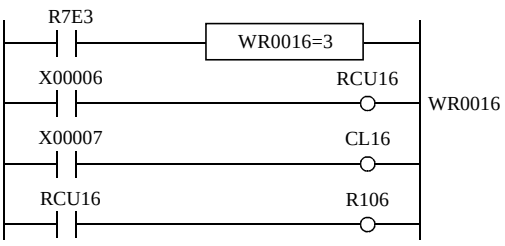
- The counter can be used up to 512 points (No. 0 to 511). However, the first 256 points (No. 0 to 255) will use the same area as the timer.
- The timer numbers and counter numbers can not overlap.
- While the counter clear CL n is on, the rise of startup condition is ignored.
- Since the startup condition of the counter is edge detection, the condition can not be detected during the first scan after the operation is started.
- If the set value is set to 0, it is regarded as a coil that is always on and controlled by the CL n.

Program example




```
LD    X00006
OUT   RCU16 3
LD    X00007
OUT   CL16
LD    RCU16
OUT   R106
```

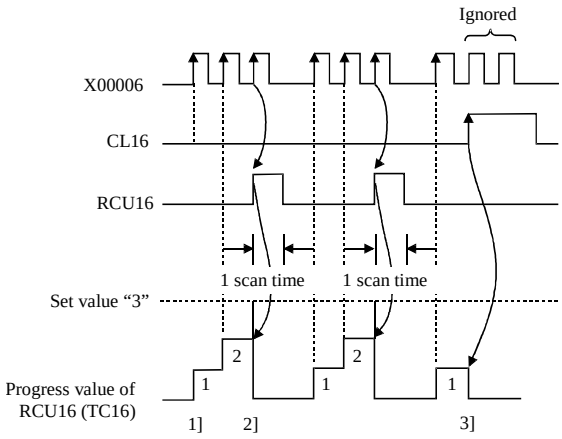
- An example of a word I/O being used as the set value for the ladder shown above.




```
LD    R7E3
[
WR0016=3
]
LD    X00006
OUT   RCU16  WR0016
LD    X00007
OUT   CL16
LD    RCU16
OUT   R106
```

Program description

[Time chart]



1] The progress value (count) is updated at the X00006 rising edge.

2] When set value = progress value, the counter coil (RCU16) turns on for one scan and the progress value is cleared.

3] When counter clear (CL16) is turned on, the progress value is cleared. The progress value is not updated while the counter clear is on.

- The clear is performed under the conditions set immediately prior to the execution of the counter coil command.

- Example using word I/O as the set value
  When RUN is commenced, the set value is set to word I/O.
  Or, designate the word I/O for the set value to store in the power failure memory beforehand.

Item number	Basic commands-29, 30	Name	Up (CTU n) and down (CTD n) of up/down counter

Ladder format	Condition code					Processing time (μs)						Remark
CTU n s CTD n	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		Upper case: CTU Lower case: CTD
	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					13.4	←	15.1	←	12.5	13.1	
OUT CTU n s	Condition			Steps				9.4				
OUT CTD n	CTU			5		12.2	←	14.1	←	11.6	11.7	
	CTD			3				8.9				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
n	Counter number												○	0 to 511 (Decimal)
s	Set value					○	○	○					○	1 to 65,535 (Decimal)

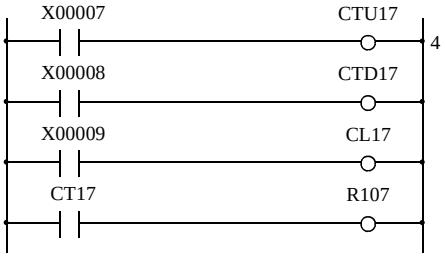
Function

- For the UP counter, increments the progress value by 1 each time the rising edge of the startup condition is detected, while it decrements the progress value by 1 for the DOWN counter. The coil switches on when the progress value is greater than or equal to the set value and switches off when the progress value is less than the set value. When the counter clear CL n switches on, the progress value is cleared to 0 and the coil switches off.
- The progress value is set in TC n, and the value will be in the range of 0 to 65,535 (decimal).
- If the progress value is updated while the system is running, the operation will be performed using the new progress value at that point.
- If an I/O is set for the set value, the set value can be changed during operation by changing the I/O value, since the set values are updated during each scan.

Cautionary notes

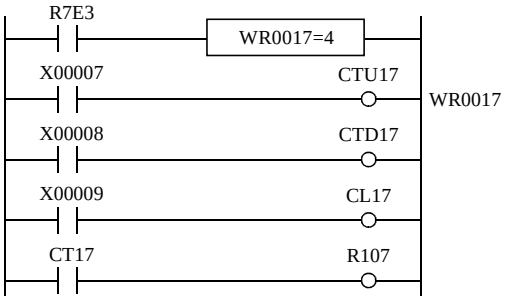
- The counter can be used up to 512 points (No. 0 to 511). However, the first 256 points (No. 0 to 255) will use the same area as the timer.
- The timer numbers and counter numbers cannot overlap.
- The numbers for the UP coil and DOWN coil must be the same.
- While the counter clear CL n is on, the rise of startup condition is ignored.
- Since the startup condition of the counter is edge detection, the condition may not be detected during the first scan after the operation is started.
- If the set value is set to 0, it is regarded as a coil that is always on and controlled by the CL n.

Program example




```
LD     X00007
OUT    CTU17  4
LD     X00008
OUT    CTD17
LD     X00009
OUT    CL17
LD     CT17
OUT    R107
```

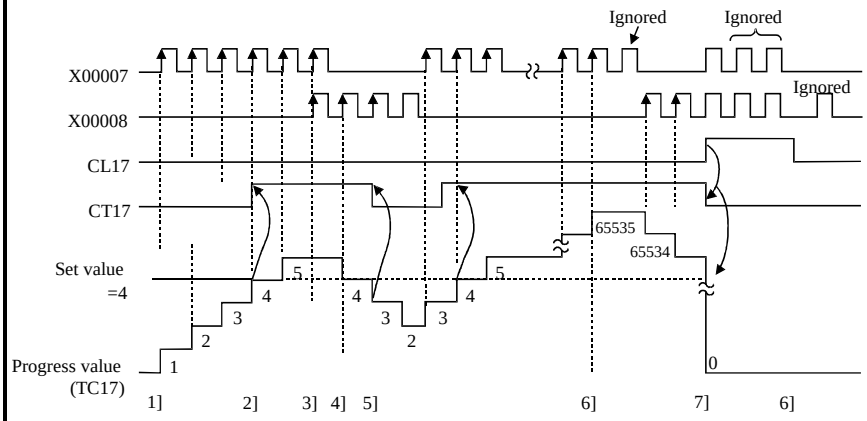
- An example of a word I/O being used as the set value for the ladder shown above.




```
LD     R7E3
[
WR0017=4
]
LD     X00007
OUT    CTU17   WR0017
LD     X00008
OUT    CTD17
LD     X00009
OUT    CL17
LD     CT17
OUT    R107
```

Program description

[Time chart]



1] The progress value is up-counted at the rising edge of X00007.

2] The counter coil (CT17) is turned on when set value ≧ progress value.

3] When the up-coil and down-coil startup conditions turn on simultaneously, the progress value does not change.

4] The progress value is down-counted at the rising edge of X00008.

5] The counter coil turns off when set value > progress value.

6] The progress value will never exceed 65,535 (decimal). Also, it will not be below 0.

7] When counter clear (CL17) turns on, the progress value and the counter coil are cleared. The progress value is not updated while the counter clear is on.

- The clear is performed under the conditions set immediately prior to the execution of the counter coil command.
- Example using word I/O as the set value
  When RUN is commenced, the set value is set to word I/O.
  Or, designate the word I/O for the set value to store in the power failure memory beforehand.

Item number	Basic commands-31	Name	Counter clear

Ladder format	Condition code					Processing time (μs)						Remark
CL n	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

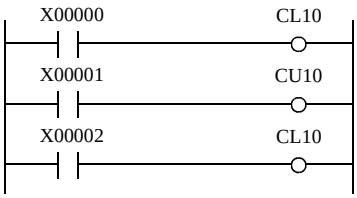
Command format	Number of steps								
	Condition	Steps	0.1	←	0.1 ..... 6.8	←	1.0	1.7	
OUT CL n s	—	1							

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
n	Counter number												○	0 to 511 (Decimal)

Function

- Clears the TD, SS and MS progress values of the integral timer and switches off the timer coil.
- In the case of WDT, a time monitor check is performed (see WDT for details).
- In the case of counters, the progress value is cleared and the counter coil is switched off.
- The clearing operation is conducted immediately before the counter or timer indicated by the clear coil executes the coil command.

Example



1) When X00000 is turned on, the CL10 immediately prior to CU10 turns on, and CU10 is cleared.

2) Even if X00002 turns on, if X00001 is off, the CL10 is turned off by the ladder before CU10 is executed. Therefore, the CU10 will not be cleared.

Cautionary notes

- The number same as on the timer or counter must be used.

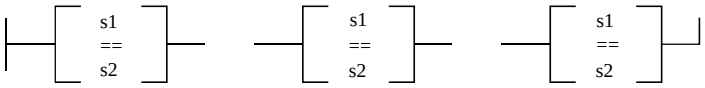
Item number	Basic commands-32	Name	=Relational box

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
(See Function column)	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					0.5 (7.5)	← (←)	23.5 51.1	91.9	76.7	185.6	
LD (s1 == s2)	Condition		Steps			15.8 (22.8)	18.4 (25.4)	26.5 71.3	123.5	100.2	130.1	
AND (s1 == s2)	Word		(See Cautionary notes)									
OR (s1 == s2)	Double word		(See Cautionary notes)									

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s1	Relational number 1					○	○	○	○	○	○	○	○	
s2	Relational number 2					○	○	○	○	○	○	○	○	

Function

[Ladder format]



- Compares s1 and s2 as unsigned numbers, and
  if s1 is equals to s2, it enters the continuity status (on) and
  if s1 is not equal to s2, enters the noncontinuity status (off).
- When s1 and s2 are words : 0 to 65,535 (decimal) or H0000 to HFFFF (hexadecimal)
  When s1 and s2 are double words : 0 to 4,294,967,295 (decimal) or H00000000 to HFFFFFFFF (hexadecimal)

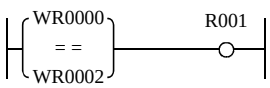
Cautionary notes

[Number of steps]

Word		
LD	(s1 == s2)	5 steps
AND	(s1 == s2)	5 steps
OR	(s1 == s2)	6 steps

Double word		LD, AND (s1==s2)	OR (s1==s2)
I/O	I/O	5 steps	6 steps
I/O	Constant	6 steps	7 steps
Constant	I/O	6 steps	7 steps
Constant	Constant	7 steps	8 steps

Program example




```
LD    (WR0000 == WR0002)
OUT   R001
```

Program description

- When WR0000 = WR0002, R001 turns on.

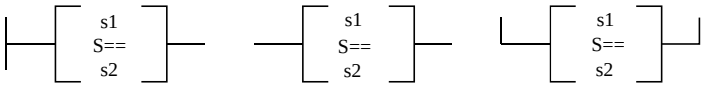
Item number	Basic commands-33	Name	Signed = Relational box

Ladder format	Condition code					Processing time (μs)						Remark
(See Function column)	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					18.4	25.4	26.5	←	76.7	130.1	
LD (s1 S== s2)	Condition		Steps									
AND (s1 S== s2)	Double word		(See Cautionary notes)					71.3	123.5			
OR (s1 S== s2)												

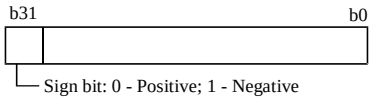
Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s1	Relational number 1									○	○	○	○	
s2	Relational number 2									○	○	○	○	

Function

[Ladder format]



- Compares s1 and s2 as signed double-word numbers, and if s1 is equals to s2, it enters the continuity status (on) and if s1 is not equal to s2, enters the noncontinuity status(off).
- s1, s2 – 2,147,483,648 to + 2,147,483,647 (decimal)
  H80000000 to H7FFFFFFF (hexadecimal)

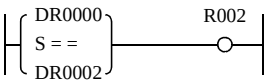


Cautionary notes

[Number of steps]

Double word		LD, AND (s1S==s2)	OR (s1S==s2)
I/O	I/O	5 steps	6 steps
I/O	Constant	6 steps	7 steps
Constant	I/O	6 steps	7 steps
Constant	Constant	7 steps	8 steps

Program example




```
LD    (DR0000 S== DR0002)
OUT   R002
```

Program description

- When DR0000 = DR0002, R002 turns on (signed).

Item number	Basic commands-34	Name	<> Relational box

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
(See Function column)	DER	ERR	SD	V	C							Upper case: W
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					0.5 (7.5)	← (←)	23.5 50.9	91.9	100.3	185.7	Lower case: DW
LD (s1 <> s2)	Condition		Steps			15.8 (22.8)	18.4 (25.4)	27.85 72.3	124.3	78.7	133.9	( ) indicates the case of OR
AND (s1 <> s2)	Word		(See Cautionary notes)									
OR (s1 <> s2)	Double word		(See Cautionary notes)									

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s1	Relational number 1					○	○	○	○	○	○	○	○	
s2	Relational number 2					○	○	○	○	○	○	○	○	

Function

[Ladder format]

- Compares s1 and s2 as unsigned numbers, and
  if s1 is equals to s2, it enters the noncontinuity status (off) and
  if s1 is not equal to s2, enters the continuity status (on).
- When s1 and s2 are words : 0 to 65,535 (decimal) or H0000 to HFFFF (hexadecimal)
  When s1 and s2 are double words : 0 to 4,294,967,295 (decimal) or H00000000 to HFFFFFFFF (hexadecimal)

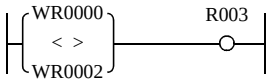
Cautionary notes

[Number of steps]

Word		
LD	(s1 <> s2)	5 steps
AND	(s1 <> s2)	5 steps
OR	(s1 <> s2)	6 steps

Double word		LD, AND (s1<>s2)	OR (s1<>s2)
I/O	I/O	5 steps	6 steps
I/O	Constant	6 steps	7 steps
Constant	I/O	6 steps	7 steps
Constant	Constant	7 steps	8 steps

Program example




```
LD    (WR0000 < > WR0002)
OUT   R003
```

Program description

- When WR0000 ≠ WR0002, R003 turns on.

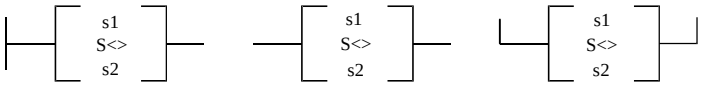
Item number	Basic commands-35	Name	Signed <> Relational box

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
(See Function column)	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					18.4	25.4	27.8	←	78.7	133.9	
LD (s1 S<> s2)	Condition		Steps									
AND (s1 S<> s2)	Double word		(See Cautionary notes)					72.3	124.3			
OR (s1 S<> s2)												

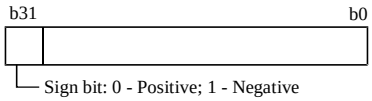
Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s1	Relational number 1									○	○	○	○	
s2	Relational number 2									○	○	○	○	

Function

[Ladder format]



- Compares s1 and s2 as signed double-word numbers, and if s1 is equals to s2, it enters the noncontinuity status (off) and if s1 is not equal to s2, enters the continuity status (on).
- s1, s2 – 2,147,483,648 to + 2,147,483,647 (decimal)
  H80000000 to H7FFFFFFF (hexadecimal)

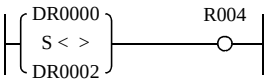


Cautionary notes

[Number of steps]

Double word		LD, AND (s1S<>s2)	OR (s1S<>s2)
I/O	I/O	5 steps	6 steps
I/O	Constant	6 steps	7 steps
Constant	I/O	6 steps	7 steps
Constant	Constant	7 steps	8 steps

Program example




```
LD    (DR0000 S < > DR0002)
OUT   R004
```

Program description

- When DR0000 ≠ DR0002, R004 turns on (signed).

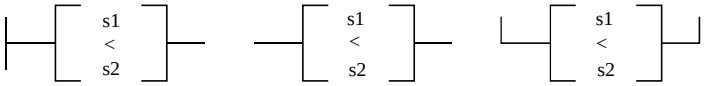
Item number	Basic commands-36	Name	<Relational box

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
(See Function column)	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					0.5 (7.5)		23.5 50.9	91.9	100.3	185.7	Upper case: W Lower case: DW ( ) indicates the case of OR
LD (s1 < s2)	Condition		Steps			16.5 (23.5)	19.4 (26.4)	28.8 74.0	126.7	80.6	136.2	
AND (s1 < s2)	Word		(See Cautionary notes)									
OR (s1 < s2)	Double word		(See Cautionary notes)									

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s1	Relational number 1					○	○	○	○	○	○	○	○	
s2	Relational number 2					○	○	○	○	○	○	○	○	

Function

[Ladder format]



- Compares s1 and s2 as unsigned numbers, and
  if s1 is less than s2, it enters the continuity status (on) and
  if s1 is greater than or equal to s2, enters the noncontinuity status (off).
- When s1 and s2 are words : 0 to 65,535 (decimal) or H0000 to HFFFF (hexadecimal)
  When s1 and s2 are double words : 0 to 4,294,967,295 (decimal) or H00000000 to HFFFFFFFF (hexadecimal)

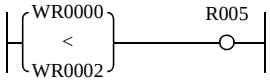
Cautionary notes

[Number of steps]

Word	
LD (s1 < s2)	5 steps
AND (s1 < s2)	5 steps
OR (s1 < s2)	6 steps

Double word		LD, AND (s1<s2)	OR (s1<s2)
I/O	I/O	5 steps	6 steps
I/O	Constant	6 steps	7 steps
Constant	I/O	6 steps	7 steps
Constant	Constant	7 steps	8 steps

Program example




```
LD    (WR0000 < WR0002)
OUT   R005
```

Program description

- When WR0000 < WR0002, R005 turns on.

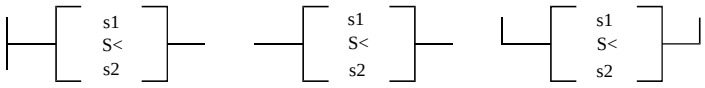
Item number	Basic commands-37	Name	Signed<Relational box

Ladder format	Condition code					Processing time (μs)						Remark
(See Function column)	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					19.4	26.4	24.8	28.8	80.9	136.7	
LD (s1 S< s2)	Condition		Steps									
AND (s1 S< s2)	Double word		(See Cautionary notes)					74.0	126.7			
OR (s1 S< s2)												

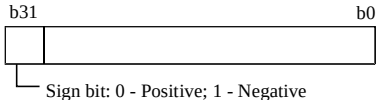
Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s1	Relational number 1									○	○	○	○	
s2	Relational number 2									○	○	○	○	

Function

[Ladder format]



- Compares s1 and s2 as signed double-word numbers, and if s1 is less than s2, it enters the continuity status (on) and if s1 is greater than or equal to s2, enters the noncontinuity status (off).
- s1, s2 – 2,147,483,648 to + 2,147,483,647 (decimal)
  H80000000 to H7FFFFFFF (hexadecimal)

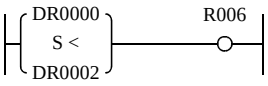


Cautionary notes

[Number of steps]

Double word		LD, AND (s1S<s2)	OR (s1S<s2)
I/O	I/O	5 steps	6 steps
I/O	Constant	6 steps	7 steps
Constant	I/O	6 steps	7 steps
Constant	Constant	7 steps	8 steps

Program example




```
LD    (DR0000 S< DR0002)
OUT   R006
```

Program description

- When DR0000 < DR0002, R006 turns on (signed).

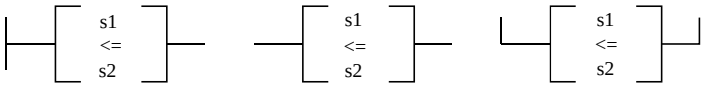
Item number	Basic commands-38	Name	≤ Relational box

Ladder format	Condition code					Processing time (μs)						Remark
(See Function column)	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		Upper case: W Lower case: DW ( ) indicates the case of OR
	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					0.5	←	23.5		100.2	185.6	
LD (s1 <= s2)	Condition		Steps			(7.5)	(←)	50.9	91.9			
AND (s1 <= s2)	Word		(See Cautionary notes)			16.5	19.4	28.85		80.7	136.4	
OR (s1 <= s2)	Double word		(See Cautionary notes)			(23.5)	(26.4)	74.0	126.7			

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s1	Relational number 1					○	○	○	○	○	○	○	○	
s2	Relational number 2					○	○	○	○	○	○	○	○	

Function

[Ladder format]



- Compares s1 and s2 as unsigned numbers, and
  if s1 is less than or equal to s2, it enters the continuity status (on) and
  if s1 is greater than s2, it enters the noncontinuity status (off).
- When s1 and s2 are words : 0 to 65,535 (decimal) or H0000 to HFFFF (hexadecimal)
  When s1 and s2 are double words : 0 to 4,294,967,295 (decimal) or H00000000 to HFFFFFFFF (hexadecimal)

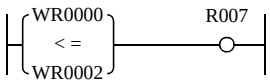
Cautionary notes

[Number of steps]

Word		
LD	(s1 <= s2)	5 steps
AND	(s1 <= s2)	5 steps
OR	(s1 <= s2)	6 steps

Double word		LD, AND (s1<=s2)	OR (s1<=s2)
I/O	I/O	5 steps	6 steps
I/O	Constant	6 steps	7 steps
Constant	I/O	6 steps	7 steps
Constant	Constant	7 steps	8 steps

Program example




```
LD    (WR0000 <= WR0002)
OUT   R007
```

Program description

- When WR0000 ≦ WR0002, R007 turns on.

Item number	Basic commands-39	Name	Signed ≤ Relational box

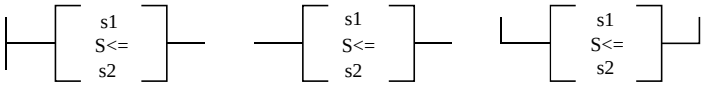
Ladder format	Condition code					Processing time (μs)						Remark
(See Function column)	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps									
LD (s1 S<= s2)	Condition	Steps	19.4	26.4	28.8	←	81.3	137.4		
AND (s1 S<= s2)	Double word	(See Cautionary notes)			74.0	126.7				
OR (s1 S<= s2)										

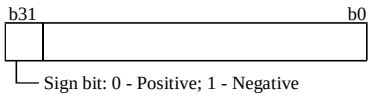
Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s1	Relational number 1									○	○	○	○	
s2	Relational number 2									○	○	○	○	

Function

[Ladder format]



- Compares s1 and s2 as signed double-word numbers, and if s1 is less than or equal to s2, it enters the continuity status (on) and if s1 is greater than s2, it enters the noncontinuity status (off).
- s1, s2 – 2,147,483,648 to + 2,147,483,647 (decimal)
  H80000000 to H7FFFFFFF (hexadecimal)

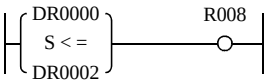


Cautionary notes

[Number of steps]

Double word		LD, AND (s1S<=s2)	OR (s1S<=s2)
I/O	I/O	5 steps	6 steps
I/O	Constant	6 steps	7 steps
Constant	I/O	6 steps	7 steps
Constant	Constant	7 steps	8 steps

Program example




```
LD    (DR0000 S<= DR0002)
OUT   R008
```

Program description

- When DR0000 ≦ DR0002, R008 turns on (signed).

Item number	Arithmeticcommands-1	Name	Substitution statement

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
d = s	DER	ERR	SD	V	C							
	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps		
	Condition	Steps	See following table
d = s	(See Cautionary notes)		

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

Function

- Substitutes the content of s into d.
- It is possible to use array variables for d and s.
- When d is a word, the constant is 0 to 65,535 or – 32,768 to + 32,767 (decimal)
  H0000 to HFFFF or H8000 to H7FFF (hexadecimal)
  When d is a double word, the constant is 0 to 4,294,967,295 or –2,147,483,648 to +2,147,483,647 (decimal)
  H00000000 to HFFFFFFFF or H80000000 to H7FFFFFFF

Cautionary notes

- When using an array variable, DER is set to 1 if the maximum usable I/O number is exceeded, and DER is reset to “0” if it is normal.
- The combinations of d and s are as follows:

d	s
Bit	Bit
Word	Word
Double word	Double word

- Step numbers and processing time are as follows:

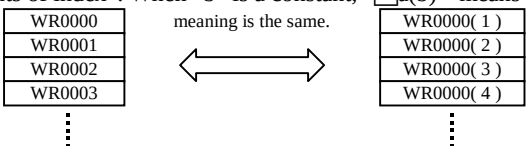
Model name	d	s	Number of steps ( ) indicates DW	Processing time (μ s)		
				Bit	Word	Double word
EH-CPU448	I/O	I/O	3 (4)	0.3	0.3	15
	I/O	Array	4	31	27	29
	Array	I/O	4 (5)	27	25	27
	Array	Array	5	43	39	42
Upper case: EH-CPU***A Lower case: EH-CPU3**	I/O	I/O	3 (4)	15.6 32	14.6 49	19.9 69
	I/O	Array	4	31.9 71	29.9 104	36.55 144
	Array	I/O	4 (5)	27.9 51	26.9 68	32.55 92
	Array	Array	5	40.05 86	37.9 119	46.55 165
Other than the above	I/O	I/O	3 (4)	71	84	69
	I/O	Array	4	172	250	173
	Array	I/O	4 (5)	101	182	125
	Array	Array	5	246	301	225

How to use Array variables

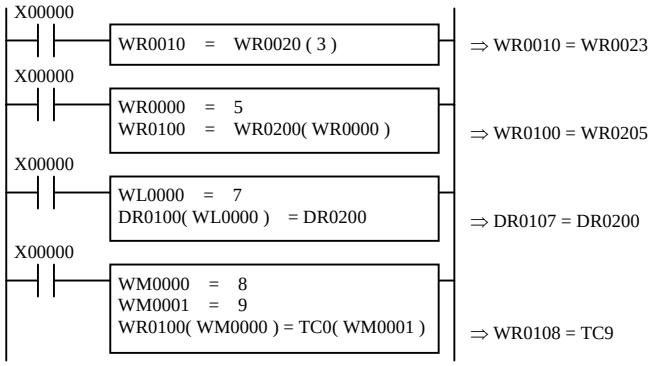
An array variable specifies I/O numbers by adding a constant or word I/O (WR, WM and WL) as an index to I/O (R, M, L, WR, WM, WL, TC, DR, DM, DL). Array variables can be used only for commands expressed by substitution formulas.

(1) Expression of array variables

An array variable is expressed by the form □a(b). □a is called "I/O of an array variable", and "b" in the parentheses is called "contents of index". When "b" is a constant, "□a(b) " means "□a + b".



(2) Example



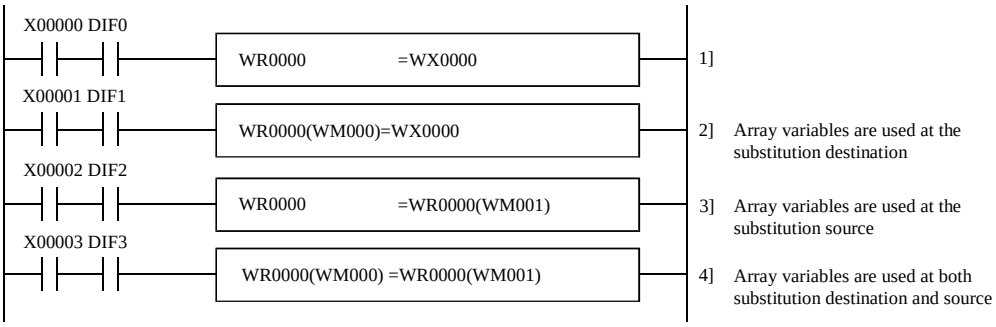
(3) Note

An array variables can be used only by the substitution formulas. The way of using it shown in the following can't be done.

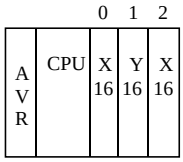
WR10(WR20) = WR100 + 1

R0 = WR10(WR20) < WR30

Program example



I/O assignment



Program description

1] The value of WX0000 is substituted into WR0000 at the rising edge of input X00000.

2] The value of WX0000 is substituted into the WR number designated by WR0000 + WM000 at the rising edge of input X00001.

1) When WM000 = H0010, it holds the same meaning as WR0010 = WX0000.

3] The word number of the I/O advanced by the amount designated by WR0000 + WM001 due to the I/O assignment is substituted into WR0000 at the rising edge of input X0002.

1) When WM001 = H0010, it hods the same meaning as WR0000 = WR0010.

4] The I/O value designated by WR0000 + WM001 at the rising edge of input X00003 is substituted into the I/O of the value designated by WR0000 + WM000.

Example) When WM000 = H0010 and WM001 = H0015, it holds the same meaning as WR0010 = WR0015.

Item number	Arithmeticcommands-2	Name	Binary addition

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
d = s1 + s2	DER	ERR	SD	V	C							
	●	●	●	↕	↕	Ave	Max	Ave	Max	Ave	Max	Upper case: W Lower case: DW

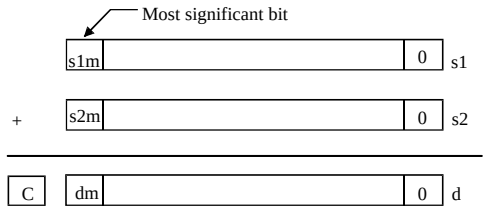
Command format	Number of steps									
	Condition	Steps	0.4	←	20	←	123	←		
d = s1 + s2					56	←				
	Word	4	22	←	31	←	100	←		
	Double word	6			80	←				

Usable I/O		Bit				Word				Double word				
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM	Constant	Other
d	Substitution destination						○	○	○		○	○		
s1	Augend					○	○	○	○	○	○	○	○	
s2	Addend					○	○	○	○	○	○	○	○	

Function

- Adds s1 and s2 as binary data, and substitutes the result into d as binary data.
- The C flag is 0 if the operation result is within the range of H0000 to HFFFF for word and H00000000 to HFFFFFFFF for double word. It is 1 otherwise.
  $C = s1m \cdot s2m + s1m \cdot \overline{dm} + s2m \cdot \overline{dm}$
- The V flag is 1 if the operation result is meaningless as signed binary data, and 0 if it is meaningful.

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



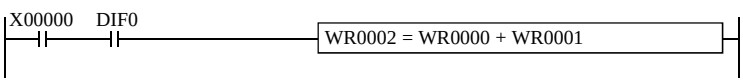
$V = s1m \cdot s2m \cdot \overline{dm} + \overline{s1m} \cdot \overline{s2m} \cdot dm$

Cautionary notes

- The combinations of d, s1 and s2 are as follows:

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

Program example




```
LD    X00000
AND   DIF0
[
WR0002 = WR0000 + WR0001
]
```

Program description

- The sum of the values in WR0000 and WR0001 is substituted into WR0002 at the rising edge of input X00000.

Item number	Arithmeticcommands-3	Name	BCD addition

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
d = s1 B+ s2	DER	ERR	SD	V	C							
	↕	●	●	●	↕	Ave	Max	Ave	Max	Ave	Max	Upper case: W Lower case: DW
Command format	Number of steps					36	←	41	←	193	←	
d = s1 B+ s2	Condition			Steps				82	←			
	Word			4		59	←	69	←	246	←	
	Double word			6				129	←			

Usable I/O		Bit				Word				Double word				Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM	Constant	
d	Substitution destination						○	○	○		○	○		
s1	Augend					○	○	○	○	○	○	○	○	
s2	Addend					○	○	○	○	○	○	○	○	

Function

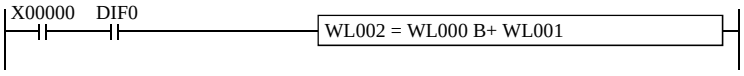
- Adds s1 and s2 as BCD data, and stores the result in d as BCD data.
- The C flag is 1 if there is a digit increase, and 0 if not.
- The DER flag is 1 if the operation result s1 and s2 are invalid BCD data. In such cases, operation is not performed and the C flag retains the previous state without outputting to d. If the s1 and s2 are valid BCD data, the DER is set to 0.
- When s1, s2 are words : 0000 to 9999 (BCD)
- When s1, s2 are double words : 00000000 to 99999999 (BCD)

Cautionary notes

- The combinations of d, s1 and s2 are as follows.

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

Program example




```
LD    X00000
AND  DIF0
[
WL002 = WL000 B+ WL001
]
```

Program description

- The sum of the values of WL000 and WL001 is substituted into WL002 as BCD data at the rising edge of input X00000.

Item number	Arithmeticcommands-4	Name	Binary subtraction

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
d = s1 – s2	DER	ERR	SD	V	C							
	●	●	●	↕	↕	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps									
	Condition	Steps	0.4	←	20	←	121	←		
d = s1 – s2					54	←				
	Word	4	20	←	31	←	96	←		
	Double word	6			80	←				

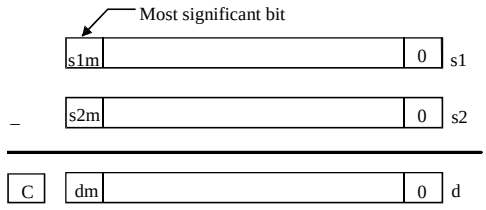
Upper case: W
Lower case: DW

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Substitution destination						○	○	○		○	○		
s1	Minuend					○	○	○	○	○	○	○	○	
s2	Subtrahend					○	○	○	○	○	○	○	○	

Function

- Subtracts s2 from s1 as binary data, and substitutes the result into d as binary data.
- The C flag is 1 if there is a digit decrease, and 0 if not.
  $C = \overline{s1m} \cdot s2m + \overline{s1m} \cdot dm + s2m \cdot dm$
- The V flag is 1 if the operation result is a meaningless signed-binary data, and 0 if it has meaning.

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



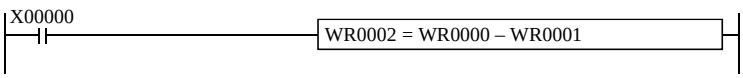
$V = \overline{s1m} \cdot s2m \cdot dm + s1m \cdot \overline{s2m} \cdot \overline{dm}$

Cautionary notes

- The combinations of d, s1 and s2 are as follows:

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

Program example




```
LD   X00000
[
WR0002 = WR0000 – WR0001
]
```

Program description

- When input X00000 is on, the difference between the values in WR0000 and WR0001 is substituted into WR0002.

Item number	Arithmeticcommands-5	Name	BCD subtraction

Ladder format	Condition code					Processing time (μs)						Remark
d = s1 B– s2	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		Upper case: W Lower case: DW
	DER	ERR	SD	V	C							
	↕	●	●	●	↕	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps									
d = s1 B– s2	Condition	Steps	34	←	39	←	193	←		
					79	←				
	Word	4	54	←	63	←	247	←		
	Double word	6			117	←				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Substitution destination						○	○	○		○	○		
s1	Minuend					○	○	○	○	○	○	○	○	
s2	Subtrahend					○	○	○	○	○	○	○	○	

Function

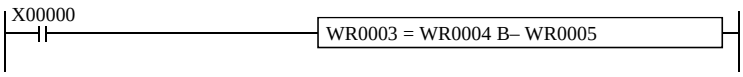
- Subtracts s2 from s1 as BCD data, and substitutes the result into d as BCD data.
- The C flag is 1 if there is a digit decrease, and 0 if not.
- The DER flag is 1 if s1 or s2 is not a valid BCD data. In such cases, operation is not performed and the C flag retains the previous state without outputting to d. If the s1 and s2 are valid BCD data, the DER is set to 0.

Cautionary notes

- The combinations of d, s1 and s2 are as follows:

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

Program example




```
LD   X00000
[
WR0003 = WR0004 B– WR0005
]
```

Program description

- When input X00000 turns on, the difference between the values in WR0004 and WR0005 is substituted into WR0003 as BCD data.

Item number	Arithmeticcommands-6	Name	Binary multiplication

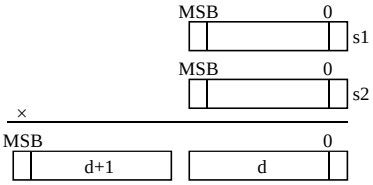
Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
d = s1 × s2	DER	ERR	SD	V	C							
	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	Upper case: W Lower case: DW

Command format	Number of steps									
	Condition	Steps	30	←	31	←	87	←		
d = s1 × s2					76	←				
	Word	4	39	←	45	←	139	←		
	Double word	6			85	←				

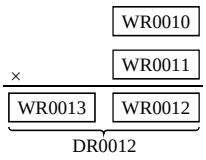
Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Substitution destination						○	○	○		○	○		
s1	Multiplicand					○	○	○	○	○	○	○	○	
s2	Multiplier					○	○	○	○	○	○	○	○	

Function

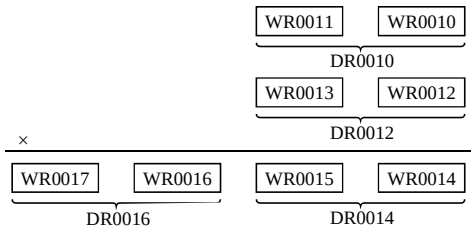
- Multiplies s1 and s2 as binary data, and substitutes the result into d+1 (upper digit) and d (lower digit) in binary.
- The DER flag is 1 if d+1 has exceeded the usable I/O range (in this case only the lower word is substituted), and 0 when it is not exceeded.



Example) WR0012 = WR0010 × WR0011



Example) DR0014 = DR0010 × DR0012



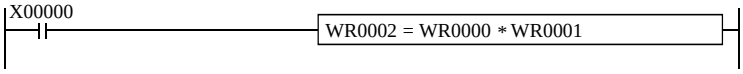
Cautionary notes

- The combinations of d, s1 and s2 are as follows:

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

- Since the operation results are always substituted into d and d + 1, exercise caution so that the word or double-word at d + 1 is not used as the I/O of others.

Program example




```
LD   X00000
[
WR0002 = WR0000 * WR0001
]
```

Program description

- When input X00000 turns on, the product of the values in WR0000 and WR0001 is substituted into WR0002.

Item number	Arithmeticcommands-7	Name	BCD multiplication

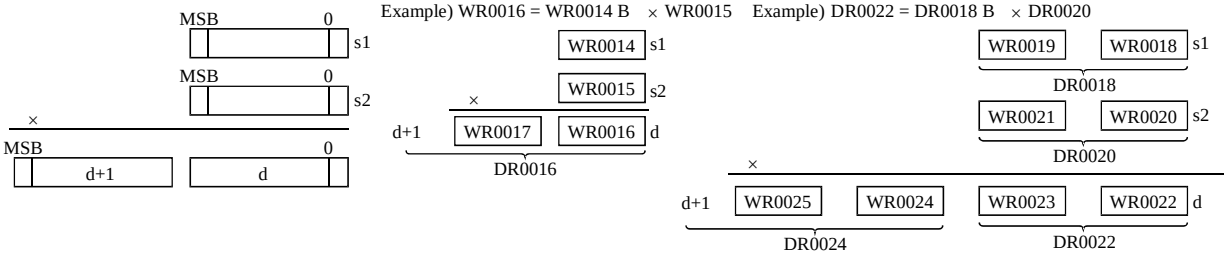
Ladder format	Condition code					Processing time (μs)						Remark
d = s1 B× s2	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		Upper case: W Lower case: DW
	DER	ERR	SD	V	C							
	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps									
d = s1 B× s2	Condition	Steps	63	←	64	←	227	←		
					110	←				
	Word	4	132	←	137	←	323	←		
	Double word	6			203	←				

Usable I/O		Bit				Word				Double word				Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM	Constant	
d	Substitution destination						○	○	○		○	○		
s1	Multiplicand					○	○	○	○	○	○	○	○	
s2	Multiplier					○	○	○	○	○	○	○	○	

Function

- Multiplies s1 and s2 as BCD data, and substitutes the result into d+1 (upper digit) and d (lower digit) as BCD.
- The DER flag is 1 if s1 or s2 is an invalid BCD data. In this case the operation is not performed. Also, if d+1 exceeds the usable I/O range, the DER flag is set to 1 and only the lower digit word is substituted. The DER flag is 0 if s1 and s2 are valid BCD data and d+1 is within the usable I/O range.



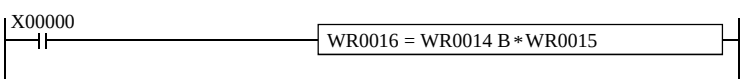
Cautionary notes

- The combinations of d, s1 and s2 are as follows:

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

- Since the operation results are always substituted into d and d + 1, exercise caution so that the word or double-word at d + 1 is not used as the I/O of others.

Program example




```
LD   X00000
[
WR0016 = WR0014 B * WR0015
]
```

Program description

- When input X00000 turns on, the product of the values in WR0014 and WR0015 is substituted into WR0016 as BCD data.

Item number	Arithmeticcommands-8	Name	Signed binary multiplication

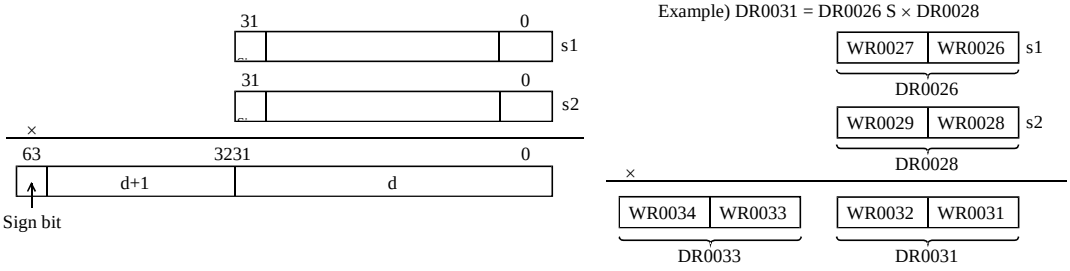
Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
d = s1 S× s2	DER	ERR	SD	V	C							
	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps										
	Condition	Steps	42	←	48 / 92	←	143	←			
d = s1 S× s2	Double word	6									

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Substitution destination										○	○		
s1	Multiplicand									○	○	○	○	
s2	Multiplier									○	○	○	○	

Function

- Multiplies s1 and s2 as signed binary data, and substitutes the result into d+1 (upper digit) and d (lower digit) as signed binary.
- The DER flag is 1 if d+1 exceeds the usable I/O range (in this case only the lower digit word is substituted), and 0 when it does not.



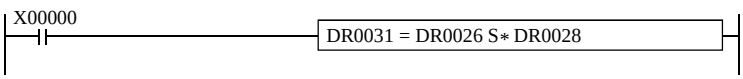
The sign of the operation result is entered in the most significant bit.

- s1, s2 – 2,147,483,648 to +2,147,483,647 (decimal)
  H80000000 to H7FFFFFFF (hexadecimal)

Cautionary notes

- The operation result is always assigned to d and d+1. Be sure not to use word or double word d+1 as the I/O of other functions.

Program example




```
LD      X00000
[
DR0031 = DR0026 S∗ DR0028
]
```

Program description

- When input X00000 turns on, the product of the values in DR0026 and DR0028 is substituted into DR0031 as signed binary data.

Item number	Arithmeticcommands-9	Name	Binary division

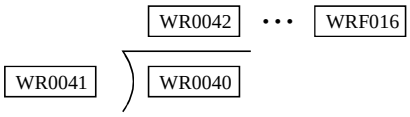
Ladder format	Condition code					Processing time (μs)						Remark
d = s1 / s2	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		Upper case: W Lower case: DW
	DER	ERR	SD	V	C							
	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					28	←	33		138	←	
d = s1 / s2	Condition			Steps				72	←			
	Word			4		32	←	42	←	111	←	
	Double word			6				95	←			

Usable I/O		Bit				Word				Double word				Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM	Constant	
d	Substitution destination						○	○	○		○	○		
s1	Dividend					○	○	○	○	○	○	○	○	
s2	Divisor					○	○	○	○	○	○	○	○	

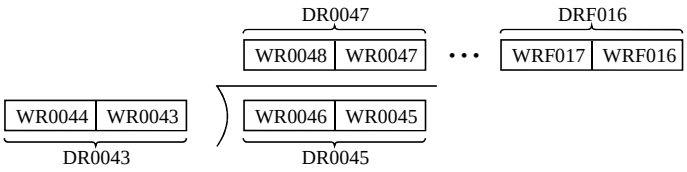
Function

- Divides s1 by s2 and substitutes the quotient into d in binary. The remainder is set in the special internal output WRF016 (DRF016 in the case of double word).
- The DER flag is 1 if s2 is 0 and the operation is not performed. As long as s2 is not 0, the flag is 0 and the operation is performed.

Example) WR0042 = WR0040 / WR0041



Example) DR0047 = DR0045 / DR0043

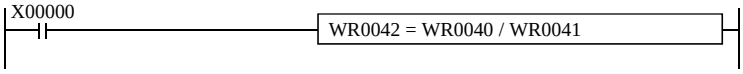


Cautionary notes

- The combinations of d, s1 and s2 are as follows:

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

Program example




```
LD   X00000
[
WR0042 = WR0040 / WR0041
]
```

Program description

- When input X00000 turns on, the value in WR0040 is divided by the value in WR0041, then substituted into WR0042. The remainder is substituted into special internal output WRF016.

Item number	Arithmeticcommands-10	Name	BCD division

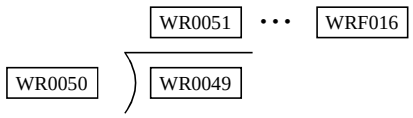
Ladder format	Condition code					Processing time (μs)						Remark
d = s1 B/ s2	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		Upper case: W Lower case: DW
	DER	ERR	SD	V	C							
	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					50	←	55	←	236	←	
d = s1 B/ s2	Condition			Steps				101	←			
	Words			4		69	←	78	←	372	←	
	Double word			6				141	←			

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Substitution destination						○	○	○		○	○		
s1	Dividend					○	○	○	○	○	○	○	○	
s2	Divisor					○	○	○	○	○	○	○	○	

Function

- Divides s1 by s2 as BCD data, and substitutes the quotient into d in BCD. The remainder is set in the special internal output WRF016 (DRF016 in the case of double word).
- The DER flag is 1 if s1 or s2 is an invalid BCD data or when s2 is 0. In this case the operation is not performed. If both s1 and s2 are valid BCD data and s2 is not 0, the operation is performed.

Example) WR0051 = WR0049 B/ WR0050



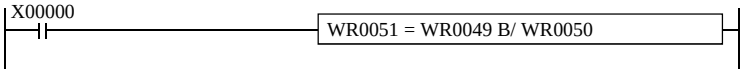
- When s1, s2 are words : 0000 to 9999 (BCD)
- When s1, s2 are double words : 00000000 to 99999999 (BCD)

Cautionary notes

- The combinations of d, s1 and s2 are as follows:

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

Program example




```
LD   X00000
[
WR0051 = WR0049 B/ WR0050
]
```

Program description

- When input X00000 turns on, the value in WR0049 is divided by the value in WR0050, then substituted into WR0051 as BCD data.
  The remainder is substituted into WRF016 as BCD data.

Item number	Arithmeticcommands-11	Name	Signed binary division

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
d = s1 S/ s2	DER	ERR	SD	V	C							
	↕	●	●	↕	●	Ave	Max	Ave	Max	Ave	Max	

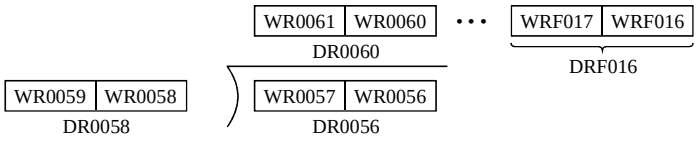
Command format	Number of steps									
	Condition		Steps	35	←	44	←	111	←	
d = s1 S/ s2	Double word		6			97				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Substitution destination										○	○		
s1	Dividend									○	○	○	○	
s2	Divisor									○	○	○	○	

Function

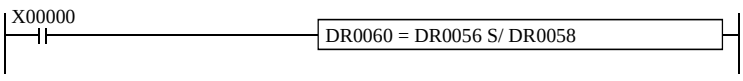
- Divides s1 by s2 as signed binary data, and substitutes the quotient into d in signed binary data. The remainder is set in the special internal output DRF016 signed binary data.
- The DER flag is 1 if s2 is 0, and the operation is not performed. As long as s2 is not 0, it is 0 and the operation is performed.
- The V flag is 1 when the quotient is a positive value and exceeds H7FFFFFFF. Otherwise, it is 0.

Eexample) DR0060 = DR0056 S/ DR0058



- s1, s2 – 2,147,483,648 to +2,147,483,647 (decimal)
  H80000000 to H7FFFFFFF (hexadecimal)

Program example




```
LD   X00000
[
DR0060 = DR0056 S/ DR0058
]
```

Program description

- When input X00000 turns on, the value of DR0056 is divided by the value in DR0058, then substituted into DR0060 as signed binary data. The remainder is substituted into special internal output DRF016 as signed binary data.

Item number	Arithmeticcommands-12	Name	Logical OR

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
d = s1 OR s2	DER	ERR	SD	V	C	Ave	Max	Ave	Max	Ave	Max	Upper case: B Middle case: W Lower case: DW
	●	●	●	●	●	0.4	←	19	←	76	←	
Command format	Number of steps							36	←			
						0.4	←	21	←	89	←	
d = s1 OR s2	Condition			Steps				55	←			
	Bit, word			4		18	←	24	←	78	←	
	Double word			6				72	←			

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Substitution destination		○	○			○	○	○		○	○		
s1	Comparand	○	○	○		○	○	○	○	○	○	○	○	
s2	Relational number	○	○	○		○	○	○	○	○	○	○	○	

Function

- Obtains OR of s1 and s2, and substitutes the result into d.

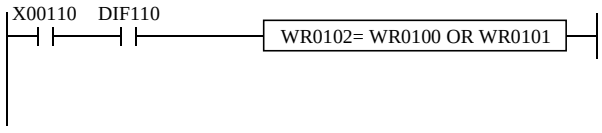
s1	s2	d
0	0	0
0	1	1
1	0	1
1	1	1

Cautionary notes

- The combinations of d, s1 and s2 are as follows:

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

Program example




```
LD    X00110
AND   DIF110
[
WR0102=WR0100 OR WR0101
]
```

Program description

- At the rising edge of X00110, the OR of WR0100 and WR0101 is set in WR0102.

WR0100 = H1234
WR0101 = H5678
WR0102 = H567C

When ⇨

WR0100 = 0001001000110100
WR0101 = 0101011001111000
WR0102 = 0101011001111100

Item number	Arithmeticcommands-13	Name	Logical AND

Ladder format	Condition code					Processing time (μs)						Remark
d = s1 AND s2	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		Upper case: B Middle case: W Lower case: DW
	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					0.4	←	19	←	76	←	
								36	←			
						0.4	←	18	←	89	←	
d = s1 AND s2	Condition		Steps					52	←			
	Bit, word		4			18	←	28	←	79	←	
	Double word		6					77	←			

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Substitution destination		○	○			○	○	○		○	○		
s1	Comparand	○	○	○		○	○	○	○	○	○	○	○	
s2	Relational number	○	○	○		○	○	○	○	○	○	○	○	

Function

- Obtains AND of s1 and s2, and substitutes the result into d.

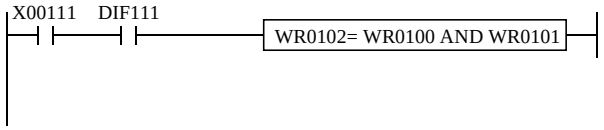
s1	s2	d
0	0	0
0	1	0
1	0	0
1	1	1

Cautionary notes

- The combinations of d, s1 and s2 are as follows:

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

Program example




```
LD    X00111
AND   DIF111
[
WR0102=WR0100 AND WR0101
]
```

Program description

- At the rising edge of X00111, the AND of WR0100 and WR0101 is set in WR0102.

WR0100 = H1234
WR0101 = H5678
WR0102 = H1230

When ⇨

WR0100 = 0001001000110100
WR0101 = 0101011001111000
WR0102 = 0001001000110000

Item number	Arithmeticcommands-14	Name	Exclusive OR

Ladder format	Condition code					Processing time (μs)						Remark
d = s1 XOR s2	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		Upper case: B Middle case: W Lower case: DW
	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					0.4	←	19	←	76	←	
								36	←			
d = s1 XOR s2	Condition			Steps		0.4	←	18	←	89	←	
								52	←			
	Bit, word			4		18	←	28	←	79	←	
	Double word			6				76	←			

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Substitution destination		○	○			○	○	○		○	○		
s1	Comparand	○	○	○		○	○	○	○	○	○	○	○	
s2	Relational number	○	○	○		○	○	○	○	○	○	○	○	

Function

- Obtains exclusive OR (XOR) of s1 and s2, and substitutes the result into d.

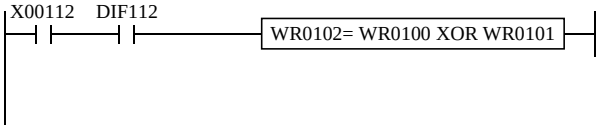
s1	s2	d
0	0	0
0	1	1
1	0	1
1	1	0

Cautionary notes

- The combinations of d, s1 and s2 are as follows:

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

Program example




```
LD    X00112
AND   DIF112
[
WR0102=WR0100 XOR WR0101
]
```

Program description

- At the rising edge of X00112, the XOR of WR0100 and WR0101 is set in WR0102.

WR0100 = H1234
WR0101 = H5678
WR0102 = H444C

When ⇨

WR0100 = 0001001000110100
WR0101 = 0101011001111000
WR0102 = 0100010001001100

Item number	Arithmeticcommands-15	Name	= Relational expression

Ladder format	Condition code					Processing time (μs)						Remark
d = s1 == s2	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		Upper case: W Lower case: DW
	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps									
d = s1 == s2	Condition	Steps	0.4	←	19	←	82	85		
					36	←				
	s is a word	4	18	←	27	←	84	85		
	s is a double word	6			44	←				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Substitution destination		○	○										
s1	Comparand					○	○	○	○	○	○	○	○	
s2	Relational number					○	○	○	○	○	○	○	○	

Function

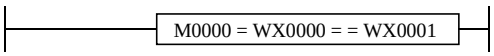
- Substitutes 1 when s1 is equal to s2 and otherwise 0 into d, assuming s1 and s2 as binary data.

Cautionary notes

- The combinations of d, s1 and s2 are as follows:

d	s1	s2
Bit	Word	Word
Bit	Double word	Double word

Program example




```
[
M0000 = WX0000 = = WX0001
]
```

Program description

- When WX0000 = WX0001, “1” is set in M0000. Otherwise, M0000 is reset to “0.”

Item number	Arithmeticcommands-16	Name	Signed = Relational expression

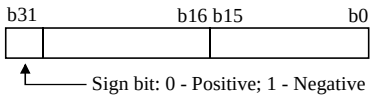
Ladder format	Condition code					Processing time (μs)						Remark
d = s1 S== s2	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps									
d = s1 S== s2	Condition	Steps	18	←	27	←	84	202		
	s is a double word	6			44	←				

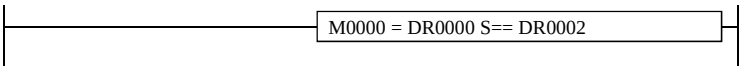
Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Substitution destination		○	○										
s1	Comparand									○	○	○	○	
s2	Relational number									○	○	○	○	

Function

- Substitutes 1 when s1 is equal to s2 and otherwise 0 into d, assuming s1 and s2 as signed binary data.
- s1 and s2 are both signed binary data. When the most significant bit is 0, the value is positive; when the most significant bit is 1, the value is negative.
  s1, s2 – 2,147,483,648 to +2,147,483,647 (decimal)
  H80000000 to H7FFFFFFF (hexadecimal)



Program example




```
[
M0000 = DR0000 S== DR0002
]
```

Program description

- When the values of DR0000 and DR0002 are equal, 1 is set in M0000. Otherwise, M0000 is reset to 0.

Item number	Arithmeticcommands-17	Name	<> Relational expression

Ladder format	Condition code					Processing time (μs)						Remark
d = s1 <> s2	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		Upper case: W Lower case: DW
	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps									
d = s1 <> s2	Condition	Steps	0.4	←	19	←	75	192		
					36	←				
	s is a word	4	18	←	27	←	84	202		
	s is a double Word	6			43	←				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Substitution destination		○	○										
s1	Comparand					○	○	○	○	○	○	○	○	
s2	Relational number					○	○	○	○	○	○	○	○	

Function

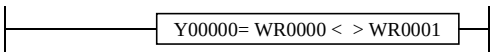
- Substitutes 1 when s1 is not equal to s2 and otherwise 0 into d, assuming s1 and s2 as binary data.

Cautionary notes

- The combinations of d, s1 and s2 are as follows:

d	s1	s2
Bit	Word	Word
Bit	Double word	Double word

Program example




```
[
Y00000= WR0000 < > WR0001
]
```

Program description

- When WR0000 ≠ WR0001, “1” is set in Y00000. Otherwise, Y00000 is reset to “0.”

Item number	Arithmeticcommands-18	Name	Signed <> Relational expression

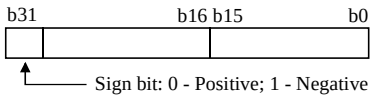
Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
d = s1 S<> s2	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps										
	Condition	Steps	18	←	27	←	85	←			
d = s1 S<> s2	s is a double word	6			43	←					

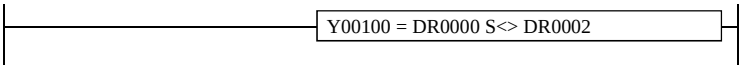
Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Substitution destination		○	○										
s1	Comparand									○	○	○	○	
s2	Relational number									○	○	○	○	

Function

- Substitutes 1 when s1 is not equal to s2 and otherwise 0 into d, assuming s1 and s2 as signed binary data.
- s1 and s2 are both signed binary data. When the most significant bit is 0, the value is positive; when the most significant bit is 1, the value is negative.
  s1, s2 – 2,147,483,648 to +2,147,483,647 (decimal)
  H80000000 to H7FFFFFFF (hexadecimal)



Program example




```
[
Y00100 = DR0000 S<> DR0002
]
```

Program description

- When the values of DR0000 and DR0002 are not equal, Y00100 is turned on. Otherwise, Y00100 is turned off.

Item number	Arithmeticcommands-19	Name	< Relational expression

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
d = s1 < s2	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	Upper case: W Lower case: DW

Command format	Number of steps										
	Condition	Steps	0.4	←	19	←	76	←			
d = s1 < s2					36	←					
	s is a word	4	19	←	27	←	86	←			
	s is a double word	6			44	←					

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Substitution destination		○	○										
s1	Comparand					○	○	○	○	○	○	○	○	
s2	Relational number					○	○	○	○	○	○	○	○	

Function

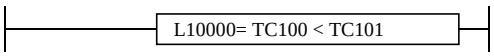
- Substitutes 1 when s1 is less than s2 and otherwise 0 into d, assuming s1 and s2 as binary data.

Cautionary notes

- The combinations of d, s1 and s2 are as follows:

d	s1	s2
Bit	Word	Word
Bit	Double word	Double word

Program example




```
[
L10000= TC100 < TC101
]
```

Program description

- When TC100 < TC101, 1 is set in L10000. Otherwise, L10000 is reset to 0.
  (TC n is the progress value of the no. n timer or counter.)

Item number	Arithmeticcommands-20	Name	Signed < Relational expression

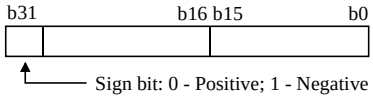
Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
d = s1 S< s2	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps										
	Condition	Steps									
d = s1 S< s2	s is a double word	6	19	←	27	←	86	←			
					45	←					

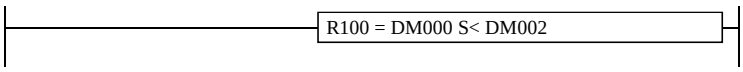
Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Substitution destination		○	○										
s1	Comparand									○	○	○	○	
s2	Relational number									○	○	○	○	

Function

- Substitutes 1 when s1 is less than s2 and otherwise 0 into d, assuming s1 and s2 as signed binary data.
- s1 and s2 are both signed binary data. When the most significant bit is 0, the value is positive; when the most significant bit is 1, the value is negative.
  s1, s2 – 2,147,483,648 to +2,147,483,647 (decimal)
  H80000000 to H7FFFFFFF (hexadecimal)



Program example




```
[
R100 = DM000 S< DM002
]
```

Program description

- When the value in DM000 is less than the value in DM002, 1 is set in R100. Otherwise, R100 is reset to 0.

Item number	Arithmeticcommands-21	Name	≤ Relational expression

Ladder format	Condition code					Processing time (μs)						Remark
d = s1 <= s2	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		Upper case: W Lower case: DW
	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps									
d = s1 <= s2	Condition	Steps	0.4	←	19	←	76	←		
					36	←				
	s is a word	4	19	←	27	←	86	←		
	s is a double word	6			45	←				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Substitution destination		○	○										
s1	Comparand					○	○	○	○	○	○	○	○	
s2	Relational number					○	○	○	○	○	○	○	○	

Function

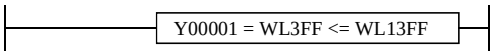
- Substitutes 1 when s1 is less than or equal to s2 and otherwise 0 into d, assuming s1 and s2 as binary data.

Cautionary notes

- The combinations of d, s1 and s2 are as follows:

d	s1	s2
Bit	Word	Word
Bit	Double word	Double word

Program example




```
[
Y00001 = WL3FF <= WL13FF
]
```

Program description

- When WL3FF ≦ WL13FF, 1 is set in Y00001. Otherwise, Y00001 is reset to 0.

Item number	Arithmeticcommands-22	Name	Signed ≤ Relational expression

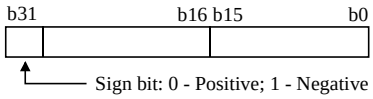
Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
d = s1 S<= s2	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps										
	Condition	Steps	19	←	27	←	86	←			
d = s1 S<= s2	s is a double word	6			45	←					

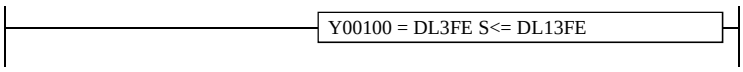
Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Substitution destination		○	○										
s1	Comparand									○	○	○	○	
s2	Relational number									○	○	○	○	

Function

- Substitutes 1 when s1 is less than or equal to s2 and otherwise 0 into d, assuming s1 and s2 as signed binary data.
- s1 and s2 are both signed binary data. When the most significant bit is 0, the value is positive; when the most significant bit is 1, the value is negative.
  s1, s2 – 2,147,483,648 to +2,147,483,647 (decimal)
  H80000000 to H7FFFFFFF (hexadecimal)



Program example




```
[
Y00100 = DL03FE S<= DL13FE
]
```

Program description

- When the value in DL3FE is less than or equal the value in DL13FE, Y00100 is turned on. Otherwise, Y00100 is turned off.

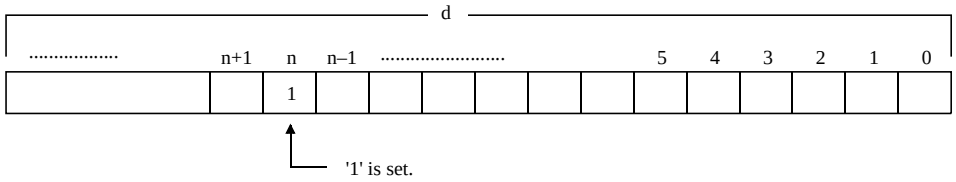
Item number	Application commands-1	Name	Bit set

Ladder format	Condition code					Processing time (μs)						Remark
BSET (d, n)	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		Upper case: W Lower case: DW
	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					13	←	15	←	86	←	
BSET (d, n)	Condition			Steps				50	←			
	—			3		15	←	17	←	66	←	
								66	←			

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	I/O to be set the bit						○	○	○		○	○		
n	Bit location to be set					○	○	○	○				○	The constant is set in decimal.

Function

- Sets the nth bit in the I/O (word or double word) specified by d to 1.
- Other bit contents are unaltered.



If d is a word : Designates the bit location depending on the contents (0 to 15) of the lower 4 bits (b3 to b0) of n (WX, WY, WR, WL, WM, TC). (Upper bits are ignored and considered 0.)
The n (constant) can be set to 0 to 15 (decimal).

If d is a double word : Designates the bit location depending on the contents (0 to 31) of the lower 5 bits (b4 to b0) of n (WX, WY, WR, WL, WM, TC). (Upper bits are ignored and considered 0.)
The n (constant) can be set to 0 to 31 (decimal).

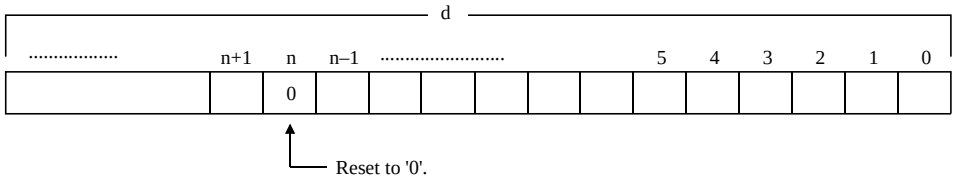
Item number	Application commands-2	Name	Bit reset

Ladder format	Condition code					Processing time (μs)						Remark
BRES (d, n)	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		Upper case: W Lower case: DW
	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					13	←	15	←	86	←	
	Condition			Steps				50	←			
BRES (d, n)	—			3		15	←	17	←	66	←	
								66	←			

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	I/O to be set the bit						○	○	○		○	○		
n	Bit location to be reset					○	○	○	○				○	The constant is set in decimal.

Function

- Sets the nth bit in the I/O (word or double word) specified by d to 0.
- Other bit contents are unaltered.



If d is a word : Designates the bit location depending on the contents (0 to 15) of the lower 4 bits (b3 to b0) of n (WX, WY, WR, WL, WM, TC). (Upper bits are ignored and considered 0.)
The n (constant) can be set to 0 to 15 (decimal).

If d is a double word : Designates the bit location depending on the contents (0 to 31) of the lower 5 bits (b4 to b0) of n (WX, WY, WR, WL, WM, TC). (Upper bits are ignored and considered 0.)
The n (constant) can be set to 0 to 31 (decimal).

Item number	Application commands-3	Name	Bit test

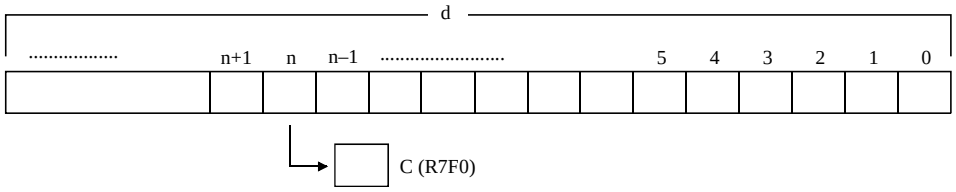
Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
BTS (d, n)	DER	ERR	SD	V	C							
	●	●	●	●	↕	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps										
	Condition		Steps		13	←	16	←	86	←	Upper case: W Lower case: DW
BTS (d, n)							50	←			
	—		3		15	←	18	←	67	←	
							67	←			

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	I/O to be tested						○	○	○		○	○		
n	Bit location to be tested					○	○	○	○				○	The constant is set in decimal.

Function

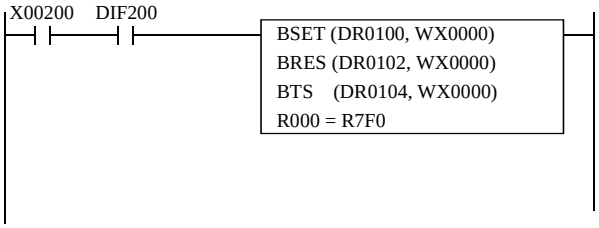
- Checks the contents of the nth bit of the I/O (word or double word) specified by d, and if the result is 1, "1" is set to C (R7F0). If the result is 0, C (R7F0) is reset to "0".
- The contents of d remains unaltered.



If d is a word : Designates the bit location depending on the contents (0 to 15) of the lower 4 bits (b3 to b0) of n (WX, WY, WR, WL, WM, TC). (Upper bits are ignored and considered 0.)
The n (constant) can be set to 0 to 15 (decimal).

If d is a double word : Designates the bit location depending on the contents (0 to 31) of the lower 5 bits (b4 to b0) of n (WX, WY, WR, WL, WM, TC). (Upper bits are ignored and considered 0.)
The n (constant) can be set to 0 to 31 (decimal).

Program example




```
LD   X00200
AND  DIF200
[
BSET (DR0100, WX0000)
BRES (DR0102, WX0000)
BTS  (DR0104, WX0000)
R000 = R7F0
]
```

Program description

When WX0000 = H1234 at the rising edge of X00200 (WX0000 = 0001001000110100)

20 (decimal)

If DR0100 = H00000000, DR0102 = HFFFFFFFF and DR0104 = H5555AAAA are set, the 20th bit of DR0100 is set to “1” by the BSET at the rising edge of X00200.

```
        b31 ——— b20 ——————— b0
DR0100=00000000000000000000000000000000
              ↑
        This bit is set to "1."
```

Also, the 20th bit of DR0102 is reset to “0” by BRES.

```
        b31 ——— b20 ——————— b0
DR0102=11111111111111111111111111111111
              ↑
        This bit is set to "0."
```

Also, the 20th bit of DR0104 is checked by BTS.

```
        b31 ——— b20 ——————— b0
DR0104=01010101010101011010101010101010
              ↑
        This bit is checked.
      Since the 20th bit is "1," (R7F0) = "1" is set.
```

Item number	Application commands-4	Name	Shift right

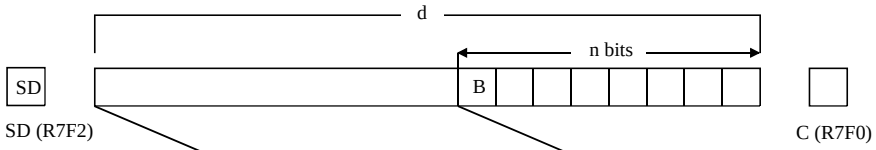
Ladder format	Condition code					Processing time (μs)						Remark
SHR (d, n)	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		Upper case: W Lower case: DW
	DER	ERR	SD	V	C							
	●	●	●	●	↕	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					17	←	19	←	90	←	
SHR (d, n)	Condition			Steps				53	←			
	—			3		18	←	21	←	71	←	
								71	←			

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	I/O to be shifted						○	○	○		○	○		
n	Number of bits to be shifted					○	○	○	○				○	The constant is set in decimal.

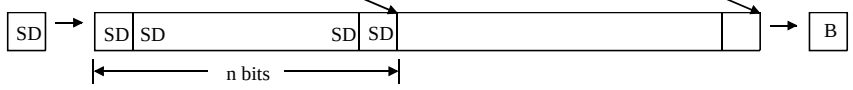
Function

- Shifts the contents of d to the right (toward the lower digits) by n bits.
- Sets n bits of SD (R7F2) contents starting with the most significant bit.
- Sets the content of the nth bit from the least significant bit in C (R7F0).

Before execution



After execution



Most significant bit (MSB) Least significant bit (LSB)

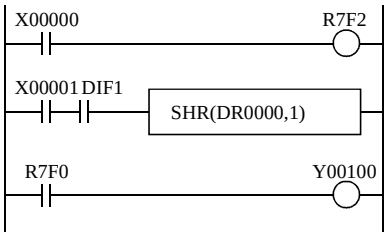
If d is a word : Designates the shift amount, depending on the contents (0 to 15) of the lower 4 bits (b3 to b0) of n (WX, WY, WR, WL, WM, TC). (Upper bits are ignored and considered 0.)
The n (constant) can be set to 0 to 15 (decimal).

If d is a double word : Designates the shift amount, depending on the contents (0 to 31) of the lower 5 bits (b4 to b0) of n (WX, WY, WR, WL, WM, TC). (Upper bits are ignored and considered 0.)
The n (constant) can be set to 0 to 31 (decimal).

Cautionary notes

- If n is equal to 0, the shift is not performed. The previous state is retained in C.

Program example



X00000 · · · · · · · Defective unit input To SD

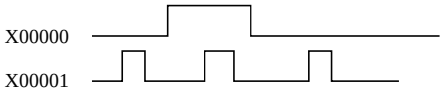
X00001 · · · · · · · Conveyor movement

Y00100 · · · · · · · Defective unit output Carry

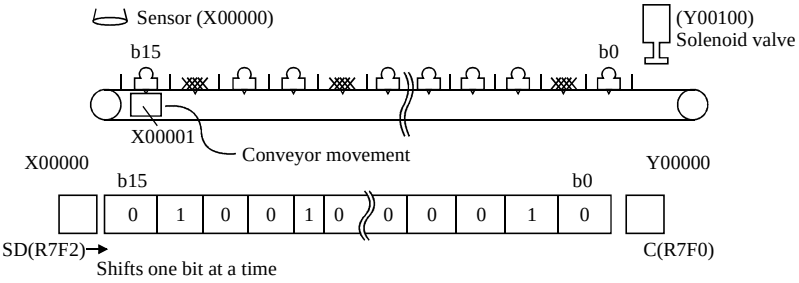
```
LD    X00000
OUT   R7F2
LD    X00001
AND   DIF1
[
SHR   (DR0000,1)
]
LD    R7F0
OUT   Y00100
```

Program description

- There exists a conveyor that has 16 stands and is moving to the right.
- Each time the conveyor moves one stand to the right, a pulse input enters X00001.
- There is a sensor on the left end of the conveyor, and when a defective unit is placed on the conveyor, X00000 turns on. X00000 (sensor input) and X00001 (conveyor movement) signals are as follows:



- As the conveyor moves to the right, the data is also shifted one bit at a time, and when data exits to the carry (on the right end of the conveyor), the (Y00100) solenoid valve turns on and rejects the defective unit.



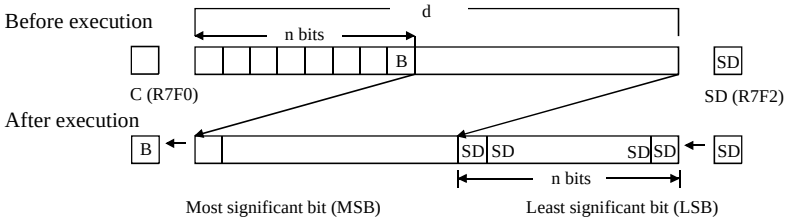
Item number	Application commands-5	Name	Shift left

Ladder format	Condition code					Processing time (μs)						Remark
SHL (d, n)	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		Upper case: W Lower case: DW
	DER	ERR	SD	V	C							
	●	●	●	●	↕	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					16	←	19	←	90	←	
								54	←			
SHL (d, n)	Condition			Steps		18	←	21	←	71	←	
	—			3				71	←			

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	I/O to be shifted						○	○	○		○	○		
n	Number of bits to be shifted					○	○	○	○				○	The constant is set in decimal.

Function

- Shifts the contents of d to the left (toward the upper digits) by n bits.
- Sets n bits of SD (R7F2) contents starting with the least significant bit.
- Sets the content of the nth bit from the most significant bit in C (R7F0).



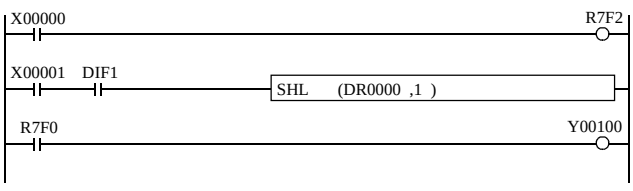
If d is a word : Designates the shift amount, depending on the contents (0 to 15) of the lower 4 bits (b3 to b0) of n (WX, WY, WR, WL, WM, TC). (Upper bits are ignored and considered 0.)
The n (constant) can be set to 0 to 15 (decimal).

If d is a double word : Designates the shift amount, depending on the contents (0 to 31) of the lower 5 bits (b4 to b0) of n (WX, WY, WR, WL, WM, TC). (Upper bits are ignored and considered 0.)
The n (constant) can be set to 0 to 31 (decimal).

Cautionary notes

- If n is equal to 0, the shift is not performed. The previous state is retained in C.

Program example




```
LD    X00000
OUT   R7F2
LD    X00001
AND   DIF1
[
SHL(DR0000,1)
]
LD    R7F0
OUT   Y00100
```

Program description

- The R7F2 value of the DR0000 after the shift is determined by the on/off of X00000.
- The content of DR0000 is shifted to the left when X00001 rises.
  At this time, the value of R7F2 is set in b0 and the value of b31 (b15 of WR0001) in R7F0.
- The Y00100 turns on/off depending on the b31 value of DR0000 (b15 of WR0001) prior to the shift.

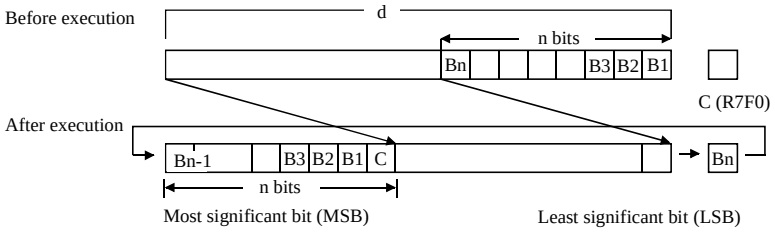
Item number	Application commands-6	Name	Rotate right

Ladder format	Condition code					Processing time (μs)						Remark
ROR (d, n)	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		Upper case: W Lower case: DW
	DER	ERR	SD	V	C							
	●	●	●	●	↕	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					21	←	23	←	96	←	
	Condition			Steps				60	←			
ROR (d, n)	—			3		22	←	25	←	77	←	
								76	←			

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	I/O to be rotated						○	○	○		○	○		
n	Number of bits to be rotated					○	○	○	○				○	The constant is set in decimal.

Function

- Rotates the contents of d to the right (toward the lower digits) by n bits.
- The content of the least significant bit is input to C (R7F0) while the content of C (R7F0) is input to the most significant bit. This is repeated n times.
- The content of C (R7F0) is set in the nth bit from the most significant bit.
- The content of the nth bit from the least significant bit is set in C (R7F0).



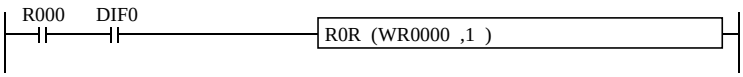
If d is a word : Designates the shift amount, depending on the contents (0 to 15) of the lower 4 bits (b3 to b0) of n (WX, WY, WR, WL, WM, TC). (Upper bits are ignored and considered 0.)
The n (constant) can be set to 0 to 15 (decimal).

If d is a double word : Designates the shift amount, depending on the contents (0 to 31) of the lower 5 bits (b4 to b0) of n (WX, WY, WR, WL, WM, TC). (Upper bits are ignored and considered 0.)
The n (constant) can be set to 0 to 31 (decimal).

Cautionary notes

- If n is equal to 0, the rotation is not performed. The previous state is retained in C.

Program example




```
LD    R000
AND   DIF0
[
R0R   (WR0000,1)
]
```

Program description

- When R000 rises, WR0000 is shifted to the right by one bit.
  At this time, the value of the least significant bit, b0, is set in R7F0, and the value of R7F0 immediately prior to the is set in the most significant bit, b15.

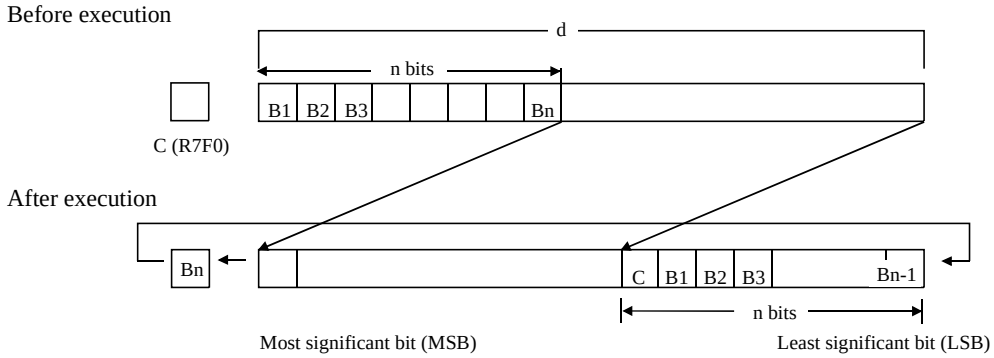
Item number	Application commands-7	Name	Rotate left

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
ROL (d, n)	DER	ERR	SD	V	C							
	●	●	●	●	↕	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					21	←	23	←	95	←	Upper case: W Lower case: DW
	Condition			Steps				59	←			
ROL (d, n)	—			3		22	←	25	←	77	←	
								76	←			

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	I/O to be rotated						○	○	○		○	○		
n	Number of bits to be rotated					○	○	○	○				○	The constant is set in decimal.

Function

- Rotates the contents of d to the left (toward the upper digits) by n bits.
- The content of C (R7F0) is set in the nth bit from the least significant bit.
- The content of the nth bit from the least significant bit is set in C (R7F0).



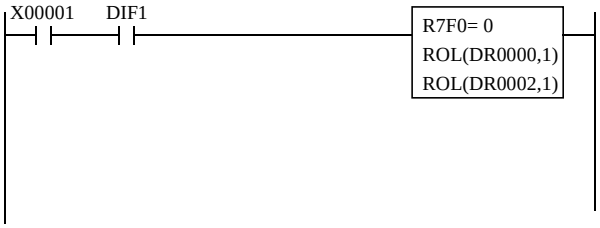
If d is a word : Designates the shift amount, depending on the contents (0 to 15) of the lower 4 bits (b3 to b0) of n (WX, WY, WR, WL, WM, TC). (Upper bits are ignored and considered 0.)
The n (constant) can be set to 0 to 15 (decimal).

If d is a double word : Designates the shift amount, depending on the contents (0 to 31) of the lower 5 bits (b4 to b0) of n (WX, WY, WR, WL, WM, TC). (Upper bits are ignored and considered 0.)
The n (constant) can be set to 0 to 31 (decimal).

Cautionary notes

- If n is equal to 0, the rotation is not performed. The previous state is retained in C.

Program example

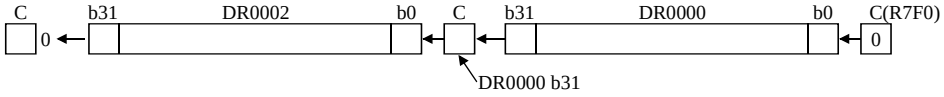



```
LD      X00001
AND     DIF1
[
R7F0= 0
ROL     (DR0000,1)
ROL     (DR0002,1)
]
```

Program description

- When X00001 rises, the 64 bit data is shifted one bit at a time.
  The open area after the shift is filled with "0."

Overall movement



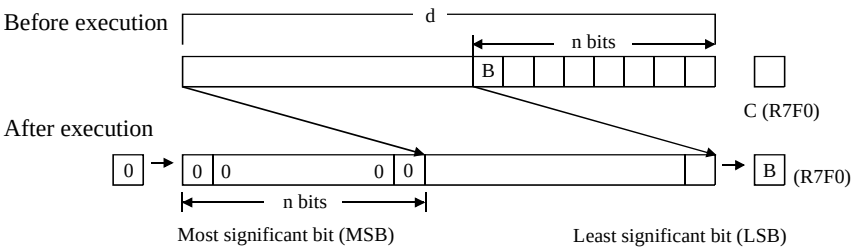
Item number	Application commands-8	Name	Logical shift right

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
LSR (d, n)	DER	ERR	SD	V	C							
	●	●	●	●	↕	Ave	Max	Ave	Max	Ave	Max	Upper case: W Lower case: DW
Command format	Number of steps					16	←	18	←	89	←	
	Condition			Steps				53	←			
LSR (d, n)	—			3		18	←	20	←	71	←	
								70	←			

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	I/O to be shifted						○	○	○		○	○		
n	Number of bits to be shifted					○	○	○	○				○	The constant is set in decimal.

Function

- Shifts the contents of d to the right (toward the lower digits) by n bits.
- "0" is set from the most significant bit to the nth bit.
- The content of the nth bit from the least significant bit is set in C (R7F0).



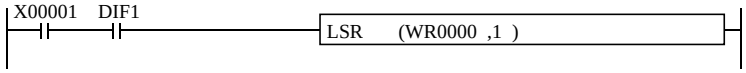
If d is a word : Designates the shift amount, depending on the contents (0 to 15) of the lower 4 bits (b3 to b0) of n (WX, WY, WR, WL, WM, TC). (Upper bits are ignored and considered "0".)
The n (constant) can be set to 0 to 15 (decimal).

If d is a double word : Designates the shift amount, depending on the contents (0 to 31) of the lower 5 bits (b4 to b0) of n (WX, WY, WR, WL, WM, TC). (Upper bits are ignored and considered "0".)
The n (constant) can be set to 0 to 31 (decimal).

Cautionary notes

- If n is equal to 0, the shift is not performed. The previous state is retained in C.

Program example




```
LD    X00001
AND   DIF1
[
LSR   (WR0000 ,1)
]
```

Program description

- When X00001 rises, the content of WR0000 is shifted to the right by one bit.
  At this time, "0" is set in b15 and the value of b0 immediately prior to the shift is set in R7F0.

Item number	Application commands-9	Name	Logical shift left

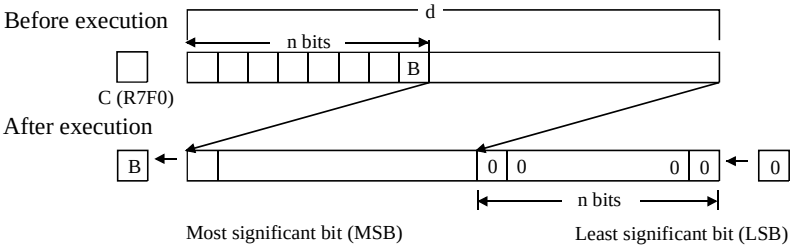
Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
LSL (d, n)	DER	ERR	SD	V	C							
	●	●	●	●	↕	Ave	Max	Ave	Max	Ave	Max	Upper case: W Lower case: DW

Command format	Number of steps		Processing time (μs)						
	Condition	Steps							
LSL (d, n)	—	3	18	←	20	←	71	←	
					70	←			

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	I/O to be shifted						○	○	○		○	○		
n	Number of bits to be shifted					○	○	○	○				○	The constant is set in decimal.

Function

- Shifts the contents of d to the left (toward the upper digits) by n bits.
- 0 is set from the least significant bit to the nth bit.
- The content of the nth bit from the most significant bit is set in C (R7F0).



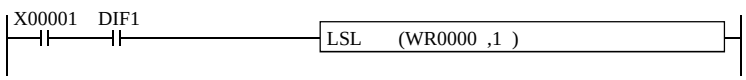
If d is a word : Designates the shift amount, depending on the contents (0 to 15) of the lower 4 bits (b3 to b0) of n (WX, WY, WR, WL, WM, TC). (Upper bits are ignored and considered "0".)
The n (constant) can be set to 0 to 15 (decimal).

If d is a double word : Designates the shift amount, depending on the contents (0 to 31) of the lower 5 bits (b4 to b0) of n (WX, WY, WR, WL, WM, TC). (Upper bits are ignored and considered "0".)
The n (constant) can be set to 0 to 31 (decimal).

Cautionary notes

- If n is equal to 0, the shift is not performed. The previous state is retained in C.

Program example




```
LD    X00001
AND   DIF1
[
LSL   (WR0000 ,1)
]
```

Program description

- When X00001 rises, the content of WR0000 is shifted to the left by one bit.
  At this time, "0" is set in b0 and the value of b15 immediately prior to the shift is set in R7F0.

Item number	Application commands-10	Name	BCD shift right

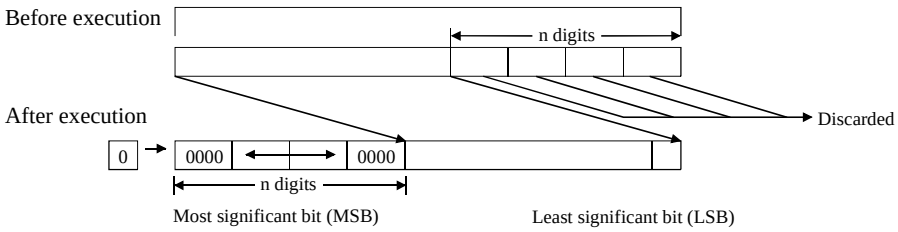
Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
BSR (d, n)	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps										
	Condition		Steps	15	←	17	←	88	←	Upper case: W Lower case: DW
BSR (d, n)	—		3			52	←				
				17	←	20	←	70	←		
						70	←				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	I/O to be shifted						○	○	○		○	○		
n	Number of digits to be shifted					○	○	○	○				○	The constant is set in decimal.

Function

- Shifts the contents of d to the right (toward the lower digits) by n digits (1 digit is equivalent to 4 bits).
- 0 is set from the most significant bit to the nth digit.
- The digits from least significant bit to the nth digit are discarded.



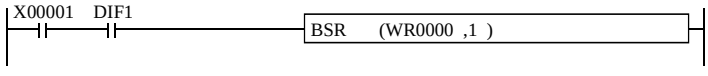
If d is a word : Designates the shift amount, depending on the contents (0 to 3) of the lower 2 bits (b1, b0) of n (WX, WY, WR, WL, WM, TC). (Upper bits are ignored and considered “0”.)
The n (constant) can be set to 0 to 3 (decimal).

If d is a double word : Designates the shift amount, depending on the contents (0 to 7) of the lower 3 bits (b2 to b0) of n (WX, WY, WR, WL, WM, TC). (Upper bits are ignored and considered “0”.)
The n (constant) can be set to 0 to 7 (decimal).

Cautionary notes

- If n is equal to 0, the shift is not performed.

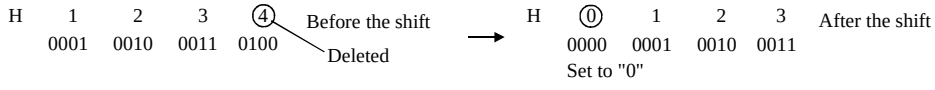
Program example




```
LD    X00001
AND   DIF1
[
BSR   (WR0000 ,1)
]
```

Program description

- When X00001 rises, the content of WR0000 is regarded as BCD code and shifted to the right by four bits.
  At this time, the values in the lower 4 bits (b3 to b0) are deleted and “0000” is set in the upper four bits (b15 to b12).



Item number	Application commands-11	Name	BCD shift left

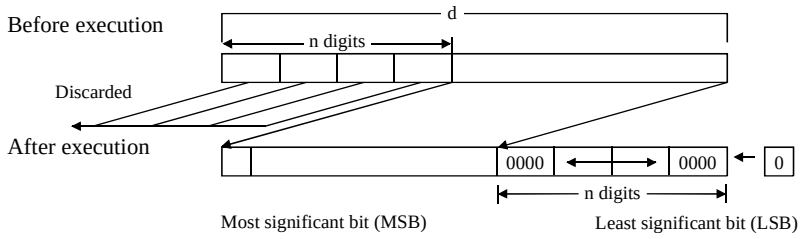
Ladder format	Condition code					Processing time (μs)						Remark
BSL (d, n)	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		Upper case: W Lower case: DW
	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps									
	Condition	Steps	15	←	17	←	88	←		
BSL (d, n)					52	←				
	—	3	17	←	20	←	70	←		
					69	←				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	I/O to be shifted						○	○	○		○	○		
n	Number of digits to be shifted					○	○	○	○				○	The constant is set in decimal.

Function

- Shifts the contents of d to the left (toward the upper digits) by n digits (1 digit is equivalent to 4 bits).
- 0 is set from the least significant bit to the nth digit.
- The digits from the most significant bit to the nth digit are discarded.



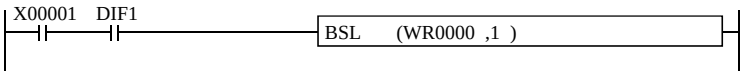
If d is a word : Designates the shift amount, depending on the contents (0 to 3) of the lower 2 bits (b1, b0) of n (WX, WY, WR, WL, WM, TC). (Upper bits are ignored and considered “0”.)
The n (constant) can be set to 0 to 3 (decimal).

If d is a double word : Designates the shift amount, depending on the contents (0 to 7) of the lower 3 bits (b2 to b0) of n (WX, WY, WR, WL, WM, TC). (Upper bits are ignored and considered “0”.)
The n (constant) can be set to 0 to 7 (decimal).

Cautionary notes

- If n is equal to 0, the shift is not performed.

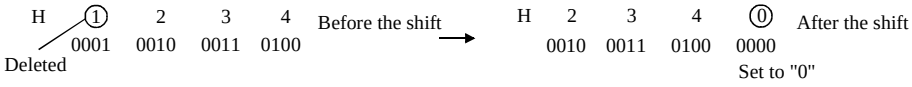
Program example




```
LD    X00001
AND   DIF1
[
BSL   (WR0000 ,1)
]
```

Program description

- When X00001 rises, the content of WR0000 is regarded as BCD code and shifted to the left by four bits.
  At this time, the data of the lower 4 bits are deleted and “0000” is set in the upper four bits.



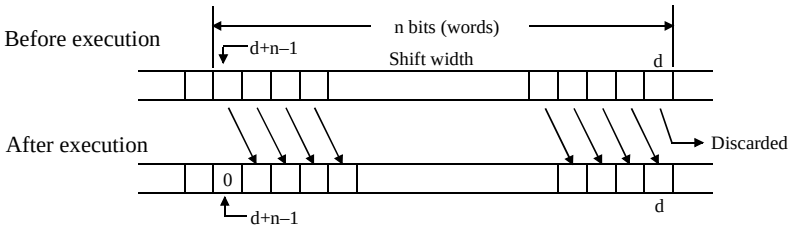
Item number	Application commands-12	Name	Batch shift right (SHIFT RIGHT BLOCK)

Ladder format	Condition code					Processing time (μs)			Remark
WSHR (d, n)	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**	EH-CPU***A EH-CPU3**	Other than left	Upper case: B Lower case: W
	DER	ERR	SD	V	C				
	↕	●	●	●	●	Ave	Ave	Ave	
Command format	Number of steps					31.3+0.08n	Note.1(margin) 54.6+0.09n	45.1+0.08n	
WSHR (d, n)	Condition			Steps					
	—			3		23.1+0.08n	22.5+0.76n 64.6+0.75n	57+0.8n	

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Head I/O to be shifted			○				○						
n	Number of bits (words) to be shifted					○	○	○	○				○	The constant is set in decimal.

Function

- Shifts n bits (words) between d and d + n – 1 to the right (toward smaller I/O number) by one bit (word).
- 0 (H0000) is set to the d + n – 1 bit (word).
- The content of d is discarded.



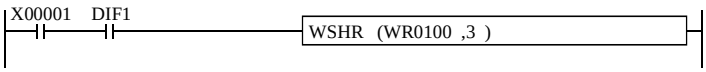
If n is a word: The contents (0 to 255) of the lower 8 bits (b7 to b0) of n (WX, WY, WR, WL, WM, TC) are set to the number of shifted bits (words).

If n is a constant: 0 to 255 (decimal) can be designated for the number of bits (words) to be shifted.

Cautionary notes

- Use this command so that d + n – 1 does not exceed the I/O range*. If the I/O range is exceeded, DER is equal to "1" and the shift is performed at the maximum range from d.
  * For I/O ranges, refer to the P3-6 and P3-7 performance specification table.
- If n is equal to 0, the block shift is not performed and DER (R7F4) will be "0".

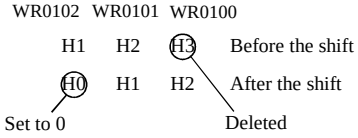
Program example




```
LD    X00001
AND   DIF1
[
WSHR (WR0100 ,3)
]
```

Program description

- When X00001 rises, the contents of WR0102, WR0101 and WR0100 are shifted to the right by one word.



Note.1: 28.9+1.32*(n/16)

Item number	Application commands-13	Name	Batch shift left (SHIFT LEFT BLOCK)

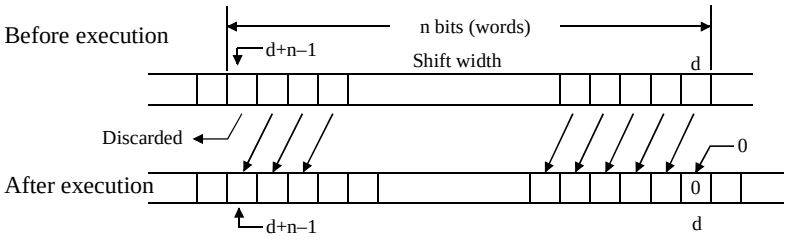
Ladder format	Condition code					Processing time (μs)			Remark
WSHL (d, n)	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**	EH-CPU***A EH-CPU3**	Other than left	Upper case: B Lower case: W
	DER	ERR	SD	V	C				
	↕	●	●	●	●	Ave	Ave	Ave	

Command format	Number of steps				
WSHL (d, n)	Condition	Steps	30.7+0.11n	Note.1(margin) 55.0+0.11n	43.3+0.11 n
	—	3	24.1+0.75n	23.6+0.75n 64.6+0.75n	58+0.75n

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Head I/O to be shifted			○				○						
n	Number of bits (words) to be shifted					○	○	○	○				○	The constant is set in decimal.

Function

- Shifts n bits (words) between d and d + n – 1 to the left (toward greater I/O number) by one bit (word).
- 0 (H0000) is set to the bit (word) for d.
- The content of d + n –1 is discarded.



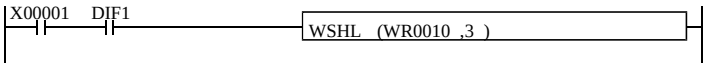
If n is a word : The contents (0 to 255) of the lower 8 bits (b7 to b0) of n (WX, WY, WR, WL, WM, TC) are set to the number of shifted bits (words).

If n is a constant : 0 to 255 (decimal) can be designated for the number of bits (words) to be shifted.

Cautionary notes

- Use this command so that d + n – 1 does not exceed the I/O range *. If the I/O range is exceeded, DER is equal to "1" and the shift is performed at the maximum range from d.
  * For I/O ranges, refer to the P3-6 and P3-7 performance specification table.
- If n is equal to 0, the block shift will not occur and DER (R7F4) will be "0".

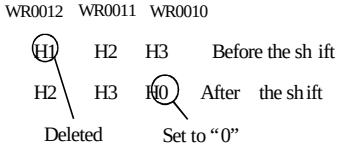
Program example




```
LD    X00001
AND   DIF1
[
WSHL  (WR0010 ,3)
]
```

Program description

- When X00001 rises, the contents of WR0010, WR0011 and WR0012 are shifted to the left by one word.



Note.1: 28.7+1.60*(n/16)

Item number	Application commands-14	Name	Batch shift right (BCD SHIFT RIGHT BLOCK)

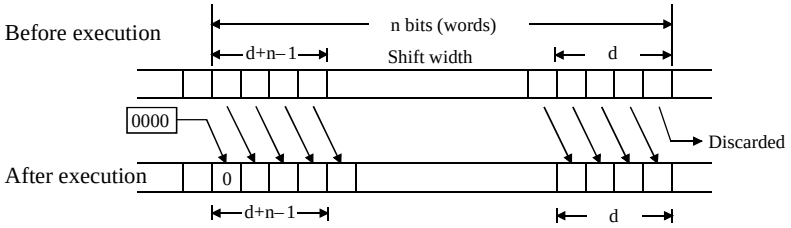
Ladder format	Condition code					Processing time (μs)			Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**	EH-CPU***A EH-CPU3**	Other than left	
WBSR (d, n)	DER	ERR	SD	V	C				
	↕	●	●	●	●	Ave	Ave	Ave	

Command format	Number of steps		Processing time (μs)			Remark
	Condition	Steps				
WBSR (d, n)	—	3	23.6+1.3n	22.8+1.3n 64.6+1.3n	57.5+1.3n	

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Head I/O to be shifted							○						
n	Number of words to be shifted					○	○	○	○				○	The constant is set in decimal.

Function

- Shifts n words between d and d + n – 1 to the right (toward smaller I/O number) by one digit (1 digit is equivalent to 4 bits) as BCD data.
- 0 is set to the most significant digit of d + n – 1.
- The content of the least significant digit of d is discarded.



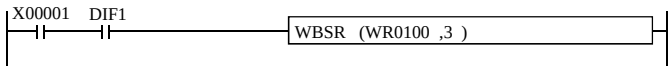
If n is a word : The contents (0 to 255) of the lower 8 bits (b7 to b0) of n (WX, WY, WR, WL, WM, TC) are set to the number of shifted words.

If n is a constant : 0 to 255 (decimal) can be designated for the number of words to be shifted.

Cautionary notes

- Use this command so that d + n – 1 does not exceed the I/O range *. If the I/O range is exceeded, DER is equal to "1" and the shift is performed at the maximum range.
  * For I/O ranges, refer to the P3-6 and P3-7 performance specification table.
- If n is equal to 0, the block shift is not performed and DER (R7F4) will be "0".

Program example




```
LD    X00001
AND   DIF1
[
WBSR  (WR0100 ,3)
]
```

Program description

- When X00001 rises, the contents of WR0100, WR0101 and WR0102 are regarded as BCD code and shifted to the right by four bits.

WR0102	WR0101	WR0100	
H1234	H5678	H1234	Before the shift
H0123	H4567	H8123	After the shift

Set to "0" Deleted

Item number	Application commands-15	Name	Batch shift left (BCD SHIFT LEFT BLOCK)

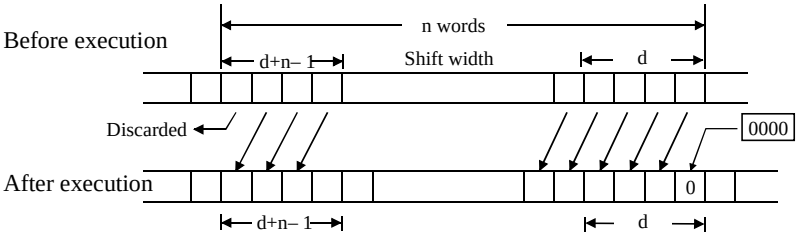
Ladder format	Condition code					Processing time (μs)			Remark
WBSL (d, n)	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**	EH-CPU***A EH-CPU3**	Other than left	
	DER	ERR	SD	V	C				
	↕	●	●	●	●	Ave	Ave	Ave	

Command format	Number of steps				
WBSL (d, n)	Condition	Steps	23.8+1.4n	23.2+1.35n 64.6+1.4n	57.5+1.4n
	—	3			

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Head I/O to be shifted							○						
n	Number of words to be shifted					○	○	○	○				○	The constant is set in decimal.

Function

- Shifts n words between d and d + n – 1 to the left (toward greater I/O number) by one digit (1 digit is equivalent to 4 bits) as BCD data.
- "0" is set to the least significant digit of d + n – 1.
- The content of the most significant digit of d is discarded.



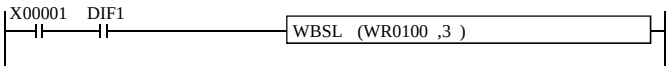
If n is a word : The contents (0 to 255) of the lower 8 bits (b7 to b0) of n (WX, WY, WR, WL, WM, TC) are set to the number of words to be shifted.

If n is a constant : 0 to 255 (decimal) can be designated for the number of words to be shifted.

Cautionary notes

- Use this command so that d + n – 1 does not exceed the I/O range *. If the I/O range is exceeded, DER is equal to "1" and the shift is performed at the maximum range.
  * For I/O ranges, refer to the P3-6 and P3-7 performance specification table.
- If n is equal to 0, the block shift is not performed and DER (R7F4) will be "0".

Program example




```
LD    X00001
AND   DIF1
[
WBSL  (WR0100 ,3)
]
```

Program description

- When X00001 rises, the contents of WR0100, WR0101 and WR0102 are regarded as BCD code and shifted to the left by four bits.

WR0102	WR0101	WR0100	
H1234	H5678	H1234	Before the shift
H2345	H6781	H2340	After the shift

Deleted

Set to "0"

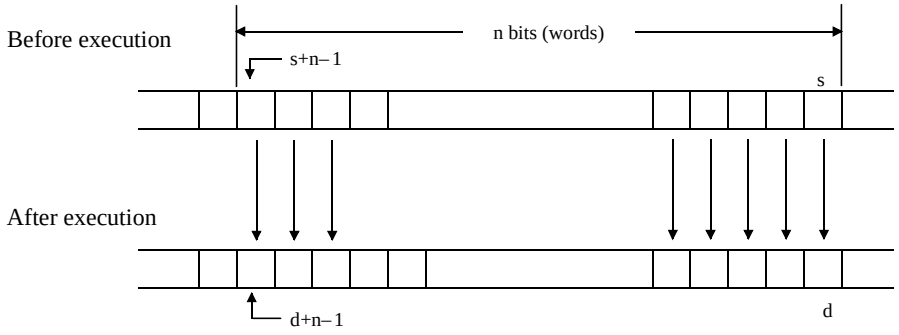
Item number	Application commands-16	Name	Block transfer (MOVE)

Ladder format	Condition code					Processing time (μs)			Remark
MOV (d, s, n)	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**	EH-CPU***A EH-CPU3**	Other than left	Upper case: B Lower case: W
	DER	ERR	SD	V	C				
	↕	●	●	●	●	Ave	Ave	Ave	
Command forma	Number of steps					172+112a a:Quotient of n/32	Note.1(margin) 260+165a a:Quotient of n/32	245+164a a:Quotient of n/32	
MOV (d, s, n)	Condition			Steps		148+80b b:Quotient of n/64	Note.2(margin) 201+91b b:Quotient of n/64	264+91b b:Quotient of n/64	
	—			4					

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Transfer destination head I/O			○				○						
s	Transfer source head I/O			○				○						
n	Number of bits (words) to be transferred					○	○	○	○				○	The constant is set in decimal.

Function

- Transfers n bits (words) between s and s + n – 1 to d + n – 1.
- The values between s and s + n – 1 are retained. However, if the transfer source and transfer destination ranges overlap, the transferred values will be used.



If n is a word : The contents (0 to 255) of the lower 8 bits (b7 to b0) of n (WX, WY, WR, WL, WM, TC) are set to the number of bits (words) to be transferred.

If n is a constant : 0 to 255 (decimal) can be designated for the number of bits (words) to be transferred.

Cautionary notes

- Use this command so that d + n – 1 and s + n – 1 do not exceed the I/O range *. If the I/O range is exceeded, DER is equal to "1" and the transfer is performed to the maximum range.

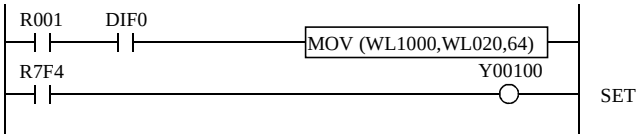
  * For I/O ranges, refer to the P3-6 and P3-7 performance specification table.
- If n is equal to 0, the block transfer is not performed and DER (R7F4) will be "0".

Note.1 : 57.33+112a(a:Quotient of n/32)+2.9b(b: Remainder of n/32) With no specification of an index.
59.9+112a(a: Quotient of n/32)+3b Those of an index with specification.

Note.2 : 58.3+80a(a:Quotient of n/64)+2.2b(b: (Remainder of n/64) / Quotient of 2) With no specification of an index.
62.2+80a(a: Quotient of n/64)+2.2b(b: (Remainder of n/64) / Quotient of 2) Those of an index with specification.

Program example

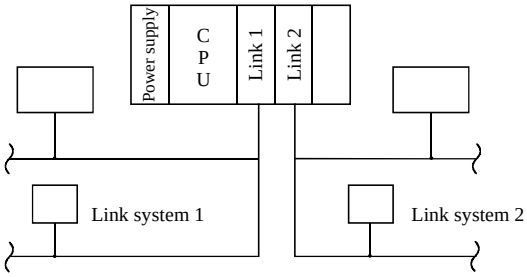
- The data in the first link area (WL020 to WL05F) is transferred to the second link area (WL1000 to WL103F).

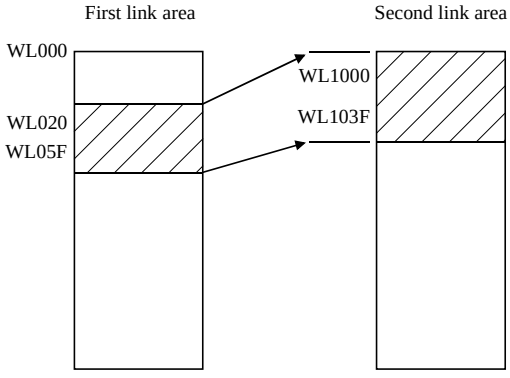



```
LD     R001
AND    DIF0
[
MOV    (WL1000,WL020,64)
]
LD     R7F4
SET    Y00100
```

Program description

- 64 words of data are transferred from link system 1 of the first link to link system 2 of the second link.
  WL020 to WL05F and WL1000 to WL103F are used as the respective transfer areas.





Item number	Application commands-17	Name	Copy

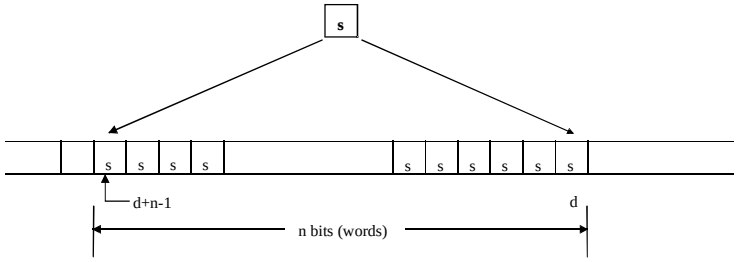
Ladder format	Condition code					Processing time (μs)			Remark
COPY (d, s, n)	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**	EH-CPU***A EH-CPU3**	Other than left	Upper case: B Lower case: W
	DER	ERR	SD	V	C				
	↕	●	●	●	●	Ave	Ave	Ave	
Command format	Number of steps					97+64a a:Quotient of n/32	Note.1(margin) 137+95a a:Quotient of n/32	124+95a a:Quotient of n/32	
	Condition		Steps						
COPY (d, s, n)	—		4			128+41b b:Quotient of n/64	Note.2(margin) 174+46b b:Quotient of n/64	173+46b b:Quotient of n/64	

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Copy destination head I/O			○				○						
s	Copy source head I/O	○	○	○		○	○	○	○				○	
n	Number of bits (words) to be copied					○	○	○	○				○	The constant is set in decimal.

Function

- The value of s (bit, word) is copied to the range d to d + n - 1.
- The value of s is retained.

A bit is copied to bits and a word is copied to words.



If n is a word: The contents (0 to 255) of the lower 8 bits (b7 to b0) of n (WX, WY, WR, WL, WM, TC) are set to the number of bits (words) to be copied.

If n is a constant: 0 to 255 (decimal) can be designated for the number of bits (words) to be copied.

Cautionary notes

- Use this command so that d + n - 1 does not exceed the I/O range *. If the I/O range is exceeded, DER is equal to "1" and the transfer is performed to the maximum range.
  * For I/O ranges, refer to the P3-6 and P3-7 performance specification table.
- If n is equal to 0, the block copy will not occur and DER (R7F4) will be "0".

Note.1 : 45.48+63.0a(a:Quotient of n/32)+1.6b(b: Remainder of n/32) d,s is at the time of R. Without index specification.
46.75+2.14a(a: Quotient of n/16) d,s is at the times other than R. Without index specification.
47.48+63.0a(a:Quotient of n/32)+1.6b(b: Remainder of n/32) d,s is at the time of R. With index specification.
34.85+2.14a(a: Quotient of n/16) d,s is at the times other than R. With index specification.
} Copy source : Constant

65.98+61.0a(a:Quotient of n/32)+1.6b(b: Remainder of n/32) d,s is at the time of R. Without index specification.
67.28+2.20a(a: Quotient of n/16) d,s is at the times other than R. Without index specification.
69.98+61.0a(a:Quotient of n/32)+1.6b(b: Remainder of n/32) d,s is at the time of R. With index specification.
70.28+2.20a(a: Quotient of n/16)+3b d,s is at the times other than R. With index specification.
} Copy source : I/O

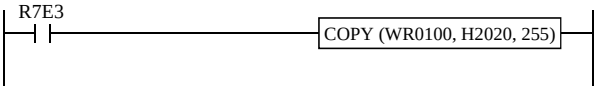
Note.2 : 37.68+1080a(a:Quotient of n/2) Without index specification.
39.68+0.80a(a: Quotient of n/2) With index specification.
} Copy source : Constant

56.24+0.81a(a:Quotient of n/2) Without index specification.
60.24+0.81a(a: Quotient of n/2) With index specification.
} Copy source : I/O

* Please refer to FUN 120-123 about index specification.

Program example

The default value (H2020) is set in the range of WR0100 to WR01FE.

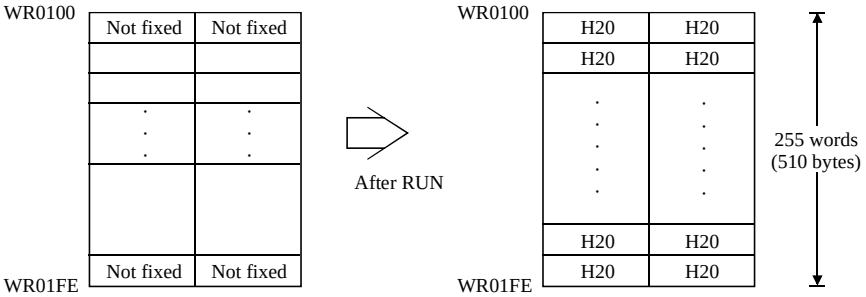



```
LD   R7E3
[
COPY (WR0100, H2020, 255)
]
```

Program description

WR0100 to WR01FE is considered as the communication data area and is filled with space codes (H20) as the default value during the first scan after RUN commencement.

R7E3: The first scan ON after RUN



Item number	Application commands-18	Name	Block exchange (EXCHANGE)

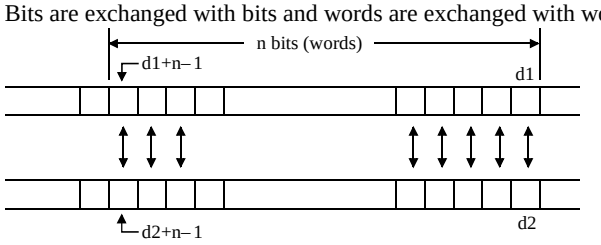
Ladder format	Condition code					Processing time (μs)			Remark
XCG (d1, d2, n)	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**	EH-CPU***A EH-CPU3**	Other than left	Upper case: B Lower case: W
	DER	ERR	SD	V	C				
	↕	●	●	●	●	Ave	Ave	Ave	

Command format	Number of steps				
XCG (d1, d2, n)	Condition	Steps	44.1+2.5n	42.2+2.53n 89.9+3.1n	76.0+3.1n
	—	4	40.1+1.4n	39.9+1.4n 123.4+1.9n	116+1.9n

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d1	Block exchange (EXCHANGE)			○				○						
d2	Exchange source head I/O			○				○						
n	Number of bits (words) to be exchanged					○	○	○	○				○	The constant is set in decimal.

Function

- Exchanges the contents of the n bits from d1 to d1 + n - 1 and the contents between d2 and d2 + n - 1.
- Bits are exchanged with bits and words are exchanged with words.



If n is a word: The contents (0 to 255) of the lower 8 bits (b7 to b0) of n (WX, WY, WR, WL, WM, TC) are set to the number of bits (words) to be exchanged.

If n is a constant: 0 to 255 (decimal) can be designated for the number of bits (words) to be exchanged.

Cautionary notes

- Use this command so that d1 + n – 1 and d2 + n - 1 do not exceed the I/O range *. If the I/O range is exceeded, DER is equal to "1" and the exchange is performed up to the maximum range with respect to the smaller number of bits (words) specified in d1 and d2.

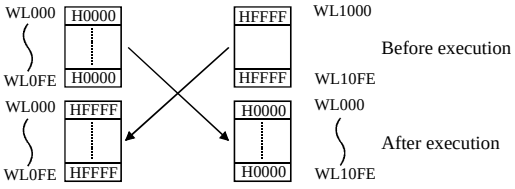
  * For I/O ranges, refer to the P3-6 and P3-7 performance specification table.
- If n is equal to 0, the block exchange is not performed and DER (R7F4) will be "0".

Program example


```
LD   X00001
AND DIF1
[
XCG (WL000, WL1000, 255)
]
```

Program description

- When X00001 rises, the contents of WL000 to WL0FE are exchanged with the contents of WL1000 to WL10FE.



Item number	Application commands-19	Name	NOT

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
NOT (d)	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	Upper case: W Lower case: DW
Command format	Number of steps					11	←	10 / 45	←	80	←	
NOT (d)	Condition			Steps								
	—			2		13	←	12 / 61	←	61	←	

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	I/O to be reversed		○	○			○	○			○	○		

Function

- Reverses the contents of d.

Before execution

1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0

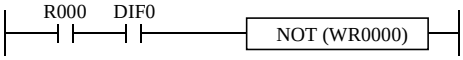
After execution

0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1

Cautionary notes

- Use edge trigger as the startup condition for this command.

Program example




```
LD    R000
AND   DIF0
[
NOT   WR0000
]
```

Program description

- When R000 rises, the content of WR0000 is reversed.
  Example) If WR0000 is H1234, WR0000 = HEDCB after the command is executed;
  WR0000 = H1234 when executed again

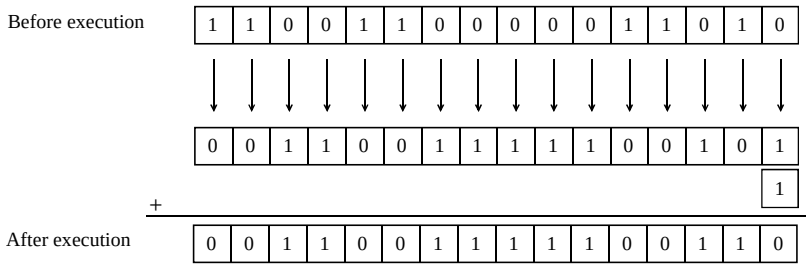
Item number	Application commands-20	Name	Two's complement (NEGATE)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
NEG (d)	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					12	←	11 / 45	←	80	←	Upper case: W Lower case: DW
	Condition			Steps								
NEG (d)	—			2		14	←	13 / 62	←	63	←	

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	I/O to take complement						○	○			○	○		

Function

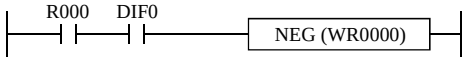
- Calculates two's complements of d (Reverses each bit contained in d and adds 1. However, C (R7F0) remains unchanged).



Cautionary notes

- Use edge trigger as the startup condition for this command.

Program example




```
LD   R000
AND  DIF0
[
NEG  WR0000
]
```

Program description

- When R000 rises, 2's complement of the content of WR0000 is obtained.
  Example) If WR0000 is H1234, WR0000 = HEDCC after the command is executed;
  WR0000 = H1234 when executed again

Item number	Application commands-21	Name	Absolute value

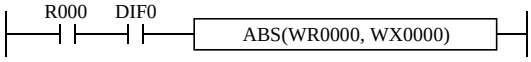
Ladder format	Condition code					Processing time (μs)						Remark
ABS (d, s)	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		Upper case: W Lower case: DW
	DER	ERR	SD	V	C							
	●	●	●	●	↕	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					13	←	15	←	85	←	
ABS (d, s)	Condition			Steps				50				
	Word			3		16	←	21	←	71	←	
	Double word			4				70				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	I/O after absolute value is taken						○	○			○	○		
s	I/O before absolute value is taken					○	○	○	○	○	○	○	○	

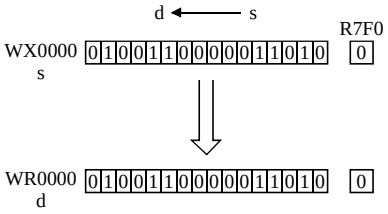
Function

- Given s is signed, set the absolute value of s in d.
- If s is positive or 0: The content of s is set to d. C (R7F0) is set to “0”.
- If s is negative: Two’s complements of the contents of s are set in d. C (R7F0) is set to “1”.
- Perform with d and s as both words or both double words.

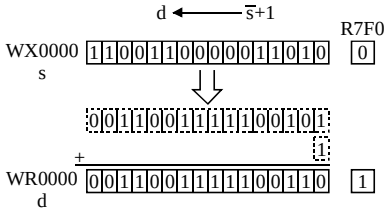
Example)



(When the value of WX is positive or 0)
WX0000 = H4C1A



(When the value of WX is negative)
WX0000 = HCC1A



- When s is a word: 0 to 32,767 (decimal) correspond to H0000 to H7FFF (hexadecimal).
  –32,768 to –1(decimal) correspond to H8000 to HFFFF (hexadecimal).
- When s is a double word: 0 to 2,147,483,647 (decimal) correspond to H00000000 to H7FFFFFFF (hexadecimal).
  –2,147,483,648 to –1 (decimal) correspond to H80000000 to HFFFFFFFF (hexadecimal).

Cautionary notes

- Use edge trigger as the startup condition for this command.

Item number	Application commands-22	Name	Sign addition (SIGN GET)

Ladder format	Condition code					Processing time (μs)						Remark
SGET (d, s)	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		Upper case: W Lower case: DW
	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					13	←	15 49	←	85	←	
SGET (d, s)	Condition			Steps								
	Word			3		16	←	20 69	←	70	←	
	Double word			4								

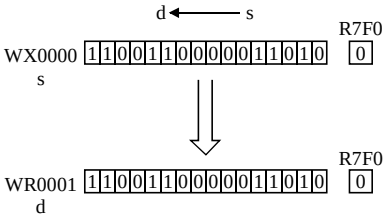
Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	I/O after absolute value is taken						○	○			○	○		
s	I/O before absolute value is taken					○	○	○	○	○	○	○	○	

Function

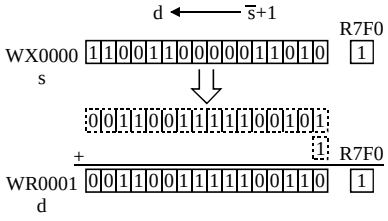
- If C (R7F0) is “0”: The content of s is set in d.
- If C (R7F0) is “1”: Two’s complements of the contents of s are set in d.
- The content of C (R7F0) remains unchanged.
- Perform with d and s as both words or both double words.

Example)

When C (R7F0) is "0"



When C (R7F0) is "1"



Cautionary notes

- Use edge trigger as the startup condition for this command.

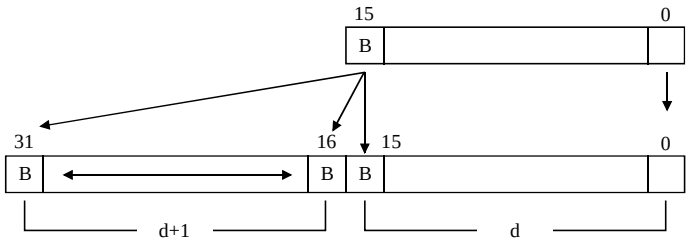
Item number	Application commands-23	Name	Sign expansion (EXTEND)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
EXT (d, s)	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					15	←	17 66	←	66	←	
	Condition			Steps								
EXT (d, s)	—			3								

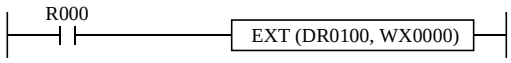
Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	I/O after sign expansion										○	○		
s	I/O before sign expansion					○	○	○	○				○	

Function

- The sign bit (MSB) of s is extended to the upper word of d.
- The lower word of d is set to the contents of s.



Program example




```
LD   R000
[
EXT (DR0100, WX0000)
]
```

Program description

- When R000 is turned on, the content of WX0000 is extended to DR0100.

When WX0000 is positive or 0
Example) WX0000 = H7FFF (+32767)
↓
DR0100 = H00007FFF (+32767)

When WX0000 is negative
Example) WX0000 = H8000 (– 32768)
↓
DR0100 = HFFFF8000 (– 32768)

Item number	Application commands-24	Name	Binary → BCD conversion

Ladder format	Condition code					Processing time (μs)						Remark
BCD (d, s)	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		Upper case: W Lower case: DW
	DER	ERR	SD	V	C							
	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

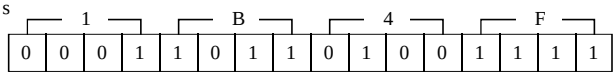
Command format	Number of steps									
BCD (d, s)	Condition	Steps	28	←	30 / 66	←	139	←		
	Word	3	31	←	35 / 86	←	191	←		
	Double word	4								

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	I/O after conversion (BCD)						○	○			○	○		
s	I/O before conversion (BIN)					○	○	○	○	○	○	○	○	

Function

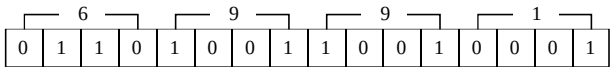
- The result of the content conversion of s from binary to BCD is output to d.
- If the conversion result of s exceeds the number of BCD data digits in d, DER (R7F4) is set to "1" and the command will not be executed.
- If s is a word: set s so that H0000 ≦ s ≦ H270F (0 to 9999).
- If s is a double word: set s so that H00000000 ≦ s ≦ H5F5E0FF (0 to 99999999).

Before execution s



(Binary)
H1B4F=6991

After execution d



(BCD)

Combinations of d and s.

d	s
Word	Word
Double word	Double word

Cautionary notes

- If data is error, the previous contents of d are retained.

Program example


```
LD  X00000
[
BCD (WY0010, WL000)
]
```

Program description

- When X00000 turns on, the content of WL000 is converted from binary to BCD and output to WY0010.
  WL000 H1B4F
  After conversion
  WY0010 H6991

Item number	Application commands-25	Name	BCD → Binary conversion

Ladder format	Condition code					Processing time (μs)						Remark
BIN (d, s)	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		Upper case: W Lower case: DW
	DER	ERR	SD	V	C							
	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					19	←	21 ... 57	←	92	93	
BIN (d, s)	Condition			Steps								
	Word			3		27	←	32 ... 83	←	83	201	
	Double word			4								

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	I/O after conversion (BIN)						○	○			○	○		
s	I/O before conversion (BCD)					○	○	○	○	○	○	○	○	

Function

- The result of the content conversion of s from BCD to binary is output to d.
- If the contents of s is not BCD data (if an A through F is included in the data), DER (R7F4) is set to "1" and the conversion will not be executed (d remains unchanged).

Before execution s

6				9				9				1			
0	1	1	0	1	0	0	1	1	0	0	1	0	0	0	1

(BCD)

After execution d

1				B				4				F			
0	0	0	1	1	0	1	1	0	1	0	0	1	1	1	1

(Binary)

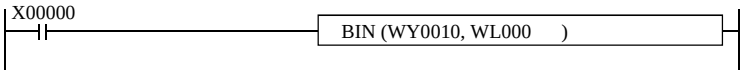
Combinations of d and s.

d	s
Word	Word
Double word	Double word

Cautionary notes

- If data is error, the previous contents of d are retained.

Program example




```
LD   X00000
[
BIN (WY0010, WL000)
]
```

Program description

- When X00000 turns on, the content of WL000 is converted from BCD to binary and output.
  WL000 H6691
  After conversion
  WY0010 H1B4F

Item number	Application commands-26	Name	Decode

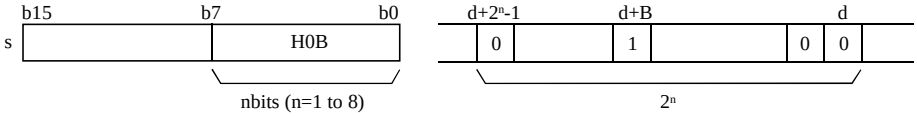
Ladder format	Condition code					Processing time (μs)			Remark
DECO (d, s, n)	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**	EH-CPU***A EH-CPU3**	Other than left	
	DER	ERR	SD	V	C				
	↕	●	●	●	●	Ave	Ave	Ave	

Command format	Number of steps				
DECO (d, s, n)	Condition	Steps	$35+1.35\times2^n$	$36+1.4\times2^n$ ⋯ $77+1.56\times2^n$	$62+1.65\times2^n$
	—	4			

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Decode destination head I/O			○										
s	Word I/O to be decoded					○	○	○	○				○	
n	Number of bits to be decoded												○	1 to 8 (decimal)

Function

- Decodes the lower n bits of s to 2^n and outputs "1" to the decoded bits in the bit rows between d and $d + 2^n - 1$ (where n = 1 to 8).
- If n is 0, the command will not be executed, and the contents of d to $d + 2^n - 1$ remain unchanged.



Cautionary notes

- Use this command so that $d + 2^n - 1$ does not exceed the I/O range*. If the I/O range is exceeded, DER is equal to "1" and the decoding is performed at the maximum range starting from d.
  * For I/O ranges, refer to the P3-6 and P3-7 performance specification table.
- Use 1 to 8 for n.

Program example


```
LD    R100
AND   DIF1
[
DECO (R000, WX0000, 4)
]
```

Program description

- When WX0000 = HFFFF, R00F, which is the 15th bit from R000 among the bits indicated by the lower four bit values of WX0000, is set to "1" upon rising of R100.

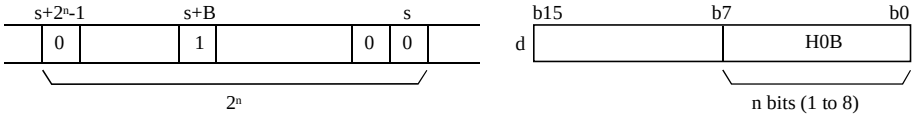
Item number	Application commands-27	Name	Encode

Ladder format	Condition code					Processing time (μs)			Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**	EH-CPU***A EH-CPU3**	Other than left	Upper case: Bit position of less than higher 16 bits Lower case: Other than the above
ENCO (d, s, n)	DER	ERR	SD	V	C				
	↕	●	●	●	↕	Ave	Ave	Ave	
Command format	Number of steps					$33+1.4\times2^n$	$39.9+1.4\times2^n$ $84+1.6\times2^n$	$84+1.6\times2^n$	
	Condition			Steps					
ENCO (d, s, n)	—			4		$84+0.1\times2^n$	$67.9+0.2\times2^n$ $170+0.2\times2^n$	$170+0.2\times2^n$	

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Decode destination head I/O						○	○						
s	Word I/O to be encoded			○										
n	Number of bits to be encoded												○	1 to 8 (decimal)

Function

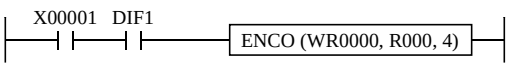
- Encodes the bit location 2^n in the range between s and $s + 2^n - 1$ where the bit is 1, and outputs the result to d (n = 1 to 8). Upper bits (16-n) of d are set to "0."
- If n is 0, the command will not be executed and the contents of d retain the original values.
- If there are more than one bits that are set to 1 between s and $s + 2^n - 1$, the upper bit location will be encoded.
- If all the bits from s to $s + 2^n - 1$ are "0", "0" is output to d, and C (R7F0) is equal to "1." In other cases, C (R7F0) is set to "0."



Cautionary notes

- Use this command so that $s + 2^n - 1$ does not exceed the I/O range*. If the I/O range is exceeded, DER is set to "1" and the encoding is performed at the maximum range starting from s.
  * For I/O ranges, refer to the P3-6 and P3-7 performance specification table.
- Use 1 to 8 for n.

Program example




```
LD    X00001
AND   DIF1
[
ENCO (WR0000, R000, 4)
]
```

Program description

- Upon the rising of X00001, the most significant bit that is set to "1" is detected within the row of bits R000 to R00F ($2^4 - 1 = 15$ bits), and a four-bit binary number is set in the word I/O of d
  Example) Of R000 to R00F, if "1" is set in the 7th and 6th bits, WR0000 is set to H0007.

Item number	Application commands-28	Name	7 segment decode

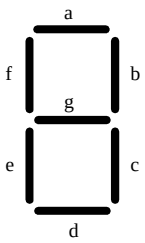
Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
SEG (d, s)	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps									
	Condition	Steps	16	←	68 / 68	←	94	98		
SEG (d, s)	—	3								

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Decode destination head I/O										○	○		
s	Decode contents					○	○	○	○				○	

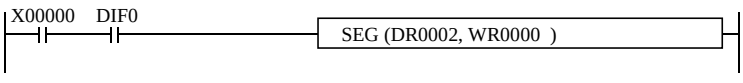
Function

- The result of the content conversion of s as 1-digit 4-bit data into 4-digit 7-segment display code is output to d.



Input data (4 bits)	Output data								Display
		g	f	e	d	c	b	a	
0	0	0	1	1	1	1	1	1	0
1	0	0	0	0	0	1	1	0	1
2	0	1	0	1	1	0	1	1	2
3	0	1	0	0	1	1	1	1	3
4	0	1	1	0	0	1	1	0	4
5	0	1	1	0	1	1	0	1	5
6	0	1	1	1	1	1	0	1	6
7	0	0	1	0	0	1	1	1	7
8	0	1	1	1	1	1	1	1	8
9	0	1	1	0	1	1	1	1	9
A	0	1	1	1	0	1	1	1	A
B	0	1	1	1	1	1	0	0	B
C	0	0	1	1	1	0	0	1	C
D	0	1	0	1	1	1	1	0	D
E	0	1	1	1	1	0	0	1	E
F	0	1	1	1	0	0	0	1	F

Program example




```
LD   X00000
AND DIF0
[
SEG  (DR0002, WR0000)
]
```

Program description

- Upon the rising of X00000, the content of WR0000 is converted to an 8-bit 4-digit 7 segment LED display data. (The most significant bit of the eight bits in each digit is always “0.”)

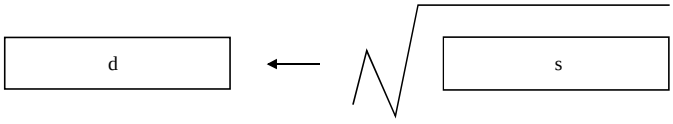
Item number	Application commands-29	Name	Square root

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
SQR (d, s)	DER	ERR	SD	V	C							
	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					86	←	90 143	←	107	108	
SQR (d, s)	Condition			Steps								
	—			4								

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Square Root (BCD)						○	○						
s	Take square root (BCD)									○	○	○	○	

Function

- The square root of the contents of s is calculated and output to d.
- Set BCD data to s.
- If s has a BCD data error, the DER (R7F4) is set to "1" and calculation is not performed (data other than H0 to H9 exists).
- The fractional portion is rounded down.



Program example


```
LD   X00000
AND DIF0
[
SQR  (WR0001, DR0020)
]
```

Program description

- Upon the rising of X00000, the square root of the value in DR0020 is calculated and is substituted into WR0001.
  Example) In the case of DR0020 = H00002159 (BCD), it becomes WR0001=H0046 (BCD) after the operation.
  ($\sqrt{2159}$ = 46.465 · · · · ·)

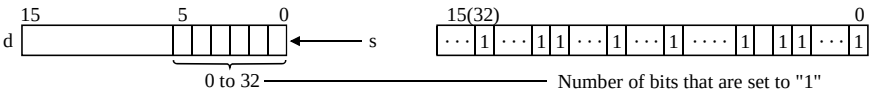
Item number	Application commands-30	Name	Bit count

Ladder format	Condition code					Processing time (μs)						Remark
BCU (d, s)	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		Upper case: W Lower case: DW
	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					14	←	16	←	96	213	
BCU (d, s)	Condition			Steps				51				
	Word			3		17	←	21	←	99	216	
	Double word			4				56				

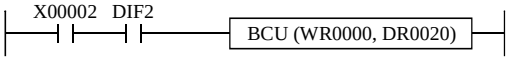
Usable I/O		Bit				Word				Double word				Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM	Constant	
d	Number of bits set to 1						○	○						
s	I/O that counts the bits set to 1					○	○	○	○	○	○	○	○	

Function

- Of the contents of s (16 bits for word and 32 bits for double word), the number of bits that are set to 1 are output to d (0 to 32).



Program example

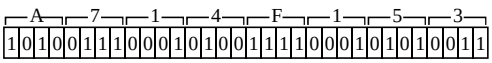



```
LD   X00002
AND  DIF2
[
BCU  (WR0000, DR0020)
]
```

Program description

- At the rising edge of X00002, the number of bits that are set to “1” among the data input to DR0020 is counted, and set to WR0000.
  Example)

  In the case of



  the number of bits set to "1" is 16 (decimal).
  Therefore, the result is WR0000 = H0010.

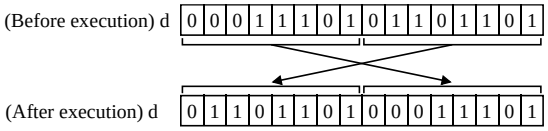
Item number	Application commands-31	Name	Swap

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
SWAP (d)	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					12	←	11	←	81	198	
	Condition			Steps								
SWAP (d)	—			2				45	←			

Usable I/O	Bit				Word				Double word			Constant	Other
	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d I/O to be exchanged						○	○						

Function

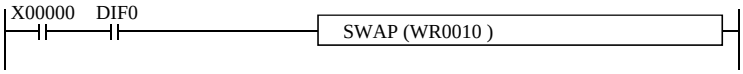
- Switches the upper 8 bits and lower 8 bits contained in d.



Cautionary notes

- Use edge trigger as the startup condition for this command.

Program example




```
LD    X00000
AND   DIF0
[
SWAP (WR0010)
]
```

Program description

- Upon rising of X00000, the upper 8 bits and lower 8 bits of WR0010 are swapped and stored in WR0010.

  WR0010 H1234 Before execution
  WR0010 H3412 After execution

Note) When there is no rising edge of DIF0, it is executed during each scan, so the upper and lower bits of WR0010 are swapped in each scan.

Item number	Application commands-32	Name	FIFO initialization

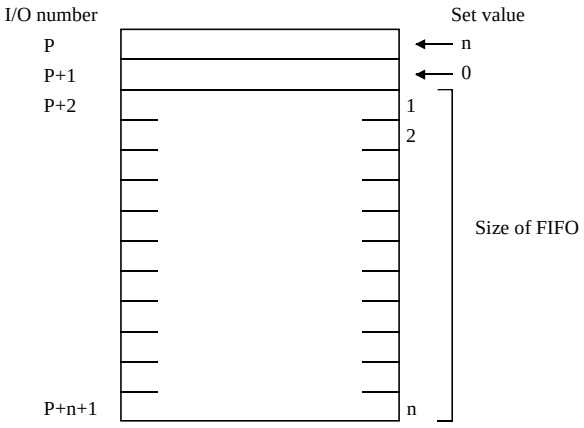
Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FIFIT (P, n)	DER	ERR	SD	V	C							
	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps									
	Condition	Steps	24	←	24	←	72	189		
FIFIT (P, n)	—	3			66	←				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
P	FIFO head I/O							○						
n	Size of FIFO												○	0 to 256

Function

- FIFO is an abbreviation for First In First Out, meaning that the data stored in the buffer first is taken out first. This command initializes the FIFO.
- Sets the FIFO head I/O number “P” and the FIFO size “n.”
  If 0 ≦ n ≦ 256: Sets n in P.
  If 257 ≦ n: Sets 256 in P.
- Sets the initial setting value of 0 as the number of used FIFO to P + 1.
- The FIFO sets n + 2 words from P to P + n + 1.



Cautionary notes

- Use this command so that P + n + 1 does not exceed the I/O range*. If the I/O range is exceeded, DER is equal to”1” and the maximum value of the range (last value - (P - 1)) is set in P.
  * For I/O ranges, refer to the P3-6 and P3-7 performance specification table.
- Set n in the range between 0 and 256. If n is greater than 256, DER (R7F4) is equal to”1” and n is set to 256.

Item number	Application commands-33	Name	FIFO write

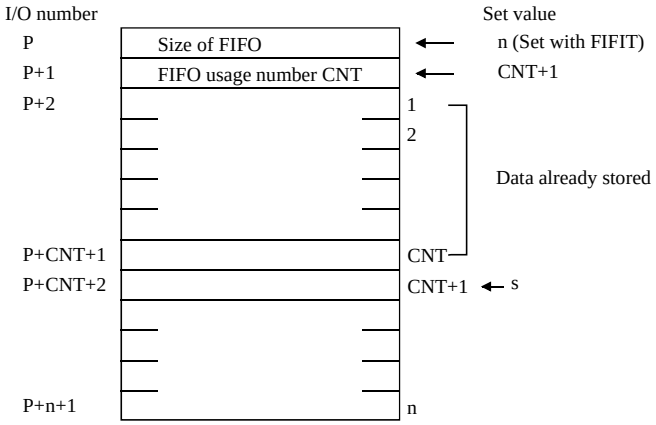
Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FIFWR (P, s)	DER	ERR	SD	V	C							
	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps									
	Condition	Steps	25	←	23	←	72	189		
FIFWR (P, s)	—	3			67	←				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
P	Head I/O of FIFO							○						
s	Data to be written to FIFO					○	○	○	○				○	

Function

- Writes data to the FIFO buffer whose head I/O number is P.
  If the used number CNT is less than the size of n:
  Writes the contents of s to P + CNT + 2, and adds 1 to the used number CNT.
  If the used number CNT is greater than the size of n:
  DER (R7F4) is set to "1" and write will not be performed.



Cautionary notes

- Use this command so that P + n + 1 does not exceed the I/O range *. If the I/O range is exceeded, DER is equal to "1" and write will not be performed.

  * For I/O ranges, refer to the P3-6 and P3-7 performance specification table.

Item number	Application commands-34	Name	FIFO read

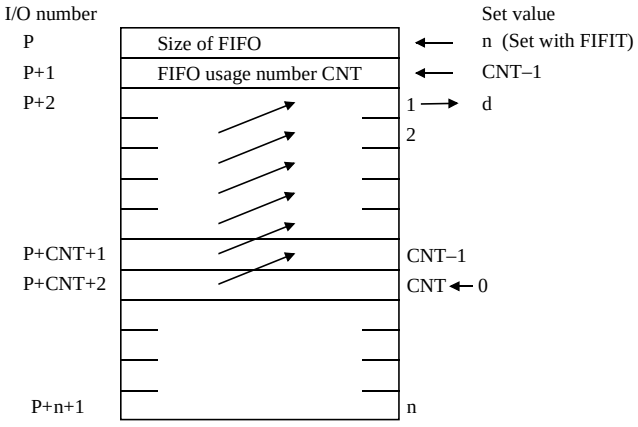
Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FIFRD (P, d)	DER	ERR	SD	V	C							
	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps									
	Condition	Steps	157	285	157	285	141	258		
FIFRD (P, d)	—	3			232	360				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
P	FIFO head I/O							○						
d	I/O that stores the read data						○	○	○				○	

Function

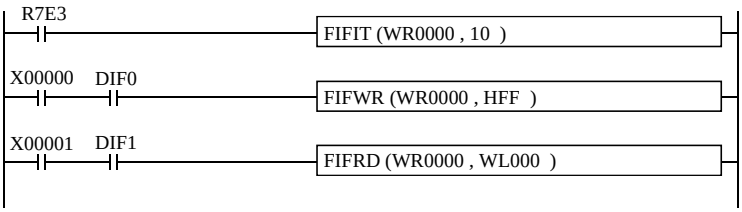
- Reads the data in the FIFO buffer whose head I/O number is P.
  When 1 ≤ used number CNT ≤ size of n:
  The contents of P + 2 is read to d.
  Moves each of the contents of P + 3 to P + CNT + 2 to its previous I/O.
  Writes 0 to P + CNT + 2.
  Subtracts 1 from the content of CNT.
  When CNT is greater than size of n or CNT is 0:
  DER (R7F4) is set to "1" and read will not be performed.



Cautionary notes

- Use this command so that p + n + 1 does not exceed the I/O range *. If the I/O range is exceeded, DER is equal to "1" and read is not performed.
  * For I/O ranges, refer to the P3-6 and P3-7 performance specification table.

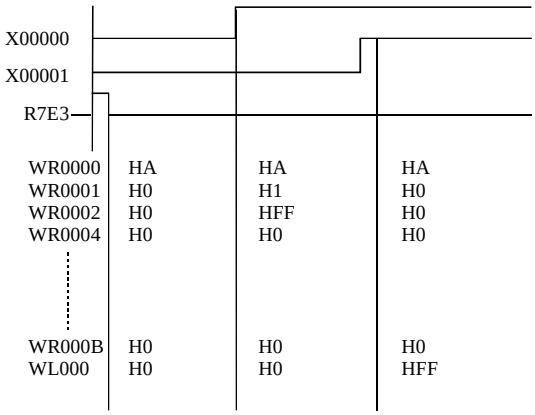
Program example




```
LD      R7E3
[
FIFIT   (WR0000, 10)
]
LD      X00000
AND     DIF0
[
FIFWR (WR0000, HFF)
]
LD      X00001
AND     DIF1
[
FIFRD  (WR0000, WL000)
]
```

Program description

- The FIFO buffer is set at WR0002 through WR000B during the first scan after RUN execution
- HFF is stored when X00000 rises.
- HFF is read to WL000 when X00001 rises.



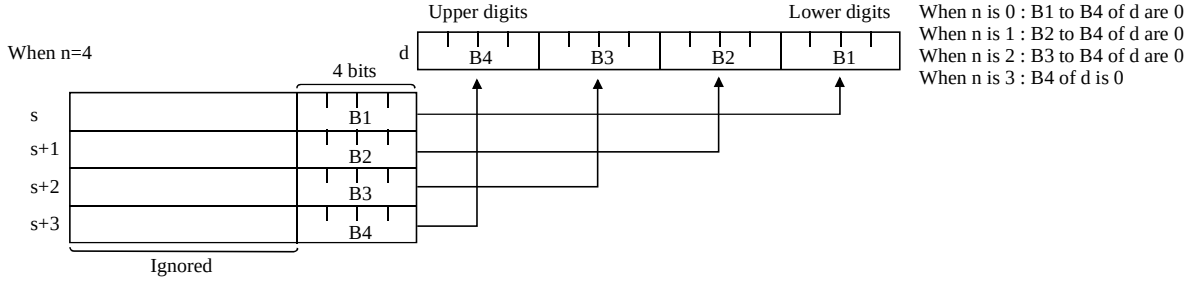
Item number	Application commands-35	Name	Unit

Ladder format	Condition code					Processing time (μs)						Remark
UNIT (d, s, n)	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
	DER	ERR	SD	V	C							
	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					30	←	32 107	←	149	267	
UNIT (d, s, n)	Condition			Steps								
	—			4								

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Unity result write destination I/O						○	○						
s	Unity destination head I/O							○						
n	Numbers of words to be united												○	n=0 to 4

Function

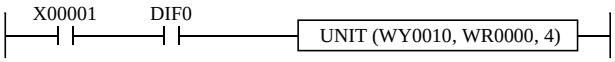
- Sets the values in the lower 4 bits of each of the n (1 to 4) words starting from s to the lower 4 bits of each word in d.
- If n is 1 to 3, the bits not set in d will be 0.
- The data stored in s to s + n – 1 will be retained even if UNIT is executed.
  Use this command so that s + n - 1 does not exceed the I/O range *. If the I/O range is exceeded, DER is equal to “1” and the lower 4 bits within the range between s and I/O will be set in d.
  * For I/O ranges, refer to the P3-6 and P3-7 performance specification table.



Cautionary notes

- When n=0, DER = “0” and 0 will be set in the write destination I/O.
- When n>5, nothing is executed.

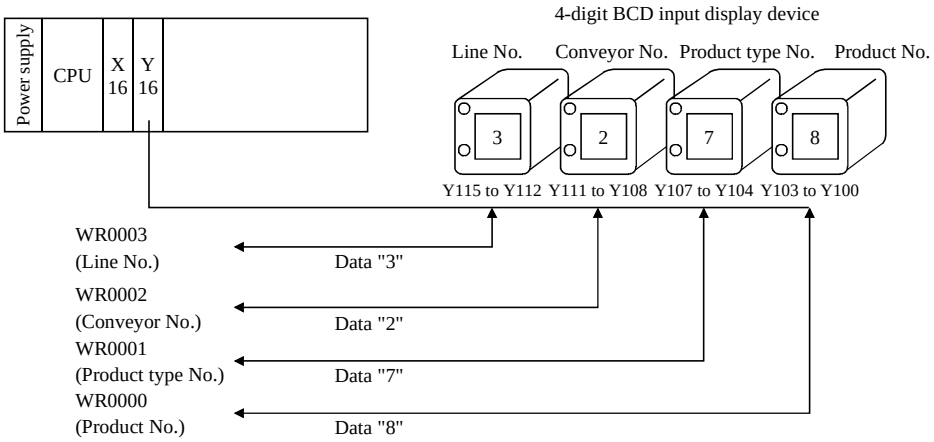
Program example




```
LD    X00001
AND  DIF0
[
UNIT (WY0010, WR0000, 4)
]
```

Program description

A 4-digit BCD input display device is connected to the WY0010, and each digit displays WR0000 to WR0003 data independently. (Only the lower four bits are considered valid data for WR0000 to WR0003.)



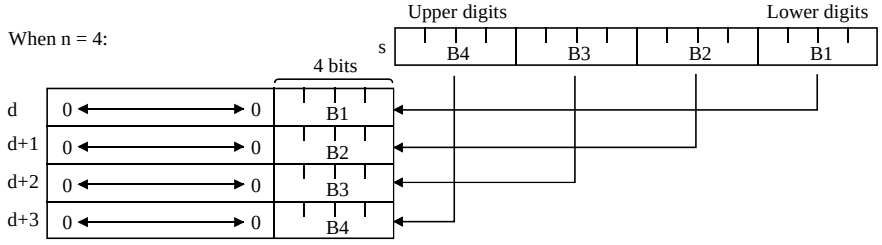
Item number	Application commands-36	Name	Distribute

Ladder format	Condition code					Processing time (μs)						Remark
DIST (d, s, n)	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
	DER	ERR	SD	V	C							
	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					27	←	29 / 71	←	80	198	
DIST (d, s, n)	Condition			Steps								
	—			4								

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Distribution result write destination head I/O							○						
s	I/O to be distributed					○	○	○	○				○	
n	Number of words to be distributed												○	n=0 to 4

Function

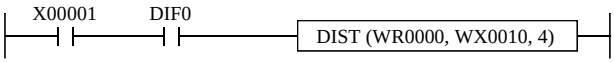
- Distributes s into 4 bit sections and sets to the lower 4 bits of the n words starting from d.
- The upper 12 bits of the range d to d + n – 1 will be 0.
- The value of s will be retained even if DIST is executed.
  Use this command so that s + n - 1 does not exceed the I/O range *. If the I/O range is exceeded, DER is equal to “1” and the distribution data for s will be set in the lower 4 bits within the range between d and the I/O.
  * For I/O ranges, refer to the P3-6 and P3-7 performance specification table.



Cautionary notes

- When n=0, DER = “0” and nothing will be executed.

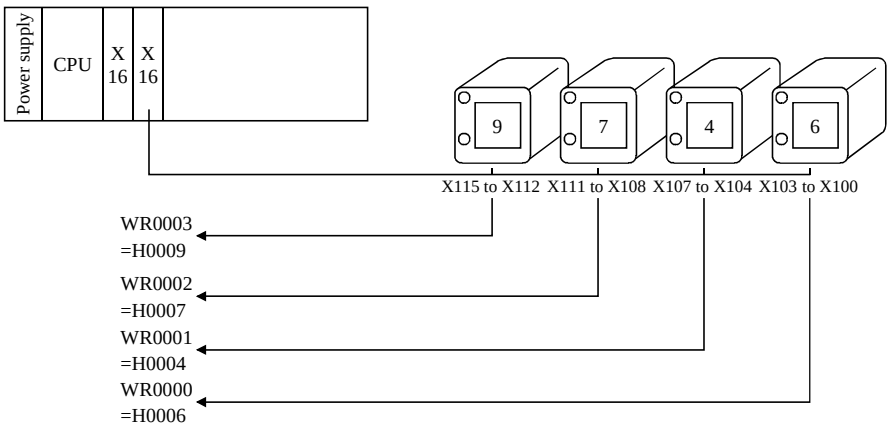
Program example




```
LD    X00001
AND  DIF0
[
DIST (WR0000, WX0010, 4)
]
```

Program description

A 4-bit 4-digit Digit switch is connected to the WX0010, and the data for each digit is stored in WR0000 to WR0003 as independent data.



Item number	Application commands-37	Name	I/O address conversion

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
ADRIO (d, s)	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	Upper case: B Lower case: W
Command format	Number of steps											
	Condition			Steps								
ADRIO (d, s)	—			3		12	←	15 49	←	84	202	

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Conversion address						○	○						
s	I/O to be converted	○*	○*	○		○*	○*	○						*The EH-CPU104A/ 208A/308A/316A/ 448 can not be used.

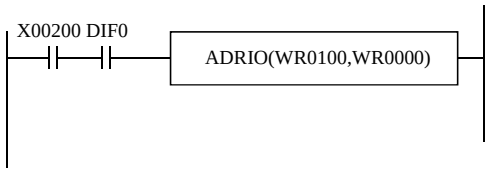
Function

- Obtains the actual address of the I/O designated by s, and sets the result in d.

Cautionary notes

- In the EH-104A/208a/308A/316A/448, ADRIO command External I/O(X,Y,WX,WY) .
  (Please see chapter 0 0.2.7 ADRIO Command, to)

Program example




```
LD    X00200
AND  DIF0
[
ADRIO  (WR0100, WR0000)
]
```

Program description

- Upon X00200 rise, the actual address of WR0000 (H3C00) is set in WR0100.
  After command execution, WR0100 becomes H3C00.

Item number	Control commands-1	Name	Normal scan end

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
END	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps										
	Condition	Steps	194	214	184	195	285	304			
END	—	1			311	320					

Usable I/O	Bit				Word				Double word			Constant	Other
	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		

Function

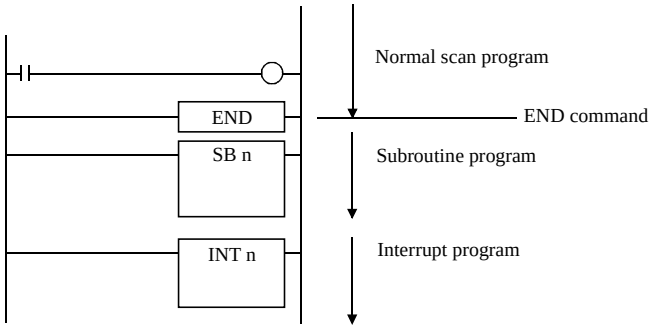
- Indicates the end of a normal scan program. (The execution of this command returns to the beginning of the program, and a normal scan is executed.)
- This command is not necessary when there are no subroutine programs or interrupt scan programs.
- If there is a subroutine program or interrupting program, write this command at the end of the normal scan program.
- This command is used only once in a program. Do not use any startup conditions.

Cautionary notes

- The END command is checked prior to the execution, and if there is an error, the following error codes are set in the special internal output WRF001. Also, the CPU error code "34" is set to special internal output WRF000.

CPU error code	Special internal output	Error code	Error description
34	WRF001	H0010	There is no END command.
		H0022	There are two or more END commands.
		H0032	A startup condition is used for the END command.

Instruction for use



Item number	Control commands-2	Name	Scan conditional end

Ladder format	Condition code					Processing time (μs)						Remark
CEND (s)	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		Upper case: Conditions do not meet Lower case: Conditions meet
	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					6	←	13	←	12	←	
CEND (s)	Condition		Steps			196	216	193	203	164	315	
	—		2					306	←			

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Scan end condition	○	○	○										

Function

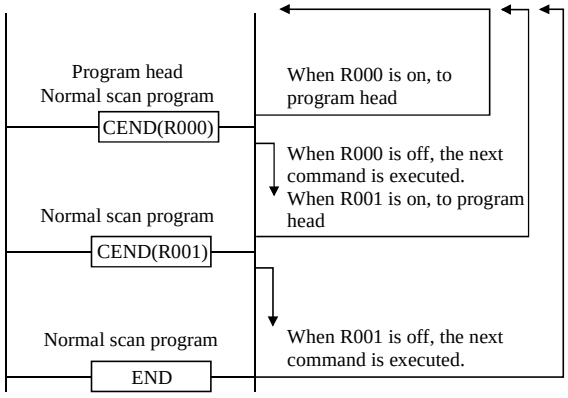
- If the scan end condition (s) is on, the execution of this command returns to the head of the scan program and executes the program.
- If (s) is off, the next command is executed.
- This command can only be used in normal scan programs, and can be used as many times as desired.
- This command can specify a startup condition. In this case, if the startup condition and (s) are both on, this command is executed.

Cautionary notes

- The CEND command is checked prior to the execution, and if there is an error, the following error codes are set in the special internal output WRF001. Also, the CPU error code “34” is set to special internal output WRF000.

CPU error code	Special internal output	Error code	Error description
34	WRF001	H0023	The CEND command exists after the END command.

Instruction for use



Item number	Control commands-3	Name	Unconditional jump (JUMP)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
JMP n	DER	ERR	SD	V	C							
	●	1]	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps									
	Condition	Steps	21	45	21	22	77	104		
JMP n	—	2			28	57				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
n	Code number												○	0 to 255 (Decimal)

Function

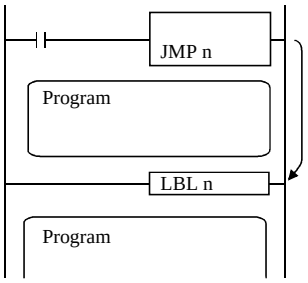
- If the startup condition of JMP n switches on, the control jumps the program from this command to the LBL n of the same code number. Always use JMP n and LBL n in pairs.
- If the startup condition is not met, the next command will be executed.
- If setting this command in conjunction with other commands in the same arithmetic-operation box, set this command at the end.
- The JMP n command is valid only within the same program. (A jump to a subroutine or interrupt scan cannot be performed from a normal scan, nor vice versa.)
- Nesting of JMP n commands is possible, but caution must be exercised so that an overload error does not occur.

Cautionary notes

- This command is checked prior to the execution, and if there is an error, the following error codes are set in the special internal outputs R7F3 and WRF015. In this case, jump is not performed and the next command will be executed.

Special internal output		Error code	Error description
R7F3=1	WRF015	H0015	There is no LBL n.
		H0040	A jump is attempted to a different program area.

Instruction for use



- When the startup condition turns on, it jumps to LBL n.
- If there is a timer within the program it jumped to, the progress value is updated, but since commands are not executed, output will not turn on even if the ON conditions are met.

Item number	Control commands-4	Name	Conditional jump

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
CJMP n (s)	DER	ERR	SD	V	C							Upper case: Conditions do not meet Lower case: Conditions meet
	●	1]	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					6	←			14	←	
	Condition			Steps				10	←			
CJMP n (s)	—			3		22	46	25 31	26 60	59	105	

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
n	Code number												○	0 to 255 (Decimal)
s	Jump condition	○	○	○										

Function

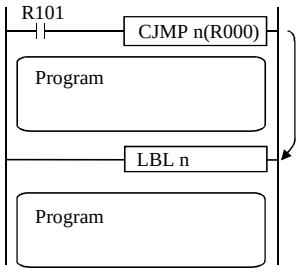
- If the jump condition (s) of CJMP n(s) switches on, the control jumps the program from this command to the LBL n of the same code number. Always use CJMP n(s) and LBL n in pairs.
- If the startup or jump condition is not met, the next command will be executed.
- If setting this command in conjunction with other commands in the same arithmetic-operation box, caution must be used because the jump takes place without performing the operations specified after the command.
- The CJMP n(s) command is valid only within the same program. (A jump to a subroutine or interrupt scan cannot be performed from a normal scan, nor vice versa.)
- Nesting of CJMP n(s) commands is possible, but caution must be exercised so that an overload error does not occur.

Cautionary notes

- This command is checked prior to the execution, and if there is an error, the following error codes are set in the special internal outputs R7F3 and WRF015. In this case, jump is not performed and the next command will be executed.

Special internal output		Error code	Error description
R7F3=1	WRF015	H0015	There is no LBL n.
		H0040	A jump is attempted to a different program area.

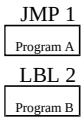
Instruction for use



- When the startup condition and the R000 jump condition bit I/O are both on, it jumps to LBL n.
- If there is a timer within the program it jumped to, the progress value is updated, but since commands are not executed, output will not turn on even if the ON conditions are met.

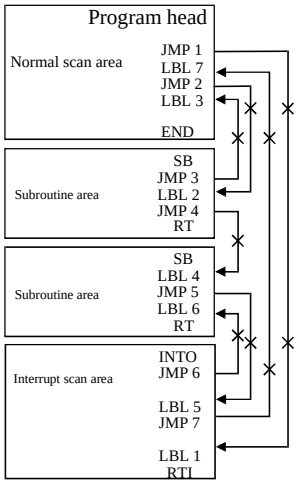
Syntax of JMP, CJMP

1) LBL n with the same code number as the code number n of the JMP command is required.



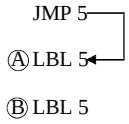
- If JMP 1 is executed when there is no LBL 1 an LBL undefined error JMP 1 will do nothing and execute the processing of program

2) Jump is not permitted to outside the area in which the JMP command resides.



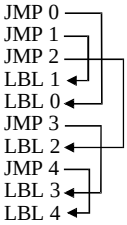
- When the JMP 1 command is executed, since LBL 1 is not in the normal scan area, a "jump outside the area" error will be generated. The JMP 1 command will do nothing and execute the next processing of program.
- JMP 2 to JMP 7 similar processing.

3) Code number n of the JMP command and the LBL n with the same code number may not be overlapped.

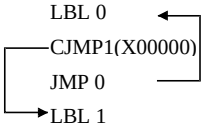


- In the pre-operation process, the label commands Ⓐ and Ⓑ have 5 as the code numbers, so a duplicate definition error will occur.

4) Nesting of JMP commands is allowed.

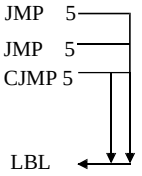


5) The JMP command can jump to a location before the command itself.

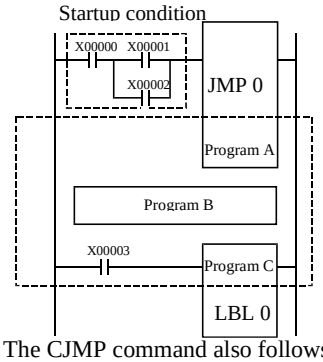


- JMP 0 will jump to LBL 0, which is a location before the JMP command.
- When input X00000 turns on, the loop between LBL 0 and JMP 0 is escaped by jumping from CJMP1 (X00000) to LBL 1.
- If there is no command as CJMP 1 (X00000) to escape from the loop, the loop from LBL 0 to JMP 0 will continue endlessly.

6) An overlap of JMP commands with the same code number is valid.



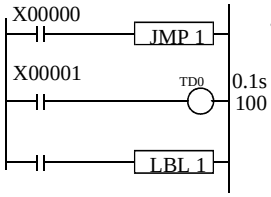
7) A startup condition can be programmed with respect to JMP commands.



- If a jump is performed fro JMP 0 to LBL 0, programs B and C will not be executed.

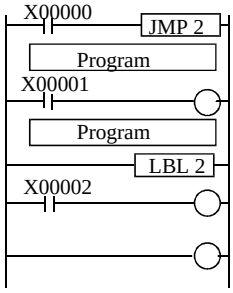
8) The CJMP command also follows the same syntax as 1) through 7).

Note 1: When a JMP command jumps to LBL, the status of each I/O between JMP and LBL is retained. However, the timer progress value will be updated.
In case of EH-104A/208A/308A/316A/448, when you use JMP command at the time of Timer starting, please keep in mind that the progress value is updateed at time of Timer command execution.
(Please see each Timer commands, to)



- If X00000 turns on after X00001 turns on, the progress value of TD0 will be updated even if a jump is performed from JMP 1 to LBL 1. If X00000 remains on, TD0 will not turn on even if its progress value exceeds 100.

Note 2: If the JMP command is used in conjunction with the MCS or MCR command, the following actions will result, so exercise caution when programming.



- When JMP 2 does not jump, Y00100 will turn on when X00001 and X00002 are both on.
- When JMP 2 does jump, if X00000 is on, Y00100 will follow the on/off of X00002 regardless of the on/off of X00001.

Note 3: Do not create a circuit that jumps to outside from between MCS and MCR.

Item number	Control commands-5	Name	Label

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
LBL n	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					0.05	←	0.65 ..... 0.65	←	1.3	←	
LBL n	Condition: —		Steps: 1									

Usable I/O	Bit				Word				Double word			Constant	Other
	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
n Code number												○	0 to 255 (Decimal)

Function

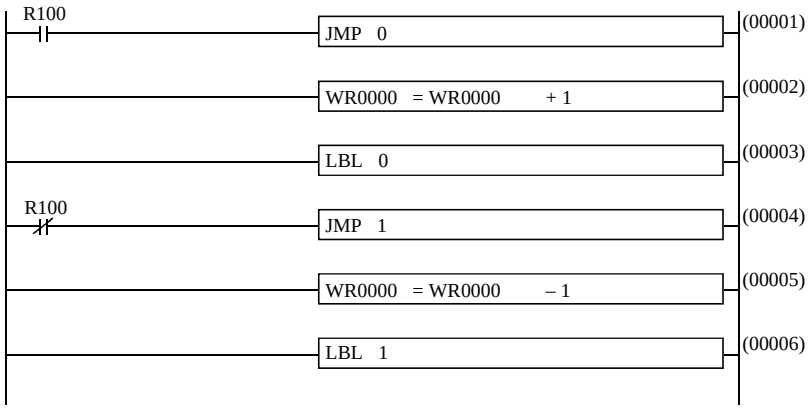
- This command indicates the destination of the jump when the JMP n or CJMP n command is executed (n is always used in pairs).
- The n in the LBL n cannot be used multiple times in the same program.
- This command alone does not do anything.
- Even if a startup condition is used for LBL n, it will be ignored.

Cautionary notes

- This command is checked prior to execution, and when there is an error, the following error code is set in the special internal output WRF001. Also, the CPU error code "34" is set to special internal output WRF000.

CPU error code	Special internal output	Error code	Error description
34	WRF001	H0001	Duplicate definition of LBL

Instruction for use



- When R100 is on, JMP 0 will be executed but JMP 1 will not be executed. Therefore, the content of WR0000 will decrement by one during each scan.
- When R100 is off, JMP 0 will not be executed but JMP 1 will be executed. Therefore, the content of WR0000 will increment by one during each scan.

Item number	Control commands-6	Name	FOR

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FOR n (s)	DER	ERR	SD	V	C							
	●	1]	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps									
	Condition	Steps	29	49	27	37	90	102		
FOR n (s)	—	3			84	102				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
n	Code number												○	0 to 49 (Decimal)
s	Number of times repeated						○	○						

Function

- Jumps from the NEXT n of the same code number to this command.
- If the number of times repeated (s) is greater than 0, the command following the FOR n (s) is executed.
- If the number of times repeated (s) is equal to 0, it jumps to the command following the NEXT n.
- Use FOR n (s) and NEXT n in pairs. Also, place the NEXT n after FOR n.
- The FOR n (s) may not be used multiple times.
- Use the FOR n (s) and NEXT n in the same program area. (It is not allowed to include FOR n (s) in the normal scan and NEXT n in the subroutine area.)
- The FOR n (s) to NEXT n nesting can be made up to 5 levels.

Cautionary notes

- This command is checked prior to execution, and when there is an error, the following error code is set in the special internal output WRF001. Also, the CPU error code "34" is set to special internal output WRF000.

CPU error code	Special internal output	Error code	Error description
34	WRF001	H0002	Duplicate definition of FOR

- If an error is generated during the execution of the command, an error code will be set in the special internal outputs R7F3 and WRF015, and the following program will be executed.

Special internal output		Error code	Error description
R7F3=1	WRF015	H0017	NEXT undefined
		H0043	FOR to NEXT error
		H0044	Area error for NEXT
		H0045	FOR to NEXT nesting error
		H0046	FOR nesting overflow

Instruction for use

For the command to use this, refer to NEXT n.

Item number	Control commands-7	Name	NEXT

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
NEXT n	DER	ERR	SD	V	C							
	●	1]	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					23	←	25 63	←	76	←	
	Condition			Steps								
NEXT n	—			2								

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
n	Code number												○	0 to 49 (Decimal)

Function

- Subtracts 1 from the number of times repeated (s) for the FORn(s) command of the same code number, then jumps to FORn(s).

Cautionary notes

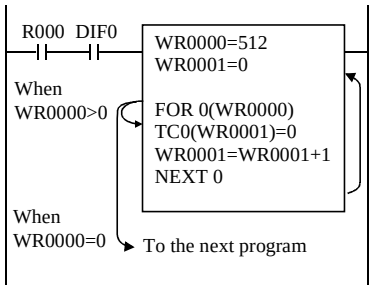
- This command is checked prior to execution, and when there is an error, the following error code is set in the special internal output WRF001. Also, the CPU error code "34" is set to special internal output WRF000.

CPU error code	Special internal output	Error code	Error description
34	WRF001	H0003	Duplicate definition of NEXT

- If an error is generated during the execution of the command, an error code will be set in the special internal outputs R7F3 and WRF015, and the following program will be executed.

Special internal output		Error code	Error description
R7F3=1	WRF015	H0016	FOR undefined
		H0046	FOR nesting overflow

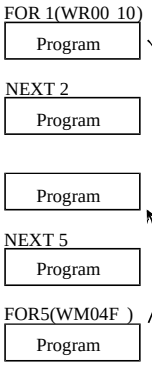
Instruction for use



- When R000 is turned on, the progress value (TC n) of the timer or counter is cleared with 0 for 512 points.
- Once the FOR to NEXT is begun, the command keeps executing until (s) is 0.
- FOR0 (WR0000) performs commands after TC0 (WR0001) = 0 while WR0000>0, subtracts 1 from WR0000 at NEXT0, then jumps to FOR0 (WR0000).
- FOR0 (WR0000) jumps to the next command within the current box upon WR0000 = 0.

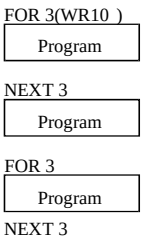
Syntax of FOR to NEXT

1] A NEXT command with the same code number as the code number n of the FOR command is required after the FOR command.



- NEXT undefined error
  The NEXT command with respect to the FOR command does not exist within the user program.
- FOR undefined error
  The FOR command does not exist before the NEXT command.
- NEXT to FOR error
  The NEXT command exists before the FOR command.

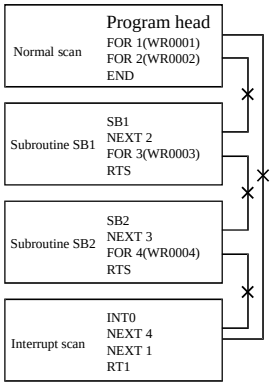
2] An overlap of FOR and NEXT commands with the same code number n is not allowed.



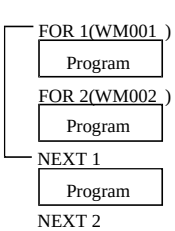
- FOR duplicate-definition error
  A FOR command with the same code number n is programmed.
- NEXT duplicate-definition error
  A NEXT command with the same code number n is programmed.

Note) FOR and NEXT duplicate-definition errors will occur during operation pre-processing.

3] FOR and NEXT must be within the same area.



4] Use FOR to NEXT as a nest.



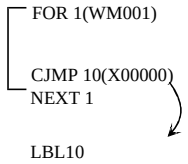
FOR 1(WM001) to NEXT 1 will execute normally.

- Nesting error

When WM002=0
Since FOR 1(WM001) to NEXT 1 is prioritized, jump will not be performed over NEXT 1 from FOR 2 to NEXT 2. At this time, NEXT 2 generates a FOR 2 undefined error.

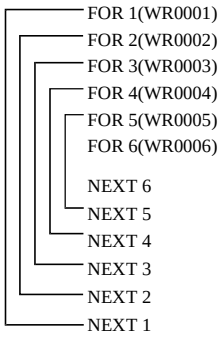
When WM002 ≠0
FOR 2 will not do anything. Therefore, NEXT 2 will generate a FOR 2 undefined error.

5] It is possible to escape from a FOR to NEXT loop using a jump command.



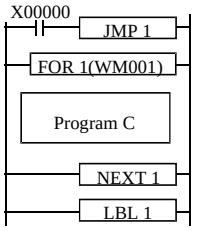
The FOR 1 to NEXT 1 loop is escaped when X00000 turns on before the loop has been repeated for the set number of repeats (content of WM001).

6] FOR to NEXT may be nested up to 5 levels. When a subroutine is included, the FOR to NEXT within the subroutine is counted.



- Nesting overflow error

7] Do not include a startup condition between FOR and NEXT. If a startup condition is required, create a circuit as shown below:

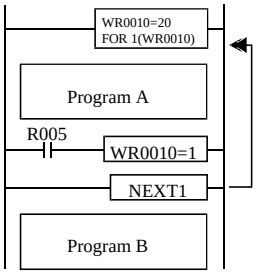


[Operation description]

When X00000 is off, program C is repeatedly executed for the number of WM 1 times.

When X00000 is on, program C is not executed since a jump is performed from JMP 1 to LBL 1.

8] The number of repeats may be modified within the program.



The content of WR0010 decrements by 1 and a jump is performed to FOR 1 (WR0010).

- When R005 is off
  Program B is executed after program A is repeated 20 times.
- When R005 is on
  The repeat counter WR0010 changes to 1, and since the NEXT 1 processing subtracts 1 from it, the content of WR0010 becomes 0. Therefore, the repeating of program A is terminated and program B is executed.

Item number	Control commands-8	Name	Call subroutine

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
CAL n	DER	ERR	SD	V	C							
	●	1]	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					19	←	20	←	43	←	
	Condition		Steps									
CAL n	—		2					23	←			

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
n	Code number												○	0 to 99 (Decimal)

Function

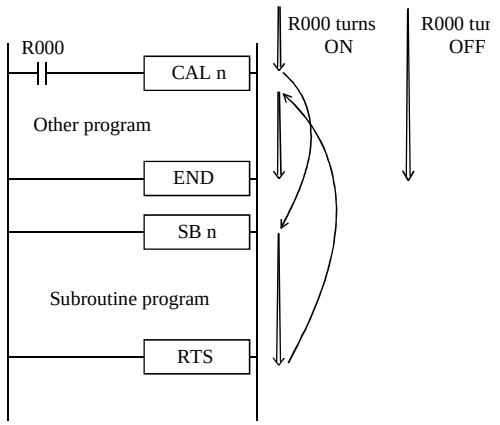
- If the startup condition of CAL n is on, this commands executes the subroutine program (the program sandwiched by SB n and RTS) of the same code number.
- If the startup condition is off, the next program is executed.
- Up to 5 levels of CAL (nesting) for another subroutine can be performed within a subroutine.
- It is possible to call a subroutine from within an interrupt scan program.

Cautionary notes

- If an error is generated during the execution of the command, an error code will be set in the special internal outputs R7F3 and WRF015, and the following program will be executed.

Special internal output		Error code	Error description
R7F3=1	WRF015	H0013	SB undefined
		H0041	Nesting error

Instruction for use



- When R000 is on, a subroutine program is executed by CAL n. After the execution, the program is re-executed from the code following the CAL n.
- When R000 is off, the subroutine program is not executed, and the next program is executed.

Item number	Control commands-9	Name	Start subroutine program

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
SB n	DER	ERR	SD	V	C							
	●	1]	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					0.05	←	0.65 ..... 0.65	←	1.3	←	
SB n	Condition: —		Steps: 1									

Usable I/O	Bit				Word				Double word			Constant	Other
	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
n Code number												○	0 to 99 (Decimal)

Function

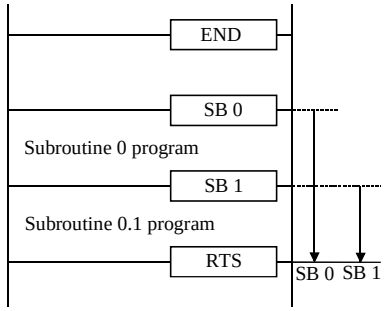
- This command indicates the start of a subroutine program (processing is not performed).
- The n in the SB n cannot be used multiple times in the same program.
- Even if a startup condition is used for SB n, it will be ignored.
- Always use SB n and RTS in pairs.
- Code the SB n to RTS subroutine program after the END command.

Cautionary notes

- This command is checked prior to execution, and when there is an error, the following error code is set in the special internal output WRF001. Also, the CPU error code "34" is set to special internal output WRF000.

CPU error code	Special internal output	Error code	Error description
34	WRF001	H0004	Duplicate definition of SB
		H0013	SB undefined

Instruction for use



- When CAL 0 is executed, SB 0 to RTS is executed as a subroutine.
- When CAL 1 is executed, SB 1 to RTS is executed as a subroutine.

SB n

Item number	Control commands-10	Name	End of subroutine program (RETURN SUBROUTINE)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
RTS	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps										
	Condition	Steps	17	←	21	←	38	←			
RTS	—	1			16.7	←					

Usable I/O	Bit				Word				Double word			Constant	Other
	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		

Function

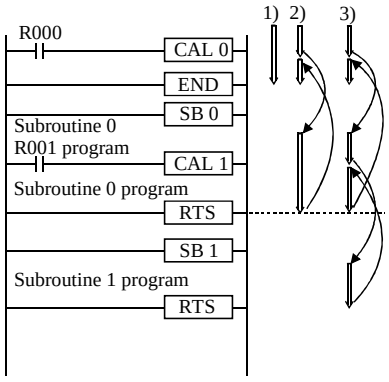
- This command declares the end of a subroutine program.
- When this command is executed, the program is resumed starting from the line following the CAL n command that called the subroutine.
- Do not set a startup condition for this command.

Cautionary notes

- This command is checked prior to execution, and when there is an error, the following error code is set in the special internal output WRF001. Also, the CPU error code "34" is set to special internal output WRF000.

CPU error code	Special internal output	Error code	Error description
34	WRF001	H0011	RTS undefined
		H0020	RTS area error
		H0030	RTS startup condition error

Instruction for use



1] Program execution when R000 and R001 are both off

2] Program execution when R000 is on and R001 is off

CAL 0 is executed, then the subroutine 0 program is executed.
CAL 1 is not executed, the subroutine 0 program is terminated and the execution is returned to the code following the CAL 0.

3] Program execution when R000 and R001 are both on

CAL 0 is executed, then the subroutine 0 program is executed.

CAL 1 is executed, then the subroutine 1 program is executed.

The subroutine 1 program is completed and execution is returned to the code following the CAL 1.

The subroutine 0 program is completed and execution is returned to the code following the CAL 0.

Item number	Control commands-11	Name	Start interrupt scan program (INTERRUPT)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
INT n	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					0.05	←	0.65 ..... 0.65	←	1.3	←	
INT n	Condition: —			Steps: 1								

Usable I/O	Bit				Word				Double word			Constant	Other
	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
n Interrupt priority												○	0 to 2 (Decimal)

Function

- This command declares the start of an interrupt scan program.
- n=0 to 2 indicates a periodical interrupt scan.
- It is set to the 10 ms periodic scan when n=0, 20 ms periodic scan when n=1, and 40 ms periodic interrupt scan when n=2.
- The smaller the number n, the higher the interrupt priority.
- Always use INT n and RTI in pairs.
- Even if a startup condition is used for INT n, it will be ignored.
- Code the INT n to RTI subroutine program after the END command.
- The n in INT n cannot be used multiple times within the same program.

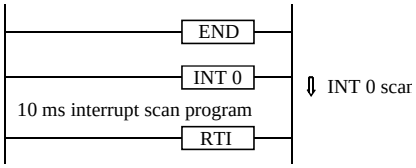
Cautionary notes

- This command is checked prior to execution, and when there is an error, the following error code is set in the special internal output WRF001. Also, the CPU error code "34" is set to special internal output WRF000.

CPU error code	Special internal output	Error code	Error description
34	WRF001	H0005	Duplicate definition of INT
		H0014	INT undefined

- The progress value is updated when the timer command is expected in the EH-CPU448. Therefore, if a program that does not scan the timer command execution section after the timer is activated is created using the interrupt scan, the timer may not turn on correctly. (The timer does not turn on correctly when the time that does not scan the timer command execution section exceeds the time calculated by multiplying the time base by 65,535.) Note that the previous progress value is also retained until the timer command is executed.

Instruction for use



- The program between INT0 and RTI is started and executed every 10 ms.

Item number	Control commands-12	Name	End interrupt scan program (RETURN INTERRUPT)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
RTI	DER	ERR	SD	V	C							
	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps									
	Condition	Steps	8	←	5	←	22	←		
RTI	—	1			5					

Usable I/O	Bit				Word				Double word			Constant	Other
	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		

Function

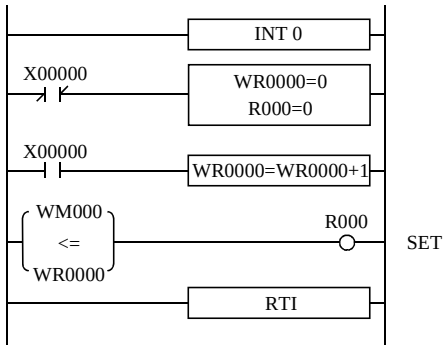
- This command declares the end of an interrupt scan program.
- When this program is executed, the processing is returned to the program that was executing before the interrupt scan was performed.
- Do not set a startup condition for this command.

Cautionary notes

- This command is checked prior to execution, and when there is an error, the following error code is set in the special internal output WRF001. Also, the CPU error code “34” is set to special internal output WRF000.

CPU error code	Special internal output	Error code	Error description
34	WRF001	H0012	RTI undefined
		H0021	RTI area error
		H0031	RTI startup condition error

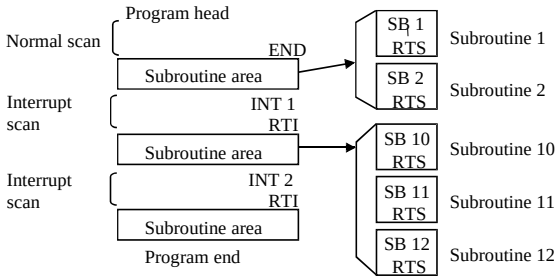
Instruction for use



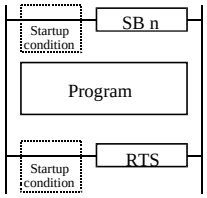
- A 0.01s timer is created using 10 ms interval interrupts.
- WM000 is used for the set value, WR0000 for the progress value and R000 for the timer coil.
- When X00000 is off, the progress value and timer coil are cleared.
- When X00000 is on, the progress value increments by 1 every 10 ms.
- The timer coil is turned on upon WM000<=WR0000.

Syntax of SB n, RTS, INT n and RT1

1] A subroutine can be programmed between a normal scan and interrupt scan, between two interrupt scans, or after the final interrupt scan.

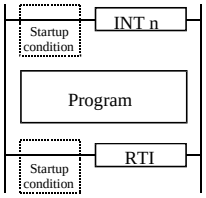


2] Program the subroutine start (SB n) and subroutine end (RTS) commands without specifying startup conditions.

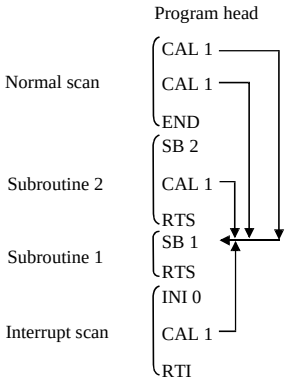


- The RTS startup condition error will occur during operation pre-processing.

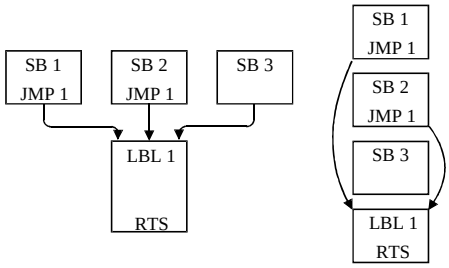
3] Program the interrupt scan start (INT n) and scan complete (RTI) commands without specifying startup conditions.



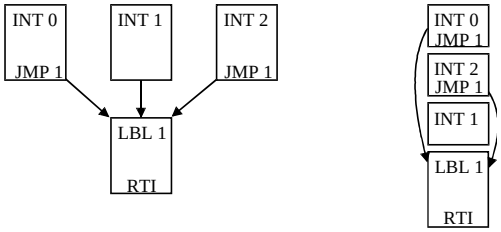
4] The same subroutine can be called from a normal scan, interrupt scan or subroutine.



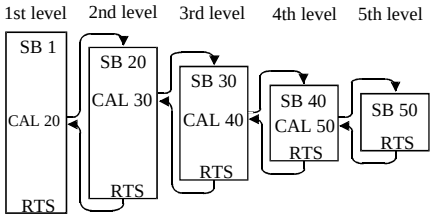
5] It is also possible to program a subroutine with multiple entry points and one exit.

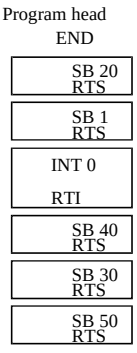


6] It is also possible to program a interrupt scan with many entry points and one exit.



7] Nesting of subroutines is allowed up to 5 levels.





(1) As shown to the left, the subroutine program order and nesting order have no relationship.

RTI

Item number	Transfer commands-1	Name	General-purpose port transmission command

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
TRNS 0 (d, s, t)	DER	ERR	SD	V	C							
	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					61	589	57	571	168	1,301	
	Condition			Steps								
TRNS 0 (d, s, t)	—			5				142	1,465			

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Module loading location						○							
s	Head of parameter area							○						s uses up to s+14.
t	Head of communication control bit			○										t uses up to t+11.

Function

(1) This is a communication command for general-purpose serial ports used in the CPU ladder program.

(2) The WY to which an arbitrary I/O is assigned is set in d. (Since this is used as a dummy, it is possible to set open n points to a slot that does not actually exist.)

(3) s is used to set the head I/O number of the parameter area, in which various communication parameters (head and size of communication data area, timeout value, receiving data length, transmission code and transmission parameter) are set.

(4) t is used to set the head I/O number of the communication control bit area, in which the start of communication, control bits for initial settings and the determination as to whether or not the communication ended properly are stored.

(5) The TRNS 0 command is a command to perform reception after transmission.

(6) Area description of s

s	1] Return code
s+1	2] System area (Cannot be used by the user)
s+3	3] Timeout time
s+4	4] Head I/O of transmitting data area
s+6	5] Size of transmitting data area
s+7	6] Head I/O of receiving data area
s+9	7] Size of receiving data area
s+10	8] Receiving data length
s+11	9] Start code
s+12	10] End code
s+13	11] Transmission speed
s+14	12] Transmission format

: User write-prohibited area

: User setting area

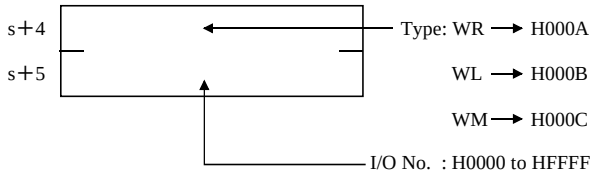
1] Return code: Set in the lower 8 bits of the execution result of the TRNS 0 command as follows:
Normal completion → 0
Abnormal completion → ≠ 0

2] System area: Used by the system processing of the TRNS 0 command when the command is executed.
This area may not be used by the user.

⚠ Don't set some value in this area by the program, the force set, and so on. A serious failure occurs in the CPU, and a movement is suspended.

3] Timeout time: Designates the timeout time from the start of TRNS 0 command execution to completion, as follows:
0: Timeout check is not performed.
≠ 0: × 10ms timeout check is performed (The maximum possible set value is HFFFF.).

4] Head I/O of transmitting data area:
When transmitted by the TRNS 0 command, designates the type and number of the head I/O of the area in which transmitting data is stored.



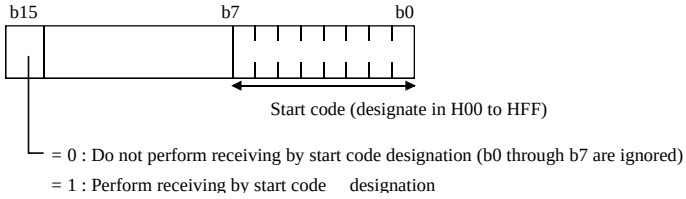
5] Size of transmitting data area:
The size of the transmitting data area is designated in word units.

6] Head I/O of receiving data area:
Designates the type and number of the head I/O of the area in which the response data, with respect to the command or data transmitted, is stored (the area composition is the same as the transmitting data area).

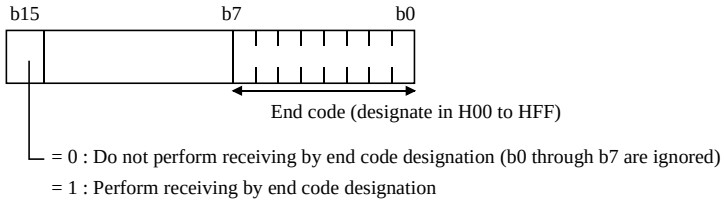
7] Size of receiving data area:
The size of the receiving data area is designated in word units.

8][*1] Receiving data length:
Receiving data length is designated in byte units. However, do not exceed the maximum value (256 bytes) or the receiving data area. If either is exceeded, DER is equal to "1" and ends abnormally.

9][*1] Start code:
Designates the receiving start code.



10][*1] End code:
Designates the receiving end code.



11] Transmission speed:
Designates the transmission speed.

Baud rate	Set value
19200 bps	H0006
9600 bps	H0005
4800 bps	H0004
2400 bps	H0003
1200 bps	H0002
600 bps	H0001
300 bps	H0000

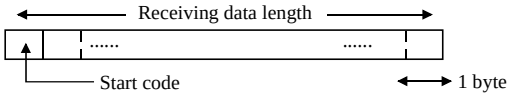
12] Transmission format:
Designates the transmission format.

Transmission code	Set value
7-bit even parity, 2 stops	H0000
7-bit odd parity, 2 stops	H0001
7-bit even parity, 1 stop	H0002
7-bit odd parity, 1 stop	H0003
8-bit no parity, 2 stops	H0004
8-bit no parity, 1 stop	H0005
8-bit even parity, 1 stop	H0006
8-bit odd parity, 1 stop	H0007

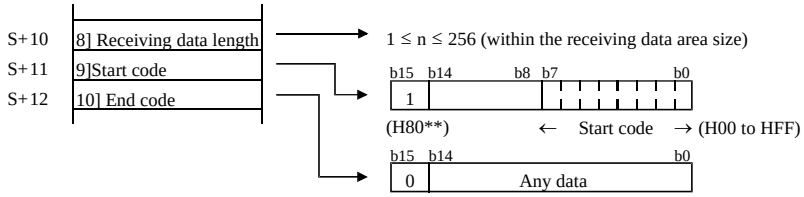
*1 The following four types of data communication methods can be designated depending on the settings of 8] through 10].

(a) Designation by the start code and receiving data length *2

(i) Receiving data structure

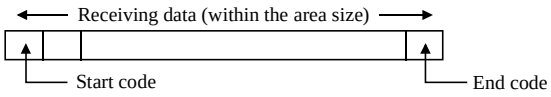


(ii) Parameter settings

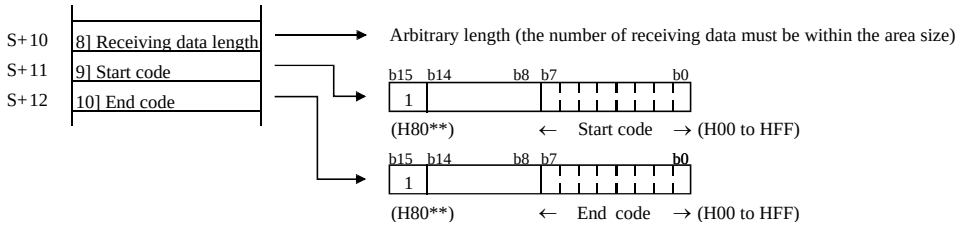


(b) Designation by the start and end codes *2

(i) Receiving data structure

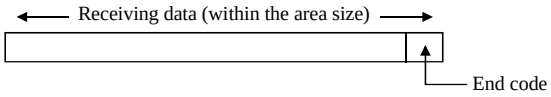


(ii) Parameter settings



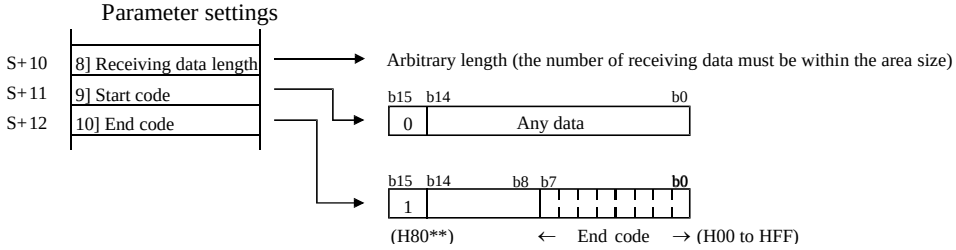
(c) Designation by the end code

(i) Receiving data structure



*2 When receiving data by specifying a start code, the receive buffer may overflow if data with a different start code is sent from the connected device. If this occurs, the data reception will not be executed. Make sure the specified start code is the same as the one used at the connected device.

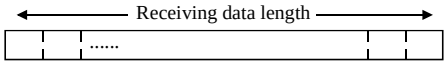
(ii) Parameter settings



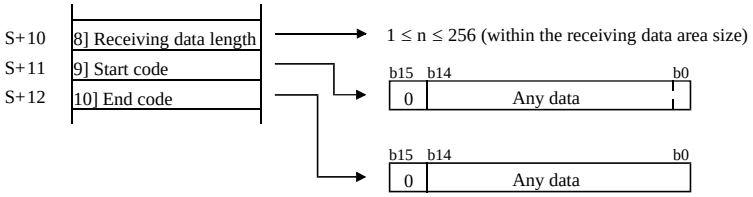
TRNS 0 (d, s, t)

(d) Designation by receiving data length

(i) Receiving data structure



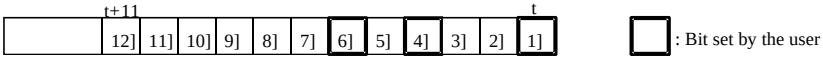
(ii) Parameter settings



Priority order among start code, end code and receiving data length

Start code	0	1	0	1
End code	0	0	1	1
Action	Receives according to the receiving data length	Received according to the start code and receiving data length	Receives according to the end code	Receives according to the codes (data length is ignored)

(7) Description of area t



1] Communication execution:
This bit is set to “1” when the communication is performed using the TRNS 0 command. After communication is completed, the TRNS 0 command resets the area to “0.”

2] Normal completion:
This bit is set to “1” when communication started by the TRNS 0 command ends normally. Also, the TRNS 0 command resets this bit to “0” when communication is commenced.

3] Abnormal completion:
This bit is set to “1” when communication started by the TRNS 0 command ends abnormally. Also, the TRNS 0 command resets this bit to “0” when communication is commenced.

4] Initialization request:
This bit is set to “1” when the TRNS 0 command is initialized. If an initialization request is issued during transmission, communication is terminated forcibly.

5] Initialization end:
This bit is set to “1” when the initialization of the TRNS 0 command is completed normally. At this time, 4] initialization request is reset to 0.”

6] Continuation:
This bit is set to “1” when receiving data immediately after transmission is complete. After the communication, the TRNS 0 command resets this bit to “0.”

7] Parity error:
This bit is set to “1” when a party error occurs during communication.

8] Framing error:
This bit is set to “1” when a framing error occurs during communication.

9] Overrun error:
This bit is set to “1” when an overrun error occurs during communication.

10] Timeout:
This bit is set to “1” when a communication times out.

11] Input buffer full:
This bit is set to “1” when the receiving buffer is full.

12] Conflict error:
This bit is set to “1” when multiple TRANS 0 commands are started simultaneously in the user program, or when a TRANS 0 command and a RECV 0 command are started simultaneously. This forces the communication to terminate.

The TRNS 0 command resets 7] through 12] to “0” during initialization or when the TRNS 0 command is started.

Supplement

- When the CPU receives data from a connected device after reception in your system, if reception is executed with the RECV 0 command after the completion of transmission with the TRNS 0 command, the reception data may not completely be received depending on the timing.
  In such systems, it is recommended that you set 6] Continuation bit to "1" and specify the reception mode after transmission.

(8) Description of the transmitting data area

This is the area to store the data to be transmitted by the TRNS 0 command. Set the data to be transmitted according to the following structure:

1] When even number of bytes are transmitted

Number of bytes to be transmitted (N)	
1st byte	2nd byte
3rd byte	4th byte
5th byte	6th byte
7th byte	8th byte
to	
N-1th byte	Nth byte

2] When odd number of bytes are transmitted

Number of bytes to be transmitted (N)	
1st byte	2nd byte
3rd byte	4th byte
5th byte	6th byte
7th byte	8th byte
to	
N-2th byte	N-1th byte
Nth byte	(Invalid data)

Transmitting data area

(9) Description of the receiving data area

This is the area to store the response with respect to the data transmitted by the TRNS 0 command. The received data is set according to the following structure:

1] When even number of bytes are received

Number of bytes to be received (N)	
1st byte	2nd byte
3rd byte	4th byte
5th byte	6th byte
7th byte	8th byte
to	
N-1th byte	Nth byte

2] When odd number of bytes are received

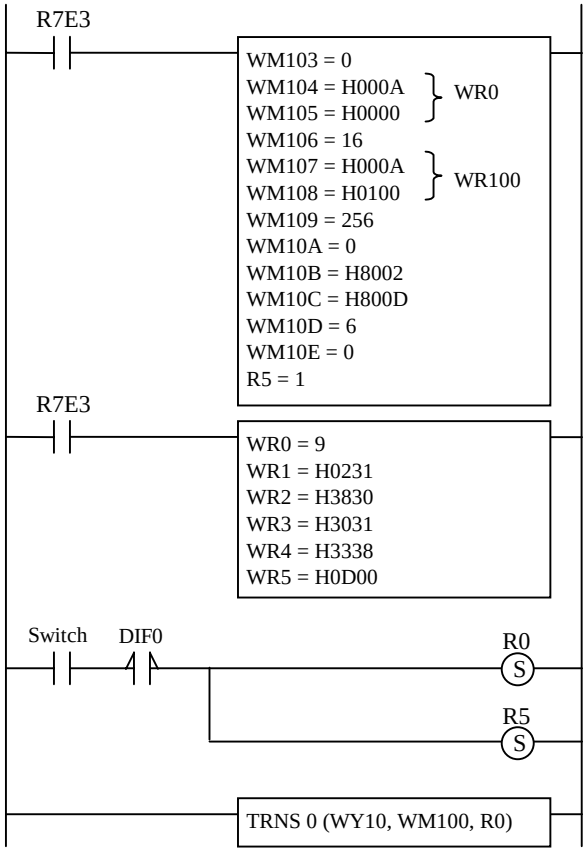
Number of bytes to be received (N)	
1st byte	2nd byte
3rd byte	4th byte
5th byte	6th byte
7th byte	8th byte
to	
N-2th byte	N-1th byte
Nth byte	(Invalid data)

Receiving data area s

Cautionary notes

- The TRNS 0 command initializes the internal work area with one scan ON.
  If there are startup conditions before the TRNS 0 command, the system software may not be able to execute initialization processing normally. Therefore, do not specify startup conditions.
- Set 1] Communication execution bit of the TRNS 0 command at the second scan or later.
- Use this command so that s+14 and t+11 do not exceed the I/O range*. If the I/O range is exceeded, DER is equal to "1" and communication will not be executed.
  * For I/O ranges, refer to the P3-6 and P3-7 performance specification table.
- When there is an error in the parameters set by s to , there may be cases in which the CPU module error "52" is set in special internal output WRF000.
- A CPU turns on an ER signal at the following timing.
  - When communication practice was accepted normally.
- A CPU turns off an ER signal at the following timing.
  - When an initials requirement was turned on during the communication. (But, an ER signal is still on in the case after communication is completed.)
  - When a CPU did STOP during the communication and RUN was done after that. (But, an ER signal is still on in the case after communication is completed.)
  - When time out occurred during the communication.
  - When a s/t parameter was rewritten during the communication and it became a range error.

Program example



R7E3 : 1st scan ON
Timeout = 0

Reserve area for data sending :
16 words from WR0

Reserve area for data receiving :
256 words from WR100

Data receiving definition
Start code : H02, End code : H0D

Communication speed : 19.2k bps
Format : 7 bits, even, 2 stop

Sent data : 9 bytes
Inverter (SJ300/L300P) command
FWD RUN for station No.18
02 31 38 30 30 31 33 38 0D
(STX 18 00 1 38 CR) [38=BCC]

When the switch is ON, execution bit R0 is ON, and data is sent out from CPU port.

R5 enables data receiving from the other device.

Description

This is a sample program for communication with Hitachi inverter (SJ300/L300P).

TRNS 0 parameter and sent data are configured at 1st scan by R7E3 contact.

When the switch is ON, execution bit R0 is ON, and CPU starts sending data.

After sending, CPU gets ready to receive response data from inverter, which is stored in WR100-.

Caution

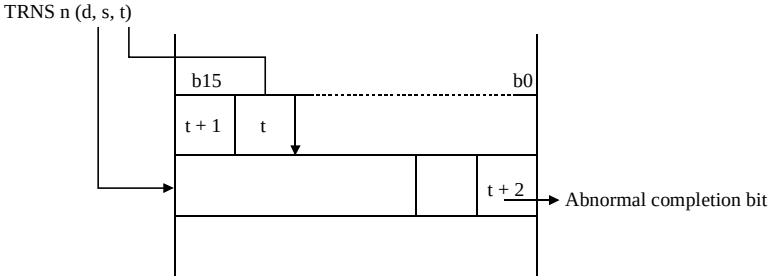
- Parameter WX10 of RECV 0 is sample value in case of input module is mounted on slot 1. Set right parameter according to your I/O configuration. (If input module is on slot 3, this should be WX30.)
- Dip switch must be general purpose port setting (2-OFF, 5-OFF).

TRNS/RECV command return code list

Return code	Name	Description	Corrective action
H00	Normal completion	Transmitting and receiving completed normally.	—
H10	Module error *1	Watchdog timer error	Check the module loading, then restart power or replace the module.
H21	Range check error	The last entry in parameters s and t exceed the I/O range.	Set each parameter area in the valid range.
H22	Transmitting area setting error	The head of the transmitting area is not set correctly.	Set the head of the receiving area in the valid range.
H23	Transmitting area range error	The last entry in the transmitting area exceeds the I/O range.	Set the transmitting area in the valid range.
H24	Receiving area setting error	The head of the receiving area is not set correctly.	Set the head of the receiving area in the valid range.
H25	Receiving area range error	The last entry in the receiving area exceeds the I/O range.	Set the receiving area in the valid range.
H26	Transmitting data length setting error	The transmitting data length is set exceeding the transmitting area length.	Perform setting so that the transmitting data length is within the transmitting area.
H27	Receiving data length setting error	The receiving data length is set exceeding the receiving area length.	Perform setting so that the receiving data length is within the receiving area.
H28	Area overlap error *2	There is an overlapped area for parameters s and t, transmitting area and receiving area.	Set each area so there is no overlap.
H30	Timeout *1	The communication processing did not complete within the set time frame.	Increase the set value or review the processing contents.
H40	Receiving area full *3	Receiving data has been stored to the maximum capacity of the receiving area and there is no more open space.	Increase the receiving area.
H41	Parity error *4	Parity error has occurred during the communication processing.	Verify the transmission path, data format, etc. of the general-purpose port.
H42	Framing error *4	Framing error has occurred during the communication processing.	
H43	Overrun error	Overrun error has occurred during the communication processing.	
H44	Conflict error	TRNS 0/RECV 0 has been started at two or more locations simultaneously.	Change the setting so that it does not startup at two or more locations simultaneously.
H45	Parameter error	The settings for TRNS 0/RECV 0 baud rate, transmission code, etc. are invalid.	Set the valid value.
H46	Port designation error	TRNS 0 or RECV 0 was started when a general-purpose port was not designated.	Check port settings.
H55	Line disconnection	No carrier is set during call origination	Check line connection
H80	Module specific error	The high-function module detected an error.	Refer to the user's manual of each module.

*1 Since the I/F designations of various high-function modules are identical from the stand point of the TRNS or RECV command, it is not possible to determine if the intended type is used. Therefore, no error indicating that the command and the module are mismatched is returned. However, when communication cannot be established due to discrepancy in the register configuration or the handshake bit location, a timeout error (if timeout has been designated) or a module error is returned as the error code.

*2 The return code for an area overlap error is H28, but when there is an area overlap as shown below, the return code may not be displayed as H28, so exercise caution.



If the head area of s and the t area are overlapped, H28 is set in the head area of s, but the abnormal completion bit is set to "1" at the same time, making the value appear as H21.

*3 It is set for the amount of the receiving area size (maximum 256 bytes).

*4 While receiving, the receiving data is not guaranteed.

Item number	Transfer commands-2	Name	General-purpose port receiving command

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
RECV 0 (d, s, t)	DER	ERR	SD	V	C							
	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					61	365	57	579	171	1,305	
	Condition			Steps								
RECV 0 (d, s, t)	—			5				142	1,479			

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Module loading location					○								
s	Head of parameter area							○						up to s + 14.
t	Head of communication control bit			○										up to t + 11.

Function

(1) This is a communication command for general-purpose serial ports used in the CPU ladder program.

(2) A WX to which an arbitrary I/O is assigned is set in d.

(3) s is used to set the head I/O number of the parameter area, in which various communication parameters (head and size of transmitting and receiving data area, timeout value, receiving data length, transmission code and transmission parameter) are set.

(4) t is used to set the head I/O number of the communication control bit area, in which the start of communication, control bits for initial settings and the determination as to whether or not the communication ended properly are stored.

(5) The RECV 0 command is a command to perform reception after transmission.

(6) Area description of s.

s	1] Return code
s+1	2] System area (Cannot be used by the user)
s+3	3] Timeout time
s+4	4] Head I/O of transmitting data area
s+6	5] Size of transmitting data area
s+7	6] Head I/O of receiving data area
s+9	7] Size of receiving data area
s+10	8] Receiving data length
s+11	9] Start code
s+12	10] End code
s+13	11] Transmission speed
s+14	12] Transmission format

: User write-prohibited area

: User setting area

1] Return code: Set in the lower 8 bits of the execution result of the RECV 0 command as follows:

Normal completion → 0

Abnormal completion → ≠ 0

2] System area: Used by the system processing of the RECV 0 command when the command is executed. This area may not be used by the user.

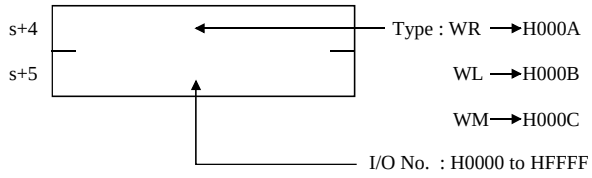
⚠ Don't set some value in this area by the program, the force set, and so on. A serious failure occurs in the CPU, and a movement is suspended.

3] Timeout time: Designates the timeout time from the start of RECV 0 command execution to completion, as follows:

0: Timeout check is not performed.

≠ 0: × 10 ms timeout check is performed (The maximum possible set value is HFFFF.).

4] Head I/O of transmitting data area:
When transmitted by the RECV 0 command, designates the type and number of the head I/O of the area in which transmitting data is stored.



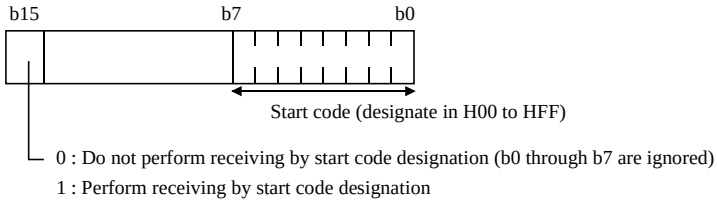
5] Size of transmitting data area:
The size of the transmitting data area is designated in word units.

6] Head I/O of receiving data area:
Designates the type and number of the head I/O of the area in which receiving data is stored (the area composition is the same as the transmitting data area).

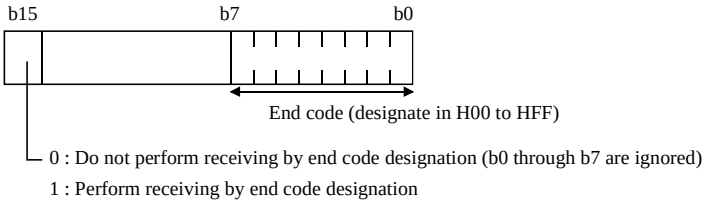
7] Size of receiving data area:
The size of the receiving data area is designated in word units.

8] Receiving data length:
Receiving data length is designated in byte units. However, do not exceed the maximum value (256 bytes) or the receiving data area. If either is exceeded, DER is equal to "1" and ends abnormally.

9] Start code:
Designates the receiving start code. *1



10] End code:
Designates the receiving end code.



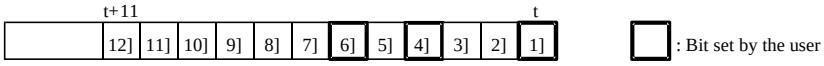
11] Transmission speed:
Designates the transmission speed.
See the TRNS 0.

12] Transmission format:
Designates the transmission format.
See the TRNS 0.

Designate 4] and 5] when transmitting after receiving by the RECV 0 command.

*1 When receiving data by specifying a start code, the receive buffer may overflow if data with a different start code is sent from the connected device. If this occurs, the data reception will not be executed. Make sure the specified start code is the same as the one used at the connected device.

(7) Area description of t



1] Communication execution:
This bit is set to "1" when the communication is performed using the RECV 0 command. After communication is completed, the RECV 0 command resets the area to "0."

2] Normal completion:
This bit is set to "1" when communication started by the RECV 0 command ends normally. Also, the RECV 0 command resets this bit to "0" when communication is commenced.

3] Abnormal completion:
This bit is set to "0" when communication started by the RECV 0 command ends abnormally. Also, the RECV 0 command resets this bit to "0" when communication is commenced.

4] Initialisation request:
This bit is set to "1" when the RECV 0 command is initialised. If an initialisation request is issued during transmission, communication is terminated forcibly.

5] Initialisation end:
This bit is set to "1" when the initialisation of the RECV 0 command is completed normally. At this time, 4] initialisation request is reset to "0."

6] Continuation:
This bit is set to "1" when transmitting after receiving is complete. After the communication, the RECV 0 command resets this bit to "0."

7] Parity error:
This bit is set to "1" when a party error occurs during communication.

8] Framing error:
This bit is set to "1" when a framing error occurs during communication.

9] Overrun error:
This bit is set to "1" when an overrun error occurs during communication.

10] Timeout:
This bit is set to "1" when a communication times out.

11] Input buffer full:
This bit is set to "1" when the receiving buffer is full.

12] Conflict error:
This bit is set to "1" when multiple RECV 0 commands are started simultaneously in the user program. This forces the communication to terminate.

The RECV 0 command resets 7] through 12] to "0" during initialisation or when the TRNS 0 command is started.

(8) Description of the transmission and receiving data area

See the TRNS 0.

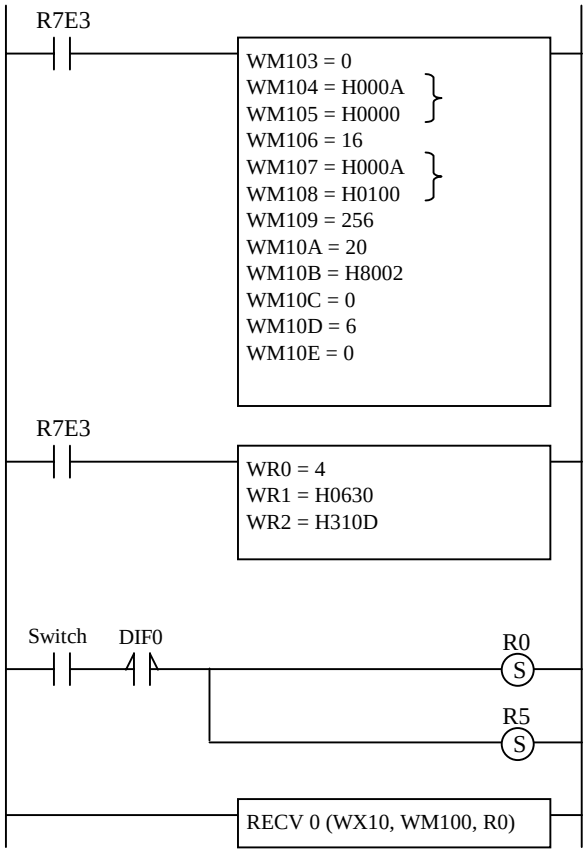
Supplement

- If the CPU transmits data from a connected device after reception in your system, it is recommended that you set 6] Continuation bit to "1" and specify the transmission mode after reception.

Cautionary notes

- The RECV 0 command initializes the internal work area with one scan ON.
  If there are startup conditions before the RECV 0 command, the system software may not be able to execute initialization processing normally. Therefore, do not specify startup conditions.
- Set 1] Communication execution bit of the RECV 0 command at the second scan or later.
- Use this command so that s+14 and t+11 do not exceed the I/O range*. If the I/O range is exceeded, DER is equal to "1" and communication will not be executed.
  * For I/O ranges, refer to the P3-6 and P3-7 performance specification table.
- When the parameters set for s and after are abnormal, there may be cases when "52" is indicated on the CPU module error display.

Program example



R7E3 : 1st scan ON
Timeout = 0

Reserve area for data sending :
16 words from WR0

Reserve area for data receiving :
256 words from WR100

Data receiving definition
Data length 20 bytes, Start code : H02

Communication speed : 19.2k bps
Format : 7 bits, even, 2 stop

Sent data
06 30 31 0D
ACK 01 CR

When the switch is ON, execution bit R0 is ON, and CPU gets ready to receive data.

R5 enables data sending after data receiving.

Description

RECV 0 parameter and sent data are configured at 1st scan by R7E3 contact.

When the switch is ON, execution bit R0 is ON, and CPU gets ready to receive data receiving.

Data ACK01CR (0630310D) is sent back from CPU after data receiving.

Caution

- Parameter WX10 of RECV 0 is sample value in case of input module is mounted on slot 1. Set right parameter according to your I/O configuration. (If input module is on slot 3, this should be WX30.)
- Dip switch must be general purpose port setting (2-OFF, 5-OFF).

Item number	Transfer commands-5	Name	Telecommunication command †

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
TRNS 8 (d, s, t)	DER	ERR	SD	V	C							
	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					128	602	68	532			
TRNS 8 (d, s, t)	Condition			Steps				153	1,100			
	—			5								

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
d	Module loading location						○							
s	Head of parameter area							○						s uses up to s+10.
t	Head of communication control bit			○										t uses up to t+13

Function

(1) This is a communication command for controlling the modem that is used by the CPU's ladder program.

(2) Set the WY to which an arbitrary I/O is assigned in d. (Since this is used as a dummy, it is possible to set open n points to a slot that does not actually exist in I/O assignments.)

(3) For s, set the head I/O number of the parameter area in which various communication parameters (control type, head and size of transmission data area, transmission code and transmission parameter) are specified.

(4) For t, set the head I/O number of the communication control bit area in which the start of communication, control bits for initial settings and the result as to whether or not the communication ended properly are stored.

(5) Description of area s

s	1] Return code
s + 1	2] System area (Cannot be used by the user)
s + 3	3] Control type
s + 4	4] Head I/O of transmission data area
s + 6	5] Size of transmission data area
s + 7	6] Dial interval
s + 8	7] Non-response monitor time
s + 9	8] Modem-PLC transmission speed
s + 10	9] Transmission format

: User write-prohibited area

: User setting area

1] Return code: The execution result of the TRNS 8 command is set in the lower 8 bits as follows:
Normal completion → 0
Abnormal completion → ≠ 0

2] System area: Used for system communication of the TRNS 8 command when the TRNS 8 command is executed.
This area cannot be used by the user.

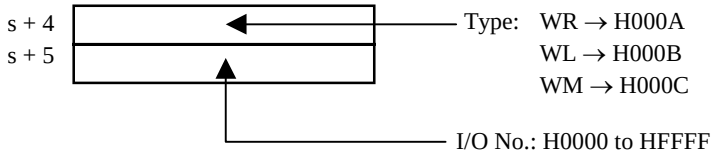
⚠ Don't set some value in this area by the program, the force set, and so on. A serious failure occurs in the CPU, and a movement is suspended.

3] Control type: Specify the control type of the TRNS 8 command.
1: Dedicated/general switch request
2: General/dedicated switch request
3: TEL origination request
4: TEL disconnection request
5: Line connection check

†: Supported by EH-CPU ***A/448/516/548 only.

Function

4] Head I/O of the transmission data area:
Set the command to be sent to the modem with the TRNS 8 command. Specify the initialization command if initialization is required when a dedicated/general switch request is made, the type and number of the head I/O of the area in which the originating telephone number is stored.



5] Transmission data area size:
Specify the size of the transmission data area in units of words.

6] Dial interval:
Set the dial wait time for the second and subsequent dialing.
=0: 5 sec.
≠0: Waits until the next dial with a time of x1s.

7] Non-response monitor time:
Specify the timeout time after a command is issued to the modem.
=0: 10 sec.
≠0: Times out with a time of 1s.

8] Modem-PLC transmission speed:
Specify the transmission speed between the modem and PLC.

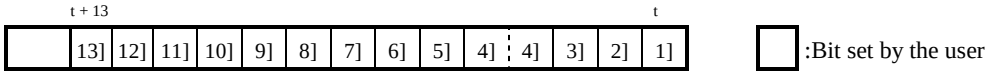
Baud rate	Setting value
115.2 kbps	H0009
57.6 kbps	H0008
38.4 kbps	H0007
19.2 kbps	H0006
9600 bps	H0005
4800 bps	H0004
2400 bps	H0003
1200 bps	H0002
600 bps	H0001
300 bps	H0000

9] Transmission format:
Specify the transmission format.

Transmission code	Set value
7-bit even parity, 2 stops	H0000
7-bit odd parity, 2 stops	H0001
7-bit even parity, 1 stop	H0002
7-bit odd parity, 1 stop	H0003
8-bit no parity, 2 stops	H0004
8-bit no parity, 1 stop	H0005
8-bit even parity, 1 stop	H0006
8-bit odd parity, 1 stop	H0007

Function

(6) Description of area t



1] Communication execution:
This bit is set to “1” when the communication is performed using the TRNS 8 command. After communication is completed, the TRNS 8 command resets the area to “0.”

2] Normal completion:
This bit is set to “1” when communication started by the TRNS 8 command ends normally. Also, the TRNS 8 command resets this bit to “0” when communication is commenced.

3] Abnormal completion:
This bit is set to “1” when communication started by the TRNS 8 command ends abnormally. Also, the TRNS 8 command resets this bit to “0” when communication is started.

4] Being connected (2 bits):
Sets the line status. When an originating call is being connected, “01” is set. When a incoming call is being connected, “10” is set. If nothing is connected, “00” is set. The TRNS 8 command resets this bit to “00” when communication is starts. This bit is set when communication is completed normally.

5] Dial busy:
This bit is set to “1” when dial busy is detected after call origination.

6] Line busy:
This bit is set to “1” when line busy is detected after call origination.

7] Received:
This bit is set to “1” when a call is received when a switch request or originating request is made.

8] No response:
This bit is set to “1” when there is no response after dedicated/general switching.

9] Parity error:
This bit is set to “1” when a party error occurs.

10] Framing error:
This bit is set to “1” when a framing error occurs.

11] Overrun error:
This bit is set to “1” when an overrun error occurs.

12] Dedicated port being connected
This bit is set to “1” when the dedicated port is being connected after a switch request is made.

13] Contension error:
This bit is set to “1” when multiple TRANS 8 commands are started simultaneously by the user program, or when the TRANS 0 and RECV 0 commands are started simultaneously. This forces the communication to terminate.

The TRNS 8 command resets 5] through 13] to “0” when the TRNS 8 command is started.

Function

(7) Description of the transmission data area

This area stores the data to be transmitted to the modem with the TRNS 8 command, and is used for the initialization command and call origination. Set the data to be transmitted according to the following format:

1] When even number of bytes are transmitted

Number of bytes to be transmitted (N)	
1st byte	2nd byte
3rd byte	4th byte
5th byte	6th byte
7th byte	8th byte
:	
N-1th byte	Nth byte

2] When odd number of bytes are transmitted

Number of bytes to be transmitted (N)	
1st byte	2nd byte
3rd byte	4th byte
5th byte	6th byte
7th byte	8th byte
:	
N-2th byte	N-1th byte
Nth byte	(Invalid data)

Transmission data area size

(i) When the initialization command is required during dedicated/general switch request

Specify the initialization command excluding the head AT.

(Example) Initialization command: To send ATE0Q0V0 & CI &SI to the modem

H000C	
H45 (E)	H30 (O)
H51 (Q)	H30 (O)
H56 (V)	H30 (O)
H26 (&)	H43 (C)
H31 (1)	H26 (&)
H53 (S)	H31 (1)

Set the following values when the modem is initialized*:

Result code: Output
Display format of result code: Numeric format
Echo back: None
DR signal: Always on
CD signal control: Follows the carrier of the opposite modem
ER signal control: Disconnects the line by switching from on to off

* The model may not operate normally if it is used with settings other than above. Be sure to set the model correctly by referring to the instruction manual of your modem.

(ii) When the initialization command is not required during dedicated/general switch request

Only AT can be sent by setting the transmission size to zero.

H0000	

Function

(iii) For call origination

Specify the telephone numbers to be dialed for call origination. Multiple numbers may be specified by delimiting each number with a space (H20). If two consecutive spaces are entered, it is considered that no telephone numbers are present thereafter. If a space is entered at the beginning, no call origination will be performed.
To redial the same telephone number, specify the same telephone number for the number of times to be called.
To redial different telephone numbers, specify different telephone numbers.

(Example) To dial TEL 03-1111-2222 three times

H0020	
H30(0)	H33(3)
H31(1)	H31(1)
H31(1)	H31(1)
H32(2)	H32(2)
H32(2)	H32(2)
H20( )	H30(0)
H33(3)	H31(1)
H31(1)	H31(1)
H31(1)	H32(2)
H32(2)	H32(2)
H32(2)	H20( )
H30(0)	H33(3)
H31(1)	H31(1)
H31(1)	H31(1)
H32(2)	H32(2)
H32(2)	H32(2)

Notes

- Use this command so that s+10 and t+13 do not exceed the I/O range*. If the I/O range is exceeded, DER is equal to “1” and communication will not be executed.
  * For the I/O ranges, see the EH-CPU448 Performance Specifications Table on page 3.
- Control type

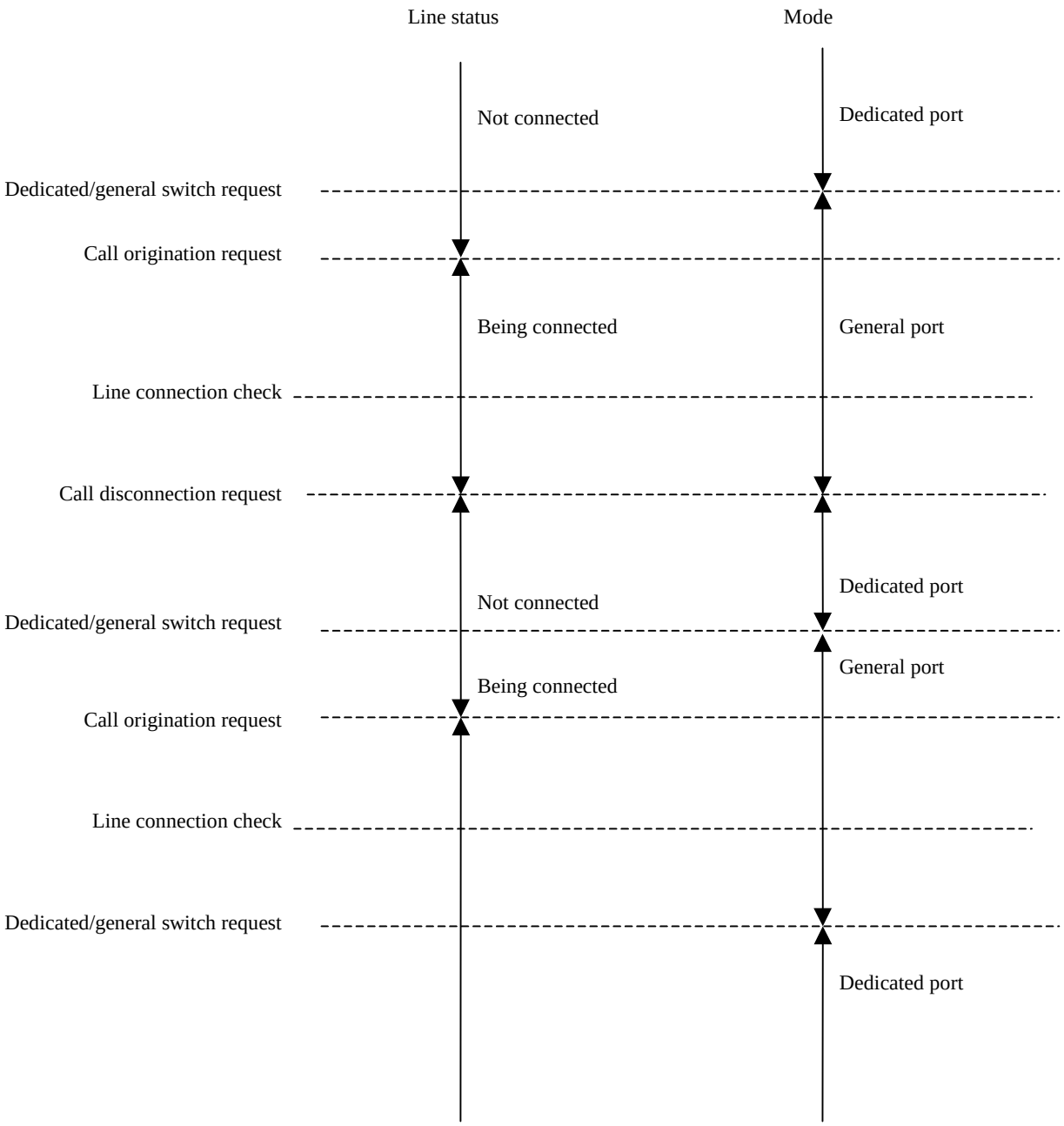
1] Dedicated/general switch request
Switches communication from the dedicated port communication to the general port communication. When the switching is completed normally, the general port is enabled. If an error occurs, the switching will not be performed. If the initialization command is required for the modem, the initialization command can be specified. If it is not required, set the transmission size to zero. This can be executed only when the line is being disconnected.

2] General/dedicated switch request
Switches communication from the general port communication to the dedicated port communication. If this switching is done while the line is being connected, the transmission speed and transmission format specified for the general port must match those for the dedicated port.

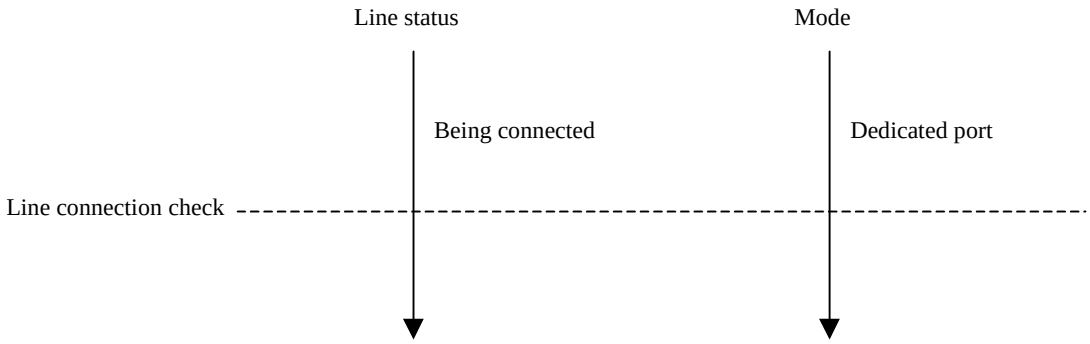
3] Call origination request
Performs a call origination to the specified telephone number.

4] Call disconnection request
Performs a call disconnection. Both the dedicated port and general port can be disconnected. When the disconnection is completed normally, the dedicated port is enabled.

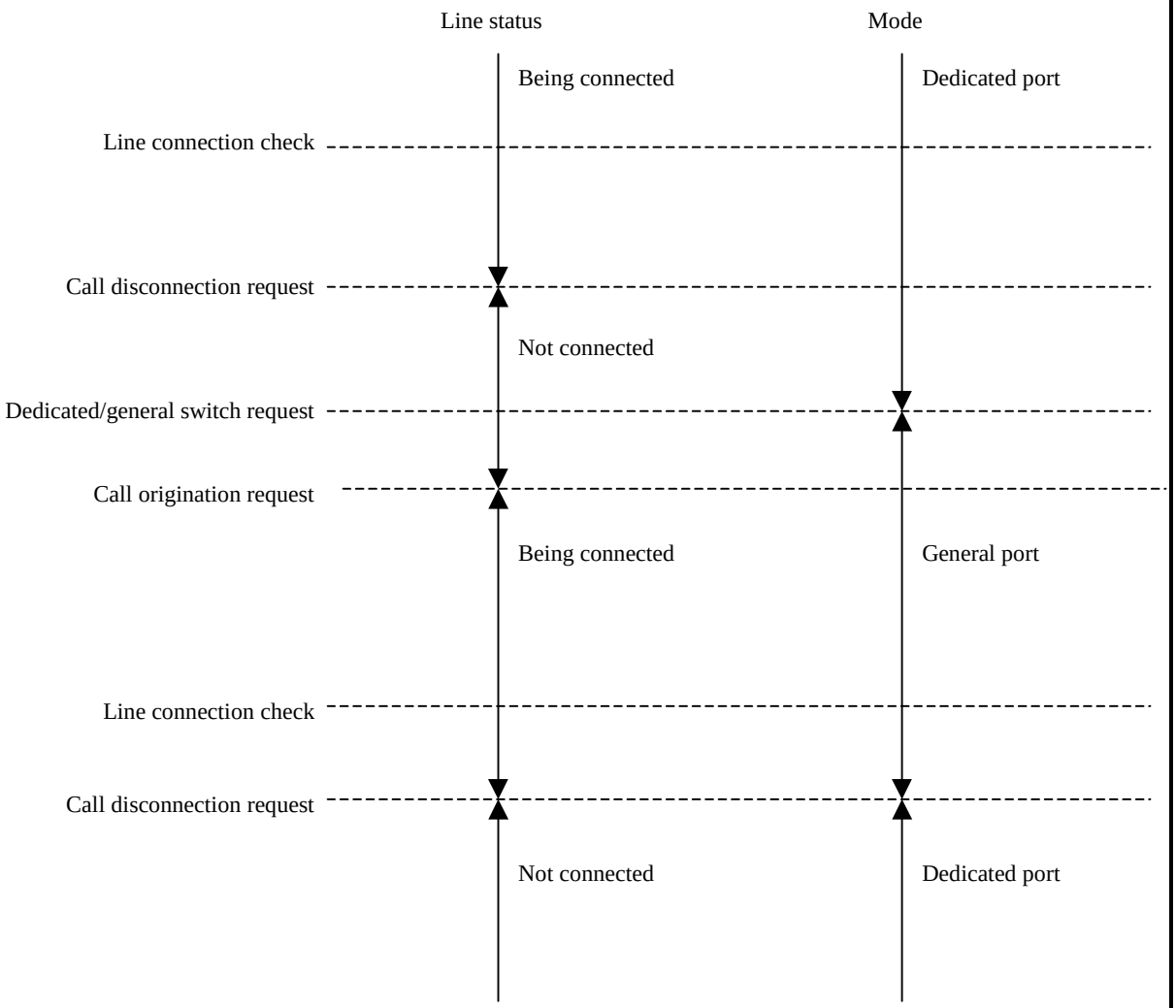
5] Line connection check
Checks the line connection status.


Line status
Mode
Not connected
Dedicated port
Dedicated/general switch request
Call origination request
Being connected
General port
Line connection check
Call disconnection request
Dedicated port
Not connected
Dedicated/general switch request
General port
Being connected
Call origination request
Line connection check
Dedicated/general switch request
Dedicated port

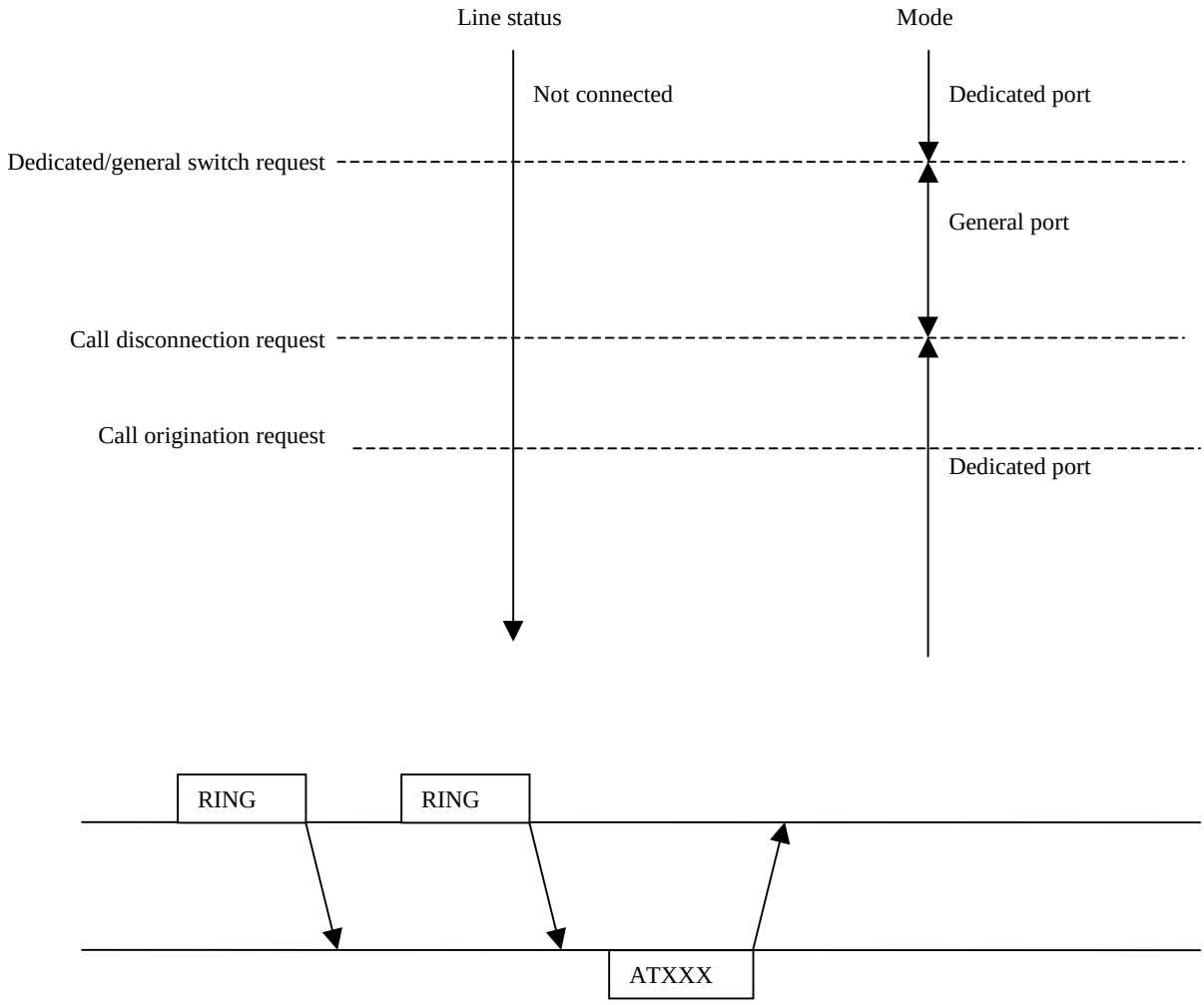
Dedicated port being connected Line check Part 1



Dedicated port being connected Line check Part 2



Dedicated port being connected Call origination



Call termination (ring reception) check is not performed before the AT command is issued.

It may result in either normal completion or abnormal completion (line busy, termination detected, no response). In case of abnormal completion, retry the operation.

When changed from RUN to STOP:

The port mode is set to the dedicated port.

If a modem is connected using the general port, the line will be placed in the disconnected status.

If the port is the dedicated port, no status change will occur.

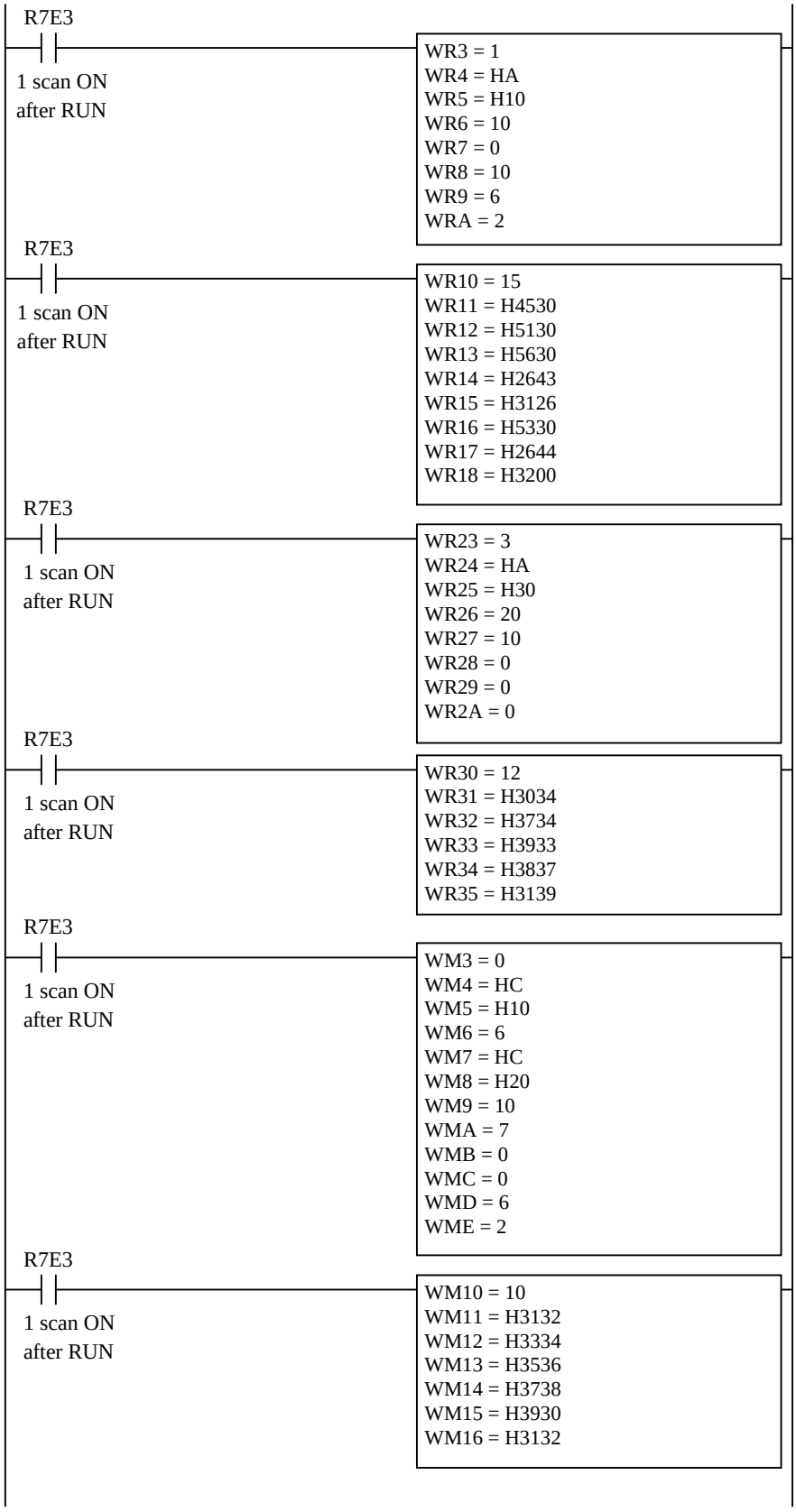
Error code

TRNS 8 error code list

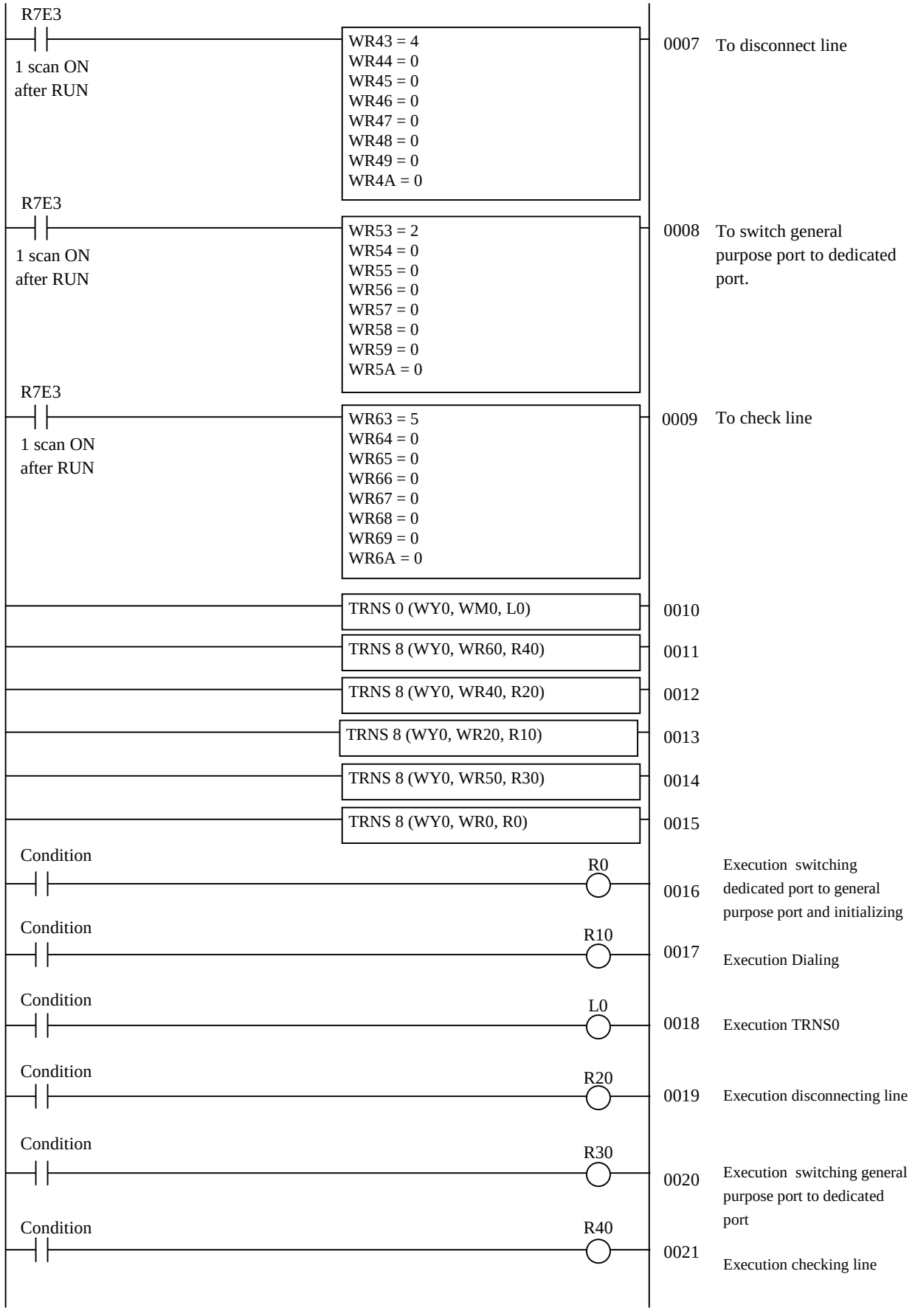
Types of errors: 1] Dedicated/general port switch request, 2] General/dedicated port switch request, 3] Call origination request, 4] Call disconnection request, 5] Line connection check

Return code	Error name	Description	Type of error
21h	Range check error	The s parameter and/or t parameter exceeded the I/O range.	1], 2], 3], 4], 5]
22h	Transmission area setting error	The starting address setting of the transmission area is incorrect.	1], 2]
23h	Transmission area range error	The transmission area exceeded the I/O range.	1], 2]
26h	Transmission data length error	The transmission data length is longer than the transmission area.	1], 2]
28h	Area overlap	The s parameter, t parameter and transmission area are overlapped.	1], 2], 3], 4], 5]
41h	Parity error	A parity error occurred during result reception.	1], 2]
42h	Framing error	A framing error occurred during result reception.	1], 2]
43h	Overrun error	An overrun error occurred during result reception.	1], 2]
44h	Contention error	More than one TRNS 8 command were started simultaneously. The TRNS 8 command was started with the TRNS 0 command simultaneously. The TRNS 8 command was started with the RECV 0 command simultaneously.	1], 2], 3], 4], 5]
45h	Parameter error	The setting values of the baud rate, transmission format and control type are incorrect.	1], when abnormal type
46h	Port specification error	The DIP switches are not set for modem connection.	1], 2], 3], 4], 5]
50h	Dial busy	Busy during call origination	2]
51h	Line busy	Line busy during call origination	2]
52h	Termination detected	When switching or during call origination	1], 2]
53h	No response	The no-response time has reached without the response of the AT command.	1]
54h	Dedicated port is being connected	Being connected to the dedicated port during switch request.	1]
55h	Line disconnection	No carrier is set during call origination.	2]
56h	Abnormal modem response	Response from the modem is abnormal.	1], 2]
57h	Abnormal port mode	The port mode was set to the dedicated port during origination.	2]
58h	Telephone number setting error	The telephone number set in the transmission data area started with a space. The number of data to be sent was zero during origination.	2]
59h	Cannot switch from general to dedicated port	Switching to the dedicated port did not complete normally.	5]

Program description



Program description



Item number	Fun commands-1	Name	PID Initialization †

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU3**A EH-CPU3**		Other than left		
FUN 0 (s)	DER	ERR	SD	V	C							
* (PIDIT (s))	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					1,661	3,154	2980 EH-CPU3**A / 7,389 EH-CPU3**	5718 EH-CPU3**A / 12,572 EH-CPU3**	4,421	8,493	
	Condition			Steps								
FUN 0 (s) * (PIDIT (s))	—			3								

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	PID control table							○						WR only

Function

- The FUN 0 (s) initializes the area in which the initialization set data required for PID operation is stored.
- The (s) in the FUN 0 (s) is used to specify the head number of WR of the PID management table.
- If there is an error in the contents specified in the PID control table, an error code will be set in error code 0 of the PID control table and initialization will not be performed.
- Once initialization is successfully completed (FUN 0 normal completion (“1”) in the PID management table), re-executing the FUN 0 will generate an error.

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- If difficulty arises when the area used by the PID operation is cleared upon operation start or recovering from a power failure, please specify the power failure memory.

† : Supported by EH-CPU 308(A)/316(A)/448(A)/516/548 only.

Item number	Fun commands-2	Name	PID operation control †

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU3**A EH-CPU3**		Other than left		
FUN 1 (s)	DER	ERR	SD	V	C							
* (PIDOP (s))	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					83	140	85 197	130 429	131	216	
	Condition			Steps								
FUN 1 (s) * (PIDOP (s))	—			3								

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	PID control table							○						WR only

Function

- The FUN 1 (s) determines the loop in which the operation is performed after reading the PID Execution flag from the bit table area of the loop and the PID Constant Change flag.
- Set (s) in the FUN 1 (s) as the head number of the PID control table. If set differently, an error will be generated and an error code will be set to error codes 0 and 1 of the PID control table, resulting in the FUN 1 not being executed.
- Program the FUN 1 (s) so that it is executed once during the 20 ms periodic scanning.

* ( ) indicates the display when the LADDER EDITOR is used.

†: Supported by EH-CPU 308(A)/316(A)/448(A)/516/548 only.

Item number	Fun commands-3	Name	PID calculation process †

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU3**A EH-CPU3**		Other than left		
FUN 2 (s)	DER	ERR	SD	V	C							
* (PIDCL (s))	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps										
	Condition	Steps									
FUN 2 (s)	—	3	87	←	91 183	←	183	—			
* (PIDCL (s))											

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Word table							○						WR only

Function

- The sampling time set in the word table for each loop determines whether or not PID calculation is performed.
- The FUN 2 (s) turns ON the PID Calculation In Progress flag of the loop that is being calculated.
- Set all of the head number of WR of the word table for each PID loop of the FUN 2 (s).
- The FUN 2 (s) will check for the output upper limit and low limit values, set value bit pattern, and range of the output value bit pattern for each loop. If an error is generated, the FUN 2 Error flag of the loop bit table will turn ON and an error code is set to error code 2 of the PID control table. The FUN 2 will be executed even if an error is generated.
- Program the FUN 2 (s) so that it is executed during the 20 ms periodic scanning.

* ( ) indicates the display when the LADDER EDITOR is used.

†: Supported by EH-CPU 308(A)/316(A)/448(A)/516/548 only.

(1) PID control table (In the case of FUN 0 (WRxxxx))

(a) Structure of PID management table (1)

Sets the header number of the WR used as the PID control table in s of FUN 0 (s). The PID control table is comprised of 2], 3], 4] and 5], and the size of the table increases by the number of loops 3]. Make sure that the maximum number of the WR is not exceeded. Otherwise, error code H0004 will be written in error code 0 2].

Address	Contents	Details	Remarks
xxxx	Error code 0 *1 (Read)	• Sets the error code generated by FUN 0 processing or some part of FUN 1 processing. • If no error is present, the prior status is maintained.	2]
xxxx + 1	Error code 1 *1 (Read)	• Sets the error code generated by FUN 1 processing. • If no error is present, the prior status is maintained.	
xxxx + 2	Error code 2 *1 (Read)	• Sets the error code generated by FUN 2 processing. • If no error is present, the prior status is maintained.	
xxxx + 3	FUN 0 Normal completion 1 (Read)	• Sets H0001 when FUN 0 (PID initialization) is executed normally. • If an error is generated, the value will be H0000, and an error code will be set in error code 0.	5]
xxxx + 4	Number of loops (Write) *2	• Sets the number of loops used in a range between 1 and 64. • If the value is 0, H0002 is written in error code 0, and the PID will not be processed. (Even if the FUN 1 and FUN 2 are programmed, PID will not be processed.)	3]
xxxx + 5	Head address of the WR of the word table for loop 1 (Write) *2	• 48 words are used per loop for PID constant input and for PID internal calculations. If the maximum WR number is exceeded, error code XX05 will be written in error code 0.	4]
xxxx + 6	Head address of the WR of the word table for loop 2 (Write) *2	• 48 words are used per loop for PID constant input and for PID internal calculations. If the maximum WR number is exceeded, error code XX05 will be written in error code 0.	
xxxx + 7	Head address of the WR of the word table for loop 3 (Write) *2	• 48 words are used per loop for PID constant input and for PID internal calculations. If the maximum WR number is exceeded, error code XX05 will be written in error code 0.	
• • •	• • •	• • •	
xxxx + 44	Head address of the WR of the word table for loop 64 (Write)*2	• 48 words are used per loop for PID constant input and for PID internal calculations. If the maximum WR number is exceeded, error code XX05 will be written in error code 0.	

*1 Error codes are expressed as a four-digit hexadecimal value. For more information, see the Error Code Details.

*2 The (Write) in the above table indicates the areas where the user enters data using a program. (It is also possible to read data.)

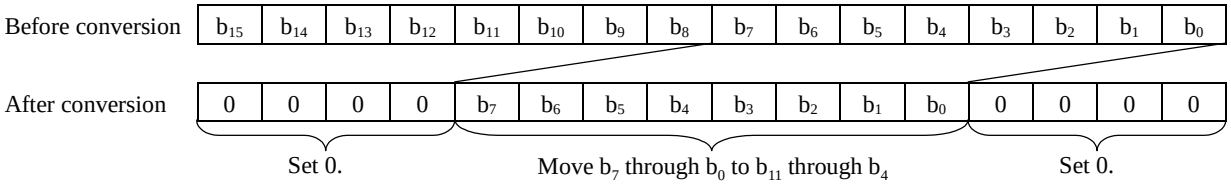
(b) Word table and bit table for each loop

(If the content of xxxx+5 in (a) is ADRIO (xxxx+5, yyyy))

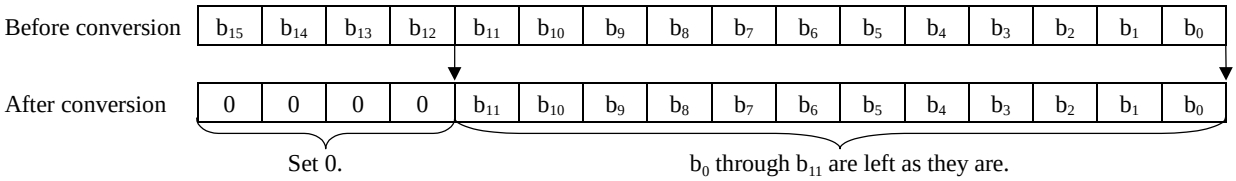
Address	Contents	Specifications	Notes	Remarks
yyyy	ADRIO (yyyy, zzzz) zzzz is the header number of the bit internal output.	Sets the header address of the bit table.	Uses 16 bits per loop. Set the actual address of the header number using the ADRIO command so the last suffix of the bit internal output is not exceeded.	11]
yyyy + 1	Sampling time TZ	When 1 to 200 (× 20 ms) analog I/O is installed in a basic base or extended base.	• Set a multiple of the minimum set value. • The minimum set value is the value set to the number of loops 3].	12]
yyyy + 2	Proportional gain KP	– 1,000 to +1,000	Corresponds to -10.00 to +10.00.	13]
yyyy + 3	Integral content Ti/TZ	1 to 32,767	Value is set to Ti/(Sampling time x 20 ms)	14]
yyyy + 4	Derivative constant TD/TZ	1 to 32,767	Value is set to Ti/(Sampling time x 20 ms)	15]
yyyy + 5	Derivative delay constant Tn/TZ	1 to 32,767	Value is set to Ti/(Sampling time x 20 ms)	16]
yyyy + 6	Output upper limit value UL	– 32,767 to 32,767	The following condition must be met. LL ≦ INIT ≦ UL	17]
yyyy + 7	Output low limit value LL	– 32,767 to 32,767		18]
yyyy + 8	Initial value INIT	– 32,767 to 32,767		19]
yyyy + 9	Set value I/O number (Write)	Set the actual address of the word number of the I/O for which the set value is set.		20]
yyyy + A	Measured Value I/O number (Write)	Set the actual address of the word number of the I/O for which the measured value is set.		21]
yyyy + B	Output value I/O Number (Write)	Set the actual address of the word number of the I/O that outputs the PID calculation results.		22]
yyyy + C	Set value bit pattern (Write)	Determine the method that is used to convert the set value to the 16-bit data in which the PID operation is performed. See *1 below and use a value between H0001 and H0004.		23]
yyyy + D	Measured value bit pattern (Write)	Determine the method that is used to convert the data read from the measured value I/O number 21] to the 16-bit data. (See the set value bit pattern 23].)		24]
yyyy + E	Output value bit pattern (Write)	• Write to the output value I/O number 22] after converting the results of the FUN 2 process or PID calculation according to the output value bit pattern 25]. • Use a value between H0001 and H0004 in *2 depending on the type of output I/O.		25]
yyyy + F ↓ yyyy + 2F	PID calculation area (Cannot be used by the user)	Do not use this in user programs because this is used by FUN 0, FUN 1, and FUN 2 processing.		26]

*1:

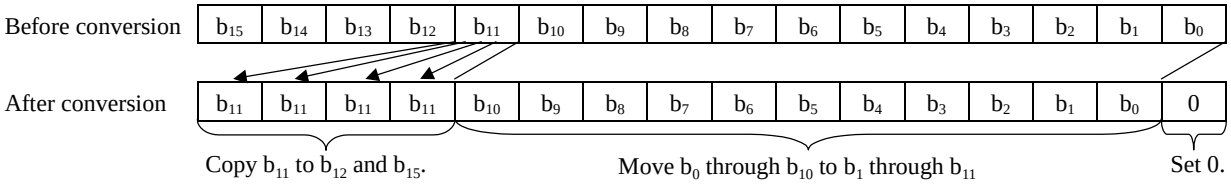
H0001: 8-bit → 16-bit



H0002: 12-bit unsigned → 16-bit



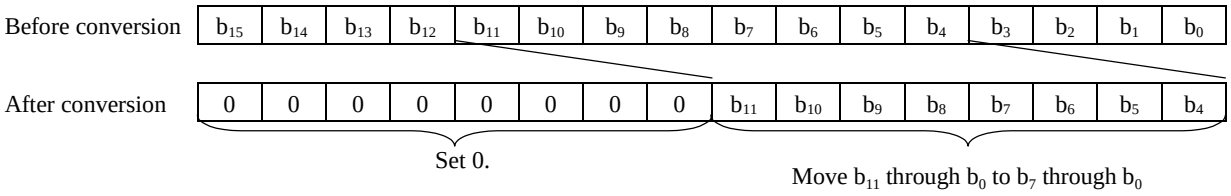
H0003: 12-bit signed → expand the sign to 16-bit



H0004: Do not convert

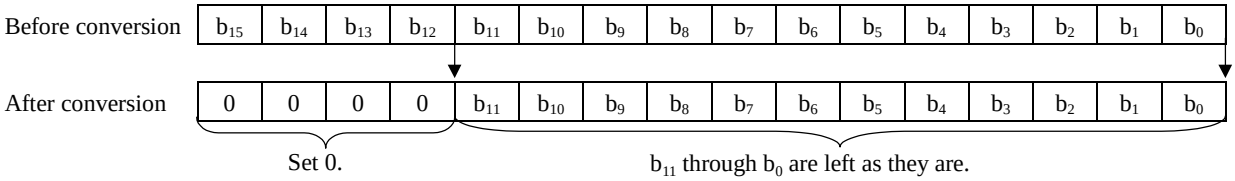
*2 :

H0001: 16-bit → 8-bit



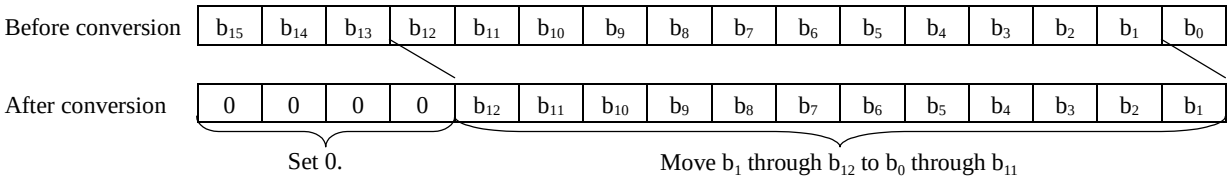
If values are H0FFF through H7FFF before conversion, the values are converted to H00FF.
If values are H8000 through HFFFF before conversion, the values are converted to H0000.

H0002: 16-bit → 12-bit



If values are H0FFF through H7FFF before conversion, the values are converted to H00FF.
If values are H8000 through HFFFF before conversion, the values are converted to H0000.

H0003: 16-bit signed → 12-bit signed



If values are H0FFF through H7FFF before conversion, the values are converted to H07FF.
If values are H8000 through HF000 before conversion, the values are converted to H0800.

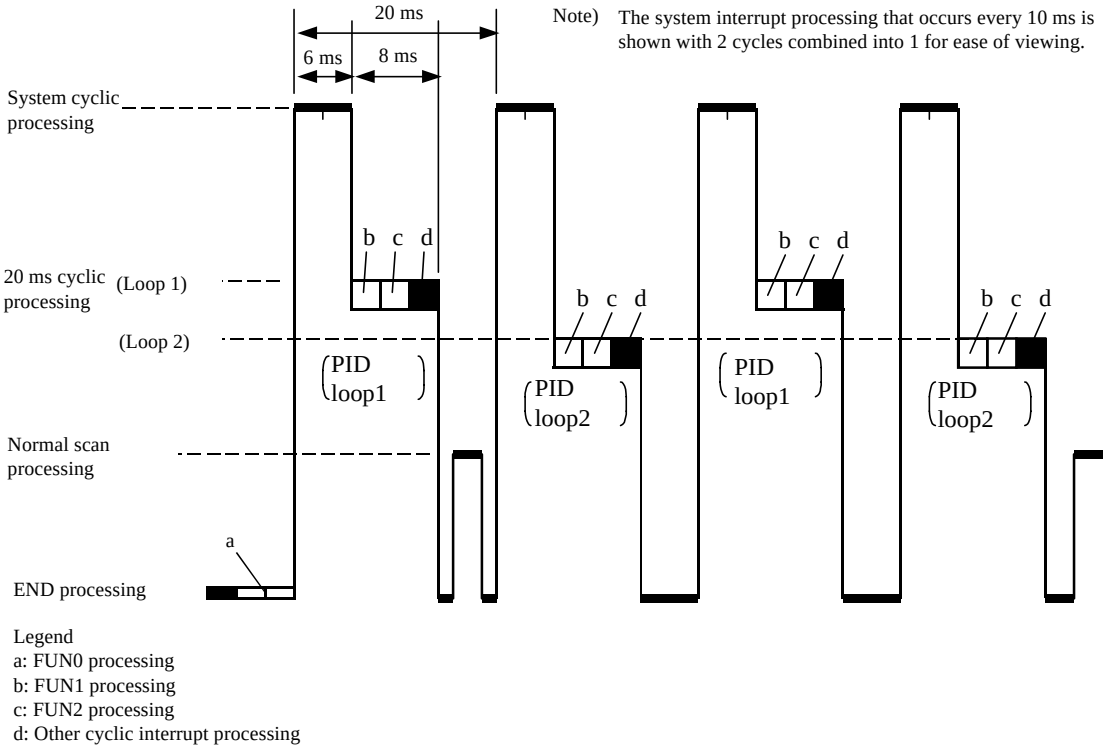
H0004: Do not convert

(c) Details of word tables used for each loop

Address	PID management table	Details	Remarks
zzzz	Execution flag (Write)	• When the Execution flag starts up (0 → 1), the PID constant at that time is checked and the PID calculation value is initialized. If successful, the PID RUN flag 58] is set to "1." If there is an error, the PID RUN flag 58] is set to "0" and PID calculation will not be performed. • PID calculation is performed while the Execution flag = 1. • When the Execution flag = 0, the PID calculation will end and the output will become "0."	50]
zzzz + 1	Non-bumpless flag (Write)	0 : Perform Bumpless processing 1 : Perform non-bumpless processing	51]
zzzz + 2	PID constant change flag (Write)	• When the PID Constant Change flag is turned from OFF → ON, the PID constant that is used for the PID calculation is read again, and this value is used to perform calculations. • After the PID constant change is complete, this flag must be turned OFF by the user. • If there is an error in the PID constant (PID Constant OK = 0), the PID calculation value based on the previous PID constant will be used and the operation will continue.	52]
zzzz + 3	S flag (Write)	When the S flag is set to "1", it reverts the output value to its initial value. It performs the following output depending on the relationship between Output Upper Limit Value 17], Output Lower Limit Value 18], and Initial Values 19]. Output Lower Limit Value 18] > Output Upper Limit Value17] …..No output Output Lower Limit Value 18] ≤ Initial Value 19] ≤ Output Upper Limit Value 17] …Outputs Initial Values 19] Output Lower Limit Value 18] ≤ Output Upper Limit Value 17] ≤ Initial Values 19] … ≤ Outputs Output Upper Limit Value 17] Initial Values 19] ≤ Output Lower Limit Value 18] ≤ Output Upper Limit Value 17] ... Outputs Output Lower Limit Value 18] The S flag takes priority over the R Flag.	53]
zzzz + 4	R flag (Write)	When the R flag is set to "1", it clears the output value to 0.	54]
zzzz + 5	D-FREI flag (Write)	"0": Calculate PID without performing integrals or derivatives. "1": Calculate PID using integrals or derivatives.	55]
zzzz + 6	Unused		
zzzz + 7	Unused		
zzzz + 8	PID RUN flag (Read)	• When the FUN 1 detects the startup of the Execution flag 50], 12] through 16] and 20] through 22] will be checked for logical validity and the result will be set to the PID RUN flag 58]. 1 : Valid 0 : Invalid • If the Execution flag 50] startup is detected by the FUN 1 when the PID RUN flag 58] = 1, PID RUN 58] becomes 0 and the PID process will end.	58]
zzzz + 9	PID calculation in progress flag (Read)	• Sets the PID Calculation in Progress flag 59] in the loop in which the FUN 2 calculates the PID to "1," and sets all PID Calculation in Progress flags in other loops to "0."	59]
zzzz + A	PID constant OK flag (Read)	• When the FUN 1 detects the startup of the PID Constant Change flag 52], the PID constants 12] through 16] will be checked for logical validity and the result will be set in the PID Constant OK Flag 60].	60]
zzzz + B	Upper limit over flag (Read)	• If the PID output value calculated by the FUN 2 is greater than the output upper limit UL 17], the Upper Limit Over flag 61] will be set to "1."	61]
zzzz + C	Lower limit over flag (Read)	• If the PID output value calculated by the FUN 2 is greater than the output lower limit LL 18], the Lower Limit Over flag 62] will be set to "1."	62]
zzzz + D	FUN 2 error flag (Read)	When there is an error in the output upper limit value 17], output lower limit value 18], or in any of the bit patterns 23] through 25] during FUN 2 processing, the FUN 2 Error 63] will be set to "1." The cause of the error is set in error code 2 2]. PID calculation will still be executed even if an error is generated. If there is no error, the FUN 2 Error flag 63] = 0. Nothing will be set to error code 2 2].	63]
zzzz + E	Unused		
zzzz + F	Unused		

(2) PID operation execution format

(Example 1) Using two loops with both loops set as TZ = 2 (× 20 ms)

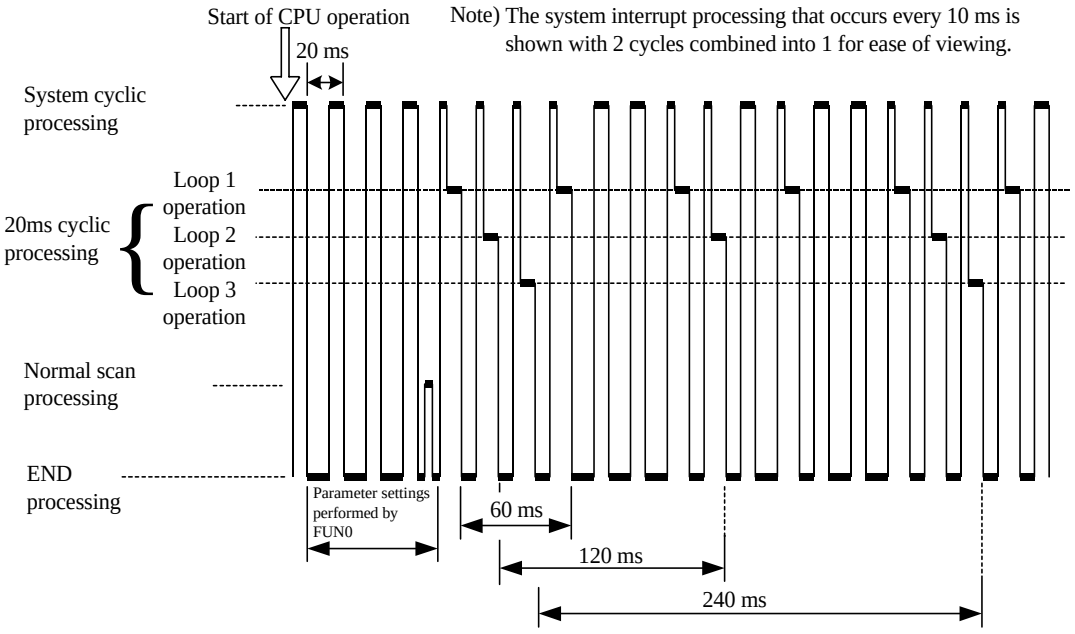


Legend
a: FUN0 processing
b: FUN1 processing
c: FUN2 processing
d: Other cyclic interrupt processing

PID Operation Execution Control (2 loops)

(Example 2) Using three loops set as follows:

Loop1: TZ = 3 (× 20 ms)
Loop2: TZ = 6 (× 20 ms)
Loop3: TZ = 12 (× 20 ms)

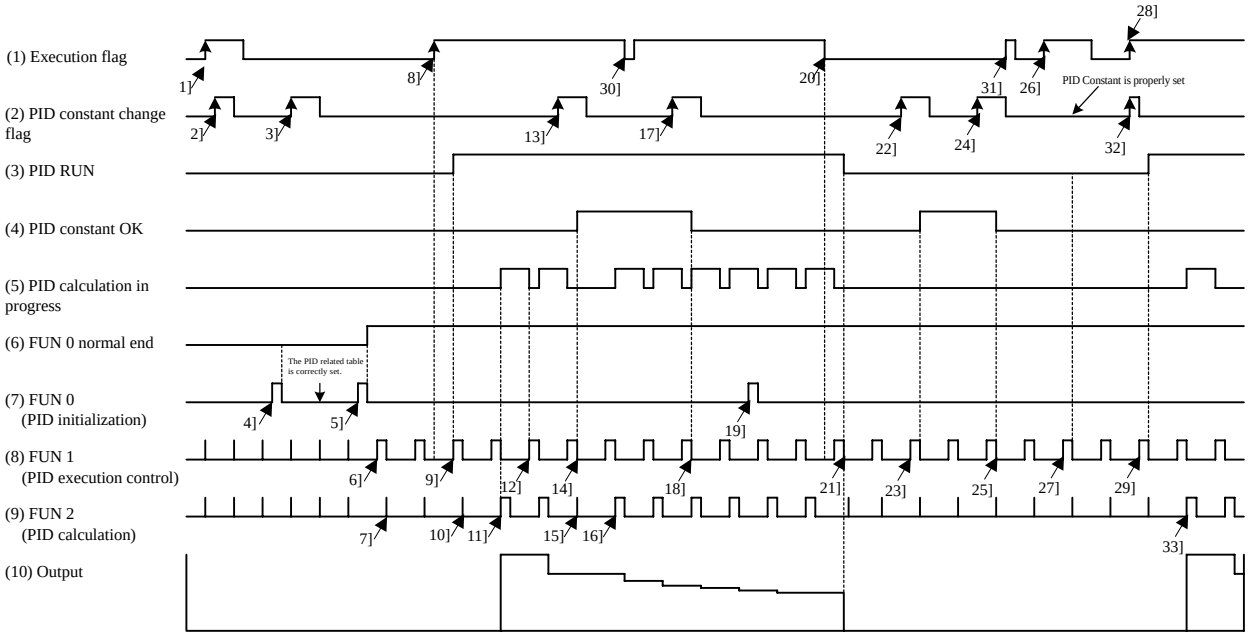


PID Operation Execution Control (3 loops)

(3) PID operation timing chart

(a) Timing chart example 1

The following timing chart shows the operation of the PID RUN flag, PID constant OK flag, PID calculation in progress flag, FUN 0, FUN 1, and FUN 2 when the execution flag and PID constant change flag is turned from ON to OFF in a single loop.



Description of timing chart example 1

1] This is ignored since FUN 0 is not executed properly even when the execution flag, 2] and 3] of the PID constant change flag are turned on.

4] No process will be performed even if FUN 1 is executed because there was an error in the PID related table during FUN 0 processing.

5] 6] FUN 1 processing will be started because the FUN 0 processing ended normally. 7] FUN2 will not perform PID calculations because the execution flag is off.

8] 9] FUN 1 will detect turning on of the execution flag and will check the PID constant. Since it is normal, the PID constant will be calculated and the PIDRUN flag will be turned on.

10] The PID calculation of FUN 2 will not be performed on the first scan, so it will start with 11] FUN 2.

11] FUN 2 will turn the PID calculation in progress flag before calculating the PID. 12] FUN 1 will turn off the PID calculation in progress flag.

13] 14] FUN 1 checks the PID constant when the PID constant change flag is turned on. Since it is normal, the PID constant OK flag is turned on and the PID constant will be changed.

15] Since PID calculations are not performed in FUN 2, PID calculations will be performed from 16] FUN 2 according to the PID constant after it has been changed.

17] When the PID constant change flag was turned on, 18] FUN 1 checked the PID constant. An error was detected, so the PID constant OK flag is turned off. The PID constant flag will not be changed.

19] FUN 0 will be ignored when re-executed during PID operation.

20] Since 21] FUN 1 detected turning off of the execution flag, the PIDRUN flag will be turned off and the output will be set to 0.

21] Since 23] FUN 1 detected turning on of the PID constant change flag when the execution flag was off, the PID constant will be checked. Since it is valid, the PID constant will be changed and the PID constant OK flag will be turned on.

24] Since 25] FUN 1 detected turning on of the PID constant change flag when the execution flag was off, the PID constant will be checked. Since there was an error, the PID constant OK flag will be turned OFF.

26] 27] FUN 1 will detect turning on of the execution flag and check the PID constant. Since an error was detected, the PIDRUN flag will be turned off.

28] Since 29] FUN 1 detected turning on of both the execution flag and the 32] PID constant change flag simultaneously, turning on of the 32] PID constant change flag will be ignored. 29] FUN 1 checks the PID constant, and since it is normal, the PIDRUN flag will be turned on. PID calculation will be started from 33] FUN 2.

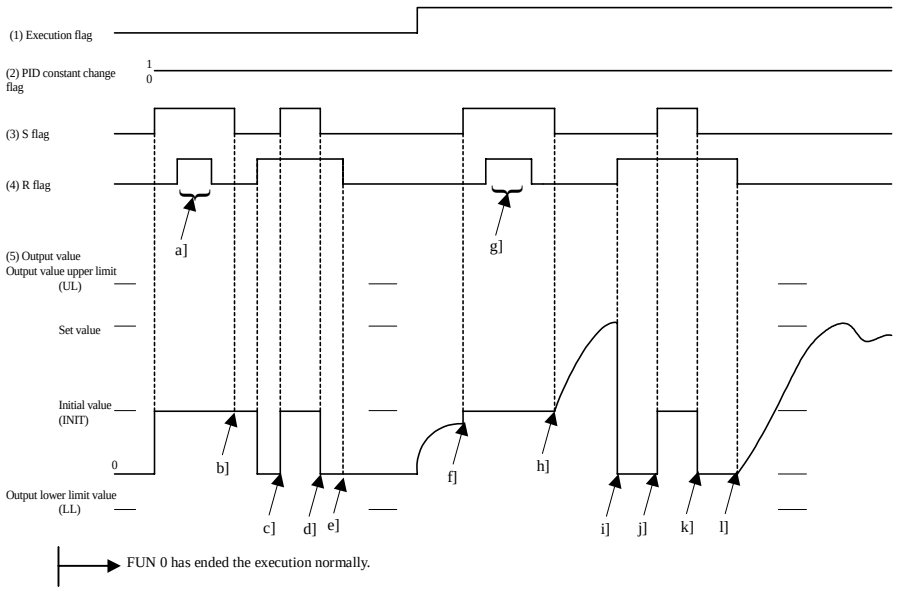
30] 31] If the execution flag turns from on to off in a timing such that the cyclic interrupt cannot detect it, it will be ignored.

(b) Timing chart example 2

The following is an operation timing chart in respect to the S flag and R flag (bumpless).

S flag…..Sets the output value to the initial value.

R flag…..Sets the output value to 0.



a] g] The output value is still INIT because the S flag takes priority.

b] e] The output value is retained since the execution flag is off.

c] j] The output value is set to INIT because the S flag takes priority.

d] k] The output value will be 0 wince the R flag is on when the S flag turns off.

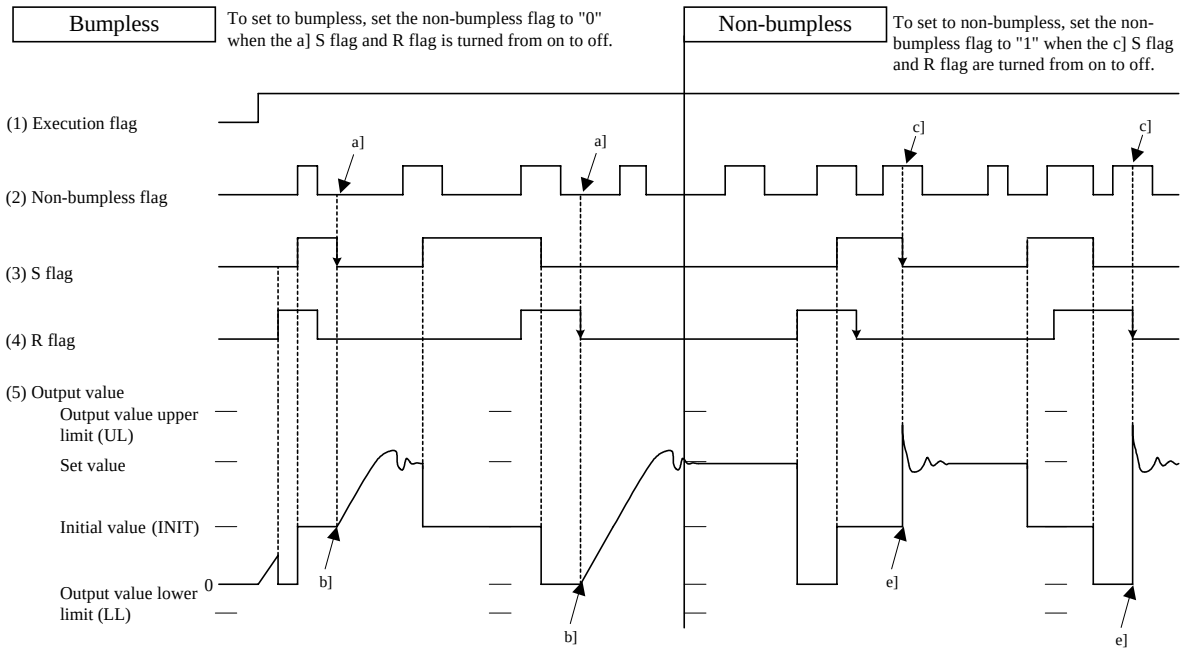
f] The output value will be INTT.

h] l] The output value will continuously move toward the target value since the execution flag is on and bumpless.

i] The output value will be 0.

(c) Timing chart example 3

Bumpless and non-bumpless

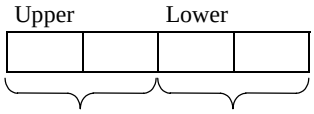


b] When the S flag and R flag turn from on to off, the output value will continuously change to move toward the set value.

e] When the S flag and R flag turn from on to off, the output value will abruptly change to move toward the set value.

(4) PID command error code details

- Error codes are shown using a 4-digit hexadecimal value.



The last 2 digits show the cause of the error.

Shows the loop number.
In the case of H00, it is an error that has no relation to loop numbers.
In the case of H01 through H04, there is an error in the loop shown in the loop number.

(a) Error code 0

The error codes generated in FUN 0 processing and some parts of FUN 1 processing are set in error code 0.
If there is no error, the previous status will be maintained.

Error code	Contents and cause	Corrective action	Remarks
0001	The FUN 0 was executed again after the FUN 0 had been successfully completed.	Do not execute the FUN 0 after it has been executed successfully.	"FUN 0 normal completion 5]" maintains the previous value.
0002	The number of loops 3] is 0.	Set the number of loops 3] to a value between the range of 1 to 64.	
0003	The number of loops 3] exceeds 65.	Set the number of loops 3] to a value between the range of 1 to 64.	
0004	The PID control table exceeds the maximum number of WR.	Change the head of PID management table or the number of loops 3] so that the maximum number of WR is not exceeded.	The size of the PID management table will change. If the number of loops 3] exceeds the suffix of the I/O, "FUN 0 normal completion 5]" will maintain the previous value.
××05	The word table of loop ×× exceeds the maximum number of WR.	Set the number in the WR for the loop 4] again.	The size of the bit table is 16 bits per loop.
××06	The bit table of loop ×× exceeds the maximum number of R.	Set the bit number for R 11] again.	The size of the bit table is 16 bits per loop.
××07	The output upper limit value 17] in loop ×× is outside of range.	Set the output upper limit value 17] to a value between –32,767 and 32,767.	
××08	The output lower limit value 18] in loop ×× is outside of range.	Set the output lower limit value 18] to a value between –32,767 and 32,767.	
××09	The initial value 19] in loop ×× is outside of range.	Set the initial value 19] to a value between –32,767 and 32,767.	
××0A	There is an error in the size relationship between the output upper limit value 17], output lower limit value 18], and initial value 19].	Perform settings so that the output lower limit value 18] (initial value 19] (output upper limit value 17] is met.	
××0B	The set value bit pattern 23] in loop ×× is outside of range.	Set the set value bit pattern 23] to a value between 1 and 4.	
××0C	The measured value bit pattern 24] in loop ×× is outside of range.	Set the measured value bit pattern 24] to a value between 1 and 4.	
××0D	The output value bit pattern 25] in loop ×× is outside of range.	Set the output value bit pattern 25] to a value between 1 and 4.	
0020 (Note)	The FUN 1 is being executed when the FUN 0 is not successfully completed.	Do not run the FUN 1 until the FUN 0 is successfully executed.	Set to the error code 0 specified by the (S) in the FUN 1 (S).
0021 (Note)	The S in the FUN 1 (S) is different from the S in the FUN 0 (S) of the PID management table.	Set the same WR for the S in the FUN 1(S) and the S in the FUN 0 (S).	Set to the error code 0 specified by the (S) in the FUN 1 (S).

(Note) Error codes 0020 and 0021 will over-write the errors generated previously (0001 to XX0D). Therefore, execute the FUN 1 after verifying that the FUN 0 is successfully executed.

(b) Error code 1

The error code generated in the FUN 1 process is set in error code 1. If there is no error, the previous condition is maintained.

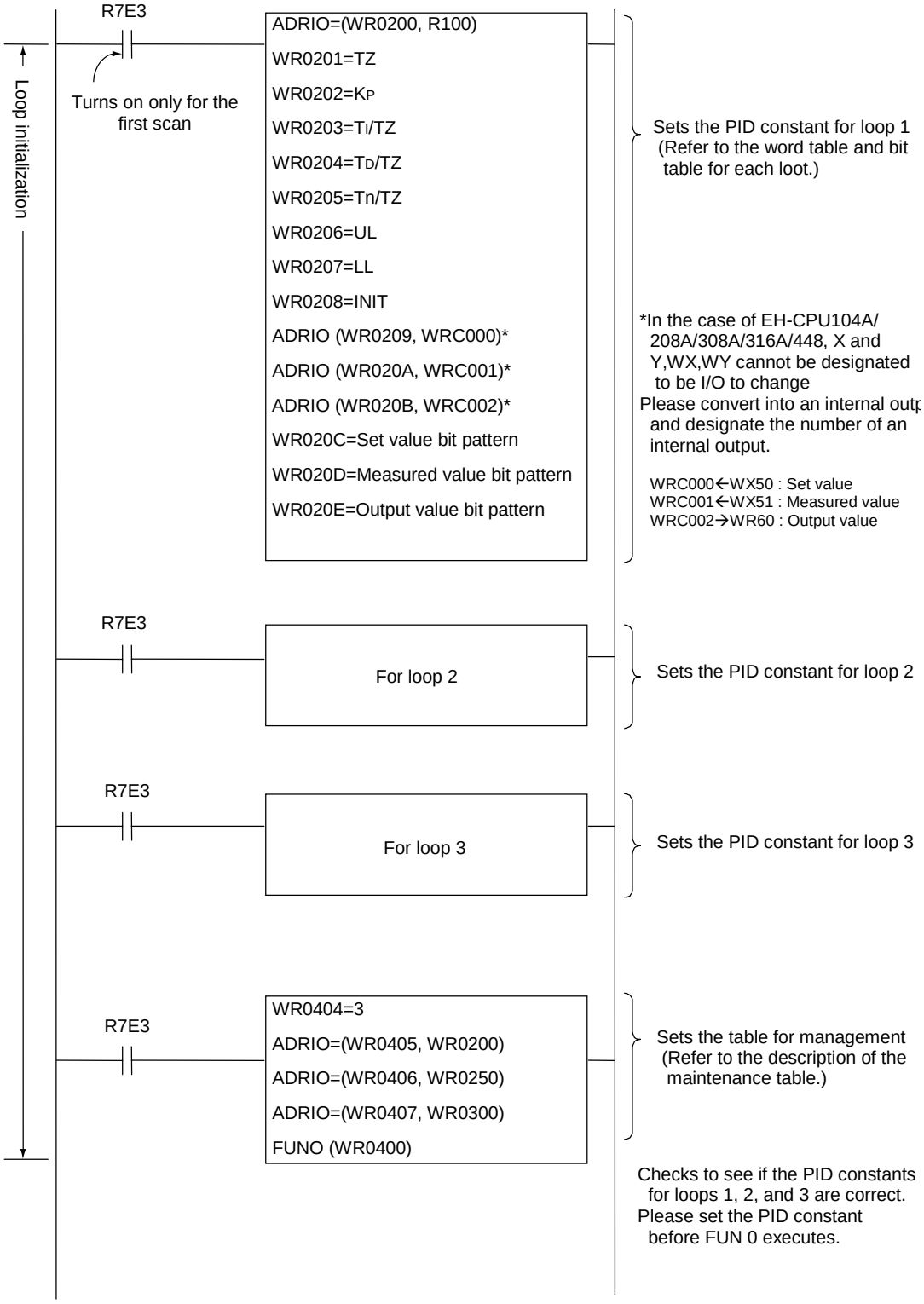
Error code	Contents and cause	Corrective action	Remarks
0020	The FUN 1 is being executed when the FUN 0 is not successfully completed.	Do not run the FUN 1 until the FUN 0 is successfully executed.	Set to the error code 0 specified by the (S) in the FUN 1 (S).
0021	The S in the FUN 1 (S) is different from the S in the FUN 0 (S) of the PID management table 1].	Set the same WR number for the S in the FUN 1(S) and the S in the FUN 0 (S).	Set to the error code 0 specified by the (S) in the FUN 1 (S).
××22	There is an error in the set value I/O number 20] in loop ××.	Set the set value I/O number 20] using the ADRIO command.	These are errors that may be generated when the Execution flag starts up.
××23	There is an error in the measured value I/O number 21] in loop ××.	Set the measured value I/O number 21] using the ADRIO command.	
××24	There is an error in the output value I/O number 22] in loop ××.	Set the output value I/O number 22] using the ADRIO command.	
××25	The sampling time 12] of loop ×× is out of range.	Set the sampling time 12] to a value within the range of 1 to 200.	These are errors that may be generated when the Execution flag starts up or when the PID Constant Change flag starts up.
××26	The sampling time 12] of loop ×× is not a multiple of the number of loops 3].	Set the sampling time 12] so that it becomes a multiple of the number of loops 3].	
××27	The proportional gain 13] of loop ×× is out of range.	Set the proportional gain 13] to a value within the range of –1,000 to 1,000.	
××28	The integral constant 14] of loop ×× is out of range.	Set the integral constant 14] to a value within the range of 1 to 32,767.	
××29	The derivative constant 15] of loop ×× is out of range.	Set the derivative constant 15] to a value within the range of 1 to 32,767.	
××2A	The derivative delay constant 16] of loop ×× is out of range.	Set the derivative delay constant 16] to a value within the range of 1 to 32,767.	
××30	There is an error in the size relationship between the output lower limit value 18] and output upper limit value 17] in loop ××.	Set the values so that the output lower limit value 18] ≦ output upper limit value 17] is satisfied.	There is a possibility that this error is generated when the S flag 53] is turned ON while the PID RUN flag 58] is OFF.
××31	There is an error in the output value I/O number 22] in loop ××.	Set the output value I/O number 22] using the ADRIO command.	There is a possibility that these errors are generated when the S flag 53] or R flag 54] is turned on while the PID RUN flag 58] is OFF.
××32	The output value bit pattern 25] in loop ×× is outside of range.	Set the output value bit pattern 25] to a value between 1 and 4.	

(c) Error code 2

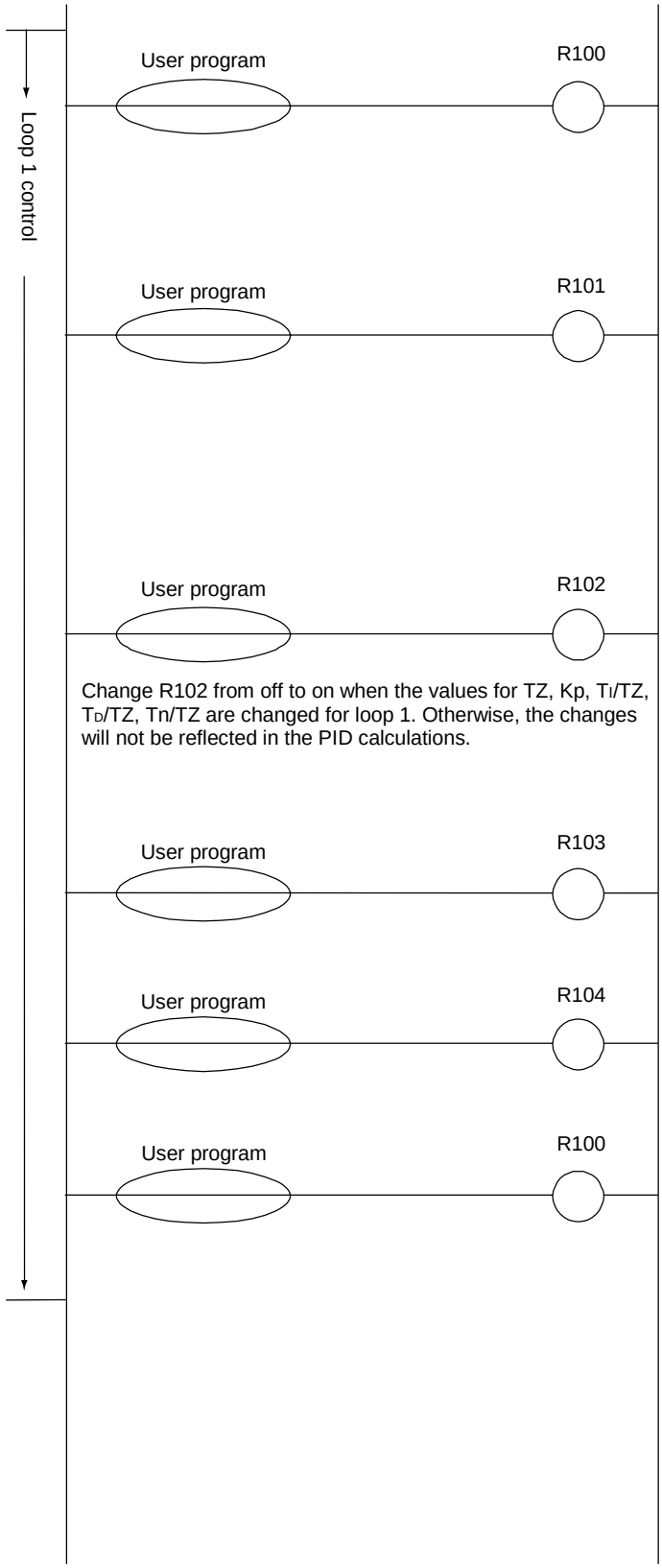
Error code	Contents and cause	Corrective action	Remarks
0040			
××41	The set value bit pattern 23] in loop ×× is outside of range.	Set the set value bit pattern 23] to a value between 1 and 4.	When the bit pattern is outside of range, the process will continue based on “4. Do not convert.”
××42	The measured value bit pattern 24] in loop ×× is outside of range.	Set the set value bit pattern 24] to a value between 1 and 4.	
××43	The output value bit pattern 25] in loop ×× is outside of range.	Set the output value bit pattern 25] to a value between 1 and 4.	
××44	There is an error in the size relationship between the output lower limit value 18] and output upper limit value 17] in loop ××.	Set the values so that the output lower limit value 18] ≦ output upper limit value 17] is satisfied.	If there is a size relationship error, the process will continue but there will be no output.

Program example

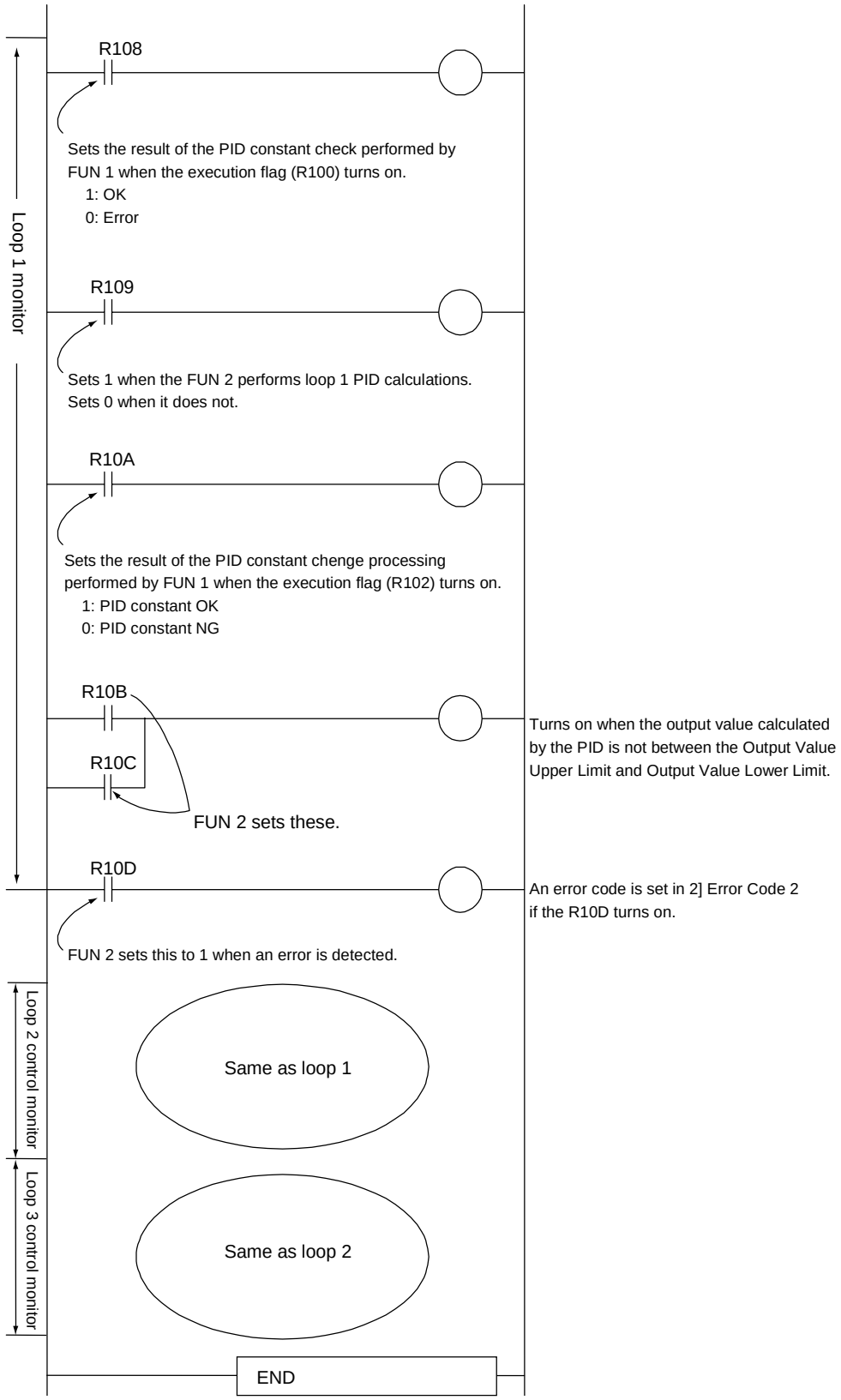
This program is an example comprised of three loops. This program also rewrites the PID constant every time the CPU starts a RUN process.



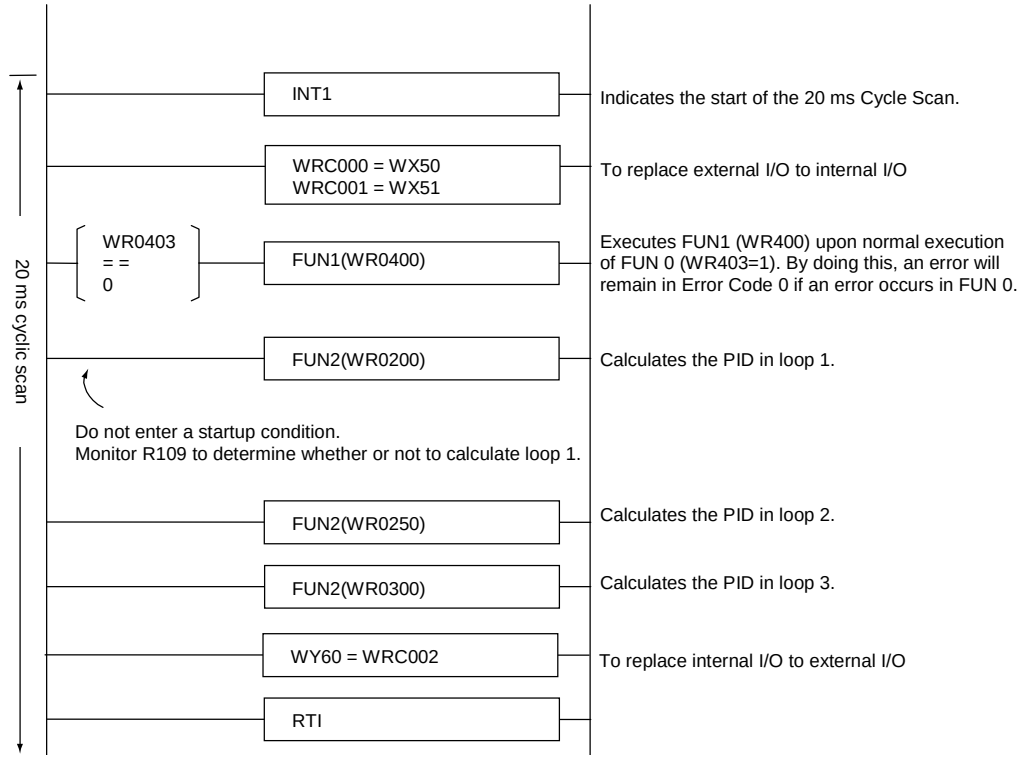
Program example



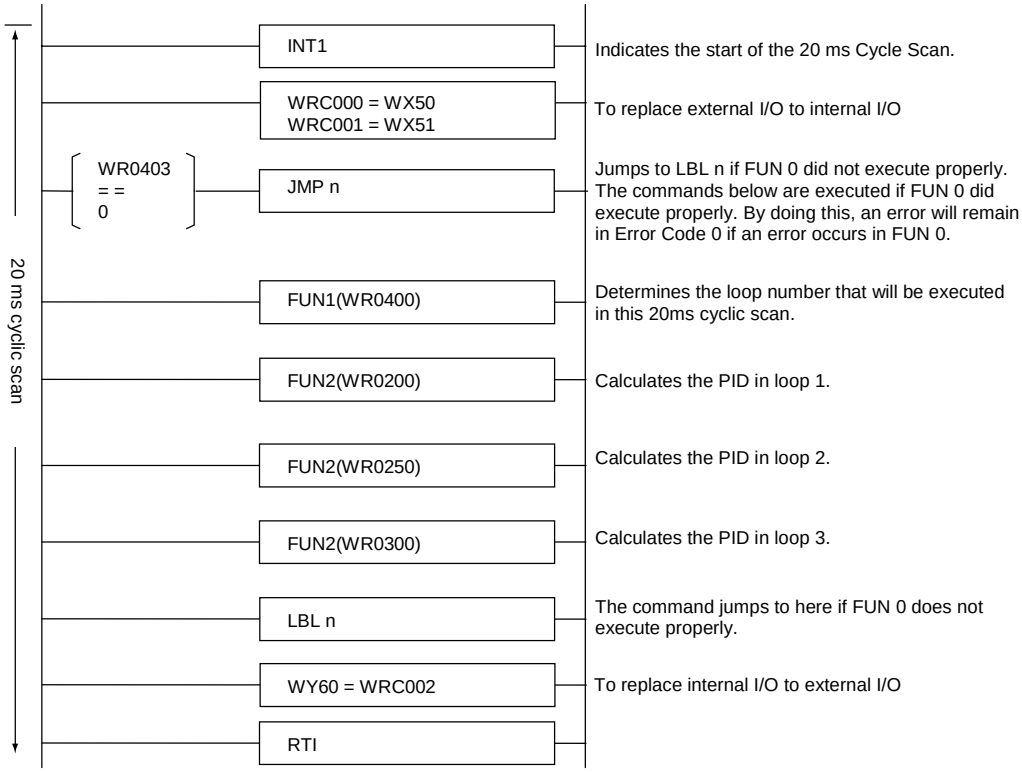
Program example



Program example



The program on this page can also be as shown below.



* Please take note that INT2 is used for the EH-CPU448 and EnhanceCPU.

Item number	Fun commands-4	Name	Process stepping

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 4 (s)	DER	ERR	SD	V	C							
* (IFR (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					208	←	191 370	←	398	←	
	Condition			Steps								
FUN 4 (s) * (IFR (s))	—			3								

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument							○						s uses up to s+3.
														s+3 is used by the system.

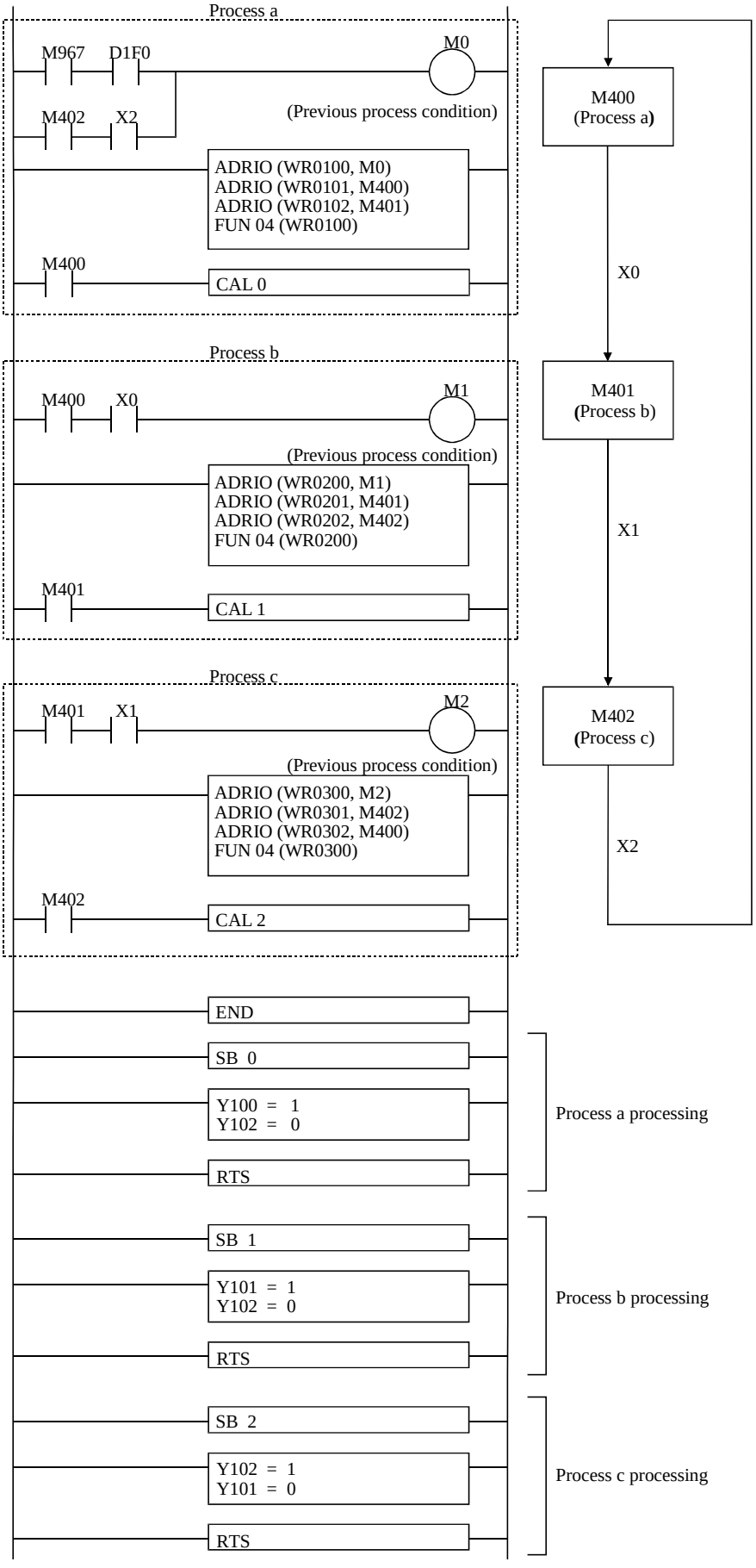
Function

s	Previous process condition I/O number
s+1	Process set I/O number
s+2	Next process (clear condition) I/O number
s+3	Used by the system

- When the I/O designated by s (previous process) switches on, the s+1 (process set) switches on and the state is retained. (The previous process condition is triggered by edge.)
- When the I/O designated by s+2 (next process) switches on, the s+1 (process set) is switched off. (The next process is triggered by level.)
- When s (previous process) and s+2 (next process) are both on, the s+2 (next process) has the priority.
- The user should designate output for each process, if necessary.

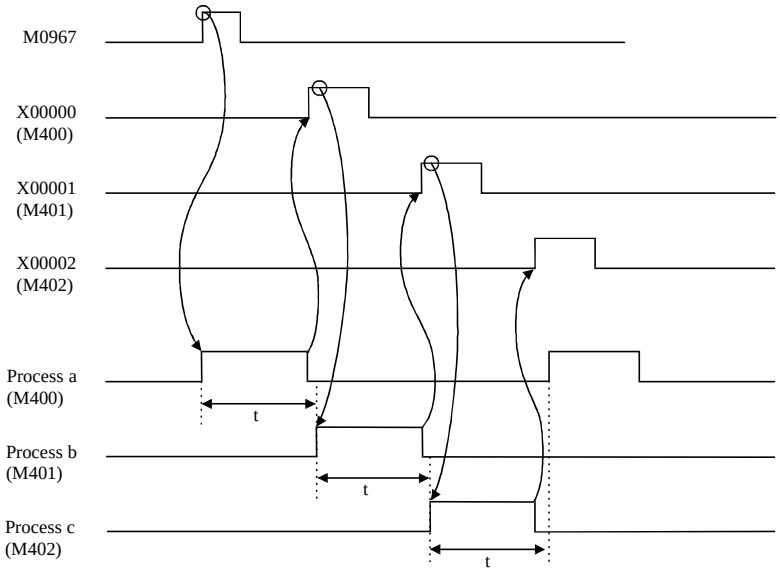
* ( ) indicates the display when the LADDER EDITOR is used.

Program example



Cautionary notes

- Set the actual R, L and M address for the parameters s through s+2 using the ADRIO command.
- If the areas designated by s to s+2 overlap, if s+1, s+2 or s+3 falls out of range, DER will be equal to “1” and the command will not be processed.
- Do not designate the same I/O for arguments of different processes, since the action of the current process is levelled by the previous process.
- Each process requires at least one scan time.



t: One scan time is necessary

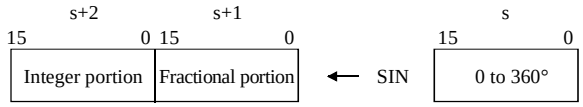
- In the program example described previously, the external I/O (X, Y) are used as switch signals of a process; thus, the time for performing I/O refresh (i.e., at least one scan period) is required for each process.

Item number	Fun commands-5	Name	SIN function

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 10 (s)	DER	ERR	SD	V	C							
* (SIN (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					39	←	40 87	←	105	←	
	Condition			Steps								
FUN 10 (s) * (SIN (s))	—			3								

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument							○						s uses up to s+2.

Function



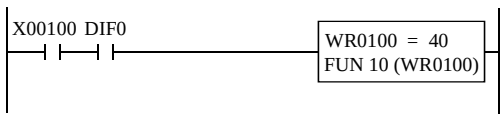
- Calculates the SIN value using the unsigned binary value designated using s as the argument, and sets the integer and fractional portions of the result in s+2 and s+1, respectively.
- The SIN value is indicated in a binary value, and negative values are indicated in two's complements.
- If the calculation is performed normally, DER is equal to "0."
- The fractional data is the value obtained by multiplying the actual value by 65,535.

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- The argument is given in degrees in the range $0 \leqq s \leqq 360$. Any other value will equal DER to "1" and the operation will not be performed.
- If s+1 and s+2 exceed the maximum value for the I/O number, DER is equal to "1" and the operation will not be performed.

Program example




```
LD      X00100
AND     DIF0
[
WR0100 = 40
FUN 10 (WR0100)
]
```

Program description

- An angle of 40° is set in WR0100.
- SIN operation is performed at the leading edge of X00100, and the fractional portion of the result is set in WR0101 and the whole number portion is set in WR0102 as binary values.
  Execution results: WR0102=H0000, WR0101=HA48E, WR0100=H0028

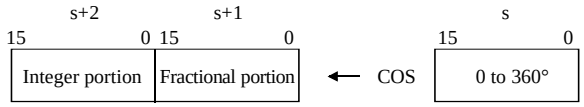
Item number	Fun commands-6	Name	COS function

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 11 (s)	DER	ERR	SD	V	C							
* (COS (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps										
	Condition	Steps									
FUN 11 (s)	—	3	40	←	41 88	←	108	←			
* (COS (s))											

Usable I/O		Bit				Word				Double word				Other	
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM	Constant		
s	Argument							○						s uses up to s+2.	

Function



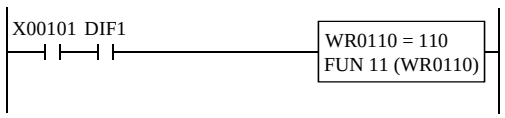
- Calculates the COS value using the unsigned binary value designated by s as the argument, and sets the integer and fractional portions of the result in s+2 and s+1, respectively.
- The COS value is indicated in a binary value, and negative values are indicated in two's complements.
- If the calculation is performed normally, DER is equal to "0."
- The fractional data is the value obtained by multiplying the actual value by 65,535.

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- The argument is given in degrees in the range $0 \leqq s \leqq 360$. Any other value will equal DER to "1" and the operation will not be performed.
- If s+1 and s+2 exceed the maximum value for the I/O number, DER is equal to "1" and the operation will not be performed.

Program example




```
LD      X00101
AND     DIF1
[
WR0110 = 110
FUN 11 (WR0110)
]
```

Program description

- An angle of 110° is set in WR0110.
- COS operation is performed at the leading edge of X00101, and the fractional portion of the result is set in WR0111 and the whole number portion is set in WR0112 as binary values.
  Execution results: WR0112=HFFFF, WR0111=HA871, WR0110=H006E

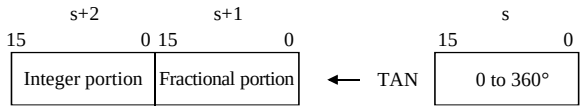
Item number	Fun commands-7	Name	TAN function

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 12 (s)	DER	ERR	SD	V	C							
* (TAN (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps										
	Condition	Steps									
FUN 12 (s)	—	3	40	←	41 88	←	109	←			
* (TAN (s))											

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument							○						s uses up to s+2.

Function



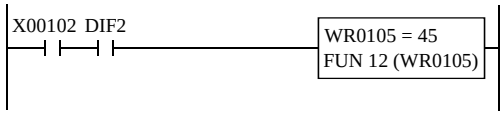
- Calculates the TAN value using the unsigned binary value designated by s as the argument, and sets the integer and fractional portions of the result in s+2 and s+1, respectively.
- The TAN value is indicated in a binary value, and negative values are indicated in two's complements.
- If the calculation is performed normally, DER is equal to "0."
- The fractional data is the value obtained by multiplying the actual value by 65,535.

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- The argument is given in degrees in the $0 \leqq s \leqq 360$. When s is equal to 90 or s is equal to 270, H7FFF and HFFFF are set for s+2 and s+1, respectively. If s falls outside the range, DER is equal to "1" and the operation will not be performed.
- If s+1 and s+2 exceed the maximum value for the I/O number, DER is equal to "1" and the operation will not be performed.

Program example




```
LD      X00102
AND     DIF2
[
WR0105 = 45
FUN 12 (WR0105)
]
```

Program description

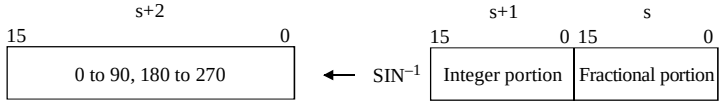
- An angle of 45° is set in WR0105.
- TAN operation is performed at the leading edge of X00102, and the fractional portion of the result is set in WR0106 and the whole number portion is set in WR0107 as binary values.
  Execution results: WR0107=H0001, WR0106=H0000, WR0105=H002D

Item number	Fun commands-8	Name	ARC SIN function

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 13 (s)	DER	ERR	SD	V	C							
* (ASIN (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					62	←	63 123	←	174	←	
	Condition			Steps								
FUN 13 (s) * (ASIN (s))	—			3								

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument (fractional portion)							○						s uses up to s+2.
s+1	Argument (integer portion)							○						

Function



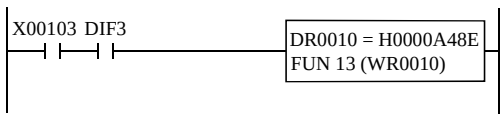
- Calculates the SIN^{-1} value using the unsigned binary value designated by s (fractional portion) and s+1 (integer portion) as the argument, and outputs s+2.
- The SIN^{-1} value is described in degrees in the range of 0 to 90 and 180 to 270.
- If the calculation is completed normally, DER is equal to "0."
- The fractional data is the value obtained by multiplying the actual value by 65,535.

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- When the argument | s+1.s | > 1, DER is equal to "1" and operation will not be performed.
- When s+1 and s+2 exceed the maximum value for the I/O number, DER is equal to "1" and operation will not be performed.

Program example




```
LD      X00103
AND     DIF3
[
DR0010 = H0000A48E
FUN 13 (WR0010)
]
```

Program description

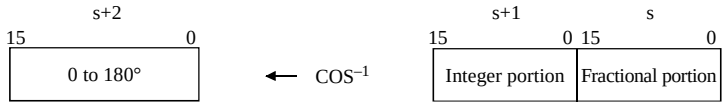
- Set data in DR0010 (WR0010, WR0011).
- SIN^{-1} operation is performs at the leading edge of X00103, and the result is set in WR0012 as a binary value.
  Execution results: WR0012=H0028, WR0011=H0000, WR0010=HA48E

Item number	Fun commands-9	Name	ARC COS function

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 14 (s)	DER	ERR	SD	V	C							
* (ACOS (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					63	←	64 125	←	177	←	
	Condition			Steps								
FUN 14 (s)	—			3								
* (ACOS (s))												

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument (fractional portion)							○						s uses up to s+2.
s+1	Argument (integer portion)							○						

Function



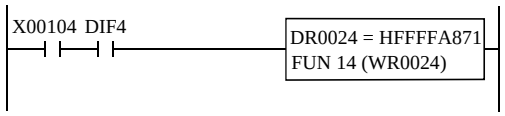
- Calculates the COS^{-1} value using the unsigned binary value designated by s (fractional portion) and s+1 (integer portion) as the argument, and outputs s+2.
- The COS^{-1} value is described in degrees in the range of 0 to 180.
- If the calculation is completed normally, DER is equal to "0."
- The fractional data is the value obtained by multiplying the actual value by 65,535.

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- When the argument | s+1.s | > 1, DER is equal to "1" and operation will not be performed.
- When s+1 and s+2 exceed the maximum value for the I/O number, DER is equal to "1" and operation will not be performed.

Program example




```
LD      X00104
AND     DIF4
[
DR0024 = HFFFFA871
FUN 14 (WR0024)
]
```

Program description

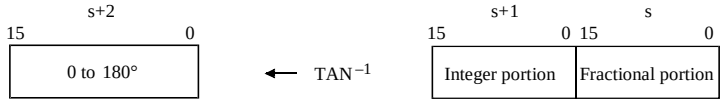
- Set data in DR0024 (WR0024, WR0025).
- COS^{-1} operation is performs at the leading edge of X00104, and the result is set in WR0026 as a binary value.
  Execution results: WR0026=H006E, WR0025=HFFFF, WR0024=HA871

Item number	Fun commands-10	Name	ARC TAN function

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 15 (s)	DER	ERR	SD	V	C							
* (ATAN (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					41	←	42 91	←	194	←	
	Condition			Steps								
FUN 15 (s)	—			3								
* (ATAN (s))												

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument (fractional portion)							○						s uses up to s+2.
s+1	Argument (integer portion)							○						

Function



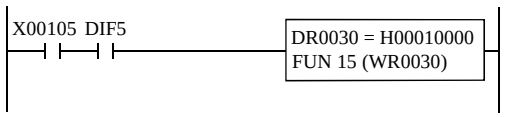
- Calculates the TAN^{-1} value using the unsigned binary value designated by s (fractional portion) and s+1 (integer portion) as the argument, and outputs s+2.
- The TAN^{-1} value is described in degrees in the range of 0 to 90 and 180 to 270.
- If the calculation is completed normally, DER is equal to "0."
- The fractional data is the value obtained by multiplying the actual value by 65,535.

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- When s+1 and s+2 exceed the maximum value for the I/O number, DER is equal to "1" and operation will not be performed.

Program example




```
LD      X00105
AND     DIF5
[
DR0030 = H00010000
FUN 15 (WR0030)
]
```

Program description

- Set data in DR0030 (WR0030, WR0031).
- TAN^{-1} operation is performs at the leading edge of X00105, and the result is set in WR0032 as a binary value.
  Execution results: WR0032=H002D, WR0031=H0001, WR0030=H0000

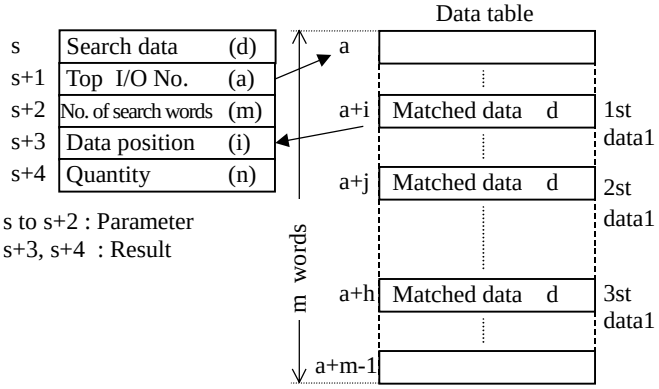
Item number	Fun commands-11	Name	Data search †

Ladder format	Condition code					Processing time (μs)		Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU5**		
FUN 20 (s)	DER	ERR	SD	V	C			
* (DSRCH (s))	↕	●	●	●	●	Ave	Max	The test condition is that searched data is 2 out of 10 words table.

Command format	Number of steps			
	Condition	Steps	86.2	86.2
FUN 20 (s)	—	3		
* (DSRCH (s))				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Search data							○						
s+1	Top I/O No. of data table							○						The real address is set.
s+2	No. of search words							○						s to s+4 are used.

Function



s to s+2 : Parameter
s+3, s+4 : Result

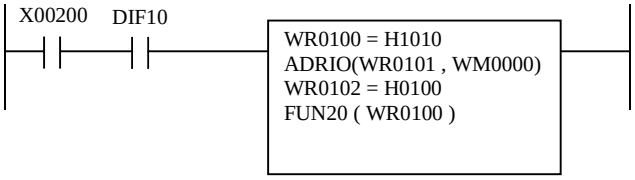
- Data specified in "s" will be searched in data specified in "s+1" and "s+2". If data is searched, the relative address from the top I/O and the quantity will be stored in "s+3" and "s+4".
- When no matched data is found, 0 is set in "s+3"
- Address in "s+1" is specified by ADRIO command.
- If operation is performed properly, DER = 0.

*Command name in ( ) is indications of the ladder editor.

Cautionary notes

- Address in "s+1" is specified to WR, WL or WM by ADRIO command., otherwise nothing executed with DER="1".
- If specified areas by "s" to "s+4" are over lapped, nothing executed with DER="1".
- If specified addresses in "s+1" and "s+2" do not exist in CPU, nothing executed with DER="1".

Program example




```
LD  X00200
AND  DIF10
 [
WR0100 = H1010
ADRIO ( WR0101,WM0000 )
WR0102 = H0100
FUN20 ( WR0100 )
 ]
```

Program description

- Data H1010 is searched for among 256 word of WM0000.
- Relative address and data quantity will be stored in WR103 and WR104.

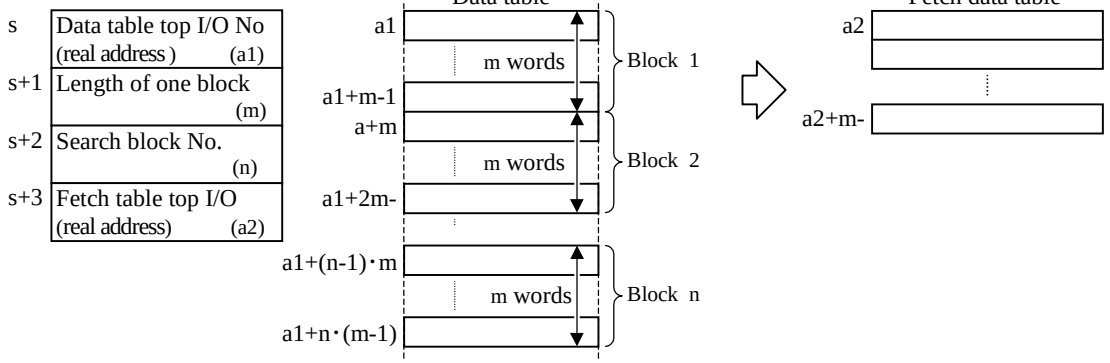
†: Supported by EH-CPU 516/548 only.

Item number	Fun commands-12	Name	Table search†

Ladder format	Condition code					Processing time (μs)		Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU5**		
FUN 21 (s)	DER	ERR	SD	V	C			
* (TSRCH (s))	↕	●	●	●	●	Ave	Max	The test condition is that searched data block is 1 block, 2 words in 10th block.
Command format	Number of steps							
FUN 21 (s)	Condition		Steps			104.2	104.2	
* (TSRCH (s))	—		3					

Usable I/O		Bit				Word				Double word				Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM	Constant	
s	Data table top I/O No.							○						The real address is set.
s+1	Length of one No.							○						s to s+3 are used.
s+2	Search block No.							○						
s+3	Fetch table top I/O No.							○						The real address is set.

Function



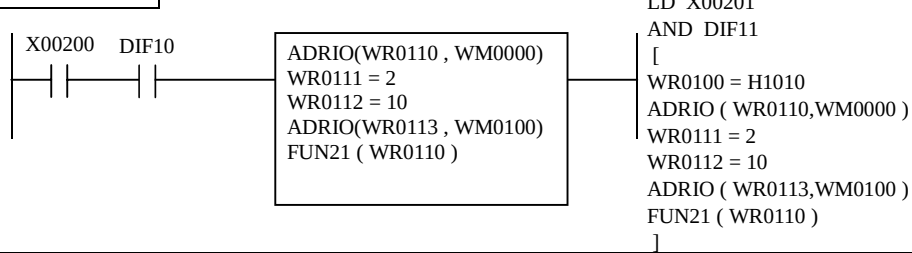
- Data will be searched (taken) in data table specified in "s" and "s+1", and stored in the address specified in "s+3".
- Addresses in "s+1" and "s+3" are specified by ADRIO command.
- If operation is performed properly, DER = 0.

*Command name in ( ) is indications of the ladder editor.

Cautionary notes

- Addresses in "s+1" and "s+3" are specified to WR, WL or WM by ADRIO command, otherwise nothing executed with DER=”1”.
- If specified areas by "s" and "s+3" are over lapped, nothing executed with DER="1".
- If specified addresses in "s" or "s+3" does not exist in CPU, nothing executed with DER="1".

Program example



Program description

- 10th data block will be taken out of data table from WM0000, which consist of 2 word block each, and stoed in WM0100.

†: Supported by the EH-CPU 516/548 only.

Item number	FUN command-13	Name	Check code calculation†

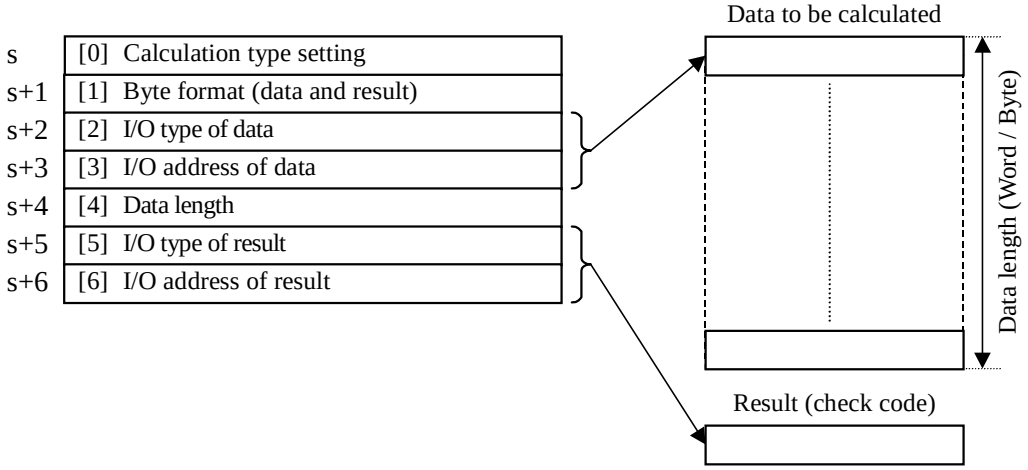
Ladder format	Condition code					Processing time				Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU5**		Other CPU		
FUN 22 (s)	DER	ERR	SD	V	C	Ave.	Max.	-	-	
	⇕	●	●	●	●	783.2	←	–	–	

Command format	Number of steps	
	Condition	Steps
FUN 22 (s)	-	3

Usable I/O		Bit				Word				Double word				Others
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, D M		
s	Starting I/O							○						s to s+6

Function

・This command creates check code to be attached to serial communication message frame.
・Calculation type is specified in the parameter "s".
・Byte format (high or low byte) is specified in the parameter "s+1".
・Data address and data length are specified in "s+2", "s+3" and "s+4".
・Result data address is specified in "s+5" and "s+6".



†: Supported by the EH-CPU 516/548 only.

Function

[0] Calculation type setting

Calculation type to be selected from 7 types as follwos.

Setting	Calculation typpe	Result (Check code)	
H0000	(B1) + (B2) + ··· +(Bn)	Byte	(ex. 12)
H0001	(B1) + (B2) + ··· +(Bn)	Word	Normal (ex.1234)
H0002	(B1) + (B2) + ··· +(Bn)	Word	Byte swapped (ex.3412)
H0003	(B1) + (B2) + ··· +(Bn)	Word	ASCII converted, normal (ex.3132)
H0004	(B1) + (B2) + ··· +(Bn)	Word	ASCII converted, swapped (ex.3231)
H0005	(W1) + (W2) + ··· +(Wn)	Word	Normal (ex. 1234)
H0006	(W1) + (W2) + ··· +(Wn)	Word	Swapped (ex. 3412)
H0010	{ (B1)xor(B2)} xor·xor(Bn)	Byte	(ex. 12)
H0011	{ (B1)xor(B2)} xor·xor(Bn)	Word	ASCII converted, normal (ex. 3132)
H0012	{ (B1)xor(B2)} xor·xor(Bn)	Word	ASCII converted, swapped (ex.3231)
H0013	{ (W1)xor(W2)} xor·xor(Wn)	Word	Normal (ex. 1234)
H0014	{ (W1)xor(W2)} xor·xor(Wn)	Word	Swapped (ex. 3412)
Others	DATA Error (DER ON)		

[1] Byte format (data and result) :

Calculation starting byte position and result storing position are specified as below in case of byte oriented calculation.

Byte type

Starting Word	(B1)	(B2)
+1	(B3)	(B4)
+2	(B5)	(B6)
	…	…
	…	…
+(m-1)	(Bn-1)	(Bn)

	(B1)
(B2)	(B3)
(B4)	(B5)
…	…
…	…
(Bn)	

Word type

(W1_H)	(W1_L)
(W2_H)	(W2_L)
(W3_H)	(W3_L)
…	…
…	…
(Wn_H)	(Wn_L)

	(W1_H)
(W1_L)	(W2_H)
(W2_L)	(W3_H)
(W3_L)	…
…	(Wn_H)
(Wn_L)	

H : High byte
L : Low byte

Wn : | Wn_H | Wn_L |

<High byte>
Calculation starting byte

<Low byte>
Result storingposition

H00xx : Calculation starts from high byte
H01xx : Calculation starts from low byte
Others : DATA Error (DER ON)

Setting value : H00xx

B	B1	B2
	B3	B4
	…	

W	W1
	W2
	…

Setting value : H01xx

B	—	B1
	B2	B3
	…	

W	—	W1_h
	W1_l	W2_h
	W2_l	…

Hxx00 : Data storing starts from high byte
Hxx01 : Data storing starts from low byte *
Others : Data Error (DER ON)
* If result is WORD, L-byte is stored in H-byte position of the next word as below.

Setting value : Hxx00

B	[1]	-

W	[1]

Setting value : Hxx01

B	-	[1]

W	-	[1]
	[1]	-

- : Existing data
[1] : Result

FUN 22 (s)

Function

[2] I/O type of data :

Type WR:H000A, WL:H000B, WM:H000C

[3] I/O address of data:

I/O address H0000 - HFFFF

[4] Data length :

Byte data : unit is byte (H0000 - HFFFF)
Word data : unit is word (H0000 - HFFFF)

[5] I/O type of result

Type WR:H000A, WL:H000B, WM:H000C

[6] I/O address of result:

I/O address H0000 - HFFFF

Sample program

<Sent data frame> Check code = XOR for each byte and ASCII conversion

STX	Data [0101000500]	C.C.	CR
(02)	(30313031303030353030)	(?)	(0D)

＜Sent data area>

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

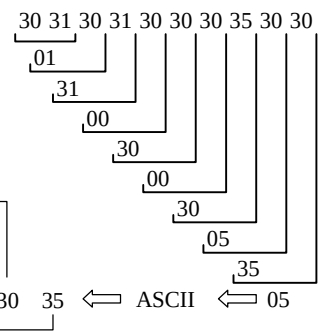
<Sample program>

```
WR0 = H0011      ——— [ 1 ]
WR1 = H0101      ——— [ 2 ]
WR2 = H000C    } ——— [ 3 ]
WR3 = H0000    }
WR4 = 10         ——— [ 4 ]
WR5 = H000C    } ——— [ 5 ]
WR6 = H0005    }
FUN 22 ( WR0 )
```

[1] Calculation type setting (Byte, ASCII, normal) : H0011
[2] Calculation starts from L-byte
Data storing from L-byte : (H0101)
[3] Data address : WM0 (H000C, H0000)
[4] Data length : 10 bytes
[5] Result address : WM5 (H000C, H0005)

＜Result>

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	**3 0**
WM6	**3 5**	0 D



FUN 22 (s)

Item number	FUN command-14	Name	Check code verifying†

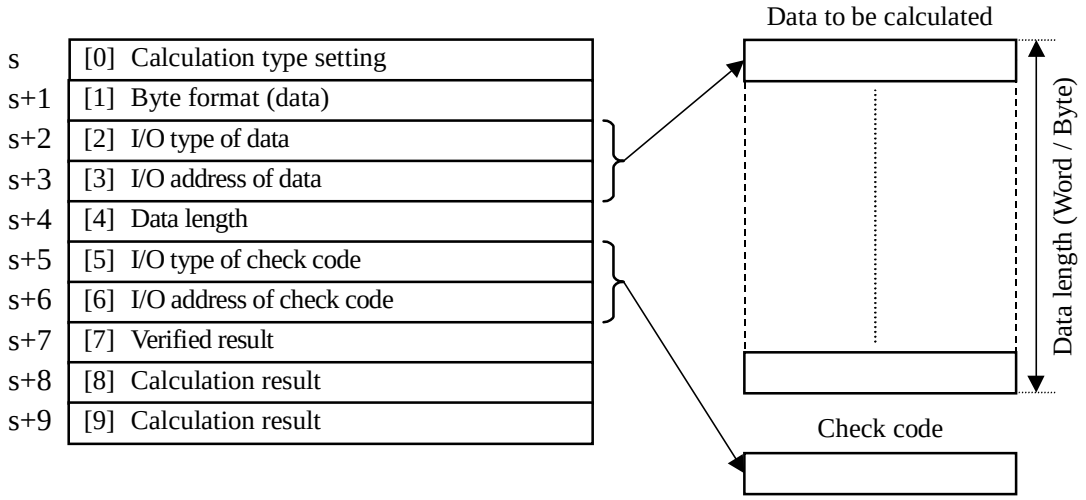
Ladder format	Condition code					Processing time				Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU5**		Other CPU		
FUN 23 (s)	DER	ERR	SD	V	C	Ave.	Max.	-	-	
	⇕	●	●	●	●	790.2	←	−	−	

Command format	Number of steps	
	Conditon	Steps
FUN 23 (s)	-	3

Usable I/O	Bit				Word				Double word				Others
	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s Starting I/O							○						s to s+9

Function

・This command verifies check code attached in received message frame.
・Calculation type is specified in the parameter "s".
・Byte format (high or low byte) is specified in the parameter "s+1".
・Data address and data length are specified in "s+2", "s+3" and "s+4".
・Check code specified in "s+5" and "s+6" is compared with calculated check code, and result is stored in the address specified in "s+7".



†: Supported by the EH-CPU 516/548 only.

Function

[0] Calculation type setting :
Calculation type to be selected from 7 types as follows.

Value	Calculation type	Result (Check code)	
H0000	(B1) + (B2) + ⋯ +(Bn)	Byte	(ex. 12)
H0001	(B1) + (B2) + ⋯ +(Bn)	Word	Normal (ex.1234)
H0002	(B1) + (B2) + ⋯ +(Bn)	Word	Byte swapped (ex.3412)
H0003	(B1) + (B2) + ⋯ +(Bn)	Byte	ASCII converted, normal (ex.3132)
H0004	(B1) + (B2) + ⋯ +(Bn)	Byte	ASCII converted, swapped (ex.3231)
H0005	(W1) + (W2) + ⋯ +(Wn)	Word	Normal (ex. 1234)
H0006	(W1) + (W2) + ⋯ +(Wn)	Word	Swapped (ex. 3412)
H0010	{ (B1)xor(B2)} xor·xor(Bn)	Byte	(ex. 12)
H0011	{ (B1)xor(B2)} xor·xor(Bn)	Byte	ASCII converted, normal (ex. 3132)
H0012	{ (B1)xor(B2)} xor·xor(Bn)	Byte	ASCII converted, swapped (ex.3231)
H0013	{ (W1)xor(W2)} xor·xor(Wn)	Word	Normal (ex. 1234)
H0014	{ (W1)xor(W2)} xor·xor(Wn)	Word	Swapped (ex. 3412)
Others	DATA Error (DER ON)		

[1] Byte format :
Verification starting byte position is specified as below in case of byte oriented calculation.

	Byte type				Word type			
Starting Word	(B1)	(B2)		(B1)	(W1_H)	(W1_L)		(W1_H)
+1	(B3)	(B4)	(B2)	(B3)	(W2_H)	(W2_L)	(W1_L)	(W2_H)
+2	(B5)	(B6)	(B4)	(B5)	(W3_H)	(W3_L)	(W2_L)	(W3_H)
	…	…	…	…	…	…	(W3_L)	…
	…	…	…	…	…	…	…	(Wn_H)
+(m-1)	(Bn-1)	(Bn)	(Bn)		(Wn_H)	(Wn_L)	(Wn_L)	

H : High byte
L : Low byte
Wn : | Wn_H | Wn_L |

<High byte>
Verification starting byte

<Low byte>
Check code starting byte

H00xx : Verification starts from high byte
H01xx : Verification starts from low byte
Others : DATA Error (DER ON)

Setting value : H00xx

B	B1	B2
	B3	B4
	…	

W	W1	
	W2	
	…	

Setting value : H01xx

B	—	B1
	B2	B3
	…	

W	—	W1_h
	W1_l	W2_h
	W2_l	…

Hxx00 : Check code starts from high byte
Hxx01 : Check code starts from low byte *
Others : Data Error (DER ON)
* If check code is WORD, L-byte is taken in H-byte position of the next word as below.

Setting value : Hxx00

B	[1]	-
W	[1]	

Setting value : Hxx01

B	-	[1]
W	-	[1]
	[1]	-

- : Existing data
[1] : Result

FUN 23 (S)

Function

[2] I/O type of data :

Type WR:H000A, WL:H000B, WM:H000C

[3] I/O address of data :

I/O address H0000 - HFFFF

[4] Data length

Byte data : unit is byte (H0000 - HFFFF)
Word dta : unit is word (H0000 - HFFFF)

[5] I/O type of check code :

Type WR:H000A, WL:H000B, WM:H000C

[6] I/O addressof check code

I/O address H0000 - HFFFF

[7] Verifying result :

OK - H8000, NG - H80FF

[8] [9] Calculation result :

Calculated value is stored in this area. If existing check code is separated in 2 words, calculated value is also stored in 2 words separately.

Sample program

<Received data frame> Check code = Sum for each byte and ASCII conversion

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

<Sample program>

```
WM0 = H0003      ——— [ 1 ]
WM1 = H0101      ——— [ 2 ]
WM2 = H000A    } ——— [ 3 ]
WM3 = H0100    }
WM4 = 10         ——— [ 4 ]
WM5 = H000A    } ——— [ 5 ]
WM6 = H0105    }
FUN 23 ( WM0 )
```

[1] Calculation type setting (Byte, ASCII, normal) :H0003
[2] Verification starts from L-byte
Check code starts from L-byte : H0101
[3] Data address : WR100 (H000A, H0100)
[4] Data length : 10 bytes
[5] Check code address : WR105 (H000A, H00105)

<Result>

WR10	0 2	3 0
WR11	3 1	3 0
WR12	3 1	3 0
WR13	3 0	3 0
WR14	3 5	3 0
WR15	3 0	**4 5**
WR16	**3 7**	0 D

31 30 31 30 30 30 35 30 30

30+31+30+31+30+30+30+35+30+30

= H 0 1 E 7 ⇨ E7

⇩ ASCII

45 37

Verifying OK (WM7=H8000) as right value. (WM7=H80FF in case of wrong value)

Item number	Fun commands-15	Name	Conversion from 16-bit unsigned binary to decimal ASCII data † (BINARY TO DECIMAL ASCII)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 30 (s)	DER	ERR	SD	V	C							
* (BINDA (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					140	←	141	←			
	Condition			Steps								
FUN 30 (s)	—			3								
* (BINDA (s))												

Usable I/O		Bit				Word				Double word				Other	
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM	Constant		
s	Argument (conversion data)							○						s uses up to s+3	

Function

16-bit unsigned binary data

s | 0 to 65535

10^n: ASCII code in the 10^n place

Decimal ASCII data

	15 8	7 0
s + 1	10^4	10^3
s + 2	10^2	10^1
s + 3	10^0	NULL

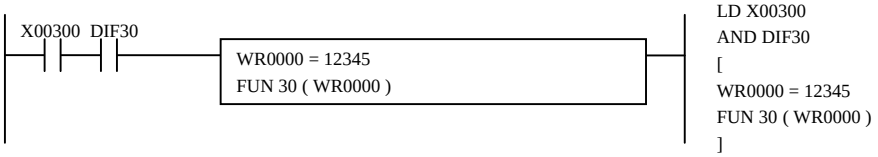
- The 16-bit unsigned binary data specified by argument s is converted to 5-digit decimal ASCII code and the result is stored in s + 1 to s + 3.
- Leading zeros of the conversion result are suppressed and these digits are replaced by H20 (space).
- The remaining digits after converting to ASCII are replaced by NULL, which indicates the end of a string.
- If the operation is performed normally, DER is set to "0."

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- If s + 1 to s + 3 exceed the maximum I/O number, DER is set to "1" and no operation is performed.

Program example




```
LD X00300
AND DIF30
[
WR0000 = 12345
FUN 30 ( WR0000 )
]
```

Program description

- The binary data 12345 stored in WR0000 is converted to ASCII data.
- The conversion result is stored in WR0001 to 3.

Execution results: WR0000=12345 (H3039), WR0001=H3132, WR0002=H3334, WR0003=H3500

†: Supported by EH-CPU ***A/448/516/548 only.

Item number	Fun commands-16	Name	Conversion from 32-bit signed binary to decimal ASCII data † (DOUBLE BINARY TO DECIMAL ASCII)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 31 (s)	DER	ERR	SD	V	C							
* (DBINDA (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps		Processing time (μs)						
	Condition	Steps							
FUN 31 (s) * (DBINDA (s))	—	3	199	←	200	←			

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument (lower)							○						-2,147,483,648 to 2,147,483,647
s	Argument (higher)							○						

Function

32-bit signed binary data

s	Lower 16-bit
s + 1	Higher 16-bit

Signs Plus: H20 (space)

Minus: H2D (“-”)

10^n: ASCII code in the 10^n place

Decimal ASCII data

	15 87	0
s + 2	Sign	10^9
s + 3	10^8	10^7
s + 4	10^6	10^5
s + 5	10^4	10^3
s + 6	10^2	10^1
s + 7	10^0	NULL

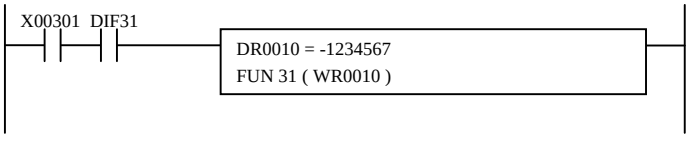
- The 32-bit signed binary data specified by arguments s (lower) and s + 1 (higher) is converted to 10-digit decimal ASCII code and the result is stored in s + 2 to s + 7.
- If the sign is a plus, it is indicated by H20 (space), and by H2D (“-“) if it is a minus.
- Leading zeros of the conversion result are suppressed and these digits are replaced by H20 (space).
- The remaining digits after converting to ASCII are replaced by NULL, which indicates the end of a string.
- If the operation is performed normally, DER is set to “0.”

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- If s + 1 to s + 7 exceed the maximum I/O number, DER is set to “1” and no operation is performed.

Program example




```
LD X00301
AND DIF31
[
DR0010 = -1234567
FUN 31 ( WR0010 )
]
```

Program description

- The binary data -1234567 stored in WR0000 (WR0010, WR0011) is converted to ASCII data.
- The conversion result is stored in WR0012 to WR0017.

Execution results: DR0010=-1234567 (HFFED2979), WR0012=H2020, WR0013=H2020, WR0014=H3132, WR0015=H3334, WR0016=H3536, WR0017=H3700

†: Supported by EH-CPU ***A/448/516/548 only.

Item number	Fun commands-17	Name	Conversion from 16-bit binary to hexadecimal ASCII data † (BINARY TO HEXA ASCII)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A / EH-CPU3**		Other than left		
FUN 32 (s)	DER	ERR	SD	V	C							
* (BINHA (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					124	←	125	←			
	Condition				Steps							
FUN 32 (s) * (BINHA (s))	—				3							

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument (conversion data)							○						s uses up to s+3

Function

16-bit unsigned binary data

s H0 to HFFFF

16^n: ASCII code in the 16^n place

Hexadecimal ASCII data

	15 8	7 0
s + 1	16^3	16^2
s + 2	16^1	16^0
s + 3	NULL	

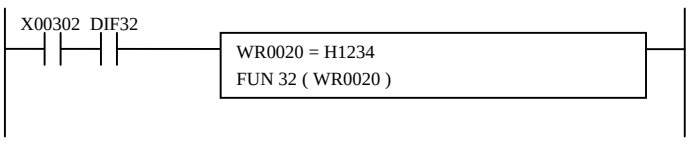
- The 16-bit unsigned binary data specified by argument s is converted to 4-digit hexadecimal ASCII code and the result is stored in s + 1 to s + 3.
- Leading zeros of the conversion result are not suppressed.
- NULL after ASCII data indicates the end of a string.
- If the operation is performed normally, DER is set to "0."

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- If s + 1 to s + 3 exceed the maximum I/O number, DER is set to "1" and no operation is performed.

Program example




```
LD X00302
AND DIF32
[
WR0020 = H1234
FUN 32 ( WR0020 )
]
```

Program description

- The binary data H1234 stored in WR0020 is converted to ASCII data.
- The conversion result is stored in WR0021 to WR0023.

Execution results: WR0020=H1234, WR0021=H3132, WR0022=H3334, WR0023=H0000

†: Supported by EH-CPU ***A/448/516/548 only.

Item number	Fun commands-18	Name	Conversion from 32-bit binary to hexadecimal ASCII data † (DOUBLE BINARY TO HEXA ASCII)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 33 (s)	DER	ERR	SD	V	C							
* (DBINHA (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps								
	Condition	Steps							
FUN 33 (s) * (DBINHA (s))	—	3	168	←	169	←			

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument (lower)							○						H00000000 to HFFFFFFFF s uses up to s+6
s + 1	Argument (higher)							○						

Function

32-bit unsigned binary data

s	Lower 16-bit
s + 1	Higher 16-bit

Hexadecimal ASCII data

	15 — 8	7 — 0
s + 2	16^7	16^6
s + 3	16^5	16^4
s + 4	16^3	16^2
s + 5	16^1	16^0
s + 6	NULL	

16^n: ASCII code in the 16^n place

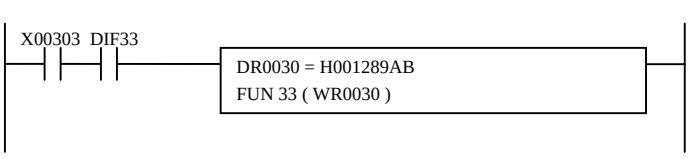
- The 32-bit signed binary data specified by arguments s (lower) and s + 1 (higher) is converted to an 8-digit hexadecimal ASCII code and the result is stored in s + 2 to s + 6.
- Leading zeros of the conversion result are not suppressed.
- NULL after ASCII data indicates the end of a string.
- If the operation is performed normally, DER is set to "0."

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- If s + 1 to s + 6 exceed the maximum I/O number, DER is set to "1" and no operation is performed.

Program example




```
LD X00303
AND DIF33
[
DR0030 = H001289AB
FUN 33 ( WR0030 )
]
```

Program description

- The binary data H001289AB stored in DR0030 (WR0030, WR0031) is converted to ASCII data.
- The conversion result is stored in WR0032 to WR0036.

Execution results: DR0030=H001289AB, WR0032=H3030, WR0033=H3132, WR0034=H3839, WR0035=H4142, WR0036=H0000

†: Supported by EH-CPU ***A/448/516/548 only.

Item number	Fun commands-19	Name	Conversion from 16-bit BCD to decimal ASCII data † (BCD TO DECIMAL ASCII)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 34 (s)	DER	ERR	SD	V	C							
* (BCDDA (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps										
	Condition	Steps									
FUN 34 (s) * (BCDDA (s))	—	3	112	←	112	←					

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument (conversion data)							○						s uses up to s+3

Function

16-bit BCD data

	15 12	11 7	4 3 0	
s	10^3	10^2	10^1	10^0

10^n: BCD code in the 10^n place

Decimal ASCII data

	15 8	7 0
s + 1	10^3	10^2
s + 2	10^1	10^0
s + 3	NULL	

10^m: ASCII code in the 10^m place

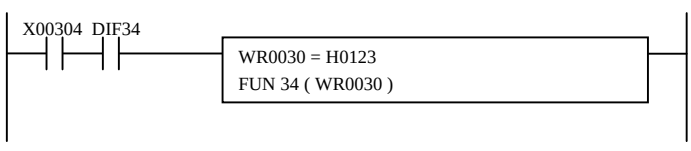
- The 16-bit BCD data specified by argument s is converted to a 4-digit decimal ASCII code and the result is stored in s +1 to s + 3.
- Leading zeros of the conversion result are suppressed and these digits are replaced by H20 (space)
- NULL after ASCII data indicates the end of a string.
- If the operation is performed normally, DER is set to "0."

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- If s is other than BCD data, DER is set to "1" and no operation is performed.
- If s + 1 to s + 3 exceed the maximum I/O number, DER is set to "1" and no operation is performed.

Program example




```
LD X00304
AND DIF34
[
WR0030 = H0123
FUN 34 ( WR0030 )
]
```

Program description

- The BCD data H0123 stored in WR0030 is converted to ASCII data.
- The conversion result is stored in WR0031 to WR0033.

Execution results: WR0030=H0123, WR0031=H2031, WR0032=H3233, WR0033=H0000

†: Supported by EH-CPU ***A/448/516/548 only.

Item number	Fun commands-20	Name	Conversion from 32-bit BCD to decimal ASCII data † (DOUBLE BCD TO DECIMAL ASCII)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 35 (s)	DER	ERR	SD	V	C							
* (DBCDDA (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps										
	Condition	Steps	171	←	171	←					
FUN 35 (s) * (DBCDDA (s))	—	3									

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument (lower)							○						s is BCD data.
s+1	Argument (higher)							○						s uses up to s+6

Function

32-bit BCD data

	15 12	11 7	4 3 0	
s	10^3	10^2	10^1	10^0
s + 1	10^7	10^6	10^5	10^4

10^n: BCD code in the 10^n place

Decimal ASCII data

	15 8	7 0
s + 2	10^7	10^6
s + 3	10^5	10^4
s + 4	10^3	10^2
s + 5	10^1	10^0
s + 6	NULL	

10^m: ASCII code in the 10^m place

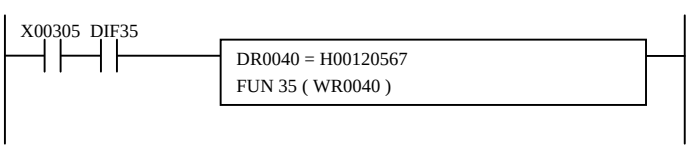
- The 32-bit BCD data specified by arguments s (lower) and s + 1 (higher) is converted to an 8-digit decimal ASCII code and the result is stored in s +2 to s + 6.
- Leading zeros of the conversion result are suppressed and these digits are replaced by H20 (space)
- NULL after ASCII data indicates the end of a string.
- If the operation is performed normally, DER is set to "0."

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- If s, s +1 is other than BCD data, DER is set to "1" and no operation is performed.
- If s + 1 to s + 6 exceed the maximum I/O number, DER is set to "1" and no operation is performed.

Program example




```
LD X00305
AND DIF35
[
DR0040 = H00120567
FUN 35 ( WR0040 )
]
```

Program description

- The BCD data H00120567 stored in DR0040 (WR0040, WR0041) is converted to ASCII data.
- The conversion result is stored in WR0042 to WR0046.

Execution results: DR0040=H00120567, WR0042=H2020, WR0043=H3132, WR0044=H3035, WR0045=H3637, WR0046=H0000

†: Supported by EH-CPU ***A/448/516/548 only.

Item number	Fun commands-21	Name	Conversion from 5-digit unsigned decimal ASCII to 16-bit binary data † (DECIMAL ASCII TO BINARY)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 36 (s)	DER	ERR	SD	V	C							
* (DABIN (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps										
	Condition	Steps	60	←	61	←					
FUN 36 (s)	—	3									
* (DABIN (s))											

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument (higher)							○						s to s + 2 will have combinations of H00, H20, and H 30 to H39. s uses up to s + 3
s+1	Argument (middle)							○						
s+2	Argument (lower)							○						

Function

Unsigned decimal ASCII data

s	10^4	10^3
s + 1	10^2	10^1
s + 2	10^0	H00

10^n: ASCII code in the 10^n place

16-bit binary data

s + 3	0 to 65535

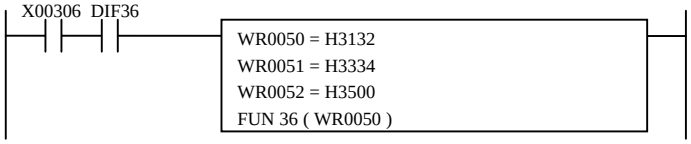
- The 5-digit unsigned decimal ASCII data specified by arguments s (upper), s + 1 (middle), and s + 2 (lower) is converted to 16-bit binary data and the result is stored in s + 3.
- Higher digit's H00 and H20 (NULL and space) are processed as H30 ("0"). (Leading-zero-suppressed digit)
- If the operation is performed normally, DER is set to "0."

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- If the 5-digit ASCII code stored in s to s + 2 is other than H30 to H39 (0 to 9), DER is set to "1" and no operation is performed. However, this does not apply to H00 and H20 (NULL and space) of leading-zero-suppressed digits.
- If s + 1 to s + 3 exceed the maximum I/O number, DER is set to "1" and no operation is performed.
- If a data value is 65,536 or higher, DER is set to "1" and no operation is performed.

Program example




```
LD X00306
AND DIF36
[
WR0050 = H3132
WR0051 = H3334
WR0052 = H3500
FUN 36 ( WR0050 )
]
```

Program description

- The ASCII data "1," "2," "3," "4," "5" stored in WR0050 to WR0052 is converted to binary data.
- The conversion result is stored in WR0053.

Execution results: WR0050=H3132, WR0051=H3334, WR0052=H3500, WR0053=12345 (H3039)

†: Supported by EH-CPU ***A/448/516/548 only.

Item number	Fun commands-22	Name	Conversion from 10-digit signed decimal ASCII to 32-bit binary data † (DOUBLE DECIMAL ASCII TO BINARY)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 37 (s)	DER	ERR	SD	V	C							
* (DDABIN (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps									
	Condition	Steps	97	←	98	←				
FUN 37 (s) * (DDABIN (s))	—	3								

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument (ASCII code)							○						Sign is H20 or H2D, and other digits are combinations of H00, H20, and H30 to H39. s uses up to s + 7
~	~							~						
s+5	Argument (ASCII code)							○						

Function

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

s + 6	Lower 16-bit
s + 7	Higher 16-bit

Signs Plus: H20 (space)

Minus: H2D (“-”)

10^n: ASCII code in the 10^n place

- The 10-digit signed decimal ASCII data specified by arguments s to s + 6 is converted to 32-bit binary data and the result is stored in s + 7 (higher) and s + 6 (lower).
- Arguments will be combinations of H00, H20, H30 to H39, and H2D (“-”).
- Higher digit’s H00 and H20 (NULL and space) are processed as H30 (“0”). (Leading-zero-suppressed digit)
- If the operation is performed normally, DER is set to “0.”
- Signed data must be in the range from –2,147,483,648 to 2,147,483,647.

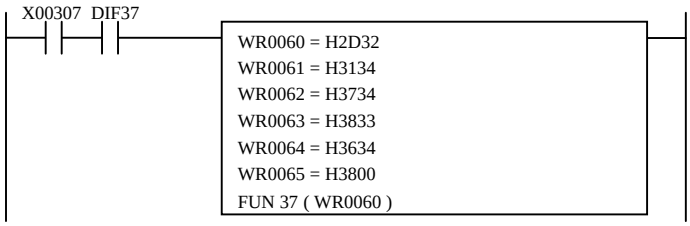
* ( ) indicates the display when the LADDER EDITOR is used.

†: Supported by EH-CPU ***A/448/516/548 only.

Cautionary notes

- If the sign is other than H20 and H2D, and other digits are other than H30 to H39 (0 to 9), DER is set to “1” and no operation is performed. However, this does not apply to H00 and H20 (NULL and space) of leading-zero-suppressed digits.
- If data is outside the range from –2,147,483,648 to 2,147,483,647, DER is set to “1” and no operation is performed.
- If s + 1 to s + 7 exceed the maximum I/O number, DER is set to “1” and no operation is performed.

Program example




```
LD X00307
AND DIF37
[
WR0060 = H2D32
WR0061 = H3134
WR0062 = H3734
WR0063 = H3833
WR0064 = H3634
WR0065 = H3800
FUN 37 ( WR0060 )
]
```

Program description

- The ASCII data “-,” “2,” “1,” “4,” “7,” “4,” “8,” “3,” “6,” “4,” “8” stored in WR0060 to WR0065 is converted to binary data.
- The conversion result is stored in WR0067 (higher) and WR0066 (lower).

Execution results: WR0060=H2D32, WR0061=H3134, WR0062=H3734, WR0063=H3833, WR0064=H3634, WR0065=H3800, DR0060=-2147483648(H80000000)

Item number	Fun commands-23	Name	Conversion from 4-digit hexadecimal ASCII to 16-bit binary data † (HEXA ASCII TO BINARY)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 38 (s)	DER	ERR	SD	V	C							
* (HABIN (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps										
	Condition	Steps	63	←	64	←					
FUN 38 (s)	—	3									
* (HABIN (s))											

Usable I/O		Bit				Word				Double word				Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM	Constant	
s	Argument (higher ASCII)							○						Combination of H00, H20, H30 to H39 and H41 to 46
s+1	Argument (lower ASCII)							○						s uses up to s + 2

Function

Hexadecimal ASCII data

	15 — 8	7 — 0
s	16^3	16^2
s + 1	16^1	16^0

16^n: ASCII code in the 16^n place

16-bit binary data

s + 2	H0 to HFFFF

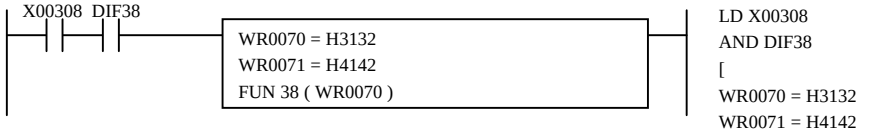
- The 4-digit hexadecimal ASCII data specified by arguments s and s + 1 is converted to binary data and the result is stored in s + 2.
- Higher digit's H00 and H20 (NULL and space) are processed as H30 ("0"). (Leading-zero-suppressed digit)
- Arguments will be combinations of H30 to H39 and H41 to H46(0 to 9 and A to F).
- If the operation is performed normally, DER is set to "0."

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- If the 4-digit ASCII code stored in s to s + 1 is other than H30 to H39, H41 to H46 (0 to 9 and A to F), DER is set to "1" and no operation is performed. However, this does not apply to H00 and H20 (NULL and space) of leading-zero-suppressed digits.
- If s + 1 to s + 2 exceed the maximum I/O number, DER is set to "1" and no operation is performed.

Program example




```
LD X00308
AND DIF38
[
WR0070 = H3132
WR0071 = H4142
FUN 38 ( WR0070 )
]
```

Program description

- The ASCII data "1," "2," "A," "B" stored in WR0070, WR0071 is converted to binary data.
- The conversion result is stored in WR0072.

Execution results: WR0070=H3132, WR0071=H4142, WR0072=H12AB

†: Supported by EH-CPU ***A/448/516/548 only.

Item number	Fun commands-24	Name	Conversion of 8-digit hexadecimal ASCII to 32-bit binary data † (DOUBLE HEXA ASCII TO BINARY)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 39 (s)	DER	ERR	SD	V	C							
* (DHABIN (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps									
	Condition	Steps	94	←	94	←				
FUN 39 (s) * (DHABIN (s))	—	3								

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument (ASCII data)							○						Combination of H00, H20, H30 to H39 and H41 to 45
~	~							~						
s	Argument (ASCII data)							○						s uses up to s + 5

Function

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 in the 16^n place

32-bit binary data

s + 4	Lower 16-bit
s + 5	Higher 16-bit

H00000000 to HFFFFFFFF

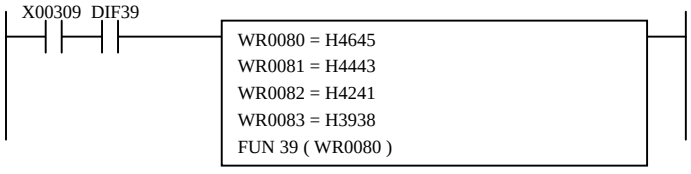
- The 8-digit hexadecimal ASCII data specified by arguments s to s + 3 is converted to binary data and the result is stored in s + 4 and s + 3.
- Higher digit's H00 and H20 (NULL and space) are processed as H30 ("0"). (Leading-zero-suppressed digit)
- The argument will be a combination of H30 to H30 and H41 to H46 (0 to 9 and A to F).
- If the operation is performed normally, DER is set to "0."

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- If the 8-digit ASCII code stored in s to s + 3 is other than H30 to H39 and H41 to H46 (0 to 9 and A to F), DER is set to "1" and no operation is performed. However, this does not apply to H00 and H20 (NULL and space) of leading-zero-suppressed digits.
- If s + 1 to s + 5 exceed the maximum I/O number, DER is set to "1" and no operation is performed.

Program example




```
LD X00309
AND DIF39
[
WR0080 = H4645
WR0081 = H4443
WR0082 = H4241
WR0083 = H3938
FUN 39 ( WR0080 )
]
```

Program description

- The ASCII data "F," "E," "D," "C," "B," "A," "9," "8" stored in WR0080 to WR0083 is converted to binary data.
- The conversion result is stored in WR0084 and WR0085.

Execution results: WR0080=H4645, WR0081=H4443, WR0082=H4241, WR0083=H3938, DR0084=HFEDCBA98

†: Supported by EH-CPU ***A/448/516/548 only.

Item number	Fun commands-25	Name	Conversion from 4-digit decimal ASCII to 16-bit BCD data † (DECIMAL ASCII TO BCD)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 40 (s)	DER	ERR	SD	V	C							
* (DABCD (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps										
	Condition	Steps	64	←	65	←					
FUN 40 (s) * (DABCD (s))	—	3									

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument (ASCII data)							○						Combination of H00, H20 and H30 toH39 s uses up to s + 2
s+1	Argument (ASCII data)							○						

Function

Decimal ASCII data

s	10^3	10^2
s + 1	10^1	10^0

16-bit unsigned BCD data

s + 2	10^3	10^2	10^1	10^0

10^m: ASCII code in the 10^m place

10^n: BCD code in the 10^n place

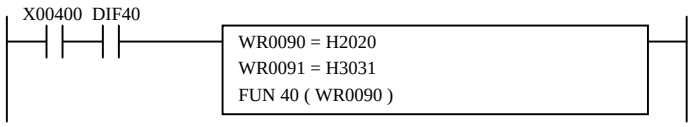
- The 4-digit decimal ASCII data specified by arguments s to s + 1 is converted to 16-bit BCD data and the result is stored in s + 2.
- Higher digit's H00 and H20 (NULL and space) are processed as H30 ("0"). (Leading-zero-suppressed digit)
- Arguments will be combinations of H30 to H39 (0 to 9).
- If the operation is performed normally, DER is set to "0."

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- If the 4-digit ASCII code stored in s to s + 1 is other than H30 to H39 (0 to 9), DER is set to "1" and no operation is performed. However, this does not apply to H00 and H20 (NULL and space) of leading-zero-suppressed digits.
- If s + 1 to s + 2 exceed the maximum I/O number, DER is set to "1" and no operation is performed.

Program example




```
LD X00400
AND DIF40
[
WR0090 = H2020
WR0091 = H3031
FUN 40 ( WR0090 )
]
```

Program description

- The ASCII data "␣" "␣" "0," "1," stored in WR0090 and WR0091 is converted to 16-bit BCD data.
- The conversion result is stored in WR0092. ("␣"=H20)

Execution results: WR0090=H2020, WR0091=H3031, WR0092=H0001

†: Supported by EH-CPU ***A/448/516/548 only.

Item number	Fun commands-26	Name	Conversion from 8-digit decimal ASCII to 32-bit BCD data † (DOUBLE DECIMAL ASCII TO BCD)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A / EH-CPU3**		Other than left		
FUN 41 (s)	DER	ERR	SD	V	C							
* (DDABCD (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps								
	Condition	Steps	93	←	94	←			
FUN 41 (s) * (DDABCD (s))	—	3							

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument (ASCII code)							○						Combination of H00, H20 and H30 to H39 s uses up to s + 5
~	~							~						
s + 3	Argument (ASCII code)							○						

Function

Decimal ASCII data

	15 8	7 0
s	10^7	10^6
s + 1	10^5	10^4
s + 2	10^3	10^2
s + 3	10^1	10^0

32-bit BCD data

	15 12	11 8	7 4	3 0
s + 4	10^3	10^2	10^1	10^0
s + 5	10^7	10^6	10^5	10^4

10^n: BCD code in the 10^n place

10^m: ASCII code in the 10^m place

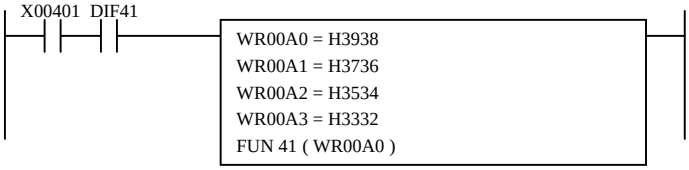
- The 8-digit decimal ASCII data specified by arguments s to s + 1 is converted to 32-bit BCD data and the result is stored in s + 4 (lower), s + 5 (higher).
- Higher digit's H00 and H20 (NULL and space) are processed as H30 ("0"). (Leading-zero-suppressed digit)
- Arguments will be combinations of H30 to H39 (0 to 9).
- If the operation is performed normally, DER is set to "0."

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- If the 8-digit ASCII code stored in s to s + 3 is other than H30 to H39 (0 to 9), DER is set to "1" and no operation is performed. However, this does not apply to H00 and H20 (NULL and space) of leading-zero-suppressed digits.
- If s + 1 to s + 5 exceed the maximum I/O number, DER is set to "1" and no operation is performed.

Program example




```
LD X00401
AND DIF41
[
WR00A0 = H3938
WR00A1 = H3736
WR00A2 = H3534
WR00A3 = H3332
FUN 41 ( WR00A0 )
]
```

Program description

- The ASCII data "9," "8," "7," "6," "5," "4," "3," "2" stored in WR00A0 to WR00A3 is converted to 32-bit BCD data.
- The conversion result is stored in WR00A4, WR00A5.

Execution results: WR00A0=H3938, WR00A1=H3736, WR00A2=H3534, WR00A3=H3332, DR00A4=H98765432

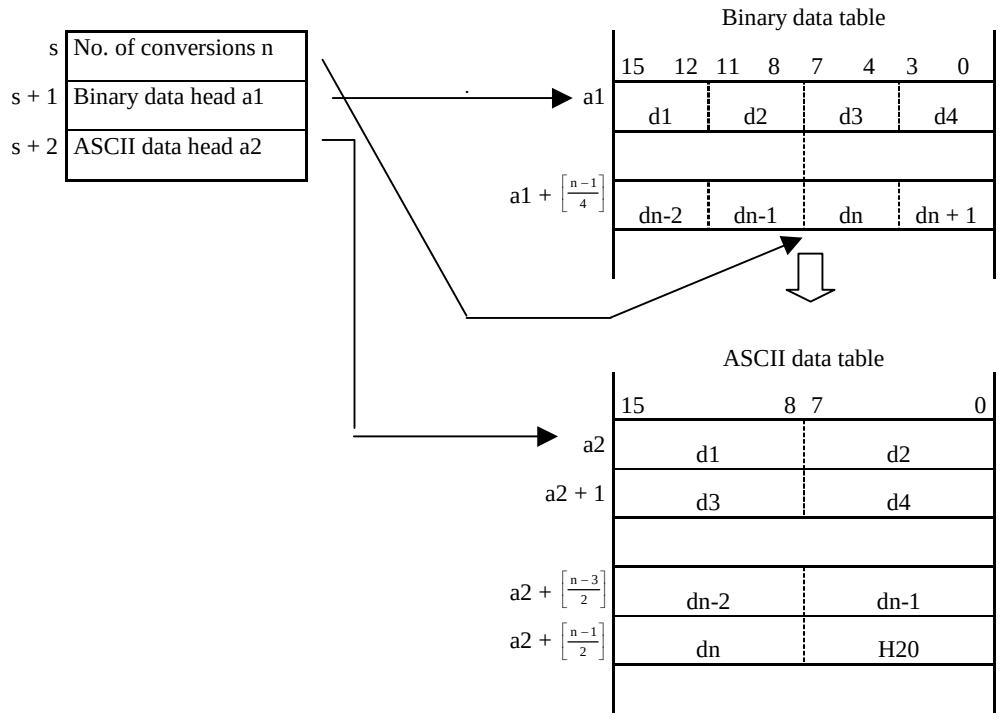
†: Supported by EH-CPU ***A/448/516/548 only.

Item number	Fun commands-27	Name	Conversion from hexadecimal binary to hexadecimal ASCII data † (BINARY TO ASCII)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 42 (s)	DER	ERR	SD	V	C							
* (ASC (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					119	←	116				10 char. conv.
	Condition			Steps								
FUN 42 (s) * (ASC (s))	—			3								

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	No. of converted characters							○						s uses up to s+2
s+1	Binary data head I/O No.							○						Actual address is set
s+2	ASCII head I/O No. after conversion							○						Actual address is set

Function



- The number of hexadecimal data characters specified by argument s is converted to hexadecimal ASCII codes beginning from the head I/O specified by argument s + 1, and the results are stored in addresses beginning from the head I/O specified by s + 2.
- If the number of characters is odd, the lower 8 bits of the data at the output destination will be H20 (space).
- Use the ADRIO command to set the actual addresses in the head I/Os of s + 1 and s + 2.
- If the operation is performed normally, DER is set to "0."

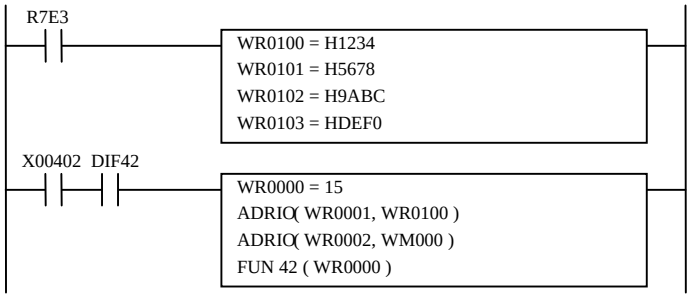
* ( ) indicates the display when the LADDER EDITOR is used.

†: Supported by EH-CPU ***A/448/516/548 only.

Cautionary notes

- The ADRIO command should be used to set the actual addresses in s + 1 and s + 2. If not, DER is set to “1” and no operation is performed.
- If s to s + 2 and the areas specified by them overlap, DER is set to “1” and no operation is performed.
- If s to s + 2 and the areas specified by s + 1 and s + 2 exceed the maximum I/O number, DER is set to “1” and no operation is performed.

Program example




```
LD R7E3
[
WR0100 = H1234
WR0101 = H5678
WR0102 = H9ABC
WR0103 = HDEF0
]
LD X00402
AND DIF42
[
WR0000 = 15
ADRIO ( WR0001, WR0100 )
ADRIO( WR0002, WM000 )
FUN 42 ( WR0000 )
]
```

Program description

1) The result is stored in the data table from WR0100 by special internal output R7E3 (single scan ON after RUN start).
2) At a rising edge of X00402, the hexadecimal binary data is converted to hexadecimal ASCII data, and the converted data is stored from WM000.

Execution results:

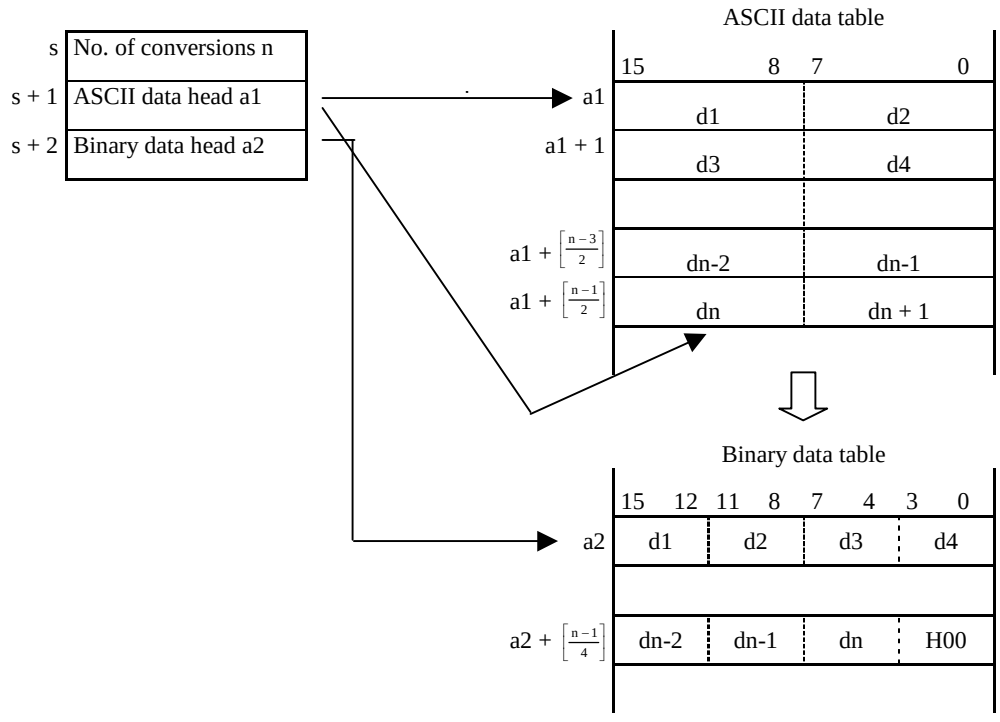
WR0100 = H1234	⇨	WM000=H3132, WM001=H3334
WR0101 = H5678		WM002=H3536, WM003=H3738
WR0102 = H9ABC		WM004=H3941, WM005=H4243
WR0103 = HDEF0		WM006=H4445, WM007=H4620 (“20” is a space.)

Item number	Fun commands-28	Name	Conversion from hexadecimal ASCII to hexadecimal binary data † (ASCII TO BINARY)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 43 (s)	DER	ERR	SD	V	C							
* (HEX (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					191	←	188	←			10 char. conv.
	Condition			Steps								
FUN 43 (s) * (HEX) (s))	—			3								

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	No. of converted characters							○						s uses up to s+2
s+1	ASCII head I/O No.							○						Actual address is set
s+2	Binary conversion data head I/O No.							○						Actual address is set

Function



- The number of hexadecimal ASCII code characters specified by argument s is converted to binary data beginning from the head of the hexadecimal ASCII code specified by argument s + 1, and the results are stored in addresses beginning from the head I/O specified by s + 2.
- If the number of characters is odd, the lower 4 bits of the data at the output destination will be "0."
- Use the ADRIO command to store the actual addresses of the head I/Os at s + 1 and s + 2.
- Higher digit's H00 and H20 (NULL and space) are processed as H30 ("0"). (Leading-zero-suppressed digit)
- If the operation is performed normally, DER is set to "0."

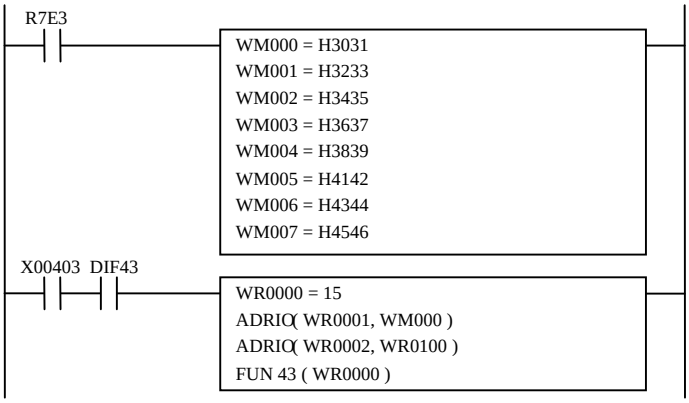
* ( ) indicates the display when the LADDER EDITOR is used.

†: Supported by EH-CPU ***A/448/516/548 only.

Cautionary notes

- The ADRIO command should be used to set the actual addresses in s + 1 and s + 2. If not, DER is set to “1” and no operation is performed.
- If s to s + 2 and the areas specified by them overlap, DER is set to “1” and no operation is performed.
- If s to s + 2 and the areas specified by s + 1 and s + 2 exceed the maximum I/O number, DER is set to “1” and no operation is performed.

Program example




```
LD R7E3
[
WM000 = H3031
WM001 = H3233
WM002 = H3435
WM003 = H3637
WM004 = H3839
WM005 = H4142
WM006 = H4344
WM007 = H4546
]
LD X00403
AND DIF43
[
WR0000 = 15
ADRIO( WR0001, WM000 )
ADRIO( WR0002, WR0100 )
FUN 43 ( WR0000 )
]
```

Program description

1) The result is stored in the data table from WR0100 by special internal output R7E3 (single scan ON after RUN start).
2) At a rising edge of X00403, the hexadecimal ASCII data is converted to hexadecimal binary data, and the converted data is stored from WM0100.

Execution results:

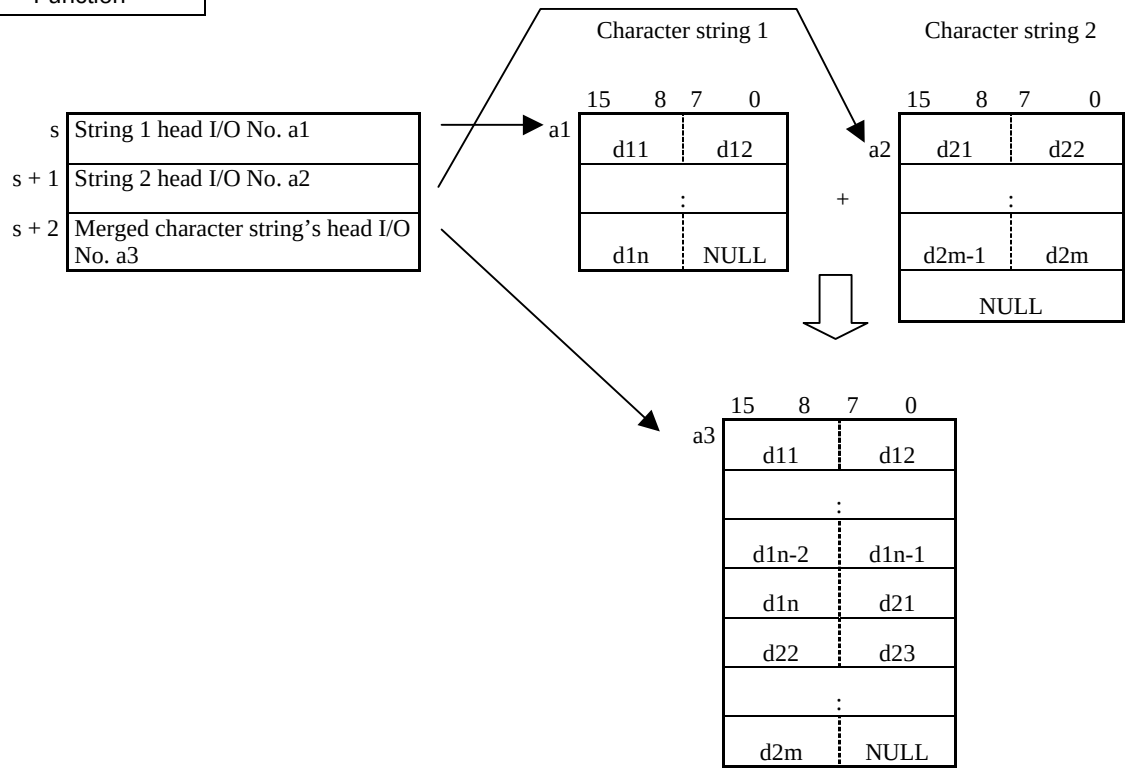
WM000=H3031, WM001=H3233	⇨	WR0100=H0123
WM002=H3435, WM003=H3637		WR0101=H4567
WM004=H3839, WM005=H4142		WR0102=H89AB
WM006=H4344, WM007=H4546		WR0103=HCDE0

Item number	Fun commands-29	Name	Merge strings †

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 44 (s)	DER	ERR	SD	V	C							10 char. + 10 char. ↓ 20 char.
* (SADD (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					196	←	184	←			
	Condition			Steps								
FUN 44 (s)	—			3								
* (SADD (s))												

Usable I/O		Bit				Word				Double word				Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM	Constant	
s	String 1 head I/O No.							○						Actual addresses are set in s to s + 2
s+1	String 2 head I/O No.							○						
s+2	Merged character string's head I/O No.							○						

Function



- The string that begins from the head I/O specified by argument s is merged with the string that begins from the head I/O specified by argument s + 1, and the result is stored in the head I/O area specified by s + 2.
- The character strings to be merged end before a NULL (H00).
- A NULL will be set after the merged character string.
- Use the ADRIO command to store the actual addresses of the head I/Os at s and s + 2.
- If the operation is performed normally, DER is set to "0."

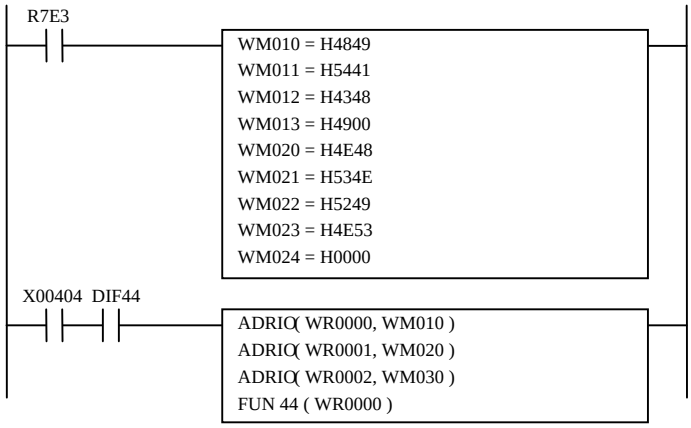
* ( ) indicates the display when the LADDER EDITOR is used.

†: Supported by EH-CPU ***A/448/516/548 only.

Cautionary notes

- The ADRIO command should be used to set the actual addresses in s to s + 2. If not, DER is set to "1" and no operation is performed.
- If s to s + 2 and the areas specified by them overlap, DER is set to "1" and no operation is performed.
- If s to s + 2 and the areas specified by s + 1 and s + 2 exceed the maximum I/O number, DER is set to "1" and no operation is performed.

Program example




```
LD R7E3
[
WM010 = H4849   "H", "I"
WM011 = H5441   "T", "A"
WM012 = H4348   "C", "H"
WM013 = H4900   "I", NULL
WM020 = H4E48   "E", "H"
WM021 = H534E   "S", "E"
WM022 = H5249   "R", "I"
WM023 = H4E53   "E", "S"
WM024 = H0000   NULL
]
LD X00404
AND DIF44
[
ADRIO( WR0000, WM010 )
ADRIO( WR0001, WM020 )
ADRIO( WR0002, WM030 )
FUN 44 ( WR0000 )
]
```

Program description

1) Sets the first character string from WM010 and the second character string from WM020 using special internal output R7E3 (single scan ON after RUN start).
2) At a rising edge of X00404, character strings are merged and output to WM030 and succeeding areas.

Execution results:

WM010=H4849		WM020=H4E48		WM030=H4849
WM011=H5441		WM021=H534E		WM031=H5441
WM012=H4348	+	WM022=H5249	⇨	WM032=H4348
WM013=H4900		WM023=H4E53		WM033=H494E
		WM024=H0000		WM034=H4853
				WM035=H4E52
				WM036=H494E
				WM037=H5300

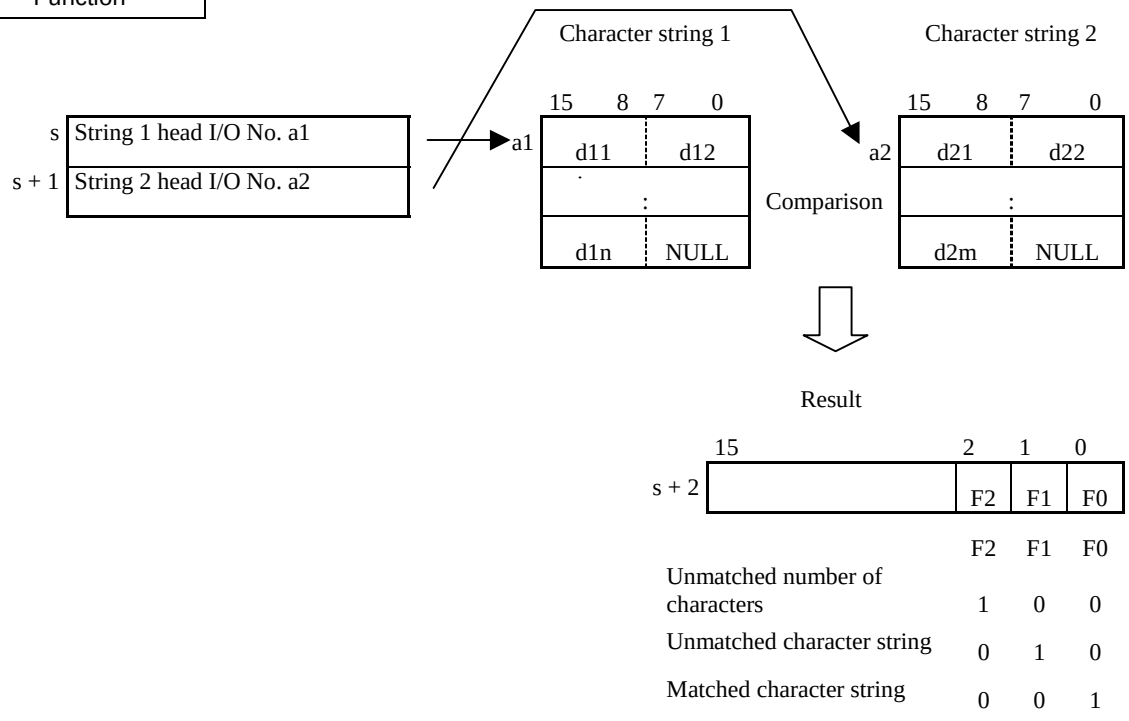
Item number	Fun commands-30	Name	Compare character strings †

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 45 (s)	DER	ERR	SD	V	C							Compar-ison between 10 char. and 10 char.
* (SCMP (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps								
	Condition	Steps	157	←	147	←			
FUN 45 (s)	—	3							
* (SCMP (s))									

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	String 1 head I/O No.							○						Actual addresses are set in s to s + 1 s uses up to s + 2
s+1	String 2 head I/O No.							○						

Function



	F2	F1	F0
Unmatched number of characters	1	0	0
Unmatched character string	0	1	0
Matched character string	0	0	1

- The character string that begins from the head I/O specified by argument s and the character string that begins from the head I/O specified by argument s + 1 are compared, and the result is stored in s + 2.
- The character strings to be compared end before a NULL (H00).
- The numbers of characters in the strings are compared first. If the numbers do not match, bit 2 is set to "1." If the numbers of characters match, the strings themselves are compared. If they do not match, bit 1 is set to "1." If both the numbers of characters and strings match, bit 0 is set to "1."
- Use the ADRIO command to set the actual addresses in the head I/Os of s and s + 1.
- If the operation is performed normally, DER is set to "0."

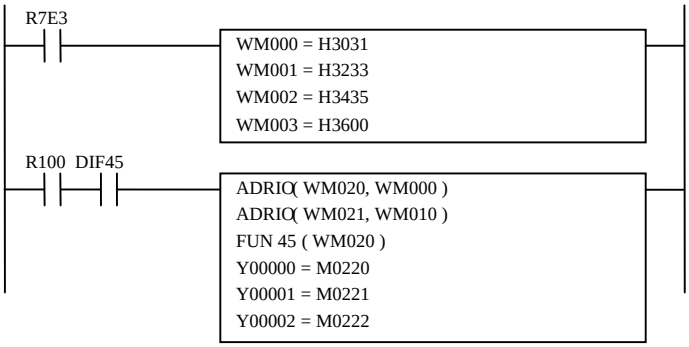
* ( ) indicates the display when the LADDER EDITOR is used.

†: Supported by EH-CPU ***A/448/516/548 only.

Cautionary notes

- The ADRIO command should be used to set the actual addresses in s and s + 1. If not, DER is set to “1” and no operation is performed.
- If s to s + 2 and the areas specified by them overlap, DER is set to “1” and no operation is performed.
- If s to s + 2 and the areas specified by s and s + 1 exceed the maximum I/O number, DER is set to “1” and no operation is performed.

Program example




```
LD R7E3
[
WM000 = H3031
WM001 = H3233
WM002 = H3435
WM003 = H3600
]
LD R100
AND DIF45
[
ADRIO( WM020, WM000 )
ADRIO( WM021, WM010 )
FUN 45 ( WM020 )
Y00000 = M0220
Y00001 = M0221
Y00002 = M0222
]
```

Program description

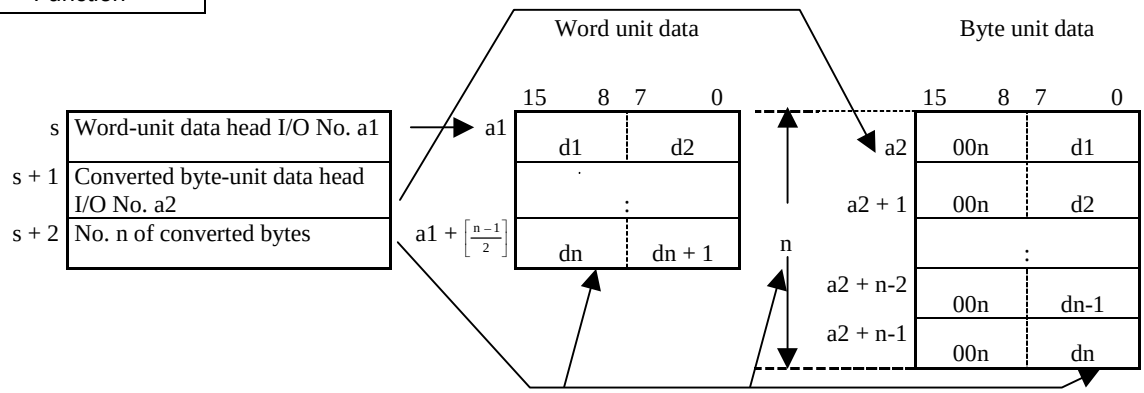
1) The compared data is stored in WM000 and succeeding areas by special internal output R7E3 (single scan ON after RUN start).
2) At a rising edge of R100, the data beginning from WM000 and the data beginning from WM010 are compared.
3) Depending on the comparison result, Y00000 to Y00002 turn on.

Item number	Fun commands-31	Name	Conversion from word units to byte units †

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 46 (s)	DER	ERR	SD	V	C							
* (WTOB (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	5 words to 10 bytes (word)
Command format	Number of steps					115	←	111	←			
	Condition			Steps								
FUN 46 (s)	—			3								
* (WTOB (s))												

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Word data head I/O No.							○						Actual addresses are set in s and s + 1 s uses up to s + 2
s+1	Byte conversion data head I/O No.							○						
s+2	No. of converted bytes							○						

Function



- The word character string data of the head I/O specified by argument s is divided into byte units for the number of bytes specified by argument s + 2, and the result is stored in the head I/O area specified by s + 1.
- Use the ADRIO command to set the actual addresses in the head I/Os of s to s + 1.
- The higher byte of the divided data is set to H00.
- If the operation is performed normally, DER is set to "0."

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- The ADRIO command should be used to set the actual addresses in s and s + 1. If not, DER is set to "1" and no operation is performed.
- If s to s + 2 and the areas specified by them overlap, DER is set to "1" and no operation is performed.
- If s to s + 2 and the areas specified by s and s + 1 exceed the maximum I/O number, DER is set to "1" and no operation is performed.

†: Supported by EH-CPU ***A/448/516/548 only.

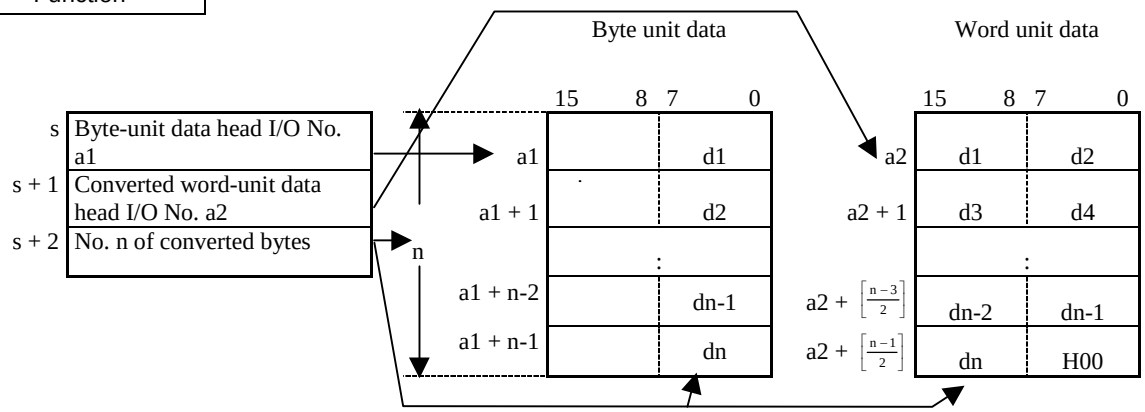
Item number	Fun commands-32	Name	Conversion from byte units to words †

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		10 bytes (word) to 5 words
FUN 47 (s)	DER	ERR	SD	V	C							
* (BTOW (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps		Processing time (μs)					
	Condition	Steps						
FUN 47 (s)	—	3	109	←	104	←		
* (BTOW (s))								

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Byte-unit data head I/O No.							○						Actual addresses are set in s and s + 1
s+1	Word-unit data head I/O No.							○						s uses up to s + 2
s+2	No. of converted bytes							○						

Function



- A byte data string is combined into word units beginning from the head I/O specified by argument s for the number of bytes specified by argument s + 2, and the result is stored in the head I/O area specified by s + 1.
- The higher byte of the byte unit data is ignored.
- If the number of converted bytes is odd, the lower 8 bits at the end of the output destination is set to H00.
- Use the ADRIO command to set the actual addresses in the head I/Os of s and s + 1.

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- The ADRIO command should be used to set the actual addresses in s and s + 1. If not, DER is set to "1" and no operation is performed.
- If s to s + 2 and the areas specified by them overlap, DER is set to "1" and no operation is performed.
- If s to s + 2 and the areas specified by s to s + 2 exceed the maximum I/O number, DER is set to "1" and no operation is performed.

†: Supported by EH-CPU ***A/448/516/548 only.

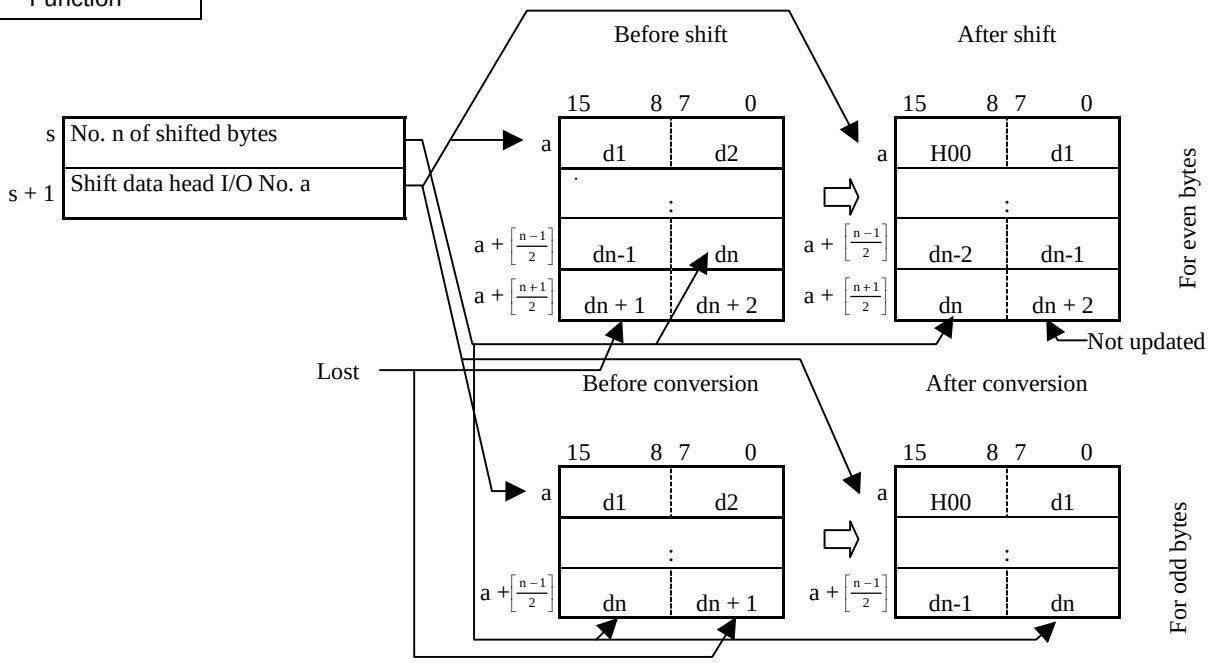
Item number	Fun commands-33	Name	Byte right shift †

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 48 (s)	DER	ERR	SD	V	C							
* (BSHR (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
						81	←	79	←			Shift 10 char. by 1 byte

Command format	Number of steps	
	Condition	Steps
FUN 48 (s) * (BSHR (s))	—	3

Usable I/O		Bit				Word				Double word				Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM	Constant	
s	No. of shifted bytes							○						Address is set in s + 1. s uses up to s + 1.
s+1	Shift data head I/O No.							○						

Function



- The data given by the number of bytes specified by argument s is shifted one byte to the right, beginning from the head I/O specified by argument s + 1.
- An H00 is inserted in an area that became empty after the shift. Note that the data after the specified number of bytes is lost by the shift operation.
- Use the ADRIO command to set the actual addresses in the head I/Os of s + 1.
- If the operation is performed normally, DER is set to "0."

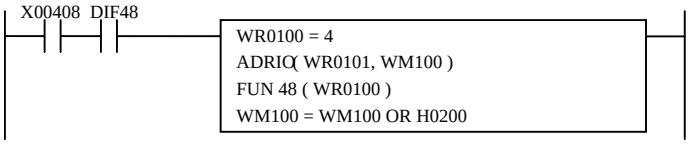
* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- The ADRIO command should be used to set the actual addresses in s + 1. If not, DER is set to "1" and no operation is performed.
- If s and s + 1 and the areas specified by them overlap, DER is set to "1" and no operation is performed.
- If s + 1 and the areas specified by s and s + 1 exceed the maximum I/O number, DER is set to "1" and no operation is performed.

†: Supported by EH-CPU ***A/448/516/548 only.

Program example




```
LD X00408
AND DIF48
[
WR0100 = 4
ADRIO( WR0101, WM100 )
FUN 48 ( WR0100 )
WM100 = WM100 OR H0200
]
```

Program description

Four bytes of transmission data is stored in WM100 and succeeding areas. Communication control code H02 (STX) is added to the head of this data.

Execution results:

WM100	"T"	"E"
WM101	"X"	"T"
WM102		

⇨

WM100	H02	"T"
WM101	"E"	"X"
WM102	"T"	

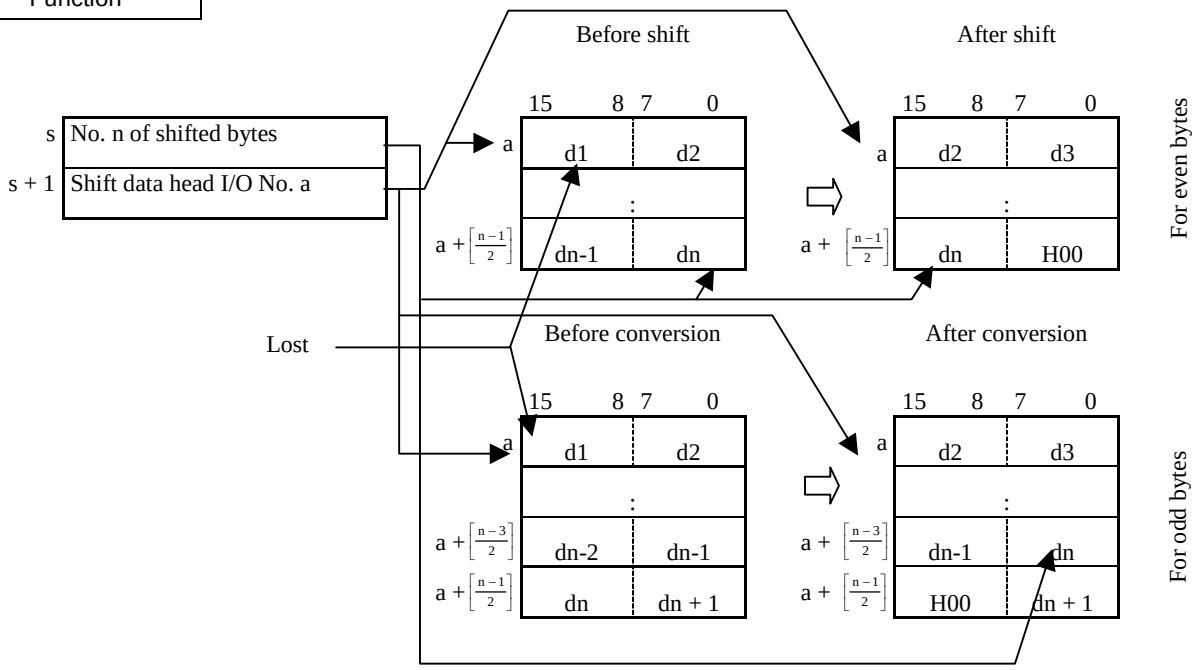
Item number	Fun commands-34	Name	Byte left shift †

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 49 (s)	DER	ERR	SD	V	C							
* (BSHL (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps											
	Condition	Steps	85	←	84	←						10 char.
FUN 49 (s)	—	3										
* (BSHL (s))												

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	No. of shifted bytes							○						Actual address is set s uses up to s + 1.
s+1	Shift data head I/O No.							○						

Function



- The data given by the number of bytes specified by argument s is shifted one byte to the left, beginning from the head I/O specified by argument s + 1,.
- An H00 is inserted in an area that became empty after the shift. Note that the head data is lost by the shift operation.
- Use the ADRIO command to set the actual addresses in the head I/Os of s + 1.
- If the operation is performed normally, DER is set to "0."

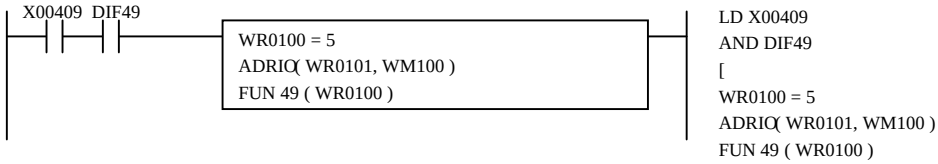
* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- The ADRIO command should be used to set the actual addresses in s + 1. If not, DER is set to "1" and no operation is performed.
- If s and s + 1 and the areas specified by them overlap, DER is set to "1" and no operation is performed.
- If s + 1 and the areas specified by s and s + 1 exceed the maximum I/O number, DER is set to "1" and no operation is performed.

†: Supported by EH-CPU ***A/448/516/548 only.

Program example




```
LD X00409
AND DIF49
[
WR0100 = 5
ADRIO( WR0101, WM100 )
FUN 49 ( WR0100 )
]
```

Program description

Five bytes of data with control code is stored in WM100 and succeeding areas. The control code is deleted from this data so that it becomes a data string containing only data.

Execution results:

WM100	H02	“T”
WM101	“E”	“X”
WM102	“T”	

⇨

WM100	“T”	“E”
WM101	“X”	“T”
WM102	H00	

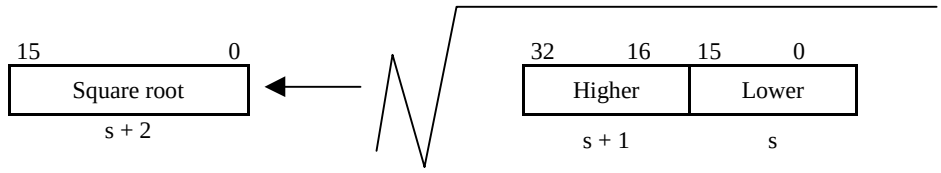
Item number	Fun commands-35	Name	Binary square root †

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 60 (s)	DER	ERR	SD	V	C							
* (BSQR (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps									
	Condition	Steps	453	←	454	←				
FUN 60 (s)	—	3								
* (BSQR (s))										

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument (lower)							○						
s+2	Argument (higher)							○						s uses up to s+2

Function



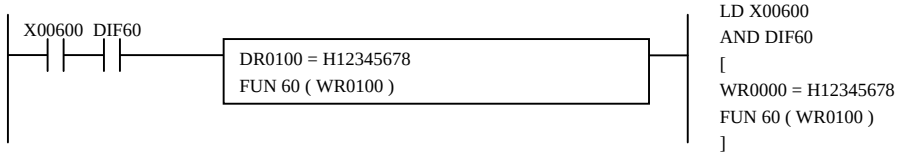
- The square root of the 32-bit binary value specified by arguments s (lower) and s + 1 (higher) is calculated and stored in s + 2.
- If the operation is performed normally, DER is set to "0."

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- If s + 1 to s + 2 exceed the maximum I/O number, DER is set to "1" and no operation is performed.

Program example




```
LD X00600
AND DIF60
[
WR0000 = H12345678
FUN 60 ( WR0100 )
]
```

Program description

- Stores the 32-bit binary value H12345678 in WR0100 and WR0101.
- The conversion result is stored in WR0102.

Execution results: DR0100=H12345678, WR0102=H4444(17476)

†: Supported by EH-CPU ***A/448/516/548 only.

Item number	Fun commands-36	Name	Dynamic scan pulse command †

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 61 (s)	DER	ERR	SD	V	C							
* (PGEN (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps										
	Condition	Steps	73	←	70	←					
FUN 61 (s)	—	3									
* (PGEN (s))											

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	No. of ON scan pulse							○						
s+1	No. of OFF scan pulse							○						
s+2	Pulse output I/O			○										Actual address is set
s+3 s+4	System area													Cannot be used by the user

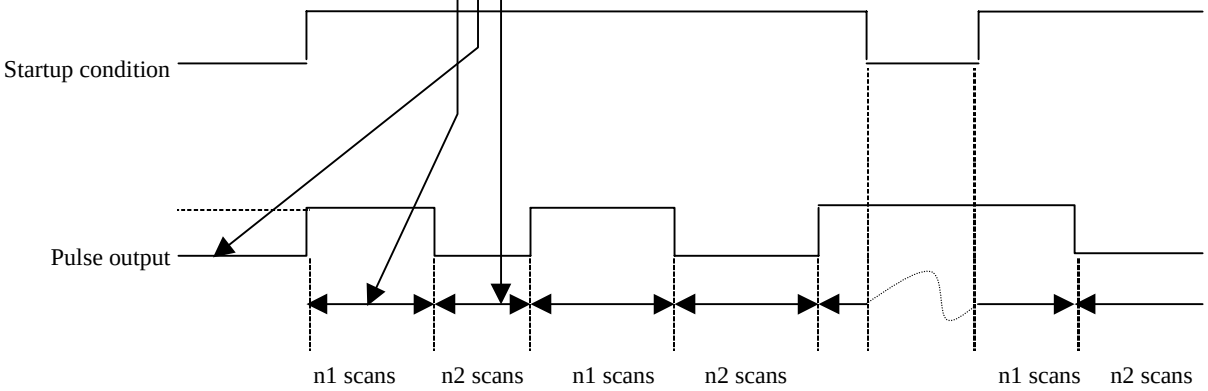
Function

Table

s	n1 of ON scan pulse
s + 1	n2 of OFF scan pulse
s + 2	Pulse output I/O No. a
s + 3	ON scan progress value
s + 4	OFF scan progress value

Relationship between number of scans and pulse activation

Number of scans		Pulse activation
n1	n2	
n1 = 0	n2 = 0	Pulse output turns OFF.
	n2 ≥ 1	
n1 ≥ 1	n2 = 0	Pulse output turns ON.
	n2 ≥ 1	Pulse output turns ON for n1 scans and turns OFF for n2 scans.



- The following operation is repeated: Turns ON the bit internal output specified by argument s + 2 for the number of scans specified by argument s and turns it OFF for the number of scans specified by argument s + 1.
- If this command is executed several times within one scan, the internal bit output turns ON and OFF according to the number of executions.
- If both s and s + 1 are 0, the output stays OFF.
- If the startup condition turns OFF, the output as well as the progress values in s + 3 and s + 4 are retained.
- Both s + 3 and s + 4 should be cleared at initialization; otherwise the pulse width of the first cycle may change.
- Use the ADRIO command to store an actual address in the pulse output I/O. ADRIO (s + 2, bit internal output)
- If the operation is performed normally, DER is set to "0."

* ( ) indicates the display when the LADDER EDITOR is used.

†: Supported by EH-CPU ***A/448/516/548 only.

Cautionary notes

- The pulse activation may at most be delayed by one scan in relation to the pulse output startup condition. Note that if the startup condition changes from ON to OFF then to ON, the pulse width of that part may change by ± one scan.
- Use the ADRIO command to store an actual address in the pulse output I/O of s + 2. If not, an error will occur and no operation will be performed
- If s + 1 to s + 4 and the areas specified by s + 2 exceed the maximum I/O number, DER is set to “1” and no operation is performed.

Item number	Fun commands-37	Name	I/O refresh (All points)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 80 (s)	DER	ERR	SD	V	C							
	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps									
	Condition	Steps	69	←	41 156	←	162	←		
FUN 80 (s)	—	3								

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument (dummy)							○						

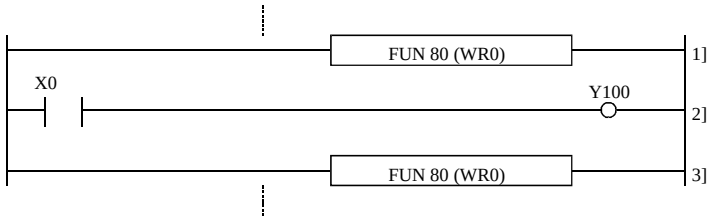
Function

- This command performs I/O refresh of all data in the external I/Os (including link area) in the middle of a scan.

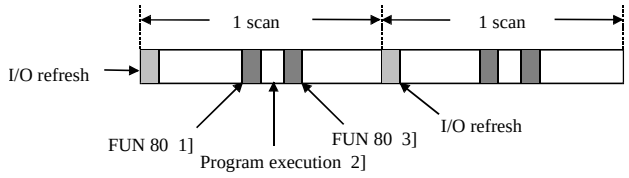
Cautionary notes

- This command performs I/O refresh of all external I/Os (including link area).
  If refresh of certain area is to be performed, use FUN81 or FUN82.
- If the argument s exceeds the maximum I/O number, DER is equal to "1" and the command will not be processed.
- Assign argument s as a one-word dummy. The I/O specified by argument s (WR, WL, WM) will not be affected.

Program example



Program description



Item number	Fun commands-38	Name	I/O refresh (Input/output designation)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 81 (s)	DER	ERR	SD	V	C							
	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps									
	Condition	Steps	44	←	45 94	←	104	←		
FUN 81 (s)	—	3								

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Type							○						

Function

s: Input type

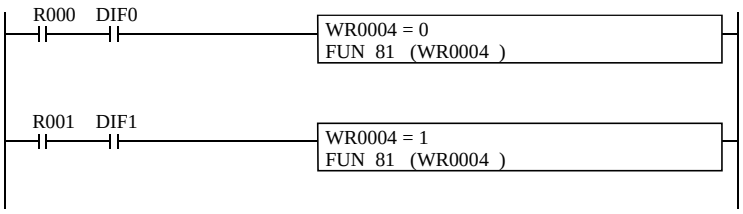
- H00: Input refresh (including remote)
- H01: Output refresh (including remote)
- H02: Link refresh

- Depending on the I/O type of the area specified by s, refresh is performed with respect to I/O modules only, output modules only or link area only.
- Refresh is performed by each slot assignment according to the I/O assignment.
- If the refresh processing is completed normally, DER is set to “0.”

Cautionary notes

- If the I/O type is other than H00, H01 or H02, DER is equal to “1” and the command will not be processed.
- If the argument s exceeds the maximum I/O number, DER is set to “1” and the command will not be processed.

Program example




```
LD    R000
AND  DIF0
[
WR0004 = 0
FUN 81 (WR0004)
]
LD    R001
AND  DIF1
[
WR0004 = 1
FUN 81 (WR0004)
]
```

Program description

- Upon rising of R000, the input module is refreshed.
- Upon rising of R001, the output module is refreshed.

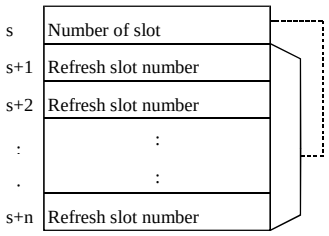
Item number	Fun commands-39	Name	I/O Refresh (slot)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 82 (s)	DER	ERR	SD	V	C							
	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps									
	Condition	Steps								
FUN 82 (s)	—	3	80	←	79 142	←	201	←		

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Number of slot							○						
s+1 and beyond	Slot location number							○						Designate the slot location.

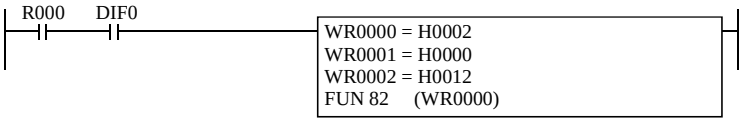
Function



n ≤ 64
Refresh slot number is designated by unit and slot number.

- Designated slot I/O is refreshed.
- The slot location numbers stored in areas s+1 and beyond are designated by the unit number and slot number.
- The maximum slot number is 64. The points exceeding 64 points are not refreshed.
- If refresh processing is completed normally, DER is equal to "0."

Program example




```
LD    R000
AND   DIF0
[
WR0000 = H0002
WR0001 = H0000
WR0002 = H0012
FUN 82     (WR0000)
]
```

Program description

- Upon rising of R000, the two slots designated after WR0001 (unit 0, slot 0) and (unit 1, slot 2) are refreshed.

Cautionary notes

- Set the unit number (0 to 1) and slot number (0 to 7) after s+1. Setting any other value will equal DER to “1” and that slot will not be processed.
- If there is no I/O assignment to the designated slot, DER is equal to “1” and that slot will not be processed.
- If the number of s+n points exceeds the maximum I/O number, DER is equal to “1” and no processing is performed.
- If the number of points exceeds 64, DER is set to 1 and the points exceeding 64 will not be processed (refresh will be performed for up to 64 points).

Slot location number

The slot locations are designated using the unit number and slot number.

The unit number and slot number are set as follows in one word units:

b15	b12	b7	b3	b0
0 to 0	0 to 0	Unit number	Slot number	

Name	Floating-point operation (FUN100 to FUN118) cautionary notes

The following describes some points of caution related to all the FUN commands (FUN100 to FUN 118) for performing floating-point operation. Data for the floating-point commands uses single-precision floating points conforming to IEEE754. The internal representation of IEEE754's single-precision floating-point numbers is explained below.

- Internal representation format of floating point
  Single-precision floating-point numbers are expressed as 32-bit data in the following format.

Contents	Sign bit (S)	Exponent part (E)	Mantissa part (M)
Bit number	b_{31}	b_{30} b_{23}	b_{22} b_0

The 32-bit data in the above table corresponds to internal outputs in the following manner. Examples of representation of internal output I/O numbers when WR100, WR101, and DR100 are used are shown in parentheses..

Word	Upper word (WR101)	Lower word (WR100)
Bit number	b_{15} b_0	b_{15} b_0
Double word	Double word (DR100)	
Bit number	b_{31}	b_0

(1) Sign Bit

Sign bit (S)	Contents
0	Real number
1	Negative number

(2) Exponent Part

Exponent part (E)	Two's exponential value (E')
FF	Indicates overflow value.
FE	127
↓	↓
80	1
7F	0
7E	–1
↓	↓
01	–126
00	Treated as 0.

(3) Mantissa Part

Mantissa part (M)	The value of mantissa part (M')
7FFFFF	$(1.11 \cdots 11)_2$
7FFFFE	$(1.11 \cdots 10)_2$
↓	↓
1	$(1.00 \cdots 01)_2$
0	$(1.00 \cdots 00)_2$

1 in the integer portion of M' in the above table does not appear in the format.

(4) Mathematical Expression

The floating-point number (F) can be expressed with the following formula using the sign bit (S), exponent part (E), and mantissa part (M) listed above.

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

- Range that can be expressed by floating-point numbers

Hexadecimal Expression		Floating Point Expression	Remark
Higher word	Lower word		
H7F7F	HFFFF	$+3.402823 \cdots \times 10^{38}$	Maximum value
H0080	H0000	$+1.175494 \cdots \times 10^{-38}$	The minimum absolute value of a positive number
↓		↓	The value in this range is treated as 0.
H8080	H0000	$-1.175494 \cdots \times 10^{-38}$	The minimum absolute value of a negative number
HFF7F	HFFF	$-3.402823 \cdots \times 10^{38}$	Minimum value

- Example of setting in interval outputs

Interval output		Sign bit	Exponent part	Mantissa part	Floating point
Higher word	Lower word				
H3F80	H0000	0	7F	0	$(1.00 \cdots 00)_2 \times 2^{7FH-7FH} = 1.0$
H4128	H0000	0	82	28	$(1.0101000 \cdots 0)_2 \times 2^{82H-7FH} = 10.5$
HBF00	H0000	1	7E	0	$(-1) \times (1.00 \cdots 00)_2 \times 2^{7EH-7FH} = -0.5$
H3F00	H0000	0	7E	0	$(1.00 \cdots 00)_2 \times 2^{7EH-7FH} = 0.5$

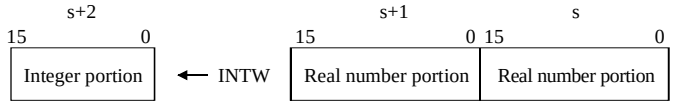
Item number	Fun commands-40	Name	Floating Point Operation † (Real to Integer (Word) Conversion)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 100 (s)	DER	ERR	SD	V	C							
* (INTW (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps										
	Condition	Steps	57	←	58	←	107	←			
FUN 100 (s)	—	3			104	←					
* (INTW (s))											

Usable I/O	Bit				Word				Double word				Other
	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM	Constant	
s Argument							○						s uses up to s+2

Function



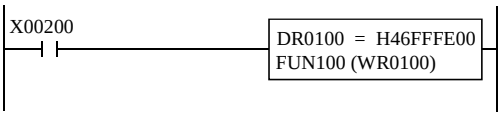
- Converts the real number specified by arguments s and s+1 to integer word data, then sets the result in s+2.
- If the calculation is completed normally, DER is equal to “0.”
- The floating point format conforms to IEEE754.

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- When the resulting integer value of the conversion of the real number specified in s and s+1 falls outside the range of –32,768 to 32,767, DER is set to "1" and s+2 does not change.
- If s to s+2 exceeds the maximum value of the I/O number, DER is set to "1" and no operation is performed.

Program example




```
LD      X00200
[
DR0100 = H46FFFE00
FUN 100 (WR0100)
]
```

Program description

- At a rising edge of X0200, the real number specified in DR0100 (WR0100, WR0101) is converted to an integer and the result is set in WR0102.

Internal output setting : WR0101 = H46FF, WR0100 = HFE00

Operation result : WR0102 = H7FFF

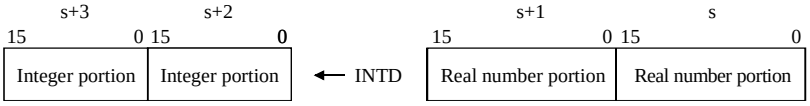
†: Supported by EH-CPU 308(A)/316(A)/448/516/548 only.

Item number	Fun commands-41	Name	Floating Point Operation † (Real to Integer (Double Word) Conversion)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU3**A EH-CPU3**		Other than left		
FUN 101 (s)	DER	ERR	SD	V	C							
* (INTD (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					69	←	69	←	123	←	
	Condition			Steps								
FUN 101 (s)	—			3				119	←			
* (INTD (s))												

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument							○						s uses up to s+3

Function



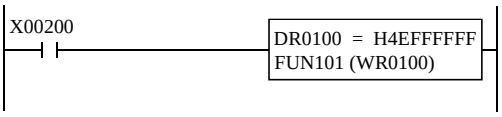
- Converts the real number specified by arguments s and s+1 to double word data, then sets the result in s+2 and s+3.
- If the calculation is completed normally, DER is equal to “0.”
- The floating point format conforms to IEEE754.

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- When the resulting integer value of the conversion of the real number specified in s and s+1 falls outside the range of –2,147,483,648 to 2,147,483,647, DER is set to "1, " and s+2 and s+3 do not change.
- If s to s+3 exceeds the maximum value of the I/O number, DER is set to "1" and no operation is performed.

Program example




```
LD      X00200
[
DR0100 = H4EFFFFFF
FUN 101 (WR0100)
]
```

Program description

- At a rising edge of X0200, the real number specified in DR0100 (WR0100, WR0101) is converted to an integer and the result is set in DR0102 (WR0102, WR0103).

Internal output setting : WR0101 = H4EFF, WR0100 = HFFFF

Operation result : WR0103 = H7FFF, WR0102 = HFF80

† : Supported by EH-CPU 308(A)/316(A)/448/516/548 only.

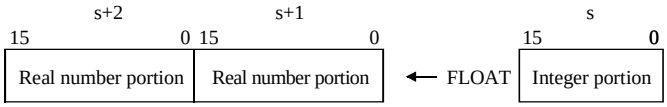
Item number	Fun commands-42	Name	Floating Point Operation † (Integer (Word) to Real Number Conversion)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU3**A EH-CPU3**		Other than left		
FUN 102 (s)	DER	ERR	SD	V	C							
* (FLOAT (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps										
	Condition	Steps	42	←	43		93	←			
FUN 102 (s)	—	3			89	←					
* (FLOAT (s))											

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument							○						s uses up to s+2

Function



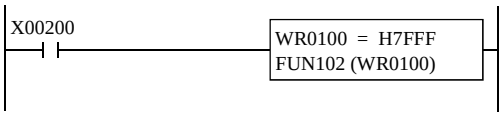
- Converts the integer word data s to a real number, then sets the result in s+1 and s+2.
- If the calculation is completed normally, DER is equal to “0.”
- The floating point format conforms to IEEE754.

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- An integer value in the range of –32,768 to 32,767 can be set for s and s+1.
- If s to s+2 exceeds the maximum value of the I/O number, DER is set to "1" and no operation is performed.

Program example




```
LD      X00200
[
DR0100 = H7FFF
FUN 102 (WR0100)
]
```

Program description

- At a rising edge of X0200, the integer specified in WR0100 is converted to a real number and the result is set in DR0101 (WR0101, WR0102).

Internal output setting : WR0100 = H7FFF

Operation result : WR0102 = H46FF, WR0101 = HFE00

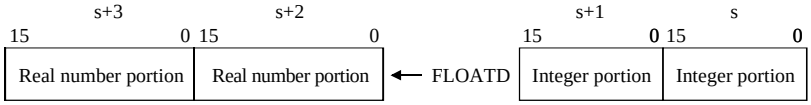
†: Supported by EH-CPU 308(A)/316(A)/448/516/548 only.

Item number	Fun commands-43	Name	Floating Point Operation † (Integer (Double Word) to Real Number Conversion)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU3**A EH-CPU3**		Other than left		
FUN 103 (s)	DER	ERR	SD	V	C							
* (FLOATD (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					47	←	47	←	99	←	
	Condition		Steps									
FUN 103 (s)	—		3					96	←			
* (FLOATD (s))												

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument							○						s uses up to s+3

Function



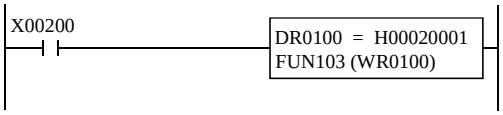
- Converts the integer double word data s and s+1 to a real number, then sets the result s+2 and s+3.
- If the calculation is completed normally, DER is equal to "0."
- The floating point format conforms to IEEE754.

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- An integer value in the range of –2,147,483,648 to 2,147,483,647 can be set for s and s+1.
- If s to s+3 exceeds the maximum value of the I/O number, DER is set to "1" and no operation is performed.

Program example




```
LD      X00200
[
DR0100 = H00020001
FUN 103 (WR0100)
]
```

Program description

- At a rising edge of X0200, the integer specified in DR0100 (WR0100, WR0101) is converted to a real number and the result is set in DR0102 (WR0102, WR0103).

Internal output setting: WR0101 = H0002, WR0100 = H0001

Operation result: WR0103 = H4800, WR0102 = H0040

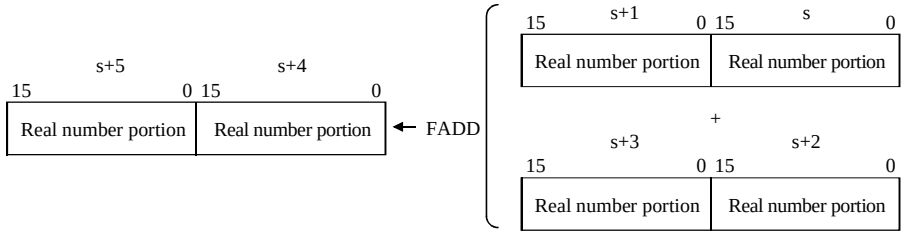
†: Supported by EH-CPU 308(A)/316(A)/448/516/548 only.

Item number	Fun commands-44	Name	Floating Point Operation † (Addition)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU3**A EH-CPU3**		Other than left		
FUN 104 (s)	DER	ERR	SD	V	C							
* (FADD (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					70	←	70	←	125	←	
	Condition			Steps								
FUN 104 (s)	—			3				122	←			
* (FADD (s))												

Usable I/O	Bit				Word				Double word				Other
	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM	Constant	
s Argument							○						s uses up to s+5

Function



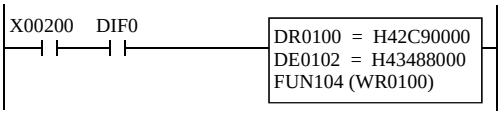
- Adds the real number (s+2, s+3) to the real number (s, s+1), then sets the result in (s+4, s+5).
- If the calculation is completed normally, DER is equal to “0.”
- The floating point format conforms to IEEE754.

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- When the operation result is not within the range of –1e+37 to 1e+37, DER is set to "1."
- If s to s+5 exceeds the maximum value of the I/O number, DER is set to "1" and no operation is performed.

Program example




```
LD      X00200
AND DIF0
[
DR0100 = H42C90000
DR0102 = H43488000
FUN 104 (WR0100)
]
```

Program description

- At a rising edge of X0200, the real number specified in DR0100 (WR0100, WR0101) is added to the real number specified in DR0102 (WR0102, WR0103), and the result is set in DR0104 (WR0104, WR0105).

Internal output setting : WR0101 = H42C9, WR0100 = H0000
WR0103 = H4348, WR0102 = H8000

Operation result : WR0105 = H4396, WR0104 = H8000

†: Supported by EH-CPU 308(A)/316(A)/448/516/548 only.

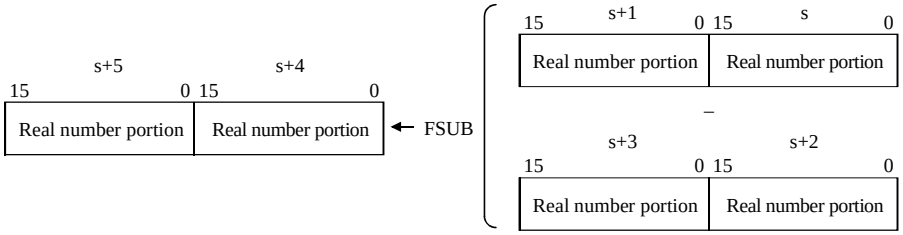
Item number	Fun commands-45	Name	Floating Point Operation † (Subtraction)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU3**A EH-CPU3**		Other than left		
FUN 105 (s)	DER	ERR	SD	V	C							
* (FSUB (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps		Processing time (μs)						Remark
	Condition	Steps							
FUN 105 (s)	—	3	70	←	70	←	125	←	
* (FSUB (s))					121	←			

Usable I/O		Bit				Word				Double word				Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM	Constant	
s	Argument							○						s uses up to s+5

Function



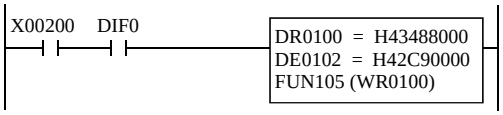
- Subtracts the real number (s+2, s+3) from the real number (s, s+1), then sets the result in (s+4, s+5).
- If the calculation is completed normally, DER is equal to “0.”
- The floating point format conforms to IEEE754.

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- When the operation result is not within the range of –1e+37 to 1e+37, DER is set to "1."
- If s to s+5 exceeds the maximum value of the I/O number, DER is set to "1" and no operation is performed.

Program example




```
LD      X00200
AND DIF0
[
DR0100 = H43488000
DR0102 = H42C90000
FUN 105 (WR0100)
]
```

Program description

- At a rising edge of X0200, the real number specified in DR0102 (WR0102, WR0103) is subtracted from the real number specified in DR0100 (WR0100, WR0101), and the result is set in DR0104 (WR0104, WR0105).

Internal output setting : WR0101 = H4348, WR0100 = H8000
WR0103 = H42C9, WR0102 = H0000

Operation result : WR0105 = H42C8, WR0104 = H0000

† : Supported by EH-CPU 308(A)/316(A)/448/516/548 only.

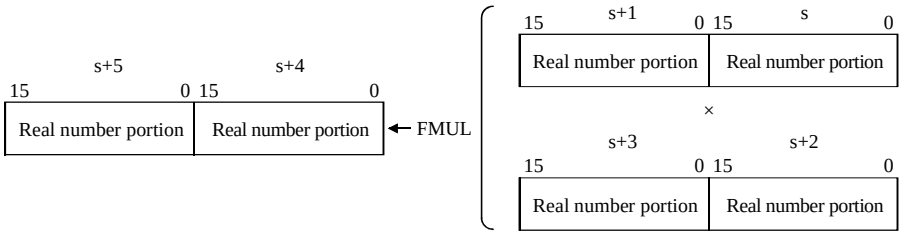
Item number	Fun commands-46	Name	Floating Point Operation † (Multiplication)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU3**A EH-CPU3**		Other than left		
FUN 106 (s)	DER	ERR	SD	V	C							
* (FMUL (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps										
	Condition	Steps	69	←	70	←	125	←			
FUN 106 (s)	—	3			121	←					
* (FMUL (s))											

Usable I/O	Bit				Word				Double word			Constant	Other
	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s Argument							○						s uses up to s+5

Function



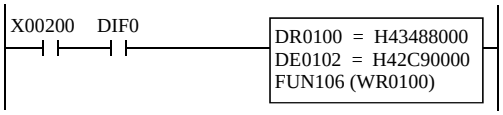
- Multiplies the real number (s, s+1) with the real number (s+2, s+3), then sets the result in (s+4, s+5).
- If the calculation is completed normally, DER is equal to “0.”
- The floating point format conforms to IEEE754.

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- When the operation result is not within the range of –1e+37 to 1e+37, DER is set to "1."
- If s to s+5 exceeds the maximum value of the I/O number, DER is set to "1" and no operation is performed.

Program example




```
LD      X00200
AND DIF0
[
DR0100 = H43488000
DR0102 = H42C90000
FUN 106 (WR0100)
]
```

Program description

- At a rising edge of X0200, the real number specified in DR0100 (WR0100, WR0101) is multiplied by the real number specified in DR0102 (WR0102, WR0103), and the result is set in DR0104 (WR0104, WR0105).

Internal output setting : WR0101 = H4348, WR0100 = H8000
WR0103 = H42C9, WR0102 = H0000

Operation result : WR0105 = H469D, WR0104 = H6C80

†: Supported by EH-CPU 308(A)/316(A)/448/516/548 only.

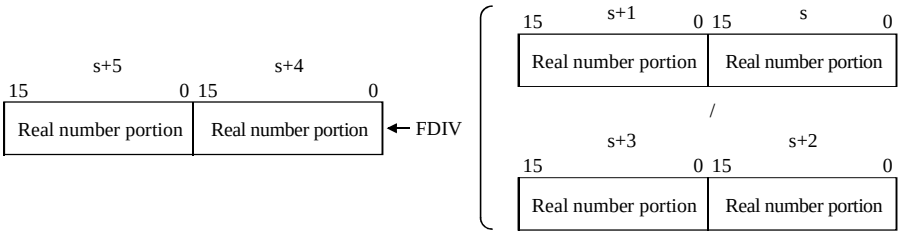
Item number	Fun commands-47	Name	Floating Point Operation † (Division)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU3**A EH-CPU3**		Other than left		
FUN 107 (s)	DER	ERR	SD	V	C							
* (FDIV (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps											
	Condition	Steps	119	←	119	←	174	←				
FUN 107 (s)	—	3			171	←						
* (FDIV (s))												

Usable I/O	Bit				Word				Double word				Other
	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM	Constant	
s Argument							○						s uses up to s+5

Function



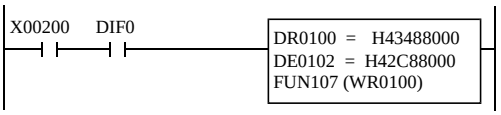
- Divides real number (s, s+1) by real number (s+2, s+3), then sets the result in (s+4, s+5).
- If the calculation is completed normally, DER is equal to “0.”
- The floating point format conforms to IEEE754.

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- When the operation result is not within the range of –1e+37 to 1e+37, DER is set to "1."
- If s to s+5 exceeds the maximum value of the I/O number, DER is set to "1" and no operation is performed.

Program example




```
LD      X00200
AND DIF0
[
DR0100 = H43488000
DR0102 = H42C88000
FUN 107 (WR0100)
]
```

Program description

- At a rising edge of X0200, the real number specified in DR0100 (WR0100, WR0101) is divided by the real number specified in DR0102 (WR0102, WR0103), and the result is set in DR0104 (WR0104, WR0105).

Internal output setting : WR0101 = H4348, WR0100 = H8000
WR0103 = H42C8, WR0102 = H8000
Operation result : WR0105 = H4000, WR0104 = H0000

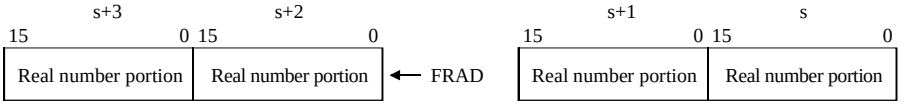
†: Supported by EH-CPU 308(A)/316(A)/448/516/548 only.

Item number	Fun commands-48	Name	Floating Point Operation † (Angle to Radian Conversion)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU3**A EH-CPU3**		Other than left		
FUN 108 (s)	DER	ERR	SD	V	C							
* (FRAD (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					63	←	64	←	117	←	
	Condition			Steps				114	←			
FUN 108 (s) * (FRAD (s))	—			3								

Usable I/O		Bit				Word				Double word				Other	
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM	Constant		
s	Argument							○						s uses up to s+3	

Function



$$\text{degrees} \times \frac{\pi}{180} = \text{radian}$$

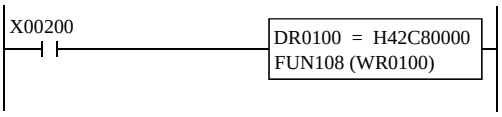
- Converts the angle units of the real number value specified in s and s+1 as the arguments to radian units, the sets the result the result in s+2 and s+3.
- If the calculation is completed normally, DER is equal to “0.”
- The floating point format conforms to IEEE754.

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- When the operation result is not within the range of –1e+37 to 1e+37, DER is set to "1."
- If s to s+3 exceeds the maximum value of the I/O number, DER is set to "1" and no operation is performed.

Program example




```
LD      X00200
[
DR0100 = H42C80000
FUN 108 (WR0100)
]
```

Program description

- At a rising edge of X0200, the real number specified in DR0100 (WR0100, WR0101) is converted to a radian and the result is set in DR0102 (WR0102, WR0103).

Internal output setting : WR0101 = H42C8, WR0100 = H0000

Operation result : WR0103 = H3FDF, WR0102 = H66F3

†: Supported by EH-CPU 308(A)/316(A)/448/516/548 only.

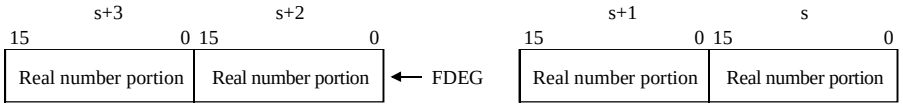
Item number	Fun commands-49	Name	Floating Point Operation † (Radian to Angle Conversion)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU3**A EH-CPU3**		Other than left		
FUN 109 (s)	DER	ERR	SD	V	C							
* (FDEG (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps										
	Condition	Steps	63	←	64	←	117	←			
FUN 109 (s)	—	3			114	←					
* (FDEG (s))											

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument							○						s uses up to s+3

Function



$$\text{radian} \times \frac{180}{\pi} = \text{degrees}$$

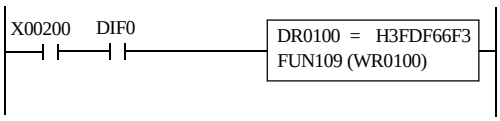
- Converts the radian units of the real number value specified in s and s+1 as the arguments to angle units, then sets the result in s+2 and s+3.
- If the calculation is completed normally, DER is equal to “0.”
- The floating point format conforms to IEEE754.

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- When the operation result is not within the range of –1e+37 to 1e+37, DER is set to "1."
- If s to s+3 exceeds the maximum value of the I/O number, DER is set to "1" and no operation is performed.

Program example




```
LD      X00200
AND DIF0
[
DR0100 = H3FDF66F3
FUN 109 (WR0100)
]
```

Program description

- At a rising edge of X0200, the real number specified in DR0100 (WR0100, WR0101) is converted to an angle and the result is set in DR0102 (WR0102, WR0103).

Internal output setting : WR0101 = H3FDF, WR0100 = H66F3

Operation result : WR0103 = H42C8, WR0102 = H0000

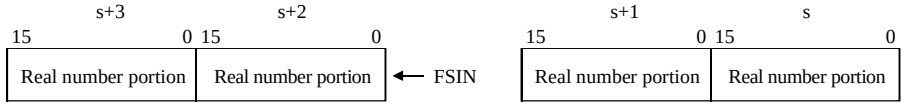
†: Supported by EH-CPU 308(A)/316(A)/448/516/548 only.

Item number	Fun commands-50	Name	Floating Point Operation † (SIN)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU3**A EH-CPU3**		Other than left		
FUN 110 (s)	DER	ERR	SD	V	C							
* (FSIN (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					281	←	282	←	358	←	
	Condition			Steps				354	←			
FUN 110 (s) * (FSIN (s))	—			3								

Usable I/O	Bit				Word				Double word				Other
	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM	Constant	
s Argument							○						s uses up to s+3

Function



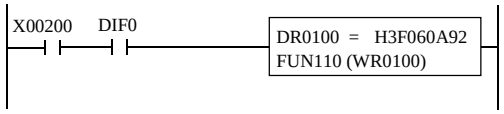
- Calculates the sine value of the real number value in radian units specified in s and s+1 as the arguments, then sets the result in s+2 and s+3.
- If the calculation is completed normally, DER is equal to “0.”
- The floating point format conforms to IEEE754.

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- When the operation result is not within the range of –1e+37 to 1e+37, DER is set to "1."
- If s to s+3 exceeds the maximum value of the I/O number, DER is set to "1" and no operation is performed.
- When the value of s, s+1 is greater than 1.414847550405688000e+16, the sine value cannot be calculated, thus DER is set to "1."
- When the value of s, s+1 is greater than 2.981568260000000000e+08, a result is obtained but the accuracy decreases, so DER is set to "1."

Program example




```
LD      X00200
AND DIF0
[
DR0100 = H3F060A92
FUN 110 (WR0100)
]
```

Program description

- At a rising edge of X0200, the SIN of the real number specified in DR0100 (WR0100, WR0101) is calculated and the result is set in DR0102 (WR0102, WR0103).

Internal output setting : WR0101 = H3F06, WR0100 = H0A92

Operation result : WR0103 = H3F00, WR0102 = H0000

†: Supported by EH-CPU 308(A)/316(A)/448/516/548 only.

Item number	Fun commands-51	Name	Floating Point Operation † (COS)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU3**A EH-CPU3**		Other than left		
FUN 111 (s)	DER	ERR	SD	V	C							
* (FCOS (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps										
	Condition	Steps	303	←	304	←	383	←			
FUN 111 (s)	—	3			380	←					
* (FCOS (s))											

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument							○						s uses up to s+3

Function

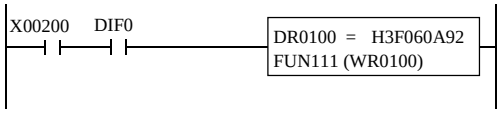
- Calculates the cosine value of the real number value in radian units specified in s and s+1 as the arguments, the sets the result in s+2 and s+3.
- If the calculation is completed normally, DER is equal to “0.”
- The floating point format conforms to IEEE754.

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- When the operation result is not within the range of –1e+37 to 1e+37, DER is set to "1."
- If s to s+3 exceeds the maximum value of the I/O number, DER is set to "1" and no operation is performed.
- When the value of s, s+1 is greater than 1.414847550405688000e+16, the cosine value cannot be calculated and DER is set to "1."
- When the value of s, s+1 is greater than 2.981568260000000000e+08, a result is obtained but the accuracy decreases, so DER is set to "1."

Program example




```
LD      X00200
AND DIF0
[
DR0100 = H3F060A92
FUN 111 (WR0100)
]
```

Program description

- At a rising edge of X0200, the cosine value of the real number specified in DR0100 (WR0100, WR0101) is calculated and the result is set in DR0102 (WR0102, WR0103).

Internal output setting : WR0101 = H3F06, WR0100 = H0A92

Operation result : WR0103 = H3F5D, WR0102 = HB3D7

†: Supported by EH-CPU 308(A)/316(A)/448/516/548 only.

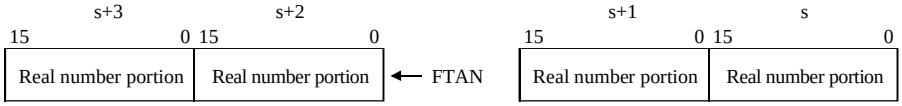
Item number	Fun commands-52	Name	Floating Point Operation † (TAN)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU3**A EH-CPU3**		Other than left		
FUN 112 (s)	DER	ERR	SD	V	C							
* (FTAN (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps										
	Condition	Steps	343	←	341	←	417	←			
FUN 112 (s)	—	3			414	←					
* (FTAN (s))											

Usable I/O	Bit				Word				Double word			Constant	Other
	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s Argument							○						s uses up to s+3

Function



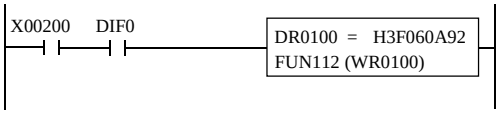
- Calculates the tangent value of the real number value in radian units specified in s and s+1 as the arguments, the sets the result in s+2 and s+3.
- If the calculation is completed normally, DER is equal to “0.”
- The floating point format conforms to IEEE754.

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- When the operation result is not within the range of –1e+37 to 1e+37, DER is set to "1."
- If s to s+3 exceeds the maximum value of the I/O number, DER is set to "1" and no operation is performed.
- When the value of s, s+1 is greater than 1.414847550405688000e+16/2, the tangent value cannot be calculated and DER is set to "1."
- When the value of s, s+1 is greater than 2.981568260000000000e+08/2, a result is obtained but the accuracy decreases, so DER is set to "1."

Program example




```
LD      X00200
AND DIF0
[
DR0100 = H3F060A92
FUN 112 (WR0100)
]
```

Program description

- At a rising edge of X0200, the tangent value of the real number specified in DR0100 (WR0100, WR0101) is calculated and the result is set in DR0102 (WR0102, WR0103).

Internal output setting : WR0101 = H3F06, WR0100 = H0A92

Operation result : WR0103 = H3F13, WR0102 = HCD3A

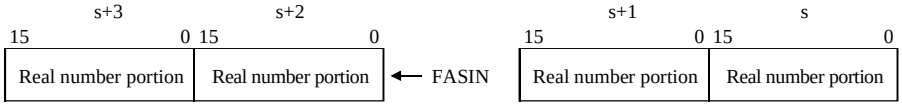
† : Supported by EH-CPU 308(A)/316(A)/448/516/548 only.

Item number	Fun commands-53	Name	Floating Point Operation † (ARC SIN)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU3**A EH-CPU3**		Other than left		
FUN 113 (s)	DER	ERR	SD	V	C							
* (FASIN (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					200	←	200	←	278	←	
	Condition			Steps				275	←			
FUN 113 (s) * (FASIN (s))	—			3								

Usable I/O	Bit				Word				Double word				Other
	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM	Constant	
s Argument							○						s uses up to s+3

Function



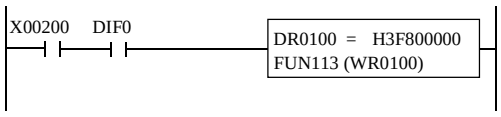
- Calculates the SIN $^{-1}$ value of the real number value specified in s and s+1 as the arguments, and sets the result in radian units in s+2 and s+3.
- If the calculation is completed normally, DER is equal to "0."
- The floating point format conforms to IEEE754.

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- When the operation result is not within the range of –1e+37 to 1e+37, DER is set to "1."
- If s to s+3 exceeds the maximum value of the I/O number, DER is set to "1" and no operation is performed.
- When the value of s, s+1 is greater than 1, DER is set to "1."

Program example




```
LD      X00200
AND DIF0
[
DR0100 = H3F800000
FUN 113 (WR0100)
]
```

Program description

- At a rising edge of X0200, the SIN $^{-1}$ value of the real number specified in DR0100 (WR0100, WR0101) is calculated and the result is set in DR0102 (WR0102, WR0103).

Internal output setting : WR0101 = H3F80, WR0100 = H0000

Operation result : WR0103 = H3FC9, WR0102 = H0FDB

† : Supported by EH-CPU 308(A)/316(A)/448/516/548 only.

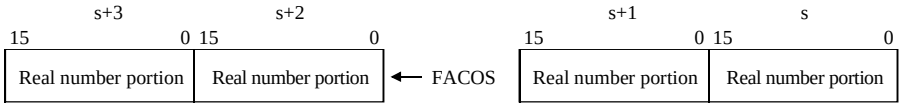
Item number	Fun commands-54	Name	Floating Point Operation † (ARC COS)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU3**A EH-CPU3**		Other than left		
FUN 114 (s)	DER	ERR	SD	V	C							
* (FACOS (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps										
	Condition	Steps	191	←	191	←	269	←			
FUN 114 (s)	—	3			266	←					
* (FACOS (s))											

Usable I/O	Bit				Word				Double word			Constant	Other
	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s Argument							○						s uses up to s+3

Function



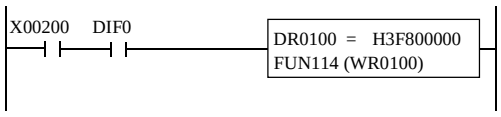
- Calculates the COS $^{-1}$ value of the real number value specified in s and s+1 as the arguments, and sets the result in radian units in s+2 and s+3.
- If the calculation is completed normally, DER is equal to "0."
- The floating point format conforms to IEEE754.

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- When the operation result is not within the range of –1e+37 to 1e+37, DER is set to "1."
- If s to s+3 exceeds the maximum value of the I/O number, DER is set to "1" and no operation is performed.
- When the value of s, s+1 is greater than 1, DER is set to "1."

Program example




```
LD      X00200
AND DIF0
[
DR0100 = H3F800000
FUN 114 (WR0100)
]
```

Program description

- At a rising edge of X0200, the COS $^{-1}$ value of the real number specified in DR0100 (WR0100, WR0101) is calculated and the result is set in DR0102 (WR0102, WR0103).

Internal output setting : WR0101 = H3F80, WR0100 = H0000

Operation result : WR0103 = H0000, WR0102 = H0000

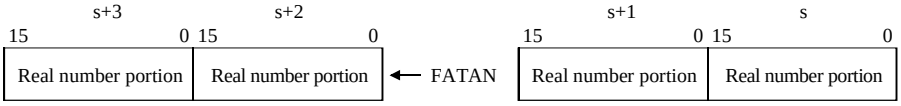
† : Supported by EH-CPU 308(A)/316(A)/448/516/548 only.

Item number	Fun commands-55	Name	Floating Point Operation † (ARC TAN)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU3**A EH-CPU3**		Other than left		
FUN 115 (s)	DER	ERR	SD	V	C							
* (FATAN (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					394	←	395	←	466	←	
	Condition			Steps								
FUN 115 (s)	—			3				463	←			
* (FATAN (s))												

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument							○						s uses up to s+3

Function



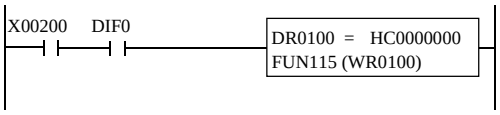
- Calculates the TAN $^{-1}$ value of the real number value specified in s and s+1 as the arguments, and sets the result in radian units in s+2 and s+3.
- If the calculation is completed normally, DER is equal to “0.”
- The floating point format conforms to IEEE754.

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- When the operation result is not within the range of –1e+37 to 1e+37, DER is set to "1."
- If s to s+3 exceeds the maximum value of the I/O number, DER is set to "1" and no operation is performed.

Program example




```
LD      X00200
AND DIF0
[
DR0100 = HC0000000
FUN 115 (WR0100)
]
```

Program description

- At a rising edge of X0200, the TAN $^{-1}$ value of the real number specified in DR0100 (WR0100, WR0101) is calculated and the result is set in DR0102 (WR0102, WR0103).

Internal output setting : WR0101 = HC000, WR0100 = H0000

Operation result : WR0103 = HBF8D, WR0102 = HB70D

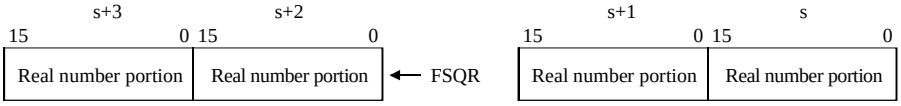
† : Supported by EH-CPU 308(A)/316(A)/448/516/548 only.

Item number	Fun commands-56	Name	Floating Point Operation † (Square Root)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU3**A EH-CPU3**		Other than left		
FUN 116 (s)	DER	ERR	SD	V	C							
* (FSQR (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					481	←	482	←	556	←	
	Condition			Steps								
FUN 116 (s)	—			3				553	←			
* (FSQR (s))												

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument							○						s uses up to s+3

Function



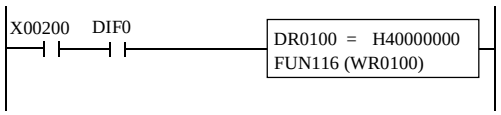
- Calculates the square root using the real number value specified in s and s+1 as the arguments, the sets the result in s+2 and s+3.
- If the calculation is completed normally, DER is equal to “0.”
- The floating point format conforms to IEEE754.

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- When the operation result is not within the range of –1e+37 to 1e+37, DER is set to "1."
- If s to s+3 exceeds the maximum value of the I/O number, DER is set to "1" and no operation is performed.
- When the value of s, s+1 is lower than 0, the value cannot be calculated and DER is set to "1."

Program example




```
LD      X00200
AND DIF0
[
DR0100 = H40000000
FUN 116 (WR0100)
]
```

Program description

- At a rising edge of X0200, the square root of the real number specified in DR0100 (WR0100, WR0101) is calculated and the result is set in DR0102 (WR0102, WR0103).

Internal output setting : WR0101 = H4000, WR0100 = H0000

Operation result : WR0103 = H3FB5, WR0102 = H04F3

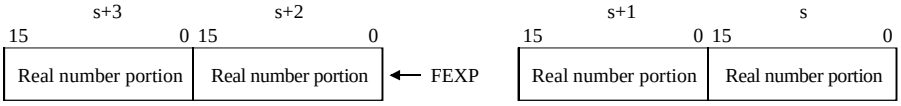
†: Supported by EH-CPU 308(A)/316(A)/448/516/548 only.

Item number	Fun commands-57	Name	Floating Point Operation † (Exponent)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU3**A EH-CPU3**		Other than left		
FUN 117 (s)	DER	ERR	SD	V	C							
* (FEXP (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					329	←	330	←	399	←	
	Condition			Steps								
FUN 117 (s)	—			3				395	←			
* (FEXP (s))												

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument							○						s uses up to s+3

Function



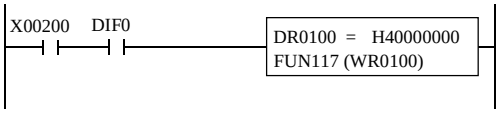
- Performs an exponent operation using the real number value specified in s and s+1 as the arguments, the sets the result in s+2 and s+3.
- An exponent operation is performed using 2.71828 as the base (e).
- If the calculation is completed normally, DER is equal to “0.”
- The floating point format conforms to IEEE754.

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- When the operation result is not within the range of –1e+37 to 1e+37, DER is set to "1."
- If s to s+3 exceeds the maximum value of the I/O number, DER is set to "1" and no operation is performed.
- Calculation cannot be performed when the value of s, s+1 is lower than –7.0839639e+02. In this case, DER is set to "1."

Program example




```
LD      X00200
AND DIF0
[
DR0100 = H40000000
FUN 117 (WR0100)
]
```

Program description

- At a rising edge of X0200, an exponent operation of the real number specified in DR0100 (WR0100, WR0101) is performed and the result is set in DR0102 (WR0102, WR0103).

Internal output setting : WR0101 = H4000, WR0100 = H0000

Operation result : WR0103 = H40EC, WR0102 = H7326

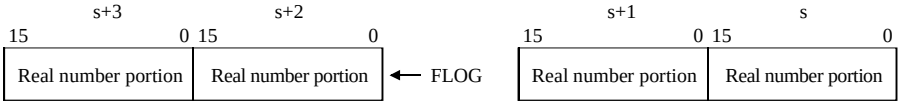
†: Supported by EH-CPU 308(A)/316(A)/448/516/548 only.

Item number	Fun commands-58	Name	Floating Point Operation † (Natural Logarithm)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU3**A EH-CPU3**		Other than left		
FUN 118 (s)	DER	ERR	SD	V	C							
* (FLOG (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					187	←	187	←	257	←	
	Condition			Steps								
FUN 118 (s)	—			3				254	←			
* (FLOG (s))												

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument							○						s uses up to s+3

Function



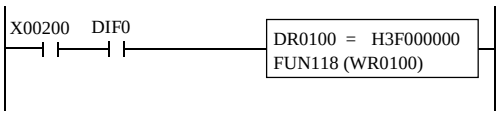
- Performs a logarithm operation for the real number value specified by arguments s and s+1 using the natural logarithm (e) as the base, then sets the result in s+2 and s+3.
- If the calculation is completed normally, DER is equal to “0.”
- The floating point format conforms to IEEE754.

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- When the operation result is not within the range of –1e+37 to 1e+37, DER is set to "1."
- If s to s+3 exceeds the maximum value of the I/O number, DER is set to "1" and no operation is performed.
- Calculation cannot be performed when the value of s, s+1 is lower than or equal to 0. In this case, DER is set to "1."

Program example




```
LD      X00200
AND DIF0
[
DR0100 = H3F000000
FUN 118 (WR0100)
]
```

Program description

- At a rising edge of X0200, the logarithm operation of the real number specified in DR0100 (WR0100, WR0101) is performed and the result is set in DR0102 (WR0102, WR0103).

Internal output setting : WR0101 = H3F00, WR0100 = H0000

Operation result : WR0103 = HBF31, WR0102 = H7218

† : Supported by EH-CPU 308(A)/316(A)/448/516/548 only.

Item number	Fun commands-59	Name	Index setting † (argument d)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 120 (s)	DER	ERR	SD	V	C							
* (INDXD (s))	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					18	←	22	←			
	Condition			Steps								
FUN 120 (s) * (INDXD (s))	—			3								

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DN, DM		
s	Index I/O No.							○						

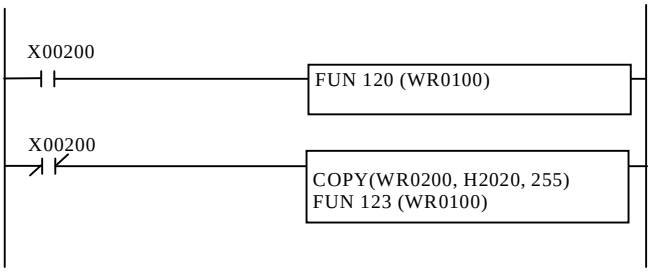
Function

- Sets the argument used as an index in respect to argument d of the MOV and COPY commands.

Cautionary notes

- The I/O value specified in the Index Area will be initialized to “0.”
- Index qualification will be performed to both the MOV and COPY commands.
- This does not check for duplicate index numbers.

Program example



Sets the WR0100 as an index in respect to argument d.

Performs copy processing after the index (WR0100) is added in respect to argument d of the COPY command (WR0200).
After the COPY command is executed, the index will be updated (+1) (FUN 123).

†: Supported by EH-CPU ***A/448/516/548 only.

Item number	Fun commands-60	Name	Index setting † (argument s)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 121 (s)	DER	ERR	SD	V	C							
* (INDXS (s))	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					18	←	22	←			
	Condition			Steps								
FUN 121 (s) * (INDXS (s))	—			3								

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DN, DM		
s	Index I/O No.							○						

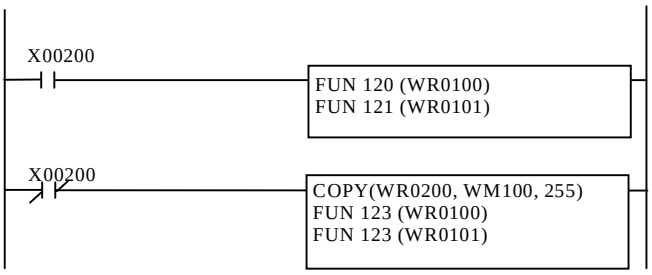
Function

- Sets the argument used as an index in respect to argument s of the MOV and COPY commands.

Cautionary notes

- The I/O value specified in the Index Area will be initialized to “0.”
- Index qualification will be performed to both the MOV and COPY commands.
- This does not check for duplicate index numbers.

Program example



Sets WR0100 as an index in respect to argument d and sets WR0101 as an index in respect to argument s.

Adds an index (WR0100) in respect to argument d (WR0200) of the COPY command. Then adds an index (WR0101) in respect to argument s (WM100). After the COPY command is executed, the index will be updated (+1) (FUN 123).

†: Supported by EH-CPU ***A/448/516/548 only.

Item number	Fun commands-61	Name	Index cancel †

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 122 (s)	DER	ERR	SD	V	C							
* (INDXC (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					18	←	21	←			
	Condition			Steps								
FUN 122 (s) * (INDXC (s))	—			3								

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DN, DM		
s	Index I/O No.							○						

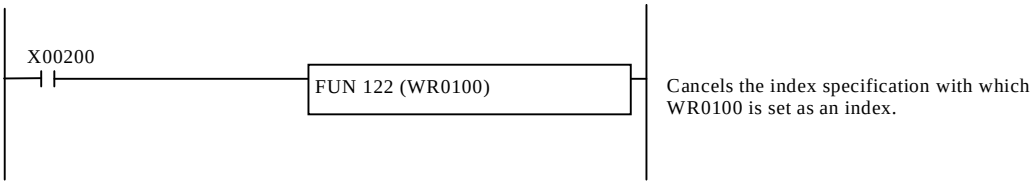
Function

- Cancels the index specification that is set for argument d or s of the MOV and COPY commands.
- Sets the I/O number specified by FUN 120 or FUN 121 to s.

Cautionary notes

- Cancels the index specification for argument d or s in respect to both the MOV and COPY commands.
- If the I/O number set in s is not specified as an index, DER will be equal to “1” and no processing will be performed.

Program example



† : Supported by EH-CPU ***A/448/516/548 only.

Item number	Fun commands-62	Name	Increment † (INC)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 123 (s)	DER	ERR	SD	V	C							
* (INC (s))	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					29	←	30	←	77	←	
	Condition			Steps								
FUN 123 (s)	—			3				74	←			
* (INC (s))												

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument							○						

Function

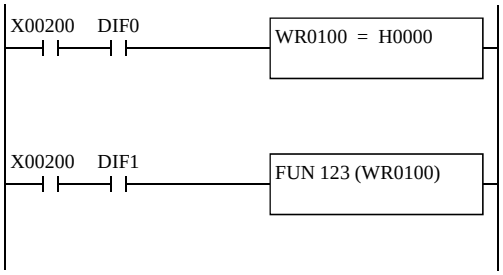
- Increments the word I/O specified in WR, WL, WM by 1 (+1).

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- When FFFFH is increased by 1 (+1), the result will be 0000H.

Program example




```
LD      X00200
AND DIF0
[
WR0100 = H0000
]
LD      X00200
AND DIF1
[
FUN 123 (WR0100)
]
```

Program description

- At a rising edge of X0200, the WR0100 is incremented by 1 (+1).

Internal output setting : WR0100 = H0000

Operation result : WR0100 = H0001

†: Supported by EH-CPU ***A/448/516/548 only.

Item number	Fun commands-63	Name	Double Word Increment † (INCD)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 124 (s)	DER	ERR	SD	V	C							
* (INCD (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					42	←	42	←	94	←	
	Condition			Steps								
FUN 124 (s)	—			3				91	←			
* (INCD (s))												

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument							○						s uses up to s+1

Function

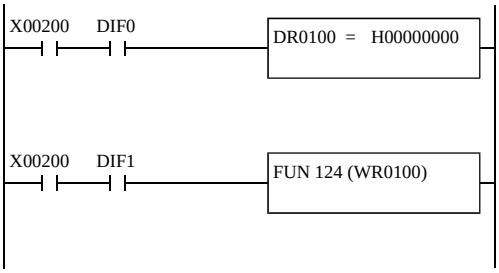
- Replaces the specified I/O in WR, WL or WM with the double word I/O in DR, DL or DM, and increments the double word I/O by 1 (+1).

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- When the area specified by s exceeds the maximum value of I/O number, DER is set to "1" and no operation is performed.
- When FFFFFFFFH is increased by 1 (+1), the result will be 00000000H.

Program example




```
LD      X00200
AND DIF0
[
DR0100 = H00000000
]

LD      X00200
AND DIF1
[
FUN 124 (WR0100)
]
```

Program description

- At a rising edge of X0200, the DR0100 is incremented by 1 (+1).

Internal output setting : DR0100 = H00000000

Operation result : DR0100 = H00000001

†: Supported by EH-CPU ***A/448/516/548 only.

Item number	Fun commands-64	Name	Decrement † (DEC)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 125 (s)	DER	ERR	SD	V	C							
* (DEC (s))	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					30	←	30	←	77	←	
	Condition			Steps								
FUN 125 (s)	—			3				73	←			
* (DEC (s))												

Usable I/O	Bit				Word				Double word			Constant	Other
	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s Argument							○						

Function

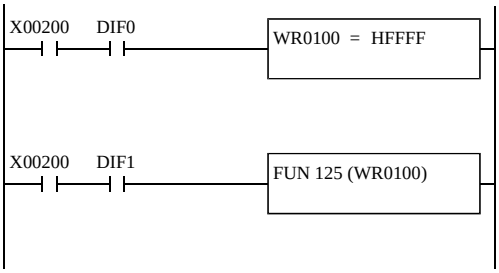
- Decrements the word I/O specified in WR, WL or WM by 1 (–1).

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- When 0000H is decreased by 1 (–1), the result will be FFFFH.

Program example




```
LD      X00200
AND DIF0
[
WR0100 = HFFFF
]
LD      X00200
AND DIF1
[
FUN 125 (WR0100)
]
```

Program description

- At a rising edge of X0200, the WR0100 is decremented by 1 (–1).

Internal output setting : WR0100 = HFFFF

Operation result : WR0100 = HFFFE

† : Supported by EH-CPU ***A/448/516/548 only.

Item number	Fun commands-65	Name	Double Word Decrement † (DECD)

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 126 (s)	DER	ERR	SD	V	C							
* (DECD (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					41	←	42	←	94	←	
	Condition			Steps								
FUN 126 (s)	—			3				91	←			
* (DECD (s))												

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument							○						s uses up to s+1

Function

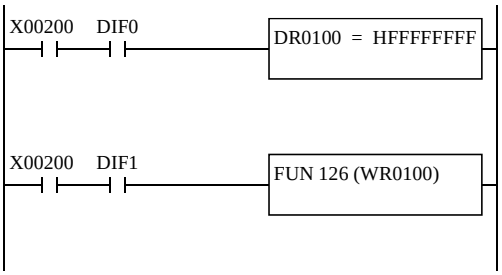
- Replaces the specified I/O in WR, WL or WM with the double word I/O in DR, DL or DM, and decrements the double word I/O by 1 (–1).

* ( ) indicates the display when the LADDER EDITOR is used.

Cautionary notes

- When the area specified by s exceeds the maximum value of I/O number, DER is set to "1" and no operation is performed.
- When 00000000H is decreased by 1 (–1), the result will be FFFFFFFFH.

Program example




```
LD      X00200
AND DIF0
[
WR0100 = HFFFFFFFF
]

LD      X00200
AND DIF1
[
FUN 126 (WR0100)
]
```

Program description

- At a rising edge of X0200, the DR0100 is decremented by 1 (–1).

Internal output setting : DR0100 = HFFFFFFFF

Operation result : DR0100 = HFFFFFFFE

†: Supported by EH-CPU ***A/448/516/548 only.

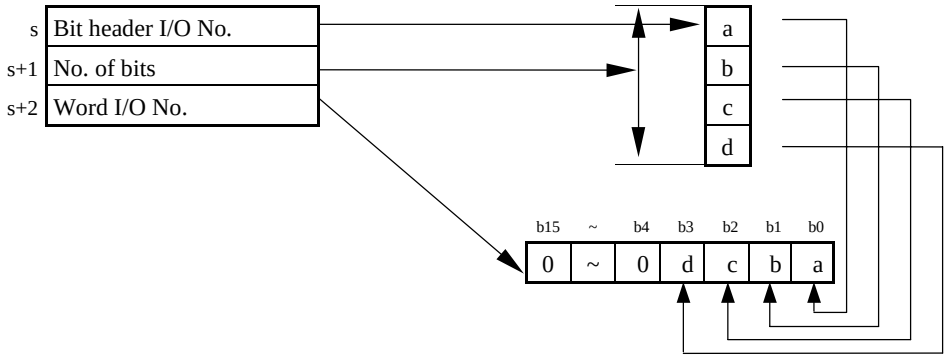
Item number	Fun commands-66	Name	Expand bit data to word data †

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 127 (s)	DER	ERR	SD	V	C							
* (BITTOW (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					139	←	133	←			
	Condition			Steps								
FUN 127 (s)	—			3								
* (BITTOW (s))												

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DN, DM		
s	Bit header I/O No.							○						Specify with actual address (Note)
s+1	No. of bits							○						0 to 16
s+2	Word I/O No.							○						Specify with actual address (Note)

Function

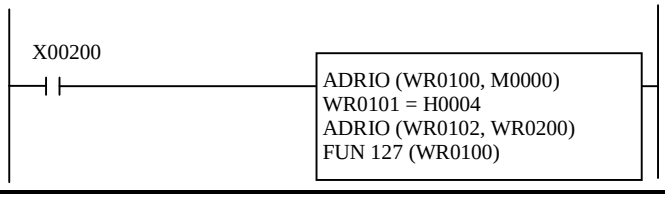
- The number of bits (s + 1) from the bit header I/O number (s) will be set in the specified word I/O number (s + 2) starting with the lowest bit.
- If the number of bits set in s + 1 is less than 16 (between 1 and 15), the upper level bit of the word I/O number specified in s + 2 will be set to “0”.



Cautionary notes

- Set the actual I/O addresses using the ADRIO command for the s and s + 2 parameters. If the actual addresses are not specified, DER will be equal to “1” and no processing will be performed.
- If s through s + 2 or the areas specified by them overlap, DER will be equal to “1” and no processing will be performed.
- If s through s + 2 exceed the maximum value of the I/O number, DER will be equal to “1” and no processing will be performed..
- If the areas specified by s through s + 2 exceed the maximum I/O number, the data will be expanded within the specified area range, but DER will be equal to “1”.
- If the number of bits (s + 1) exceeds 16, DER will be equal to “1” and no processing will be performed.
- If the number of bits (s + 1) is 0, no processing will be performed. DER will be equal to “1”.

Program example



†: Supported by EH-CPU ***A/448/516/548 only.

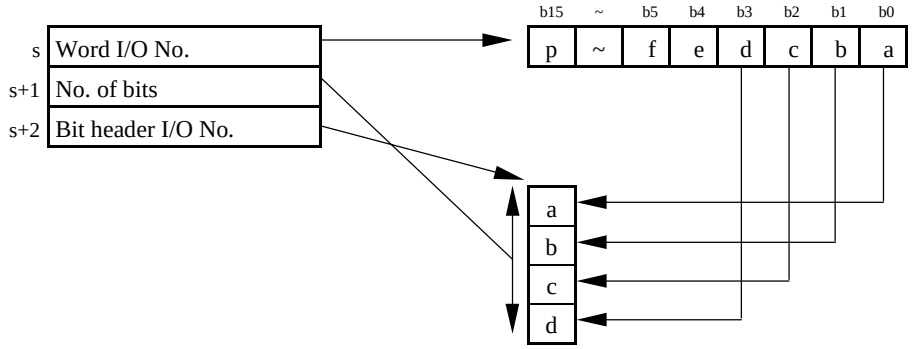
Item number	Fun commands-67	Name	Compress word data to bit data (DeviceNet) †

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 128 (s)	DER	ERR	SD	V	C							
* (WTOBIT (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					145	←	139	←			
	Condition			Steps								
FUN 128 (s)	—			3								
* (WTOBIT (s))												

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DN, DM		
s	Word I/O No.							○						Specify with actual address (Note)
s+1	No. of bits							○						0 to 16
s+2	Bit header I/O No.							○						Specify with actual address (Note)

Function

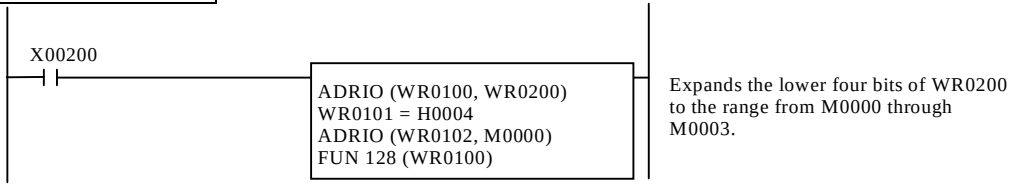
- Sets the specified number of bits (s + 1), from the lowest bit of the I/O (s) specified by the word I/O number, in the area starting with the specified header I/O number (s + 2).



Cautionary notes

- Set the actual I/O addresses using the ADRIO command for the s and s + 2 parameters. If the actual addresses are not specified, DER will be equal to “1” and no processing will be performed.
- If s through s + 2 or the areas specified by them overlap, DER will be equal to “1” and no processing will be performed.
- If s through s + 2 exceed the maximum value of the I/O number, DER will be equal to “1” and no processing will be performed..
- If the areas specified by s through s + 2 exceed the maximum I/O number, the data will be expanded within the specified area range, but DER will be equal to “1”.
- If the number of bits (s + 1) exceeds 16, DER will be equal to “1” and no processing will be performed.
- If the number of bits (s + 1) is 0, no processing will be performed. DER will be equal to “1”.

Program example



†: Supported by EH-CPU ***A/448/516/548 only.

Item number	Fun commands-68	Name	Explicit message execution †

Ladder format	Condition code					Processing time (μs)		Remark
FUN 162 (S)	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU5**		
	DER	ERR	SD	V	C	Ave	Max	
	⇕	●	●	●	●	127	388	

Command format	Number of steps	
FUN 162 (S)	Condition	Steps
	—	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, CU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument							○						

Function

・This is to execute explicit command for EH-RMD module.
・Put this command without any contact together with FUN 163 command.

Caution

・Argument "s" is dummy parameter. Assign to WR, WM or WL. Actual address is not influenced anything by this command.
・If EH-RMD module is not mounted, operation is not executed with DER=”1”.
・Do not use any contact with this command.
・Use this command in normal scan cycle.

†: Supported by EH-CPU 516/548 only.

Item number	Fun commands-69	Name	Explicit message configuration (DeviceNet) †

Ladder format	Condition code					Processing time (μs)		Remark
FUN 163 (s)	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU5**		
	DER	ERR	SD	V	C	Ave	Max	
	⇕	●	●	●	●	37	150	

Command format	Number of steps	
FUN 163 (s)	Condition	Steps
	—	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, CU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument							○						s uses up to s+5

Function

・Flag table and sending/receiving area address for explicit message are configured in this command.

・Since 4 times of EH-RMD can be used in case of remote assignment, this command has parameter area for 4 modules.

S parameter

+0	[0] Number of EH-RMD modules	1st EH-RMD
+1	[1] Slot No. for 1st EH-RMD	
+2	[2] Control flag address [ADRIO command]	
+3	[3] Sending area address [ADRIO command]	
+4	[4] Receiving area address [ADRIO command]	
+5	[5] Sending error code	
+6	[6] Receiving error code	
+7	[1] Slot No. for 2nd EH-RMD	
-	-	
+H13	[1] Slot No. for 4th EH-RMD	4th EH-RMD
+H14	[2] Control flag address [ADRIO command]	
+H15	[3] Sending area address [ADRIO command]	
+H16	[4] Receiving area address [ADRIO command]	
+H17	[5] Sending error code	
+H18	[6] Receiving error code	

[0] Number of EH-RMD modules. Maximum is 4.
[1] Slot number for 1st EH-RMD (0 to 7)
[2] Control flag address configured by ADRIO command.
(Possible I/O type : R, L and M.)
[3] Sending area address configured by ADRIO command.
(Possible I/O type : WR, WL and WM)
[4] Receiving area address configured by ADRIO command.
(Possible I/O type : WR, WL and WM)
[5] Sending error code set by CPU
[6] Receiving error code set by CPU

Caution

・Parameter tables are addressed by ADRIO command for [2] to [4].

・If I/O address of "s" does not exist in CPU, the command is not executed with DER=1.

・Be careful to map each table since area overlapping is not checked by system.

†: Supported by EH-CPU 516/548 only.

Control flag details

+0	[0] Send data flag
+1	[1] Initializing flag

[0] Send data flag :
Set 1 by user program to send explicit message. This flag is cleared after communication completed.

[1] Initializing flag :
Set 1 by user program to initialize the FUN command or to clear timeout. Received message is cleared as well.

Sending area details

+0	Size	
+1	Service	MACID
+2	Class	
+3	Instance	
+4	Service data 1	Service data 0
	Max. 64 byte for service data	
	Service data 63	Service data 62

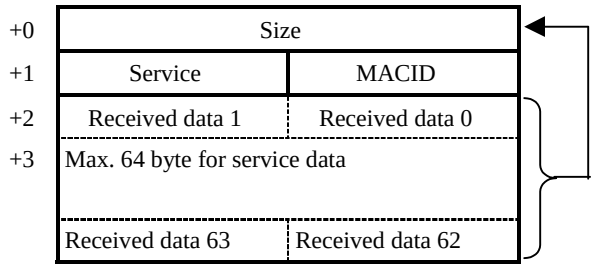
Size :
Service data (sending area from s+4 to s+n) size with byte unit
*) If the byte size is odd number, the last byte is stored in lower byte.

Sending error code

Error code

Error code	Name	Remarks
0001H	Timeout error	Detected by EH-RMD (timeout 3 sec.)
0002H	Data size error	Detected by EH-RMD
0003H	Mail box error	Detected by EH-RMD
0101H	Timeout error	Detected by CPU (timeout 5 sec.)
0202H	Range error	Data size exceeds configured sending area.
0203H	Slot number error	Slot number must be 0 to 7.
0204H	No module error	EH-RMD is not mounted on configured slot.

Receiving area details



Size :
Service data (receiving area from s+4 to s+n) size with byte unit
*) If the byte size is odd number, the last byte is stored in lower byte.

Receiving error

Error code

Error code	Description	Remarks
0000H	Command message sending ready	
0001H	Command message sending completed properly.	
0002H	Message being sent.	
0004H	Node off line error	Detected by EH-RMD
0005H	DeviceNet port off line	Detected by EH-RMD
0006H	Invalid Txid	Detected by EH-RMD
0007H	Txid duplicated	Detected by EH-RMD
0009H	sending/receiving buffer full	Detected by EH-RMD
000CH	Response data size over	Detected by EH-RMD
000EH	Size error	Detected by EH-RMD
000FH	Response time out error	Detected by EH-RMD (Timeout : 3 sec.)
0101H	Time out error	CPU←→EH-RMD timeout 5 sec.
0102H	Receiving time out error	Timeout 5 sec.
0201H	Txid unmatch	
0202H	Area range error	Receiving data is out of receiving range *1

*1) Data withing the range is kept.

Item number	Fun commands-70	Name	Inverter control command †

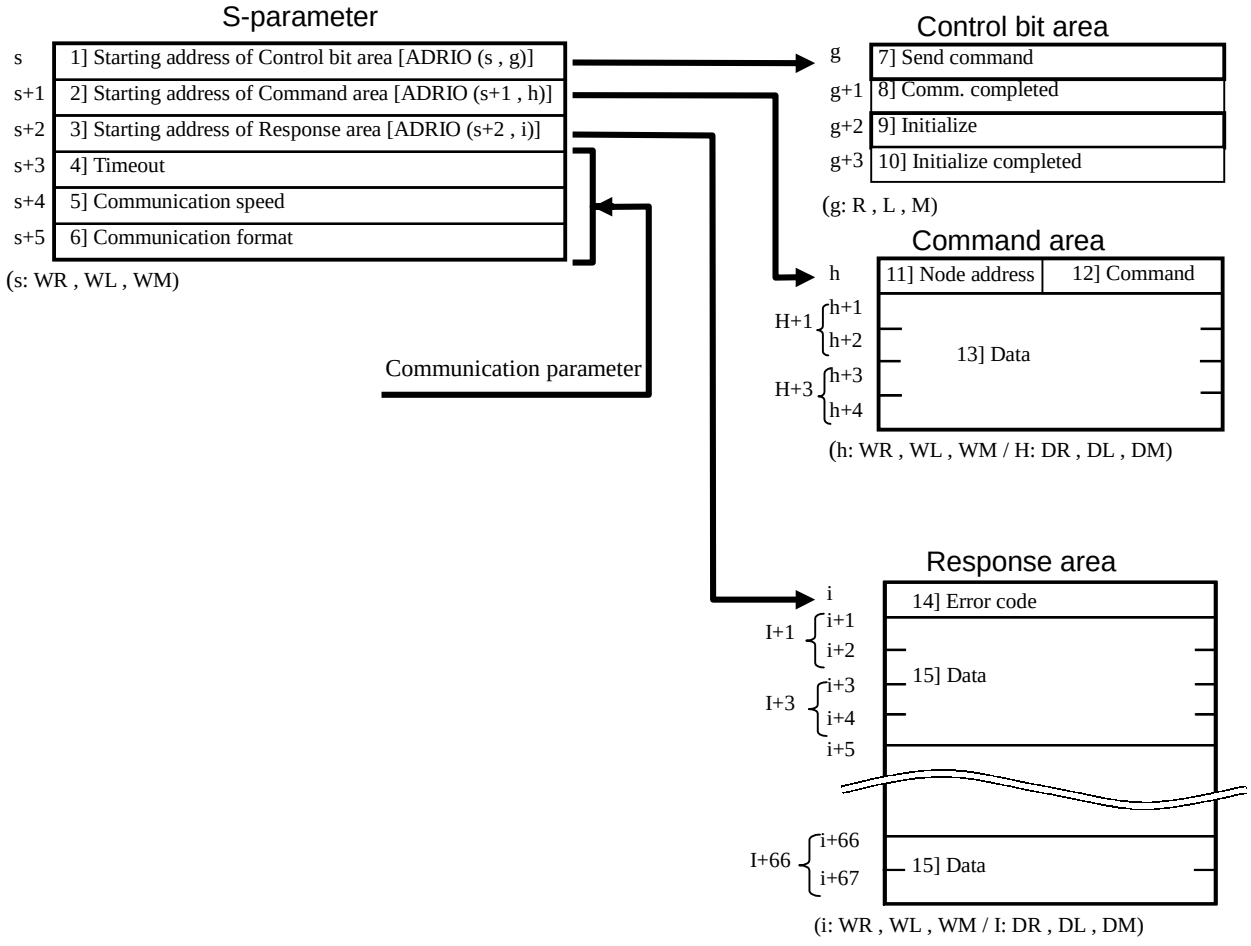
Ladder format	Condition code					Processing time (μs)		Remark
FUN 190 (S)	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU5**		
	DER	ERR	SD	V	C	Ave	Max	
	⇕	●	●	●	●	136.4	2736.4	

Command format	Number of steps	
FUN 190 (S)	Condition	Steps
	—	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, CU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument							○						s uses up to s+5
g	Control bit area			○										g uses up to g+3
h	Command area							○				○		h uses up to h+3
i	Response area							○				○		i uses up to i+3

Function

(1) This command is to control Hitachi inverter (SJ300/L300P/SJH300/HFC-VAN3 series) through CPU port 1.
(2) Configure communication port 1 as a RS-485 . (dip switch 5 Off, port type in WRF036)
(Please see Chapter 4 System Equipment 4.2 CPU Module, Chapter13 System Internal Outputs 13.2 Word Specical Internal Output Area)
(3) Configure s parameters to control inverter as below.
(4) s parameter table



†: Supported by CPU516/548.

Function

1] Starting address of control bit area : Assign control bit area by ADRIO command[*1].

2] Starting address of command area: Assign inverter command area by ADRIO command[*1].

3] Starting address of response: Assign response area by ADRIO command[*1].

*1 Refer to the Chapter 5 Command Specifications, 5.3 Command Specification Details (3) Application command for ADRIO command.

4] Timeout: The timeout from the execution of FUN190 command to the command completion.
= 0: Timeout is not checked
≠ 0: ×10ms timeout check (The maximum possible set value is HFFFF)

5] Communication speed: Configure as same speed as inverter's.

· Baud rate	· Set value
· 2400 bps	· H0000
· 4800 bps	· H0001
· 9600 bps	· H0002
· 19200 bps	· H0003

6] Transmission format: Configure as same format as inverter's.

Transmission code	Set value
7 bit even parity 2 stops	H0000
7 bit odd parity 2 stops	H0001
7 bit even parity 1 stop	H0002
7 bit odd parity 1 stop	H0003
8 bit no parity 2 stops	H0004
8 bit no parity 1 stop	H0005
8 bit even parity 1 stop	H0006
8 bit odd parity 1 stop	H0007

(5) Control bit area
Set 1 in user program to send command or initialize the command.
7] Send command: Set 1 to send command to inverter in user program.
CPU sets 0 automatically when the operation completed.

8] Communication completed : CPU sets 1 automatically when the operation completed
When 7] Send command bit or 9] Initialize bit is set, it is cleared by CPU.

9] Initialize: Set 1 to initialize the FUN 190 command in user program.
CPU sets 0 automatically when the operation completed.

10] Initialize completed: CPU sets 0 automatically when the operation initialized.
When 7] Send command bit or 9] Initialize bit is set, it is cleared by CPU.

(6) Command area
Configure Node address number and command according to below description.

11] Node address: Set Node address number of inverter from 01 up to 32(all BCD data). Node address number FF (at BCD data)[*2] is for broadcast sending, which is supported only by commands 00, 01, 02, 07, 08, 0A, and 0B[*3]. Inverters do not reply to this broadcast command.

*2 If configured Node address number does not exist, no error will be detected. Use timeout instead.
*3 If Node address number FF is used with command, which does not support FF, error message given in " (7)Response area 14] Error code".

Function

12] Command list of inverter

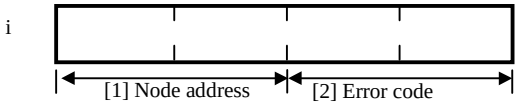
Command	Description	Broadcast
00	Forward / Reverse / Stop command	✓
01	Setting of frequency in standard profile	✓
02	Setting of intelligent terminal state	✓
03	Read all monitor date (back read)	
04	Read inverter status	
05	Read trip history	
06	Read a single parameter value	
07	Write a single parameter value	✓
08	Set inverter parameters to default values	✓
09	Verifies that the requested setting can be written to EEPROM	
0A	Writes a parameter value to EEPROM	✓
0B	Requests the recalculation of internal constant	✓

13] Data : The setting value set as an inverter by each command is input.

(7) Response area

The response from inverter is stored in this area.

14] Error code : If FUN 190 command is not completed correctly, error code is given in the following format.



[1] Node address : Inverter Node address number of communication error is given in BCD format.
(Node address No.24 ➔ H24) When PLC detects errors, Node address is displayed on "HEE".
[2] Error code: Detected by either inverter error or PLC error .

Error code list

Inverter error		PLC error	
Error code	Description	Error code	Description
	Parity error		Parity error
H02	Check sum error	H22	Framing error
H03	Framing error	H23	Overrun error
H04	Overrun error	H24	Timeout
H05	Protocol error	H25	Receiving buffer overrun error
H06	ASCII code error	H26	Sum check error
H07	Receive buffer overrun error	H27	Protocol error*4
H08	Receive time-out error	H28	ASCII code error
H11	Abnormal command code/value error	H29	Node address error
H13	Execution impossible error	H30	Command unusual error
H16	Abnormal parameter code/value error	H31	Conflict error*5
		H32	Port designation error
		H33	Command area range error
		H34	Response area range error
		H35	Overlap error
		H36	Node address unmatched

*4 No [CR] is detected in right position.
*5 This error is detected when FUN190 is executed while TRNS0 or RECV0 command is operated. In this case, FUN190 command will be terminated.

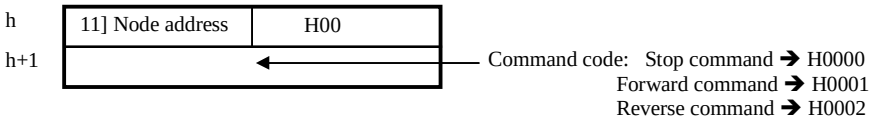
15] Data : The response data is stored in the following format depending on 03,04,05,06,09 command.

Function

(8) Command area/ Response area by each command.

(i) Command 00 : Forward / Reverse / Stop command

(a) Command area

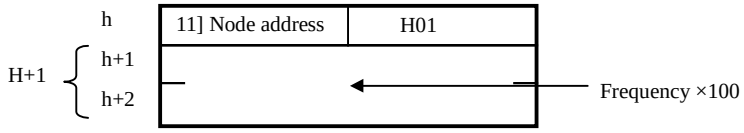


(b) Response area

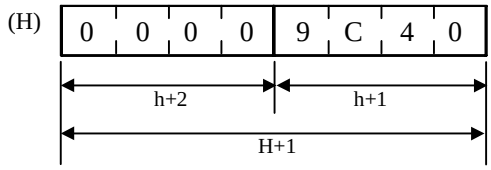
No response normally. If operation does not work properly, error code is given as shown in "14] Error code".

(ii) Command 01: Setting of frequency in standard profile

(a) Command area



Example) 400Hz ➔ H+1 : 40000 = 400 × 100 [40000 (DEC) = 9C40 (HEX)

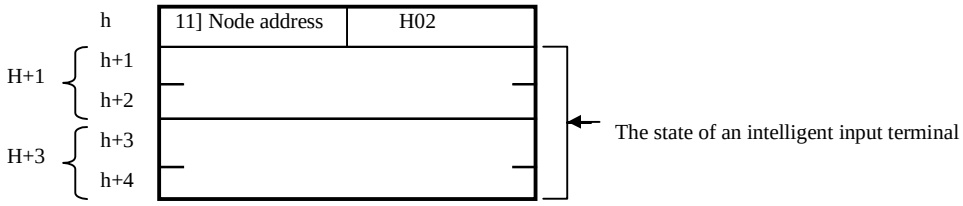


(b) Response area

No response normally. If operation does not work properly, error code is given as shown in "14] Error code".

(iii) Command 02: Setting of intelligent terminal state.

(a) Command area

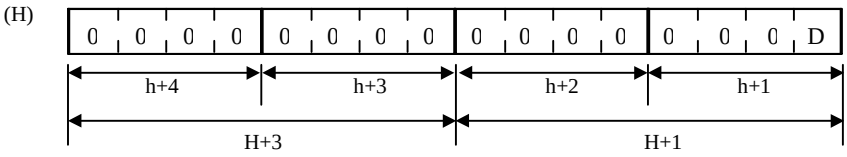


Example) [FW], [CF1] and [CF2] to be active.

[FW] [CF1] [CF2] (Input Data)

H0000000000000001 + H0000000000000004 + H0000000000000008 = H000000000000000D

Input Data



(b) Response area

No response normally. If operation does not work properly, error code is given as shown in "14] Error code".

FUN 190 (s)

Function

(iv) Command 03: Read all monitor data

(a) Command area

h	11] Node address	H03

(b) Response area

No response normally. If operation does not work properly, error code is given as shown in "14] Error code".

	i	14] Error code
I+1	i+1	[1] Output frequency × 100*7
	i+2	
	i+3	[2] Output current × 10
	i+4	[3] Direction of rotation
I+5	i+5	[4] PID feedback*8
	i+6	
	i+7	[5] Intelligent input
	i+8	[6] Intelligent output
I+9	i+9	[7] Frequency converting
	i+10	× 100
	i+11	[8] Output torque× 10*11
	i+12	[9] Output voltage× 10
	i+13	[10] Electric power × 10*12
	i+14	[11] Reserved
I+15	i+15	[12] Run Mode time
	i+16	
I+17	i+17	[13] Power ON time
	i+18	

[1] Output frequency*7 : (value × 100)

*7 As for SJH300 / HFC-VAH3 Series, the value of output frequency × 10 is displayed.

[2] Output current : The value of output current × 10 of an Inverter is displayed.

[3] The rotation direction:

0 : Stop, 1 : FWD, 2 REV

[4] PID feedback monitor*8: (value × 100)

*8 Not supported by SJH300 / HFC-VAH3 Series. ("H0" is displayed)

[5] Intelligent input monitor:

Terminal	Monitor Item	Set value
[FW]	Forward input	H0100
1	Input 1	H0001
2	Input 2	H0002
3	Input 3	H0004
4	Input 4	H0008
5	Input 5	H0010
6*9	Input 6	H0020
7*9	Input 7	H0040
8*9	Input 8	H0080

*9 Not supported by L300P Series.

[6] Intelligent output monitor

Terminal	Monitor Item	Set value
AL	Alarm relay	H0020
11	Output 1	H0001
12	Output 2	H0002
13*10	Output 3	H0004
14*10	Output 4	H0008
15*10	Output 5	H0010

*10 Not supported by L300P Series..

Function

[7] Frequency conversion monitor: (value × 100)
[8] Output torque*11:
*11 Not supported by SJH300 / HFC-VAH3 Series. ("H0" is displayed)
[9] Output voltage monitor: (value × 10)
[10] Electric power monitor*12: (value × 10)
*12 Not supported by HFC-VAH3 Series. ("H0" is displayed)
[11] Reserve: (always 0)
[12] RUN time monitor:
[13] Power ON time monitor:

(V) Command 04: Read inverter status
(a) Command area

h	11] Node address	H04

(b) Response area
If operation does not work properly, error code is given as shown in “14] Error code”.

		14] Error code		
i		Status C		Status A
I+1	i+1			Status B
	i+2	Status A	Status B	Status C

Status A code

code	Status
H00	Initial state
H01	-
H02	On Stopping
H03	On Running
H04	On Free-run stop
H05	On Jog
H06	On Dynamic braking
H07	On Retry
H08	On TRIP
H09	On Under-voltage

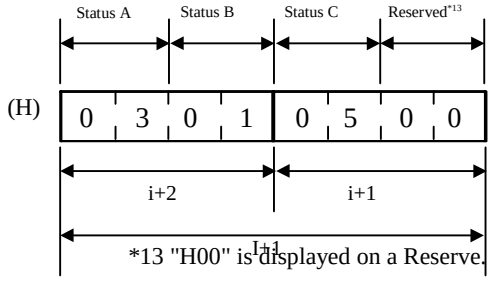
Status B code

code	Status
H01	On Running
H02	On Tripping

Status C code

code	Status
	-
H01	Stop
H02	Deceleration speed
H03	Constant speed
H04	Acceleration speed
H05	Forward
H06	Reverse
H07	Reverse from forward
H08	Forward from reverse
H09	Forward start
H10	Reverse start

Example) Status A: On Running + Status B: On Running + Status C: Forward
(03) (01) (05)



*13 "H00" is displayed on a Reserve.

FUN 190 (s)

Function

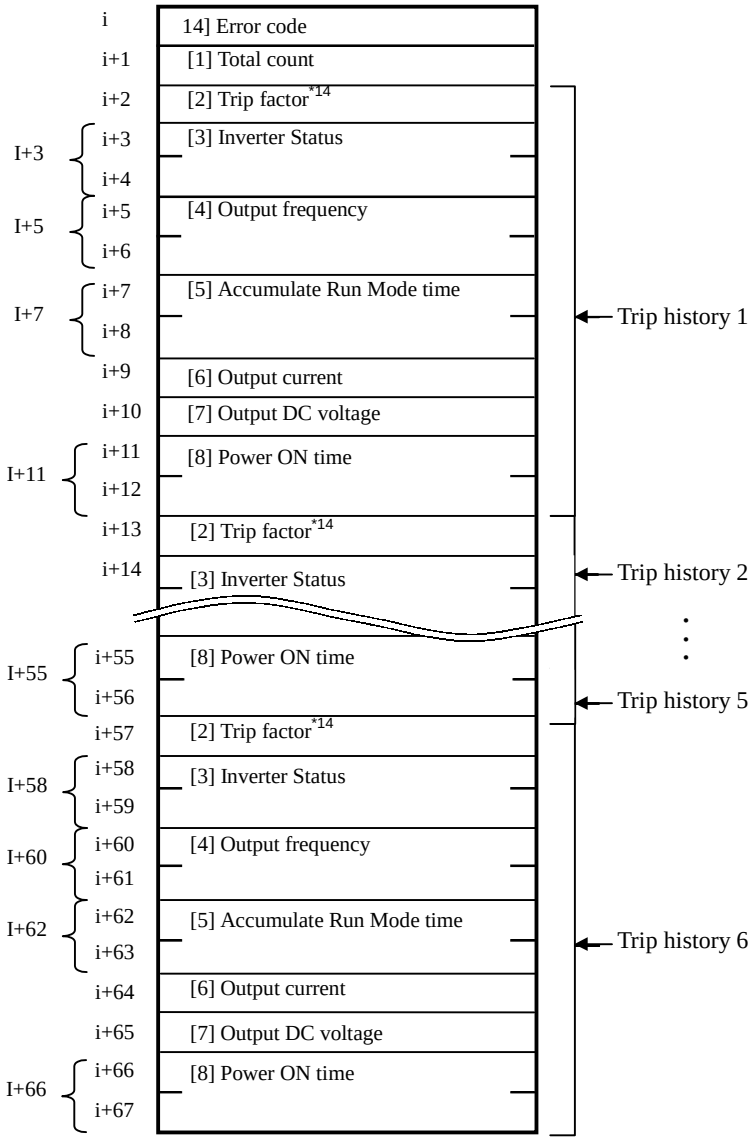
(Vi) Command 05: Read trip history

(a) Command area

h	11] Node address	H05

(b) Response area

No response normally. If operation does not work properly, error code is given as shown in “14] Error code”.



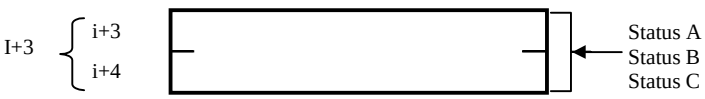
The trip history is shown with total count [1], and the last 6 times data are kept.
"0" is stored if there is no history.

[1] Total count : The total accumulated number of trips

[2] Trip factor*14:

*14 Please refer to inverter manual for further information.

[3] Inverter Status:



Status A: Status at inverter trip.
Status B: Trip factor detected by gate array.
Status C: Detailed status at inverter trip.

FUN 190 (s)

Function

Status A code

Code	Condition
H00	Initial status
H01	-
H02	On Stopping
H03	On Running
H04	On Free-Run Stop
H05	On Jog
H06	On Dynamic
H07	On retry
H08	On Trip
H09	On Under-Voltage

Status B code

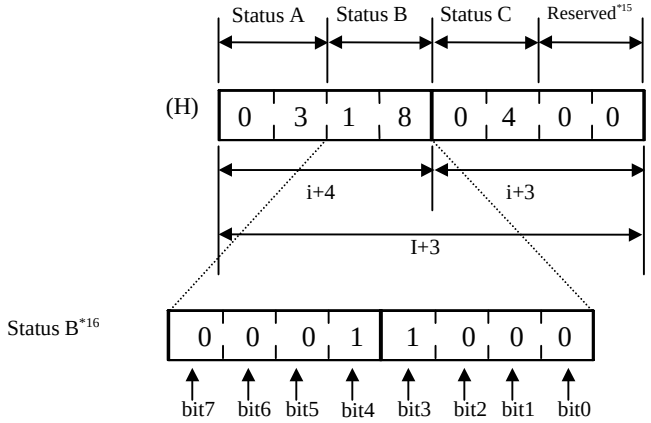
Bit	Condition
bit0	Option earth fault
bit1	IGBT error U phase
bit2	Undervoltage
bit3	Overvoltage protection
bit4	Abnormal temperature
bit5	IGBT error W phase
bit6	IGBT error Vphase
bit7	Gate array error

Status C code

Code	Condition
	On Resettg
H01	On Stopping
H02	On Decelerating
H03	On Constant speed
H04	On Accelerating
H05	On Running 0Hz
H06	On Starting
H07	On Dynamic Braking
H08	On overload protection

Example) Inverter trip by abnormal temperature and Overvoltage protection during acceleration of Running (Trip history 1)

Status A : On Running (03)
Status B : Abnormal temperature (bit4) ,and Overvoltage protection (bit3)
Status C : On Accelerating (04)

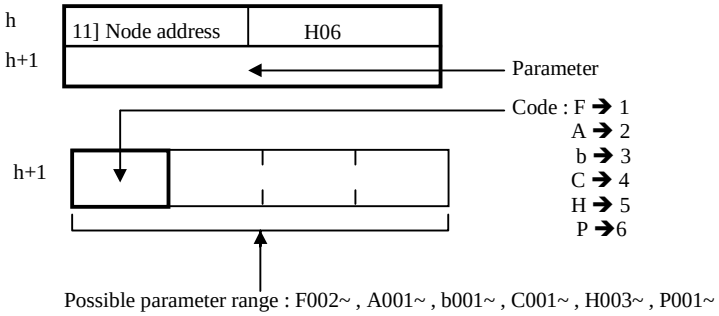


*15 "H00" is displayed on a Reserve.
*16 Trip factor (status B) is given as 8 bits format data. "1" is set in corresponding bit.

[4] Output frequency: (value × 10)
[5] Accumulate Run time :
[6] Output current: (value × 10)
[7] Output DC voltage: (value × 10)
[8] Power ON time:

(Vii) Command 06: Read a single parameter value

(a) Command area



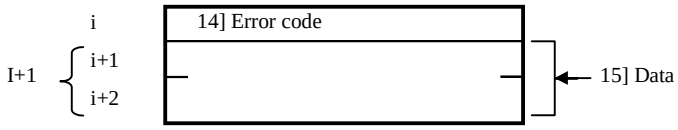
Example) "PID proportional gain": A072

Input parameter h+1 : H2072

Function

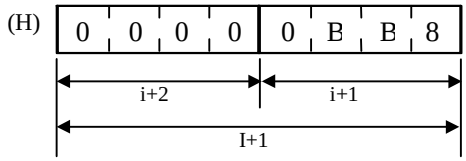
(b) Response area

When FUN 190 command does an unusual end, Error code is displayed on “14] Error code”.



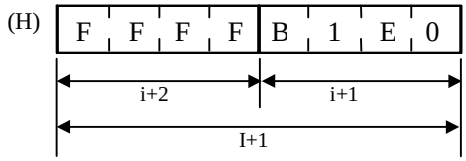
15] Data:

Example 1) Parameter of the 1st acceleration time (F002)
and a response is 30.00s
I+1 : 3000 = 30 × 100 [3000 (DEC) = BB8 (HEX)] *17



*17 10 or 100 times value is given depending on each command. Refer to inverter manual for further information..
Please refer to inverter manual for further information.

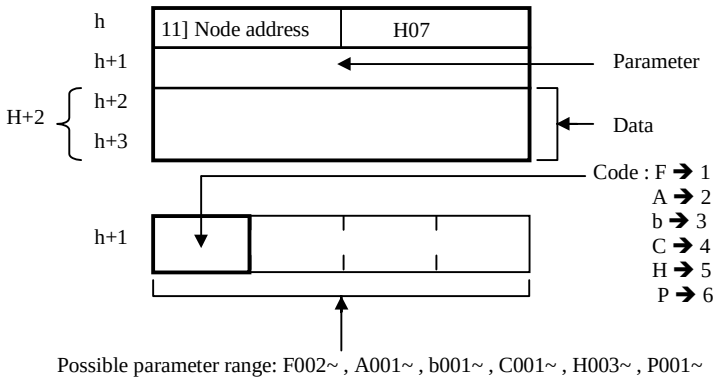
Example 2) When the parameter of the 02 start (A111) is read by command
and a response is -200%
I+1 : -200 × 100 = -20000 [-20000 (DEC) = FFFFB1E0 (HEX)] *17,*18



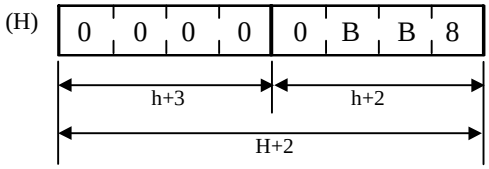
*18 The data of the parameter of A111~ A114 is given as two’s complement.

(Viii) Command 07: Write a single parameter value

(a) Command area



Example 1.) "Acceleration (F002)" ➔ h+1 : H1002
30.00 sec. ➔ H+2 : 3000 (30×100) *19 [3000 (DEC) = BB8 (HEX)]

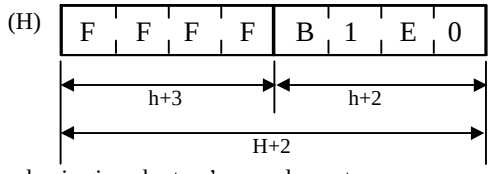


*19 Set 10 or 100 times value depending on each command. Refer to inverter manual for further information

FUN 190 (s)

Function

Example 2.) 02-L input active range start frequency (A111) ➔ h+1 : H2111
-200 % ➔ -20000 (-200×100) (Two's complement) *20 ➔ H+2 : HFFFFB1E0
[-20000 (DEC) = FFFFB1E0 (HEX)] *20



*20 Negative value is given by two's complement

(b) Response area
No response normally. If operation does not work properly, error code is given as shown in "14] Error code".

(ix) Command 08: Set inverter parameters to default values
(a) Command area

h	11] Node address	H08

(b) Response area
No response normally. If operation does not work properly, error code is given as shown in "14] Error code".

(X) Command 09: Verifies that the requested setting can be written to EEPROM
(a) Command area

h	11] Node address	H09

(b) Response area
No response normally. If operation does not work properly, error code is given as shown in "14] Error code".

i	14] Error code
i+1	15] Data

15] Data: The result checked by Command 09 is displayed.

Check result code

Code	Condition
H00	Save is impossible
H01	Save is possible

(Xi) Command 0A: Writes a parameter value to EEPROM
(a) Command area

h	11] Node address	H0A

(b) Response area
No response normally. If operation does not work properly, error code is given as shown in "14] Error code".

(Xii) Command 0B: Requests the recalculation of internal constant
(a) Command area

h	11] Node address	H0B

(b) Response area
No response normally. If operation does not work properly, error code is given as shown in "14] Error code".

Cautionary notes

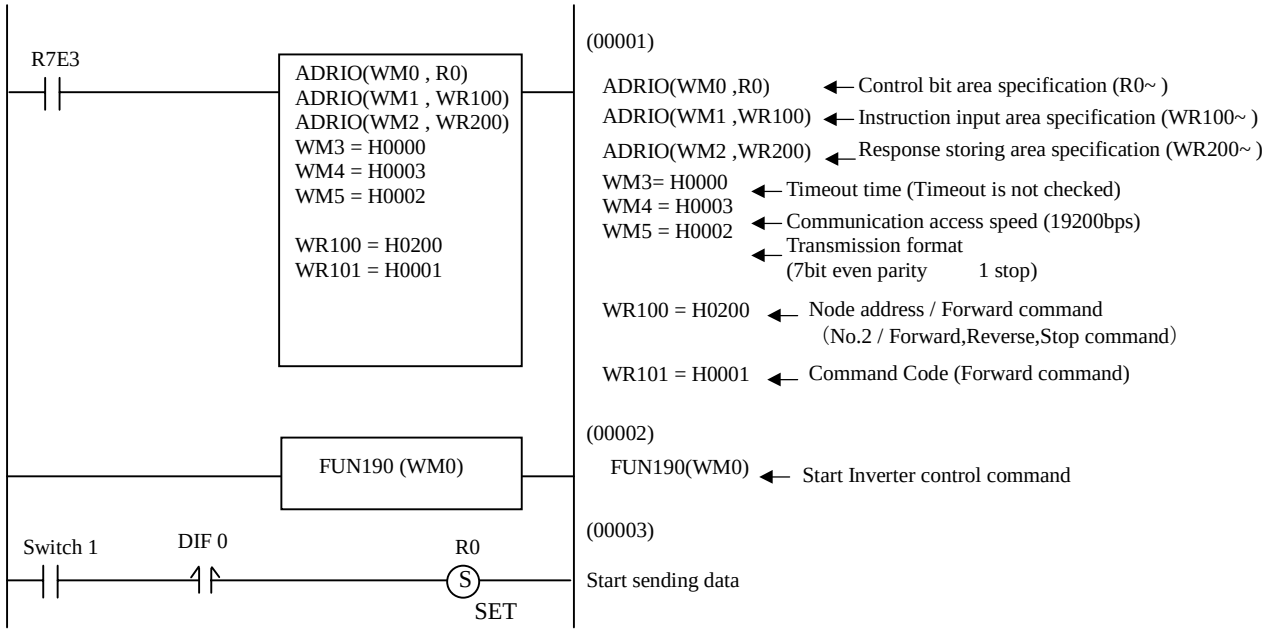
- FUN190 cannot carry out multiplex execution. When multiplex execution is carried out, FUN190 is not performed normally.
- Supported inverters are Hitachi SJ300/L300P/SJH300/HFC-VAN3 series.
- Refer to the inverter manuals for further information.。
- Be sure to configure the RS-485 communication port of inverter side.
- Configure each parameter within the I/O range, otherwise this command does not work. In this case, error code 52 is stored in WRF000, and DER bit is set.

Occupied words for each command

Command	Command data	Response data
00	2 words	1 words
01	3 words	1 words
02	5 words	1 words
03	1 word	19 words
04	1 word	3 words
05	1 word	68 words
06	2 words	3 words
07	4 words	1 words
08	1 word	1 words
09	1 word	2 words
0A	1 word	1 words
0B	1 word	1 words

- When the parameter set as s-parameter is unusual, error code 52 is stored in WRF000, and DER bit is set.
- When the PLC communication port is not set as RS-485, error code 52 is stored in WRF000, and DER bit is set.

Program example



Program description

- When the switch 1 is ON, inverter No.2 starts operation in forward direction.

Item number	Fun commands-64	Name	Extended X/Y area read/write command †

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU3**A EH-CPU3**		Other than left		
FUN 200 (s)	DER	ERR	SD	V	C							
* (XYR/W (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

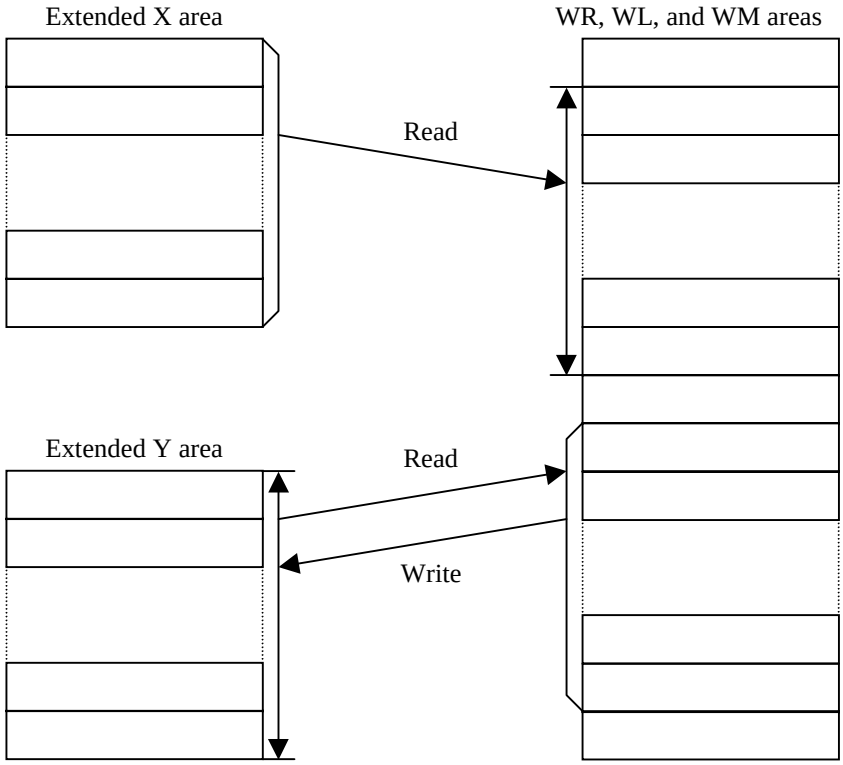
Command format	Number of steps		Processing time (μs)					
	Condition	Steps	EH-CPU4** EH-CPU5**		EH-CPU3**A EH-CPU3**		Other than left	
FUN 200 (s)	—	3	65	337	57	217		
* (XYR/W (s))								

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Header of parameter area							○						s uses up to s+7

Function

- This command reads and writes data between the CPU and a module using the extended X and Y areas.
- The maximum read and write data sizes are determined by the number of module's I/O assignment points. For details on using this command, refer to the applicable module manual.
- By specifying the control type, the read/write method by performing or not performing handshaking with the module can be selected. However, in order to use the read/write method by performing handshaking, the module side must support handshaking. For more details, refer to the applicable module manual.
- s sets the head I/O number of the parameter area that sets various data read/write parameters.

* The values displayed inside parenthesis ( ) show the display when LADDER EDITOR is used.



† : Supported by EH-CPU 308(A)/316(A)/448(A)516/548 only.

Description of S parameter area

s+0	1] Error code
s+1	2] System area
s+2	(not available to user)
s+3	3] Control type
s+4	4] Header of target area
s+5	5] Read/write control bit I/O No.
s+6	6] Transfer source (destination) header I/O No.
s+7	7] Size

[1] Error code:
The result of FUN 200 command execution is set.
Normal end $\rightarrow = 0$
Abnormal end $\rightarrow \neq 0$

[2] System area:
This is used by the system processes of the FUN 200 command when the FUN 200 command is executed. This area cannot be used by the user.

[3] Control type
Specifies the control type.
H0001: Read request from the X area (with handshaking)
H0002: Read request from the X area (without handshaking)
H0003: Write request to the Y area (with handshaking)
H0004: Forced write request to the Y area (without handshaking)
H0005: Forced read request from the Y area (without handshaking)

[4] Header of target area:

b15	b11	b7	b0
Unit No.	Slot No.	Word location	

Sets the unit number (0 to 7)
Sets the slot number (0 to 7)
Sets the word location (from 0) *1

[5] Read/write control bit I/O number:
Sets the actual addresses of R, L, and M in the read/write control bit I/O number using the ADRIO command.

[6] Transfer source (destination) header I/O number:
Sets the actual addresses of WR, WL, and WM in the transfer source (destination) header I/O number using the ADRIO command.

[7] Size:
Sets the size to be read or written (from 1) in word units. *1

Description of read/write control bit table

+0	1] Execute flag
+1	2] Normal end flag
+2	3] Abnormal end flag

1] Execution flag:
Set to “1” using a user program when performing read or write operation using the FUN 200 command. When the read or write operation is complete, the FUN 200 command resets this to “0”.

2] Normal end flag:
Set to “1” when read or write operation by the FUN 200 command is normally completed. When read or write operation is started, the FUN 200 command resets this to “0”.

3] Abnormal end flag:
Set to “1” when read or write operation by the FUN 200 command is abnormally completed. When read or write operation is started, the FUN 200 command resets this to “0”.

* Description of borders

User setting area
User write prohibited area

*1 The ranges in which the word location and size can be specified will differ depending on assignment. The equation for calculating the ranges of the word location and size is as follows.
Size = 3Fh – Number of input points – Number of output points

(Example) If assignment is 4W/4W,
3Fh – 4 – 4 = 37h
The word location range is 00h through 36h.
The size range is 0001h through 0037h.

Cautionary notes

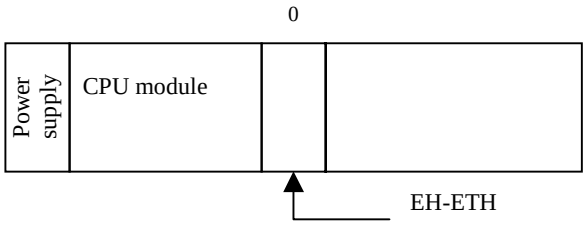
- When an error is generated, an error code is set, DER will be equal to "1," and no processing will be performed.
- If the specified size is 0, the process will end normally with performing anything.
- Execute using only normal scan without entering any startup conditions.

For more information on the error codes that can be set, see the Error Code Details.

Program example

In this program example, a program performs information settings on the Ethernet module (EH-ETH) by writing to the Y area. After the information is set normally, the program reads the information settings of the Ethernet module (EH-ETH) by reading the X area.

(1) EH-ETH installation



The EH-ETH is installed in the "0" slot of the basic base. Therefore, the I/O assignment of the EH-ETH will be WX0000, WY0001.

(2) Internal output assignments

A sample program will be created using the assignments shown below. Change the I/O number, etc. according to the application in an actual usage.

I/O	NO.	Usage
WR	0000 to 0007	FUN 200 command (Y area writing) Parameter area (S to S + 7)
M	0000 to 0002	FUN 200 command (Y area writing) Control bit table
WM	100 to 117	Transfer source data area (24 words)
WR	0010 to 0017	FUN 200 command (X area reading) Parameter area (S to S + 7)
M	0010 to 0012	FUN 200 command (X area reading) Control bit table
WM	120 to 137	Transfer source data area (24 words)
R	100 101	Execution request (FUN 200 execution Y area writing) Execution request (FUN 200 execution X area reading)

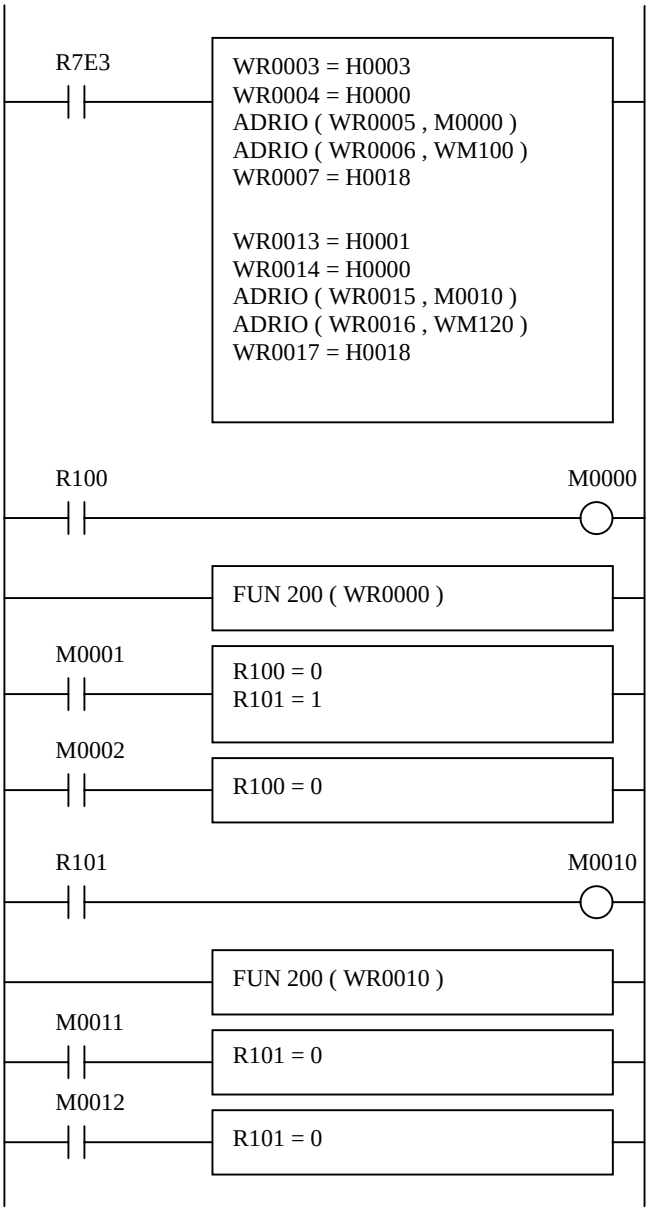
Program example

(3) Data passed from the CPU to the EH-ETH

Using the following program example, the data to be set in the transfer source data area that will be passed from the CPU to the EH-ETH is described.
For more details, see the EH-ETH manual.

I/O NO.	Data	Description	Remark
WM100	HC000	Higher IP address	192.000
WM101	H0000	Lower IP address	000.000
WM102	HFFFF	Higher subnet mask address	255.255
WM103	HFF00	Lower subnet mask address	255.000
WM104	HC000	Higher IP address for sending and receiving tests	192.000
WM105	H00FF	Lower IP address for sending and receiving tests	000.255
WM106	H0FA0	Port number for sending and receiving tests	4000
WM107	H0000	Higher gateway address 1	000.000
WM108	H0000	Lower gateway address 1	000.000
WM109	H0000	Higher gateway address 2	000.000
WM10A	H0000	Lower gateway address 2	000.000
WM10B	H0000	Higher gateway address 3	000.000
WM10C	H0000	Lower gateway address 3	000.000
WM10D	H0000	Higher gateway address 4	000.000
WM10E	H0000	Lower gateway address 4	000.000
WM10F	H0BBC	Task code 1 port number	3004
WM110	H0001	Task code 1 communication protocol	TCP/IP
WM111	H0BBD	Task code 2 port number	3005
WM112	H0001	Task code 2 communication protocol	TCP/IP
WM113	H0BBE	Task code 3 port number	3006
WM114	H0001	Task code 3 communication protocol	TCP/IP
WM115	H0BBF	Task code 4 port number	3007
WM116	H0001	Task code 4 communication protocol	TCP/IP
WM117	H0000	Task code timeout	No timeout check

Program example



Perform parameter area initialization settings with 1 scan execution on.
Write request to the Y area
Set M0000 in the read control bit I/O number from unit number 0, slot number 0, word location word 0.
Set WM100 in the transfer source header I/O number.
Set the read size to 24 words.

Read request from the X area
Set M0010 in the read control bit I/O number from unit number 0, slot number 0, word location word 0.
Set WM120 in the transfer source header I/O number.
Set the read size to 24 words.

Start reading when the execution flag M0000 turns on

Contact point R100 turns off when the normal end flag M0001 turns on.

Contact point R101 turns on when the normal end flag M0001 turns on.

Contact point R100 turns off when the abnormal end flag M0002 turns on.

After writing to the Y area is completed normally, start reading from the X area.

Start reading when the execution flag M0010 turns on.

Contact point R101 turns on when the normal end flag M0011 turns on.

Contact point R101 turns off when the abnormal end flag M0012 turns on.

Item number	Fun commands-65	Name	Status control read/write command †

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU3**A EH-CPU3**		Other than left		
FUN 201 (s)	DER	ERR	SD	V	C							
* (SCR/W (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

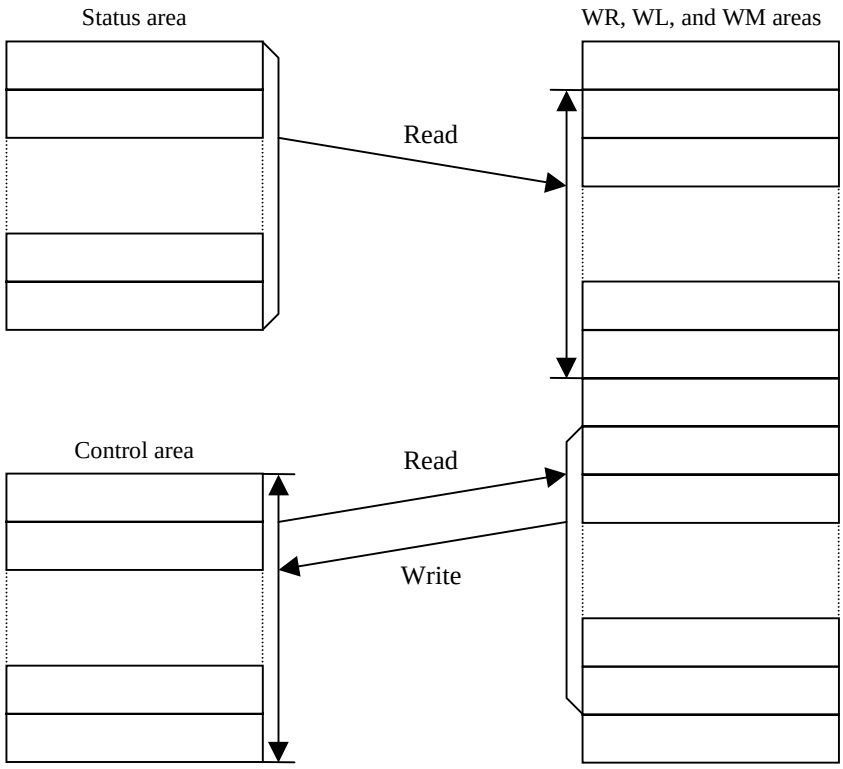
Command format	Number of steps										
	Condition	Steps	66	338	58	218					
FUN 201 (s)	—	3									
* (SCR/W (s))											

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Header of parameter area							○						s uses up to s+7

Function

- This command reads and writes data between the CPU and a module using the status control area.
- The maximum read and write data sizes are different depending on the module. For details on using this command, refer to the applicable module manual.
- By specifying the control type, the read/write method by performing or not performing handshaking with the module can be selected. However, in order to use the read/write method by performing handshaking, the module side must support handshaking. For more details, refer to the applicable module manual.
- s sets the head I/O number of the parameter area that sets various data read/write parameters.

* The values displayed inside parenthesis () show the display when LADDER EDITOR is used.



† : Supported by EH-CPU 308(A)/316(A)/448(A)516/548 only.

Description of S parameter area

s+0	1] Error code
s+1	2] System area
s+2	(not available to user)
s+3	3] Control type
s+4	4] Header of target area
s+5	5] Read/write control bit I/O No.
s+6	6] Transfer source (destination) header I/O No.
s+7	7] Size

[1] Error code:
The result of FUN 201 command execution is set.
Normal end $\rightarrow = 0$
Abnormal end $\rightarrow \neq 0$

[2] System area:
This is used by the system processes of the FUN 201 command when the FUN 201 command is executed. This area cannot be used by the user.

[3] Control type
Specifies the control type.
H0001: Read request from the status area (with handshaking)
H0002: Read request from the status area (without handshaking)
H0003: Write request to the control area (with handshaking)
H0004: Forced write request to the control area (without handshaking)
H0005: Forced read request from the control area (without handshaking)
HA55A: Software reset

[4] Header of target area:

b15	b11	b7	b0
Unit No.	Slot No.	Word location	

Sets the unit number (0 to 7)
Sets the slot number (0 to 7)
Sets the word location (from 0) *1

[5] Read/write control bit I/O number:
Sets the actual addresses of R, L, and M in the read/write control bit I/O number using the ADRIO command.

[6] Transfer source (destination) header I/O number:
Sets the actual addresses of WR, WL, and WM in the transfer source (destination) header I/O number using the ADRIO command.

[7] Size:
Sets the size to be read or written (from 1) in word units. *1

Description of read/write control bit table

+0	1] Execute flag
+1	2] Normal end flag
+2	3] Abnormal end flag

1] Execution flag:
Set to “1” using a user program when performing read or write operation using the FUN 201 command. When the read or write operation is complete, the FUN 201 command resets this to “0”.

2] Normal end flag:
Set to “1” when read or write operation by the FUN 201 command is normally completed. When read or write operation is started, the FUN 201 command resets this to “0”.

3] Abnormal end flag:
Set to “1” when read or write operation by the FUN 201 command is abnormally completed. When read or write operation is started, the FUN 201 command resets this to “0”.

* Description of borders

User setting area
User write prohibited area

*1

- When performing read or write operation by controlling handshaking (when the control type is H001 or H003)
  The ranges that can be specified for the word location and size are different for each module. Refer to the applicable module manual.

- When performing forced read or forced write operation
  The word location range will be 00h to 3Dh.
  The size range will be 0001h to 003Eh.

Cautionary notes

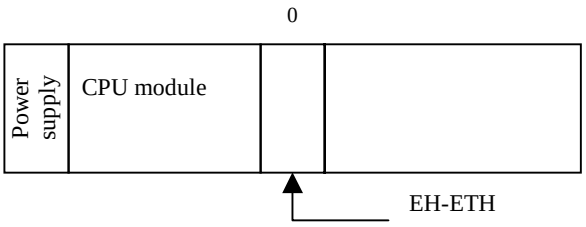
- When an error is generated, an error code is set, DER will be equal to "1," and no processing will be performed.
- If the specified size is 0, the process will end normally with performing anything.
- Execute using only normal scan without entering any startup conditions.

For more information on the error codes that can be set, see the Error Code Details.

Program example

The following is an example of a program that sets the transmission information of the Ethernet module (EH-ETH) by writing to the control area. Another example of a program that reads the communication status and error information of the Ethernet module (EH-ETH) by reading the status area is also shown.

(1) EH-ETH installation



The EH-ETH is installed in the "0" slot of the basic base. Therefore, the I/O assignment of the EH-ETH will be WX0000, WY0001.

(2) Internal output assignments

A sample program will be created using the assignments shown below. Change the I/O number, etc. according to the application in an actual usage.

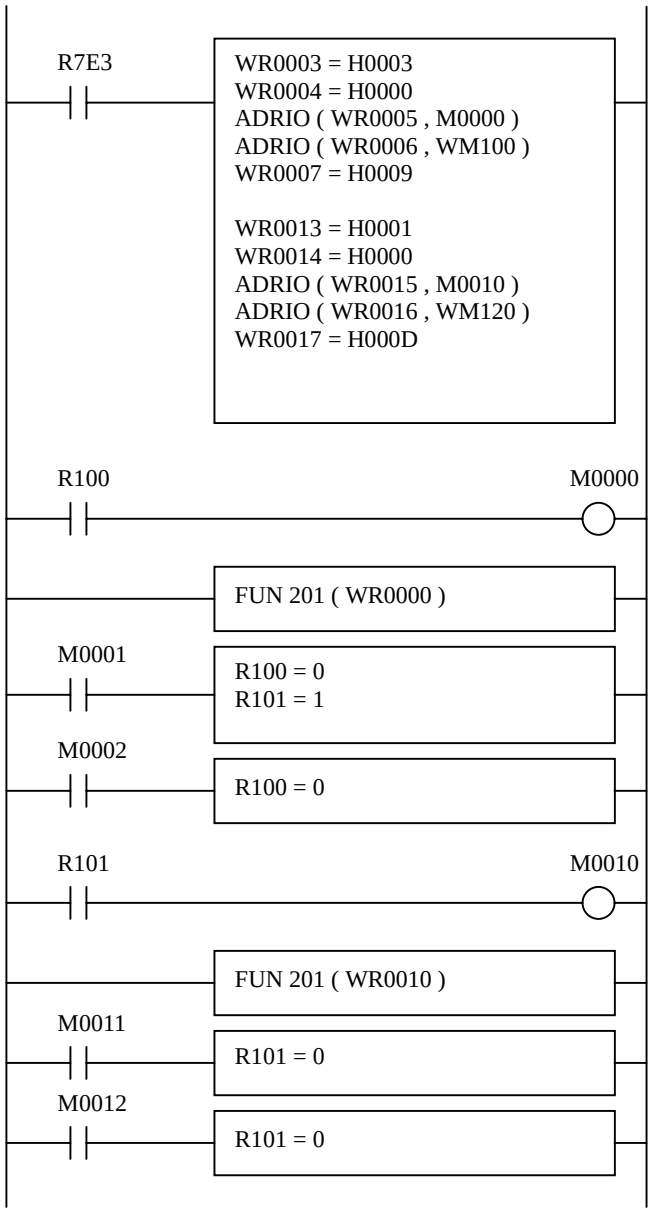
I/O	NO.	Usage
WR	0000 to 0007	FUN 201 command (control area writing) Parameter area (S to S + 7)
M	0000 to 0002	FUN 201 command (control area writing) Control bit table
WM	100 to 108	Transfer source data area (9 words)
WR	0010 to 0017	FUN 201 command (status area reading) Parameter area (S to S + 7)
M	0010 to 0012	FUN 201 command (status area reading) Control bit table
WM	120 to 12D	Transfer source data area (13 words)
R	100 101	Execution request (FUN 201 execution control area writing) Execution request (FUN 201 execution status area reading)

Program example

(3) Data passed from the CPU to the EH-ETH

Using the following program example, the data to be set in the transfer source data area that will be passed from the CPU to the EH-ETH is described.
For more details, see the EH-ETH manual.

I/O NO.	Data	Description	Remark
WM100	H000F	Module control register	
WM101	H0000	Connection open/close request	
WM102	H003F	Dedicated transmission control	
WM103	H003F	Automatic transmission control	
WM104	H003F	Dedicated reception control	
WM105	H003F	Automatic reception control	
WM106	H3F3F	Error clear 1 control	
WM107	H3F3F	Error clear 2 control	
WM108	H3F3F	Error clear 3 control	



Perform parameter area initialization settings with 1 scan execution on.
Write request to the control area
Set M0000 in the read control bit I/O number from unit number 0, slot number 0, word location word 0.
Set WM100 in the transfer source header I/O number.
Set the read size to 9 words.

Read request from the status area
Set M0010 in the read control bit I/O number from unit number 0, slot number 0, word location word 0.
Set WM120 in the transfer source header I/O number.
Set the read size to 13 words.

Start reading when the execution flag M0000 turns on

Contact point R100 turns off when the normal end flag M0001 turns on.

Contact point R101 turns on when the normal end flag M0001 turns on.

Contact point R100 turns off when the abnormal end flag M0002 turns on.

After writing to the control area is completed normally, start reading from the status area.

Start reading when the execution flag M0010 turns on.

Contact point R101 turns on when the normal end flag M0011 turns on.

Contact point R101 turns off when the abnormal end flag M0012 turns on.

Error code details

FUN200/FUN201 command error code list

Return code	Name	Description	Corrective action
0000	Normal end	Read/write operation ended normally.	
0002	S parameter area range error	Argument s has exceeded the maximum word I/O number.	Change the header I/O number of argument s.
0004	Control bit table range error	The control bit table has exceeded the maximum bit I/O number.	Change the bit header I/O number of the control bit table.
0006	Control type setting error	The control type setting is incorrect.	Set the correct value for the control type.
0008	Target area header setting error	The header setting of the target area is incorrect.	Set the unit number within a range of 0 to 7, set the slot number within a range of 0 to 7, and set the word location within the proper range.
000A	Target area range error*1	The target area header word location + size – 1 has exceeded the last word location.	Set the target area within the correct range.
000C	Transfer source (destination) header I/O number range error	The transfer source (destination) header I/O number has exceeded the maximum word I/O number.	Change the transfer source (destination) header word I/O number.
000D	Transfer source (destination) area range error*1	The transfer source (destination) header I/O + size – 1 has exceeded the maximum word I/O number.	Change the transfer source (destination) area within the correct range.
000E	Overlap error	The area for argument s and the area specified by argument s overlap.	Perform settings so the area for argument s and the area specified by argument s do not overlap.
0010	No assignment	There is no assignment in the specified unit and slot.	Perform assignment.
0020	Installation error	A module that supports handshaking is not installed.	Change the control type to forced read operation or forced write operation.
0022	Non-existent read or write area	There is no area in which read or write operation can be performed while controlling handshaking.	Change the control type to forced read operation or forced write operation.
0030	Timeout*2	No response from the module.	Check the attachment to the CPU basic base and restart. Or exchange the module.
0040	Conflict error	Read or write operation was simultaneously attempted to the same target area, or the module was reset while reading or writing.	Do not startup simultaneously or change the target area.

*1 An error will be generated, but the read/write processing will be performed within the specified range.

*2 1] A timeout will be generated if a response is not received within 100 ms after handshake is controlled and read/write operation is started.
A timeout will be generated if a response is not received within 5 s after software reset is started.
2] Note that processing new is not received during timeout detection.
Please perform the next processing after checking an abnormal end flag and a normal end flag.

Item number	Fun commands-66	Name	Data logging initialization setting †

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU3**A EH-CPU3**		Other than left		
FUN 210 (s)	DER	ERR	SD	V	C							
* (LOGIT (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					1,728	3,121	2906 10,087	5492 19,550	6,163	11,663	
	Condition			Steps								
FUN 210 (s) * (LOGIT (s))	—			3								

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Data logging management table							○						

Function

- Initializes the data logging function (common area).
- The initialization settings will be used to check and initialize the parameters for log data write and log data clear processes.
- Always perform the initialization setting once during operation.

* ( ) indicates the display when the LADDER EDITOR is used.
For the details of the data logging management table, see “Management table details.”
For the error codes that are set, see “Error code details.”

Cautionary notes

- If an error is generated, an error code will be set to the error code in the data logging management table, and DER will be set to “1.” In this case, initialization setting will not be performed.
- An error will be generated if initialization is performed again after the initialization setting has already been performed once (initialization of the data logging management table has been successfully completed).
- An error will be generated if the argument S (data logging management table) exceeds the maximum value for the I/O number.
- An error will be generated if there is an error in the verification process of items that have been specified for initialization in (S+3).

† : Supported by EH-CPU 308(A)/316(A)/448(A)516/548 only.

Program example

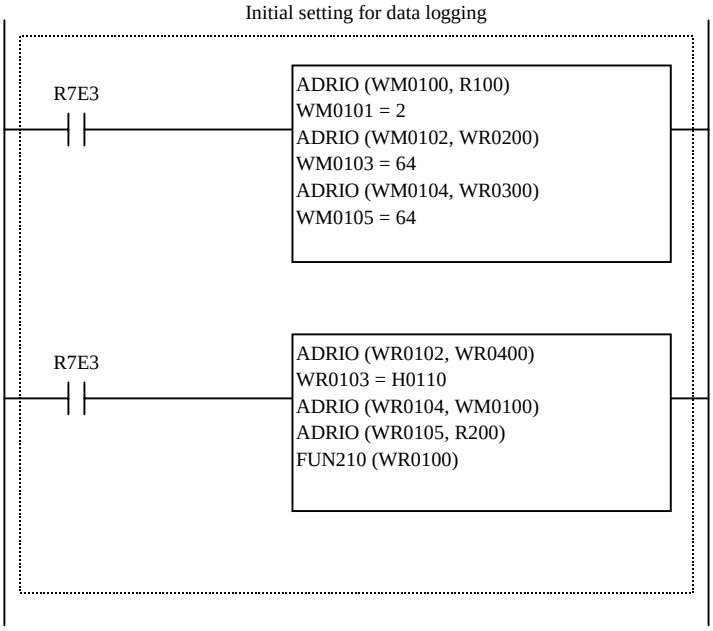
The following shows a sample program that performs initial settings for data logging.

This program initializes the log write parameter table and log clear parameter bit table.

Assigning internal outputs

This sample program is created using the following assignments. In actual cases, change the I/O numbers and other items according to the application.

I/O	No.	Usage	Remarks
WR	0100 to 0105	FUN 210 command Data logging management table (s to s+5)	See the data logging management table
WR	0400 to 0404	Log information table	See the log information table
WM	0100 to 0105	Log write parameter table	See the log write parameter table
R	0100 to 0103	Log write control bit table	See the log write control bit table
WR	0200 to 023F	Log data group 1	This I/O content is logged
WR	0300 to 033F	Log data group 2	This I/O content is logged
R	0200 to 0202	Log clear parameter bit table	See the log clear parameter bit table



Setting of actual address for log information table.
Initialization of log write parameter table and log clear parameter bit table.
Setting of actual address for log write parameter table.
Setting of actual address for log clear parameter bit table.
Execution of initial setting for data logging.

Program example

1. This program performs the initial settings of the log information table, log write table and log clear parameter bit table when a single scan execution (R7E3) is on.

Management table overview

See “Management table details” for detailed description.

(a) Logging management table (6 words)

s + 0	1) Error codes
s + 1	2) Initialization result
s + 2	3) Logging information table I/O number
s + 3	4) Initialization specification
s + 4	5) Log write parameter I/O number
s + 5	6) Log clear parameter I/O number

(b) Logging information table (5 words)

+ 0	Number of logs allowed (Lower)
+ 1	Number of logs allowed (Upper)
+ 2	Log size (words/log)
+ 3	Number of logs (Lower)
+ 4	Number of logs (Upper)

Exists when initialization specification (S+3) for log write is enabled.

(c) Log write parameter table (2 + MAX (128 *2) words)

+ 0	Log write control bit I/O Number
+ 1	Number of log groups (n) MAX 128
+ 2	Log groups 1 head I/O number
+ 3	Log groups 1 size
+ 4	Log groups 2 head I/O number
+ 5	Log groups 2 size
	_ _
+ (n + 2) + 0	Log groups n head I/O number
+ (n + 2) + 1	Log groups n size

MAX (128 *2) words

(d) Log write control bit table (4 bits)

+ 0	EXECUTE flag
+ 1	RUN flag
+ 2	WRITE flag
+ 3	ABNORMAL COMPLETION flag

Exists when initialization specification (S+3) for log clear is enabled.

(e) Log clear parameter bit table (3 bits)

+ 0	EXECUTE flag
+ 1	RUN flag
+ 2	ABNORMAL COMPLETION flag

* Description of cells

Area set by the user
Area where the user cannot write

Management table details

(a) Explanation of data logging management table (1/2)

1] Data logging management table

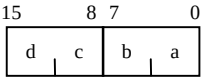
S	Error code	2]
S + 1	Initialization result	3]
S + 2	Logging information table I/O number	4]
S + 3	Initialization specifications	5]
S + 4	Log write parameter I/O number	6]
S + 5	Log clear parameter I/O number	7]

1] Data logging management table

1. Set in the FUN 210 (S) the word head I/O number used as the data logging management table.
2. The data logging management table consists of 2) through 7) and the size is 6 words. Make sure the maximum value for the word head I/O number is not exceeded.
   If the maximum value is exceeded, H0002 will be set to the error code.

2] Error code (Read)
Sets the error code generated in the FUN 210 process.

3] Initialization result (Read/Common area)
Sets the initialization result, and whether or not execution is allowed.

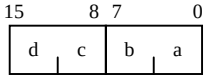


a. (b0 to b3) Whether or not re-initialization is allowed is set.
   0 : Allowed
   Other than 0 : Not allowed
b. (b4 to b7) Whether or not execution of the FUN 211 (Log Data Write) is allowed is set.
   0 : Not allowed
   Other than 0 : Allowed
c. (b8 to b11) Whether or not execution of the FUN 212 (Log Data Clear) is allowed is set.
   0 : Not allowed
   Other than 0 : Allowed
d. (b12 to b15) Whether or not execution of the FUN 213 (Log Data Read) is allowed is set.
   0 : Not allowed
   Other than 0 : Allowed

* If recovery is not possible by use of commands, b0 to b3 will become other than 0 and b4 to b15 will become 0.

4] Logging information table I/O number (Write/Common area)
Sets the word head I/O number that stores the current logging information data.
See “(b) Explanation of logging information word table.”

5] Initialization specifications (Write)
Sets the parameter table used.



a. (b0 to b3) Open (Unused)
b. (b4 to b7) Initialization of log write parameter table
   0 : Not initialized
   Other than 0 : Initialized
c. (b8 to b11) Initialization of log clear parameter table
   Other than 0 : Initialized
d. (b12 to b15) Open (Unused)

Management table details

(a) Explanation of data logging management table (2/2)

6] Log write parameter I/O number (Write)

Sets the word head I/O number used in the FUN 211 (Log Data Write) process.
When the write Function is not used, set H0000 as a dummy argument.
See “(c) Explanation of log write parameter word table.”

7] Log clear parameter I/O number (Write)

Sets the bit head I/O number of the parameter used in the FUN 212 (Log Data Clear) process.
When the clear function is not used, set H0000 as a dummy argument.
See “(e) Explanation of log clear parameter bit tables.”

Management table details

(b) Explanation of log information word table

1] Log information word table

+ 0	Number of logs allowed	: lower (R)	2]
+ 1	Number of logs allowed	: upper (R)	
+ 2	Log size (words/log)	(R)	3]
+ 3	Number of logs	: lower (R)	4]
+ 4	Number of logs	: upper (R)	

(R) Read word

1] Log information word table

1. Set the actual address (WR, WL, WM) of the word head I/O number that is used as the table into the data logging management table 4) using the ADRIO command.
2. The table consists of 2) through 4), and the size is 5 words. Make sure the maximum value for the word I/O number is not exceeded. If the maximum value is exceeded, H0003 will be set to the error code in the data logging management table.

2] Number of logs allowed

1. Indicates the number of logs currently allowed.
2. The value calculated by the formula shown below is set.
   Number of logs allowed = 384 k (393216) words ÷ 3) Log size
   (*) Round off below the decimal point.

3] Log size (words/log)

Indicates the number of words (Total number of words of respective log groups) per data groups written for a single log.

4] Number of logs

The number of logs that have been written at present is set. The number does not include the log whose data is currently being written. The count is increased by 1 when all data for that log has been written.

Management table details

(c) Explanation of log write parameter word table

1] Log write parameter word table

+ 0	Log write control bit I/O number	2)
+ 1	Log groups quantity (n)	3)
+ 2	Log groups 1 head I/O number	4)
+ 3	Log groups 1 size	
+ 4	Log groups 2 head I/O number	
+ 5	Log groups 2 size	
•	•	
•	•	
+ (n × 2) + 0	Log groups n head I/O number	
+ (n × 2) + 1	Log groups n size	

1] Log write parameter word table

1. Set the actual address (WR, WL, WM) of the word head I/O number that is used as the table into the data log management table 6) using the ADRIO command.
2. The table consists of 2) through 4), and the size increases depending on the log groups quantity 3). Make sure the maximum value for the word head I/O number is not exceeded. If the maximum value is exceeded, H0005 will be set to the error code in the data logging management table.

2] Log write control bit I/O number

Set the bit I/O number used in FUN 211 control.
See (d), “Explanation of log write control bit table.”

3] Log groups quantity

Set the number of logging groups in the range between 1 and 128.
If the specified value exceeds the allowable range, H0004 will be set to the error code in the data logging management table. If this error occurs, data logging will not be performed.

4] Log selection

Select the elements that comprise a single logging.

1. Head I/O number 1 to n ... Set each number using the ADRIO command by making sure the maximum value for the word head I/O number (WR, WL, WM, WX), from which the data will be logged, is not exceeded.
2. Size ····· Set the range for the number of words logged from the word head I/O number.
3. If the maximum I/O number is exceeded, Hxx08 will be set to the error code in the data logging management table. If this error occurs, data logging will not be performed.
4. If the specified size is not between 1 and 128, H0007 will be set to the error code in the data logging management table. If this error occurs, data logging will not be performed.
5. The total size (number of words) of the selected data items must be in the range between 1 and 128 words. If the above range is exceeded, H0009 will be set to the error code in the data logging management table. If this error occurs, data logging will not be performed.

Management table details

(d) Explanation of log write control bit table

1] Log write control bit table

+ 0	EXECUTE flag	(W)	2)
+ 1	RUN flag	(R)	3)
+ 2	WRITE flag	(R)	4)
+ 3	ABNORMAL COMPLETION flag	(R)	5)

(W) Set and read bit

(R) Read bit

1] Log write control bit table

1. Set the actual address (R, L, M) of the bit head I/O number that is used as the table into the log write parameter table 2) using the ADRIO command.
2. The table consists of 2) through 5), and the size is 4 bits. Make sure the maximum value for the bit I/O number is not exceeded. If the maximum value is exceeded, H0006 will be set to the error code in the data logging management table.

2] EXECUTE flag

1. When the EXECUTE flag rises (0 → 1), the system waits for the start of data logging.
2. If the execution ready state is achieved normally, the RUN flag 3) becomes 1. If there is an error, the RUN flag 3) becomes 0.
3. Logging is performed while the EXECUTE flag is 1.

3] RUN flag

1. When the FUN 211 detects the rise of the EXECUTE flag 2), it checks if execution is allowed and sets the result.
   1 : Check result is normal. Data logging is started.
   0 : Check result is abnormal. Data logging is not performed. The EXECUTE flag 2) becomes 0 and the cause of error is set to the FUN 211 (log data write) error code.
2. When the FUN 211 detects the fall of the EXECUTE flag 2) while the RUN flag is 1, and if the WRITE flag 4) is 1 (data is being written), the RUN flag will become 0 when the write is completed and the data logging will end.

4] WRITE flag

This flag indicates whether or not the data that has been stored temporarily via the FUN 211 is currently being written.

1 : Data write is being executed.

0 : Data write is yet to be executed.

5] ABNORMAL COMPLETION flag

This flag indicates whether or not the FUN 211 (data logging write) has been executed normally.

1 : Abnormal completion. The cause of error is set to the FUN 211 (log data write) error code.

0 : Normal completion. Indicates a state where no error has been generated.

Management table details

(e) Explanation of log clear parameter bit table

1] Log clear parameter bit table

+ 0	EXECUTE flag	(W)	2)
+ 1	RUN flag	(R)	3)
+ 2	ABNORMAL COMPLETION flag	(R)	4)

(W) Set and read bit

(R) Read bit

1] Log clear parameter bit table

1. Set the actual address (R, L, M) of the bit head I/O number that is used as the table into the data log management table 7) using the ADRIO command.
2. The table consists of 2) through 4), and the size is 3 bits. Make sure the maximum value for the bit I/O number is not exceeded. If the maximum value is exceeded, H000A will be set to the error code in the data logging management table.

2] EXECUTE flag

1. When the EXECUTE flag rises (0 → 1), the system waits for the start of log data clear.
2. If the execution ready state is achieved normally, the RUN flag 3) becomes 1. If there is an error, the RUN flag 3) becomes 0.

3] RUN flag

1. When the FUN 212 detects the rise of the EXECUTE flag 2), it checks if execution is allowed and sets the result.
   1 : Check result is normal. Log data clear is started.
   0 : Check result is abnormal. Log data clear is not performed. The EXECUTE flag 2) becomes 0 and the cause of error is set to the FUN 212 (log data clear) error code.
2. When the clear is completed, both the RUN flag and EXECUTE flag 2) become 0. If the FUN 212 detects the fall of the EXECUTE flag 2) while the RUN flag is 1 (data is being cleared), the RUN flag will become 0 when the clear is completed and the log data clear will end.

4] ABNORMAL COMPLETION flag

This flag indicates whether or not the FUN 212 (log data clear) has been executed normally.

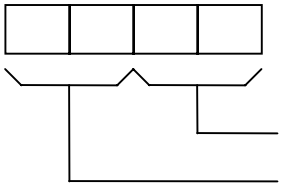
1 : Abnormal completion. The cause of error is set to the FUN 212 (log data clear) error code.

0 : Normal completion. Indicates a state where no error has been generated.

Error code details

(a) Explanation of log function error code

Each error code is expressed as a four-digit hexadecimal number.



The last 2 digits indicate the cause of error.

These digits indicate the log groups number.

If H00, it indicates that this error is not related to the log groups number.

If between H01 and H80, it indicates that there is an error regarding the log groups number.

In the error code list, the error codes in which the first 2 digits are indicated by ×× have a log groups number between H01 and H80 in place of ××. Numbers in brackets and ○ used in the table below indicate the corresponding numbers in "Management table details."

Error code	Check point/output destination				Description and cause	Corrective action
	Initial	Write	Clear	Read		
0001	○	×	×	×	The FUN 210 was executed again after its re-initialization had become disabled.	Modify the program so that the FUN 210 will not be executed after its initialization is completed.
0002	○	○ *3	○ *3	○	The argument S exceeds the maximum value for the word I/O number.	Change the head I/O number of the argument S.
0003	○	×	×	×	The log information table (a) 4) exceeds the maximum value for the word I/O number.	Change the head word I/O number (a) 4).
0004	○ *1	×	×	×	The log groups quantity (c) 3) exceeds the range between 1 and 128.	Set the log groups quantity (c) 3) in the range between 1 and 128.
0005	○ *1	×	×	×	The log write parameter table (a) 6) exceeds the maximum value for the word I/O number.	Change the head word I/O number (a) 6)/ log groups quantity (c) 3).
0006	○ *1	×	×	×	The log write control bit table (c) 2) exceeds the maximum value for the bit I/O number.	Change the head bit I/O number (c) 2).
××07	○ *1	×	×	×	Groups size (c) 4) number ×× exceeds the range between 1 and 128.	Set the groups size (c) 4) corresponding to ×× in the range between 1 and 128.
××08	○ *1	×	×	×	Logging groups (c) 4) number XX exceeds the maximum value for the world I/O number.	Change the head word I/O number (c) 4)/ groups size (c) 4) corresponding to XX.
0009	○ *1	×	×	×	The total size of the log groups exceeds the range between 1 and 128.	Set the total size of the log groups so it falls in the range between 1 and 128.
000A	○ *2	×	×	×	The log clear parameter table (a) 7) exceeds the maximum value for the bit I/O number.	Change the head bit I/O number (a) 7).
000B	×	○	○	○	A FUN was executed although its execution was not allowed (3) initialization result was "execution not allowed").	Execute the FUN after confirming that the initialization result has become "execution allowed."
000C	×	×	×	○	The specified file address does not exist.	Set the file address between H0000 and HFFFF (lower) or H0000 and H0005 (upper).
000D	×	×	×	○	The acquisition source area exceeds the maximum value for the word I/O number.	Change the head I/O number or size of the acquisition source.
000E	○	×	×	○	The area of the argument S and the area specified by the argument S are overlapping.	Set the area of the argument S so it does not overlap with the area specified by the argument S.
000F	×	×	×	○	There is no log data in the specified file address.	Calculate the value of file address again by referring to the log information table (a) 4).
0010	○	○	×	×	The size of the single log that was previously written to the memory board is different from the total size of the log groups.	Modify the program according to the previous log size. Or, execute the FUN 212 command to clear the data, then execute again.
0011	×	×	×	○	The read size exceeds 128.	Set the read size in the range between 1 and 128.

Error code details

Error code	Check point/output destination				Description and cause	Corrective action
	Initial	Write	Clear	Read		
0012	×	○	×	×	The log data area became full and no new log data could be added.	To recover this error, read the current data using the FUN 213 command, then clear the data using the FUN 212 command. When the data has been cleared, execute again.
0020	○	○	○	○	No memory board is installed, or the memory board does not support the logging function.	Set a memory board that supports the logging function.
0021	×	○	○	×	Operation was attempted in the write-prohibited mode.	Change the dip switch setting to the write-enable mode.
0022	○	○	○	×	Hardware error was detected in the memory board.	The memory board may be damaged. Replace the memory board.
0023	○	○	×	×	The previous logging has failed.	To recover this error, execute the FUN 212 command to clear the data, then execute again.
0024	×	○	○	○	Operation was attempted in a mode other than the data logging mode.	Change the dip switch setting to the data logging mode.

* Initial: FUN 210 (initial setting)
Write: FUN 211 (write)
Clear: FUN 212 (clear)
Read: FUN 213 (read)

*1: When the initialization of log write is specified.

*2: When the initialization of log clear is specified.

*3: Cannot be output.

Item number	Fun commands-67	Name	Log data write †

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU3**A EH-CPU3**		Other than left		
FUN 211 (s)	DER	ERR	SD	V	C							
* (LOGWRT (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					645	1,188	527 / 1,025	959 / 1,587	973	1,535	
	Condition			Steps								
FUN 211 (s) * (LOGWRT (s))	—			3								

Usable I/O	Bit				Word				Double word			Constant	Other
	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s Error code							○						

Function

- This command is used to log (i.e., write to the memory board) the items specified by the FUN 210 (initial setting for data logging).

* ( ) indicates the display when the LADDER EDITOR is used.

For the data logging management table, see “Management table details.”
For the error codes to be set, see “Error code details.”

Cautionary notes

- If an error occurs, an error code is set and DER becomes equal to “1.” If this happens, the log data write will not be performed.
- If this command is executed when the initialization result in the data logging management table corresponding to the FUN 210 (initial setting for data logging) is “execution not allowed,” an error will occur.
- An error will occur if the argument S exceeds the maximum value for the I/O number.
- Do not set a startup condition for this command. Execute it during normal scans only.
- An error will occur if the dip switch setting on the memory board specifies write prohibited or any mode other than the data logging mode.
- If the memory board is currently in use for other processing, DER becomes equal to “0” and R7F7 becomes equal to “1.” If this happens, the processing will not be performed.

† : Supported by EH-CPU 308(A)/316(A)/448(A)516/548 only.

Program example

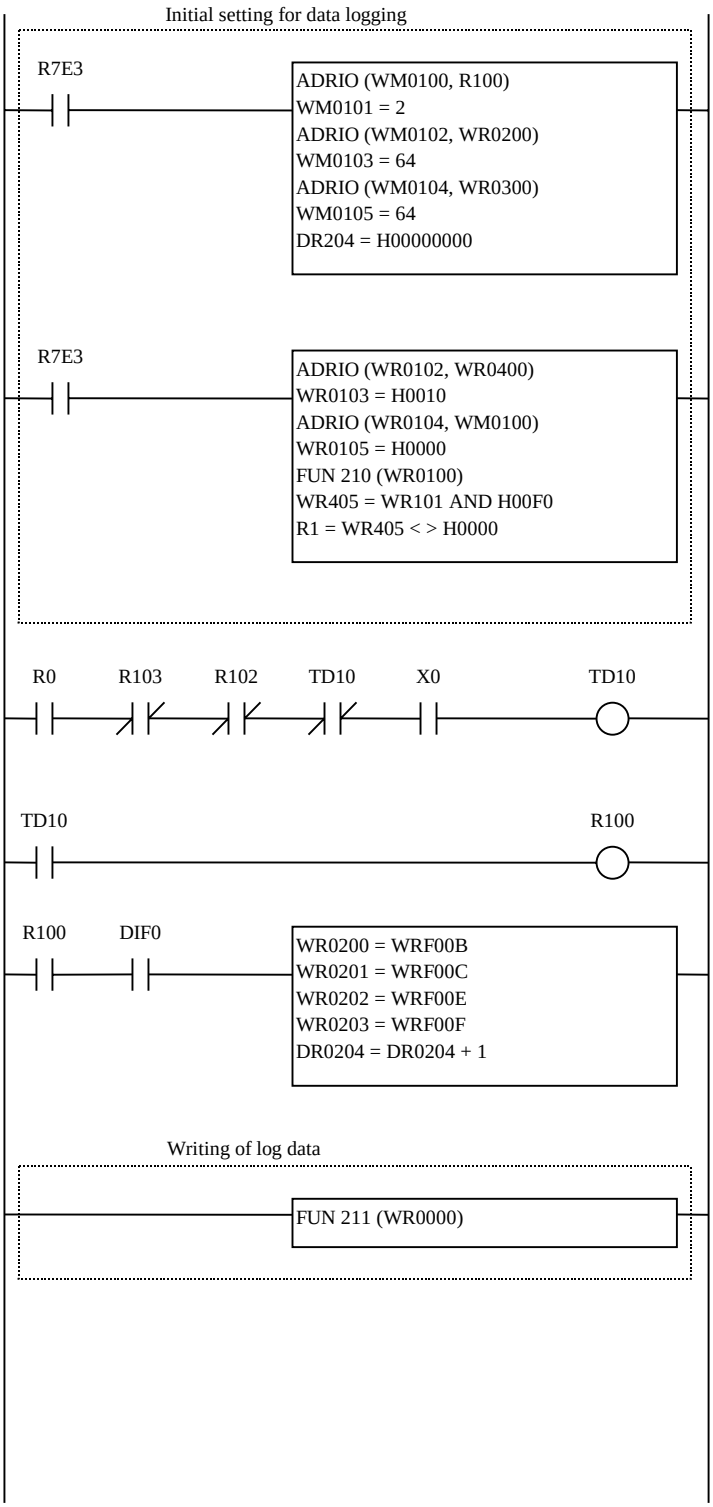
The following shows a sample program that writes into the logging area a total of 128 words including WR200 through WR23F (64 words) and WR300 through WR33F (64 words) at intervals of 10 minutes after ON of X00000 (the start bit), together with date data (year/month, day/hour, minute/second) and sequence number.

Assigning internal outputs

The sample program is created using the following assignments. In actual cases, change the I/O numbers and other items according to the application.

I/O	No.	Usage	Remarks
WR	0100 to 0105	FUN 210 command Data logging management table (s to S+5)	See the data logging management table
WR	0400 to 0404	Log information table	See the log information table
WM	100 to 105	Log write parameter table	See the log write parameter table
R	100 to 103	Log write control bit table	See the log write control bit table
WR	0200 to 023F	Log data groups 1	This data is logged
WR	0300 to 033F	Log data groups 2	This data is logged
DR	0204	Sequence number	Increments by 1 after each logging
WR	0405	Save area for initialization result	Only the result of the FUN 211 is set
R	0	Save bit for initialization result	If ON, it indicates normal completion (execution allowed)
X	0	Log start bit	Logging interval (10 minutes) monitoring start bit
TD	10	10 minute monitoring timer	Turns on when 10 minutes have elapsed
WR	0000	FUN 211 command error code (s)	The error code that has been generated as a result of the log data write is set

Program example



Create a write parameter in WM0100 that logs 128 words (64 words from WR200 and 64 words from WR300) upon turning ON of single scan execution.
Clears the sequence number to 0.

Specify initial setting so that the specified item (WM0100) is logged in WR0100 upon turning ON of single scan execution.
Sets the initialization result in R000 after the initial setting.

1S 600 (Turns ON when 10 minutes have elapsed)
The progress value is updated while the start bit is ON but only when both the initial setting and logging have been completed normally.

Logging is started when TD10 is turned ON (when 10 minutes have elapsed).

Date data (year/month, day/hour, minute/second) and sequence number are appended to the head of log data (WR200 to WR205) when the logging execution flag is turned ON (R100 OFF → ON).

Executes the logging.

Item number	Fun commands-68	Name	Log data clear †

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU3**A EH-CPU3**		Other than left		
FUN 212 (s)	DER	ERR	SD	V	C							
* (LOGCLR (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					163	164	79 / 238	79 / 443	391	←	
	Condition			Steps								
FUN 212 (s) * (LOGCLR (s))	—			3								

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Error code							○						

Function

- This command is used to forcibly erase (clear) the logged data in the memory board.

* ( ) indicates the display when the LADDER EDITOR is used.
For the data logging management table, see “Management table details.”
For the error codes to be set, see “Error code details.”

Cautionary notes

- If an error occurs, an error code is set and DER becomes equal to “1”. If this happens, the log data clear will not be performed.
- If this command is executed when the initialization result in the data logging management table corresponding to the FUN 210 (initial setting for data logging) is “execution not allowed,” an error will occur.
- An error will occur if the argument S exceeds the maximum value for the I/O number.
- Do not set a startup condition for this command. Execute it during normal scans only.
- An error will occur if the dip switch setting on the memory board specifies write prohibited or any mode other than the data logging mode.
- If the memory board is currently in use for other processing, DER becomes equal to “0” and R7F7 becomes equal to “1.” If this happens, the processing will not be performed.

† : Supported by EH-CPU 308(A)/316(A)/448(A)516/548 only.

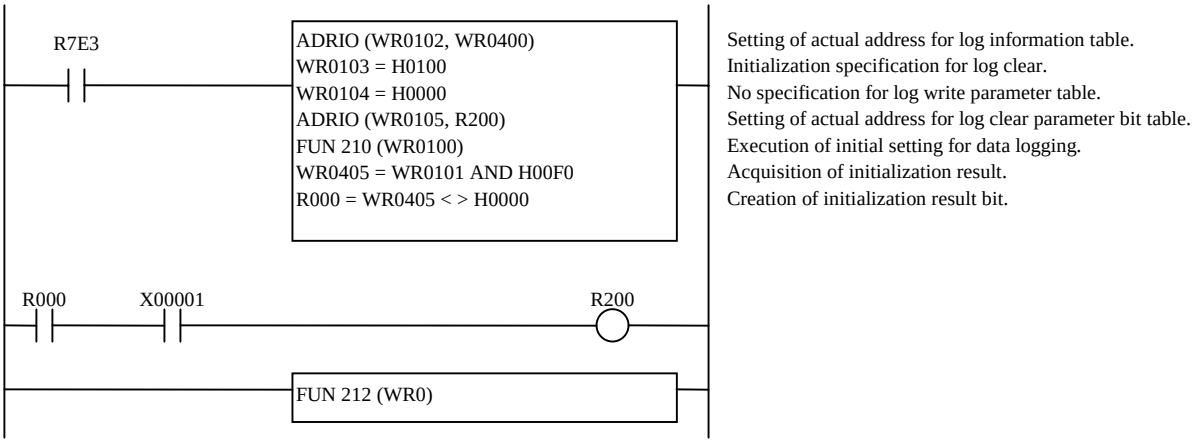
Program example

The following shows a sample program that performs initial setting for logging area clear and then clears the logging area.

Assigning internal outputs

The sample program is created using the following assignments. In actual cases, change the I/O numbers and other items according to the application.

I/O	No.	Usage	Remarks
WR	0100 to 0105	FUN 210 command Data logging management table (s to S+5)	See the data logging management table
WR	0400 to 0404	Log information table	See the log information table
R	200 to 202	Log clear parameter bit table	See the log clear parameter bit table
WR	0405	Save area for initialization result	Only the result of the FUN 212 is set
R	000	Save bit for initialization result	If ON, it indicates normal completion (execution allowed)
X	00001	Clear start bit	Logging clear start bit
WR	0000	FUN 212 command error code (s)	The error code that has been generated as a result of the log data clear is set



Program description

The log area is cleared when X00001 (the clear start bit) is turned ON after the initial setting for log area clear has been executed upon turning ON of single scan execution (R7E3) and completed normally.

Item number	Fun commands-69	Name	Log data read †

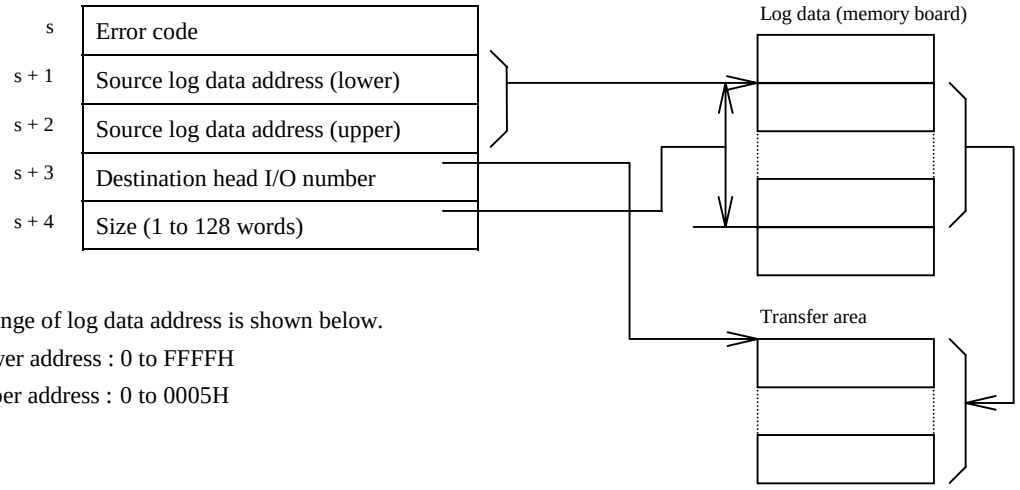
Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU3**A EH-CPU3**		Other than left		
FUN 213 (s)	DER	ERR	SD	V	C							
* (LOGRED (s))	↕	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	

Command format	Number of steps										
	Condition	Steps	268	357	210	299	560	645			
FUN 213 (s)	—	3			547	633					
* (LOGRED (s))											

Usable I/O	Bit				Word				Double word			Constant	Other
	X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s Argument							○						

Function

The logged data in the memory board is transferred to the specified I/O number area by the amount of data corresponding to the number of words being specified by the source log data address (word position from the head of log data).



The range of log data address is shown below.

Lower address : 0 to FFFFH

Upper address : 0 to 0005H

- The valid maximum log data address is calculated by the formula shown below :
  (Log size in the log information table × Number of logs) – 1
- When the log data read is completed normally, DER becomes equal to "0."
- If the specified size is 0, nothing will be done and the process will complete normally.
- For the S+3 parameter, set the actual address (WR, WL, WM) using the ADRIO command.

* ( ) indicates the display when the LADDER EDITOR is used.

For the data logging management table, see "Management table details."

For the error codes to be set, see "Error code details."

† : Supported by EH-CPU 308(A)/316(A)/448(A)516/548 only.

Cautionary notes

- If an error occurs, an error code is set and DER becomes equal to “1”. If this happens, the log data read will not be performed.
- If this command is executed when the initialization result in the data logging management table corresponding to the FUN 210 (initial setting for data logging) is “execution not allowed,” an error will occur.
- An error will occur if the specified log data address does not exist.
- If the specified log data does not exist, an error will occur after the read.
- An error will occur if the area specified by the argument S or S+3 exceeds the maximum value for the I/O number.
- An error will occur if the area specified by the argument S and the one specified by S+3 overlap.
- An error will occur if the dip switch setting on the memory board specifies any mode other than the data logging mode.
- If the memory board is currently in use for other processing, DER becomes equal to “0” and R7F7 becomes equal to “1.” If this happens, the processing will not be performed.

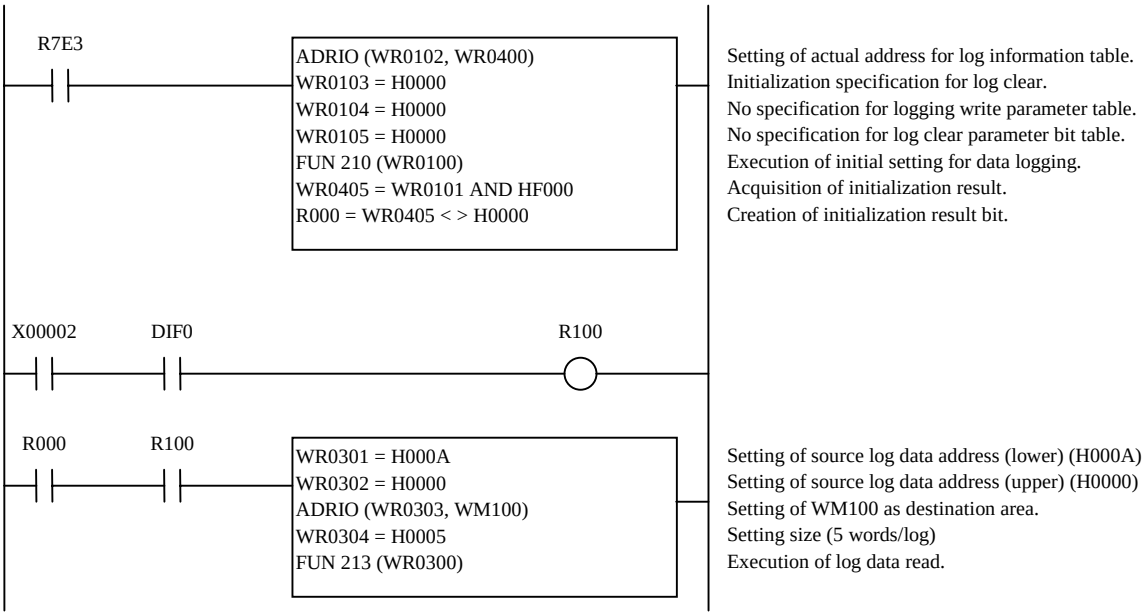
Program example

The following shows a sample program that performs initial setting for log data read and then reads the logged data.

Assigning internal outputs

The sample program is created using the following assignments. In actual cases, change the I/O numbers and other items according to the application.

I/O	No.	Usage	Remarks
WR	0100 to 0105	FUN 210 command Data logging management table (s to S+5)	See the data logging management table
WR	0400 to 0404	Data logging information table	See the log information table
WR	0405	Save area for initialization result	Only the result of the FUN 213 is set
R	000	Save bit for initialization result	If ON, it indicates normal completion (execution allowed)
X	00002	Read start bit	Logging read start bit
WR	0300 to 0304	FUN 213 command parameter area (s to S+4)	
WM	0100 to 0104	Log data transfer area	The data that has been read is set

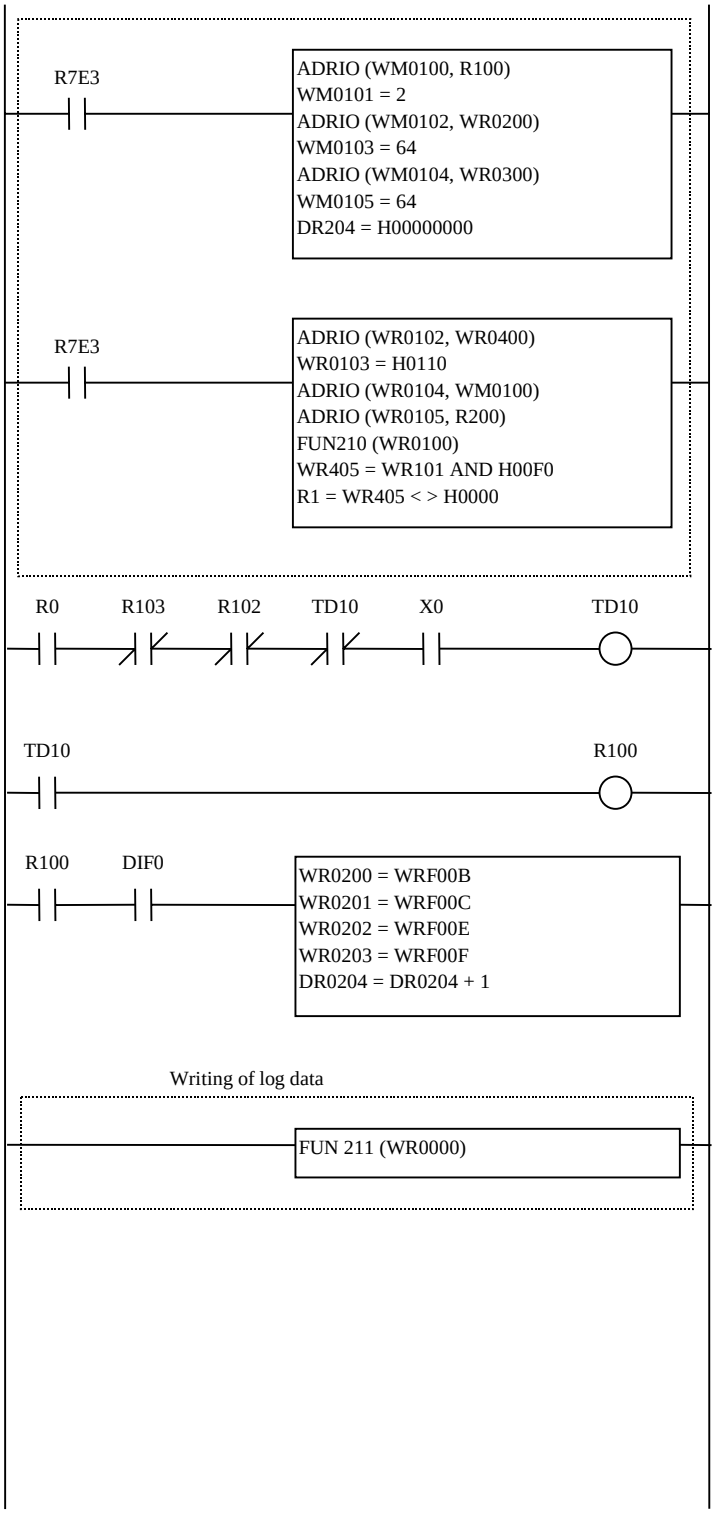


Program description

Of the three logs of data that have been written, with each log consisting of 5 words, the content of the third log (in the memory board) is transferred to the destination (WM100) by the amount of data specified for a single log (5 words), with X00002 (the read start bit) ON.

Program example

The following shows an example of a program that starts logging at every 10 minutes after X0 turns ON, reads the first and last log data when X0 is OFF, and then clears the log data.



Setting of actual address for logging information table
Initializes the log write parameter table and log clear parameter table.
Log write parameter table not specified
Log clear parameter bit table not specified
Data logging initialization setting executed

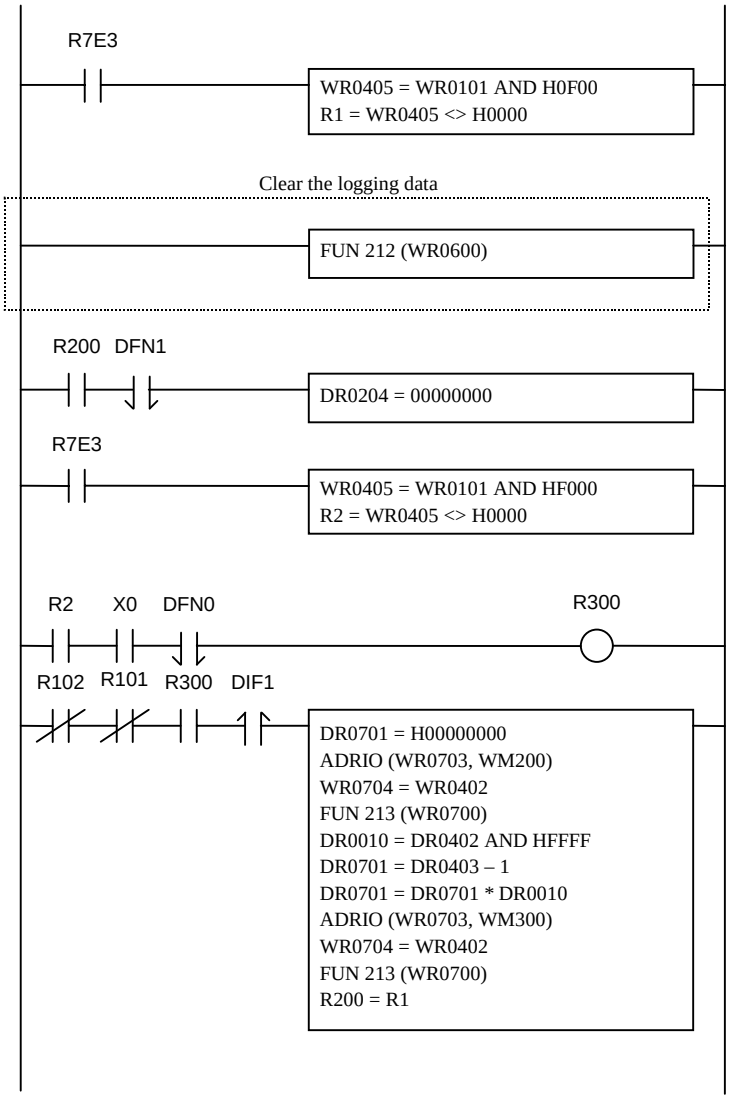
Specify initial setting so that the specified item (WM0100) is logged/cleared in WR0100 upon turning ON of single scan execution.
Sets the initialization result in R000 after the initial setting.

1S 600 (Turns ON when 10 minutes have elapsed)
The progress value is updated while the start bit is ON but only when both the initial setting and logging have been completed normally.

Logging is started when the TD10 (10 min has elapsed) is ON.

Date data (year/month, day/hour, minute/second) and sequence number are appended to the head of log data (WR200 to WR205) when the logging execution flag is turned ON (R100 OFF → ON).

Program example



Acquisition of initialization result.
Creation of initialization result bit (log clear).

Clears the sequence number upon completion of log data clear.

Acquisition of initialization result.
Creation of initialization result bit (log read).

Setting of head log address (H0000).
Setting of WM200 as destination area.
Setting size (128 words/log)
Execution of log data read (first log item).
Calculation of last log data address.

5.
Setting of WM300 as destination area
Setting size (128 words/log)
Execution of log data read (last log item).
Execution of log clear.

Item number	Fun commands-70	Name	BOX comment

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 254 (s)	DER	ERR	SD	V	C							
* (BOXC (s))	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					0.15	←	14 14	←	20	←	
	Condition			Steps								
FUN 254 (s) * (BOXC (s))	—			3								

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument (dummy constant)							○						

Function

- This command does not perform any operations. It is used to print comments on the right side of the calculation box in conjunction with the LADDER EDITOR.
- A comment can contain a maximum of 32 characters.

* ( ) indicates the display when the LADDER EDITOR is used.

Item number	Fun commands-71	Name	Memo comment

Ladder format	Condition code					Processing time (μs)						Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	EH-CPU4** EH-CPU5**		EH-CPU***A EH-CPU3**		Other than left		
FUN 255 (s)	DER	ERR	SD	V	C							
* (MEMC (s))	●	●	●	●	●	Ave	Max	Ave	Max	Ave	Max	
Command format	Number of steps					0.15	←	14.4 ..... 14.4	←	20	←	
	Condition			Steps								
FUN 255 (s) * (MEMC (s))	—			3								

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, L, M	TD, SS, WDT, MS, TMR, CU, RCU, CT	WX	WY	WR, WL, WM	TC	DX	DY	DR, DL, DM		
s	Argument (dummy constant)							○						

Function

- This command does not perform any operations. It is used to print comments on the right side of the calculation box in conjunction with the LADDER EDITOR.
- A comment can contain a maximum of one screen (66 characters × 16 lines).

* ( ) indicates the display when the LADDER EDITOR is used.

Chapter 6 I/O Specifications

Classification of I/O that can be used with the EH-150 as well as the I/O points are indicated in Table 6.1.

Table 6.1 Usable I/O classification and points

Function		Symbol	Size	10 / 16	Name	CPU104 /104A	CPU208 /208A	CPU308 /308A	CPU316 /316A	CPU448 /448A	CPU516	CPU548
External I/O	External I/O	X	B	10*	Bit external input	512 points **	1,024 points **				2,112 points **	3,520 points **
		Y	B	10*	Bit external output							
		WX	W	16	Word external input	8 words	16 words					
		WY	W	16	Word external output							
		DX	D	16	Double word ex. input							
		DY	D	16	Double word ex. output							
	Remote I/O	X	B	10*	Bit remote ex. input	-					4,096 points	
		Y	B	10*	Bit remote ex. output							
		WX	W	16	Word remote ex. input	-					256 words	
		WY	W	16	Word remote ex. output							
		DX	D	16	Double word remote ex. in.							
		DY	D	16	Double word remote ex. out.							
CPU link area		L	B	16	Bit link area 1	16,384 points						
		WL	W	16	Word link area 1	1,024 words						
		DL	D	16	Double word link area 1							
		L	B	16	Bit link area 2	16,384 points						
		WL	W	16	Word link area 2	1,024 words						
		DL	D	16	Double word link area 2							
Internal output	Bit	R	B	16	Bit internal output	1,984 points						
		R	B	16	Bit special internal output	64 points						
	Word	WR	W	16	Word internal output	4,096 words	8,192 words	17,408 words	22,528 words	50,176 words	22,528 words	50,176 words
		DR	D	16	D. word internal output							
		WR	W	16	Word special internal output	512 words						
		DR	D	16	D. word special internal output							
	Shared bit/word	M	B	16	Bit internal output	16,384 points						
		WM	W	16	Word internal output	1,024 words						
		DM	D	16	D. word internal output							
Others	Edge detection	DIF	B	10	Rise edge	512 points						
		DFN	B	10	Fall edge	512 points						
	Master control	MCS	B	10	Master control set	50 points						
		MCR	B	10	Master control reset							
	Timer, Counter	TD	B	10	On delay timer	Timer 256 points (0.01 s timer has only 0 to 63) * The same No. cannot be used to timer, counter. * TM is available with only EH-CPU448. Counter 512 points (The same area as timer is used up to 256 points) * The same No. cannot be used to timer, counter.						
		SS	B	10	Single-shot timer							
		WDT	B	10	Watchdog timer							
		MS	B	10	Monostable timer							
		TMR	B	10	Accumulative timer							
		CU	B	10	Up counter							
		RCU	B	10	Ring counter							
		CTU	B	10	Up-down counter up input							
		CTD	B	10	Up-down counter down input							
		CT	B	10	Up-down counter down output							
		CL	B	10	Progress value clear							
		TC	W	10	Timer/counter accumulator							
		TM	B	10	On delay timer	-					2,048 points 0.01s fixed	
		TV	W	10	Timer accumulator							

* The lower 2 digits of external bit X,Y are decimal (0-95) however, next digit (slot number) is hexa-decimal format (0-A). See table 6.3 for further information.

** In case 64 points module used.

6.1 External I/O

When operation start is performed for the EH-150, the user program is executed (scanned) after the input refresh processing (receiving external input data) is performed. Operations are performed according to the contents of the user program, and the next input refresh processing and output refresh processing (operation results are reflected in the external output) are performed. After that, the next user program is executed (scanned). This series of operations is continually repeated until operation is stopped or until something occurs where operation can no longer continue.
When operation is stopped or if something occurs so that operation can no longer continue, the CPU performs an output refresh with all output data as off data and stops operation, regardless of the execution status for the user program.
Diagram outlining this series of operations is indicated in Figure 6.1 below.

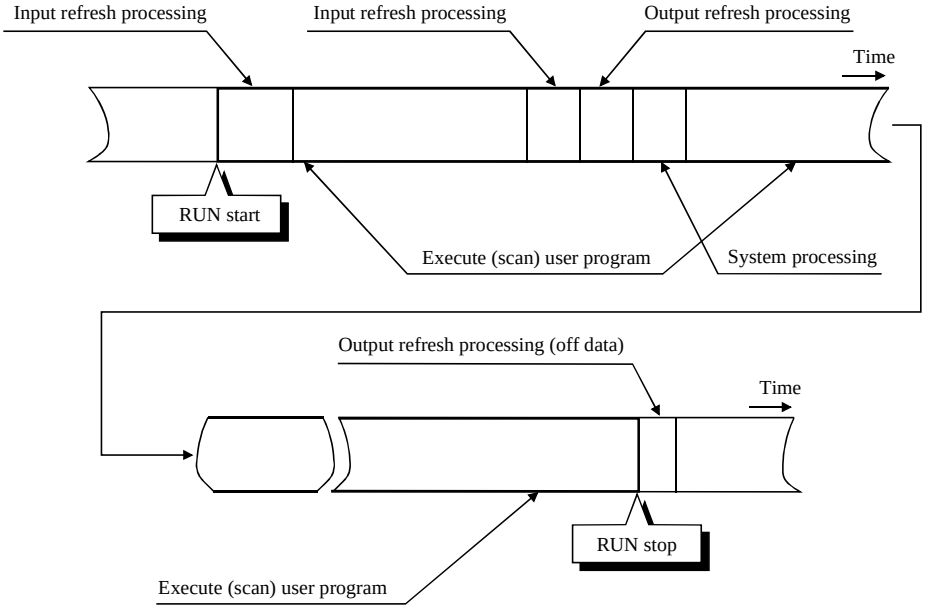


Figure 6.1 Overview of user program execution and refresh processing

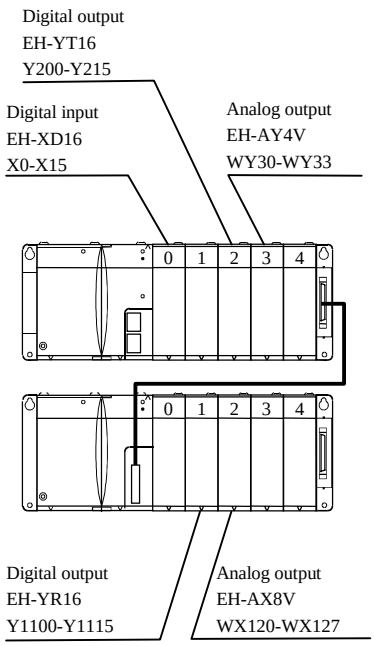
The user programs are executed in order, usually from the program at the beginning of the scan area to the end of the program or until the END command. After that, I/O data is refreshed prior to the execution of the next user program.
As shown above, external I/O data is updated in group in the refresh processing after the user program is executed. If it is necessary to update (refresh) the I/O data while the user program is being executed, use the refresh command (FUN80 to FUN82).
When designing the system, take into account the above refresh operation from when the input data is received and operated on until output data is obtained.

The next section explains the assignment of external I/O. The external I/O numbers for the EH-150 system are expressed with the following rules.

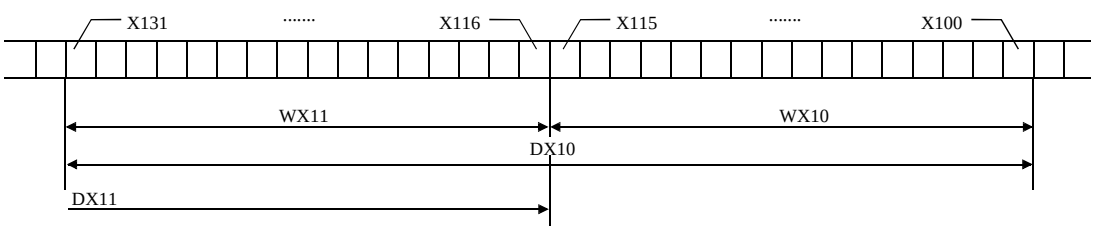
Table 6.2 List of external I/O classification and data type

Classification	I/O classification	Data type	Remarks
X	External input	Bit type	Corresponds to the signal of each terminal block.
WX		Word type (16-bit)	Data in the range 0 to 15 is batch processed. 16-bit synchronicity guaranteed.
DX		Double word type (32-bit)	Two word data are batch expressed. 32-bit synchronicity is not guaranteed.
Y	External output	Bit type	Corresponds to the signal of each terminal block.
WY		Word type (16-bit)	Data in the range 0 to 15 is batch processed. 16-bit synchronicity guaranteed.
DY		Double word type (32-bit)	Two word data are batch expressed. 32-bit synchronicity is not guaranteed.

Table 6.3 List of I/O number rules for external I/O

Data type	Numbering rule	Example
Bit type (basic/expansion)	X □ □ □ □□ ← External input Y □ □ □ □□ ← External output Bit number (00 - 95) Slot number (0 - A) Unit number (0 - 4) Remote number (0 - 4)	
Word type	WX □ □ □ □ WY □ □ □ □ Word number (0 - 7) Slot number (0 - A) Unit number (0 - 4) Remote number (0 - 4)	
Double word type	DX □ □ □ □ DY □ □ □ □ Word number (0 to 6) in the module Slot number (0 to A) Unit number (0 to 4) Remote number (0 to 4)	

The external I/O word type is a collection of 16 points, and double word type is a collection of 32 points of the relevant bit type.
(Example) Relationship between DX10, WX10 and X100 to X115 is,



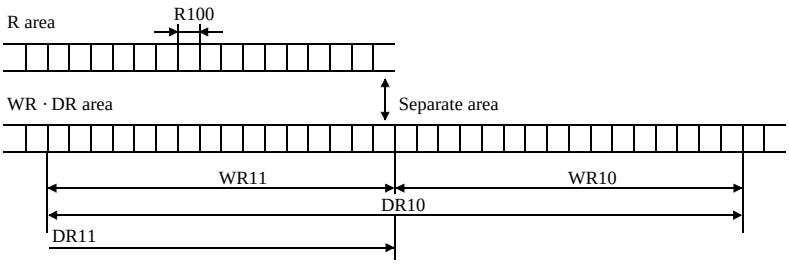
6.2 Internal Output

Memory in the CPU module is available as the area for internal outputs. There are three areas: bit dedicated area (R), word dedicated area (WR), and bit/word common area (M/WM).

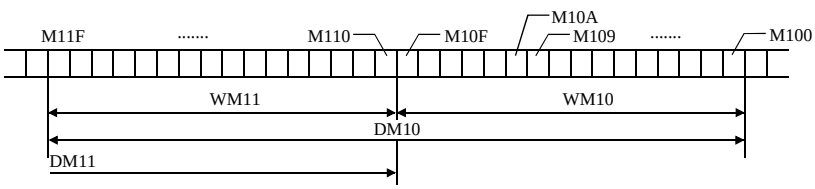
Table 6.4 List of I/O number rules for external I/O

Data type	Numbering rule	Example
Bit dedicated type	R □□□ Normal area H000 to H7BF Special area H7C0 to H7FF Both are expressed by hexadecimal	R0 R105 R23C R7E7
Word dedicated type	<Word> WR□□□□ Normal area H0000 to Special area HF000 to Both are expressed by hexadecimal	WR0 WR11 WR1234 WRF004
	<Double word> DR□□□□ Normal area H0000 to Special area HF000 to Both are expressed by hexadecimals. Expresses WR for 2 words in continuatio	DR0 DR11 DR1234 DRF004
Bit/word common type	<Bit> M □□□□ H0000 to	M0 M11 M1234
	<Word> WM□□□ H000 to M120F M1200 WM120	WM0 WM11 WM123
	<Double word> DM□□□□ H0000 to Expressed by hexadecimals. Expresses DM for 2 words in continuatio	DM0 DM11 DM234

- Internal output R and WR, DR are completely separate areas. Bit-based operations cannot be performed in WR.
  (Example) Relationship between R100 and WR10, DR10



- Since internal output M, WM and DM share the same area, bit-based operations are allowed.
  (Example) Relationship between M100 and WM10, DM10



6.3 Virtual remote I/O for internal output

Virtual remote I/O area (WY1000-) is available by assigning "Remote2" at empty slot in basic base as follows.

(1) Assign "Remote2" at empty slot in basic base.

(2) Assign any of "Empty 16/32/64/128" at I/O configuration table of slave station. Then virtual area is kept according to this setting points.

(3) Select "RUN" at operation mode of I/O assignment unmatched in your programming software.

(4) Y, WY, DY can be used as WM according to the table below.

[例] 割付の例

Basic unit	
0	X16
1	X16
2	X16
3	Y16
4	Y16
5	Y16
6	X8W
7	Y8W
8	**Remote2**
9	
A	

Remote master 1	
Empty 64	Y10000-Y10063, WY1000-WY1004
Empty 64	Y10100-Y10163, WY1000-WY1014
Empty 64	Y10200-Y10263, WY1000-WY1024
Empty 64	Y10300-Y10363, WY1000-WY1034
Empty 64	Y10400-Y10463, WY1000-WY1044
Empty 64	Y10500-Y10563, WY1000-WY1054
Empty 64	Y10600-Y10663, WY1000-WY1064
Empty 64	Y10700-Y10763, WY1000-WY1074

*1: It is possible to assign "Empty 128" however, bit access is up to Y95 while word access is 8 words.

*2 :Use Y,WY,DY. Input X,WX, DX are not available.

Basic specification

I/O points	1,024 points (64 words) / remote master 4 remote master / CPU (basic base) Total 4,095 points (256 words)
Remote I/O address	Remote master 1 : Y10000- / WY1000- Remote master 2 : Y20000- / WY2000- Remote master 3 : Y30000- / WY3000- Remote master 4 : Y40000- / WY4000-
Supported CPU	CPU516/548

MEMO

Chapter 7 Programming

7.1 Memory Capacity

The specifications for the user program in EH-150 are given in Table 7.1.

Table 7.1 User program specifications

No.	Item		CPU104(A)	CPU208(A)	CPU308(A)	CPU316(A) CPU516	CPU448(A) CPU548
1	Program capacity		3.5 k steps	7.6 k steps	7.6 k steps	15.7 k steps	48.5 k steps
2	Command size		32 bits/1 step				
3	Memory specification	SRAM	Backup is possible by installing the battery				
		FLASH	Backup is possible by using FLASH memory				
4	Programming language		H-series ladder/command language				
5	Program creation		Created with H-series programming device				
6	Program modification	During STOP	Can be done as desired by using the programming device.				
		During RUN	Can be done by using the modification during RUN operation (except control commands). Control commands can be changed with special operations. *1 (When a change is made during RUN, control operation stops while the program is being modified.).				
7	Program protection		The program cannot be changed unless with a WRITE occupancy. (Occupancy is automatically controlled by the programming device).				
8	Password		A password can be set using the programming device (the program cannot be displayed when setting the password. The program can be downloaded to the programming device).				
9	Check function		A sum check function for the program is always running. An address check with the I/O assignment table is done at the time RUN begins.				
10	Program name		The program name is set using the programming device and is stored together with the program.				

*1: Refer to the peripheral device manual for details.

Note:

- Comment data that has been created with peripheral device is not stored in the CPU.
- Save the user program to a separate floppy disk or other media, in case a mishap should occur.

7.2 Programming Method

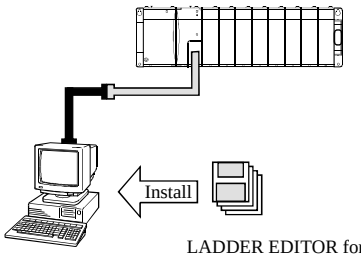
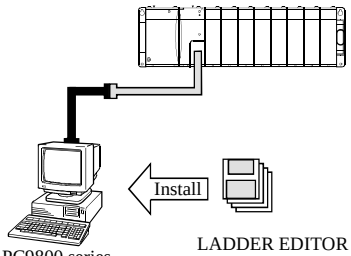
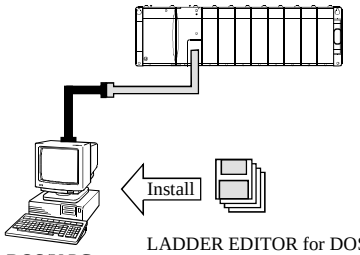
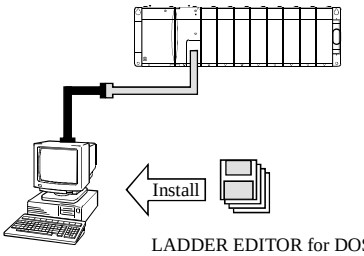
The following methods are allowed for creating the user program.

Table 7.2 Programming methods

No.	Programming device used	Operation concept	Remarks
1	Personal computer software (LADDER EDITOR, etc.)	[For off-line / on-line operation] The I/O assignment table created, program to be created is input, and transfered to the EH-150 CPU.	• When each EH-150 module is loaded to the base unit, the I/O assignment information can be read. • Initialize the EH-150 CPU when using it for the first time after it is unpacked or when a battery error occurs.
2	Dedicated programming console (GPCL01H, etc.)	[For direct operation] As the programs are input one by one, the program input to the CPU is directly written to the CPU, each time.	
3	Portable graphic programmer (PGM-GPH) (Do not use an option box.)	As the programs are input one by one, the program input to the CPU is directly written to the CPU, each time.	
4	Ccommand language programmer (PGM-CHH)		

The system configuration and procedure for creating a user program when using personal computer software are indicated below. It is necessary to take note that cables, etc. differ depending on the personal computer and the software used.

Table 7.3 System configuration when using a personal computer

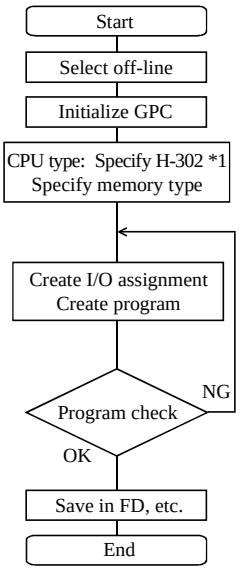
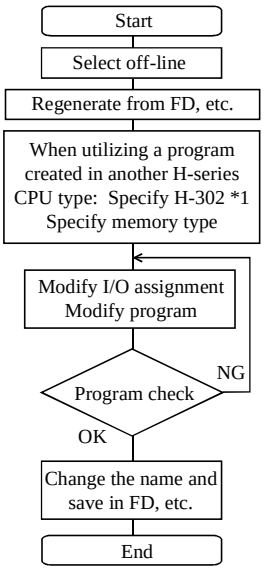
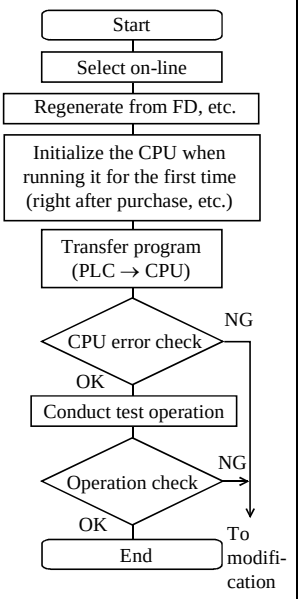
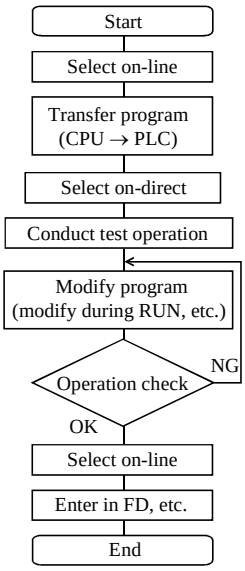
<table>
<tr><th>No.</th><th colspan="2">Personal computer software used</th><th>DOS/V PC</th><th>PC9800 series personal computer</th></tr>
<tr><td>1</td><td colspan="2">LADDER EDITOR
(Windows® version)</td><td> 
</td><td> 
</td></tr>
<tr><td></td><td colspan="2">CPU setting</td><td colspan="2">Specify H-302. Specify EH-150 after version 2.0.*2</td></tr>
<tr><td></td><td rowspan="5">Memory setting</td><td>CPU104(A)</td><td colspan="2">Specify RAM-04H (4 k memory).</td></tr>
<tr><td></td><td>CPU208(A)</td><td colspan="2">Specify RAM-08H (8 k memory).</td></tr>
<tr><td></td><td>CPU308(A)</td><td colspan="2">Specify RAM-08H (8 k memory).</td></tr>
<tr><td></td><td>CPU316(A)/516</td><td colspan="2">Specify RAM-16H (16 k memory).</td></tr>
<tr><td></td><td>CPU448(A)/548</td><td colspan="2">Specify RAM-48H (48 k memory).*2</td></tr>
<tr><td></td><td colspan="2">Cable (EH-150 side)</td><td>EH-RS05</td><td>EH-RS05</td></tr>
<tr><td></td><td colspan="2">Cable
(personal computer side)</td><td>WVCB02H</td><td>WPCB02H</td></tr>
<tr><td></td><td colspan="2">Mode setting switch *1
(Serial port 1 setting)</td><td>SW 1 2 3 4 5 6 7 8 Toggle
Status OFF — ON OFF ON — OFF OFF —</td><td>SW 1 2 3 4 5 6 7 8 Toggle
Status OFF — ON OFF ON — OFF OFF —</td></tr>
<tr><td></td><td colspan="2">Mode setting switch *1
(Serial port 2 setting)</td><td>SW 1 2 3 4 5 6 7 8 Toggle
Status OFF — — — — OFF OFF OFF ON</td><td>SW 1 2 3 4 5 6 7 8 Toggle
Status OFF — — — — OFF OFF OFF ON</td></tr>
<tr><td>2</td><td colspan="2">LADDER EDITOR
(DOS version)</td><td> 
</td><td> 
</td></tr>
<tr><td></td><td colspan="2">CPU setting</td><td colspan="2">Specify H-302.</td></tr>
<tr><td></td><td rowspan="5">Memory setting</td><td>CPU104(A)</td><td colspan="2">Specify RAM-04H (4 k memory).</td></tr>
<tr><td></td><td>CPU208(A)</td><td colspan="2">Specify RAM-08H (8 k memory).</td></tr>
<tr><td></td><td>CPU308(A)</td><td colspan="2">Specify RAM-08H (8 k memory).</td></tr>
<tr><td></td><td>CPU316(A)/516</td><td colspan="2">Specify RAM-16H (16 k memory).</td></tr>
<tr><td></td><td>CPU448(A)/548</td><td colspan="2">Specify RAM-48H (48 k memory).</td></tr>
<tr><td></td><td colspan="2">Cable (EH-150 side)</td><td rowspan="2">EH-VCB02</td><td>EH-RS05</td></tr>
<tr><td></td><td colspan="2">Cable
(personal computer side)</td><td>PCCB02H</td></tr>
<tr><td></td><td colspan="2">Mode setting switch *1
(Serial port 1 setting)</td><td>SW 1 2 3 4 5 6 7 8 Toggle
Status OFF — ON ON ON — OFF OFF —</td><td>SW 1 2 3 4 5 6 7 8 Toggle
Status OFF — ON ON ON — OFF OFF —</td></tr>
<tr><td></td><td colspan="2">Mode setting switch *1
(Serial port 2 setting)</td><td>SW 1 2 3 4 5 6 7 8 Toggle
Status OFF — — — — OFF OFF OFF OFF</td><td>SW 1 2 3 4 5 6 7 8 Toggle
Status OFF — — — — OFF OFF OFF OFF</td></tr>
</table>

*1 Restart the power supply after the mode setting switch has been adjusted. The settings that are made while the power is on will not be effective until the power is restarted.

*2 When using the EH-CPU448(A)/548, please use the LADDER EDITOR for Windows® Ver. 2.13 and later. When using before Ver 2.12, please specify H-302. (Memory setting is RAM-48H)

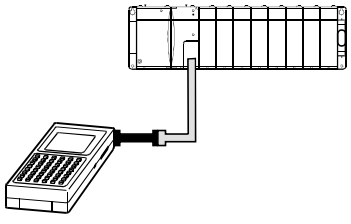
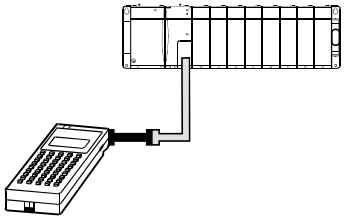
Note: Refer to the manual that comes with each software on how to install the software (LADDER EDITOR).

Table 7.4 List of procedures for creating a program

Item	Create new program	Modify	Test operation, adjustment	
	Off-line	Off-line	On-line	On-direct
Operation procedure summary				
Circumstance	When creating a new program	When modifying a program	When transferring a program to the CPU for the first time	When modifying a program during test operation
Point	A program can be created without having the actual EH-150.	When using a program that was used in another H-series, specify H-302 as the CPU type.	When performing CPU error check, make sure the I/O assignment matches the loaded module. (The match can be forced using the loading read function).	Match the personal computer and the contents of CPU memory to enter the on-direct mode. Afterwards, the modification will be reflected in both the computer memory and CPU memory.

*1: For LADDER EDITOR for DOS version and LADDER EDITOR 1.×× for Windows ®.
Specify EH-150 after LADDER EDITOR 2.×× for Windows ®. When you will set the 48k to the Memory setting, please use after LADDER EDITOR 2.13 for Windows®.

Table 7.5 System configuration when using the programmer

Item		Portable graphic programmer (PGM-GPH)	Command language programmer (PGM-CHH)
System configuration			
	Cable (EH-150 side)	EH-RS05	EH-RS05
	Cable (programmer side)	PGCB02H	PGCB02H

Note: 1. Do not use the option box (model: PGMIF1H) for the portable graphic programmer. Because of the large current consumption, the EH-150 system may go down.
Also, use a cable of 2 m (6.56 ft.) in length (model: PGCB02H) on the programmer side to connect a programmer. If a 5 m (16.4 ft.) cable (model: PGCB05H) is used, the programmer operation may become unstable.

2. The programmer can only be connected to serial port 2. For more information on setting modes, see Table 10.6.

With the EH-150, the user program is controlled in ladder units and as shown with Figure 7.1, 1 ladder can describe 9 contacts (a-type contact or b-type contact) and 7 coils.

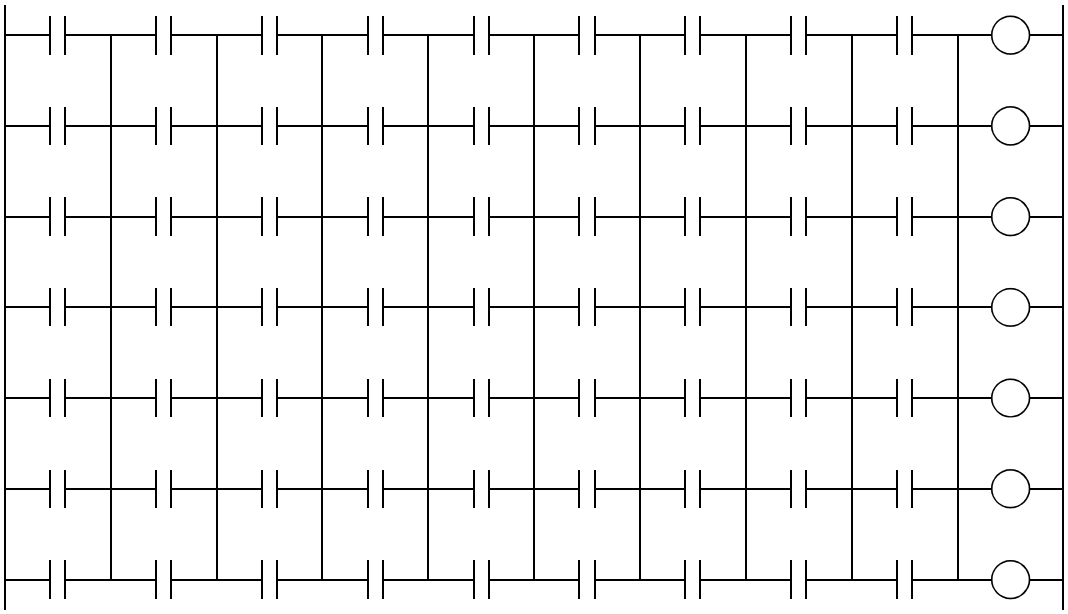


Figure 7.1 Max. size of one ladder

Or, one relational box can be described using the width of 3 contacts. The relational box can be thought of as an a-type contact that turns on when the conditions in the box are satisfied (Figure 7.2).

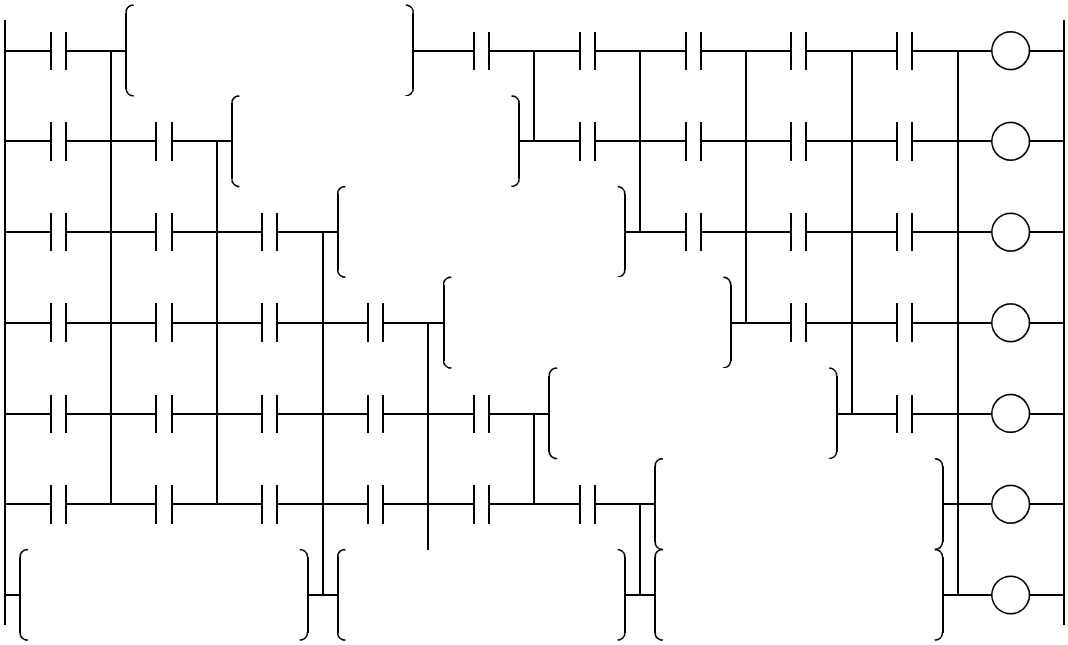


Figure 7.2 Using comparison box

In addition, if return symbols are used, a ladder containing up to 57 contacts and one coil can be input within seven lines. (Figure 7.3)

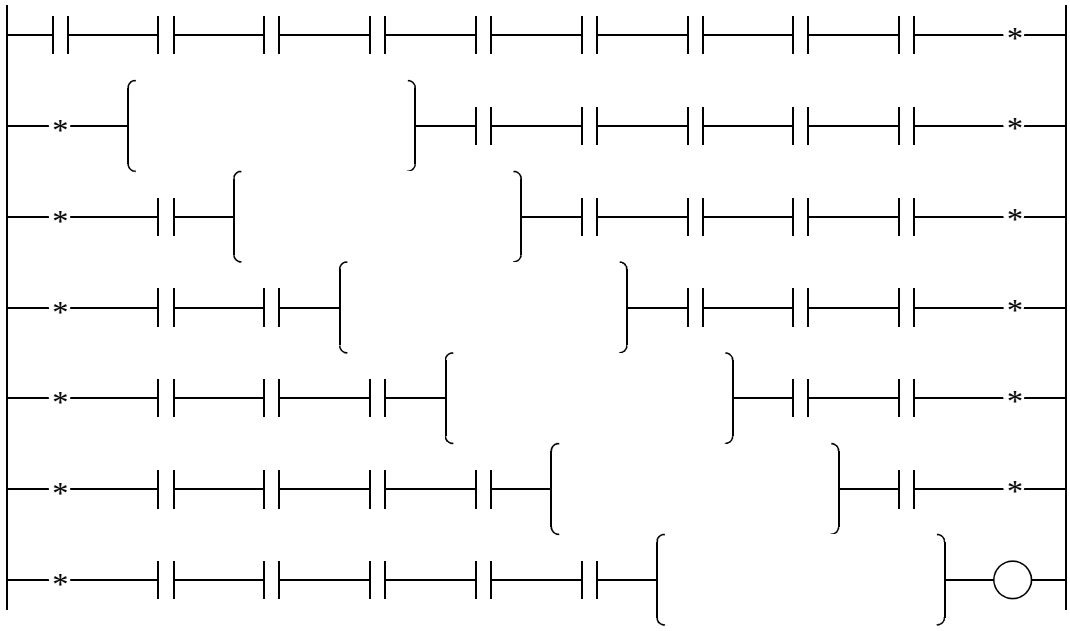


Figure 7.3 Example of using return symbols

A processing box can be placed in the coil position. Processing commands, application commands, control commands, transfer command and FUN commands can be described in a processing box. A maximum of 19 commands can be described in one processing box. The processing box is executed when the condition in the contact section that is connected directly before it, is satisfied and not executed if the condition is unsatisfied.
Refer to “Chapter 5 Command Specifications” for details on each command.

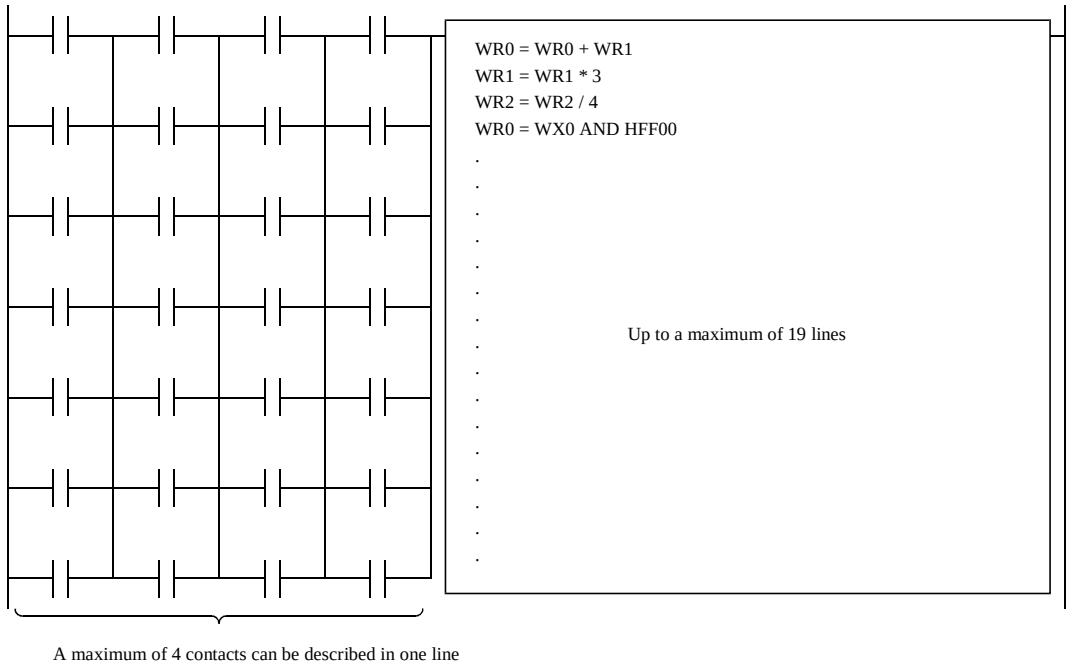


Figure 7.4 Using a processing box

Note: For the LADDER EDITOR for Windows®, the processing box can be displayed in 1 contact point width, so a ladder of 9 contact and 1 processing box can be input.
For details, refer to the user's manual for the LADDER EDITOR for Windows®.

MEMO

Chapter 8 PLC Operation

The EH-150 can switch its operation status and stop status through various types of operations. This feature is shown in Figure 8.1.

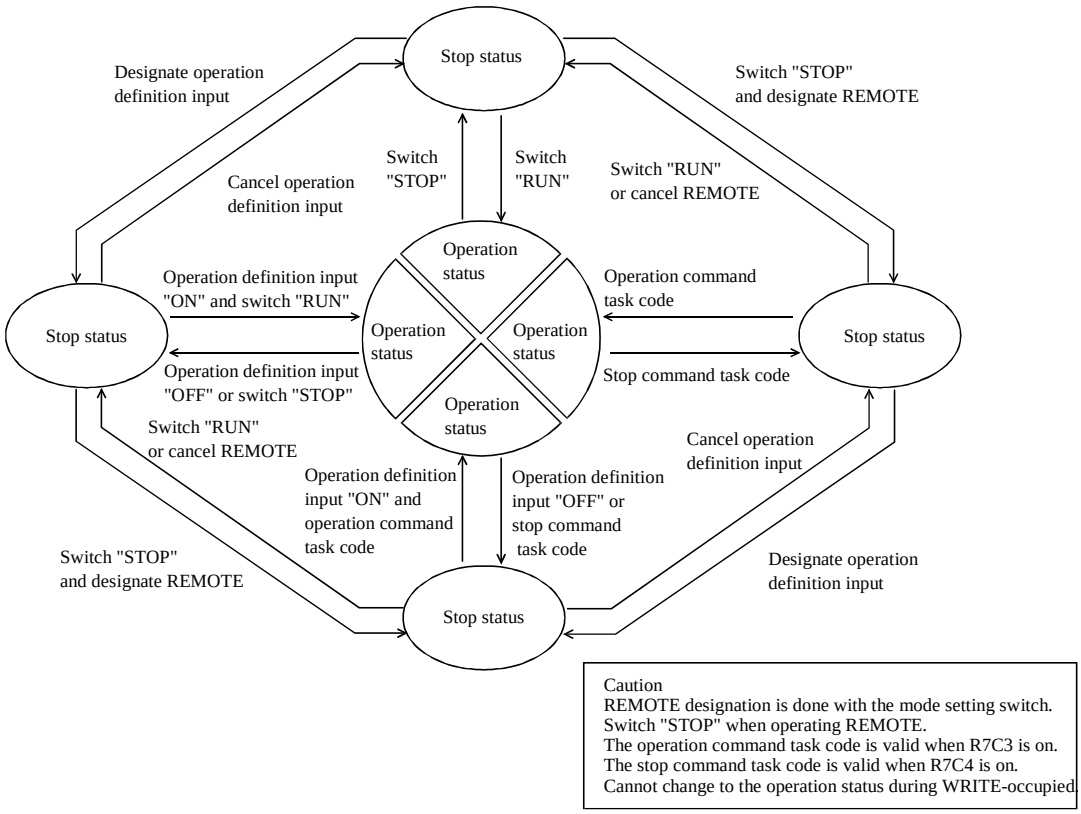


Figure 8.1 Diagram of status transition between operation and stop

The EH-150 can be operated or stopped under the conditions as shown in Figure 8.1. If an error is detected during operation or stop, output is shut off, an error is displayed and the EH-150 stops. Error includes serious failure, medium failure, minor failure and warning. The operation status for each failure is shown in Table 8.1.

Table 8.1 Description of each failure and operation status

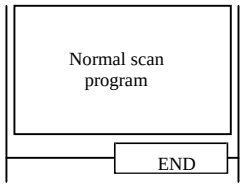
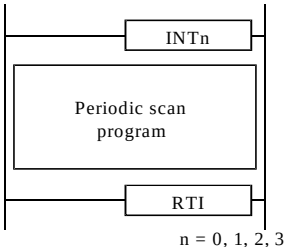
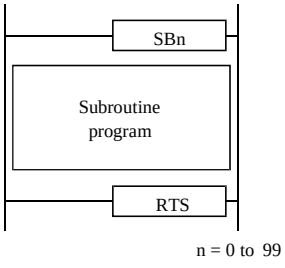
Classification	Description	Run/Stop
Serious failure	This indicates there is a serious, unrecoverable error, such as power failure, microcomputer error, system ROM error, system RAM error, system bus error, etc.	Stop
Medium failure	This indicates there is an error such as data memory abnormal, system program abnormal, user memory abnormal, user memory size error, grammatical assemble error, etc., which will cause a malfunction if operation is continued.	Stop
Minor failure	These are errors such as I/O information verify error, remote abnormal, overload error, excessively assigned I/O points. The operation can continue when there is a continue operation setting in the user program.	Stop (continued operation is possible if specified)
Warning	These are abnormal such as a transfer error, where it is possible to continue the operation.	Operation continues

Note
If CPU type is 448(A)/516/548, "RUN continue" can not be specified at scan time error for cyclic scan. Make your program of periodical scan carefully so that scan time error is not detected.

8.1 RUN Start

When the EH-150 switches to operation status, the user program is executed in sequence from the head. The user program consists of a normal scan program and periodic scan program. In addition to these programs, there is a subroutine area defined as a subroutine.

Table 8.2 Program classification

No.	Program classification	Description	Expression
1	Normal scan program	This is the program that is normally executed. When the program has been executed to the END command, execution starts again from the head. Overload error monitoring is performed according to the overload check time set by the user. Monitoring is from the top of the program until the END command. If it is designated to continue during overload, the operation continues.	
2	Cyclic scan program	This program is executed periodically at intervals of 5 ms, 10 ms, 20 ms, or 40 ms *. Each execution cycle time becomes the overload error monitoring time. If it is specified to continue during overload, the operation is terminated in the middle. If CPU type is 448(A)/516/548, "RUN continue" can not be specified at scan time error for cyclic scan. Make your program of periodical scan carefully so that scan time error is not detected.	Described in the area after the END command. 
3	Subroutine	This is a program called out by the CALL command.	Described in the area after the END command 

* Cyclic scan time depends on CPU types

	CPU104/208/308/316	CPU***A/448/516/548
INT 0	10 ms	5 ms
INT 1	20 ms	10 ms
INT 2	40 ms	20 ms
INT 3	-	40 ms

8.1.1 Scan operation

Each program is executed in the order of the priority shown in Figure 8.2. Each program is executed while monitoring the execution time of each program area. If the monitored time exceeds the prescribed time, this causes an overload error and operation stops. When continued operation has been specified, operation continues.
The timing for scan execution is shown in Figure 8.2. System processing is performed at set periods (every 5 ms), followed by communication system processing. The maximum execution time of communication system processing *1 equals the duration of time until the next periodic system processing is started. If the communication system processing ends before the maximum execution time is up, execution of scan processing is started upon completion of the communication system processing. When the next periodic processing is performed, scanning is performed until the next periodic processing is performed.
*1: Communication system processing is executed every 10 ms.

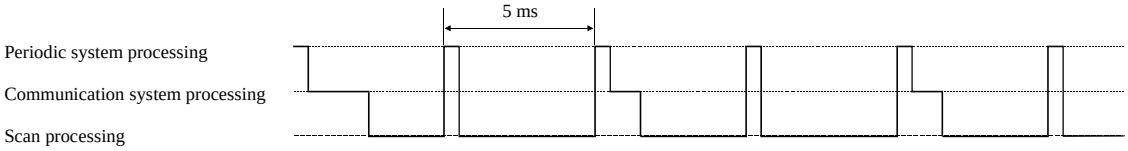


Figure 8.2 Relationship between system processing and scanning

As shown in Figure 8.3, scan processing is done while periodic scanning is performed. Periodic scanning is processed at the point when switching to normal scan. Periodic scans are performed at intervals of every 5 ms, 10 ms, 20 ms, or 40 ms. (5 ms periodic scan is supported only for the EH-CPU***A/448/516/548.) In terms of priority of execution, 5 ms scans have the highest priority. Use the refresh command when you wish to perform data processing for the external I/O (X,Y) during the periodic scan.
Update of timer progress value is performed as part of system processing. (For the EH-CPU***A/448/516/548, it is performed when each timer command is executed.)

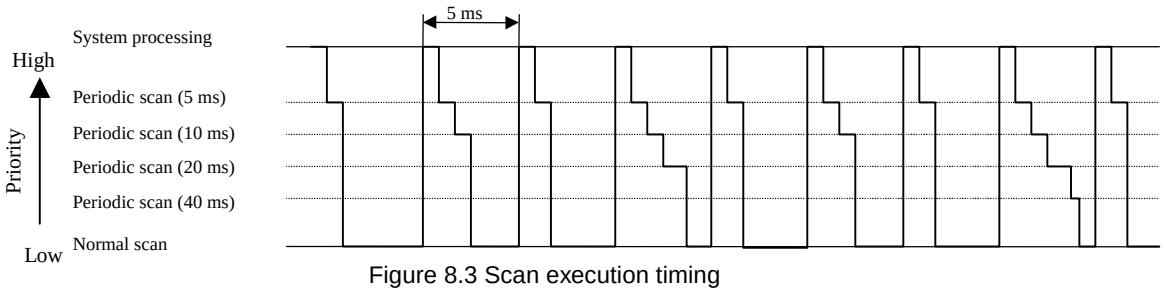


Figure 8.3 Scan execution timing

8.1.2 Setting System Processing Time (EH-CPU***A/448/516/548)

System processing time can be assigned from 1 to 8 ms out of 10 ms for EH-CPU***A/448/516/548. It is thus possible to select the optimal system processing time according to the purpose of control.

(1) Operation and application

Figure 8.4 shows the scan operation when the system processing time is set to 3 ms. In this figure, the system operation time corresponds to "periodical system processing" + "communication system processing,". And you can set up time to assign this processing as a parameter.

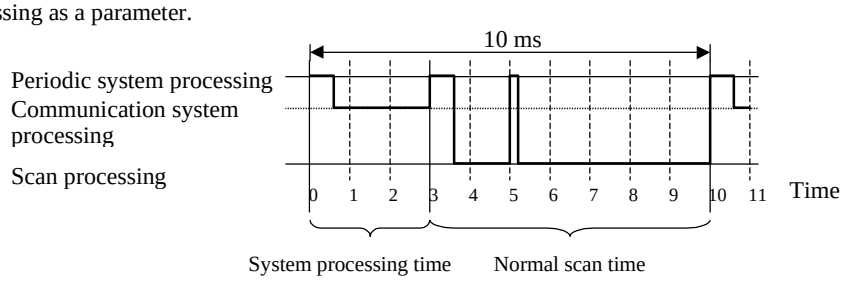


Figure 8.4 Scan operation when the system processing time is set to 3 ms

Because the scan processing time for executing user programs can be made longer, the operation processing efficiency per unit time can be improved.

(2) Setup method

Set the system processing time in special internal output WRF038.

Bit:	15 ... 8	7 ... 0
WRF038	User setting time	Current system processing time

Figure 8.5 Special internal output for setting the system processing time

Enter a value from 1 to 8 in 1-digit BCD format representing the user setting time. If a value outside this range is selected, the system sets the value which can be used with the system. Furthermore, the processing time of the system currently in operation, which is set in the lower eight bits of the current system processing time field, can be checked by monitoring WRF038 with a peripheral unit. This setting range changes as shown in the following table due to the existence of 5ms periodical scan (INT0).

Table 8.3 The correspondence of the user setting value and the actual system processing time

Setting value(*1)	H0	H0	H0	H0	H0	H0	H0	H0	H0	H0	Remark
INT0(5ms periodical scan) Not exist	H2	H1	H2	H3	H4	H5	H6	H7	H8	H8	The part of __ shows the value changed forcibly with the system.
INT0(5ms periodical scan) Exist	H2	H1	H2	H3	H2	H2	H2	H2	H2	H2	

*1: Setting value is the lower digit BCD format of an upper 8 bits of WRF038. The value of an upper digit BCD format is ignored.

Ex.1): In order to set the system processing time to 3 ms, enter H0300 in WRF038.

Ex.2): If H10 is set as the user setting time, the value of an upper digit BCD format is ignored. Therefore, a system is judged H0 to be set up, and system set to 3 ms the system operation time. It is indicated with H1002 when WRF038 is seen at this time.

This setting becomes valid immediately after setting.

The existence of 5 ms periodical scan (INT0) is confirmed with the system at the time of the RUN start. System processing time is changed forcibly when INT0 is contained in the program and user setting time is beyond H4. (Table 8.3 reference.)

(3) Precautions

Please make a program referring to "8.1.6 (4) Cautions when using 5 ms periodical scan " when you use the 5 ms periodical scan (INT0), which is supported only by the EH-CPU***A/448/516/548.

8.1.3 Setting System Processing Time (EH-CPU308/316)

> NOTE:
> This function is available only for EH-CPU308/316.
> EH-CPU***A/448/516/548 is not applied.

The system processing time of the EH-CPU308/316* can be fixed at 5 ms. By fixing the system processing time, the normal scan time will not fluctuate and it is executed for 5 ms in 5 ms cycles.
* The EH-CPU308/316 of ROM Version 03 or earlier do not support this function.

(1) Operation and application

Figure 8.6 shows the scan operation when the system processing time is fixed. In this figure, the system operation time corresponds to "periodical system processing" + "communication system processing," thus it is possible to set this parameter.
Conventionally, the operation shifts to normal scan processing when the system processing is finished. If, however, the system processing time is fixed, the normal scan does not start until 5 ms have elapses, even if the system processing is finished.

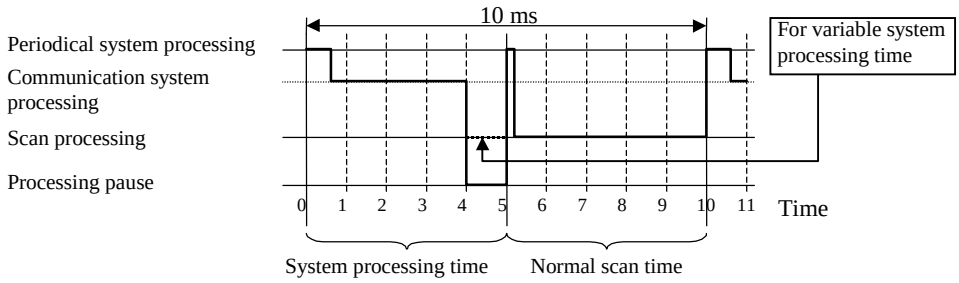


Figure 8.6 System processing time (fixed at 5 ms)

The scan processing time for executing user programs can be fixed. Use this function if it is not desirable to allow the normal scan time to fluctuate.

(2) Setup method

Set the system processing time in special internal output WRF038.

Bit:	15 ... 8	7 ... 0
WRF038	User setting time	Current operation status

Figure 8.7 Special internal output for setting the system processing time

Enter H*5 as the user setting time. When the setting is completed, H05 is set in the current operation status field. If a value other than H*5 is set in the user setting time field, the setting value is stored in the user setting time field but H00 is stored in the current operation status field, and the system processing time becomes variable as usual.
Example: In order to fix the system processing time, enter H0500 in WRF038. If WRF038 is monitored at this point, it can be observed that H0505 is set.
If a value other than H*5, e.g. H0300, is set as the system processing time, the system processing time becomes variable and the value that can be monitored in WRF038 is H0300.
This setting becomes valid immediately after setting.

8.1.4 Normal Scan

(1) Definition and operation

The normal scan refers to the calculations and execution of the ladder/instruction language program (or the programs written in Pro-H) until the END scan processing caused by the END instruction (excluding interrupt programs). The time required for one scan, from the beginning of a normal scan program to the END scan processing, is called the normal scan time.

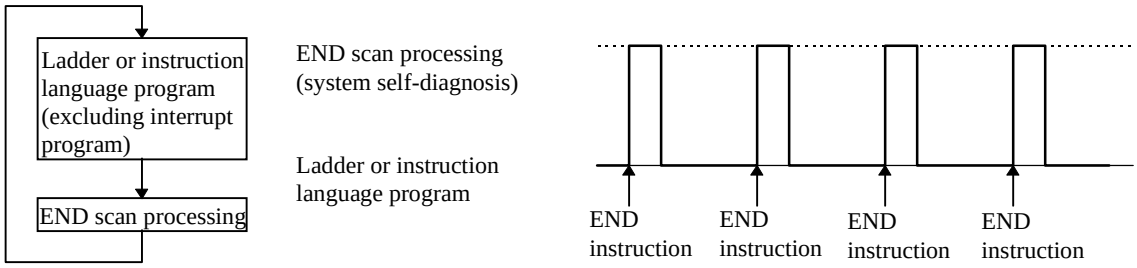


Figure 8.8 Operation of normal scan

(2) Causes of congestion errors at normal scan

Congestion errors may occur at normal scan because of the following two possible reasons. In particular when using a periodical scan program together, care must be taken to create the program in such a way that the total scan time does not exceed the congestion check time.

(a) When only a normal scan program is used

The scan time exceeded the congestion check time because the time required for one scan was too long.

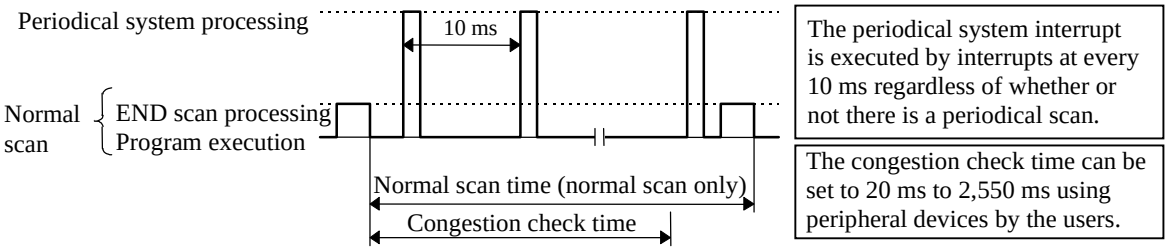


Figure 8.9 Congestion error at normal scan (a)

(b) When both a normal scan program and a periodical scan program are used

The congestion check time was exceeded because the periodical scan program was executed and the normal scan time became longer. (Figure 8.10 shows an example of operation by CPUs other than the EH-CPU104A/208A/308A/316A/448.)

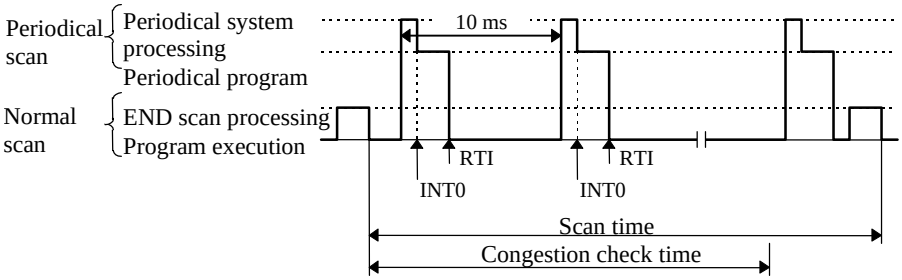


Figure 8.10 Congestion error at normal scan (b)

(3) Continuation of operation after a congestion error occurred

When the special internal output R7C0, which specifies whether the operation should continue after a congestion error occurred, is turned on, the normal scan executes the scan until the end regardless of the congestion check time, and after executing the END scan processing, executes the normal scan from the beginning again.

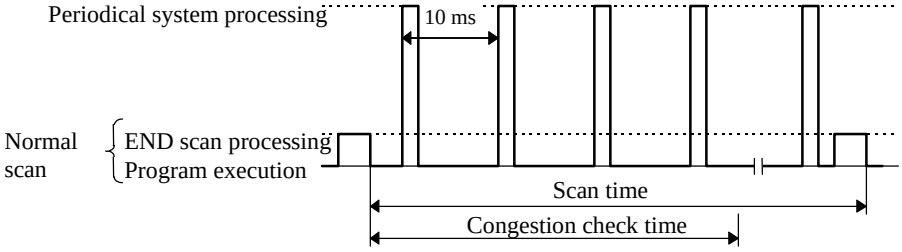


Figure 8.11 Operation when operation continuation at congestion error is set

However, note that this setting does not stop the execution of the scan when a congestion error occurred even when an infinite loop is formed within the normal scan by the JMP instruction.

8.1.5 Periodical Scan (In case of the EH-CPU104/208/308/316)

(1) Definition and operation

This scan executes operations of interrupt programs (periodical scan programs) while the CPU is operating with a cycle time specified by the user (10 ms, 20 ms or 40 ms). For the EH-CPU***A/448/516/548, 5 ms periodical scan has newly been added. The details about the periodical scan of the EH-CPU***A/448/516/548 are explained in Section 8.1.6. This section explains about the EH-CPU104/208/308/316.

Enter the periodical scan program to be executed between instructions INT0 and RT1 if it should be started up with a 10 ms cycle time, and between INT1 and RT1 if it should be started up with a 20 ms cycle time.

The periodical system processing is executed every 10 ms regardless of whether or not there is a periodical scan program.

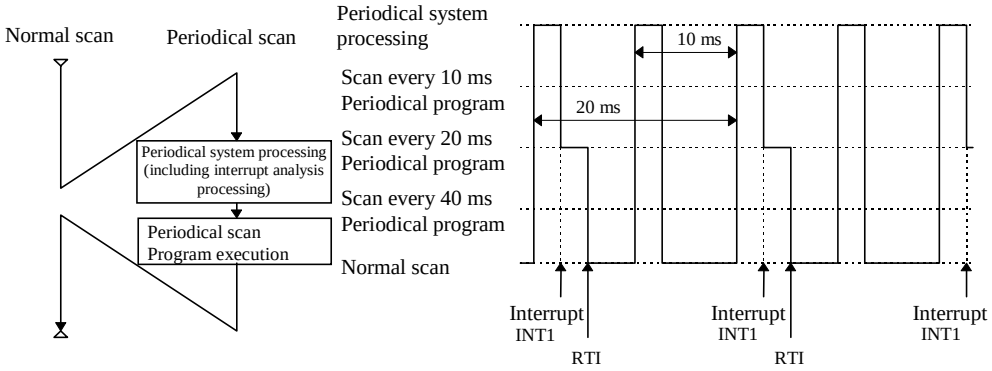


Figure 8.12 Operation of periodical scan (in case of INT1)

(2) Causes of congestion errors at periodical scan

If there are periodical scans at every 10 ms as well as scans at every 20 ms or 40 ms, a congestion error occurs and the scan is stopped if the periodical scan at 10 ms is started up again before all the periodical scans are completed (i.e., the periodical system processing at INT0 to INT2 does not end within 10 ms).

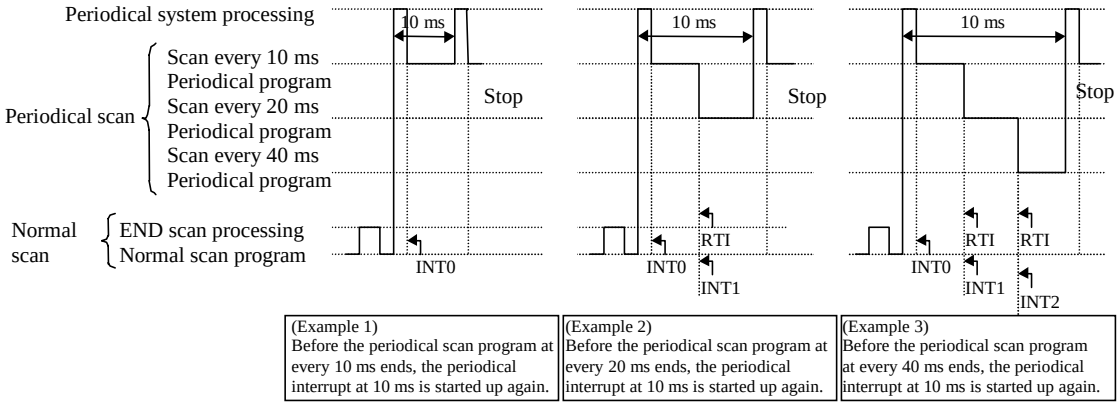


Figure 8.13 Congestion error at periodical scan (10 ms)

Similarly, when executing with a periodical scan at every 20 ms or with a combination of periodical scans at every 20 ms and 40 ms, a congestion error occurs if the periodical scan at 20 ms is started up again before all the periodical scans are completed (i.e., the periodical system processing at INT1 to INT2 does not end within 20 ms). Finally, when using a periodical scan at every 40 ms, a congestion error occurs if the periodical scan at 40 ms is started up again before all the periodical scans are completed (i.e., the periodical system processing at INT2 does not end within 40 ms).

(3) Continuation of operation after a congestion error

If a congestion error occurs when the special internal output bit R7C1, which specifies whether the operation should continue after a congestion error, is turned on, the execution of the periodical scan is stopped and the periodical scan is executed from the beginning again. If the operation continuation specification for the normal scan is Off when this happens, the scan stops as a congestion error at a normal scan. If the operation continuation specification for the normal scan is On, only the periodical scan continues to be executed in the event of a periodical congestion error. Care must be taken because the normal scan is not executed under this condition.

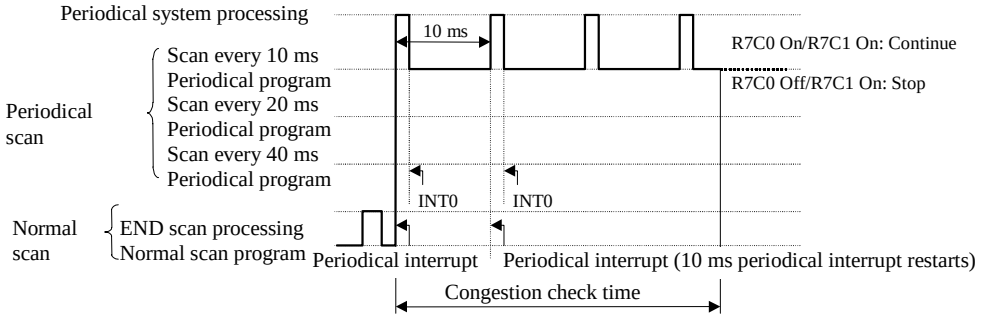


Figure 8.14 Operation when operation continuation at congestion error is set

8.1.6 Periodical Scan (In case of EH-CPU***A/448/516/548)

(1) Definition and operation

In addition to the conventional periodical scan, a “5 ms periodical scan” is now available in the EH-CPU***A/448/516/548. The basic operations, such as normal operation and operations at congestion errors, are the same as the operations available at the periodic scan described in Section 8.1.5.

Table 8.4 Interrupt labels and cycles time

No.	Interrupt label	Startup cycle	
		EH-CPU***A/516/548	EH-CPU104/208/ 308/316
1	INT0	Every 5 ms	Every 10 ms
2	INT1	Every 10 ms	Every 20 ms
3	INT2	Every 20 ms	Every 40 ms
4	INT3	Every 40 ms	-

(2) Causes of congestion errors at periodical scan

In the same way as in CPUs other than the EH-CPU104A/208A308A/316A/448, a congestion error occurs when the execution time of the periodic scan exceeds the minimum cycle (time) of periodical program steps used in the program.
Ex. 1) If all the interrupt labels, INT0 to INT3, are used
Since the minimum cycle of the periodical scans being used is 5 ms, a congestion error occurs if the total scan time of each periodical scan exceeds 3 ms. Refer to (4) for INT0 specially.
Ex. 2) When INT2 and INT3 are used
Since the minimum cycle of the periodical scans being used is 20 ms, a congestion error occurs if the total scan time of each periodical scan exceeds 20 ms.

(3) Continuation of operation after the occurrence of a congestion error

The CPU except for EH-CPU448(A)/516/548 can continue operation even if a congestion error occurs in periodical scan when special internal output “R7C1” is turned on. But, even if this setting is done, operation isn't continued with EH-CPU448(A)/516/548. Please make a periodical scan program so that a congestion error may not occur in periodical scanning.
And, when the time to assign it to the system processing is lengthened, the time that is assigned to periodical scan becomes short. Therefore carry out the change of the system processing time after you do enough confirmation.
Note) When the congestion error occurs in the CPU except EH-CPU448, the periodical scan program aborts on the way.
When the next period scan begins, it executes again from the beginning of the period scan program.
Therefore, please modify the period scan program to finish in period scan, because the period scan program of the congestion error occurrence since then isn't executed

(4) Cautions when using 5 ms periodical scan

The congestion check time of 5 ms periodical scan (INT0) is the time when system processing time, the completion processing time of periodical scanning, trance/receiving processing time of the communication, and so on were deducted from 5 ms. The allowable time when 5 ms periodical scan (INT0) can be carried out in the table 8.5 is shown. Please make a program to be in time when the value that the processing time of each command was summed up shows it in the table.

Table 8.5 Standard of 5 ms periodical scan executive time

Setting value	System processing time	Equation	INT0 Allowable time
H0	H2 (0 to 3 ms)	(5 ms - 3 ms - 0.5 ms) × 60%	0.9 ms
H1	H1 (0 to 2 ms)	(5 ms – 2 ms - 0.5 ms) × 60%	1.5 ms
H2	H2 (0 to 3 ms)	(5 ms – 3 ms - 0.5 ms) × 60%	0.9 ms
H3	H3 (0 to 4 ms)	(5 ms – 4 ms - 0.5 ms) × 60%	0.3 ms
H4 to H9	H2 (0 to 3 ms)	(5 ms – 3 ms - 0.5 ms) × 60%	0.9 ms

8.2 Online Change in RUN

The user program can be modified during operation while retaining the output status as is. This is called the "online change in RUN" function. Modifying the user program requires programming software specifically for this or a programmer. Refer to the individual manuals on how to do this.
Online change in RUN cannot be done in the following situations. Perform this operation after completing the conditions.

Table 8.4 Conditions for performing online change in RUN

No	Conditions under which online change in RUN cannot be done	Specific situation	How to satisfy the conditions
1	When READ-occupying	Other programming device is connected.	Change other programming devices to off-line.
2		When a personal computer or panel, etc. is connected and monitoring is being done.	Change the personal computer or panel to off-line. (When monitoring, it is convenient to use the occupancy unnecessary task code.)
3	END command is not executed.	A program that runs in an endless loop is being executed.	Correct the program so that it does not run in an endless loop.
4	Trying to modify a program that contains control commands.	Doing an online change in RUN for a ladder containing a control command may cause operation to stop depending on the type of the program modification error.	An explanation of how to do an online change in RUN for a ladder that contains a control command is given in the programming software manual.
5	A password has been set.	A program protected by a password cannot be modified.	Perform the revision after having the system supervisor remove the password.

(When the CPU is stopped, the modification is executed without displaying a message confirming online change in RUN.)

8.2.1 Cautionary Items for Changing Programs in RUN

The system operation of EH-CPU***A/448/516/548 is slightly different from EH-CPU104/208/308/316. The following describes cautionary items for changing programs in RUN when using EH-CPU***A/448/516/548.

(1) Updating the timer elapsed value

The timer elapsed value of EH-CPU104A/208A/308A/316A/448 is calculated by comparing with the previous elapsed value when the timer coil command is executed. The execution of a user program halts temporarily when it is changed in RUN. Therefore, the timer elapsed value will not be updated when the CPU is in the HALT state.

> Reference: The timer elapsed values of EH-CPU104/208/308/316 are updated periodically. Therefore, these timer elapsed values are updated even in the HALT state.

(2) Transferring changed programs to the FLASH memory

EH-CPU***A/448/516/548 loads the programs that have been changed in RUN to the FLASH memory after the change in RUN display on the peripheral unit goes off. If the CPU power is turned off while a changed program is being transferred to the FLASH memory, that program may be damaged.
To prevent this, the special internal output R7EF which indicates that programs are being transferred to the FLASH memory (turns on while programs are being transferred to the FLASH memory) has been added. Before turning off the CPU power, be sure that this special internal output R7EF is off, or wait for about two minutes after the change in RUN operation.
The EH-150 operation when the user program is changed in RUN is shown below.

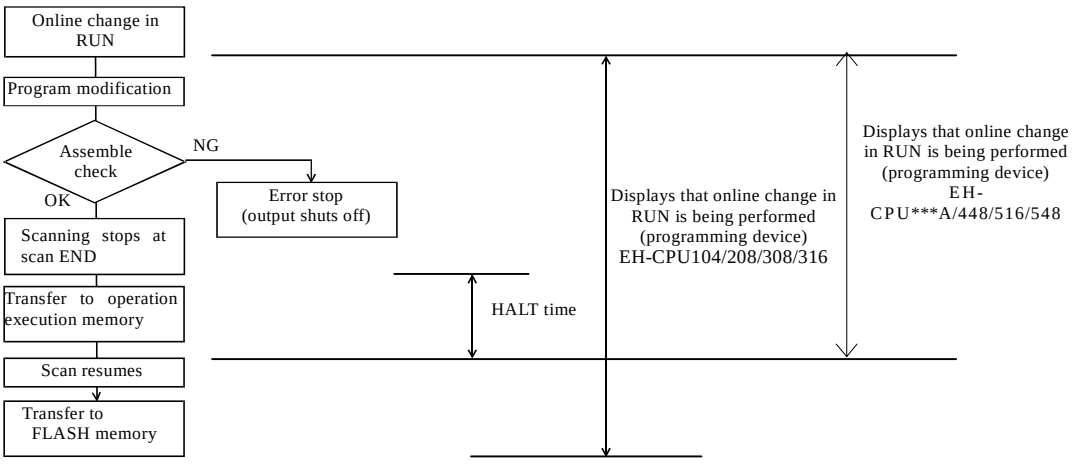


Figure 8.15 Internal processing for an online change in RUN

8.2.2 HALT time

When performing online change in RUN, the program to be written to the CPU is checked if there are no errors, then the CPU is halted temporarily (RUN → HALT).
The program of the modified area is written to the CPU while it is halted, and the CPU is set to operate (HALT → RUN) again.
At this time, the approximate time the CPU is halted will is obtained in the equation below (it is not necessarily the maximum value).

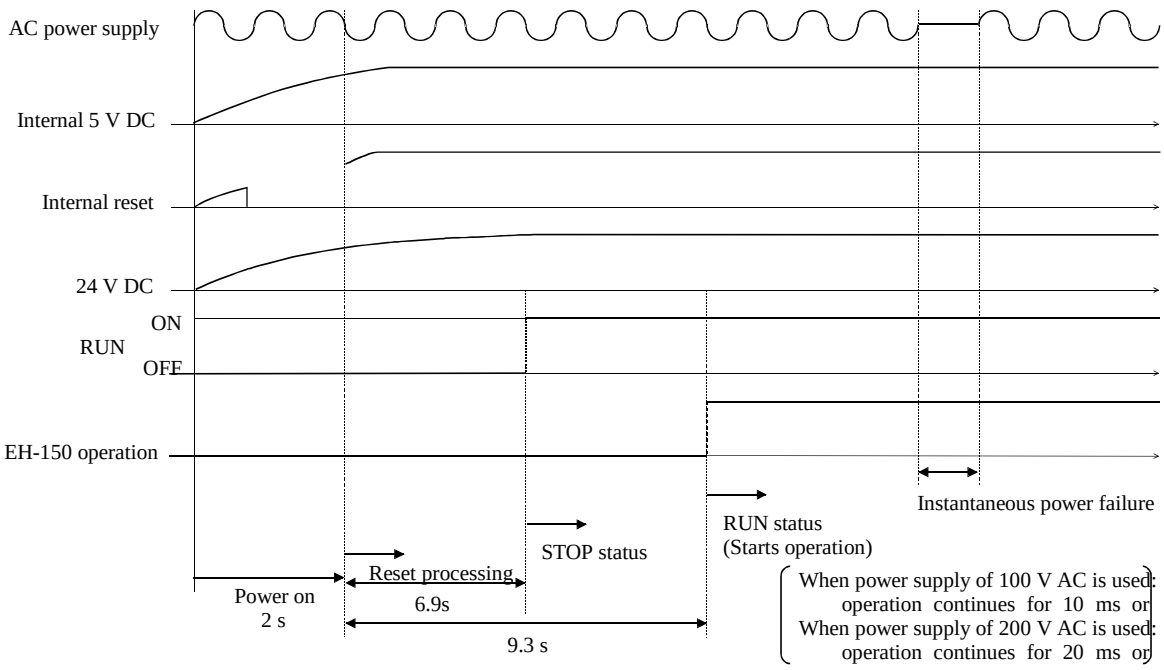
HALT time (ms) = 45 × Program capacity (ks) + 20 (EH-CPU104//208)
= 38 × Program capacity (ks) + 10 (EH-CPU308/316)
= 18 × Program capacity (ks) + 15 (EH-CPU104A-316A)
= 29 × Program capacity (ks) + 10 (EH-CPU516)
= 29 × Program capacity (ks) + 60 (EH-CPU448(A)/548)

An example of a calculation of the HALT time using the above equation is shown below.

Program size (k step)	HALT time (ms)				
	CPU104/208	CPU308/316	CPU104A-316A	CPU516	CPU448(A)/548
4	200	162	87	126	176
8	380	314	159	242	292
16	-	618	303	474	524
48	-	-	-	-	1,452

8.3 Instantaneous Power Failure

What happens when the power supply to the EH-150 shuts off, is shown below.



(1) Powering on

The EH-150 starts operations after a maximum of 3.5 seconds have elapsed after power-up. If the power for input module is not completely started when the operation is commenced, the input that is supposed to be on will be received as OFF and operation proceeds, so make sure that the power for I/O module is completely started before operation is commenced.

Note: When expansion unit used, turn on the power supply of the expansion side, or turn on the power for the basic side and expansion side at the same time. If the basic side is powered up first, an error will occur.

(2) Instantaneous power failure actions

(a) When 100 V AC is supplied

Operation continues if power failure time is 10 ms or less.

(b) When 200 V AC is supplied

Operation continues if power failure time is 20 ms or less.

Note: Since EH-150 detects power failure by voltage level of internal 5VDC, the power failure resistance time depends on mounted module configuration. For this reason, only expansion unit can be reset due to instantaneous power failure. If this is not accepted, connect the AC power of PLC to AC input module and use it as interlock.

Note: Power for input should be supplied as long as CPU is running. If the power for PLC and AC input module is the same source, CPU can recognize input data as OFF at instantaneous power failure.

Note: If power failure time is more than the above resistance time, CPU will be reset and all data will be cleared.

8.4 Operation Parameter

The settings of "parameters," which are required to perform tasks such as creating programs, transferring programs to the CPU, are performed. The setting contents are explained below.

<table>
<tr><th>Item</th><th>Function</th><th>Description</th><th>When to use the function</th></tr>
<tr><td>1</td><td>Password</td><td>○ Register a password to a program in the four-digit hexadecimal format. The program with a password will not allow program operation nor changes unless the correct password is entered, so please exercise caution.
<u>Caution: The user will not be able to reset the password when it is forgotten, so exercise sufficient caution when using a password.</u>
Password is not set at the time of shipment.</td><td>Use when protecting the confidentiality of the program.</td></tr>
<tr><td>2</td><td>CPU type</td><td>○ Set the CPU name used to perform programming.
For EH-150, set to the following:
Set H-302 with the EH-150. However, set EH-150 with the LADDER EDITOR (Windows ® version) Ver 2.0 or later.</td><td>Always perform these settings when programming.</td></tr>
<tr><td>3</td><td>Memory assignment</td><td>○ Set the memory capacity.
For EH-150, set to the following:<table><tr><td>CPU</td><td>EH-CPU104/104A</td><td>EH-CPU208(A)/308(A)</td><td>EH-CPU316(A)/516</td></tr><tr><td>Memory type</td><td>RAM-04H</td><td>RAM-08H</td><td>RAM-16H</td></tr></table><table><tr><td>CPU</td><td>EH-CPU448/548</td></tr><tr><td>Memory type</td><td>RAM-48H</td></tr></table>Set the “CPU type” of Item 2 to H-302 for EH-CPU448. However in case of LADDER EDITOR for Windows® Ver. 2.13 and later , set the EH-150.</td><td>Always perform these settings when programming.</td></tr>
<tr><td>4</td><td>Operation parameters</td><td>○ Operation control
Perform these settings when controlling the operation and stopping using a specific I/O. If this is not set, operation will commence automatically when the RUN switch is set to "RUN."
○ Overload check time
Set this when you wish to stop the CPU operation when the set maximum processing time for a normal scan is exceeded. When the setting is not made, this is automatically set to initial value 100 ms.
○ Operation mode at abnormal occurrence
Set this when you wish to continue the CPU operation when the error generated by the CPU is minor. However, do not use this when not debugging.</td><td>Set according to the user's operation purposes.</td></tr>
<tr><td>5</td><td>I/O assignment</td><td>○ This sets the I/O assignment information of the CPU.
It is recommended to use the EH-150's convenient I/O assignment copy function.</td><td>Always perform these settings when programming.</td></tr>
<tr><td>6</td><td>Program name</td><td>Set the program name using a maximum of 16 kana or alphanumeric characters. The set program names can be written into the CPU along with the program, which will make the program verification and maintenance works easier.</td><td>Set this when easy program verification and maintenance are desired.</td></tr>
<tr><td>7</td><td>Power failure memory</td><td>This sets the range in which the data in a specified area in the CPU is to be stored upon CPU power off or when commencing RUN. Settings for R, WR, WM, TD, DIF, DFN are possible.</td><td>Set this when there is data you wish to maintain when operation is stopped.
The special internal output data is unconditionally saved for power failure by I/O number.</td></tr>
</table>

8.5 Test Operation

(1) Verification of interlock

Verify the performance of the interlock in preparation for unexpected accidents.
Create ladders such as an emergency stop ladder, protective ladder and interlock ladder outside the program controller. With regard to the relay output module, however, do not control the relay drive power supply to interlock with external loads.

(2) Operation without load

Before actually operating the loads in the system, test the program only and verify its operation.
Always perform this if there is any possibility of damaging the other party's equipment due to unexpected occurrences caused by program errors or other problems.

(3) Operation using actual loads

Supply power to the external input and external output to verify the actions.

MEMO

Chapter 9 PLC Installation, Loading, Wiring

9.1 Installation

(1) Installation location and environment

(a) When installing the EH-150, use the module within the [3.3 general specification] environment.

(b) Mount the PLC onto a metal plate.

(c) Install the PLC in a suitable enclosure such as a cabinet that opens with a key, tool, etc.

(2) Installing the base unit

(a) Precautions when installing the base unit

1] When installing the base unit, fix it securely with screws in 4 places (M4, length 20 mm (0.79 in.) or more) or DIN rail.

2] To keep the unit within the ambient temperature range when using,

a) Allow ample space for air circulation. (50 mm (1.97 in.) or more at top and bottom, 10 mm (0.39 in.) or more to the left and right)

b) Avoid installing the unit directly above equipment that generates a lot of heat (heater, transformer, large-capacity resistance, etc.)

c) When the ambient temperature reaches more than 55 °C, install a fan or cooler to lower the temperature to below 55 °C.

3] Avoid mounting inside a panel where high-voltage equipment is installed.

4] Install 200 mm (7.87 in.) or more away from high-voltage lines or power lines.

5] Avoid upside down, vertical or horizontal mounting.

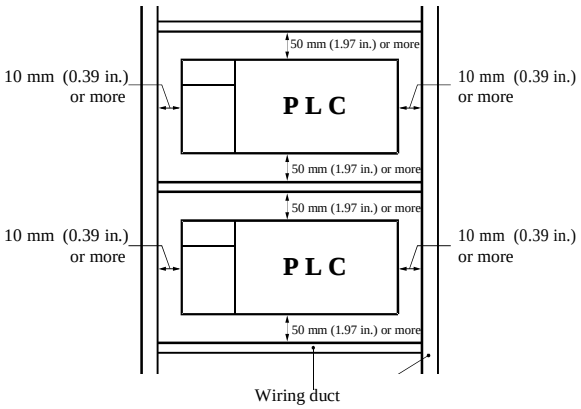


Figure 9.1 Installation space

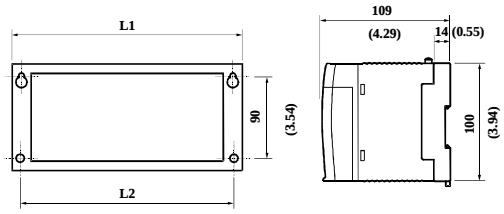


Figure 9.2 External dimensions

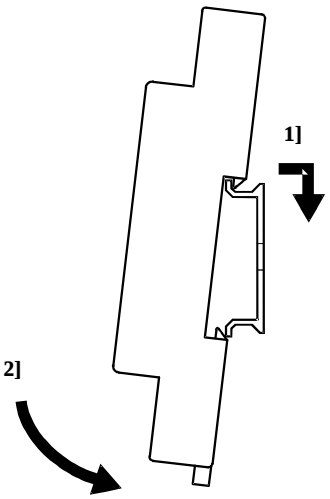
Dimensional table

Base	L1 (Outer dimensions)	L2 (Mounted dimensions)
3 slots	222.5 (8.76)	207 (8.15)
5 slots	282.5 (11.12)	267 (10.51)
8 slots	372.5 (14.67)	357 (14.06)
11 slots	462.5 (18.21)	447 (17.60)

Unit: mm (in.)

(b) Mounting to a DIN rail

Attaching to a DIN rail

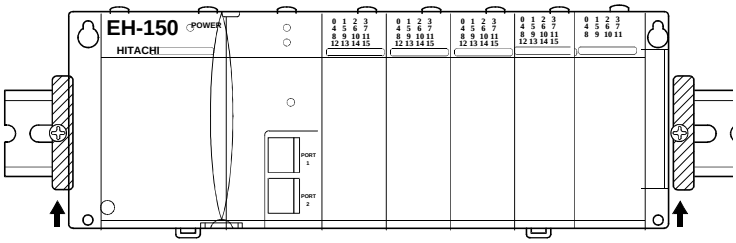


1] Hook the claw fixed at the bottom of the base unit, to the DIN rail.

2] Press the base unit into the DIN rail until it clicks.

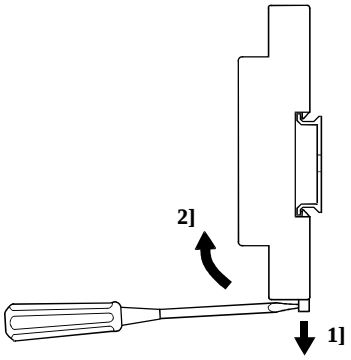
Note: After installation, check to make sure the base unit is securely fixed.

Fixing the unit



Secure the unit by installing DIN rail fixing brackets from both sides. (The product may go out of place if not secured with the fixing brackets.)

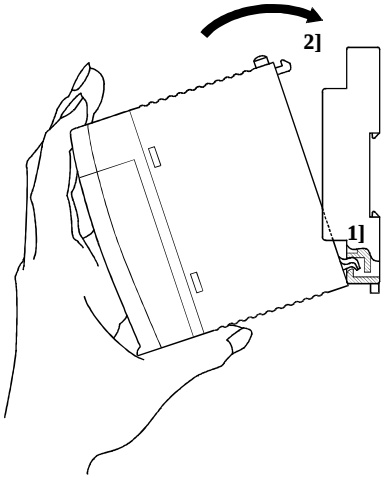
Removing the unit from the DIN rail



1] While lowering the DIN rail fixing mounting lever toward the bottom, raise the base upward to remove.

9.2 Loading the Module

(1) Installing



1] Hook the bottom part of the module to the hole in the base.

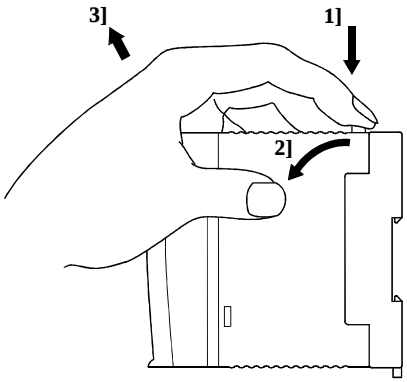
2] Press in the upper side of the module until it clicks.

Note 1: After loading the module, check to make sure it does not come out.

Note 2: Load the power module at the leftmost side of the base unit.

Note 3: Load the CPU module and I/O controller to the left of the power module.

(2) Removing



1] Push in the lock button.

2] With the lock button pushed in, pull the top of the module toward the front.

3] Raise it toward the top and pull it out.

Note: For the power module, pull it out while pushing down the two lock buttons.

9.3 Wiring

(1) Separation of the power system

For the power supply, there is power for the EH-150 PLC unit/power for the I/O signals/power for general equipment. These power supplies should be wired from separate systems as much as possible.
When these power supplies are supplied from one main power source, separate the wiring with a transformer or similar device, so that each power supply is a separate system.

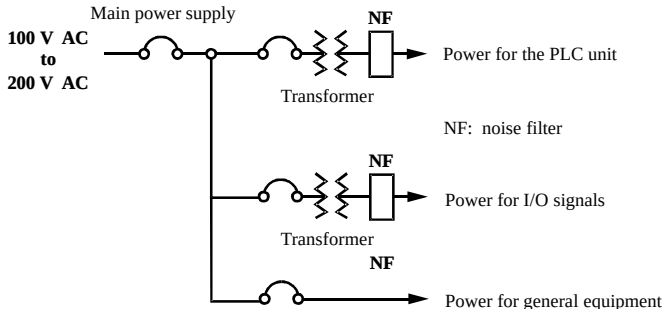


Figure 9.3 Example of power system diagram

(2) Regarding fail safe

1] Construct an interlock circuit outside the PLC.

When the PLC power supply is turned on or off, the lag time and difference in startup time between the PLC unit power and the external power (particularly DC power supply) for the PLC I/O module signals, may temporarily cause the I/O not to operate normally.
Do not control the power for the EH-YR12 relays to have it perform an interlock with the external load, etc. The relays may turn on even when power is not being supplied by an aluminum electrolytic condenser inside the module to drive the relays.
Also, it is plausible that a fault in the external power supply or a failure in the PLC unit will lead to abnormal actions. To prevent such actions from causing abnormal operation of the entire system, and from the viewpoint of creating a fail-safe mechanism, construct ladders such as an emergency stop circuit, the protect circuit, and the interlock circuit, for the sections that lead to a mechanical breakdown or accident from abnormal operation outside the PLC.

2] Install a lightning arrester

To prevent damage to equipment as a result of being struck by lightning, it is recommended that a lightning arrester be set up for each PLC power supply ladder.

The EH-150 detects power failures from a voltage drop of the internal 5 V DC power supply. For this reason, when the load in the unit's internal 5 V DC is light, the 5 V DC is retained for a long time and operations may continue for more than 100 ms. Therefore, when using the AC input module, an off delay timer for coordinating with the internal 5 V DC is necessary since the AC input signal turns off more quickly than the internal 5 V DC.

(3) Wiring to the power module

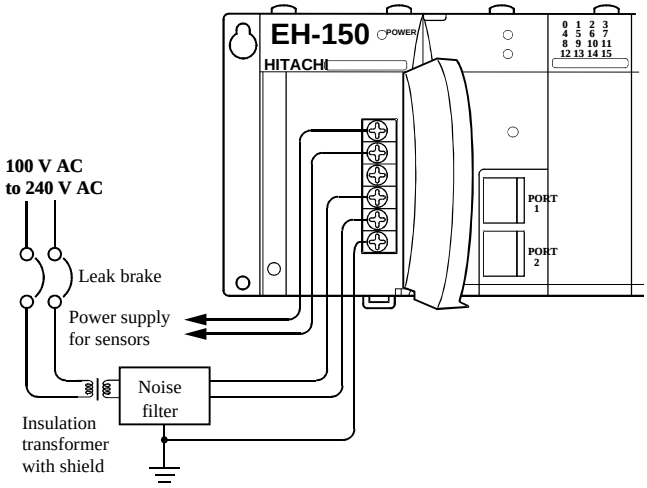


Figure 9.4 Wiring power diagram

(a) For power supply wiring, use a cable of 2 mm^2 (0.0031 in.) or more to prevent a voltage drop from occurring.

(b) The function ground terminal (FE terminal) should use cable of 2 mm^2 (0.0031 in^2.) or more and Class D grounding (100 Ω or less). The appropriate distance for ground cable is within 20 m (65.62 ft.).

1] Shared with instrumentation panel, relay panel grounding

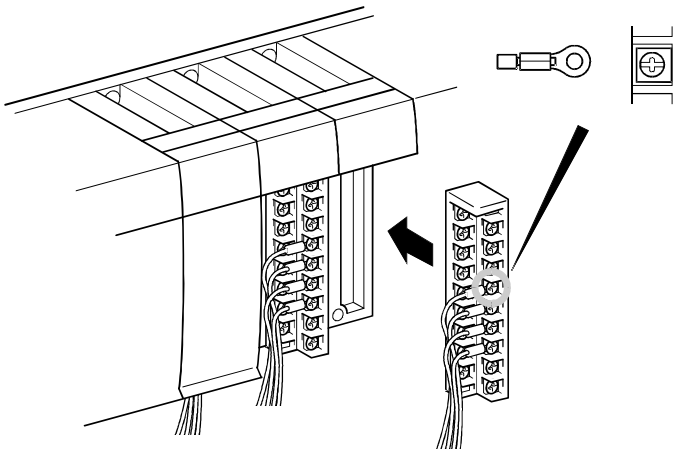
2] Avoid joint grounding with equipment that can generate noise such as high-frequency heating furnace, large power panel (several kW or more), thyristor exchanger, electric welders, etc.

3] Be sure to connect a noise filer (NF) to the power cable.

(c) The terminal screw is an M3. When wiring, tighten screws within a torque range of 0.49 to 0.78 N·m.

(d) Use the same power supply system for the basic and expansion units.

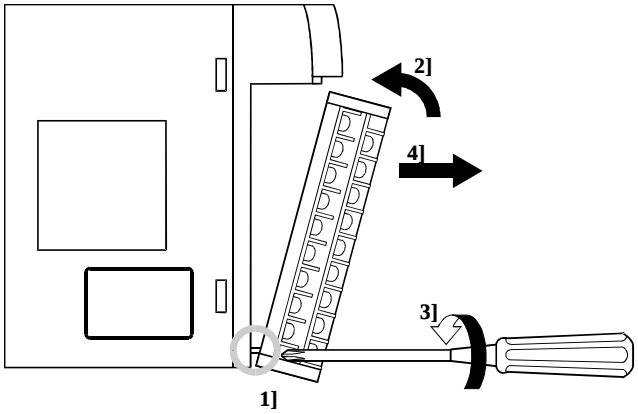
(4) Wiring cable for I/O signals



Screws for all terminals are M3.
Tighten within a torque range of 0.49 to 0.78 N·m.
When using a crimp terminal, use one with an outer diameter of 6 mm (0.24 in.) or less.
Use only up to two crimp terminals in the same terminal. Avoid clamping down more than three at the same time.
Use a cable thickness of maximum 0.75 mm^2 (0.0011 in^2.). (Use a 0.5 mm^2 (0.00075 in^2.) cable when adding two crimp terminals in the same terminal).

Note: When corresponding to CE marking EMC command is necessary, use shielded cable for the relay output module.

Attaching the terminal block



1] Align the tip of the terminal block mounting screws to the screw section of the I/O cover insertion fittings.
2] Push in the top of the terminal block until the I/O cover claw section locks with a click.
3] While holding down the upper part of the terminal block, tighten the terminal block mounting screws.
4] Pull on the top of the terminal block to make sure that it is locked and cannot come out.

Note: If the terminal block is removed, always reinstall it following the instructions above.

(5) Input wiring for thep input module

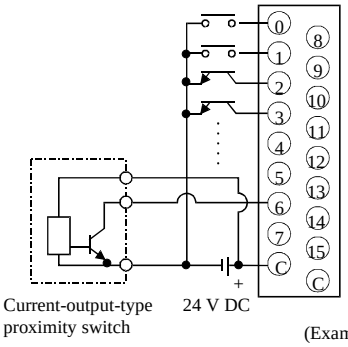
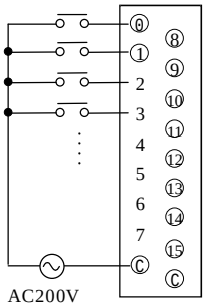
DC input (EH-XD16, EH-XD8)	AC input (EH-XA16, EH-XAH16)
	
Current-output-type proximity switch 24 V DC (Example of EH-XD16)	AC200V (Example of EH-XAH16)

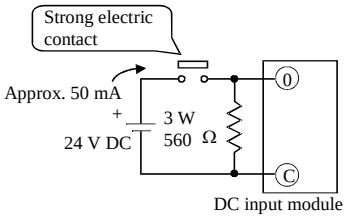
Figure 9.5 Input wiring

(a) DC input module

1] When all input terminals (X0, X1, ...) and the common terminal (C) are loaded with 24 V DC, inputs change to ON, and approximately 6.9 mA of current in the case of the EH-XD8, and approximately 4mA in the case of the EH-XD16, flow to the external input contacts.

2] For sensors such as a proximity switch or photoelectric switch, current output type (transistor open collector) can be directly connected. For voltage-output-type sensors, connect them to the input terminal after first going through the transistor.

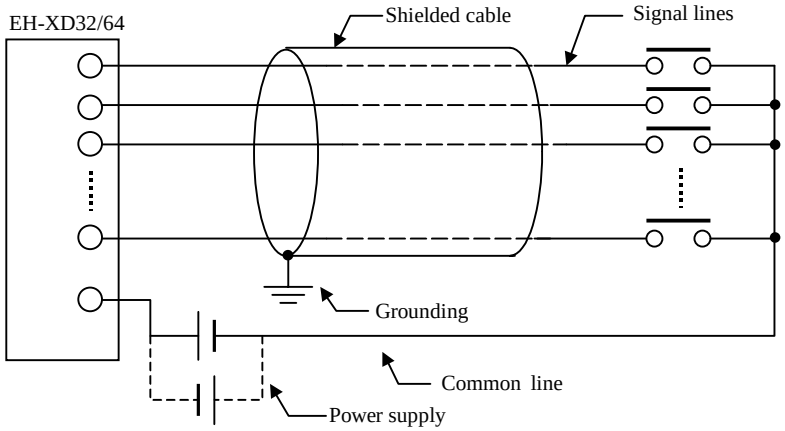
3] Measures to prevent faulty contact in a strong electric contact



The current that flows to a contact when external contacts are closed is approximately 6.9 mA for the EH-XD8, and approximately 4 mA for the EH-XD16. If the use of a strong electric contact cannot be avoided, add resistance as shown in the diagram at left and supply sufficient current to the contact to prevent a faulty contact.

4] Max. cable length is 30 m (98.43 ft.)

(b) 32/64-point DC input module (Based on CE marking)



Caution

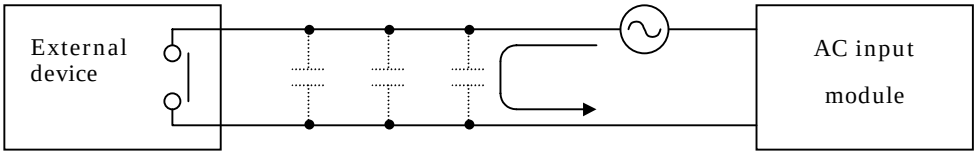
1] Wire only the signal line through the shielded cable. Cable shield should be grounded.

2] Do not wire the common line with signal lines in shielded cable. Be sure to wire it independently and separately from the power line, I/O lines or power supply line.

3] Power supply should be installed at common terminal of input module as close as possible.

(c) AC input module

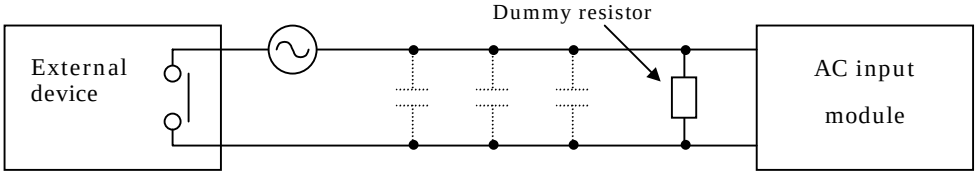
In case of AC input module, input voltage may exist if input wiring is long although no device drives. This phenomenon is caused from leakage current due to floating capacitance between lines.



The countermeasures are [1] or [2] as follows. This voltage due to electrostatic coupling must be half of max. OFF voltage or less.

[1] To install dummy resistor in parallel so that impedance of input module is lower.

[2] To replace power supply at drive (external device) side.



(6) Output wiring for the output module

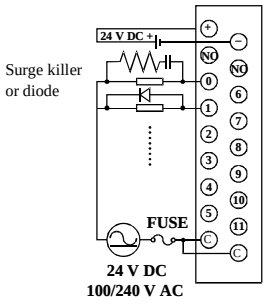
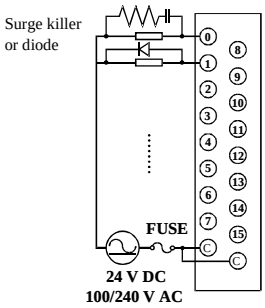
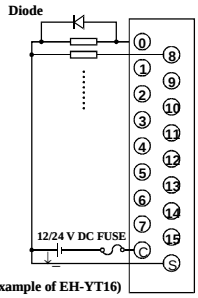
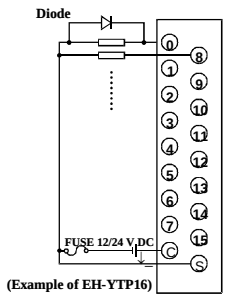
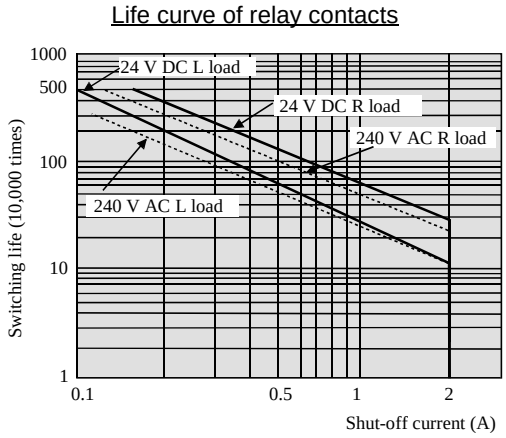
Item	Relay output (EH-YR12)	Relay output (EH-YR16)	Transistor output (sink type) (EH-YT16, EH-YT8)	Transistor output (source type) (EH-YTP16, EH-YTP16S, EH-YTP8)
External wiring				

Figure 9.6 Output wiring

(a) Wiring for the relay output module

1] Life of relay contacts



Life of the contact is also in squared reverse proportion to the current, so be aware that interrupting rush current or directly driving the condenser load will drastically reduce the life of the relay.

When switching is done with high frequency, use a transistor output module.

2] Surge killer

For inductive load, connect a surge killer (condenser 0.1 μF, + resistance of around 100 Ω) in parallel to the load. Also, for DC load, connect a flywheel diode.

3] Fuse

A fuse is not built into this module. Install a 6 A fuse in the common to prevent the external wiring from burning out.

4] Power supply for driving the relays

If a 24 V DC power supply is connected to drive the relays, take care with respect to the polarity when connecting. There is a risk that the internal circuit will be damaged if the wiring is done incorrectly. Also, do not perform an interlock, etc. to the external load with the power supply for driving the relays.

(b) Wiring for the transistor output module

5] Flywheel diode

For inductive load, connect a flywheel diode in parallel.

6] S and C terminals

Always connect an S terminal and C (common) terminal. If the module is used without connecting these terminals, the internal flywheel diode does not function and there is a risk that the module will malfunction or breakdown.

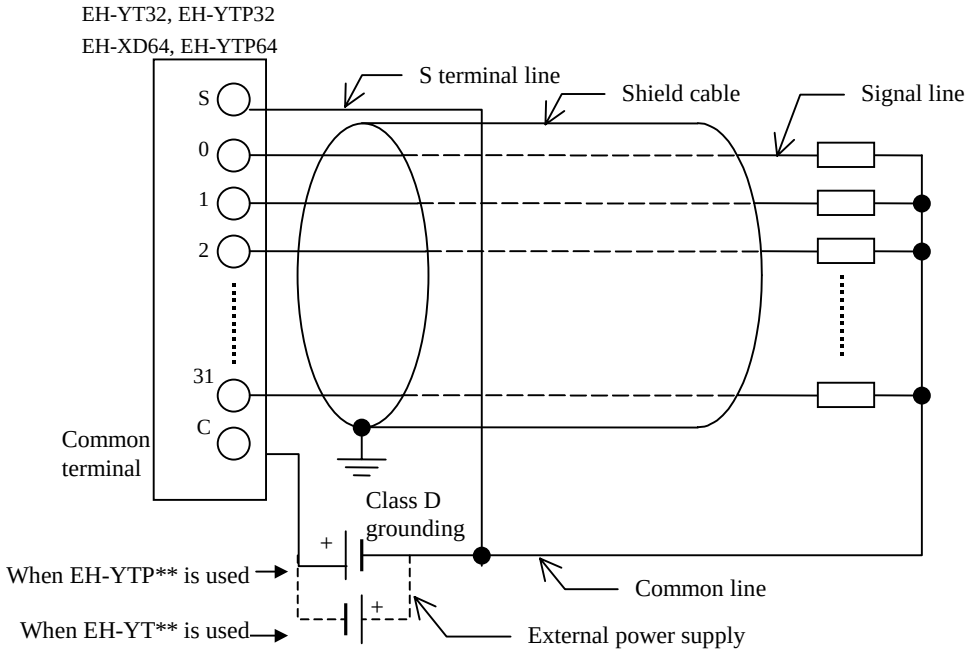
7] Fuse

A fuse is inserted in the common to prevent the external wiring from burning out, but this does not protect transistor elements. Therefore, when the external load is shorted, these elements are destroyed, so exercise caution. If the external load shorts, please contact us for repair.

Also, if the fuse blows, there will be no output even if the LED lights up. (The fuse out lamp for the module at this time as well as a CPU module error will not be displayed.)

Note: If the fuse is melted or blown, do not supply power to the module after changing the fuse without eliminating the source of the problem. Damage escalation, smoke, etc., may otherwise result.

(c) Wiring for the 32-point/64 point output module EH-YT32/YTP32, EH-YT64/YTP64 (Based on CE marking)



Cautionary notes

1] Wire only the signal line through the shielded cable, and provide class D grounding on the shielded cable side.

2] Do not wire the common line or S terminal line through the shielded cable. Be sure to wire them independently and separately from the power line, I/O lines or power supply line.

3] The supply line to the external power supply should be wired as close as possible to the common terminal of the output module.

(7) Analog module I/O wiring

- Do not apply excess voltage to the analog input module beyond the rated input voltage. Similarly, do not subject the module to current that exceeds the rated input current. Connecting the analog input module to a power supply other than the specified types may cause damage to the product or burning of its internal components.
- For unused channels of the analog input module, short the input terminals before use.
- For unused channels of the analog output module (unused current output channel, 2 to 3 channels), short the outputs before use.
- When wiring the external lines of the analog module, route them through the shielded cables while separating them from other power lines or signal lines subject to differential voltage. Shielded cables must be grounded on one side. However, whether it is more effective to ground on one side or leave both sides open, depends on the noise environment condition in the actual use. Provide appropriate grounding based on the noise environment.
- Use separate piping for the AC power supply line and the signal/data lines.
- Wire the signal lines and data lines as close as possible to the grounded surface of the cabinet or a metal bar.

(8) Wiring to the module terminal

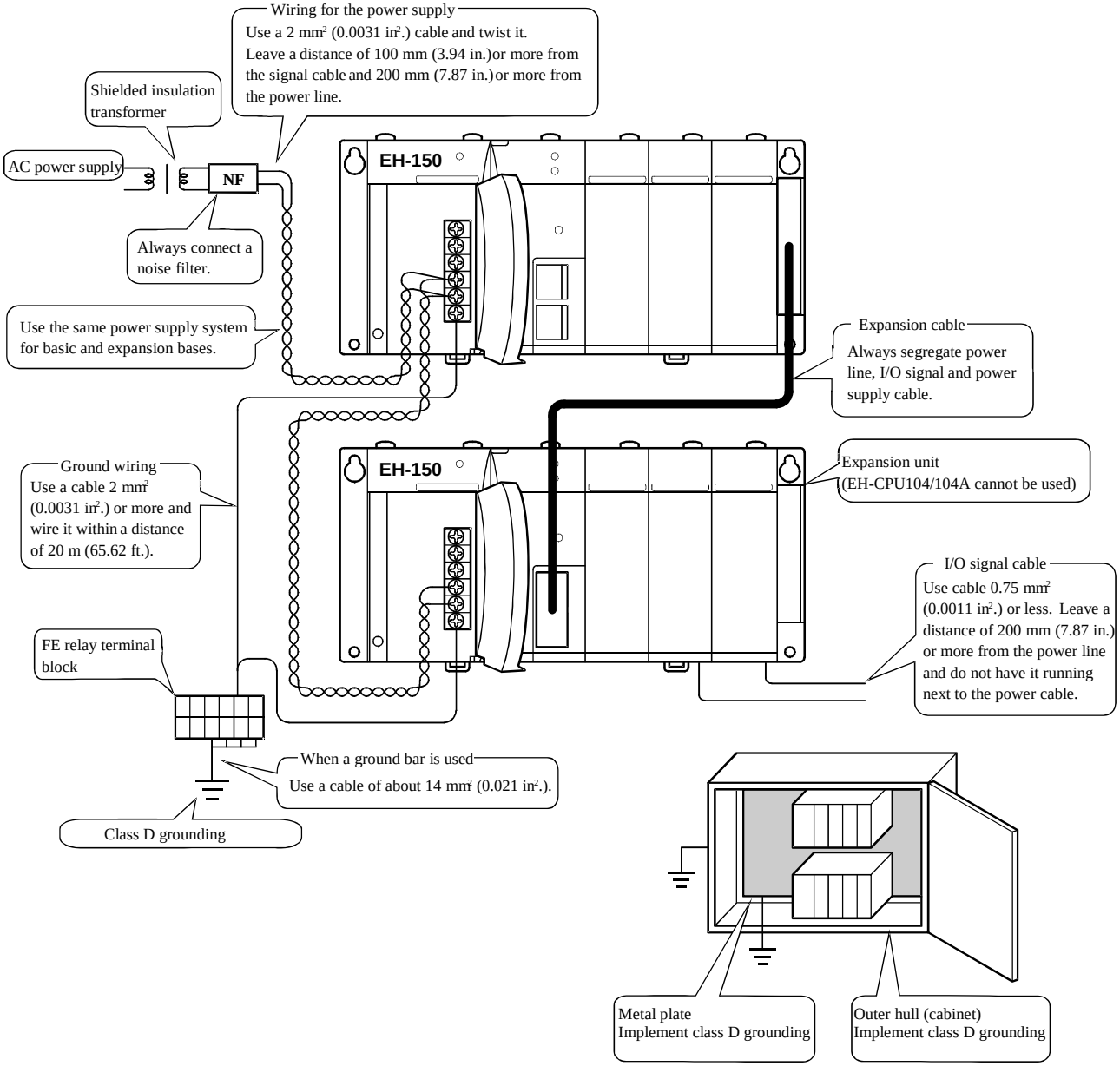


Figure 9.7 Example of wiring

MEMO

Chapter 10 Communication Specifications

10.1 Features

The EH-150 has two communication ports. Supported function of each port is shown in the below table.

10.1.1 Communication port functions

Port type		RS-232C			RS-422/485				
		Dedicated port *		General purpose port	Dedicated port *				General purpose port
		Procedure 1	Procedure 2		Procedure 1		Procedure 2		
					Std. (1:1)	St. No. (1:N)	Std. (1:1)	St. No. (1:N)	
Connected devices		PC, Programmer, Modem, HMI, etc.	PC etc.	PC etc	PC, Programmer, HMI etc.	PC etc.	PC etc.	PC etc.	PC etc.
Port 1	CPU104/208	✓	-	✓	-	-	-	-	-
	CPU104A/208A	✓	✓	✓	-	-	-	-	-
	CPU308/316	✓	-	✓	-	-	-	-	✓
	CPU308A/316A	✓	✓	✓	✓	✓	✓	✓	✓
	CPU448/448A	✓	✓	✓	✓	✓	✓	✓	✓
	CPU516/548	✓	✓	✓	✓	✓	✓	✓	✓
Port 2	All CPU	✓	-	-	-	-	-	-	-

* Dedicated port = Programming port for PC, HMI, SCADA.

10.1.2 Port 1 setup method

Set the mode setting switch, power up again and set the special internal output values according to table below.

(1) General port setup

In case of general purpose port, with using the TRNS 0 and RECV 0 commands in a user program, set the interface according to the table below. After the setting, the system overwrites the following values, and switches the interface type. If user set value is out of the possible range, the system will not accept. Be sure to check the safety of the unit to be connected before setting.

Table 10.2 General purpose port setting for port 1

mode SW		WRF036				Interface
		Built-in termination resistor OFF *		Built-in termination resistor ON *		
2	5	Use set value	System set value	Use set value	System set value	
OFF	OFF	H0003	H0002	-	-	RS-232C
		H0005	H0004	H0015	H0014	RS-422
		H0009	H0008	H0019	H0018	RS-485

* If the built-in termination resistor is set ON, built-in 100Ω resistor will be connected between lines (between receiver lines in case of RS-422). So no external termination is necessary.

(2) Dedicated port mode setup (Not supported by EH-CPU104(A))

When a modem is used, configure the following setup:

Table 10.3 Modem settings of port 1

Mode SW		WRF037			Contents of setting	
2	5	User set value	System set value *1	Value displayed after power restart	Interface	Control procedure
ON	OFF	H8000	H0000	H0000	RS-232C	Transmission control procedure 1
		HC000	H4000	H4000		Transmission control procedure 2

(3) Dedicated port setup

If the port 1 of EH-***A/448/516/548 is used as a dedicated port, configure the following setup.

Table 10.4 Dedicated port settings of port 1(EH-CPU***A/448/516/548)

No.	Mode SW		WRF037			Contents of setting	
	2	5	User set value	System set value *2	Value displayed after power restart	Interface	Control procedure
1	-	ON	H8000	H0000	H0000	RS-232C *1	Transmission control procedure 1
2			HC000	H4000	H4000		Transmission control procedure 2
3			H8100	H0100	H0500	RS-422 (Built-in termination OFF) *3	Transmission control procedure 1
4			HC100	H4100	H4500		Transmission control procedure 2
5			HA1**	H21**	H25**		Transmission control procedure 1 with station No.
6			HE1**	H61**	H65**		Transmission control procedure 2 with station No.
7			H8200	H0200	H0A00	RS-485 (Built-in termination OFF) *3	Transmission control procedure 1
8			HA2**	H22**	H2A**		Transmission control procedure 1 with station No.
9			HE2**	H62**	H6A**		Transmission control procedure 2 with station No.
			H9100	H1100	H1500	RS-422 (Built-in termination ON) *3	Transmission control procedure 1
			HD100	H5100	H5500		Transmission control procedure 2
			HB1**	H31**	H35**		Transmission control procedure 1 with station No.
			HF1**	H71**	H75**		Transmission control procedure 2 with station No.
			H9200	H1200	H1A00	RS-485 (Built-in termination ON) *3	Transmission control procedure 1
			HB2**	H32**	H3A**		Transmission control procedure 1 with station No.
			HF2**	H72**	H7A**		Transmission control procedure 2 with station No.

**: This means station number in 1:N communication. Set[*3] a 2-digit BCD value from 00 to 31. If a value outside this range is specified as a station number, the specified value will be displayed, but the system will operate using a BCD value of 31.

*1 EH-CPU104A/208A supports only No.1 and 2 (RS232C only).

*2 The current interface status is displayed by b11 and b10 of WRF037. For this reason, other values may be displayed as system set values depending on previous setting. Please refer to chapter 13 for further information about WRF037.

*3 If the built-in termination resistor is set ON, built-in 100Ω resistor will be connected between lines (between receiver lines in case of RS-422). So no external termination is necessary.

10.2 Dedicated Port

The specification of dedicated port is shown in table 10.5.
By the dedicated port, CPU can be programmed or monitored by peripheral devices or PC or HMI. In addition, supervisor system or special sytem can be connected as well by making communication software.
Be sure to check cables and setting switches carefully beforehand.

Table 10.5 Specifications for a dedicated port

<table>
<tr><th>Item</th><th colspan="6">Specification</th></tr>
<tr><td rowspan="7">Transmission speed</td><td colspan="6">4,800 bps, 9,600 bps, 19,200 bps, 38,400 bps
Configured by setting switch.</td></tr>
<tr><td colspan="4">Port 1</td><td colspan="2">Port 2 *1</td></tr>
<tr><td>SW3</td><td>SW4</td><td colspan="2">Transmission speed setting</td><td>SW6</td><td>Transmission speed setting</td></tr>
<tr><td>ON</td><td>ON</td><td colspan="2">4,800 bps</td><td rowspan="2">ON</td><td>PHL = Low: 9,600 bps</td></tr>
<tr><td>OFF</td><td>ON</td><td colspan="2">9,600 bps</td><td>PHL = High: 38,400 bps</td></tr>
<tr><td>ON</td><td>OFF</td><td colspan="2">19,200 bps</td><td rowspan="2">OFF</td><td>PHL = Low: 4,800 bps</td></tr>
<tr><td>OFF</td><td>OFF</td><td colspan="2">38,400 bps</td><td>PHL = High: 19,200 bps</td></tr>
<tr><td>Interface*2</td><td colspan="2">RS-232C</td><td>RS-422</td><td>RS-485</td><td colspan="2">RS-232C</td></tr>
<tr><td>Max cable length</td><td colspan="2">15 m</td><td>500 m</td><td>500 m</td><td colspan="2">15 m</td></tr>
<tr><td>Connection mode (max connected units)</td><td colspan="2">1 : 1</td><td>1 : N (32 units)</td><td>1 : N (32 units)</td><td colspan="2">1 : 1</td></tr>
<tr><td>Communication system</td><td colspan="6">Half duplex</td></tr>
<tr><td>Synchronization system</td><td colspan="6">Start-stop synchronization</td></tr>
<tr><td>Startup system</td><td colspan="6">One-sided startup using the host side command</td></tr>
<tr><td>Transmission system</td><td colspan="6">Serial transmission (bit serial transmission)</td></tr>
<tr><td>Transmission code</td><td colspan="6">ASCII</td></tr>
<tr><td>Transmission code configuration</td><td colspan="6">ASCII: 7-bit data, 1 start, 1 stop, even number parity</td></tr>
<tr><td>Transmission code outgoing sequence</td><td colspan="6">Sent out from the lowest bit in character units</td></tr>
<tr><td>Error control</td><td colspan="6">Vertical parity check, sum check, overrun check, framing check</td></tr>
<tr><td>Transmission unit</td><td colspan="6">Message unit (variable length)</td></tr>
<tr><td>Maximum message length</td><td colspan="6">503 bytes*3 (including control characters)</td></tr>
<tr><td>Control procedure</td><td colspan="6">H-series dedicated procedure (high protocol)
Standard procedure 1 (transmission control procedure 1),
simplified procedure (transmission control procedure 2)*4</td></tr>
<tr><td>Connector used</td><td colspan="6">CPU side: TM5RJ3-88 8-pin modular connector (Hirose)
Partner side: Comparable to TM10P-88P (Hirose)</td></tr>
</table>

*1 When the PHL switch is set high, +12 V DC is supplied from the connector pin 4 of port 2.
*2 RS-422/485 is supported by EH-CPU308A/316A/448(A)/516/548 only.
*3 When H series' dedicated procedure with station number is used in the EH-CPU308A/316A/448(A)/516/548, the maximum message length will be 505 bytes (including control character).
*4 Transmission control procedure 2 are supported by the EH-CPU***A/448/516/548 only.

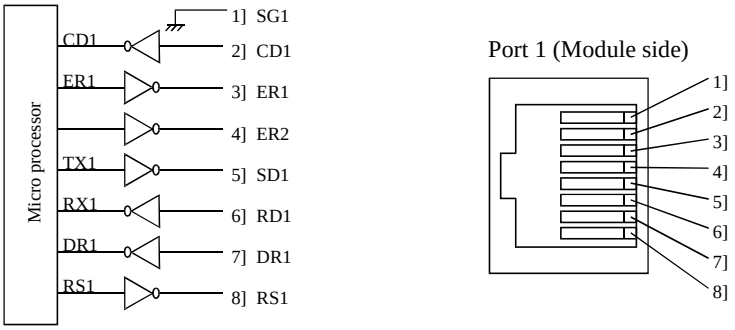


Figure 10.1 Circuit diagram and pin numbers for port 1

Table 10.6 List of port 1 signals

Pin No.	Signal abbreviation	Direction		Meaning
		CPU	Host	
1]	SG1	⟷		Ground for signal
2]	CD1	←		Notification signal during carrier received
3]	ER1	→		Communication enabled signal When this signal is high level, communication is possible.
4]	ER2	→		Outputs High
5]	SD1	→		Data sent by the CPU
6]	RD1	←		Data received by the CPU
7]	DR1	←		Peripheral units connected signal When this signal is high level, indicates that dedicated peripherals are connected.
8]	RS1	→		Transmission request signal When this signal is high level, indicates that the CPU can receive data.

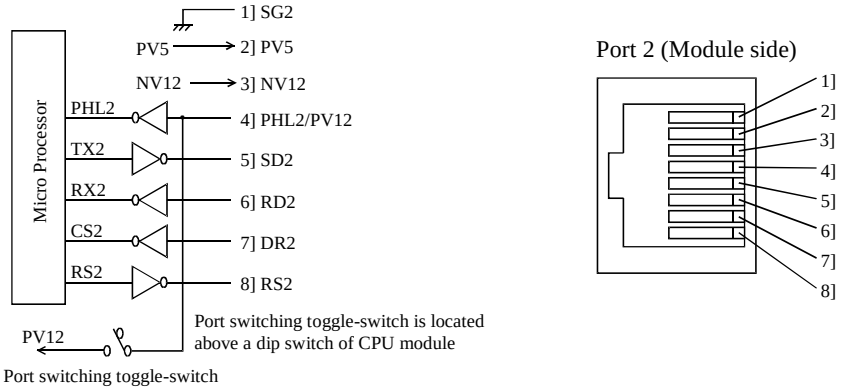


Figure 10.2 Circuit diagram and pin numbers for port 2

Table 10.7 List of port 2 signals

Pin No.	Signal abbreviation	Direction		Meaning
		CPU	Host	
1]	SG2	⟷		Signal ground
2]	PV5	→		5 V DC is supplied.
3]	NV12	→		–12 V DC is supplied.
4]	PHL2 / PV12	←		Peripheral units connected signal/+12 V DC output. Indicates that a dedicated peripheral is connected. (when the switching switch is off). 12 V DC is supplied. (when the switching switch is on).
5]	SD2	→		Data sent by the CPU
6]	RD2	←		Data received by the CPU
7]	DR2	←		Receive enabled signal When this signal is high level, indicates that connected devices can receive data.
8]	RS2	→		Transmission request signal When this signal is high level, indicates that the CPU can receive data.

10.3 General Purpose Port

Port 1 can be used as a general-purpose port. In this case, user program must be prepared with using serial communication command TRNS 0 and RECV 0. The specifications of general purpose port is given in Table 10.8 below.
Be sure to check the connecting cables and setting switches carefully beforehand. Refer to P4-6 for setting switch.

Table 10.8 Specifications for a general-purpose port

Item	Specification		
Transmission speed	300, 600, 1,200, 2,400, 4,800, 9,600, 19,200 bps		
Communication system	Half duplex		
Synchronization system	Start-stop synchronization		
Startup system	One-sided startup using the host side command		
Transmission system	Serial transmission (bit serial transmission)		
Transmission code	User defined		
Transmission code configuration	User setting: (1 start, 7 or 8 bit data, NON or EVEN or ODD parity, 1 or 2 stop)		
Transmission code outgoing sequence	Sent out from the lowest bit in character units		
Error control	Vertical parity check, overrun check, framing check		
Transmission unit	Message unit (variable length)		
Maximum message length	CPU104/208/308/316 : 256 bytes (including control characters) CPU***A/448/516/548 : 1,024 bytes (including control characters)		
Interface *	RS-232C	RS-422	RS-485
Maximum cable length	15 m (49.2 ft.)	500 m (1640.5 ft.)	500 m (1640.5 ft.)
Connection mode (max connected units)	1 : 1	1 : N (32 units)	1 : N (32 units)
Control procedure	No procedure		
Control code	User defined		
Connector used	CPU side: TM5RJ3-88 8-pin modular connector (Hirose) Partner side: Comparable to TM10P-88P (Hirose)		

* EH-CPU 104(A)/208(A) supports only RS-232C.

TRNS0, RECV0 commands specifications
Transmission speed set value (For other specification details, refer to the item in the list of instructions.)

Baud rate	Set value
19,200 bps	H0006
9,600 bps	H0005
4,800 bps	H0004
2,400 bps	H0003
1,200 bps	H0002
600 bps	H0001
300 bps	H0000

10.3.1 RS-232C interface

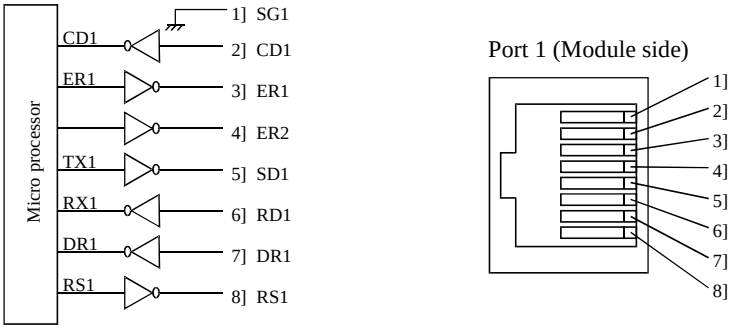


Figure 10.3 Circuit diagram and pin numbers

Table 10.9 List of port 1 signals

Pin No.	Signal abbreviation	Direction		Meaning
		CPU	Host	
1]	SG1	⟷		Ground for signal
2]	CD1	←		Notification signal during carrier received
3]	ER1	→		Transmission enabled signal When this signal is high level, communication is possible.
4]	ER2	→		Outputs High.
5]	SD1	→		Data sent by the CPU
6]	RD1	←		Data received by the CPU
7]	DR1	←		Reception request signal When this signal is high level, indicates that dedicated the host can send data. Connect it to RTS signal on the host.
8]	RS1	→		Transmission request signal When this signal is high level, indicates that the CPU can receive data. Connect it to CTS on the host.

10.3.2 RS-422/485 interface

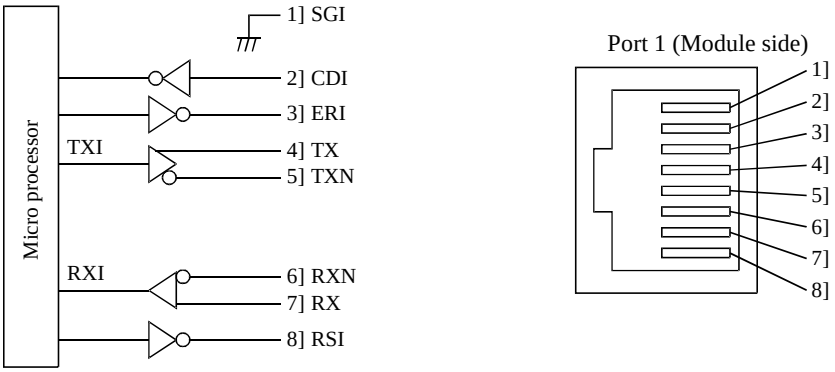


Figure 10.4 Circuit diagram and pin numbers

Table 10.10 List of port 1 signal

Pin No.	Signal abbreviation	Direction		Meaning
		CPU	Host	
1]	SG1	⟷		Signal ground
2]	CD1	←		Not used. Do not connect.
3]	ER1	→		Not used. Do not connect.
4]	TX	→		Data sent by the CPU +
5]	TXN	→		Data sent by the CPU –
6]	RXN	←		Data received by the CPU –
7]	RX	←		Data received by the CPU +
8]	RS1	→		Not used. Do not connect.

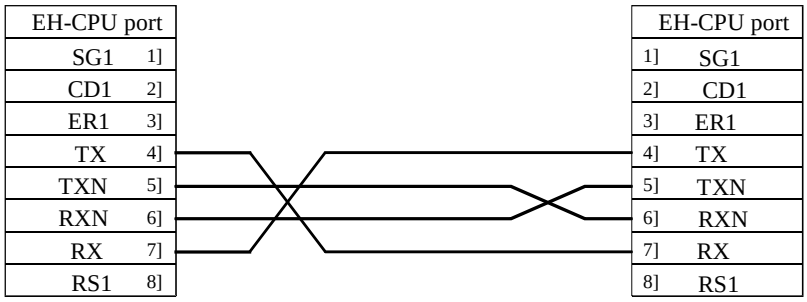


Figure 10.5 RS-422 signal connection diagram

If used as RS-485 interface, connect pin numbers 4] and 7], and 5] and 6], externally.

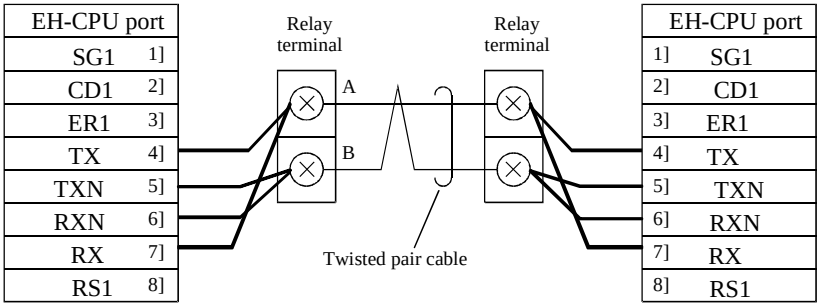


Figure 10.6 RS-485 signal connection diagram

10.3.3 1: N communication (RS-485)

(1) Precautions

When performing 1 to N communication using RS-485, do so in polling/selecting mode. When creating a ladder program, observe the following precautions.

1] Perform communication by making sure the master station and slave station are using the same start code.

2] At the master station, send a request by specifying the station number of the slave station.

3] At the slave station, send a response only when the request received from the master station is addressed to the own station. Set the station so that it will reset the mode and wait for the next request in the event it received a request addressed to other station.

4] At the master station, send a new request after at least 20 ms (t_p in figure below) has elapsed from the time it completed receiving the last response from the slave station.

5] At the slave station, send a response after at least 20 ms (t_s in figure below) has elapsed from the time it completed receiving the request from the master station.

Figure 10.7 shows an example of 1 to N transmission sequence. This example shows a sequence in which the master station sends a series of requests to slave stations 1 to 3, with each slave station sending a response to the received request.

In Figure 10.7, a plateau expressed in solid line indicates that the station received a transmission addressed to the own station. Similarly, a plateau expressed in dotted line indicates that the station received a transmission addressed to other station.

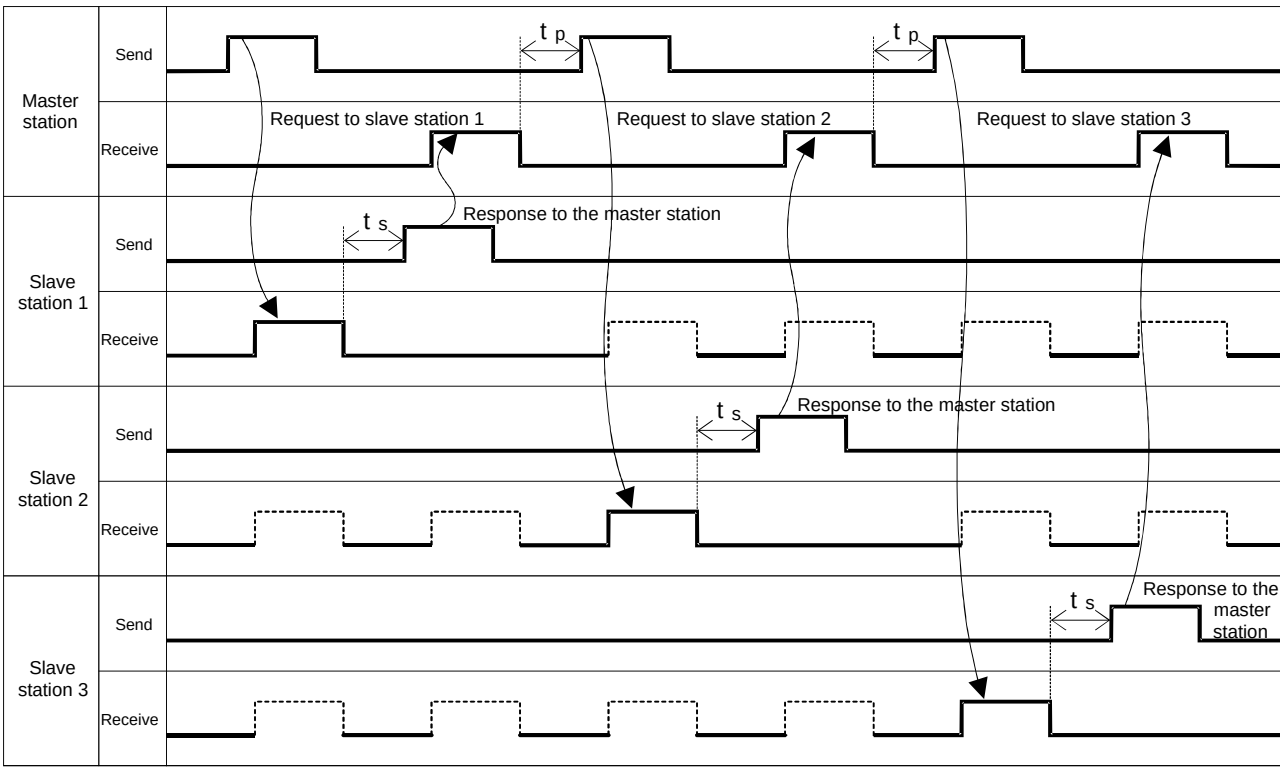


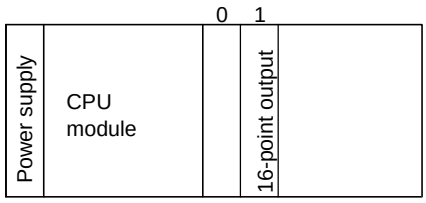
Figure 10.7 1:N transmission and reception sequence

(2) Sample Program

The following shows a sample program that performs communication between one master station and three slave stations using RS-485.

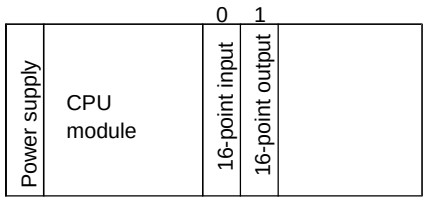
1] Mounting the module

(a) Master station side



Mount the 16-point output module in slot "1" of the basic base.

(b) Slave station side



Mount the 16-point input module in slot "0" of the basic base and mount the 16-point output module in slot "1" of the basic base.

2] Assigning internal outputs

The sample program is created using the following assignments. In actual cases, change the I/O numbers and other items according to the application.

(a) Assigning internal outputs on the master station side

I/O	No.	Usage
WM	100 to 10E	TRNS 0 instruction Parameter area (s to s+14)
R	000 to 00B	TRNS 0 instruction Communication control bit area (t to t+11)
	100	Transmission data setting completion flag
WR	0000 to 001F	Transmission data area (32-word)
	0100 to 011F	Reception data area (32-word)
	4000	Station number of target slave station
	4001	Number of slave stations
WL	001 to 003	Reception data storage area

(b) Assigning internal outputs on the slave station side

I/O	No.	Usage
WM	0000 to 000E	RECV 0 instruction Parameter area (s to s+14)
	200 to 21F	Transmission data area (32-word)
	300 to 31F	Reception data area (32-word)
WR	0000 to 000E	TRNS 0 instruction Parameter area (s to s+14)
	0200 to 021F	Transmission data area (32-word)
	0300 to 031F	Reception data area (32-word)
	4000	Station number of communicating slave station
	4001	Own station number
WL	001 to 003	Reception data storage area

3] Transmission format

The following shows the formats that are used when transmitting data between the master station and slave station.

(a) Request format from the master station to slave station (a maximum of 3 bytes)

Start code	Slave station number	End code
02H	1 to 3	0DH

(b) Response format from the slave station to master station (a maximum of 5 bytes)

Start code	Own station number	Data		End code
02H	1 to 3	Any *	Any *	0DH

*: Any data can be set except for the end code (0DH). The slave station number is set in this sample program.

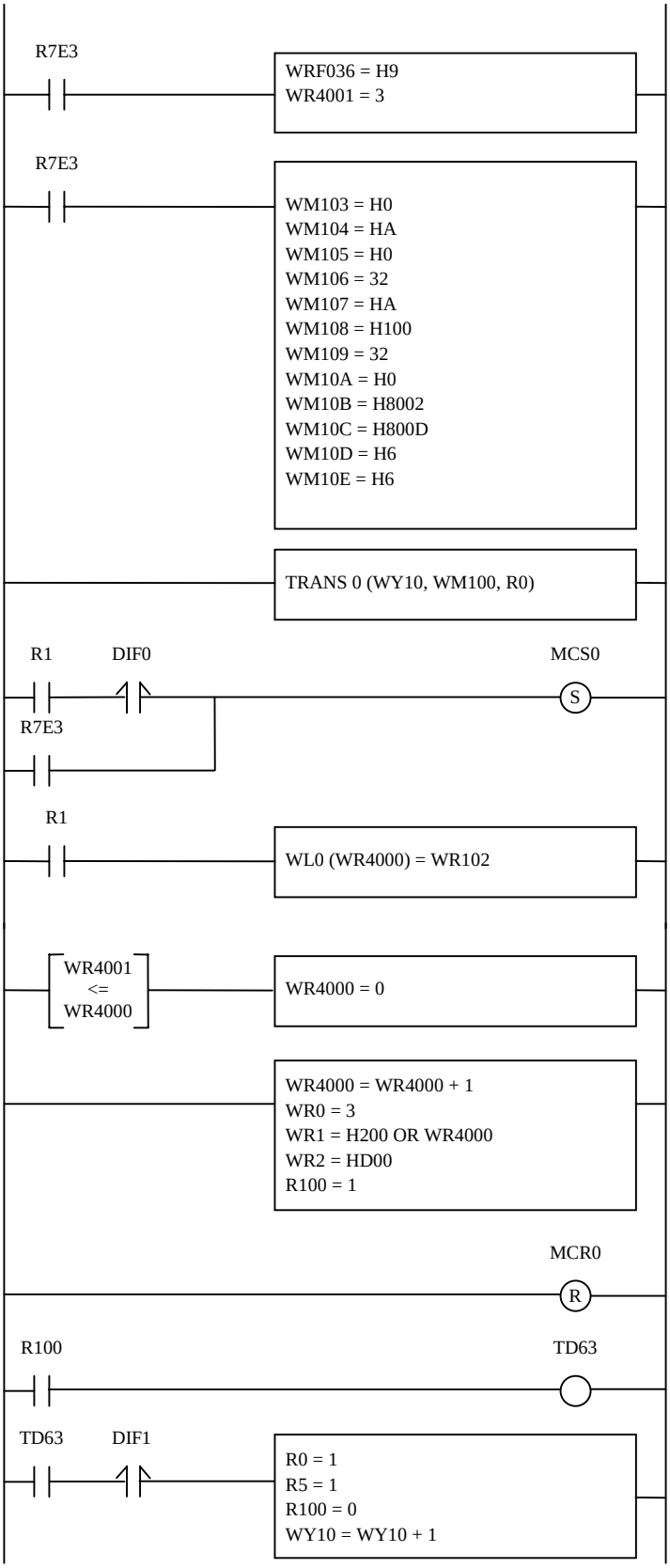
4] Reception result on the master station side

When the transmissions between slave stations 1 to 3 have completed successfully, the corresponding data are set in the WL area of the master station as shown below. In this sample program, the slave stations set their own station number as part of the data.

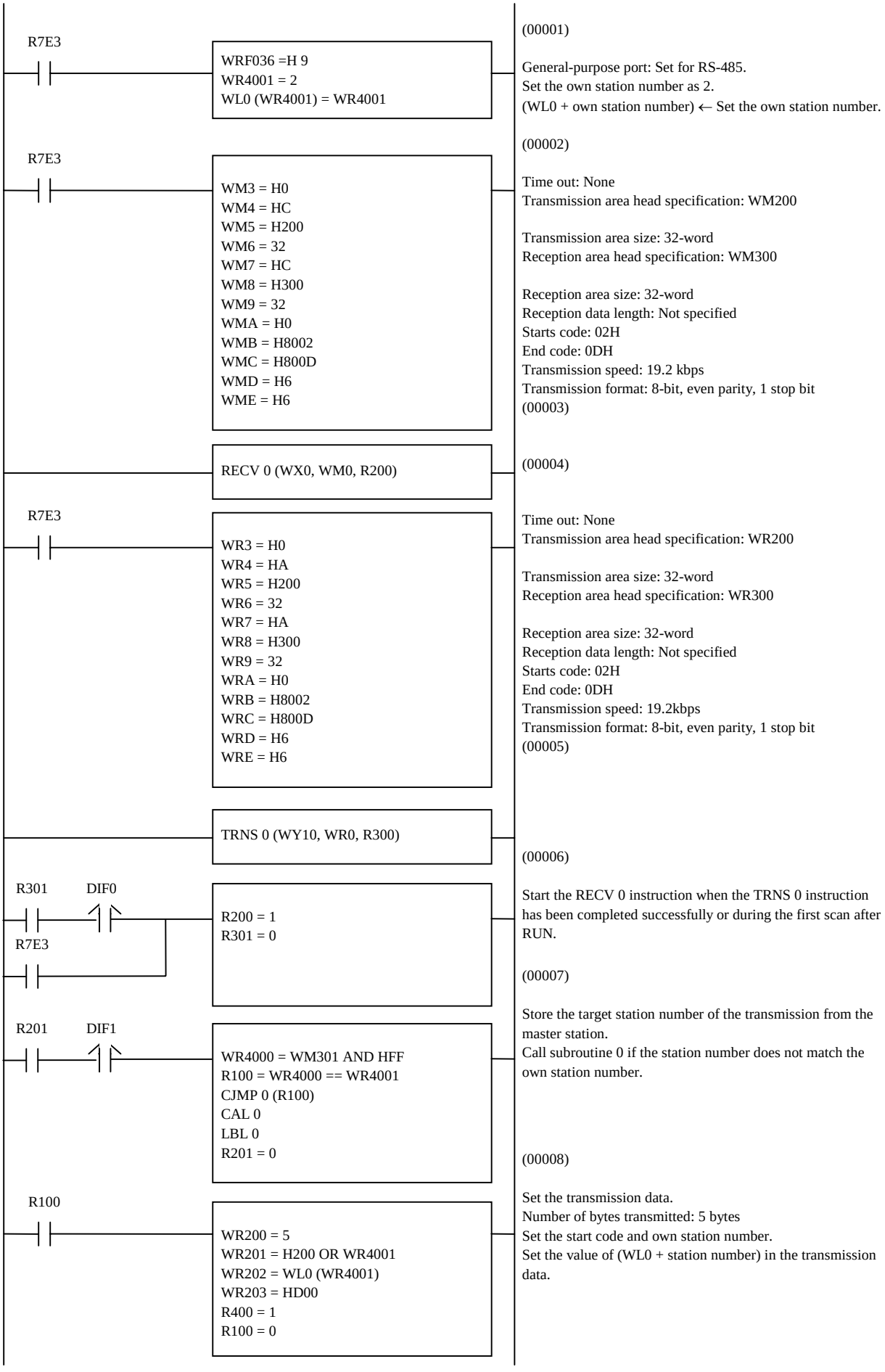
Address	Set value	Contents
WL0001	0001H	The data received from slave station 1.
WL0002	0002H	The data received from slave station 2.
WL0003	0003H	The data received from slave station 3.

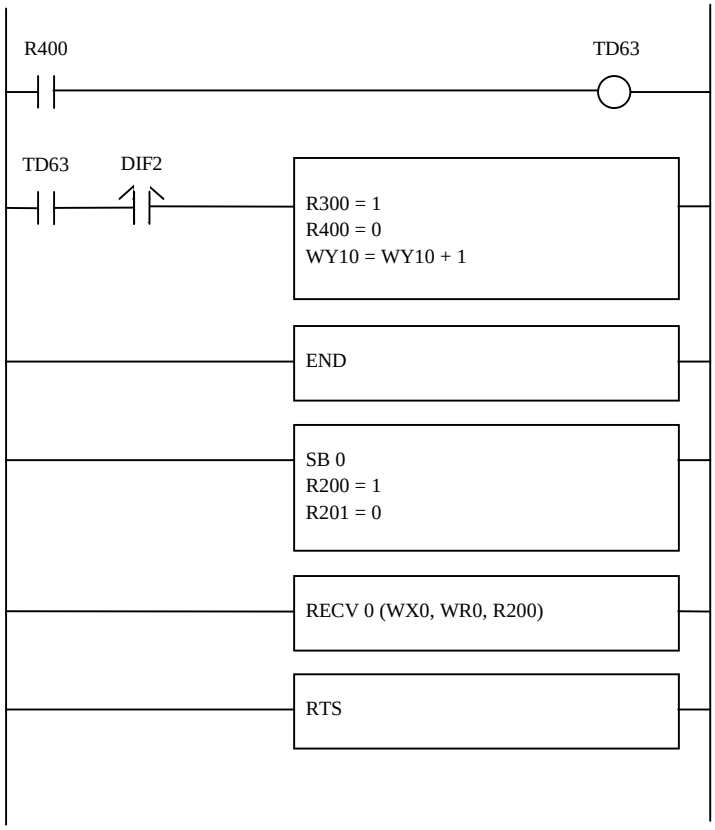
5] Program

(a) Program on the master station side (with three slave stations)



(b) Program on the slave station side (slave station number 2)



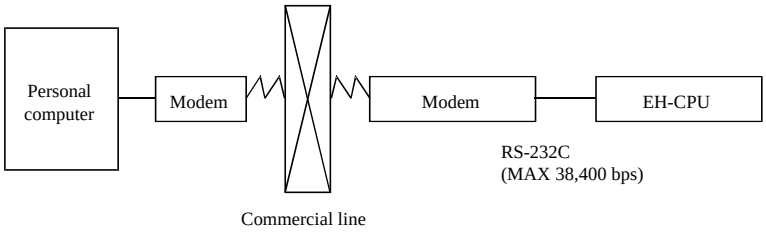

R400
TD63
(00009)
01s 2
TD63
DIF2
(00010)
Start the TRNS 0 instruction after 20 ms has elapsed.
R300 = 1
R400 = 0
WY10 = WY10 + 1
(00011)
END
(00012)
SB 0
Subroutine 0
R200 = 1
Turn on the reception execution flag.
R201 = 0
Turn off the reception normal completion flag.
(00013)
RECV 0 (WX0, WR0, R200)
(00014)
RTS

10.4 Modem Control Function

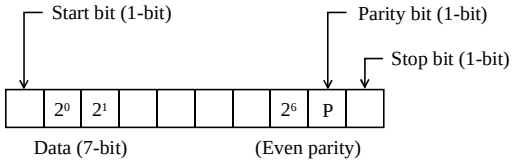
All CPU except CPU104(A) is equipped with a modem control function. The modem control function can be controlled using task codes.
To use this function, the mode setting switch must be set. For setting switches, refer to P4-6.

Connecting the two operating modems may be difficult if there is a large difference between them in communication speeds. Match to eliminate any difference in communication speeds.

10.4.1 Configuration



10.4.2 Connection specifications

Item	Specification
Transmission speed	2,400 bps, 4,800 bps, 9,600 bps, 19,200 bps, 38,400 bps * Communication speed between the modem and PLC depends on the setting for the special internal output WRF01A.
Communication system	Full duplex (communication program is half-duplex control)
Synchronization system	Start-stop synchronization
Transmission system	Serial transmission (bit serial transmission)
Transmission code	ASCII code
Transmission code configuration	
Transmission code outgoing sequence	Sent out from the lowest bit (2^0) in character units
Error detection	Vertical parity check, overrun check, framing check
Interface	Conforms to RS-232C
Control procedure	H-series dedicated procedure (high protocol)
Startup system	One-sided startup using the host side command

* Since ER signal cannot be controlled, be sure to use command or another I/O to disconnect line.

Table 10.11 List of port 1 signals when a modem is connected

Pin No.	Signal abbreviation	Direction		Meaning
		CPU	Modem	
1]	SG1	←	←	Signal ground
2]	CD1	←	←	Carrier receive in-progress notification signal Connected to CD in the modem.
3]	ER1	→	→	Communication enabled signal of the terminal
4]	ER2	→	→	Unused
5]	SD1	→	→	Data sent by the CPU Connected to SD in the modem.
6]	RD1	←	←	Data received by the CPU Connected to RD in the modem.
7]	DR1	←	←	Communication enabled signal of the modem Connected to DR in the modem.
8]	RS1	→	→	Transmission request signal Connected to RS in the modem.

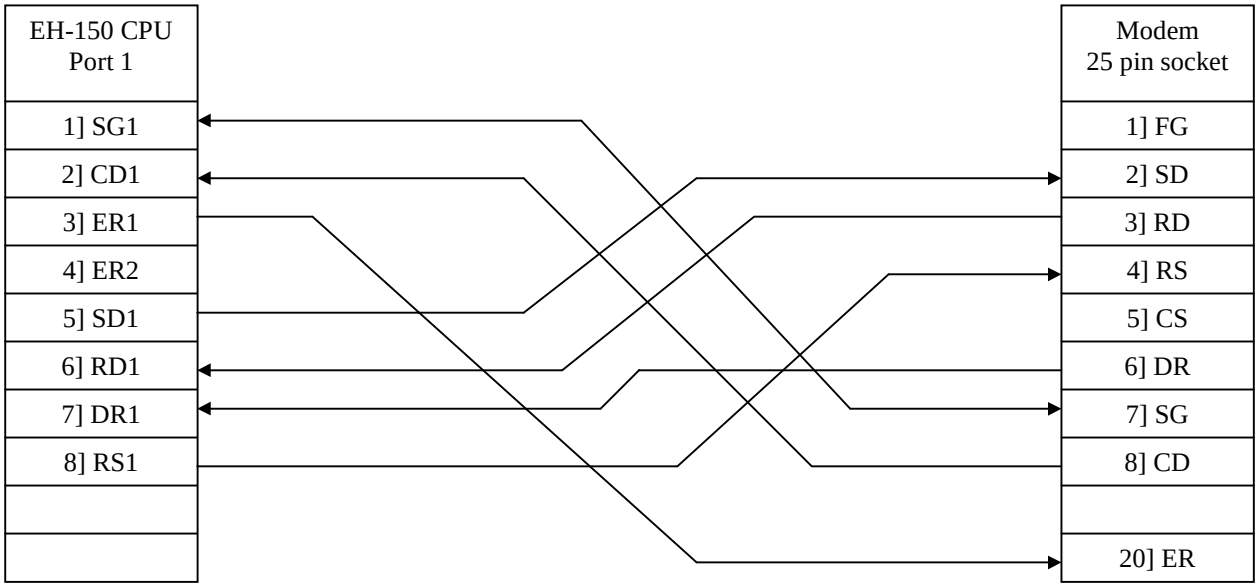


Figure 10.8 Cable connection between a modem (25 pin socket) and port 1

10.4.3 Additional task code

The existing task codes are supported, and ladder disconnection processing is supported as an additional function. The specification is the same as the current post communication (destination processing, ladder disconnection processing), except for the NCU control.

Line disconnection task code

Line disconnection request

H1C

Response (normal response only)

H00	H1C

10.4.4 AT commands

In AT commands, an instruction sent to the modem from the host is called a "command" and the character string in response to the "command" returned to the host from the modem is called a "result code".
AT commands always begin with the character string "AT" and a return code is input at the end of the command. However, A/ is excluded. The command that follows after the "AT" can have multiple inputs in a single line.

(1) Format

1] AT command format

A	T	command parameter command parameter · · · ·	CR

2] Result code format

CR	LF	result code (word)	CR	LF

Result code (number)	CR

(2) List of commands (extract)

1] AT commands

Command	Function overview		Example
AT	Automatically recognizes data format		—
A/	Re-executes the response directly preceding		—
ATDmm	Dial		ATD12345678
ATHn	Line ON/OFF	0: On hook (disconnect) 1: Off hook	ATH0 ATH1
ATPn	Pulse setting	0, 1: 10 pps 2: 20 pps	ATP0, ATP1 ATP2
ATT	Tone setting		ATT
ATSn = X	Sets S register value		ATS0 = 0
ATVn	Result code display format	0: Number 1: Word	ATV0 ATV1
AT&Cn	CD signal control	0: Always on 1: Depends on the carrier of counter-party modem	AT&C0 AT&C1
AT&Dn	ER signal control	0: Always on 2: Turning from on to off during communication disconnects line 3: Turning from on to off resets the software	AT&D0 AT&D2 AT&D3
AT&Sn	DR signal	0: Always on 1: Depends on sequence 2: Depends on CD signal	AT&S0 AT&S1 AT&S2
AT&Rn	RI (CI) signal control	0: Turns on from calling start until communication begins 1: Turns on from calling start until communication ends 2: Turns on/off in synchronization with the call signal	AT&R0 AT&R1 AT&R2

2] S register

S register	Set value	Function
S0	0 no automatic reception 1 to 255	Setting for automatic reception/reception ring count
S2	0 to 127 (43 [+])	Escape code setting
S3	0 to 127 (13 [CR])	CR code setting
S4	0 to 127 (10 [LF])	LF code setting

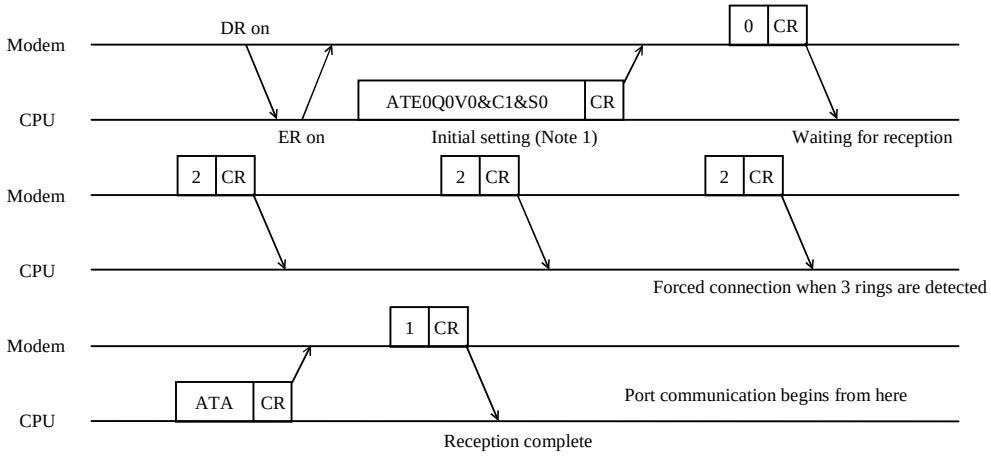
3] Result codes

Number format	Word format	Meaning
0	OK	Normal execution
1	CONNECT	Connection complete
2	RING	Reception detected
3	NO CARRIER	Line disconnected
4	ERROR	Command error
5	CONNECT 1200	1,200 bps connection
6	NO DIAL TONE	Cannot hear dial tone
7	BUSY	Busy signal detected
8	NO ANSWER	No tone heard
10	CONNECT 2400	2,400 bps connection
11	CONNECT 4800	4,800 bps connection
12	CONNECT 9600	9,600 bps connection
13	CONNECT 14400	14,400 bps connection

(3) Sequence

An example of a communication sequence using the Omron-made modem ME3314A is given below.

(a) Reception sequence

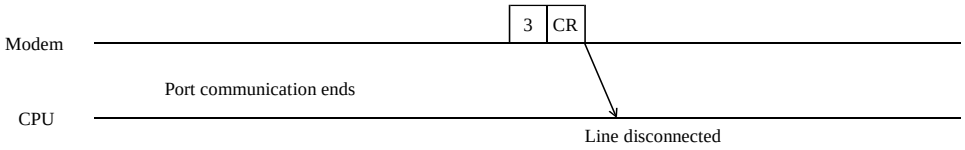


1] The PLC generates the AT command that performs the initial setting of the modem.

2] If initial setting is OK, the modem returns "0."

3] When the PLC is in reception wait status and detects the result code "2" three times, it connects the modem.

(b) Disconnect sequence



The PLC disconnects the line when the result code "3" is returned.

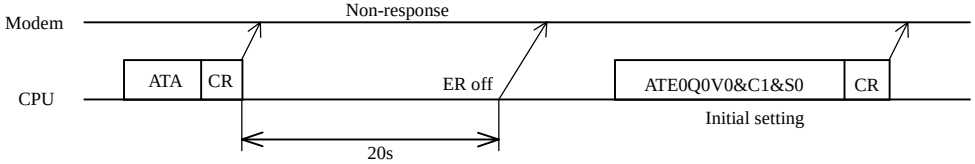
Note 1: Since the modem initial setup sets only minimal items from the PLC side, connect a personal computer and perform necessary settings before making the connection. (Set the DR signal to always on.)

Note 2: The CPU version showed by Table 10.12 support time out setting at special internal output (WRF03B). The default setting is “no time out”. Refer to the “Chapter 13 Special Internal Outputs” for details.

Table 10.12 CPU version

CPU type	CPU516/548	CPU448	CPU316A	CPU308A	CPU208A
ROM version (WRF050, WRF051)	All	C416	B205	A205	B105

Note: the time out of the former CPU version of Table 10.12 is fixed 20ms EH-CPU104/104A do not support modem control function.



Note

Special internal outputs(WRF01A,WRF03B,WRF03C) about modem connection are maintained by the battery during power failuer.It is not necessary to set up every time the power is turned ON. However, when a battery error occurs, please re-set up.

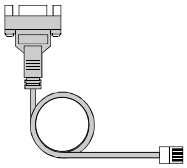
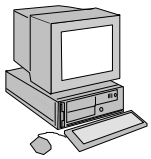
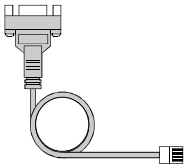
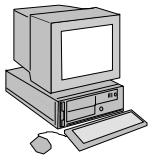
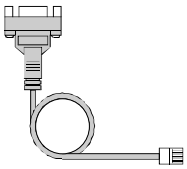
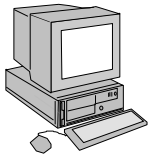
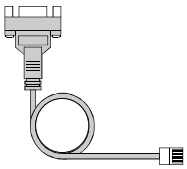
10.5 Port and Peripheral Unit Connection

Tables 10.13 and 10.14 show the settings of the mode setting switches and the connecting cables to connect a peripheral unit to ports 1 and 2 of the EH-150, respectively.

Table 10.13 Settings of the mode setting switches for connecting peripheral units

Condition		A	B	C	D	E	F
PORT	SW	Windows® LADDER EDITOR (DOS/V)	Windows® LADDER EDITOR (PC98)	DOS LADDER EDITOR (PC98)	DOS LADDER EDITOR (AT compatible)	LADDER EDITOR For GPCL / HILDRL	Programmer (PGM-GPH) (PGM-CHH)
1	8	OFF	OFF	OFF	OFF	Cannot connect	Cannot connect
	7	OFF	OFF	OFF	OFF		
	6	—	—	—	—		
	5	ON	ON	ON	ON		
	4	OFF	OFF	ON	ON		
	3	ON	ON	ON	ON		
	2	—	—	—	—		
	1	OFF	OFF	OFF	OFF		
	Toggle	—	—	—	—		
2	8	OFF	OFF	OFF	OFF	OFF	OFF
	7	OFF	OFF	OFF	OFF	OFF	OFF
	6	OFF	OFF	OFF	OFF	OFF	OFF
	5	—	—	—	—	—	—
	4	—	—	—	—	—	—
	3	—	—	—	—	—	—
	2	—	—	—	—	—	—
	1	OFF	OFF	OFF	OFF	OFF	OFF
	Toggle	ON	ON	OFF	OFF	OFF	ON
Software type		HLW-PC3	HLW-PC3	HL-PC3	HL-AT3E	HL-GPCL HILDRL	(Built-in programmer)
Cable type		WVCB02H + EH-RS05 / EH-VCB02	WPCB02H + EH-RS05	PCCB02H + EH-RS05	EH-VCB02	GPCB02H + EH-RS05	PGCB02H + EH-RS05

Table 10.14 Peripheral unit connection configuration

<table>
<tr><th>Peripheral unit</th><th colspan="2">Cable</th></tr>
<tr><td>GPCL01H
(LADDER EDITOR, HI-Ladder)
  </td><td>GPCB02H
  </td><td>EH-RS05
  </td></tr>
<tr><td>LADDER EDITOR
  </td><td>PCCB02H
  </td><td>EH-RS05
  </td></tr>
<tr><td rowspan="2">LADDER EDITOR for Windows®
  </td><td>WPCB02H
 
WVCB02H
  </td><td>EH-RS05
  </td></tr>
<tr><td colspan="2">EH-VCB02
  </td></tr>
<tr><td rowspan="2">Pro-H
  </td><td>WVCB02H
  </td><td>EH-RS05
  </td></tr>
<tr><td colspan="2">EH-VCB02
  </td></tr>
</table>

10.6 Connection method for RS-422/485 communication

Communication can be performed with an interface of RS-422/485 at the port 1 of the EH-150. Using the H series' dedicated control procedure (high protocol) or general port commands (TRNS 0 and RECV 0), communication of 1:N stations can be performed. Figures 10.9 and 10.10 show examples when a connection is made for 1:N stations at the port 1. Note that, for the connection for communication in 1:1 mode, only the first EH-150 in these figures is connected.

(1) In case of RS-422

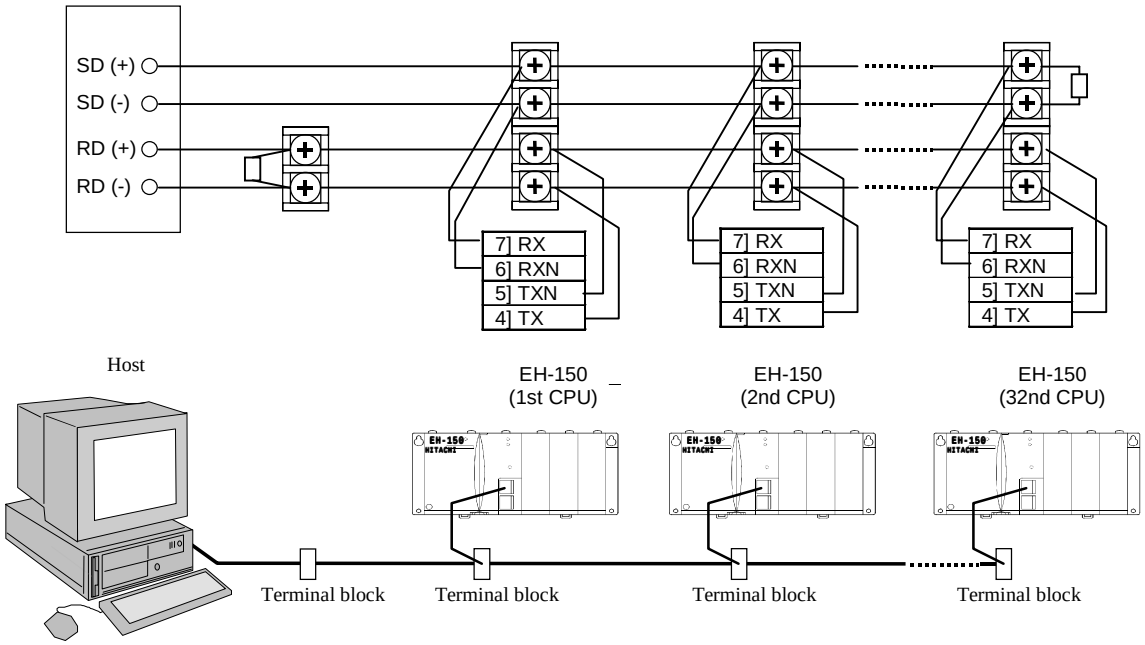


Figure 10.9 Connection for 1:n station communication by RS-422

(2) In case of RS-485

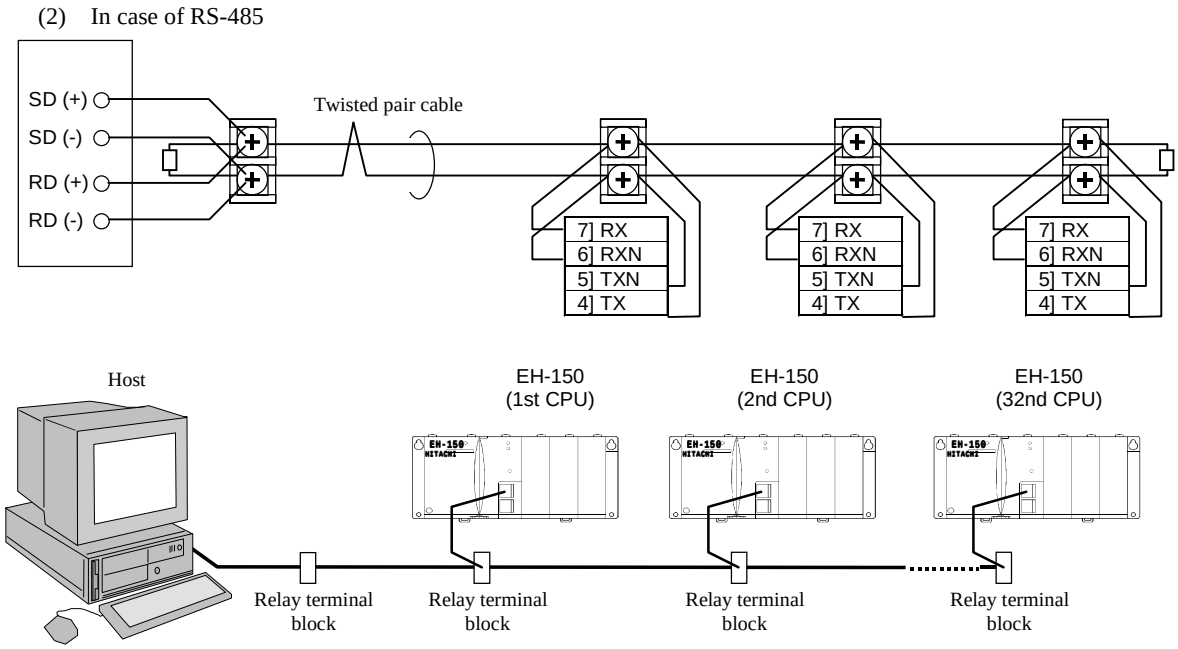


Figure 10.10 Connection for 1:n station communication by RS-485

Chapter 11 Real Time Clock and Memory Board

11.1 Real Time Clock Function

The EH-CPU208/308/316/448 includes a function that manages the time and date of PLC data. The clock function can be operated with either special internal output or a task code.
Clock data is retained by battery power. Note that the battery is not connected when the module is shipped. When using the clock function, connect the battery and set the clock data.
Clock function is not supported by EH-CPU104(A).

11.1.1 Operation using a special internal output

(1) Reading clock data

By turning the read request (R7F8) on, the clock data at the time of the request is stored in the read value area (WRF01B to WRF01F).

(2) Clock data setting

The clock data setting is done by first set the data to store in the set value area (WRF01B to WRF01F), and then turning on the setting request (R7F9). At this time, if there is an error in the set value, the setting data error (R7FB) turns on. When the setting request (R7F9) turns off and the setting data error (R7FB) is off, setting is complete.

(3) Clock data ±30 seconds adjustment

By turning on the ±30 second adjustment request (R7FA), the seconds value will be set as follows depending on the value at this time.

- When the digit for seconds is 00 to 29: the digit becomes 00
- When the digit for seconds is 30 to 59: the time becomes +1 minute and the digit for seconds becomes 00.

(4) Defining special internal output

- Operation bit

Item	I/O number	Name	Function
1	R7F8	Request to read calendar and clock data	Calendar and clock data is read out to WRF01B-F01F.
2	R7F9	Request to write calendar and clock data	Calendar and clock data in WRF01B-F01F is written to the current data in WRF00B-F00F.
3	R7FA	Clock ± 30 seconds adjustment request	Sets the second digits of the RTC to 00.
4	R7FB	Calendar and clock setting data error	Turns on when the setting data is abnormal.

- Current data monitor area : Current data of the clock given always (all BCD data).

Item	I/O number	Name	Description
1	WRF00B	Year	4-digit year [yyyy]
2	WRF00C	Month/day	[mmdd]
3	WRF00D	Day of the week	0 to 6 : Sunday to Saturday
4	WRF00E	Hour	[hhmm] (24-hour system).
5	WRF00F	Seconds	[00ss]

- Reading/writing area : Clock data to be read or written (all BCD data).

Item	I/O number	Name	Description
1	WRF01B	Year	4-digit year [yyyy]
2	WRF01C	Month/day	[mmdd]
3	WRF01D	Day of the week	0 to 6 : Sunday to Saturday
4	WRF01E	Hour	[hhmm] (24-hour system).
5	WRF01F	Seconds	[00ss]

Note 1: Day of the week data is as follows (3 upper digits are always 000):
0 - Sunday, 1 - Monday, 2 - Tuesday, 3- Wednesday, 4- Thursday, 5- Friday, 6 - Saturday

11.1.2 Operation using task codes

The following can be performed with the clock setting and reading task codes.

- Read clock data
- Set clock data
- Clock data ±30 seconds adjustment

Refer to the task code list (Appendix 3) for more detail on task codes.

11.2 Memory Board Function

The EH-CPU308(A)/316(A)/448/516/548 have the program transfer function for transferring and comparing programs between the CPU and either the EH-MEMP or EH-MEMD memory board, and the data logging function via the FUN command from user programs using the EH-MEMD memory board. Table 11.1 shows the DIP switch settings of the EH-MEMP and EH-MEMD memory boards for switching to each mode.
* The EH-CPU104/104A/208/208A do not support the memory board function.

Table 11.1 Operation modes settings with dip switch of memory board

Function	SW4	SW3	SW2	SW1	Operation details	Remarks
Program transfer					Starts up in the normal operating mode (Invalid memory board)	
				ON	From memory board to CPU (Program transfer)	
			ON		From CPU to memory board (Program transfer)	
			ON	ON	Between memory board and CPU (Check program)	
		ON			Starts up in the normal operating mode (Invalid memory board)	*
		ON		ON	From memory board to CPU (Program transfer)	*
		ON	ON		Dip switch setting error	*
		ON	ON	ON	Between memory board and CPU (Check program)	*
Data logging	ON		Invalid		Starts up in the normal mode (Data logging command can be executed)	
	ON	ON	Invalid		Starts up in the normal mode (Data logging command can be executed)	*

*: If the SW3 is ON, writing to the memory board is prohibited.
In the case of program transfer mode, the dip switch setting becomes effective only after turning on the power. In the case of data logging mode, SW3 becomes effective in real time. (However, if the setting is changed while a command is being executed, the new setting becomes effective upon execution of the next command.)

11.2.1 Program transfer function

The EH-CPU308/308A/316/316A/448 have the program transfer function for transferring and comparing programs between the CPU and either the EH-MEMP or EH-MEMD memory board by setting the DIP switches of the memory board in advance and turning on the power. Since the CPU starts up in a special mode, the CPU will not switch to the RUN mode in this case. Table 11.2 shows the operation modes that can be executed between each CPU model and the EH-MEMP/EH-MEMD memory board. As shown in Table 11.2, the maximum program size that can be transferred to the EH-MEMD memory board is 16 k steps. Thus, to transfer a program of larger than 16 k steps and a maximum of 48 k steps in the EH-CPU448, use the EH-MEMP memory board.

Table 11.2 EH-MEMP/MEMD function

Item	EH-MEMP	EH-MEMD
User program size	Max. 48 k step	Max. 16 k step
Data memory size	-	384 k words
Program copy (CPU→MEMP/D)	✓	✓
Program copy (CPU←MEMP/D)	✓	✓
Program verify (CPU←→MEMP/D)	✓	✓
Data logging	-	✓

Table 11.3 shows the LED displays when an error occurs during the execution of the program transfer function.

Table 11.3 CPU LED indications in the program transfer mode

Operation mode	Memory board → CPU		Memory board ← CPU		Memory board = CPU	
	RUN	ERR	RUN	ERR	RUN	ERR
Dip switch setting error	—	—	●	◕	—	—
Sum error in memory board	●	○	—	—	●	○
Sum error in CPU memory	—	—	◑	○	◑	○
Memory size error	◑ *1	◑ *1	—	—	—	—
Transfer and checking in progress	◑	●	◑	●	◑	●
FLASH error (inside memory board) *4	—	—	○	◑	—	—
Unmatched check result	●	◔	●	◔	●	◔
Normal completion	○→●	●	○→●	●	○→●	●

○: ON ●: OFF ◕: Flashing (1 s ON, 1 s OFF) ◑: Flashing (500 ms ON, 500 ms OFF)
◔: Flashing (250 ms ON, 250 ms OFF) ○→●: 3 s OFF after ON
*1: RUN and LED of ERR flash at the same time
*2: Erase error and write error of FLASH

11.2.2 Logging Function

The EH-CPU 308(A)/316(A)/448(A)/516/548 provides the function for log data using the EH-MEMD.
* This function cannot be used with the EH-MEMP.

(1) Log commands

There are four types of log commands, as shown below.

FUN 210 (s) * (LOGIT (s))	Initial setting for data logging
FUN 211 (s) * (LOGWRT (s))	Log data write
FUN 212 (s) * (LOGCLR (s))	Log data clear
FUN 213 (s) * (LOGRED (s))	Log data read

* ( ) indicates the display when the LADDER EDITOR is used.

(2) Overview of logging function

The logging function is used to transfer a group of specified data groups (word internal output/external input) to the log data area (memory board) as a single log, or read a group of data stored in the log data area.

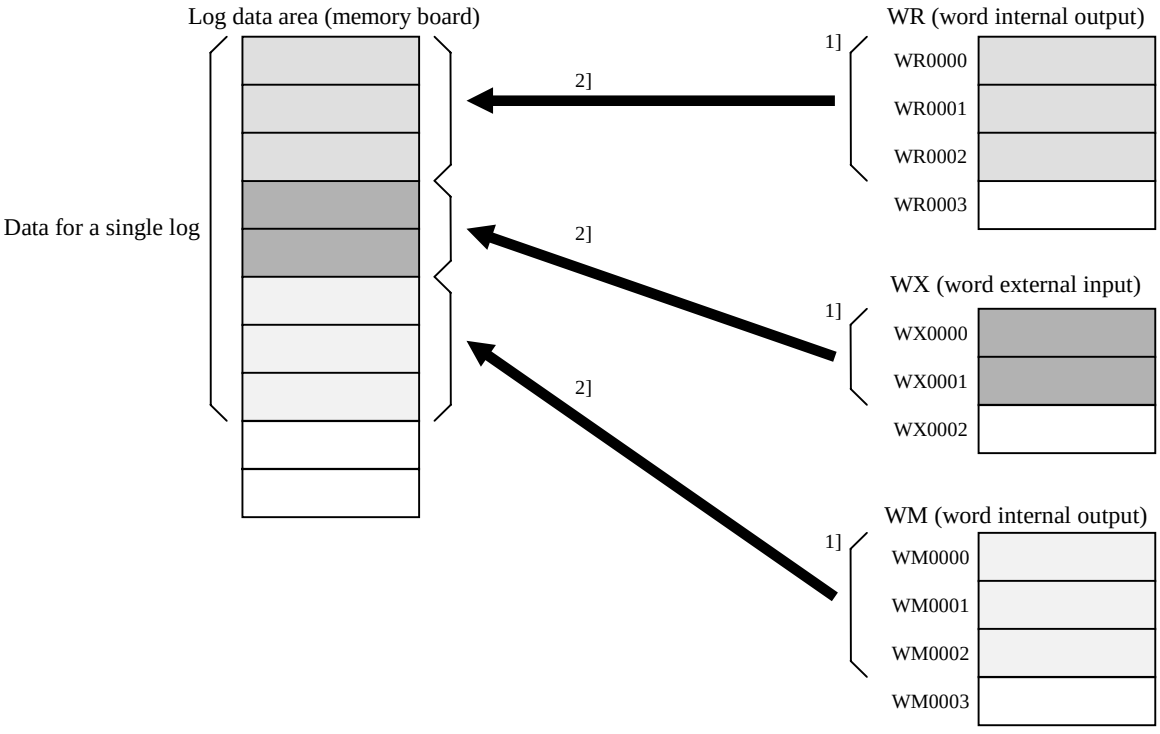
Use example:

1] FUN 210 (Initial setting for data logging)

Sets a total of 8 words from three groups of the log data including WR0000 through WR0002 (3 words), WX0000 and WX0001 (2 words), and WM0000 through WM0002 (3 words), as a data for a single log.

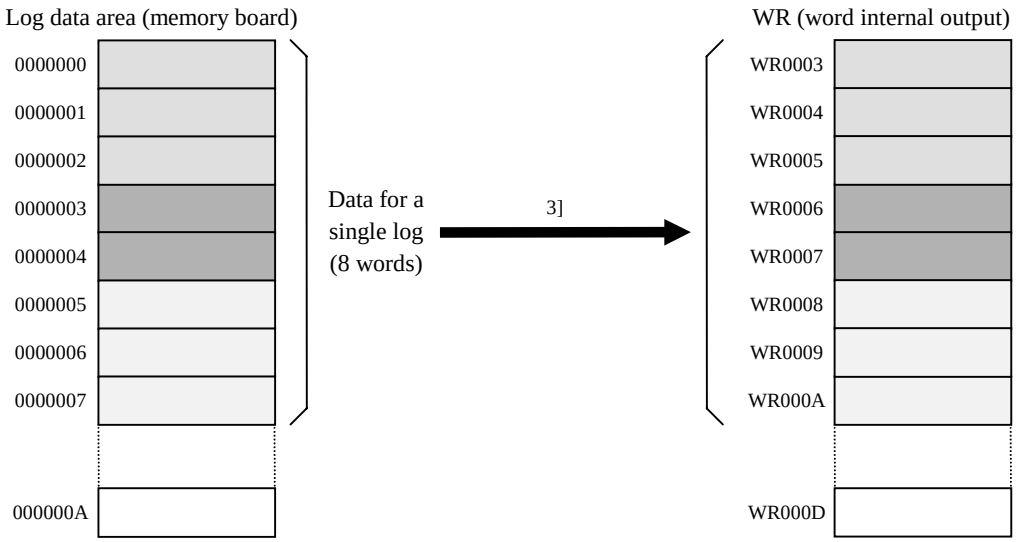
2] FUN 211 (log data write)

Writes to the log data area (memory board) the log data for a single log specified in the initial setting in 1].



3] FUN 213 (log data read)

Reads to the specified destination after WR0003 the contents of the log data area written in 2] by the amount of data for a single log (8 words) starting from the head of the log data area.



4] FUN 212 (log data clear)

Clears the contents of the log data area in the memory board.

(3) Applicable log area

The table below shows the I/O areas that can be used for logging and their allowable ranges.

	Symbol	Allowable range		
		EH-CPU308/308A	EH-CPU316/316A	EH-CPU448
Word external input	WX	16 words		Cannot be used
CPU link area 1	WL	WL0 to WL3FF		
CPU link area 2	WL	WL1000 to WL13FF		
Word internal output	WR	WR0 to WR43FF	WR0 to WR57FF	WR0 to WRC3FF
	WM	WM0 to WM3FFF		

* If an area outside the corresponding range is specified, the DER (R7F4) bit is turned ON and the processing will not be performed.

(4) Log data size and write interval

- Size of log data area
  384 K (393,216) words
- Size of log data
  Data between 1 and 128 words can be specified for a single log.
- Write interval
  The table below shows the number of words in a single log and the corresponding logging interval.

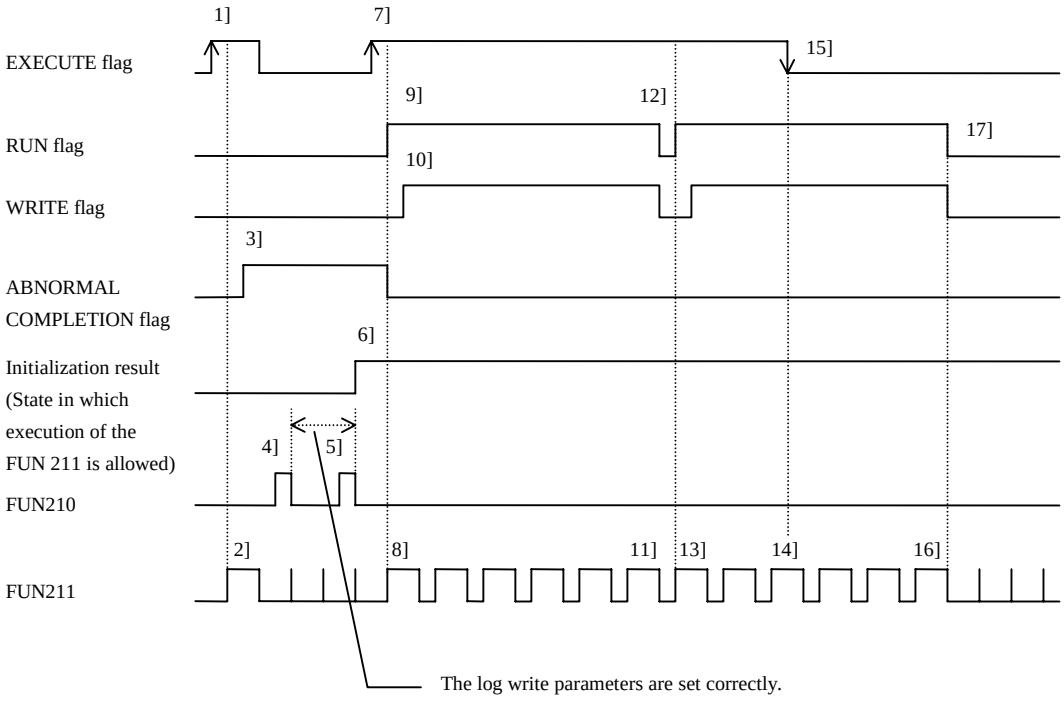
	Minimum	Intermediate	Maximum
Log data size (words/log)	1 word per log	n words per log	128 words per log
Minimum write interval (ms)	Approx. 10 ms	Approx. (5n + 5) ms	Approx. 645 ms

* In the above table, n indicates an integer between 1 and 128.

No new data can be logged in between the write intervals (ms) shown above.

(5) Log write (FUN 211) time chart

The diagram below shows the relationship between the log write control bit table (EXECUTE flag, RUN flag, WRITE flag, ABNORMAL COMPLETION flag) and the execution of the FUN 211 (log write).



The EXECUTE flag rises 1]. However, since the initialization result indicates that execution of the FUN 211 is not allowed, the FUN 211 turns on 2] the ABNROMAL COMPLETION flag 3].

A log write parameter error is found during the processing of the FUN 210 4]. As a result, the initialization result does not indicate that execution of the FUN 211 is allowed.

The write parameters are correctly set 5]. As a result, the initialization result indicates that execution of the FUN 211 is allowed 6].

The FUN 211 detects 8] the rise of the EXECUTE flag 7] and checks if the execution is allowed. Since the check result is normal, the RUN flag is turned on 9] and the log data is stored temporarily. When all data has been stored, the WRITE flag is turned on 10] and the log data is written to the memory board (starting of write).

When the writing of data for the first log is completed 11], the RUN flag and WRITE flag are turned off 12].

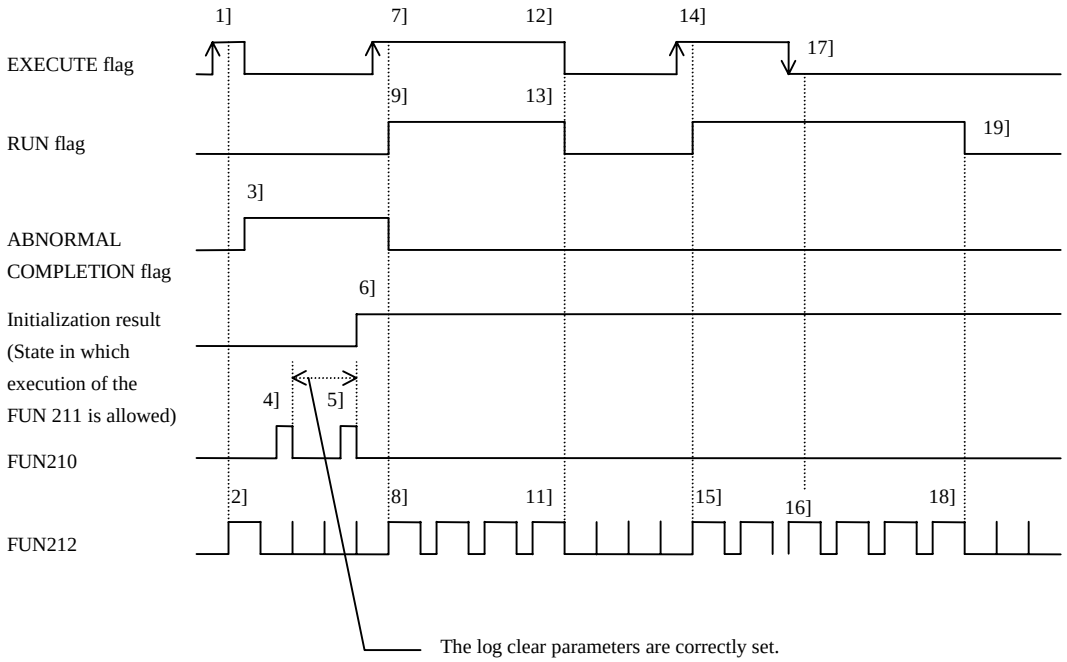
Since the EXECUTE flag is still on, writing of the second log is started 13].

The FUN 211 detects 14] the fall of the EXECUTE flag 15]. However, the processing continues since the writing of data for the second log is not completed yet.

When the writing of data for the second log is completed 16], the RUN flag and WRITE flag are turned off 17].

(6) Log clear (FUN 212) time chart

The diagram below shows the relationship between the log clear parameter bit table (EXECUTE flag, RUN flag, ABNORMAL COMPLETION flag) and the execution of the FUN 211 (log clear).



The EXECUTE flag rises 1]. However, since the initialization result indicates that execution of the FUN 211 is not allowed, the FUN 211 turns on 2] the ABNROMAL COMPLETION flag 3].

A log clear parameter error is found during the processing of the FUN 210 4]. As a result, the initialization result does not indicate that execution of the FUN 211 is allowed.

The log clear parameters are correctly set 5]. As a result, the initialization result indicates that execution of the FUN 211 is allowed 6].

The FUN 212 detects 8] the rise of the EXECUTE flag 7] and checks if the execution is allowed. Since the check result is normal, the RUN flag is turned on 9] and the log data is cleared.

When the log data clear is completed 11], the EXECUTE flag 12] and the RUN flag 13] are turned off.

The FUN 212 detects 15] the rise of the EXECUTE flag 14] and starts the second data clear.

The FUN 212 detects 16] the fall of the EXECUTE flag 17]. However, since the second data clear is not completed yet, the processing continues.

When the second data clear is completed, the FUN 212 turns off 18] the RUN flag 19].

(7) Log error details

When started in the logging mode, the system checks the log data upon power ON. If the system detects an error during the check, it sets to the special internal output R7FE or WRF07F an applicable error code from among the following error codes, and then executes the program shown below. At the same time, the CPU error code “A0” is set to the special internal output WRF000. The table below shows the description of each error.

CPU error code	Special internal output		Error code	Description of error
A0	R7FE = 1	WRF07F	H0001	Different memory board from the one previously installed
			H0002 to H0009	Invalid log data
			H00FF	Memory board cannot be accessed

(8) Corrective actions to be taken upon detection of log error

When the system detects any of the log errors described in (7), take appropriate actions as shown below.

When H0001 occurred:
Reinstall the memory board previously installed, and restart the system.
If a new memory board is installed, clear the log data using the log data clear command (FUN 212).

When any of the errors between H0002 and H0009 occurred:

Clear the log data using the log data clear command (FUN 212), and then restart the system.

When H00FF occurred:

Check if the memory board is properly installed, and then restart the system. If the error persists, there may be a hardware error in the memory board.

MEMO

Chapter 12 Error Code List

12.1 Error Codes

The table below indicates the self-diagnostic error codes. (Refer to Chapter 15 Troubleshooting for corrective action.) Error codes are output as a hexadecimal to the special internal output WRF000. (This special internal output is saved during a power failure, and is retained even when the cause of the error disappears. Also, when more than one error is generated, the one that is classified with the greatest severity is stored.)

Note: LED example

○ : ON ● : OFF ◕ : Flashing (1 s ON, 1 s OFF) ◑ : Flashing (500 ms ON, 500 ms OFF) ◔ : Flashing (250 ms ON, 250 ms OFF)

Error code	Error name [detection timing]	Classification	Nature of error	RUN LED	ERR LED	Operation	Related special internal output	
							Bit	Word
11	System ROM error [when power turns on]	Serious failure	The system ROM has a sum error or cannot be read	●	○	Stops	R7DB	—
12	System RAM error [when power turns on]	Serious failure	System RAM cannot be read and written properly	●	○	Stops	R7DB	—
13	Micro processor error [always checking]	Serious failure	Address error interrupt, undefined command interrupt occurred in the micro computer	●	○	Stops	R7C8	—
15	System bus time out error [when bus is accessed]	Serious failure	Micro computer detects a system bus time out	●	○	Stops	R7D8	—
16	System program abnormal [always checking]	Serious failure	System program in FLASH memory has a sum error	●	○	Stops	R7C8	—
—	Power off Power error [always checking]	Serious failure	No power supply from the power module	●	●	Stops	—	—
—	Microcomputer overload error [Constantly checked]	Serious failure	The watchdog timer detected a microcomputer overload because the microcomputer did not operate according to program (the microcomputer is not functioning correctly).	— *1	○	Stops	—	—
22	Sequence processor error	Medium failure	The microcomputer detected a sequence processor did not operate. (the sequence processor is not functioning correctly).	●	○	Stops	R7DB	—
23	Undefined code [check during operation]	Medium failure	Unknown code is found by microprocessor.	◔	○	Stops	R7C9	—
27	Data memory error [when power ON, at initialization]	Medium failure	Data memory cannot be read/written properly.	●	○	Stops	—	—
31	User memory abnormal [when power turns on, when RUN starts, when parameters are changed, at initialization]	Medium failure	A sum error is detected in user memory or RUNNING memory	◔	○	Stops	R7CA	—
33	User memory size error [when RUN starts]	Medium failure	User program capacity set by the parameter is larger than actual user memory capacity	◔	○	Stops	R7CC	—
34	Grammar/assemble error [when RUN starts, online change in RUN]	Medium failure	There is a grammar error in the user program	◑	○	Stops	R7D4	WRF001
41	I/O information verify error [always checking]	Minor failure	I/O assignment information and actual loading of module do not match	◑	○	Stops *2	R7CD	WRF002

*1: Condition prior to error occurrence is restored. (If operation was in progress, the RUN LED turns ON; if operation was stopped, the RUN LED turns off)

*2: Depending on the run parameter setting, operation can continue even when an abnormal occurs.

Error code	Error name [detection timing]	Classification	Nature of error	RUN LED	ERR LED	Operation	Related special internal output	
							Bit	Word
43	Remote abnormal [always checking]	Minor failure	• I/O assignment verify mismatch occurs in the remote slave station module • Communication error occurs between the remote master station module and CPU • There is an error in the remote master station module, and transmission to the slave station has stopped	◑	○	Stops *2	R7D0	WRF006 WRF080 to WRF0DF
44	Overload error (Normal scan) [during operation]	Minor failure	Execution time for normal scan exceeded the overload check time set by the parameter	◑	○	Stops *2	R7D1	—
45	Overload error (periodic scan) [periodic processing]	Minor failure	Execution time for periodic scan exceeded the execution period	◑	○	Stops *2	R7D2	—
47	I/O assignment points over [when power turns on, when RUN starts, during RUN, when parameters changed]	Minor failure	Slot number exceeds the possible range.	◑	○	Stops *2	R7D6	—
51	I/O module abnormal [always checking]	Warning	• High function module, communication module faults out	— *1	●	Runs	R7D5	WRF005 WRF080 to WRF0DF
52	I/O transmission error [when high function module is transmitting, when transfer command is being executed]	Warning	• Error occurs during high function module transmission • TRNS, RECV command parameter error	— *1	●	Runs	—	—
53	I/O inappropriate assignment [check during operation]	Warning	Assignment of communication module is incorrect	— *1	●	Runs	—	—
54	Communication module abnormal [always checking]	Warning	Communication module hardware abnormal	— *1	●	Runs	R7D7	WRF004
55	Communication module transmission error [when a peripheral device is connected to the communication module]	Warning	Error occurred during transmission to the communication module	— *1	●	Runs	R7D7	WRF004
57	Communication module I/O assignment over [always checking]	Warning	The number of communication module assignments exceeds the maximum	— *1	●	Runs	R7DD	—
58	Communication module I/O verify error [always checking]	Warning	• Mismatch between communication module assignment information and module loading • Communication module hardware error	— *1	●	Runs	R7CE	WRF003
59	Link module abnormal [always checking]	Warning	• Link module hardware abnormal • Link parameter abnormal	— *1	●	Runs	R7DE	WRF007 WRF0E0 to WRF19F
61	Port 2 transmission error (parity) [when transmitting]	Warning	Parity error detected during transmission	— *1	●	Runs	—	—
62	Port 2 transmission error (framing/overrun) [when transmitting]	Warning	Framing error or overrun error detected during transmission	— *1	●	Runs	—	—
63	Port 2 transmission error (time out) [when transmitting]	Warning	Time out error detected during transmission	— *1	●	Runs	—	—

*1: Condition prior to error occurrence is restored. (If operation was in progress, the RUN LED turns ON; if operation was stopped, the RUN LED turns off)

*2: Depending on the run parameter setting, operation can continue even when an abnormal occurs.

Error code	Error name [detection timing]	Classification	Nature of error	RUN LED	ERR LED	Operation	Related special internal output	
							Bit	Word
64	Port 2 transmission error (protocol error) [when transmitting]	Warning	Protocol (transmission procedure) error detected during transmission	— *1	●	Runs	—	—
65	Port 2 transmission error (BCC error) [when transmitting]	Warning	Sum error detected during transmission	— *1	●	Runs	—	—
67	Port 1 transmission error (parity) [when transmitting]	Warning	Parity error detected during transmission	— *1	●	Runs	—	—
68	Port 1 transmission error (framing/overrun) [when transmitting]	Warning	Framing error or overrun error detected during transmission	— *1	●	Runs	—	—
69	Port 1 transmission error (time out) [when transmitting]	Warning	Time out error detected during transmission	— *1	●	Runs	—	—
6A	Port 1 transmission error (protocol error) [when transmitting]	Warning	Protocol (transmission procedure) error detected during transmission	— *1	●	Runs	—	—
6B	Port 1 transmission error (BCC error) [when transmitting]	Warning	Sum error detected during transmission	— *1	●	Runs	—	—
71	Battery error [always checking]	Warning	• Battery voltage dropped below specified value • Battery not installed	— *1	◔	Runs	R7D9	—
	FLASH memory error (CPU448(A)/516/548) *3 [when program writing is executed]	Warning	Data cannot be written to the backup memory.	— *1	◔	Runs	R7D9	
94	Port 1 No modem response (when modem is connected)	Warning	There is no response from the modem connected to the port.	— *1	●	Runs	—	—
95	Port 1 Dial busy (when modem is connected)	Warning	The modem of the opposite party is communicating with another terminal or not ready for communication.	— *1	●	Runs	—	—
96	Port 1 CS ON time out (when modem is connected)	Warning	The CS signal of the modem connected to the port does not turn on within the designated time.	— *1	●	Runs	—	—
97	Port 1 Modem time out *4 (when modem is connected)	Warning	The modem connected to the port does not respond the complete of connecttion within the setting time (WRF03B).	— *1	●	Runs	—	—
A0	Data logging error	Warning	Detected an error in the memory board or log data.	— *1	●	Runs	R7FE	WRF07F

*1: Condition prior to error occurrence is restored. (If operation was in progress, the RUN LED turns ON; if operation was stopped, the RUN LED turns off)

*2: Depending on the run parameter setting, operation can continue even when an abnormal occurs.

*3: When the CPU fails in writing to the backup memory, EH-CPU448 does the treatment which is the same as the battery error. A backup memory error occurred to the CPU module if the same error after the battery exchange as well seems to occur. In this case, be careful because your program and so on isn't stored.

*4: This error code is supported by only CPU448 (Software version C416 later), EH-CPU316A (Software version B205 later), EH-CPU308A (Software version A205 later) and EH-CPU208A (Software version B105 later), CPU516/548.

How to clear the error code

Set 1 in the special internal output R7EC to clear the error code.
It is cleared also by pressing the reset switch for retentive area (R.CL) on the front of the CPU while CPU is in stop mode. Please note that if the reset switch pressed, all the retentive area is cleared.

12.2 Grammar and Assemble Error Codes

Descriptions of the grammar/assemble error codes are given below. The error codes are output as a hexadecimal to the internal output WRF001.

Error code	Error item	Description of error	Corrective action
H0001	Double definition of LBL	There are 2 or more LBL commands with the same number in the program	Limit the LBL command that has 2 or more of the same number to 1.
H0002	Double definition of FOR	There are 2 or more FOR commands with the same number in the program	Limit the FOR command that has 2 or more of the same number to 1.
H0003	Double definition of NEXT	There are 2 or more NEXT commands with the same number in the program	Limit the NEXT command that has 2 or more of the same number to 1.
H0004	Double definition of SB	There are 2 or more SB commands with the same number in the program	Limit the SB command that has 2 or more of the same number to 1.
H0005	Double definition of INT	There are 2 or more INT commands with the same number in the program	Limit the INT command that has 2 or more of the same number to 1.
H0010	END undefined	There is no END command prior to the INT, SB commands	Define the END command before the INT or SB command.
H0011	RTS undefined	There is no RTS command corresponding to the SB command	Define the RTS command after the SB command.
H0012	RTI undefined	There is no RTI command corresponding to the INT command	Define the RTI command after the INT command.
H0013	SB undefined	There is no SB command corresponding to the RTS command	Define the SB command before the RTS command.
H0014	INT undefined	There is no INT command corresponding to the RTI command	Define the INT command before the RTI command.
H0020	RTS area error	There is an RTS command in the normal scan area or periodic scan program area	Define the RTS command within the subroutine area.
H0021	RTI area error	There is an RTI command in the normal scan area or subroutine program area	Define the RTI command within the interrupt scan area.
H0022	END area error	There is an END command in the periodic scan area or subroutine program area	Define the END command at the end of the normal scan area.
H0023	CEND area error	There is a CEND command in the periodic scan area or subroutine program area	Define the CEND command within the normal scan area.
H0030	RTS start condition error	There is a start condition in the processing box that contains an RTS command	Delete the startup condition of the processing box.
H0031	RTI start condition error	There is a start condition in the processing box that contains an RTI command	Delete the startup condition of the processing box.
H0032	END start condition error	There is a start condition in the processing box that contains an END command	Delete the startup condition of the processing box.

12.3 Operation Error Codes

If an error occurs when a control command is executed, “1” is set in the operation error (ERR) special internal output “R7F3” and an error code indicating the nature of the error is set in WRF015.
To do a zero clear of the operation error, execute “R7F3=0” using forced setting from the program or peripheral device.
To do a zero clear of the error code, execute “WRF015=0” using forced setting from the program or peripheral device.

Error code	Error name	Description of error	Originating command
H0013	SB undefined	SBn command corresponding to the command number n in the CALn command is not programmed	CAL
H0015	LBL undefined	LBLn command corresponding to the command number n in the JMPn, CJMPn commands is not programmed	JMP CJMP RSRV
H0016	FOR undefined	FORn command corresponding to the command number n in the NEXTn command is not programmed	NEXT
H0017	NEXT undefined	NEXTn command corresponding to the command number n in the FORn command is not programmed	FOR
H0040	LBL area error	LBLn command corresponding to the command number n in the JMPn, CJMPn commands is not programmed in the same program area	JMP CJMP RSRV
H0041	CAL nesting overflow	There are more than 6 levels of subroutine nesting	CAL
H0042	CAL undefined	RTS command was executed without executing a CAL command	RTS
H0043	FOR to NEXT error	There is a NEXTn with the same command number n prior to the FORn command	FOR
H0044	NEXT area error	There is no NEXTn command with the same command number n as the FORn command, in the same program area	FOR
H0045	FOR to NEXT nesting overflow	The FORn and NEXTn commands are not nested	FOR
H0046	FOR nesting overflow	There are more than 6 nesting levels of FOR to NEXT	FOR NEXT

MEMO

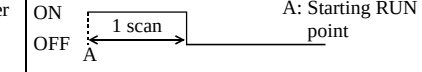
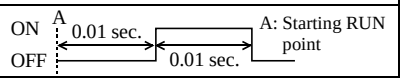
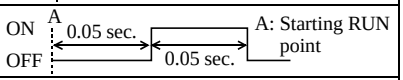
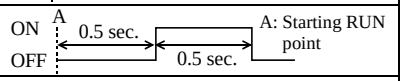
Chapter 13 Special Internal Outputs

The EH-150 has a special internal output area for performing status display and various other settings. The special internal output area is always kept as power failure memory.

13.1 Bit Special Internal Output Area

Definitions of the bit special internal output area (R7C0 to R7FF) are given below.

No.	Name	Meaning	Description	Setting condition	Resetting condition
R7C0	Continue when overload error occurs (normal scan)	0: Stop running when overload error occurs 1: Continue running when overload error occurs	Designates continue/stop running when a normal scan overload error occurs	Set by user	Cleared by user or cleared when retentive area cleared.
R7C1	Continue when overload error occurs (periodic scan)	0: Stop running when overload error occurs 1: Continue running when overload error occurs	Designates continue/stop running when a periodic-scan overload error occurs		
R7C2	Continue when overload error occurs (interrupt scan)	0: Stop running when overload error occurs 1: Continue running when overload error occurs	Designates continue/stop running when an interrupt-scan overload error occurs		
R7C3	REMOTE RUN allowed	0: RUN prohibited 1: RUN allowed	Designates whether operation based on the task code is allowed		
R7C4	REMOTE STOP allowed	0: STOP prohibited 1: STOP allowed	Designates whether stop based on the task code is allowed		
R7C5	Undefined	Do not use.			
R7C6	Undefined	Do not use.			
R7C7	Modification during RUN allowed	0: Modification during RUN prohibited 1: Modification during RUN allowed	Designates whether online change in RUN is allowed in the user program	Set by user	Cleared by user or cleared when retentive area cleared.
R7C8	Serious failure flag	0: No serious failure 1: Serious failure	Indicates whether there is an abnormal in the system	Set by system	
R7C9	Microcomputer abnormal	0: Normal 1: Abnormal	Indicates whether there is an abnormal in the microcomputer		
R7CA	User memory abnormal	0: Normal 1: Abnormal	Indicates whether there is an abnormal in the user memory		
R7CB	Undefined	Do not use.			
R7CC	Memory size over	0: Normal 1: Abnormal	Indicates whether the capacity set by the parameter exceeds loaded memory capacity	Set by system	Cleared by user or cleared when retentive area cleared.
R7CD	I/O verify mismatch	0: Normal 1: Unmatched	Indicates whether I/O assignment and loading are matched (Mismatched information output to WRF002)		
R7CE	Communication module assignment verify mismatch	0: Normal 1: Unmatched	Indicates whether I/O assignment and loading are matched (Mismatched information output to WRF003)		
R7CF	Undefined	Do not use.			
R7D0	Remote error	0: Normal 1: Abnormal	Indicates whether the remote module is normal (Abnormal slot number display to WRF006, detailed information output to WRF080 to WRF0DF)	Set by system	Cleared by user or cleared when retentive area cleared.
R7D1	Overload error (normal scan)	0: Normal 1: Scan time over	Indicates whether the scan execution time has exceeded the designated time		
R7D2	Overload error (periodic scan)	0: Normal 1: Scan time over	Indicates whether the periodic scan was completed within cycle time		
R7D3	Undefined	Do not use.		Set by system	Cleared by user or cleared when retentive area cleared.
R7D4	Grammar/assemble error	0: Normal 1: Error	Indicates whether there is a grammar error in the user program (Detailed information output to WRF001)		
R7D5	I/O module error	0: Normal 1: Abnormal	Indicates whether there is an abnormal in the I/O module (Error slot number in WRF005)		

No.	Name	Meaning	Description	Setting condition	Resetting condition
R7D6	Number of I/O assignment points over	0: Normal 1: I/O assignment points over	Indicates whether the number of I/O assigned points has exceeded the maximum number	Set by system	Cleared by user or cleared when retentive area cleared.
R7D7	Communication module abnormal	0: Normal 1: Abnormal	Indicates whether there is an abnormal in the communication module (Abnormal slot number output to WRF004)		
R7D8	System bus abnormal	0: Normal 1: Abnormal	Indicates whether there is an abnormal when the system bus is accessed		
R7D9	Battery error	0: Normal 1: Abnormal	Indicates battery voltage is low, or back up memory failure.		Cleared by system *
R7DA	Undefined	Do not use.			
R7DB	Self-diagnostic error	0: Normal 1: Error	Indicates whether there is a self-diagnostic error (Detailed information output to WRF000)	Set by system	Cleared by user or cleared when retentive area cleared.
R7DC	Undefined	Do not use.			
R7DD	Communication module assignment over	0: Normal 1: Error	Indicates whether the communication module assignment exceeds the maximum value	Set by system	Cleared by user or cleared when retentive area cleared.
R7DE	Link module abnormal	0: Normal 1: Abnormal	Indicates whether there is an abnormal in the link module (Abnormal slot number displayed in WRF007, detailed information output to WRF0E0 to WRF19F)		
R7DF	Undefined	Do not use.			
R7E0	Operation switch position (STOP)	0: Other than below. 1: RUN switch position is STOP and REMOTE setting switch position is off.	One of these is on	Set by system	Cleared by system
R7E1	Operation switch position (REMOTE)	0: Other than below. 1: When RUN switch position is STOP and REMOTE setting switch is on			
R7E2	Operation switch position (RUN)	0: RUN switch position is STOP 1: RUN switch position is RUN			
R7E3	1 scan ON after RUN	0: From the second scan after RUN 1: 1 scan after RUN			
R7E4	Always ON	0: Non-status of 0 1: Always	Always outputs 1 regardless of CPU status		Always ON
R7E5	0.02 second clock	0: 0.01 seconds 1: 0.01 seconds			Cleared by system
R7E6	0.1 second clock	0: 0.05 seconds 1: 0.05 seconds			
R7E7	1.0 second clock	0: 0.5 seconds 1: 0.5 seconds			
R7E8	Occupied flag	0: Unoccupied 1: Occupied	Indicates occupancy status from the peripheral device		
R7E9	RUN prohibited	0: Operation allowed 1: Operation prohibited	Indicates whether it is operation prohibited status		

* Battery error (R7D9) will turn off when the error factor is gone by exchanging the batteries, etc.

No.	Name	Meaning	Description	Setting condition	Resetting condition
R7EA	Executing an 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	Set by system	Cleared by system
R7EB	Undefined	Do not use.			
R7EC	Clear error special internal output	Clear with 1	Clears error special internal outputs (WRF000 to F00A, R7C8 to 7DE)	Set by user	Cleared by system
R7ED	Undefined	Do not use.			
R7EE	Battery error display	0: Upon error, ERR lamp flashes 1: Upon error, ERR lamp turns off	Indicates whether or not the ERR lamp is used to notify battery errors	Set by user	Cleared by user or cleared when retentive area cleared.
R7EF	Backup memory writing execution flag (EH-CPU104A/208A/ 308A/318A/448 only)	0: Write complete 1: Write in progress	Indicates whether or not data is being written to the backup memory.	Set by system	Cleared by system
R7F0	Carry flag (C)	0: No carry 1: Carry	Indicates whether there is a carryover from the operation result		
R7F1	Overflow flag (V)	0: No overflow 1: Overflow	Indicates whether there is overflow in the operation result		
R7F2	Shift data (SD)	0: Shift data "0" 1: Shift data "1"	Designates the shift data used in shift commands, etc.	Set by user	Cleared by user
R7F3	Operation error (ERR)	0: Normal 1: Error	Indicates whether there is an operation error when operation is executed	Set by system	Cleared by system
R7F4	Data error (DER)	0: Normal 1: Error	Indicates whether there is a data error when operation is executed		
R7F5	Undefined	Do not use.			
R7F6	Scan time base value specification (EH-CPU***A /448/516/548) Undefined (EH-CPU104/208/ 308/316)	0: 10 ms 1: 1 ms Do not use.	Specifies the time base of the scan time display (WRF010-WRF012).	Set by user	Cleared by user
R7F7	Memory board busy	0: Standby 1: Busy	Indicates whether the memory board that was accessed during the execution of the FUN command (FUN 211 through FUN 213) is currently in use for other processing	Set by system	Cleared by system
R7F8	Calendar, clock read request Undefined for the EH-CPU104	1: Read	Read present value of calendar, clock and set in WRF01B to WRF01F	Set by user	Cleared by system
R7F9	Calendar, clock setting request Undefined for the EH-CPU104	1: Set	Set the data set in WRF01B to WRF01F in the calendar, clock		
R7FA	Clock ± 30 second adjustment request Undefined for the EH-CPU104	1: Request adjustment	When second data is 0 to 29, it becomes 0 seconds and when it is 30 to 59, +1 minute is added and second data becomes 0		
R7FB	Calendar, clock set data error Undefined for the EH-CPU104	0: Normal 1: Error	Indicates whether there is an error in calendar, clock set data	Set by system	
R7FC to R7FD	Undefined	Do not use.			
R7FE	Data logging error	0: Normal 1: Error	Indicates whether or not there is an error in the memory board or log data	Set by system	Cleared by user or cleared when retentive area cleared.
R7FF	Undefined	Do not use.			

13.2 Word Special Internal Output Area

Definitions of the word special internal output area (WRF000 to WRF1FF) are given below.

No.	Name	Storage data	Description	Setting condition	Resetting condition
WRF000	Self diagnosis error code	Error code (2 digit hexadecimal, upper 2 digits are 00)	Stores the error number deleted in the CPU as a binary code	Set by system	Cleared by user
WRF001 (R7D4)	Grammar/assemble error details	Grammar/assemble error code (4 digit hexadecimal)	Error code for user program grammar/assemble error is stored		
WRF002 (R7CD)	I/O verify mismatch details	Mismatched slot number *1	15 12 11 8 7 4 3 0 0 a b 0 a: Unit number (0 to 1) b: Slot number (0 to 7)		
WRF003 (R7CE)	Communication module I/O verify mismatch details	Mismatched module slot number *1	Mismatched slot number is stored (lower 4 bits: 0 to 7, upper bit 0)		
WRF004 (R7D7)	Communication module abnormal slot number	Slot number of the communication module with abnormal *1	Error slot number (lower 4 bits: 0 to 7, upper bit 0)		
WRF005 (R7D5)	I/O module abnormal slot number	Slot number of the I/O module with abnormal *1	15 12 11 8 7 4 3 0 0 a b 0 a: Unit number (0 to 1) b: Slot number (0 to A)		
WRF006 (R7D0)	Remote I/O master station module abnormal slot number	Slot number of the module with abnormal *1			
WRF007 (R7DE)	Link module abnormal slot number	Slot number of the module with abnormal *1			
WRF008	Undefined	Do not use.			
WRF009	Undefined	Do not use.			
WRF00A	Undefined	Do not use.			
WRF00B	Calendar and clock present value (4 digit BCD) (Nor supported by CPU104(A))	Year	Always displays the 4 digit year	Set by system	-
WRF00C		Month/day	Month/day data always displayed		
WRF00D		Day of the week	Day of the week data always displayed (Sunday: 0000 to Saturday: 0006)		
WRF00E		Hour/minute	Hour/minute data always displayed (24-hour system)		
WRF00F		Seconds	Seconds data always displayed (Lower 2 digits, upper 2 digits are 00)		
WRF010	Scan time (maximum value)	Maximum execution time for a normal scan	Maximum execution time for a normal scan is stored in 10 ms units *2	Set by system	Cleared by system (when RUN starts)
WRF011	Scan time (present value)	Present value of execution time for a normal scan	Present value of execution time for a normal scan is stored in 10 ms units *2		
WRF012	Scan time (minimum value)	Minimum execution time for a normal scan	Minimum execution time for a normal scan is stored in 10 ms units (the first scan after RUN is HFFFF)		
WRF013	CPU status	15 11 10 8 7 6 5 4 3 2 1 0 Unused a b c d e f g h i a: CPU type (011), b: Battery error (1-error, 0-no error), c: Unused, d: Unused, e: Unused, f: Error (1-error, 0-no error), g: Unused, h: Halt (1-In halt, 0-Out of halt), I: CPU operation (1-RUN, 0-STOP)			-

*1 If error slot number to be checked, existing information should be cleared by setting off the respective bit of special internal output (address in ( )) once, or setting on R7EC.

*2 Time base of scan time can be switched to 1ms by setting R7F6 for CPU448(A)/516/548.

<table>
<tr><th>No.</th><th>Name</th><th>Storage data</th><th>Description</th><th>Setting condition</th><th>Resetting condition</th></tr>
<tr><td>WRF014</td><td>Word internal output capacity</td><td>Number of words for word internal output (WR)</td><td>CPU104(A): H1000, CPU208(A): H2000, CPU308(A): H4400, CPU316(A): H5800, CPU448(A): HC400, CPU516 : H5800, CPU548 : HC400</td><td rowspan="5">Set by system</td><td>-</td></tr>
<tr><td>WRF015</td><td>Operation error code</td><td>Operation error code</td><td>Operation error code is stored (4 digit hexadecimal)</td><td>Cleared by user</td></tr>
<tr><td>WRF016</td><td>Division remainder register (lower)</td><td rowspan="2">Remainder data when division command executed</td><td>For a double word operation: WRF017 (upper), WRF016 (lower)</td><td rowspan="3">Cleared by system</td></tr>
<tr><td>WRF017</td><td>Division remainder register (upper)</td><td>For a word operation: WRF016 only</td></tr>
<tr><td>WRF018</td><td>Communication module startup flag</td><td colspan="2">15 9 8 7 6 5 4 3 2 1 0
Unused | a | b | c | d | e | f | g | h | i
Bit number corresponds to the slot number
1-startup complete 0-startup incomplete</td></tr>
<tr><td>WRF019</td><td>Undefined</td><td colspan="2">Do not use.</td><td></td><td></td></tr>
<tr><td>WRF01A</td><td>Baud rate set value between CPU and modem</td><td colspan="2">15 6 5 4 3 2 1 0
Unused | a | b | c | d | e | f
a-57,600 bps b-38,400 bps c-19,200 bps
d-9,600 bps e-4,800 bps f-2,400 bps
The corresponding bit is set to 1. The lower bit takes priority.
If all bits are 0, it is 2,400 bps.</td><td>Set by user</td><td>Cleared by user</td></tr>
<tr><td>WRF01B</td><td rowspan="4">Read and set values for calendar and clock
(4 digit BCD)
Not supported by CPU104(A)</td><td>Year [yyyy]</td><td>4 digit year to be read or written.</td><td rowspan="4">Set by system or by user</td><td rowspan="4">Cleared by user</td></tr>
<tr><td>WRF01C</td><td>Month/day [mmdd]</td><td>Month/date to be read or written.</td></tr>
<tr><td>WRF01D</td><td>Day of the week data (Sunday: 0000 to Saturday: 0006)</td><td>Day of the week to be read or written.</td></tr>
<tr><td>WRF01E</td><td>Hour/minute (24-hour system) [hhmm]</td><td>Stores the read hour/minute value or sets the set value</td></tr>
<tr><td>WRF01F</td><td></td><td>Seconds [00ss]</td><td>Stores the read second value or sets the set value</td><td></td><td></td></tr>
<tr><td>WRF020 to F021</td><td>Communication module status (slot 0)</td><td>Status data</td><td rowspan="9">15 8 7 0
Status 1 | Status 2
Status 3 | Status 4
Refer to communication module specifications for details</td><td rowspan="9">Set by system</td><td rowspan="9">Cleared by system</td></tr>
<tr><td>WRF022 to F023</td><td>Communication module status (slot 1)</td><td>Status data</td></tr>
<tr><td>WRF024 to F025</td><td>Communication module status (slot 2)</td><td>Status data</td></tr>
<tr><td>WRF026 to F027</td><td>Communication module status (slot 3)</td><td>Status data</td></tr>
<tr><td>WRF028 to F029</td><td>Communication module status (slot 4)</td><td>Status data</td></tr>
<tr><td>WRF02A to F02B</td><td>Communication module status (slot 5)</td><td>Status data</td></tr>
<tr><td>WRF02C to F02D</td><td>Communication module status (slot 6)</td><td>Status data</td></tr>
<tr><td>WRF02E to F02F</td><td>Communication module status (slot 7)</td><td>Status data</td></tr>
<tr><td>WRF030 to F031</td><td>Communication module status (slot 8)</td><td>Status data</td></tr>
<tr><td>WRF032 to F035</td><td>Undefined</td><td colspan="2">Do not use.</td><td></td><td></td></tr>
</table>

<table>
<tr><th>No.</th><th>Name</th><th>Meaning</th><th>Description</th><th>Setting condition</th><th>Resetting condition</th></tr>
<tr><td>WRF036</td><td>Port 1 setting
(General purpose port)

(CPU308(A)/316(A)/ 448(A)/516/548)

This setting is valid when port 1 is general purpose port.
(DIP SW No.5:OFF)</td><td colspan="2">15 … 0
Unused | a | b | c | d | e
a- Built-in termination resistor 0:OFF, 1:ON b- RS-485 c- RS-422
d- Setting bit
<table><tr><th></th><th colspan="2">Built-in term. resistor OFF</th><th colspan="2">Built-in term. resistor ON</th></tr><tr><th></th><th>Set by user</th><th>Set by CPU</th><th>Set by user</th><th>Set by CPU</th></tr><tr><td>RS-232C</td><td>H0003</td><td>H0002</td><td>-</td><td>-</td></tr><tr><td>RS-422</td><td>H0005</td><td>H0004</td><td>H0015</td><td>H0014</td></tr><tr><td>RS-485</td><td>H0009</td><td>H0008</td><td>H0019</td><td>H0018</td></tr></table>- Please set this register manually by programming software or programming device, not by user program. (If you need to set it by user program, please do it carefully so not to be changed frequently.)
- This setting will be valid immediately after changed, and saved to the backup memory. (When the setting is effective, “e” bit is reset.)
- When this value is undefined at the power ON, CPU will work with default value (RS-232C).
- This setting is valid only when DIP SW5 is OFF (Port 1 for general purpose).</td><td rowspan="2">Set by system or by user</td><td rowspan="2">Cleared by user</td></tr>
<tr><td>WRF037</td><td>Port 1 setting
(Dedicated port)
(EH-CPU***A/448/ 516/548)

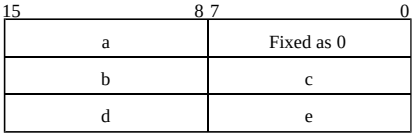
This setting is valid when port 1 is dedicated port.
(DIP SW No.5:ON)</td><td colspan="2">15 14 13 12 11 10 9 8 7 … 0
a | b | c | d | e | f | g | h | Station No.
a: Setting bit 1 - Set This bit is cleared after the setting completed.
b: Transmission control procedure
0 - Procedure 1 1 - Procedure 2
c: Station number presence
0 - No station number 1 - With Station number
d: Built-in termination resistor 0 - OFF 1 - ON
e, f: Communication interface status display (set by system)
<table><tr><th>e</th><th>f</th><th>Interface</th></tr><tr><td>0</td><td>0</td><td>RS-232C</td></tr><tr><td>0</td><td>1</td><td>RS-422 (Not supported by CPU104A/208A)</td></tr><tr><td>1</td><td>0</td><td>RS-485 (Not supported by CPU104A/208A)</td></tr></table>g, h: Communication interface setting (set by the user)
<table><tr><th>g</th><th>h</th><th>Interface</th></tr><tr><td>0</td><td>0</td><td>RS-232C</td></tr><tr><td>0</td><td>1</td><td>RS-422 (Not supported by CPU104A/208A)</td></tr><tr><td>1</td><td>0</td><td>RS-485 (Not supported by CPU104A/208A)</td></tr></table><table><tr><th>Interface</th><th>Set by user</th><th>Set by CPU</th></tr><tr><td>RS-232C</td><td>H*0**</td><td>H*0**</td></tr><tr><td>RS-422</td><td>H*1**</td><td>H*5**</td></tr><tr><td>RS-485</td><td>H*2**</td><td>H*A**</td></tr></table>* : Setting value except for e,f,g,h.
But, when setting is completed, bit “a” is turned off with the system.
Station number: 0 to 31 in 2-digit BCD If the station number exceeds this range, the system will set 31.
For more details, see Chapter 10, “Communication Specifications.”
- Please set this register manually by programming software or programming device, not by user program. (If you need to set it by user program, please do it carefully so not to be changed frequently.)
- This setting will be valid at the next power ON, and saved to the backup memory. Please turn off the power after the setting. (When the setting is effective, “a” bit is reset.)
- When this value is undefined at the power ON, CPU will work with default value (Transmission control procedure 1, No station number, RS-232C).</td></tr>
</table>

<table>
<tr><th>No.</th><th>Name</th><th>Meaning</th><th>Description</th><th>Setting condition</th><th>Resetting condition</th></tr>
<tr><td>WRF038</td><td>Setting of system processing time (EH-***A/448/516/548)</td><td colspan="2">15 8 7 0
User setting time | Current operating condition

User setting time: 3 to 9 in 1-digit BCD If the user setting time exceeds this range, the system will set 2.
Current system processing time: The system sets the processing time of the system currently in operation.
The value which is actually setting in the system complies with the setting value as the following table.
The value which an underline has is changed forcibly with the system
<table><tr><td>Setting value</td><td>H0</td><td>H1</td><td>H2</td><td>H3</td><td>H4</td><td>H5</td><td>H6</td><td>H7</td><td>H8</td><td>H9</td></tr><tr><td>INT0 (*1) Not exist</td><td>H2</td><td>H1</td><td>H2</td><td>H3</td><td>H4</td><td>H5</td><td>H6</td><td>H7</td><td>H8</td><td>H8</td></tr><tr><td>INT0 (*1) Exist</td><td>H2</td><td>H1</td><td>H2</td><td>H3</td><td>H2</td><td>H2</td><td>H2</td><td>H2</td><td>H2</td><td>H2</td></tr></table>(*1) INT0 is 5ms periodical scan with CPU***A/448/516/548.
- Please set this register manually by programming software or programming device, not by user program. (If you need to set it by user program, please do it carefully so not to be changed frequently.)
- This setting will be valid immediately after changed, and saved to the backup memory. (When the setting is effective, “User setting time” is set on “current operating condition”.)
- When this value is undefined at the power ON, CPU will work with default value (H02).
- For more details, see Chapter 8, “Operating and Stopping EH-150.”</td><td rowspan="2">Set by system or by user</td><td rowspan="2">Cleared by user</td></tr>
<tr><td>Setting of system processing time (EH-CPU308/316 only of ROM Ver. 04 or later)</td><td colspan="2">15 8 7 0
User setting time | Current operating condition

User setting time: 5 in BCD If a value other than 5 is specified, the system processing time will vary.
Current operating time: 00 - System processing time varies.
05 - System processing time is fixed at 5 ms.
*The current operating condition section is set by system.
- Please set this register manually by programming software or programming device, not by user program. (If you need to set it by user program, please do it carefully so not to be changed frequently.)
- This setting will be valid immediately after changed, and saved to the backup memory. (When the setting is effective, “H00” or “H05” is set on “Current operating condition” by “User setting time”.)
- When this value is undefined at the power ON, CPU will work with default value (System processing time varies).
- For more details, see Chapter 8, “Operating and Stopping EH-150.”</td></tr>
<tr><td></td><td>Undefined (EH-CPU104/208)</td><td colspan="2">Do not use.</td><td></td><td></td></tr>
<tr><td>WRF039 to F03A</td><td>Undefined</td><td colspan="2">Do not use.</td><td></td><td></td></tr>
<tr><td>WRF03B</td><td>Mode time out at reception sequence of Modem control, (port 1) (EH-208A/308A/316A/ CPU448(A)/516/548)</td><td colspan="2">15 8 7 0
a | Unused | Time out setting time

a: 1 = Valid time out
0 = Invalid time out
Time out setting time(second) : H01 to HFF (0 to 255)
In case the most significant bit (a) is 0, the setting time is cleared when power on. See 10.4.4 At Command (3) Sequence, Note 2, Page 10-17.</td><td>Set by user</td><td>Cleared by user</td></tr>
<tr><td></td><td>Undefined</td><td colspan="2">Do not use.</td><td></td><td></td></tr>
<tr><td>WRF03C</td><td>Modem time out (port 1)</td><td colspan="2">H80**: Time out monitoring is performed for ** seconds.
(The timer is set again upon reception of STX, ENQ or NAK.)</td><td>Set by user</td><td>Cleared by user</td></tr>
<tr><td>WRF03D to F03F</td><td>Undefined</td><td colspan="2">Do not use.</td><td></td><td></td></tr>
</table>

<table>
<tr><th>No.</th><th>Name</th><th colspan="2">Meaning</th><th>Description</th><th>Setting condition</th><th>Resetting condition</th></tr>
<tr><td>WRF040 to F042</td><td>Occupied member registration area 1</td><td colspan="2" rowspan="4">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 (*)
(*) When dedicated port:
Port 1 is H02 and port 2 is H01.</td><td rowspan="4">Set by system</td><td rowspan="4">Cleared by system</td></tr>
<tr><td>WRF043 to F045</td><td>Occupied member registration area 2</td></tr>
<tr><td>WRF046 to F048</td><td>Occupied member registration area 3</td></tr>
<tr><td>WRF049 to F04B</td><td>Occupied member registration area 4</td></tr>
<tr><td>WRF04C to F04F</td><td>Undefined</td><td colspan="2">Do not use.</td><td></td><td></td></tr>
<tr><td>WRF050</td><td>System use area</td><td colspan="2">System ROM version information</td><td></td><td></td></tr>
<tr><td>WRF051</td><td>System use area</td><td colspan="2">System FLASH ROM version information</td><td></td><td></td></tr>
<tr><td>WRF052 to F07D</td><td>Undefined</td><td colspan="2">Do not use.</td><td></td><td></td></tr>
<tr><td>WRF07E</td><td>Link area clear mode selection (EH-CPU308/316 of ROM Ver. 04 or later, CPU***A/448/516/548)</td><td colspan="2">15 0
0 0 0 0 0 0 0 0 0 0 d c 0 0 b a
When a mode to clear a link area is selected, the specified link area will be cleared to 0 when the CPU status changes from RUN to STOP.
a: 1 = Link area 1 mode switch request (this is set to 0 by system when mode switch is complete.)
b: Link area 1: 0 = Do not clear; 1 = Clear
c: 1 = Link area 2 mode switch request (this is set to 0 by system when mode switch is complete.)
d: Link area 2: 0 = Do not clear; 1 = Clear
<table><tr><th></th><th>Selection mode</th><th>User set value</th><th>Display after mode switch</th></tr><tr><td rowspan="2">Link 1</td><td>Clear</td><td>H0003</td><td>H00*2</td></tr><tr><td>Not clear</td><td>H0001</td><td>H00*0</td></tr><tr><td rowspan="2">Link 2</td><td>Clear</td><td>H0030</td><td>H002*</td></tr><tr><td>Not clear</td><td>H0010</td><td>H001*</td></tr></table>*: Retains the status before switching.
- Please set this register manually by programming software or programming device, not by user program. (If you need to set it by user program, please do it carefully so not to be changed frequently.)
- This setting will be valid immediately after changed, and saved to the backup memory. (When the setting is effective, “a” bit and “c” bit is reset.)
- When this value is undefined at the power ON, CPU will work with default value (The link area 1,2 is not cleared by zero) .</td><td>Set by system or by user</td><td>-</td></tr>
<tr><td></td><td>Undefined (CPU104(A)/208(A))</td><td colspan="2"></td><td></td><td></td></tr>
<tr><td>WRF07F</td><td>Log error code</td><td>Log error code</td><td>Stores the log error code.
For details, see the data logging specifications.</td><td></td><td>Set by system</td><td>Cleared by user</td></tr>
<tr><td>WRF080 to F097</td><td>Remote master 1 error flag</td><td colspan="3" rowspan="4">Remote error information
(refer to separate sheet for details)</td><td rowspan="5">Set by system</td><td rowspan="5">Cleared by system</td></tr>
<tr><td>WRF098 to F0AF</td><td>Remote master 2 error flag</td></tr>
<tr><td>WRF0B0 to F0C7</td><td>Remote master 3 error flag</td></tr>
<tr><td>WRF0C8 to F0DF</td><td>Remote master 4 error flag</td></tr>
<tr><td>WRF0E0 to F13F</td><td>Link 1 error flag</td><td colspan="3" rowspan="2">Link information
(refer to separate sheet for details)</td></tr>
<tr><td>WRF140 to F19F</td><td>Link 2 error flag</td></tr>
</table>

13.3 Remote Error Flag Area

Details of the remote error flag error are given below.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Slave station participation flag	a																+ 00
Slave station error flag	b																+ 01
Master station error detail information	c	d	e	f	g	h	i	j	Number of times transmission errors *1								+ 02
Slave station No. 0 detail information	k	l	m	n	o	p	q	r	Number of times transmission errors *2								+ 03
Slave station No. 1 detail information	k	l	m	n	o	p	q	r	Number of times transmission errors *2								+ 04
Slave station No. 2 detail information	k	l	m	n	o	p	q	r	Number of times transmission errors *2								+ 05
Slave station No. 3 detail information	k	l	m	n	o	p	q	r	Number of times transmission errors *2								+ 06
Slave station No. 4 detail information	k	l	m	n	o	p	q	r	Number of times transmission errors *2								+ 07
Slave station No. 5 detail information	k	l	m	n	o	p	q	r	Number of times transmission errors *2								+ 08
Slave station No. 6 detail information	k	l	m	n	o	p	q	r	Number of times transmission errors *2								+ 09
Slave station No. 7 detail information	k	l	m	n	o	p	q	r	Number of times transmission errors *2								+ 0A
Slave station No. 8 detail information	Undefined																+ 0B
Slave station No. 9 detail information	Undefined																+ 0C
Slave station No. 10 detail information	Undefined																+ 0D
Slave station No. 11 detail information	Undefined																+ 0E
Slave station No. 12 detail information	Undefined																+ 0F
Slave station No. 13 detail information	Undefined																+ 10
Slave station No. 14 detail information	Undefined																+ 11
Slave station No. 15 detail information	Undefined																+ 12
I/O verify mismatch slot No.	0									Slave station No.				Slot No.			+ 13
I/O error slot No.	0									Slave station No.				Slot No.			+ 14
Refresh time (maximum)	(unit: ms)																+ 15
Refresh time (minimum)	(unit: ms)																+ 16
Refresh time (present)	(unit: ms)																+ 17

a: Bit number corresponds to the slave station number (1: participating, 0: non-participating)
b: Bit number corresponds to the slave station number (1: error, 0: no error)
c: Time out error (1: error, 0: no error)
d: Frame error (1: error, 0: no error)
e: System bus error (1: error, 0: no error)
f: Slave station I/O error (1: error, 0: no error)
g: Duplicate station number (1: error, 0: no error)
h: Slave station connection mismatch (1: error, 0: no error)
i: I/O information mismatch (1: error, 0: no error)
j: Remote points error (1: error, 0: no error)
k: Same as c
l: Same as d
m: Undefined
n: Same as f
o: Same as g
p: Same as h
q: Same as i
r: Undefined

*1 “Number of times transmission error” is a cumulative total of the number of c or d error occurrences.
*2 “Number of times transmission error” is a cumulative total of the number of k or l error occurrences.

13.4 Link Error Flag Area

Details of the link error flag area are given below.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Local station error information	Undefined		a	b	c	d	e	f	Undefined		Disconnected station number						+ 00
	15	to														0	+ 01
Link participation flag (g)	31	to														16	+ 02
	47	to														32	+ 03
	63	to														48	+ 04
	15	to														0	+ 05
Link operation status flag (h)	31	to														16	+ 06
	47	to														32	+ 07
	63	to														48	+ 08
	3				2				1				0				+ 09
CPU status flag (i)	:																:
	:																:
	63				62				61				60				+ 18
	15	to														0	+ 19
Error status flag (j) *1	31	to														16	+ 1A
	47	to														32	+ 1B
	63	to														48	+ 1C
	k	l	m	Undefined							Number of times transmission errors *2						+ 1D
Station 0 to 63	:																:
Error detail information	:																:
	k	l	m	Undefined							Number of times transmission errors *2						+ 5C
Refresh time (maximum)	(Unit: ms)																+ 5D
Refresh time (minimum)	(Unit: ms)																+ 5E
Refresh time (present)	(Unit: ms)																+ 5F

a: System bus error (1-error, 0-no error)
b: Undefined
c: Area error (1-error, 0-no error)
d: Duplicate area error (1-error, 0-no error)
e: Station number error (1-error, 0-no error)
f: Transmission path disconnected (1-error, 0-no error)
g: Number indicates the station number (1-participation, 0-non-participation)
h: Number indicates the station number (1-operating, 0-stopped)
i: Number indicates the station number (of the 4 bits, 1] 1-CPU error, 0-normal 2] Undefined 3] 1-HALT status, 0-other than HALT status 4] 1-running, 0-stopped)

1]	2]	3]	4]

j: Number indicates the station number (1-error, 0-no error)
k: Time out error (1-error, 0-no error)
l: Frame error (1-error, 0-no error)
m: Abnormal between CPU and link (1-abnormal, 0-normal)

*1 "Error status flag" is set to 1 when one of the k, l and m errors is generated in "Error detail information." k, l and m generate error when an error is detected while the peripheral device is communicating with the CPU of other station.
*2 "Number of times transmission errors" is a cumulative total of the number of k or l error occurrences.

Chapter 14 Troubleshooting

14.1 Error Indication and Countermeasure Procedures

The indication locations of errors detected by individual equipment in the EH-150 system are shown in Figure 14.1. When errors occur, remedy the errors using the error code list.

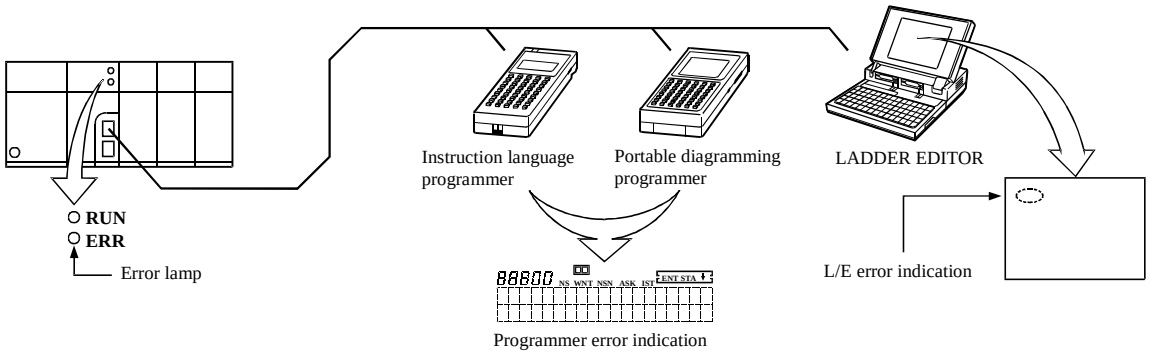


Figure 14.1 Error indication of EH-150

(1) Error indication

(a) CPU module error indication

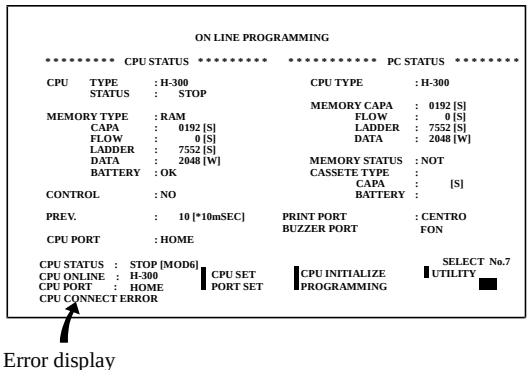
The CPU module will perform self-diagnostic tests using the microcomputer, and when there is an error the contents are indicated in the combination of the ERR and RUN lamps located in the front of the CPU module being lit/flashing/unlit. For Chapter 12 error codes and countermeasures, refer to the self-diagnosis error code list.

(b) Programmer error indication

Error codes encountered during program device operation, such as duplicate definition error, undefined error, operation error, program over, etc., will be displayed on the programming device. For detailed error codes, refer to the error code list in the programming device manual.

(c) GPCL error display

The error detected by the CPU during the GPCL operation is displayed at the bottom left of the screen.



Error display

If the cable is disconnected or the power to the master unit is turned off during the GPCL operation, “CPU CONNECTION ERROR” is displayed to indicate a connection error.

For the details of error codes, see the list of error codes in the GPCL manual.

(d) Setting in the special internal output

An error code is set in the special internal output area (such as WRF000). The smaller the error code value, 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 the same, the cause code generated last will be displayed.

The clearing of error special internal output is performed by setting the special internal output R7EC to 1. The R7EC can be set to 1 either by connecting the programming device or by including a subprogram that sets the R7EC using external input within the program. (If turning R7EC on by the program, always set it on after the error cause has been verified. However, if R7EC is turned on by a program that would generate a watchdog error, the system may clear the error cause and rerun after detecting a watchdog error.)

Note: Error codes are set in hexadecimal values. Verify error codes by setting the monitor to hexadecimal display.

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
9	Operation microcomputer abnormal
A	User memory abnormal
B	I/O bus abnormal
C	Memory size over
D	I/O verify mismatch
E	Communication module assignment verify mismatch
R7CF	(Undefined)
R7D0	Remote abnormal
1	Overload error (normal scan)
2	Overload error (periodical scan)
3	Overload error (interrupt scan)
4	Grammar/assemble error
5	I/O module abnormal
6	I/O assignment point over
7	Communication module abnormal
8	System bus error
9	Battery error
A	(Undefined)
B	Self-diagnostic error
C	(Undefined)
D	Communication module assignment over
R7DE	Link module abnormal

No.	Word special internal output
WRF000	Self-diagnostic error code
1	Grammar/assemble error details
2	I/O verify mismatch details
3	Communication module I/O verify mismatch details
4	Communication module abnormal slot number
5	I/O module abnormal slot number
6	Remote I/O master station module abnormal slot number
7	Link module abnormal slot number
8	(Undefined)
9	(Undefined)
WRF00A	(Undefined)

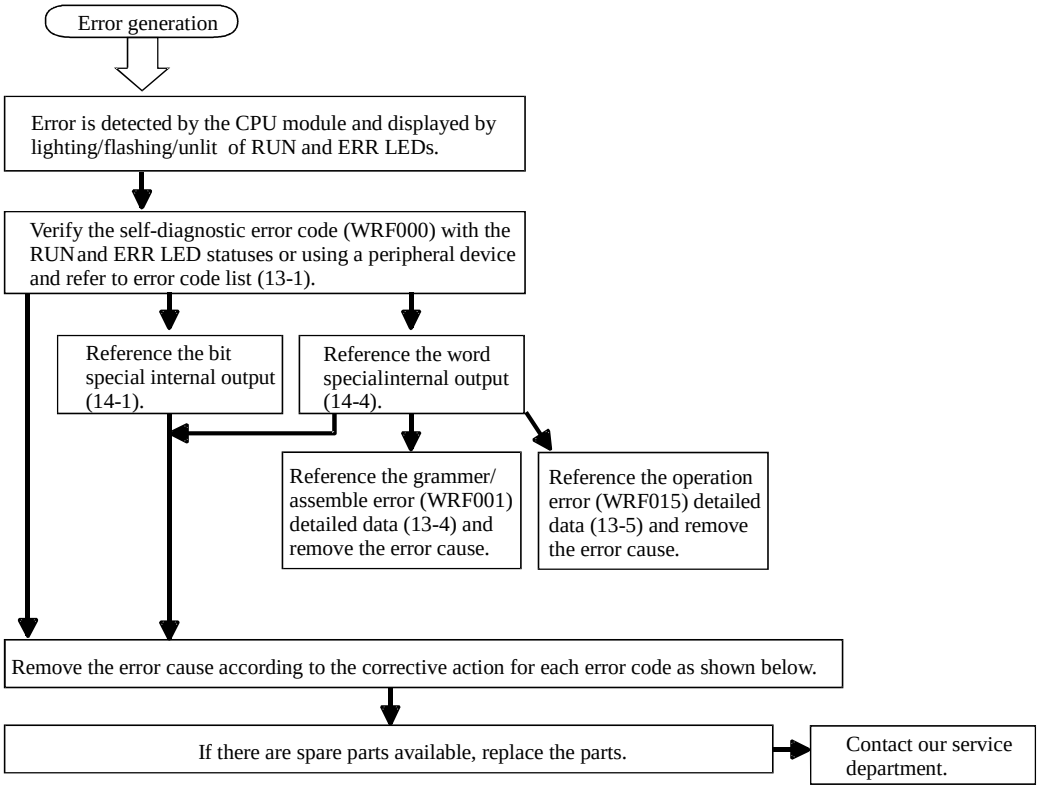
When all of the special internal output data cannot be cleared for program execution reasons, refer to the self-diagnostic error code list and clear only the respective error flags by using forced set of the programmer or peripheral device.

Caution

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.

(2) Corrective actions when errors are generated

The process flow when an error is generated is shown below.



Error code	Error name	Corrective action
11	System ROM error	Re-inspect the fixation of the CPU module to the basic base unit and restart the power supply. If the same error occurs, it is a hardware error in the CPU module, so replace the CPU module with a spare. Make sure that there are no machines, etc. that generate excessive noise.
12	System RAM error	
13	Microcomputer error	
15	System bus timeout error	
16	System program error	
22	Sequence processor error	
23	Undefined command	
27	Data memory error	
—	Power shut-off, power supply error	Check the basic and extended power supply voltage.
31	User memory error	The contents of the user program is destroyed. Perform initialization and transfer the program again. This is displayed when the machine is stored with a worn-out battery or without battery for a long period of time.
33	User memory size error	This may be displayed when the contents of the memory within the CPU module is unstable. If the same error is generated after initialization, replace the CPU module with a new one.
34	Grammar/assemble error	There is a grammar or assemble error in the user program. Verify the program and I/O assignment.
41	I/O information verification error	Check the I/O assignment. Check each I/O module, I/O controller fixation and expansion cable connection.
43	Remote error	Perform module settings, reset, etc. according to the error code of the malfunctioning remote module.
44	Overload error (normal scan)	Change the program so that the scan time of the user program is shorter or change the watchdog check time.
45	Overload error (periodic scan)	Change the program so that the periodic interrupt program execution time is shorter.
47	I/O assignment point over	Perform I/O assignment so that the maximum I/O points of the CPU module are not exceeded.

Error code	Error name	Corrective action
51	I/O module abnormal	Check for errors in I/O modules and replace the malfunctioning module.
52	I/O transmission error	Reset the power supply for the high-function module. Set the TRNS, RECV parameters correctly.
53	I/O invalid interrupt	Check to see if the links and remote modules are assigned properly in the I/O assignment.
54	Communication module abnormal	Perform error recovery procedures according to the error code of the malfunctioning communication module.
55	Communication module transmission error	
57	Communication module I/O over assignment	Perform assignment so that the maximum number of communication module assignments is not exceeded.
58	Communication module I/O information verify error	Change the parameters or properly install the communication function module.
59	Link module abnormal	Perform error recovery procedures according to the error code of the malfunctioning link module.
61	Port 2 transmission error (parity)	Check the connection of the connector cable. Check the settings such as the transmission speed. Check to see if there are any sources of noise near the cable.
62	Port 2 transmission error (framing/overrun)	
63	Port 2 transmission error (time out)	Check the connection of the connector cable. Check to see if there are any sources of noise near the cable. Verify the protocol designations, evaluate the host computer processing and correct any errors.
64	Port 2 transmission error (protocol error)	
65	Port 2 transmission error (BCC error)	
67	Port 1 transmission error (parity)	Check the connection of the connector cable. Check the settings such as the transmission speed. Check to see if there are any sources of noise near the cable.
68	Port 1 transmission error (framing/overrun)	
69	Port 1 transmission error (timeout)	Check the connection of the connector cable. Check to see if there are any sources of noise near the cable. Verify the protocol designations, evaluate the host computer processing and correct any errors.
6A	Port 1 transmission error (protocol error)	
6B	Port 1 transmission error (BCC error)	
71	Battery error	Replace the battery with a new one. Verify the connection of the battery connector.

Perform the following procedures to erase the error display.

(a) When CPU is stopped

Turn the CPU RUN switch to “STOP,” then to “RUN” again.

If the cause of the error has been corrected, the ERR lamp turns off. However, the error information remains in the error special internal output, which stores the CPU error types and details. (This makes it possible to analyze the error after recovery.) To reset the error information, perform the procedures shown in (b) or press the reset switch for the power failure memory protection (R.CL SW).

(b) When the CPU is still running

1] Clear the ERR lamp only.

Set “1” in the special internal output, R7EB.

2] Clear the ERR lamp display and error special internal output.

Set “1” in the special internal output, R7EC.

14.2 Checklist for troubles

If an error is generated in the EH-150 system, check the following items. If there are no problems in the following items, contact our service department.

(a) 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 capacity of the power supply module greater than the total of module consumption current?

(b) CPU related items

- Are the initial settings (CPU initialization, I/O assignment, parameter settings, etc.) proper?
- Are there any error codes that are output to the special internal output?
- Is the RUN switch in the proper location?
- Are batteries mounted properly? Is the battery life still remaining?
- Are the CPU connectors properly connected to the base connectors?

(c) Input module related items

- Is the input voltage within the specifications for the module?
- Is there any noise or chattering in the input?
- Do the I/O assignment numbers in the program match?
- Is the wiring done properly?

(d) Output module related items

- Do the module and the load power supply type (DC/AC) match?
- Do the load voltage and current match the module specifications?
- 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?

(e) Wiring related items

- Is the FE terminal of the power supply module grounded using class-3 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 module?

Caution

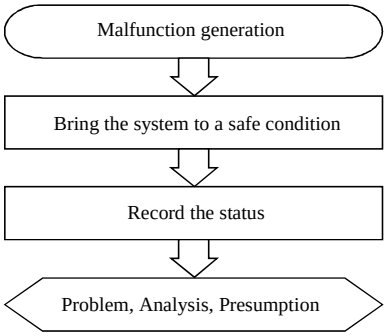
(a) Be sure to replace the modules with the power supply turned off.

(b) When returning the module for repair, please notify us of the malfunctioning effects in as much detail as possible (including error codes, malfunctioning output bit number, will not turn on or off, etc.).

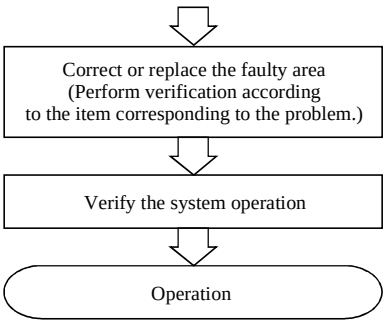
(c) The tools and devices necessary for troubleshooting are roughly as follows:
Phillips/flathead drivers, digital MultiMate, tester, oscilloscope (necessary depending on the case) etc.

14.3 Procedures to Solve

The processing flow when a malfunction has occurred is as follows:

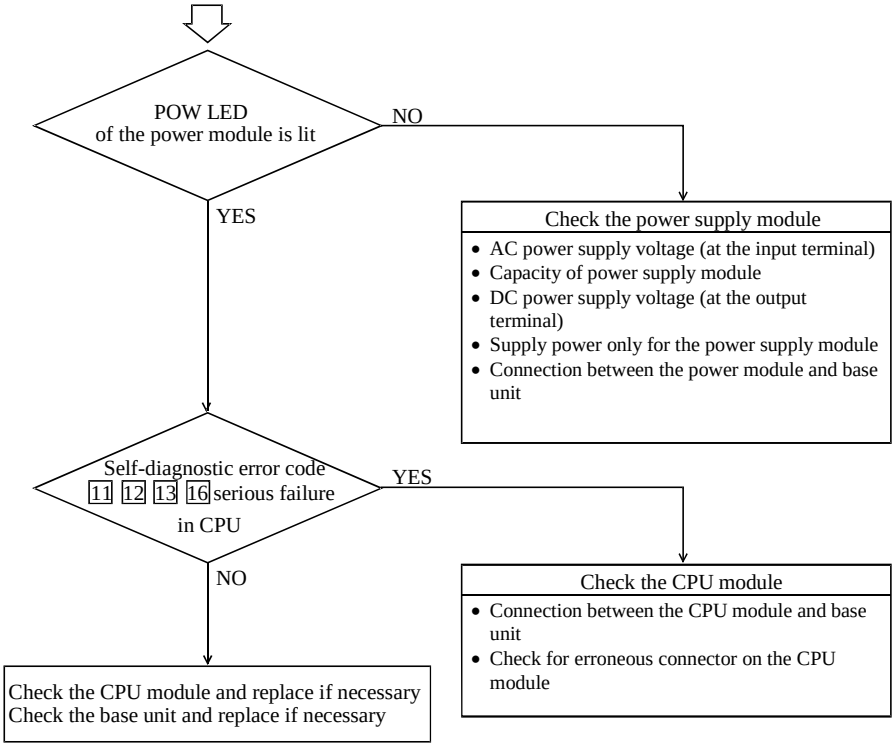


Major problems	Verification points	Typical causes of the problem	Refe-rence item
PLC does not start	Power LED, CPU error code	Power supply abnormal, power shut off, power supply capacity shortage, module connector error, serious CPU failure	(a)
PLC does not operate (Not in RUN mode)	CPU error code, CPU LED, Internal output of error	I/O assignment abnormal, incorrect parameter settings, incorrect user program, syntax error, drive conditions not met, write-occupied status	(b)
Operation stopped (RUN stopped)	Power LED, CPU LED, CPU error code	Power supply abnormal, expansion power supply abnormal/shut off, CPU abnormal, memory abnormal, communication module abnormal, base abnormal	(c)
Input error	CPU LED, I/O LED Monitoring by peripheral devices	User program timings, input power supply, bad connection, input module abnormal, I/O inductive noise	(d)
Output error	CPU LED, I/O LED, Monitoring by peripheral devices, Forced set	User programming, bad connection, output module abnormal, I/O inductive noise	(e)
Peripheral device error	CPU error code, CPU, peripheral devices	Serious CPU failure, peripheral device abnormal, peripheral drive setting error, cable abnormal	(f)



(a) PLC does not start

[The CPU ERR LED does not turn off even when power is started, nor peripheral devices cannot be connected on-line.]

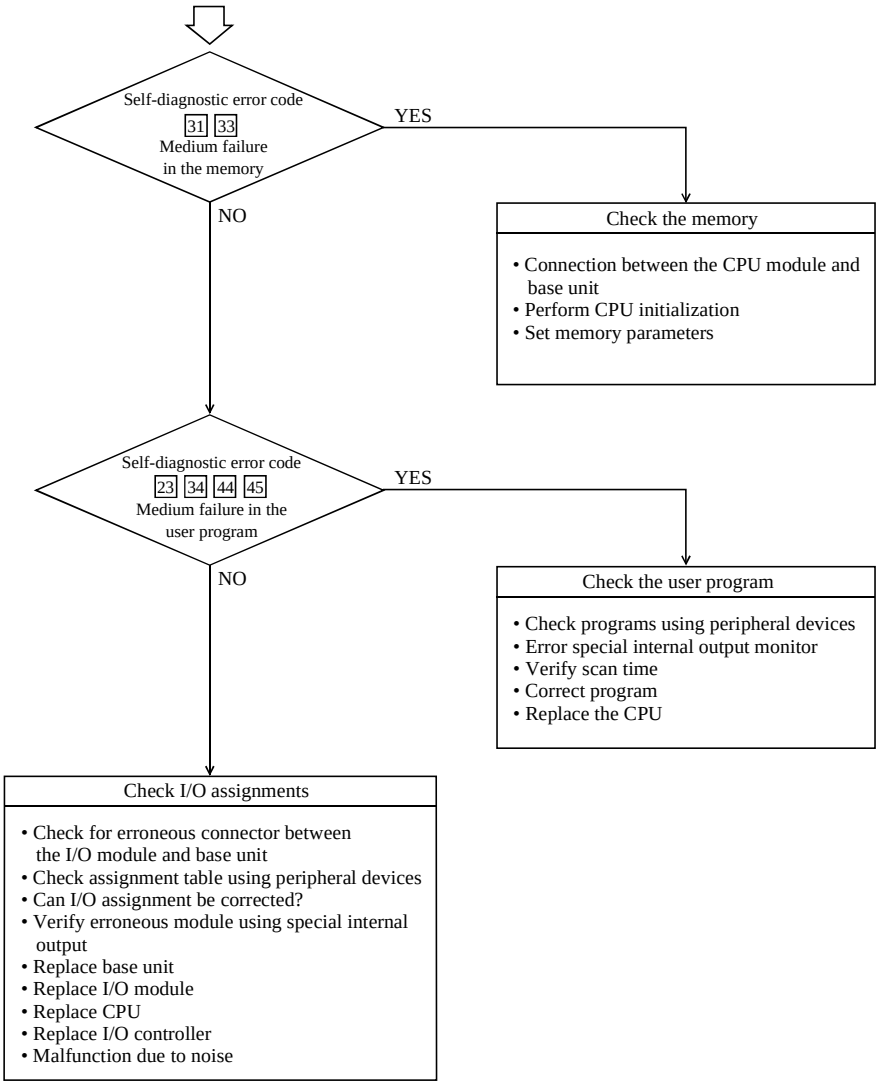


(b) PLC does operate (Not in RUN mode)

Even if the PLC operation conditions are met, the CPU does not operate (the RUN LED does not turn on) and remains stopped. However, the peripheral devices go on-line.

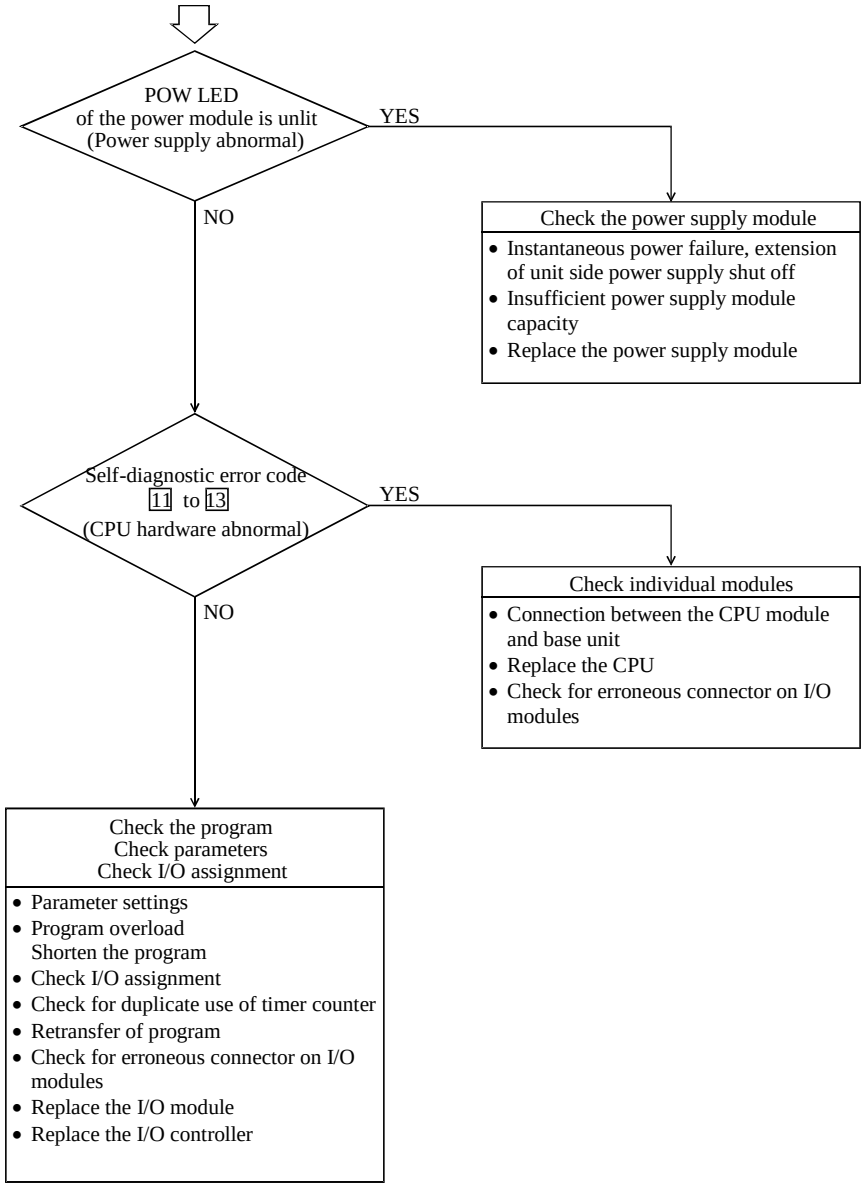
Caution

If the CPU is WRITE-occupied, the CPU will not run even if the RUN switch is switched from "STOP" to "RUN." The CPU starts running by pressing the GRS key after peripheral devices are connected.



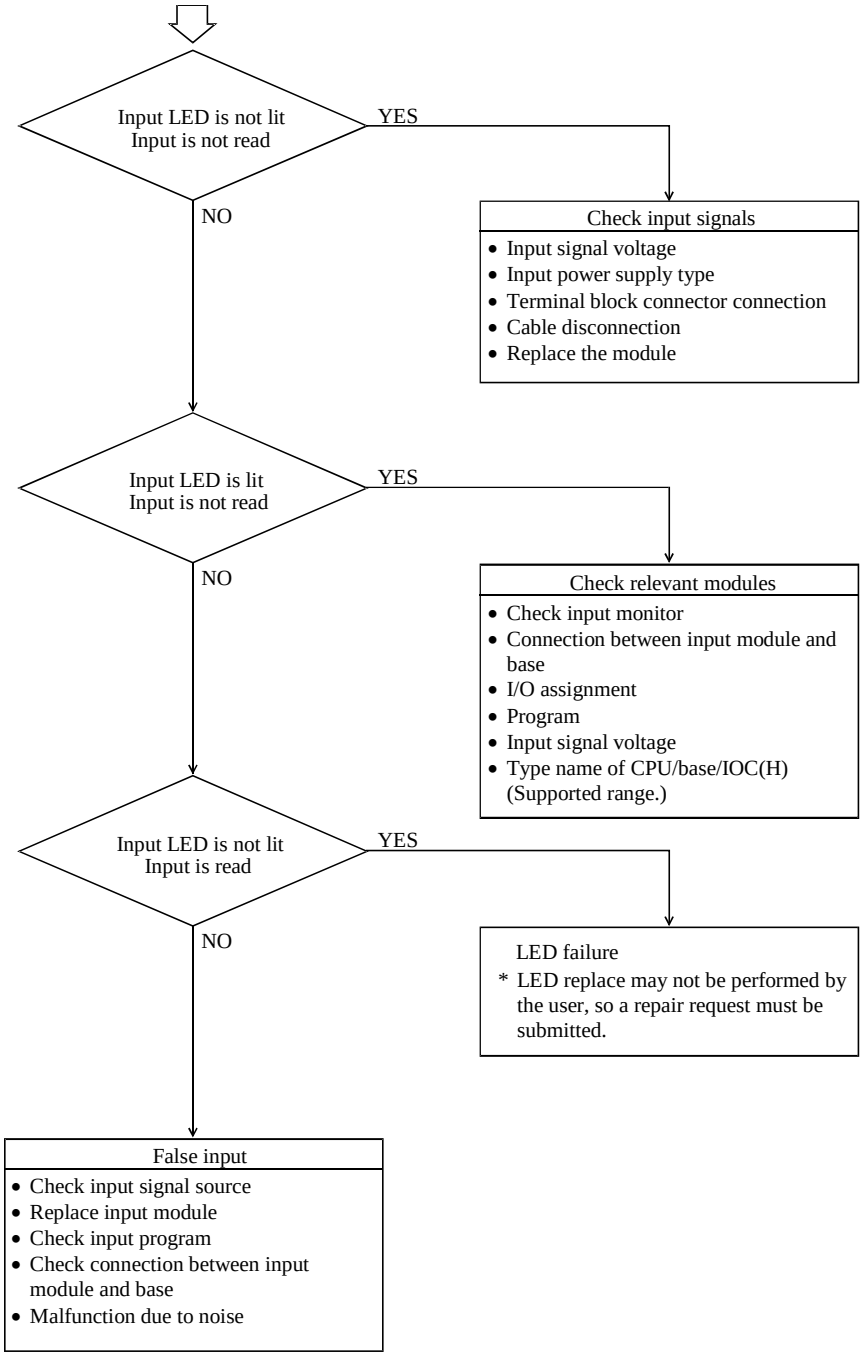
(c) Operation stopped (RUN stopped)

〔 During normal operation, the CPU suddenly stops (the RUN LED turns off). 〕

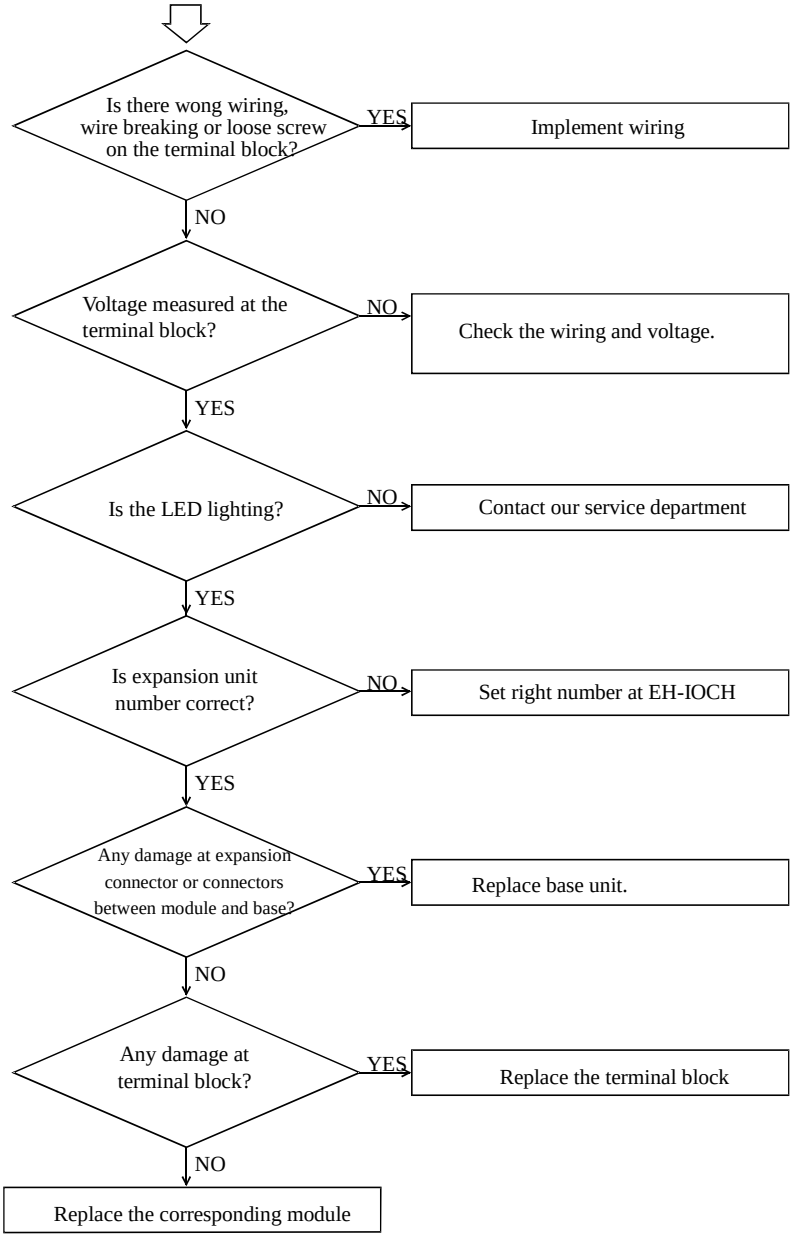


(d) Wrong input at input module, or input module does not work (abnormal operation)

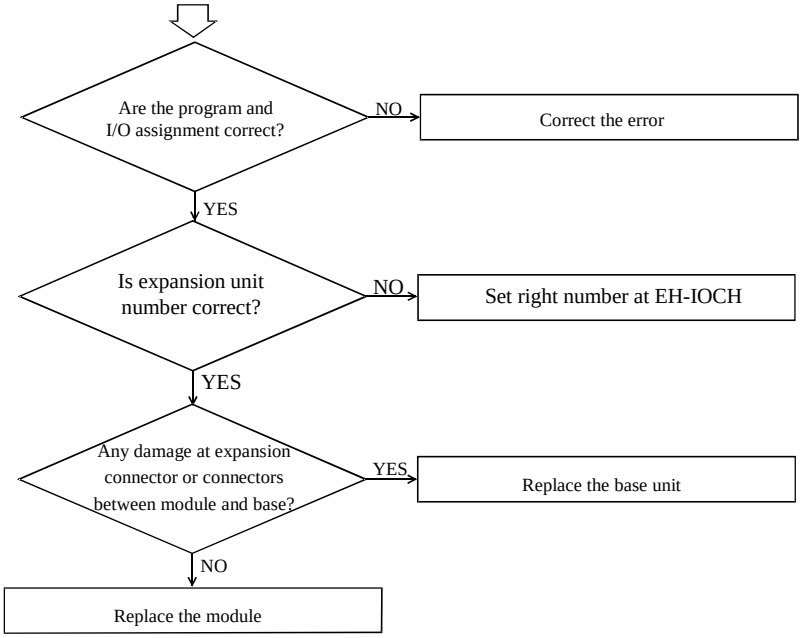
[CPU runs, but the input data is not correct.]



〔 No input signal. 〕

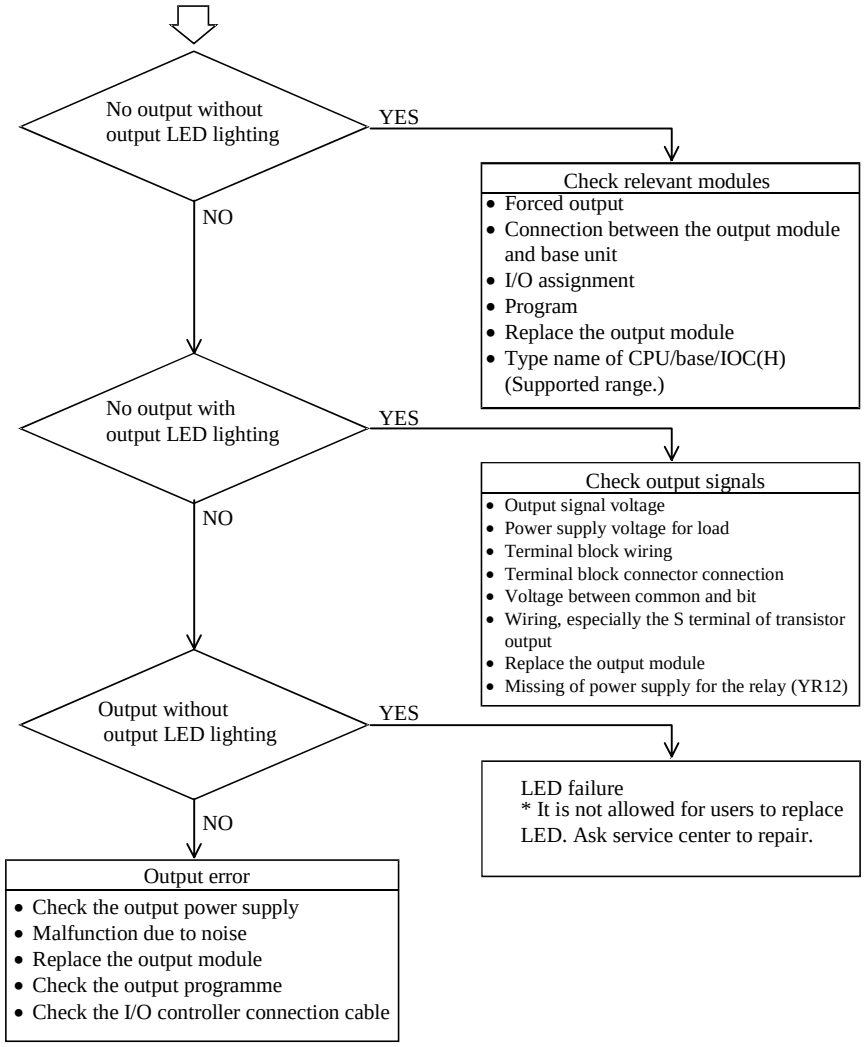


[I/O assignment error is generated, but data is read.]

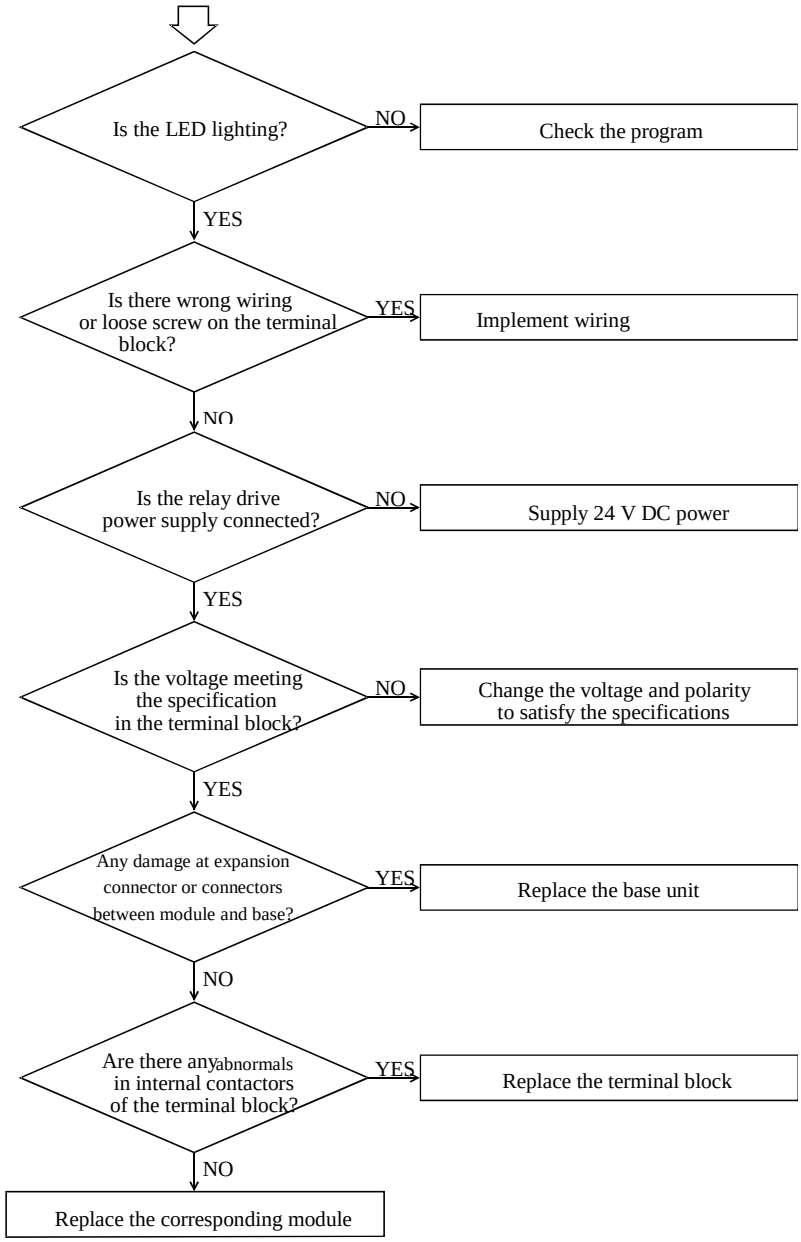


(e) Wrong output from output module or output module will not output (abnormal operation)

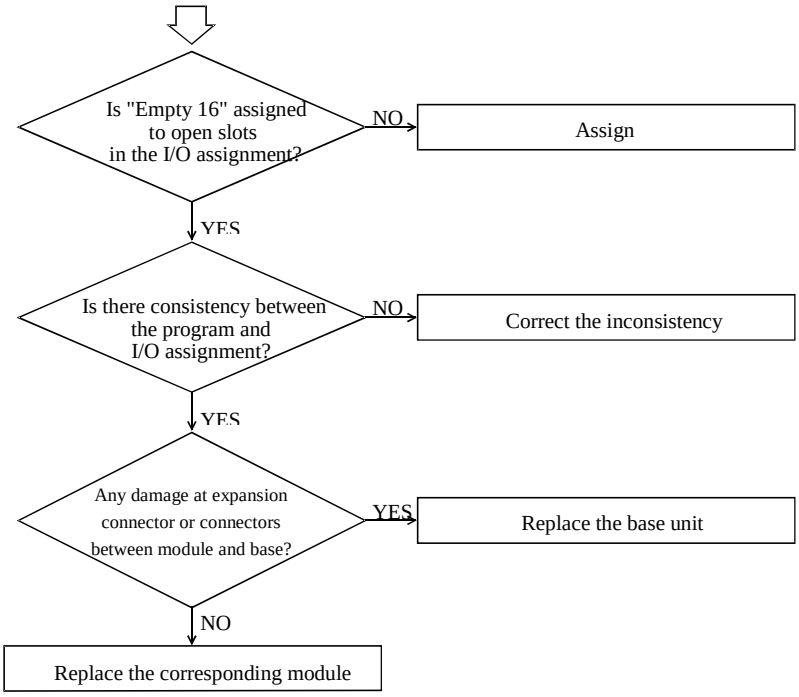
[The CPU operates, but output signals are not correct.]



[The CPU operates, but output signals are not detected.]

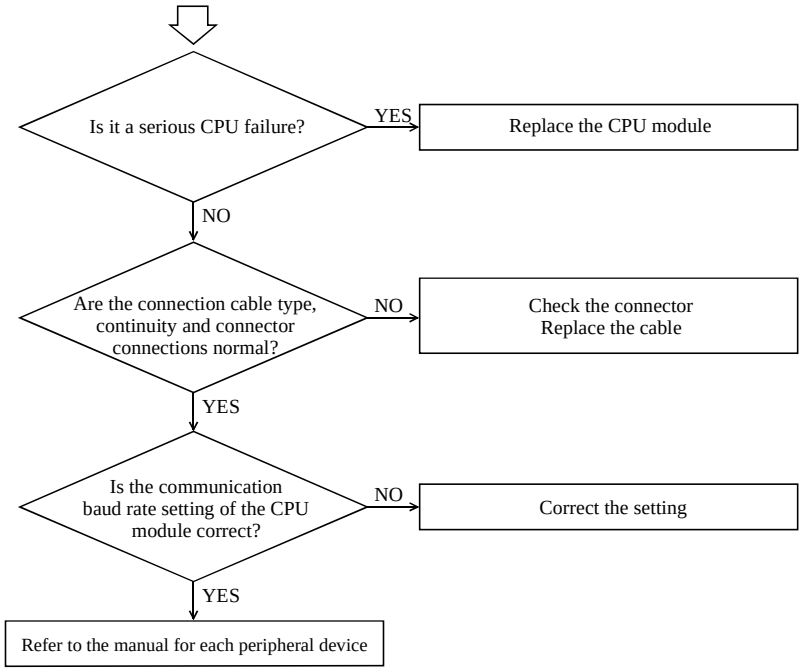


〔 I/O assignment error is generated, but output is normal. 〕



(f) Peripheral devices abnormal

〔 Peripheral devices cannot be connected. 〕



Chapter 15 Operation Examples

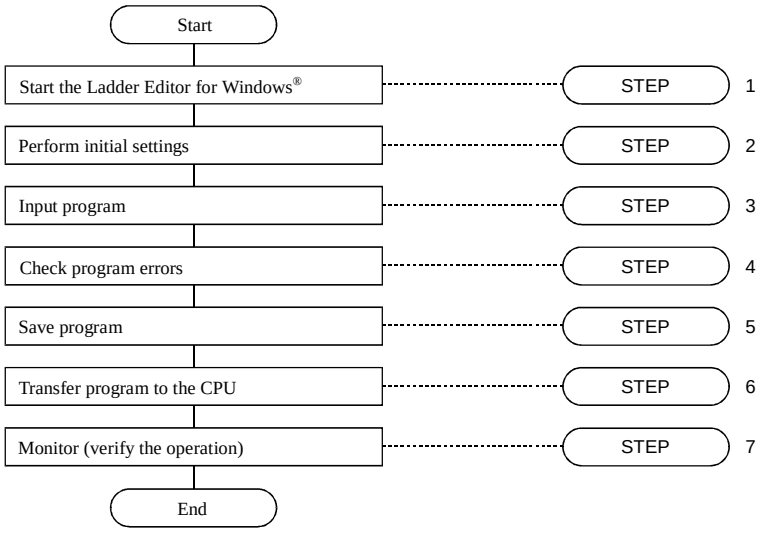
To understand the basic operation of the EH-150, this chapter explains samples of operations such as inputting simple programs and verifying operations.
The following programming devices can be used:

	Peripheral device name	Form
1	Portable graphic programmer	PGM-GPH
	ROM pack for portable graphic programmer	PGMPK2H
2	Command language programmer	PGM-CHH
3	H series ladder diagram command language software Ladder Editor	HL-PC3 HL-AT3E
4	H series ladder diagram command language software Ladder Editor for Windows® version	HLW-PC3 HLW-PC3E

* The graphic input unit (model name: GPCL01H) can be used. However, the PGMIF1H cannot be used.

(1) Operation verification procedures

An operation is verified according to the following procedures:



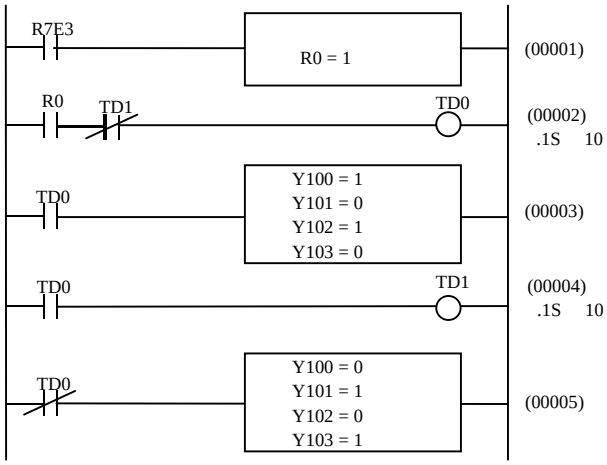
A personal computer and Ladder Editor for Windows® are used as the peripheral devices in the example. For details, refer to the user's manual for each peripheral device.

(2) Detailed operation example

The following explains an operation example using the module and sample program from step 1.

Power supply: EH-PSA
Basic base: EH-BS3
CPU: EH-CPU448
Slot 0: EH-XD64(64-point input)
Slot 1: EH-YT32(32-point output)
Slot 2: EH-DUM(dummy)

Operation of program
Turn Y100 and Y 102 on and Y101 and Y103 off and vice versa, alternating at one second intervals.



STEP 1 Starting the Ladder Editor for Windows®

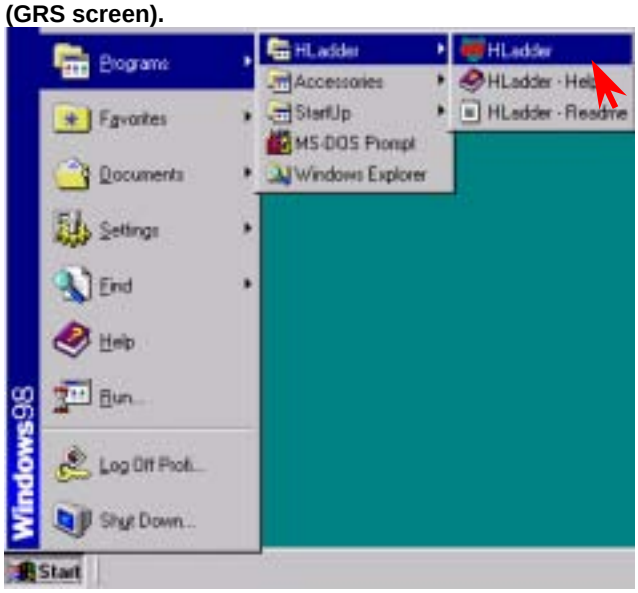
1. Start the personal computer.

Start the personal computer.

2. Start the Ladder Editor for Windows® system (GRS screen).

From the Start menu of Windows®, click **[Program] → [Hladder] → [Hladder].**
As Ladder Editor for Windows® is started, the GRS screen is displayed.

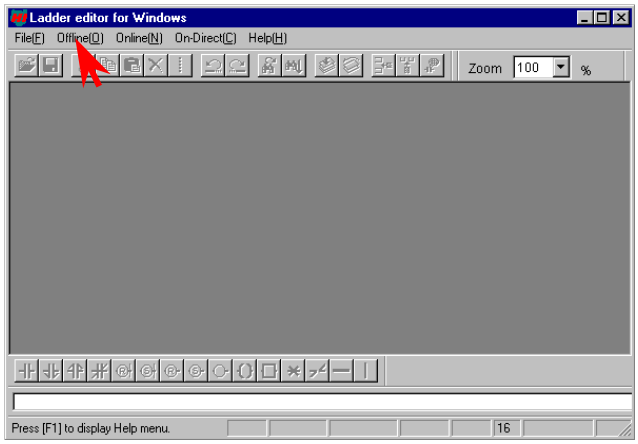
Startup



3. Switching to Offline mode.

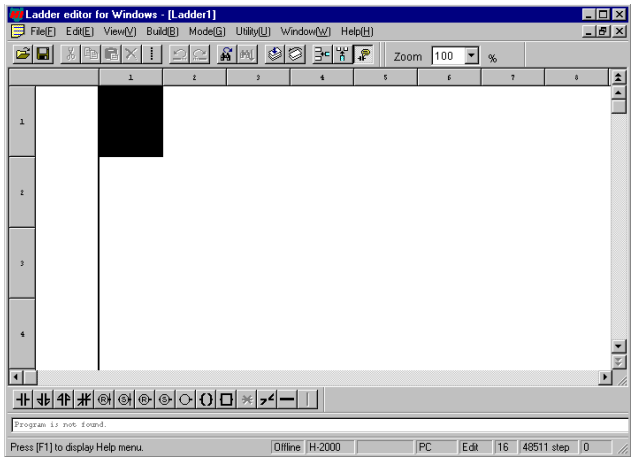
Click **[Offline]** in the Menu bar.

GRS screen



The Read/Edit screen is displayed.

Mode switching



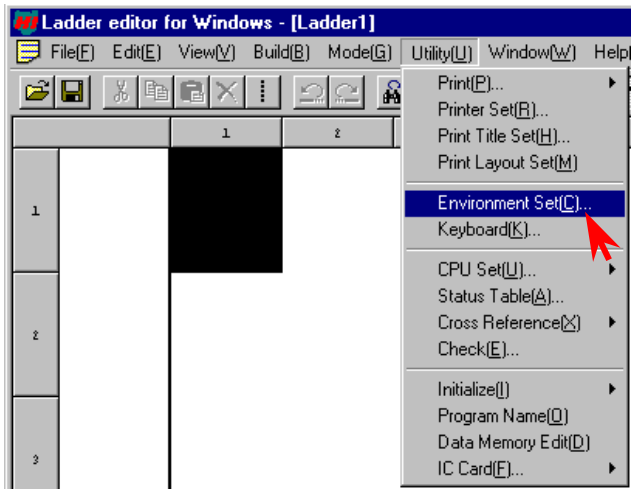
STEP 2 Initialization

Settings for the CPU type, memory type and I/O assignment are performed.

1. Setting the CPU type

Click **[Utility]** → **[Environment Settings]** in the Menu bar.

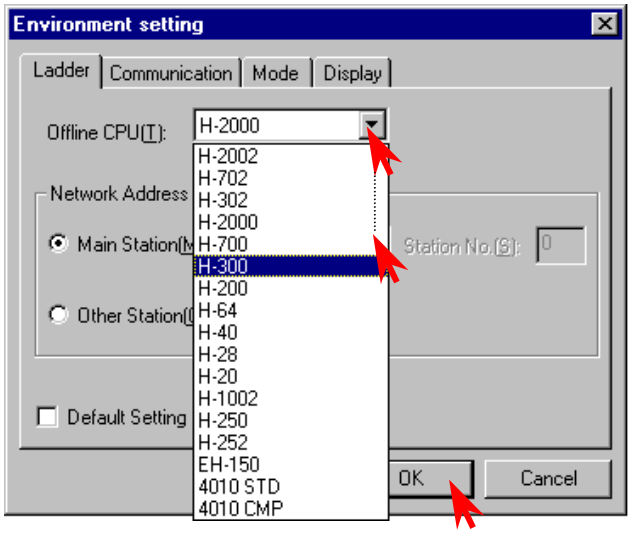
Pull-down menu



The Environment Setting dialogue box is displayed. Specify the CPU type from the Ladder tag.

- Click the ▼ of the Offline CPU field to show the available CPU types in the pull-down display. Select the CPU type.
- Click the **[OK]** button.

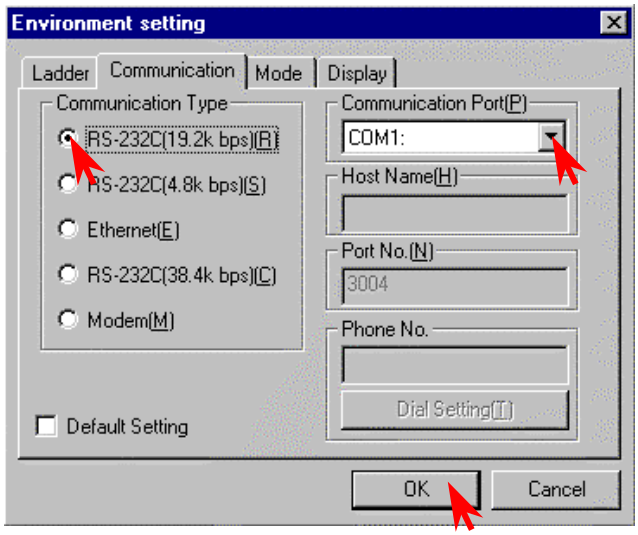
Pull-down display of CPU types



Specify the transmission speed from the Communication tag.

- Select the transmission speed set with the DIP switches of the EH-150 main unit (in case of 10-point type CPU, the transmission speed is fixed at 4800bps).
- Specify the communication port.
- Click the **[OK]** button.

Pull-down display of CPU types

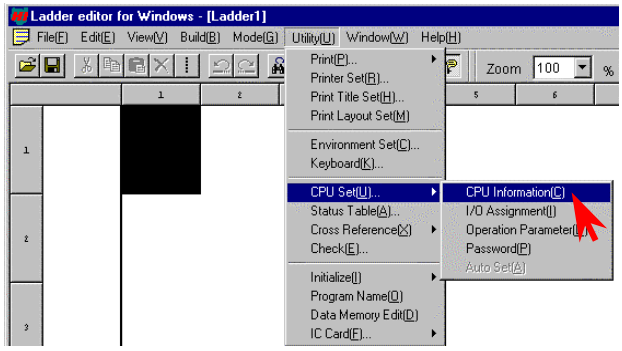


For the CPU type, select “EH-150.” (However, if LADDER EDITOR for Windows® of earlier than Ver. 2.0, HL-PC3, or GPCL01H is used, select “H-302.”)
* If the EH-CPU448 is used, select “H-302.” (If “EH-150” is selected, 48K steps cannot be specified for the memory type.)

2. Setting the memory type

Click **[Utility]** → **[CPU Setting]** → **[CPU Information]** in the Menu bar.

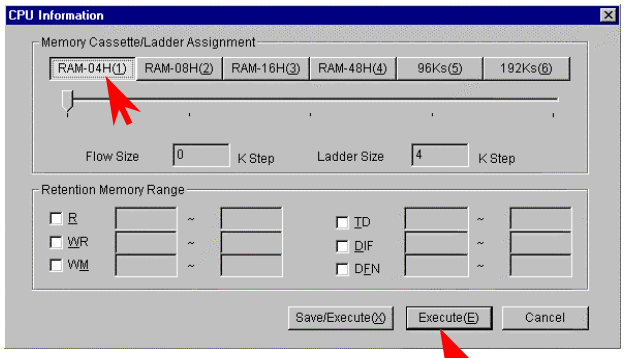
Pull-down menu



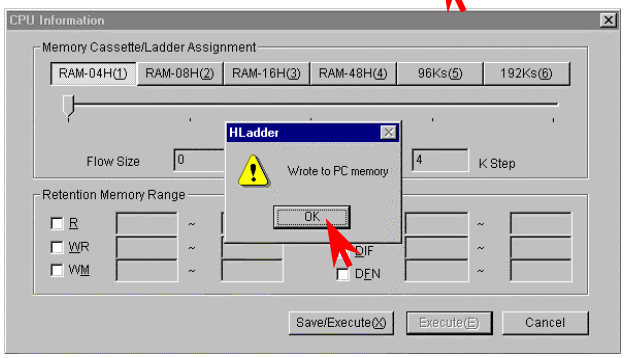
The CPU Information dialogue box is displayed.

- Click the Memory Cassette/Ladder Assign button and select the memory cassette size.
- Click **[Execute]** or the **[Memory/Execute]** button.

CPU Information dialogue box



- Click the [OK] button in the confirmation dialogue box.

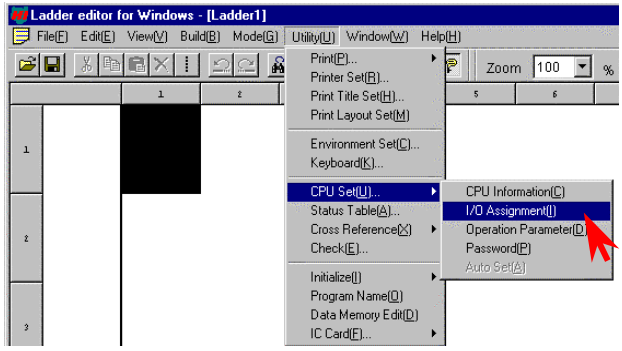


Select the memory cassette size according to the CPU model by referring to Section 7.2, If the EH-CPU448 is used, select “H-302” for the CPU type.
[Execute]: Save to the PC memory
[Memory/Execute]: Save to the PC memory and Window registry.

3. Assigning I/O

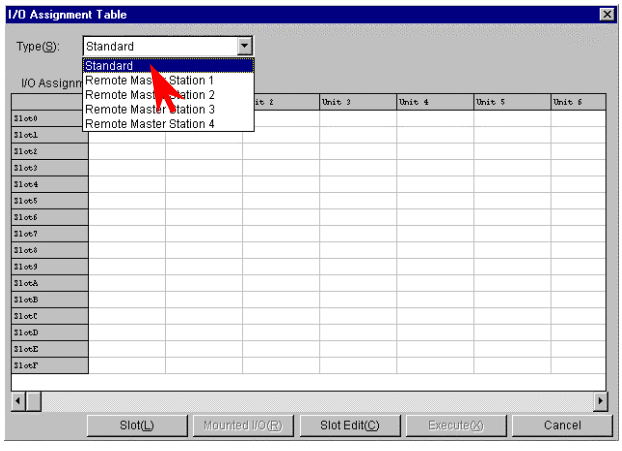
Click **[Utility]** → **[CPU Setting]** → **[I/O Assign]** in the Menu bar.

Pull-down menu



The I/O Assign List dialogue box is displayed.
Click the ▼ of the Types field and select **[Standard]** from the pull-down display.

I/O Assign List dialogue box



There are two setting methods for the subsequent procedures.

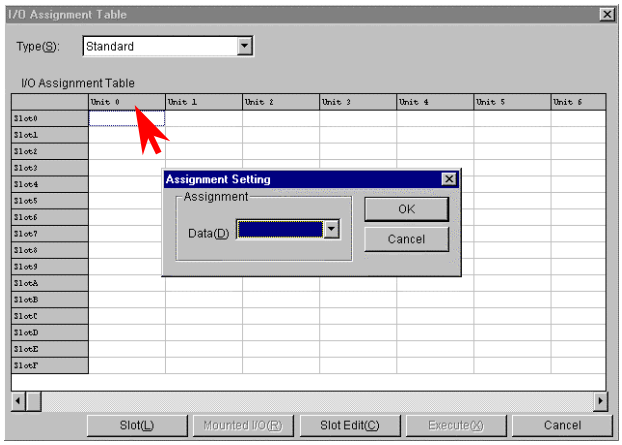
- From the I/O Assign List
- From the I/O Assign List → Slot Setting Status

[Setting from the I/O Assign List]

1] Double-click the cell for the unit number and slot number to be set.

The Assignment Setting dialogue box is displayed.

The Assignment Setting dialogue box.

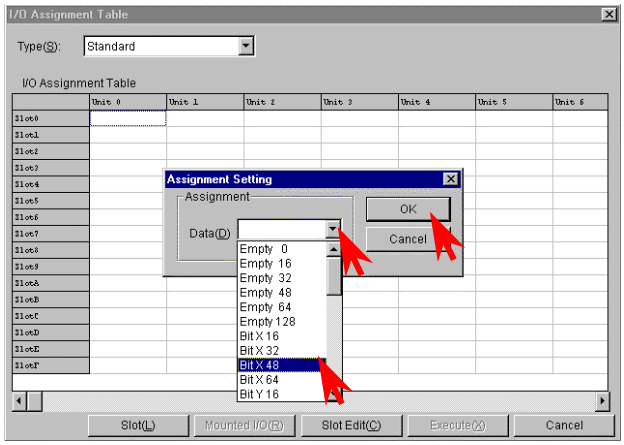


2] Click the ▼ of the data and select I/O type from the pull-down display.

3] Click the **[OK]** button to close the Assignment Setting dialogue box.

I/O type is set to the specified cell.

Setting of I/O type

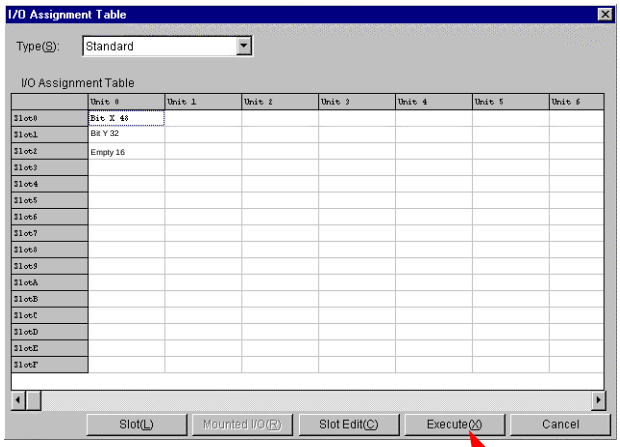


In the same way, repeat steps 1] to 3] to assign Y32 and 16 vacant points to Slot 1 and 2 respectively.

If a wrong value has been entered, the slot is left blank by assigning [Vacant 0] and is treated as though nothing is assigned to it.

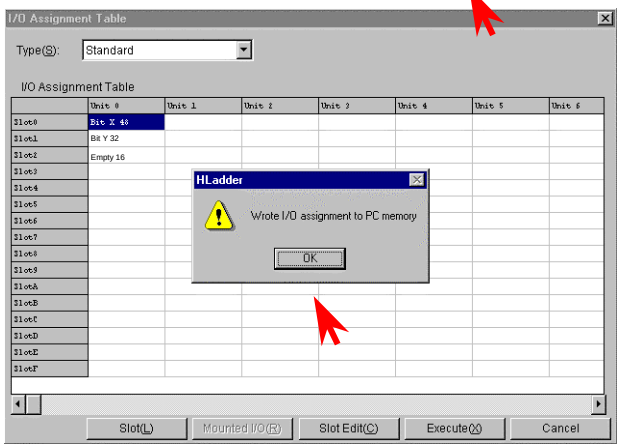
4] Click the **[Execute]** button.

The information assigned to the PC memory is written.



5] Click the **[OK]** button in the confirmation dialogue box to close the I/O Assignment List dialogue box.

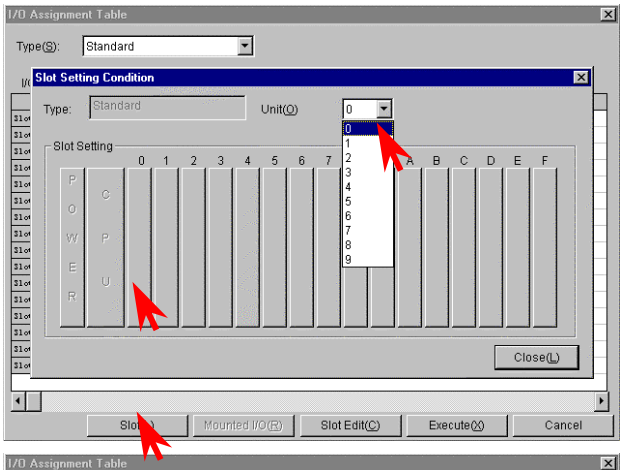
Confirmation dialogue box



[Setting from the Slot Setting Status]
Click the **[Slot]** button to display the Slot Setting Status dialogue box.

1] Click the ▼ of the unit and select the unit number from the pull-down display.
2] Click the button of the slot number to be set.

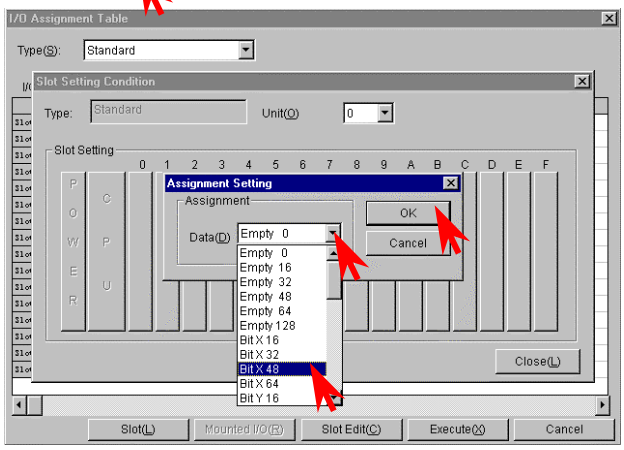
Slot Setting Status dialogue box.



3] Click the ▼ of the data and select the I/O type from the pull-down display.
4] Click the **[OK]** button and close the Assignment Setting dialogue box.

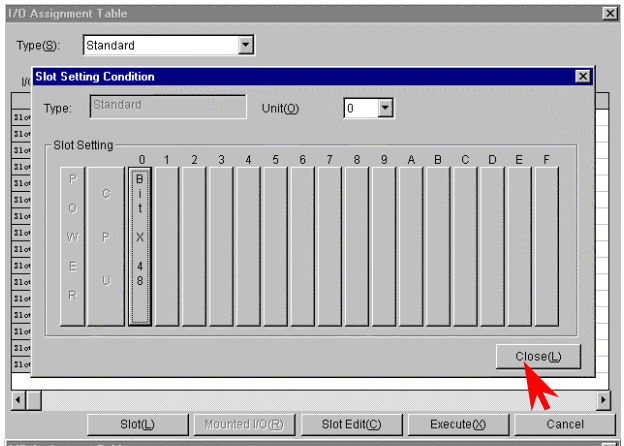
I/O type is set to the specified slot no.

Specification of I/O type



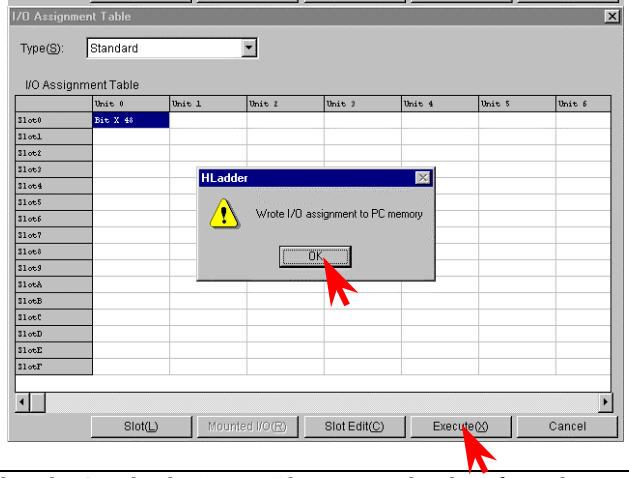
In the same way, repeat the steps 1] and 2] to 4] to set other unit and slot numbers in order to perform I/O assignment according to the unit to be used.
In this example, Y32 and 16 vacant points are assigned to slots 1 and 2 respectively.

5] Click the **[Close]** button to close the Slot Setting Status dialogue box.
Enter the I/O assignment set in the Slot Setting Status into the I/O Assignment List.



6] Click the **[Execute]** button to write the assigned information to the PC memory.
7] Click the **[OK]** button in the confirmation dialogue box to close the I/O Assignment List dialogue box.

Confirmation dialogue box

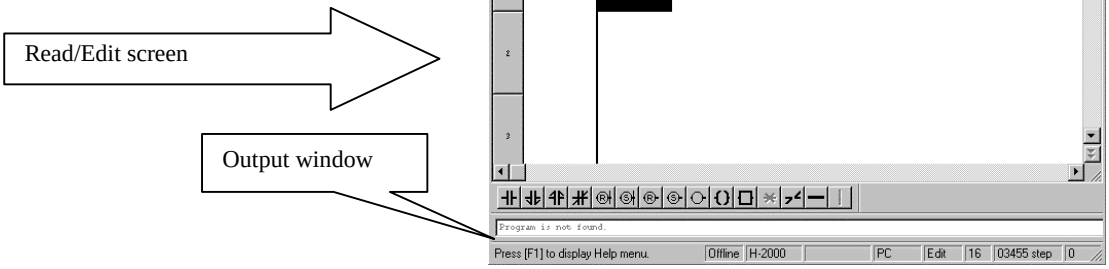


For online mode, it is possible to read the I/O mounted on the CPU by the “Mount” button. For details, refer to the “Reading Mounted I/O” of the programming device.

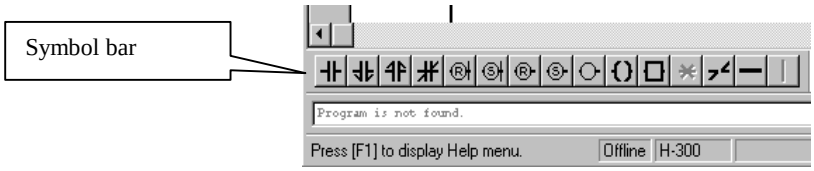
STEP 3 Program input

1. Input a program.

At first, the output window displays “there is no program” in the bottom left of the Read/Edit screen.
The cursor ■, which indicates the program input position, is placed at the top left of the screen.

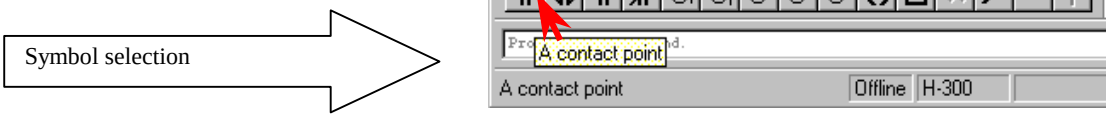


[Input procedure of ladder program]
Repeat steps 1] to 4] to proceed with symbol input. The usual operations found in other Windows applications, such as cut, copy, paste, and move, can be performed on already input symbols.
1] Specify the input position. (Move the cursor ■ by clicking the mouse or the arrow keys.)
2] Click symbols in the Symbol bar.

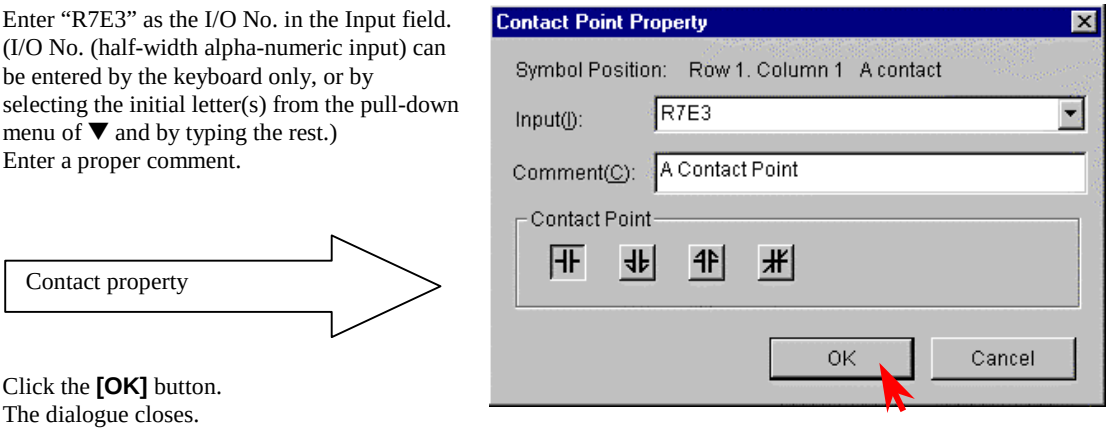


3] Input the desired function (I/O, comparison expression, arithmetic expression) in the dialogue box for the symbol displayed.
4] Click the **[OK]** button in the dialogue box.

[Example of entering a contact]
1] Begin from the cursor position at the top.
2] Click the symbol for contact A.
The dialogue box for contacts is displayed.



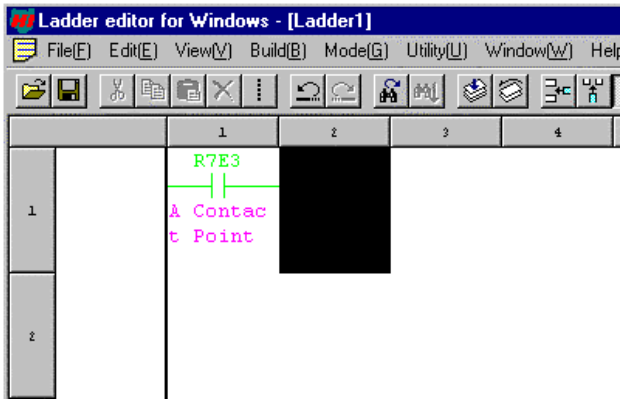
3] Enter “R7E3” as the I/O No. in the Input field. (I/O No. (half-width alpha-numeric input) can be entered by the keyboard only, or by selecting the initial letter(s) from the pull-down menu of ▼ and by typing the rest.)
Enter a proper comment.



4] Click the **[OK]** button.
The dialogue closes.

When the dialogue box closes, the symbol is displayed in the Read/Edit screen and the cursor shifts.

Display of symbol



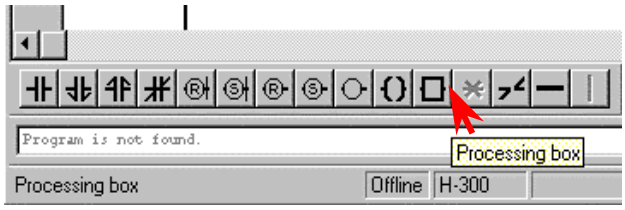
The comment is displayed under the symbol.

[Example of entering a Processing Box]

1] The specification of the input position can be omitted when entering symbols into the same ladder as the contact above.

2] Click the symbol for Processing Box.

Symbol selection

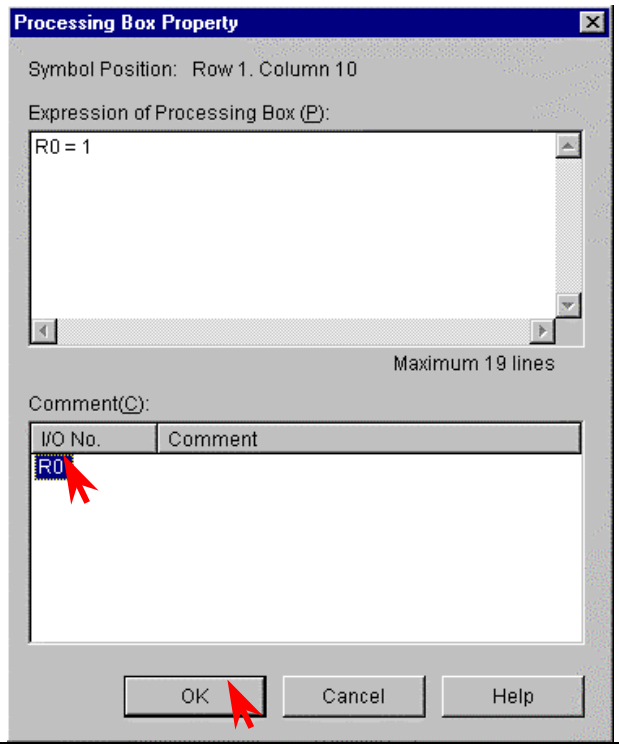


The cursor moves to the far-right portion of the screen automatically.
The dialogue box for the processing box symbol is displayed.

3] Input arithmetic expressions in the Expression in Processing Box text field.

Multiple lines (a maximum of 19) can be input by including line breaks

Processing Box properties

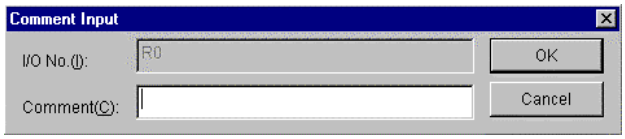


The comment for the I/O No. written to the Processing Box is displayed by clicking the Comment column.
If there are no comments, only the I/O No. is displayed.

Always enter a space before and after "=".

- The Comment Input dialogue box is displayed by double-clicking the I/O No. displayed in the Comment column.
- Input a comment and click the **[OK]** button.

Comment Input dialogue box



4] Click the **[OK]** button in the Processing Box.

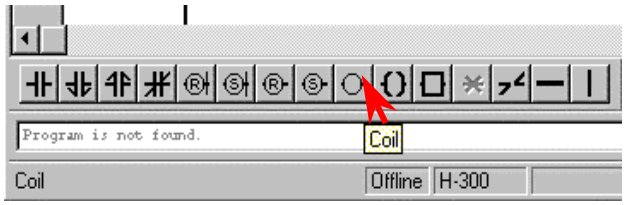
The input of the horizontal line symbol, which connects between symbols, may be omitted. (Symbols are connected by horizontal lines by the automatic wiring function at ladder write.)

[Example of entering a timer]

1] Specify the input position, or omit the specification if entering it in the same ladder.

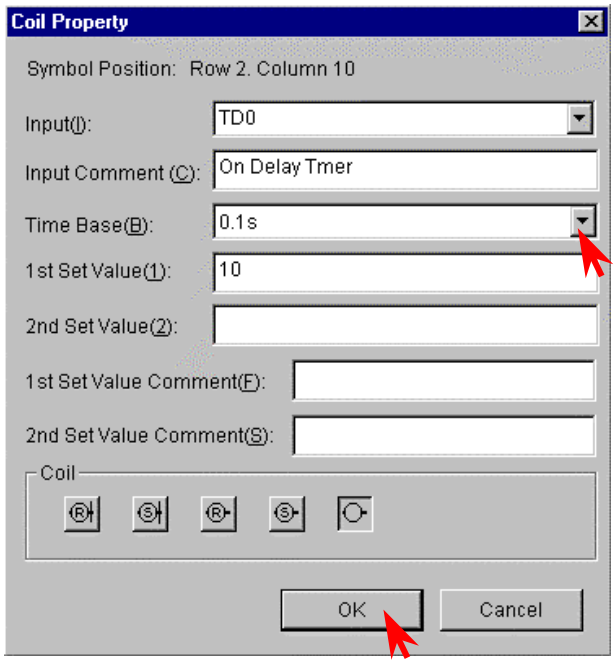
2] Click the symbol for coil.
When the specification of the input position is omitted, the cursor automatically moves to the far-right portion of the screen.

Symbol selection



3] Input I/O No., time base, and the first setting value.

Coil property



The following initials of various I/O numbers can be selected from the pull-down display of the Input field:

R, L, M, Y, TD, SS, WDT, MS, TMR, CU, RCU, CTU, CTD, CL

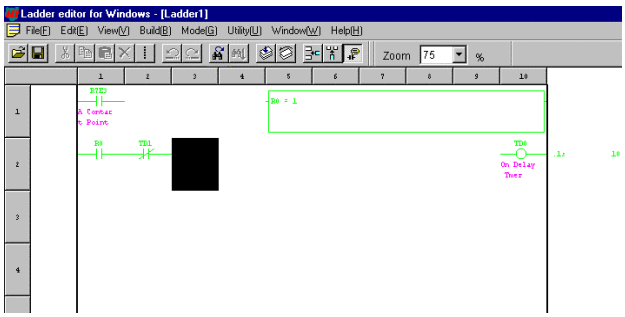
Input values in the necessary items, such as the time base, the first setting value, and second setting value, according to the I/O No.

(Example) Coil
It is only necessary to enter values in the Input and Comment items.

4] Click the **[OK]** button to display the symbol at the cursor at the far-right portion of the ladder.

Symbols whose input positions for coils, arithmetic expressions, etc. are determined are automatically flushed to the right.

Display of symbols



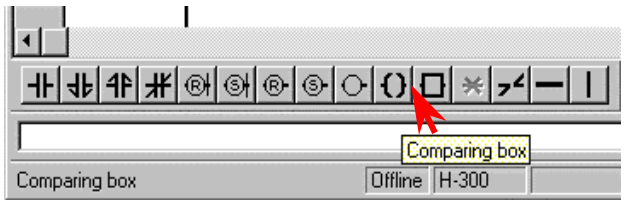
After displaying the coil, the cursor moves to the top of the next ladder.

[Example of entering a Comparison Box]

1] Specify the input position

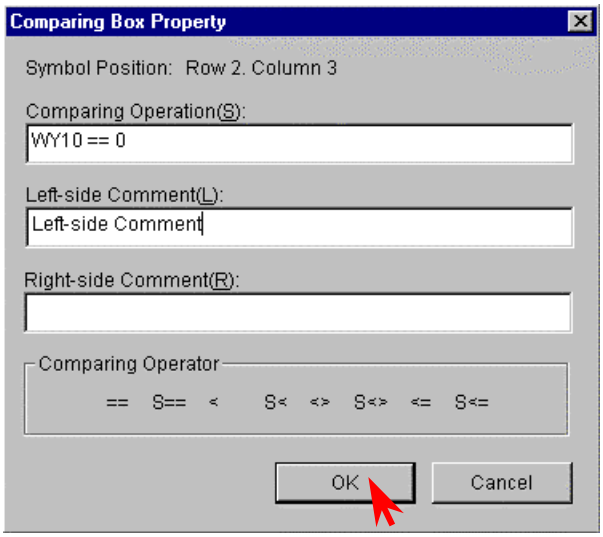
2] Click the symbol for Comparison Box.

Symbol selection



3] Input comparison expression and comment.
4] Click the **[OK]** button.

Comparison Box property

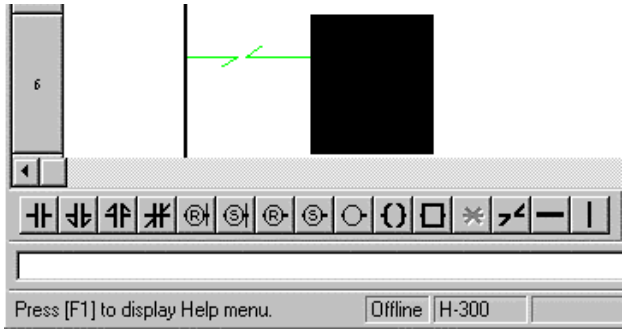


The comment input is valid only for I/O numbers. In this example, entering a comment for the value on the right side of the expression will not generate a comment.

Always enter a space between an I/O number and comparison operator (in this case, between "WY10" and "= ="), as well as between a comparison operator and comparison data ("= =" and "0").

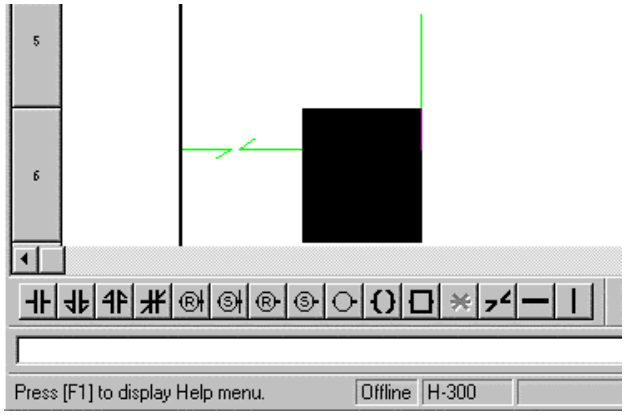
[Example of entering a Knot]
1] Specify the input position.
2] Click the symbol for Knot.
The symbol is displayed and the cursor moves to the right.

Display of symbols



[Example of entering a Vertical Line]
1] Specify the input position.
2] Click the symbol for Vertical Line.
The symbol is displayed on the right side of the cursor.
The cursor does not move.

Display of symbol



In case of the Horizontal Line symbol, the cursor does move to the right after displaying the symbol, in the same way as in the Knot symbol.

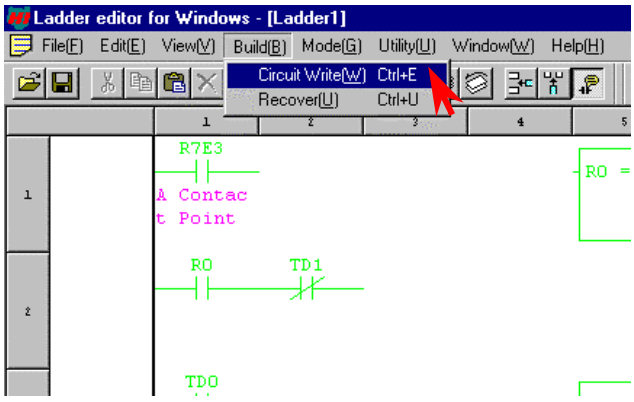
2. Writing to the program memory

Perform a "ladder write" operation by either of the following methods in order to write the ladder to the program memory.

1] Click **[Build]** → **[Ladder write]** in the Menu bar.

2] Click the **[ladder write]** icon  in the tool bar.

Ladder write operation

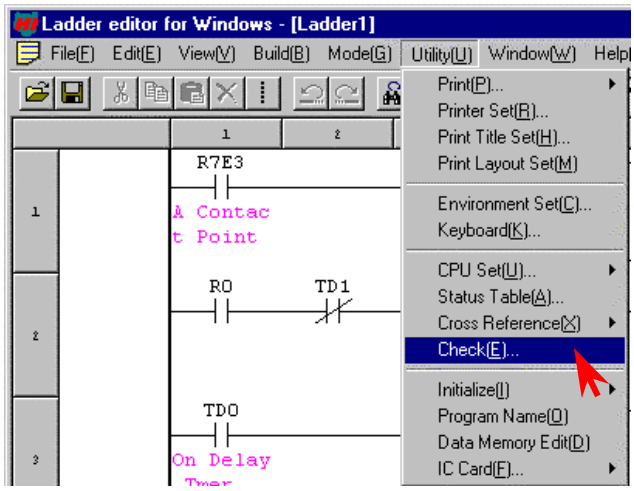


STEP 4 Checking program errors

Check to see if the program in the memory is correct.

Click **[Utility]** → **[Check]** in the Menu bar.
The Check dialogue box is displayed.

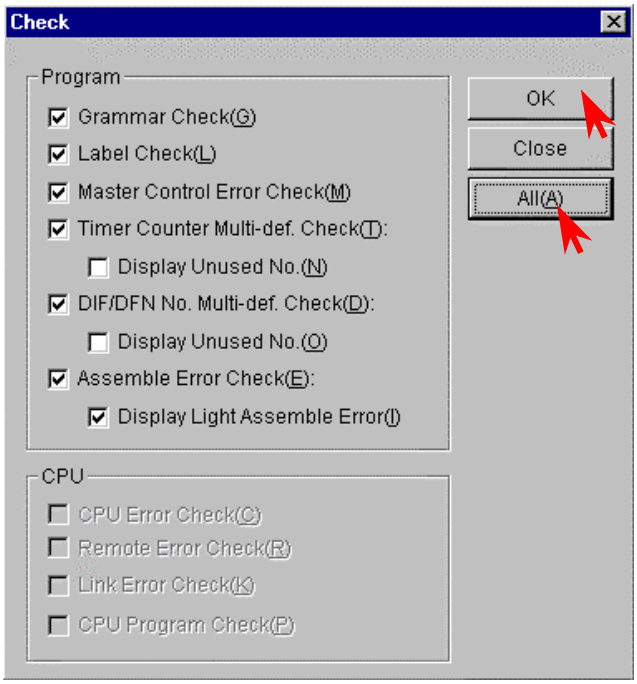
Pull-down menu



- Click the **[All items]** or the individual check column to specify the items to be checked.
- Click the **[Execute]** button.

The Check Result dialogue box is displayed.

Check dialogue box

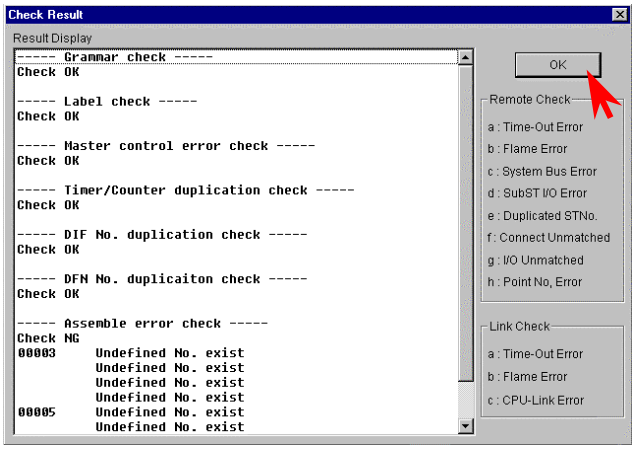


The checking of the CPU can be specified at online mode.

- Click the **[OK]** button.

The Check Result dialogue box closes.

Check Result dialogue box

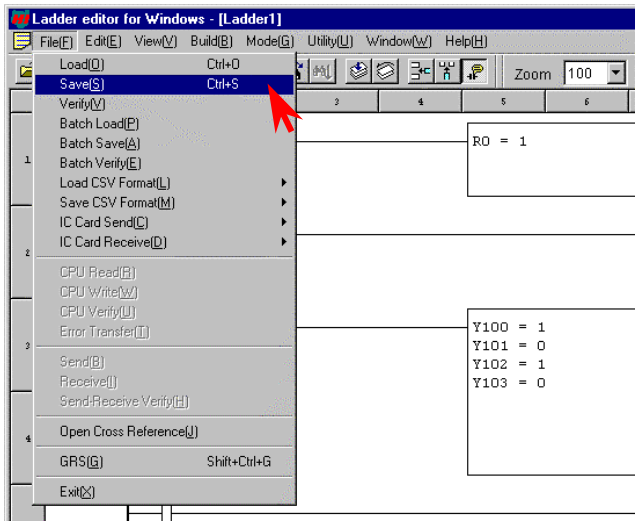


(Note)
For example, if the I/O assignment of bit Y32 is missing for unit 1, WY10 of the sample is treated as undefined; the error is displayed as in the figure to the right.
If there are any errors, correct the errors of the program before check the program again.

STEP 5 Saving the program

Save the program and comment that has been created to a floppy disk.

Click **[File]** → **[Record]** in the Menu bar, the Record icon  , or **[File]** → **[Batch Record]**. The dialogue for Record or Batch Record is displayed.

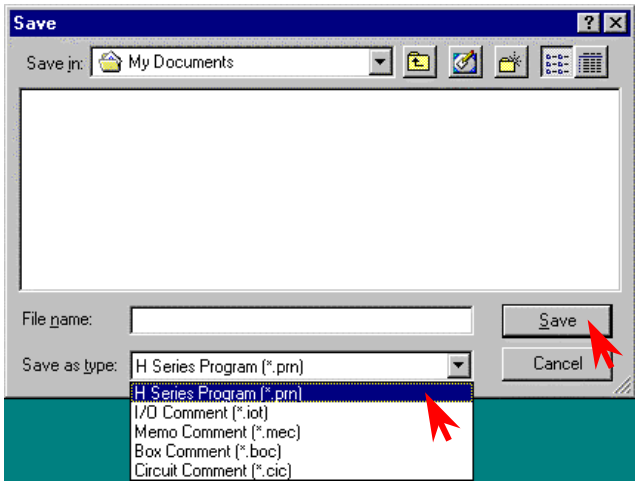


Pull-down menu.

Record  : Specify the file type and save.
Batch Record: Saves a program and all the comment files.

Record dialogue box:
Specify the directory to save in, file name, and file type.
Batch Record dialogue box:
Specify the place to save and file name.

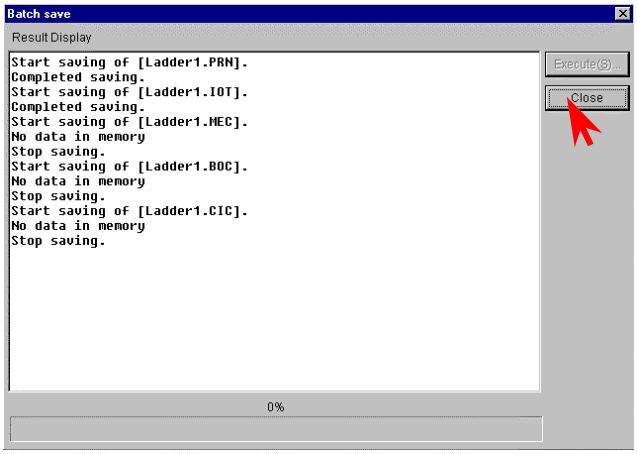
Click the **[Save]** button to save.



Record dialogue box

File name extensions are not necessary to input.

Record and Batch Record display the results of the save operations for one file and five files respectively.
The figure to the left shows an example of a result display for the Batch Record.



Batch Record Result dialogue box.

STEP 6 Program transfer to CPU

Write the program that has been input, to the CPU. However, verify the following:

- The CPU and the personal computer connection cable are properly connected.
- The CPU power is on.
- CPU mode switch is set to "STOP."

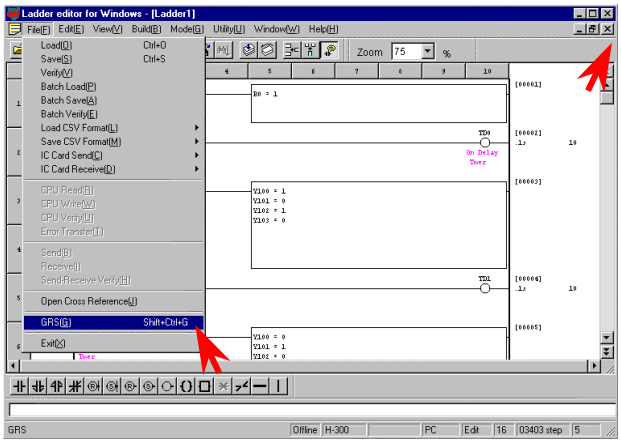
1. Switching to online mode.

Move to the GRS screen from the offline mode.
This can be done in two ways.

1] Click **[File]** → **[GRS]** in the Menu bar.

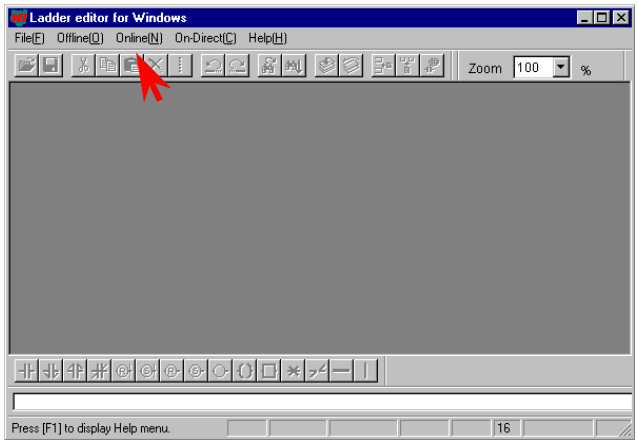
2] Click ✕ (lower button) on the upper right of the screen.

GRS screen



In the GRS screen, click the **[Online]** item in the Menu bar.
The Read/Edit screen of the online mode is displayed.

GRS screen

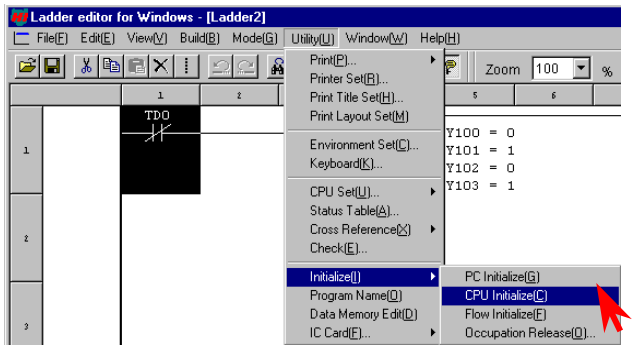


Note) Verify again that the DIP switches are set to the transmission speed selected in the Environment Setting in step 2. (For 10-point type, it is fixed to 4,800 bps.)

2. Initializing the CPU

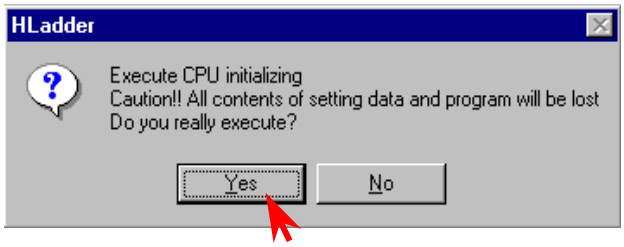
Click **[Utility]** → **[Initialize]** → **[CPU initialize]** in the Menu bar.

Pull-down menu

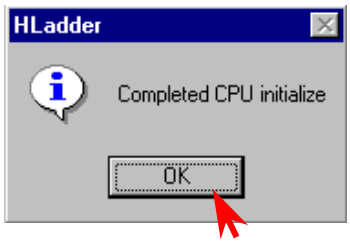


Note) Please note that programs etc. in the personal computer will be erased if [PC initialize] is selected.

The Confirmation dialogue box is displayed; click the **[Yes]** button and start the CPU initialization.



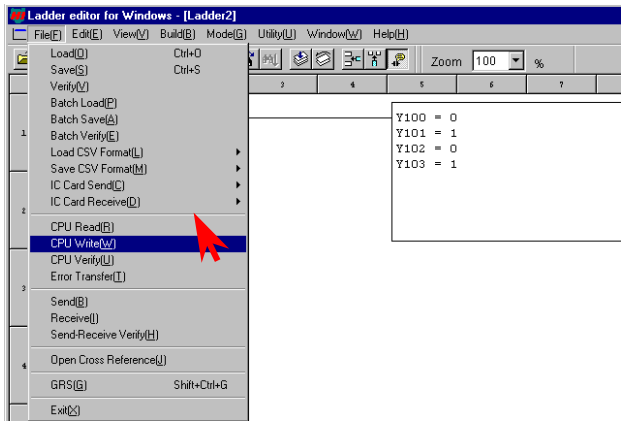
The Exit dialogue box is displayed; click the **[OK]** button to close the dialogue.



3. Transferring to the CPU

Click **[File]** → **[CPU write]** in the Menu bar.

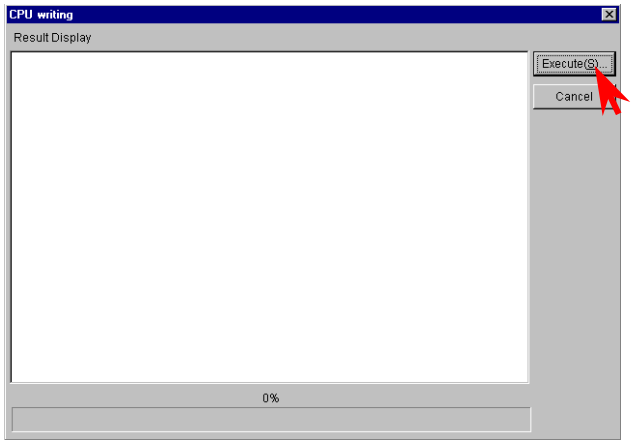
Pull-down menu



Program transfer
CPU Read: PC (personal computer) ← CPU
CPU Write: PC (personal computer) → CPU

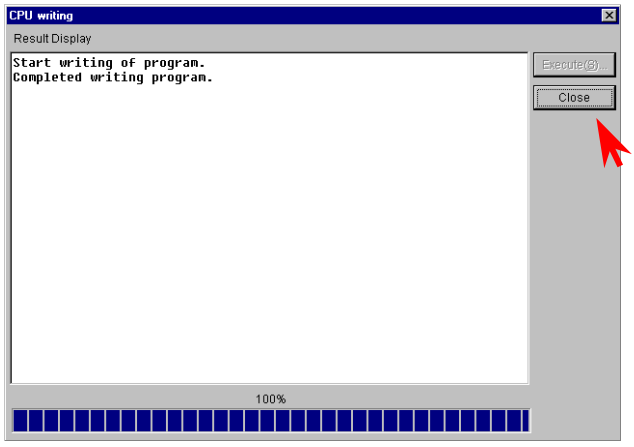
The CPU Write dialogue box is displayed.
Click the **[Execute]** button.

CPU write dialogue box



When the writing is completed, the result is displayed.
Click the **[Close]** button to close the dialogue box.

Display of write result.

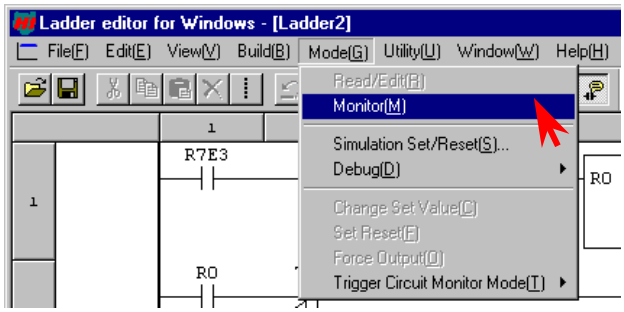


STEP 7 Monitoring (verifying the operation)

Monitor the program execution status in the CPU.

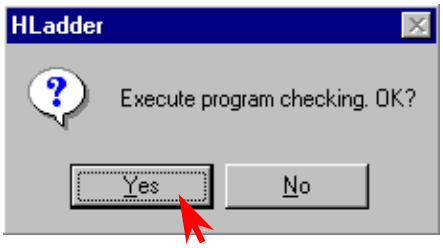
[Ladder monitor]
Click **[Mode]** → **[Monitor]** in the Menu bar.

Pull-down menu



The Confirmation dialogue box for the program match check between PC and the CPU is displayed.
Click the **[Yes]** button.

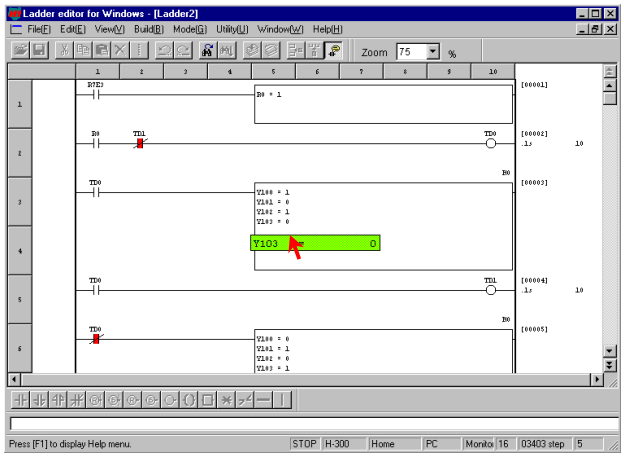
Match check



Set the CPU's RUN switch to “RUN” to begin the CPU operation.

The on/off status of the contact, timer, and current counter value are displayed.

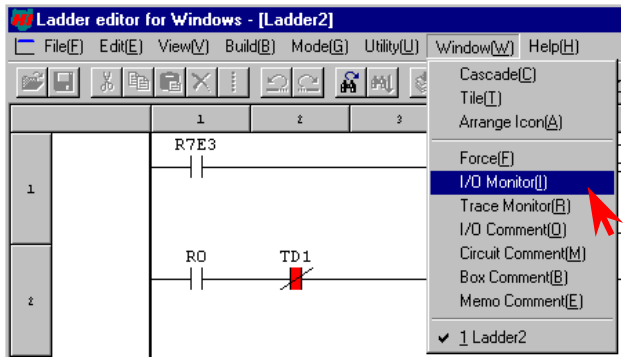
Display of ladder monitor



To monitor and display the current value and progress value, select comparison expression, arithmetic box, and coil (timer, counter, etc.) with the mouse arrow.

[I/O monitor]
The I/O monitor can be operated while in monitor mode.
Click **[Window]** → **[I/O Monitor]** in the Menu bar.
The I/O Monitor dialogue box is displayed.

Pull-down menu



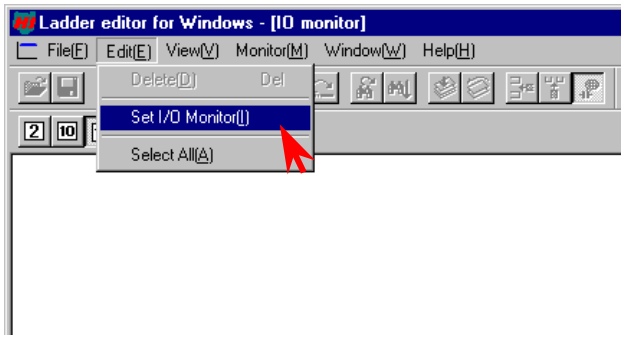
The I/O Monitor dialogue box is displayed on the Read/Edit screen at its maximum size.

The I/O monitor can be specified in the following two ways.

1] Click **[Edit]** → **[I/O monitor setting]** in the Menu bar.

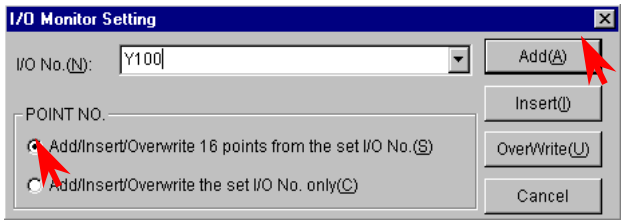
2] Click the  icon in the Symbol bar.

I/O Monitor dialogue box



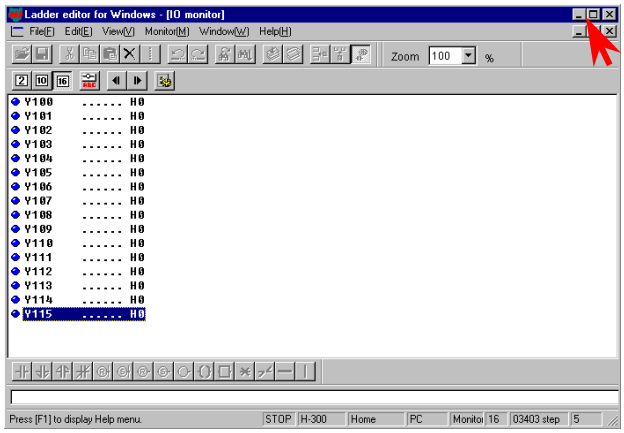
- Enter the starting I/O No.
- Click the number of points to be monitored.
- Click on either the **[Add]**, **[Insert]**, or **[Overwrite]** buttons.

I/O Monitor Setting dialogue box.



Monitor and display 16 points from Y100.

I/O monitor



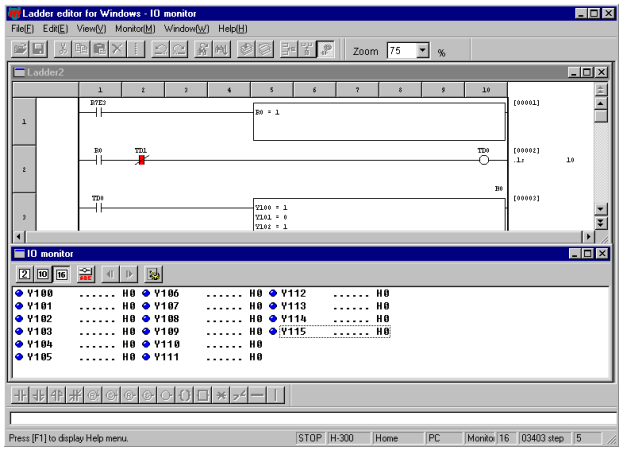
The I/O monitor can display up to 64 I/O points (up to 64 including words/double-words).

Click the I/O No. being I/O monitored and click **[Edit]** → **[Delete]** to delete it from the monitor.

The display size of the I/O Monitor dialogue box can be changed by clicking  .

Both the ladder monitor in the Read/Edit screen and the I/O Monitor can be displayed by making their display sizes smaller to check the operation.

Display of ladder and I/O monitor



Chapter 16 Daily and Periodic Inspection

In order to use the EH-150 functions in the most desirable condition and maintain the system to operate normally, it is essential to conduct daily and periodic inspections.

(1) Daily inspection

Verify the following items while the system is running.

Table 16.1 Items for daily inspection

Item	LED display	Normal status	Main cause of error
LED of Power supply	POW	Lighting	Power supply error, etc.
LED of CPU *1	RUN	Lighting (running)	Off: Microcomputer malfunction, memory error, etc. Flashing: Grammar error, watchdog error, etc.
	ERR	off	Lighting: Microcomputer malfunction, memory error, etc. Flashing: Battery error *2

*1 The EH-150 indicates error contents by lighting pattern of RUN/ERR LED. Refer to the Chapter 13 error code list for further information.

*2 If PLC power is off over one week after battery error indication, memory data may be destroyed. If PLC power is off for a long time, battery error might not be detected. In this case, memory data is lost.

(2) Periodic inspection

Turn off the power for the external I/O ladder and check the following items once every six months.

Table 16.2 Items for periodic inspection

Part	Item	Check criteria	Remarks
Programming device to CPU	Check operation of programming device	All switches and display lamps work normally.	
Power supply	Check for voltage fluctuations	85 to 264 V AC	Tester
I/O module	Output relay life	Electrical life 200,000 times Mechanical life 10 million times	Refer to the relay contact life curve (Chapter 9).
	LED	working properly	
	External power voltage	Within the specification for each I/O module	Refer to the I/O module specifications(Chapter 4).
Battery (Lithium battery)	Check voltage, life	ERR LED not to flash Within 2 years after replacement	
Installation and connecting areas	(1) All modules are securely fixed (2) All connectors fit snugly (3) All screws are tight (4) All cables are normal	No defects	 Tighten Check insertion Tighten Visual check
Ambient environment	(1) Temperature (2) Humidity (3) Other	0 to 55 °C 20 to 90 % RH (no condensation) No dust, foreign matter, vibration	Visual check
Spare parts	Check number of parts, storage condition	No defects	Visual check
Program	Check program contents	Compare the contents of the latest program saved and CPU contents, and make sure they are the same	Check both master and backup.

(3) Life of the power module

Numbers of electrolytic capacitors are used in the power module. Electrolytic condensers have a lifetime and it is supposed that the life is reduced by half when the ambient temperature rise 10 °C.

When stocking spare parts, the standard for consideration is that the power module has a life of approximately 5 years when used at the rated ambient temperature (30 °C). Also, to lengthen the life of the module, consider the air circulation around the module and ambient temperature when installing it.

(4) Life of the battery

- The length of the battery life is expressed as the total time during which the power supply for the basic unit is off.
- The battery life can be determined by checking for the flashing of the ERR lamp.
- The battery life is also displayed in the bit special internal output “R7D9.”
  An example of a ladder using “R7D9” is shown below.

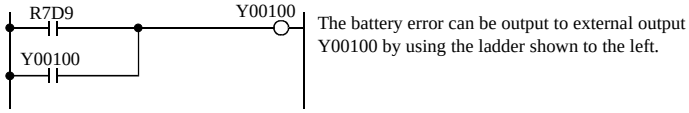


Figure 16.1 Battery error detection ladder

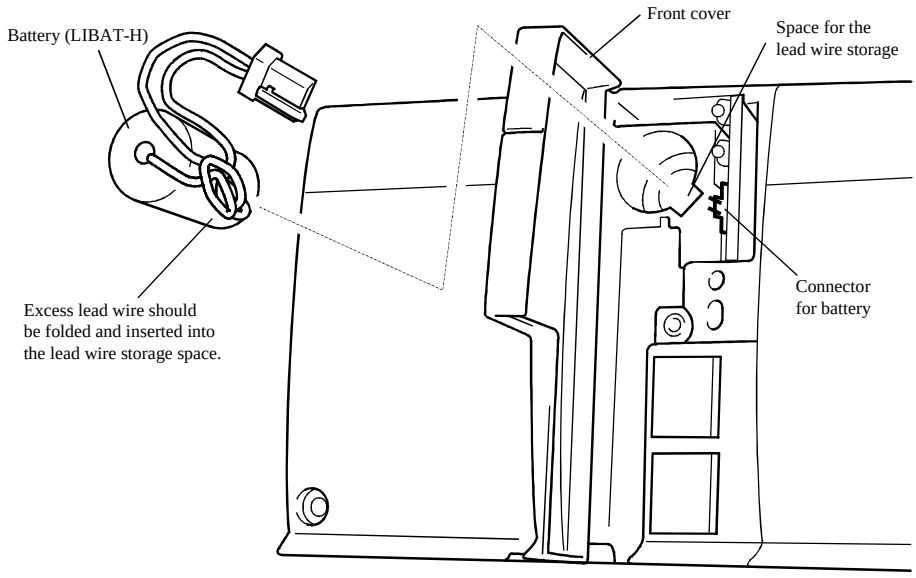
- The self-diagnostic error code “71” indicates that the battery is not loaded or that it has reached its life.

Use the reference table below to determine the remaining life of battery.
As a guideline, replace the battery every two years even when the total power failure time is less than the guaranteed value shown in the table.

Battery life (Total power failure time) [Hr]	
Guaranteed value (MIN) @55°C	Actual value (MAX) @25°C
2,000	32,000

(Note)
In case of CPU448(A)/516/548, battery error flag (R7D9) is set on if backup memory writing fails. If this bit R7D9 is still on after battery replaced, backup memory is probably broken.

(5) How to replace the battery



1] Prepare a new battery (LIBAT-H).

2] Confirm that the newest program is saved on floppy disks. If it is not saved, always save a backup of the program on floppy disks for safety purposes.

3] Replace the battery while the power supply to the basic base is turned on.

4] Remove the old lithium battery from the battery case, and remove the connector on the battery side.

5] Insert the connector on the battery side to the CPU module connector.
Insert it so that the red lead is ⊕, and the black lead is ⊖.

6] Fold the excess lead and store it in the lead storage space.
(If excess lead is not dressed properly, the wire may be severed by the front cover.)

* When exchanging while the basic base unit power turned off, perform steps 4], 5] and 6], in less than one minute.

⚠ Danger

Battery handling

Be sure to use LIBAT-H. Another battery may cause to explosion. The followings are strictly prohibited : inverted battery connection, charging, disassembling, heating, throwing into fire and short circuit.

⚠ Caution

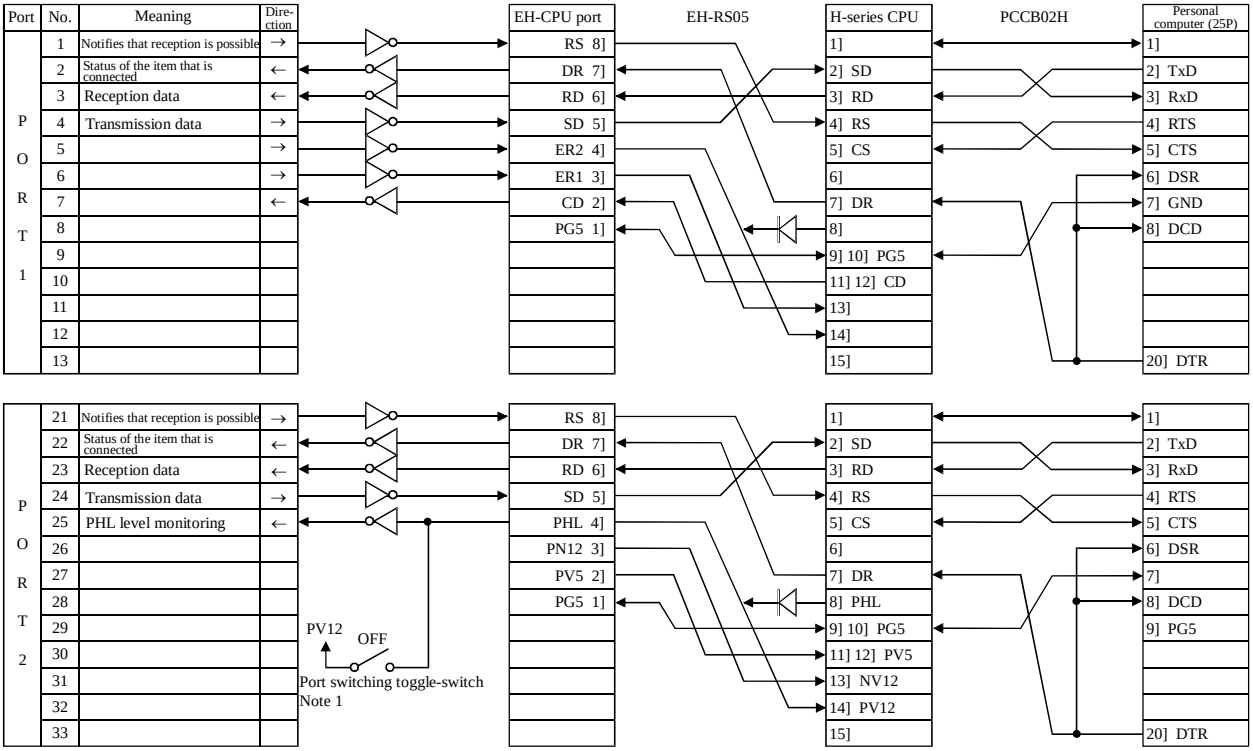
Battery discard (Recycling)

Exchanged old battery must be discarded properly according to local regulation.

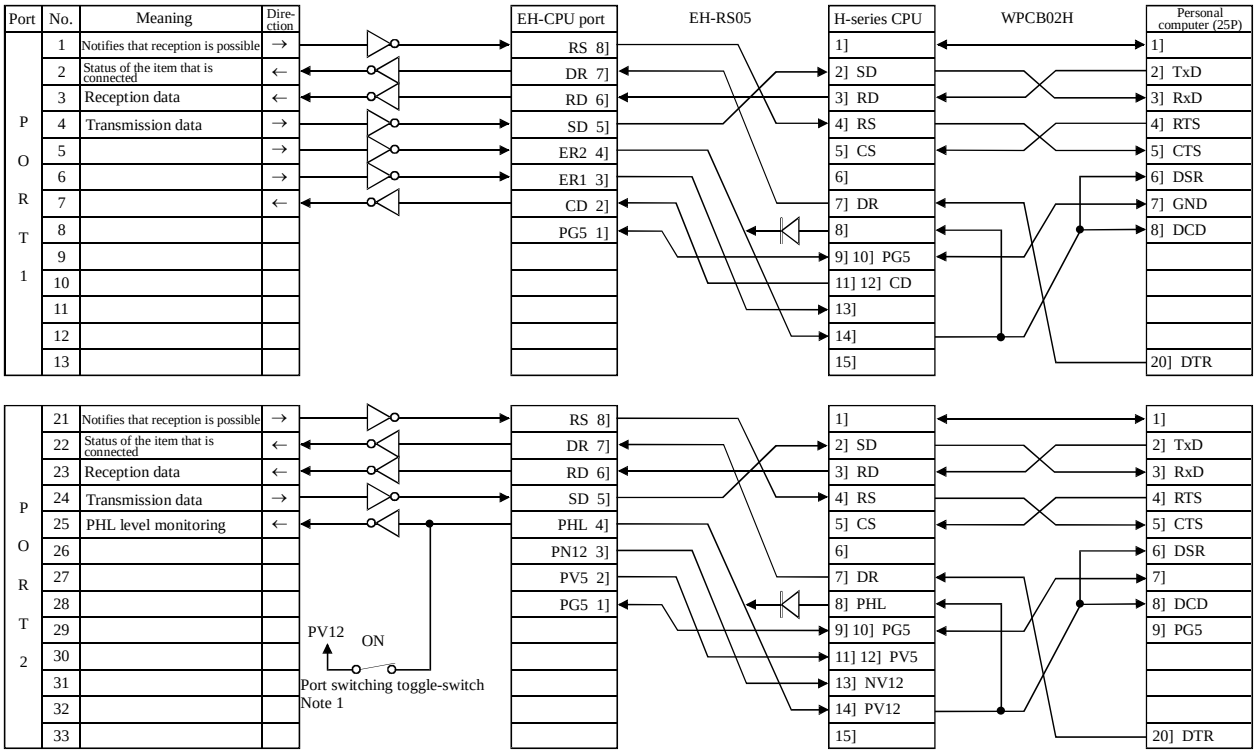
MEMO

Appendix 1 Cable Connection Diagram

EH-150 port and cable connection [when using LADDER EDITOR for PC9801 (HL-PC3)]
<Cable: EH-RS05 + PCCB02H>



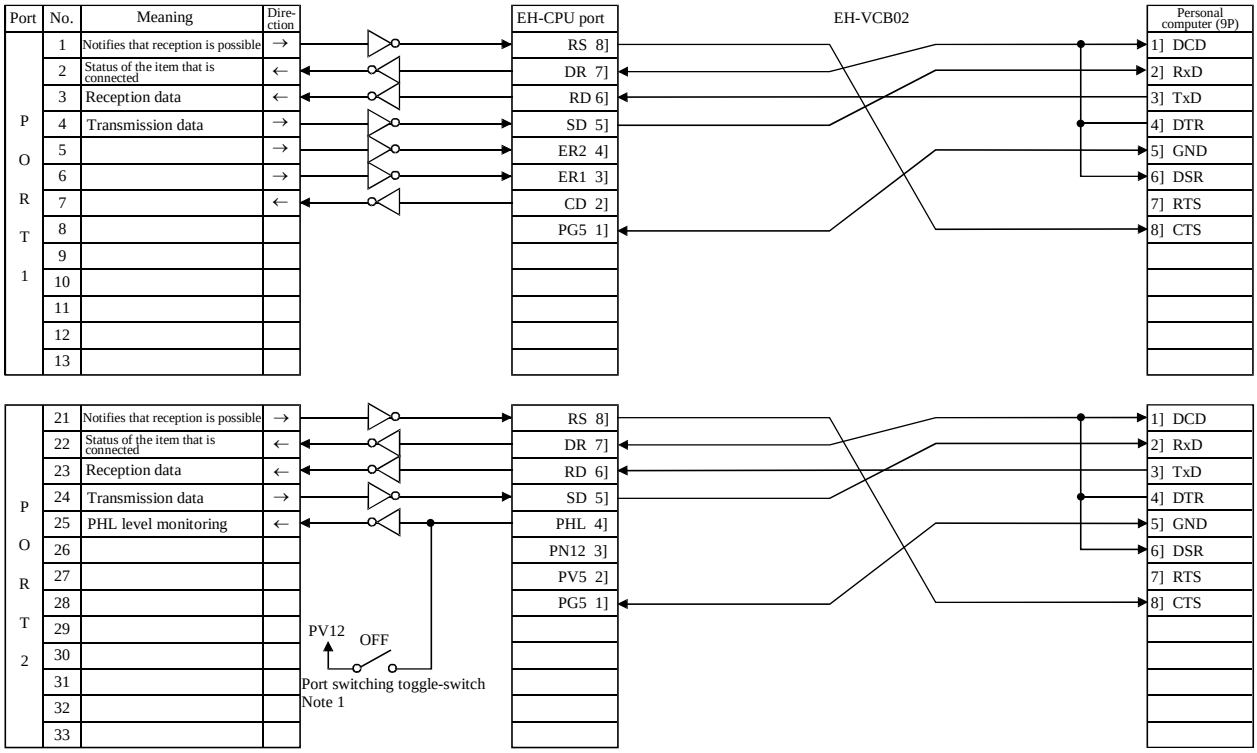
EH-150 port and cable connection [when using LADDER EDITOR for Windows® for PC9801]
<Cable: EH-RS05 + WPCB02H>



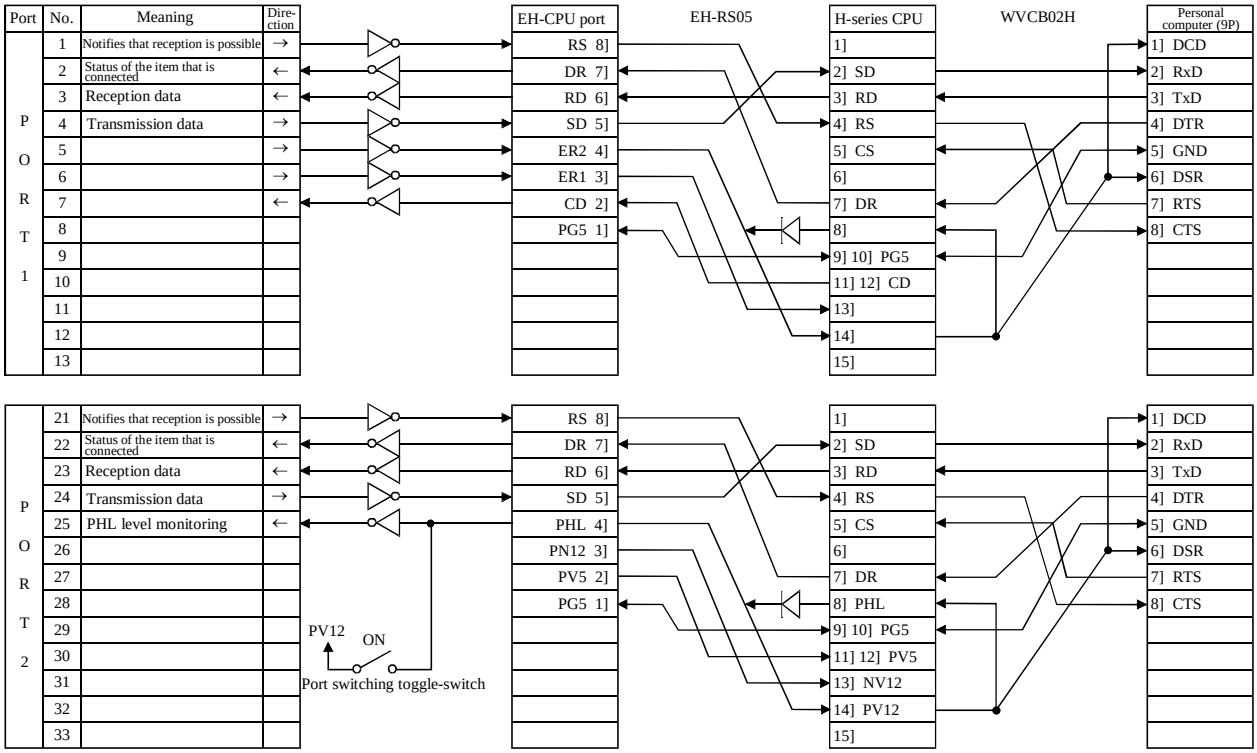
*The pin numbers ([1] to [8]) of the EH-CPU port have been changed beginning with this manual (NJI-281B(X)). For the correspondence between pin numbers and connectors, see the figure in Chapter 10, “Communication Specifications” of this manual.

Note 1: This switch is located above a dipswitch of CPU module.

EH-150 port and cable connection [when using LADDER EDITOR (HL-AT3E) for AT compatibles]
<Cable: EH-VCB02>



EH-150 port and cable connection [when using LADDER EDITOR Windows® for AT compatibles]
<Cable: EH-RS05 + WVCB02H (for Windows®)>

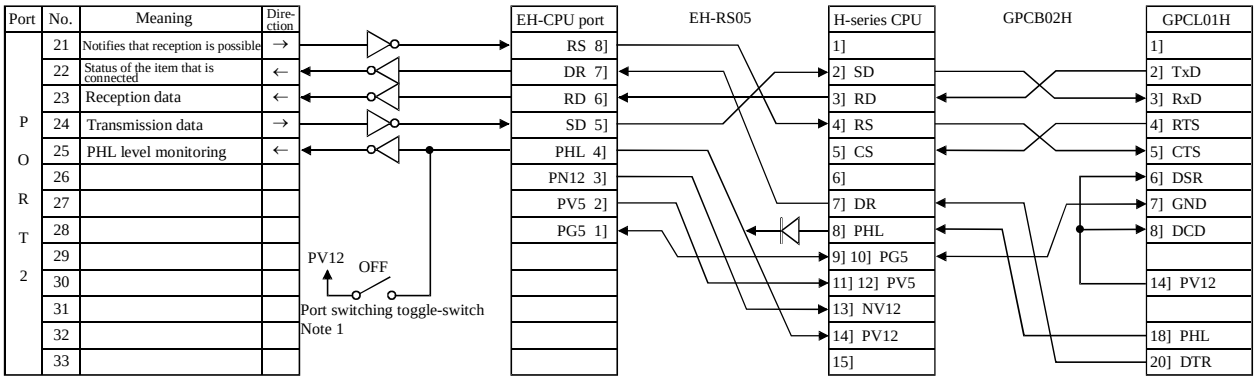


*The pin numbers ([1] to [8]) of the EH-CPU port have been changed beginning with this manual (NJI-281B(X)). For the correspondence between pin numbers and connectors, see the figure in Chapter 10, “Communication Specifications” of this manual.

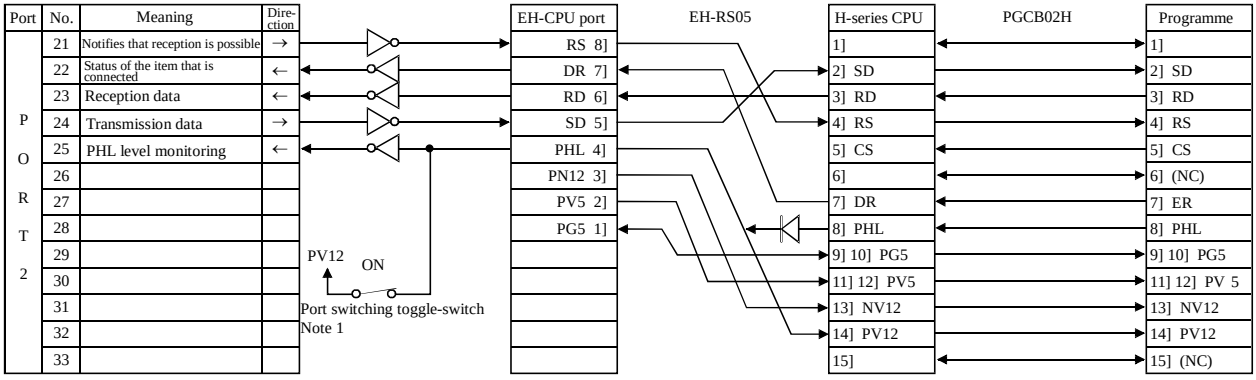
Note 1: This switch is located above a dipswitch of CPU module.

Note 2: Set on or off according to communication speed.

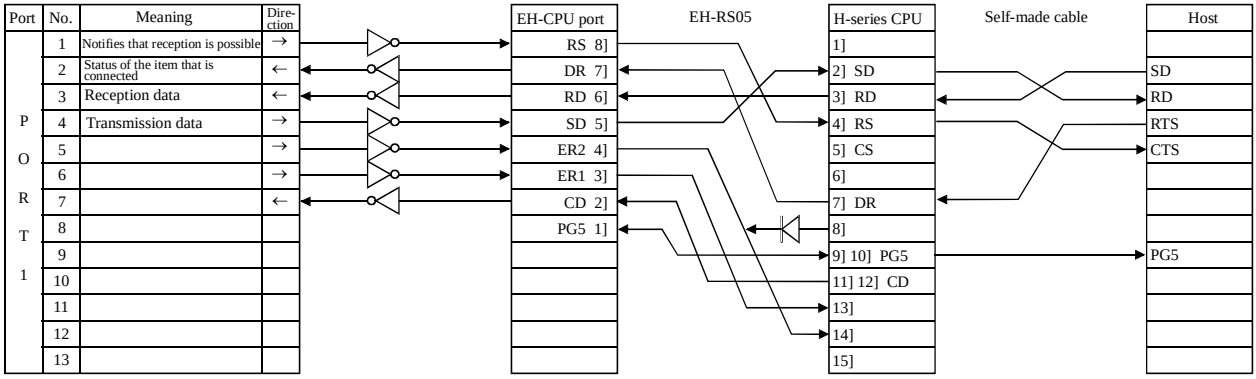
EH-150 port and cable connection [when using GPCL01H] <Cable: EH-RS05 + GPCB02H>



EH-150 port and cable connection [when using a programmer] <Cable: EH-RS05 + PGCB02H>



EH-150 port and cable connection [when using the host] <Cable: EH-RS05 + Self-made cable>



*The pin numbers ([1] to [8]) of the EH-CPU port have been changed beginning with this manual (NJI-281B(X)). For the correspondence between pin numbers and connectors, see the figure in Chapter 10, “Communication Specifications” of this manual.

Note 1: This switch is located above a dipswitch of CPU module.

MEMO

Appendix 2 H-series Command Support Comparison Chart

[Basic commands and sequence commands]

No.	Command format	Command name	EH-150	H-64 to H-20	H-200	H-250	H-252	H-2000 H-700 H-300	H-2002 H-1002 H-702 H-302	H-4010
1	LD	Logical operation start	○	○	○	○	○	○	○	○
2	LDI	Logical negation operation start	○	○	○	○	○	○	○	○
3	AND	Contact series connection	○	○	○	○	○	○	○	○
4	ANI	Contact series connection	○	○	○	○	○	○	○	○
5	OR	Contact parallel connection	○	○	○	○	○	○	○	○
6	ORI	Contact parallel connection	○	○	○	○	○	○	○	○
7	NOT	Negation	○	○	○	○	○	○	○	○
8	AND DIF	Rising edge detection	○	○	○	○	○	○	○	○
9	OR DIF	Rising edge detection	○	○	○	○	○	○	○	○
10	AND DFN	Falling edge detection	○	○	○	○	○	○	○	○
11	OR DFN	Falling edge detection	○	○	○	○	○	○	○	○
12	OUT	I/O output	○	○	○	○	○	○	○	○
13	SET	Set I/O output	○	○	○	○	○	○	○	○
14	RES	Reset I/O output	○	○	○	○	○	○	○	○
15	MCS	Set master control	○	○	○	○	○	○	○	○
16	MCR	Reset master control	○	○	○	○	○	○	○	○
17	MPS	Save operation result	○	○	○	○	○	○	○	○
18	MRD	Read operation result	○	○	○	○	○	○	○	○
19	MPP	Clear operation result	○	○	○	○	○	○	○	○
20	ANB	Logical block series connection	○	○	○	○	○	○	○	○
21	ORB	Logical block parallel connection	○	○	○	○	○	○	○	○
22	[ ]	Processing box start and end	○	○	○	○	○	○	○	○
23	( )	Relational box start and end	○	○	○	○	○	○	○	○

* ○: Supported ×: Not supported

[Basic commands and timer/counter]

No.	Command format	Command name	EH-150	H-64 to H-20	H-200	H-250	H-252	H-2000 H-700 H-300	H-2002 H-1002 H-702 H-302	H4010
1	OUT TD	On delay timer	○	○	○	○	○	○	○	○
2	OUT TM	On delay timer	○5	×	×	×	×	×	×	×
3	OUT SS	Single shot	○	○	○	○	○	○	○	○
4	OUT MS	Mono stable timer	○	×	×	○	○	○	○	○
5	OUT TMR	Integral timer	○	×	×	○	○	○	○	○
6	OUT WDT	Watchdog timer	○	×	×	○	○	○	○	○
7	OUT CU	Counter	○	○	○	○	○	○	○	○
8	OUT RCU	Ring counter	○	×	×	○	○	○	○	○
9	OUT CTU	Up and down of up counter	○	○	○	○	○	○	○	○
10	OUT CTD	Up and down of down counter	○	○	○	○	○	○	○	○
11	OUT CL	Counter clear	○	○	○	○	○	○	○	○

* ○: Supported ○5: Supported by CPU516/548 ×: Not supported

[Basic commands and relational box]

No.	Command format	Command name	EH-150	H-64 to H-20	H-200	H-250	H-252	H-2000 H-700 H-300	H-2002 H-1002 H-702 H-302	H4010
1	LD(s1 == s2)	= Relational box	○	○	○	○	○	○	○	○
2	AND(s1 == s2)	= Relational box	○	○	○	○	○	○	○	○
3	OR(s1 == s2)	= Relational box	○	○	○	○	○	○	○	○
4	LD(s1 S== s2)	Signed = Relational box	○	×	×	○	○	○	○	○
5	AND(s1 S== s2)	Signed = Relational box	○	×	×	○	○	○	○	○
6	OR(s1 S== s2)	Signed = Relational box	○	×	×	○	○	○	○	○
7	LD(s1 < > s2)	< > Relational box	○	○	○	○	○	○	○	○
8	AND(s1 < > s2)	< > Relational box	○	○	○	○	○	○	○	○
9	OR(s1 < > s2)	< > Relational box	○	○	○	○	○	○	○	○
10	LD(s1 S< > s2)	Signed < > Relational box	○	×	×	○	○	○	○	○
11	AND(s1 S< > s2)	Signed < > Relational box	○	×	×	○	○	○	○	○
12	OR(s1 S< > s2)	Signed < > Relational box	○	×	×	○	○	○	○	○
13	LD(s1 < s2)	< Relational box	○	○	○	○	○	○	○	○
14	AND(s1 < s2)	< Relational box	○	○	○	○	○	○	○	○
15	OR(s1 < s2)	< Relational box	○	○	○	○	○	○	○	○
16	LD(s1 S< s2)	Signed < Relational box	○	×	×	○	○	○	○	○
17	AND(s1 S< s2)	Signed < Relational box	○	×	×	○	○	○	○	○
18	OR(s1 S< s2)	Signed < Relational box	○	×	×	○	○	○	○	○
19	LD(s1 <= s2)	<= Relational box	○	○	○	○	○	○	○	○
20	AND(s1 <= s2)	<= Relational box	○	○	○	○	○	○	○	○
21	OR(s1 <= s2)	<= Relational box	○	○	○	○	○	○	○	○
22	LD(s1 S<= s2)	Signed <= Relational box	○	×	×	○	○	○	○	○
23	AND(s1 S<= s2)	Signed <= Relational box	○	×	×	○	○	○	○	○
24	OR(s1 S<= s2)	Signed <= Relational box	○	×	×	○	○	○	○	○

* ○: Supported ×: Not supported

[Arithmetic commands]

No.	Command format	Command name	EH-150	H-64 to H-20	H-200	H-250	H-252	H-2000 H-700 H-300	H-2002 H-1002 H-702 H-302	H4010
1	d = s	Substitution statement	○	○	○	○	○	○	○	○
2	d = s1 + s2	Binary addition	○	○	○	○	○	○	○	○
3	d = s1 B+ s2	BCD addition	○	○	○	○	○	○	○	○
4	d = s – s2	Binary subtraction	○	○	○	○	○	○	○	○
5	d = s1 B– s2	BCD subtraction	○	○	○	○	○	○	○	○
6	d = s1 × s2	Binary multiplication	○	○	○	○	○	○	○	○
7	d = s1 B× s2	BCD multiplication	○	○	○	○	○	○	○	○
8	d = s1 S× s2	Signed binary multiplication	○	×	×	○	○	○	○	○
9	d = s1 / s2	Binary division	○	○	○	○	○	○	○	○
10	d = s1 B/ s2	BCD division	○	○	○	○	○	○	○	○
11	d = s1 S/ s2	Signed binary division	○	×	×	○	○	○	○	○
12	d = s1 OR s2	Logical OR	○	○	○	○	○	○	○	○
13	d = s1 AND s2	Logical AND	○	○	○	○	○	○	○	○
14	d = s1 XOR s2	Exclusive OR	○	○	○	○	○	○	○	○
15	d = s1 == s2	= Relational expression	○	○	○	○	○	○	○	○
16	d = s1 S== s2	Signed = Relational expression	○	×	×	○	○	○	○	○
17	d = s1 < > s2	≠ Relational expression	○	○	○	○	○	○	○	○
18	d = s1 S< > s2	Signed ≠ Relational expression	○	×	×	○	○	○	○	○
19	d = s1 < s2	< Relational expression	○	○	○	○	○	○	○	○
20	d = s1 S< s2	Signed < Relational expression	○	×	×	○	○	○	○	○
21	d = s1 <= s2	<= Relational expression	○	○	○	○	○	○	○	○
22	d = s1 S<= s2	Signed <= Relational expression	○	×	×	○	○	○	○	○

* ○: Supported ×: Not supported

[Application commands] (1/2)

No.	Command format	Command name	EH-150	H-64 to H-20	H-200	H-250	H-252	H-2000 H-700 H-300	H-2002 H-1002 H-702 H-302	H4010
1	BSET (d, n)	Bit set	○	○	○	○	○	○	○	○
2	BRES (d, n)	Bit reset	○	○	○	○	○	○	○	○
3	BTS (d, n)	Bit test	○	○	○	○	○	○	○	○
4	SHR (d, n)	Shift right	○	○	○	○	○	○	○	○
5	SHL (d, n)	Shift left	○	○	○	○	○	○	○	○
6	ROR (d, n)	Rotate right	○	○	○	○	○	○	○	○
7	ROL (d, n)	Rotate left	○	○	○	○	○	○	○	○
8	LSR (d, n)	Logical shift right	○	○	○	○	○	○	○	○
9	LSL (d, n)	Logical shift left	○	○	○	○	○	○	○	○
10	BSR (d, n)	BCD shift right	○	○	○	○	○	○	○	○
11	BSL (d, n)	BCD shift left	○	○	○	○	○	○	○	○
12	WSHR (d, n)	Batch shift right	○	×	×	○	○	○	○	○
13	WSHL (d, n)	Batch shift left	○	×	×	○	○	○	○	○
14	WBSR (d, n)	Batch BCD shift right	○	×	×	○	○	○	○	○
15	WBSL (d, n)	Batch BCD shift left	○	×	×	○	○	○	○	○
16	MOV (d, s, n)	Block transfer	○	×	×	○	○	○	○	○
17	COPY (d, s, n)	Copy	○	×	×	○	○	○	○	○

* ○: Supported ×: Not supported

[Application commands] (2/2)

No.	Command format	Command name	EH-150	H-64 to H-20	H-200	H-250	H-252	H-2000 H-700 H-300	H-2002 H-1002 H-702 H-302	H4010
18	XCG (d, d2, n)	Block exchange	○	×	×	○	○	○	○	○
19	NOT (d)	Reverse	○	○	○	○	○	○	○	○
20	NEG (d)	Two's complement	○	○	○	○	○	○	○	○
21	ABS (d, s)	Absolute value	○	○	○	○	○	○	○	○
22	SGET (d, s)	Sign addition	○	×	×	○	○	○	○	○
23	EXT (d, s)	Sign expansion	○	×	×	○	○	○	○	○
24	BCD (d, s)	Binary → BCD conversion	○	○	○	○	○	○	○	○
25	BIN (d, s)	BCD → Binary conversion	○	○	○	○	○	○	○	○
26	DECO (d, s, n)	Decode	○	○	○	○	○	○	○	○
27	ENCO (d, s, n)	Encode	○	○	○	○	○	○	○	○
28	SEG (d, s)	7 segment decode	○	×	×	○	○	○	○	○
29	SQR (d, s)	Square root	○	×	×	○	○	○	○	○
30	BCU (d, s)	Bit count	○	○	○	○	○	○	○	○
31	SWAP (d)	Swap	○	○	○	○	○	○	○	○
32	FIFIT (P, n)	FIFO initialization	○	×	×	○	○	○	○	○
33	FIFWR (P, s)	FIFO write	○	×	×	○	○	○	○	○
34	FIFRD (P, d)	FIFO read	○	×	×	○	○	○	○	○
35	UNIT (d, s, n)	Unit	○	○	○	○	○	○	○	○
36	DIST (d, s, n)	Distribute	○	○	○	○	○	○	○	○
37	ADRIO (d, s)	I/O address conversion	○	×	×	×	○	○	○	○

* ○: Supported ×: Not supported

[Control commands]

No.	Command format	Command name	EH-150	H-64 to H-20	H-200	H-250	H-252	H-2000 H-700 H-300	H-2002 H-1002 H-702 H-302	H4010
1	END	Normal scan end	○	○	○	○	○	○	○	○
2	CEND (s)	Scan conditional end	○	○	○	○	○	○	○	○
3	JMP n	Unconditional jump	○	○	○	○	○	○	○	○
4	CJMP n (s)	Conditional jump	○	○	○	○	○	○	○	○
5	RSRV n	Reserve	×	×	×	×	×	○	○	○
6	FREE	Reserve cancel	×	×	×	×	×	○	○	○
7	LBL n	Label	○	○	○	○	○	○	○	○
8	FOR n (s)	FOR	○	×	×	○	○	○	○	○
9	NEXT n	NEXT	○	×	×	○	○	○	○	○
10	CAL n	Call subroutine	○	○	○	○	○	○	○	○
11	SB n	Start subroutine program	○	○	○	○	○	○	○	○
12	RTS	RETURN SUBROUTINE	○	○	○	○	○	○	○	○
13	START n	Start BASIC task	×	×	×	×	×	○	○	○
14	INT n	Start interrupt scan program	○	○	○	○	○	○	○	○
15	RTI	RETURN INTERRUPT	○	○	○	○	○	○	○	○

* ○: Supported ×: Not supported

[High-function module transfer commands]

No.	Command format	Command name	EH-150	H-64 to H-20	H-200	H-250	H-252	H-2000 H-700 H-300	H-2002 H-1002 H-702 H-302	H4010
1	TRNS 0 (d, s, t)	General-purpose port transmission command	○	×	×	×	×	×	○	○
2	RECV 0 (d, s, t)	General-purpose port receiving command	○	×	×	×	×	×	○	○
3	TRNS 1 (d, s, t)	Data communication command for SIO, CLOCK	×	×	×	×	○	×	○	○
4	QTRNS1 (d, s, t)	High data communication command for SIO, CLOCK	×	×	×	×	×	×	○	○
5	TRNS 2 (d, s, t)	Data communication command for ASCII	×	×	×	×	×	×	○	○
6	QTRNS2 (d, s, t)	High data communication command for ASCII	×	×	×	×	×	×	○	○
7	TRNS 3 (d, s, t)	Data transmission command for POSIT-H	×	×	×	×	×	×	○	○
8	QTRNS3 (d, s, t)	High data transmission command for POSIT-H	×	×	×	×	×	×	○	○
9	RECV 3 (d, s, t)	Data receiving command for POSIT-H	×	×	×	×	×	×	○	○
10	TRNS 4 (d, s, t)	Data communication command for POSIT-2H, POSITA2H	×	×	×	×	○	×	○	○
11	QTRNS 4 (d, s, t)	High data communication command for POSIT-2H, POSITA2H	×	×	×	×	×	×	○	○
12	TRNS 5 (d, s, t)	Data communication command for XCU-001H	×	×	×	×	×	×	○	○
13	TRNS 6 (d, s, t)	Data communication command for XCU-232H	×	×	×	×	×	×	○	○
14	TRNS 8 (d, s, t)	Telecommunication command	○3	×	×	×	×	×	×	×

○ : Supported ○1 : Supported by CPU***A/448/516/548
○2 : Supported by CPU308(A)/316(A)/448(A)/516/548
○3 : Supported by CPU208A/308A/316A/448(A)/516/548
○4 : Supported by CPU308/316/***A/448/516/548
○5 : Supported by CPU516/548
× : Not supported

[FUN commands] (1/5)

No.	Command format	Command name	EH-150	H-64 to H-20	H-200	H-250	H-252	H-2000 H-700 H-300	H-2002 H-1002 H-702 H-302	H4010
1	FUN 0 (s) (PIDIT (s))	PID operation initialization	○2	×	×	×	○	×	○	○
2	FUN 1 (s) (PIDOP (s))	PID operation execution control	○2	×	×	×	○	×	○	○
3	FUN 2 (s) (PIDCL (s))	PID operation execution	○2	×	×	×	○	×	○	○
4	FUN 4 (s) (IFR (s))	Process stepping	○	×	×	×	×	×	×	○
5	FUN 10 (s) (SIN (s))	SIN function	○	×	×	×	○	×	○	○
6	FUN 11 (s) (COS (s))	COS function	○	×	×	×	○	×	○	○
7	FUN 12 (s) (TAN (s))	TAN function	○	×	×	×	○	×	○	○
8	FUN 13 (s) (ASIN (s))	ARC SIN function	○	×	×	×	○	×	○	○
9	FUN 14 (s) (ACOS (s))	ARC COS function	○	×	×	×	○	×	○	○
10	FUN 15 (s) (ATAN (s))	ARC TAN function	○	×	×	×	○	×	○	○
11	FUN 20 (s) (DSRCH (s))	Data search	○5	×	×	×	○	×	○	○
12	FUN 21 (s) (TSRCH (s))	Table search	○5	×	×	×	○	×	○	○
13	FUN 22 (s)	Check code calculation	○5	×	×	×	×	×	×	×
14	FUN 23 (s)	Check code verifying	○5	×	×	×	×	×	×	×
15	FUN 30 (s) (BINDA (s))	Binary → decimal ASCII conversion (16 bit)	○1	×	×	×	○	×	○	○
16	FUN 31 (s) (DBINDA (s))	Binary → decimal ASCII conversion (32 bit)	○1	×	×	×	○	×	○	○
17	FUN 32 (s) (BINHA (s))	Binary → hexadecimal ASCII conversion (16 bit)	○1	×	×	×	○	×	○	○
18	FUN 33 (s) (DBINHA (s))	Binary → hexadecimal ASCII conversion (32 bit)	○1	×	×	×	○	×	○	○
19	FUN 34 (s) (BCDDA (s))	BCD → decimal ASCII conversion (16 bit)	○1	×	×	×	○	×	○	○
20	FUN 35 (s) (DBCDDA (s))	BCD → decimal ASCII conversion (32 bit)	○1	×	×	×	○	×	○	○
21	FUN 36 (s) (DABIN (s))	Without signed 5 digit Decimal ASCII conversion → binary conversion	○1	×	×	×	○	×	○	○
22	FUN 37 (s) (DDABIN (s))	Signed 10 digit Decimal ASCII conversion → binary conversion	○1	×	×	×	○	×	○	○
23	FUN 38 (s) (HABIN (s))	4 digit hexadecimal ASCII → binary conversion	○1	×	×	×	○	×	○	○
24	FUN 39 (s) (DHABIN (s))	8 digit hexadecimal ASCII → binary conversion	○1	×	×	×	○	×	○	○
25	FUN 40 (s) (DABCD (s))	4 digit decimal ASCII → BCD conversion	○1	×	×	×	○	×	○	○

○ : Supported ○1 : Supported by CPU***A/448/516/548 ○2 : Supported by CPU308(A)/316(A)/448(A)/516/548
○3 : Supported by CPU208A/308A/316A/448(A)/516/548 ○4 : Supported by CPU308/316/***A/448/516/548
○5 : Supported by CPU516/548 × : Not supported

[FUN commands] (2/5)

No.	Command format	Command name	EH-150	H-64 to H-20	H-200	H-250	H-252	H-2000 H-700 H-300	H-2002 H-1002 H-702 H-302	H4010
26	FUN 41 (s) (DDABCD (s))	8 digit decimal ASCII → BCD conversion	○1	×	×	×	○	×	○	○
27	FUN 42 (s) (ASC (s))	Hexadecimal binary → ASCII conversion (digit designation)	○1	×	×	×	○	×	○	○
28	FUN 43 (s) (HEX (s))	Hexadecimal ASCII → binary conversion (digit designation)	○1	×	×	×	○	×	○	○
29	FUN 44 (s) (ASDD (s))	Character unit	○1	×	×	×	○	×	○	○
30	FUN 45 (s) (SCMP (s))	Character relational	○1	×	×	×	○	×	○	○
31	FUN 46 (s) (WTOB (s))	Word → byte conversion	○1	×	×	×	○	×	○	○
32	FUN 47 (s) (WTOW (s))	Byte → word conversion	○1	×	×	×	○	×	○	○
33	FUN 48 (s) (BSHR (s))	Byte unit right shift	○1	×	×	×	○	×	○	○
34	FUN 49 (s) (BSHL (s))	Byte unit left shift	○1	×	×	×	○	×	○	○
35	FUN 50 (s) (TRSET (s))	Sampling trace set	×	×	×	×	○	×	○	○
36	FUN 51 (s) (TRACE (s))	Sampling trace execution	×	×	×	×	○	×	○	○
37	FUN 52 (s) (TRRES (s))	Sampling trace reset	×	×	×	×	○	×	○	○
38	FUN 60 (s) (BSQR (s))	Binary square	○1	×	×	×	○	×	○	○
39	FUN 61 (s) (PGEN (s))	Dynamic scan pulse	○1	×	×	×	○	×	○	○
40	FUN 70 (s)	High-speed counter mode set	×	○	×	×	×	×	×	×
41	FUN 71 (s)	High-speed counter process value read	×	○	×	×	×	×	×	×
42	FUN 72 (s)	High-speed counter process value write	×	○	×	×	×	×	×	×
43	FUN 73 (s)	High-speed counter set value read	×	○	×	×	×	×	×	×
44	FUN 74 (s)	High-speed counter set value write	×	○	×	×	×	×	×	×
45	FUN 80 (s) (ALREF (s))	I/O refresh (All points)	○	×	×	×	○	×	×	○
46	FUN 81 (s) (IOREF (s))	I/O refresh (Input/output designation)	○	×	×	×	○	×	×	○
47	FUN 82 (s) (SLREF (s))	I/O refresh (Any slot)	○	×	×	×	○	×	×	○
48	FUN 90 (s) (ETDIT (s))	Expansion timer initial setting	×	×	×	×	×	×	×	○
49	FUN 91 (s) (ETD (s))	Expansion timer execution	×	×	×	×	×	×	×	○
50	FUN 97 (s) (WNRED (s))	Expansion link area read	×	×	×	×	×	×	×	○
51	FUN 98 (s) (WNWRT (s))	Expansion link area write	×	×	×	×	×	×	×	○

○ : Supported ○1 : Supported by CPU***A/448/516/548
○2 : Supported by CPU308(A)/316(A)/448(A)/516/548
○3 : Supported by CPU208A/308A/316A/448(A)/516/548
○4 : Supported by CPU308/316/***A/448/516/548
○5 : Supported by CPU516/548
× : Not supported

[FUN commands] (3/5)

No.	Command format	Command name	EH-150	H-64 to H-20	H-200	H-250	H-252	H-2000 H-700 H-300	H-2002 H-1002 H-702 H-302	H4010
52	FUN 100 (s) (INT (s))	Flow decimal point operation (real number → integral number (word) conversion)	○2	×	×	×	×	×	×	○
53	FUN 101 (s) (INTD (s))	Flow decimal point operation (real number → integral number (double word) conversion)	○2	×	×	×	×	×	×	○
54	FUN 102 (s) (FLOAT (s))	Flow decimal point operation (integral number (word) → real number conversion)	○2	×	×	×	×	×	×	○
55	FUN 103 (s) (FLOATD (s))	Flow decimal point operation (integral number (double word) → real number conversion)	○2	×	×	×	×	×	×	○
56	FUN 104 (s) (FADD (s))	Flow decimal point operation (addition)	○2	×	×	×	×	×	×	○
57	FUN 105 (s) (FSUB (s))	Flow decimal point operation (subtraction)	○2	×	×	×	×	×	×	○
58	FUN 106 (s) (FMUL (s))	Flow decimal point operation (multiplication)	○2	×	×	×	×	×	×	○
59	FUN 107 (s) (FDIV (s))	Flow decimal point operation (division)	○2	×	×	×	×	×	×	○
60	FUN 108 (s) (FRAD (s))	Flow decimal point operation (angle → radian conversion)	○2	×	×	×	×	×	×	○
61	FUN 109 (s) (FDEG (s))	Flow decimal point operation (radian → angle conversion)	○2	×	×	×	×	×	×	○
62	FUN 110 (s) (FSIN (s))	Flow decimal point operation (SIN)	○2	×	×	×	×	×	×	○
63	FUN 111 (s) (FCOS (s))	Flow decimal point operation (COS)	○2	×	×	×	×	×	×	○
64	FUN 112 (s) (FTAN (s))	Flow decimal point operation (TAN)	○2	×	×	×	×	×	×	○
65	FUN 113 (s) (FASIN (s))	Flow decimal point operation (ARC SIN)	○2	×	×	×	×	×	×	○
66	FUN 114 (s) (FACOS (s))	Flow decimal point operation (ARC COS)	○2	×	×	×	×	×	×	○
67	FUN 115 (s) (FATAN (s))	Flow decimal point operation (ARC TAN)	○2	×	×	×	×	×	×	○
68	FUN 116 (s) (FSQR (s))	Flow decimal point operation (square)	○2	×	×	×	×	×	×	○
69	FUN 117 (s) (FEXP (s))	Flow decimal point operation (characteristic)	○2	×	×	×	×	×	×	○
70	FUN 118 (s) (FLOG (s))	Flow decimal point operation (napierian logarithm)	○2	×	×	×	×	×	×	○
71	FUN 120 (s) (INDXD (s))	Index setting (argument d)	○1	×	×	×	×	×	×	○
72	FUN 121 (s) (INDXS (s))	Index setting (argument s)	○1	×	×	×	×	×	×	○
73	FUN 122 (s) (INDXC (s))	Index cancel	○1	×	×	×	×	×	×	○
74	FUN 123 (s) (INC (s))	Increment (INC)	○4	×	×	×	×	×	×	○
75	FUN 124 (s) (INCD (s))	Double word increment (DINC)	○4	×	×	×	×	×	×	○

○ : Supported ○1 : Supported by CPU***A/448/516/548
○2 : Supported by CPU308(A)/316(A)/448(A)/516/548
○3 : Supported by CPU208A/308A/316A/448(A)/516/548
○4 : Supported by CPU308/316/***A/448/516/548
○5 : Supported by CPU516/548
× : Not supported

[FUN commands] (4/5)

No.	Command format	Command name	EH-150	H-64 to H-20	H-200	H-250	H-252	H-2000 H-700 H-300	H-2002 H-1002 H-702 H-302	H4010
76	FUN 125 (s) (DEC (s))	Decrement (DEC)	○4	×	×	×	×	×	×	○
77	FUN 126 (s) (DECD (s))	Double word decrement (DECD)	○4	×	×	×	×	×	×	○
78	FUN 127 (s) (BITTOW (s))	Expansion of bit data to word data	○1	×	×	×	×	×	×	○
79	FUN 128 (s) (WTOBIT (s))	Expansion of word data to bit data	○1	×	×	×	×	×	×	○
80	FUN 133 (s) (FIBMOV (s))	File memory block transfer	×	×	×	×	×	×	×	○
81	FUN 134 (s) (FIBCHG (s))	File memory block exchange	×	×	×	×	×	×	×	○
82	FUN 135 (s) (FIWRED (s))	File memory word unit read	×	×	×	×	×	×	×	○
83	FUN 136 (s) (FIWWRT (s))	File memory word unit write	×	×	×	×	×	×	×	○
84	FUN 162 (s)	Explicit message execution (DeviceNet)	○5	×	×	×	×	×	×	×
85	FUN 163 (s)	Explicit message setting (DeviceNet)	○5	×	×	×	×	×	×	×
86	FUN 190 (s)	Inverter control command	○5	×	×	×	×	×	×	×
87	FUN200 (s) (XYR/W (s))	X, Y area read/write command	○1	×	×	×	×	×	×	×
88	FUN201 (s) (SCR/W (s))	Status control area read/write command	○1	×	×	×	×	×	×	×
89	FUN 210 (s) (LOGIT (s))	Initial setting for data logging	○2	×	×	×	×	×	×	×
90	FUN 211 (s) (LOGWRT (s))	Log data write	○2	×	×	×	×	×	×	×
91	FUN 212 (s) (LOGCLR (s))	Log data clear	○2	×	×	×	×	×	×	×
92	FUN 213 (s) (LOGRED (s))	Log data read	○2	×	×	×	×	×	×	×
93	FUN 254 (s) (BOXC (s))	BOX comment	○	○	○	○	○	○	○	○
94	FUN 255 (s) (MEMC (s))	Memo comment	○	○	○	○	○	○	○	○

○ : Supported ○1 : Supported by CPU***A/448/516/548
○2 : Supported by CPU308(A)/316(A)/448(A)/516/548
○3 : Supported by CPU208A/308A/316A/448(A)/516/548
○4 : Supported by CPU308/316/***A/448/516/548
○5 : Supported by CPU516/548
× : Not supported

MEMO

Appendix 3 Index of Instruction

[H]

[I]

[J]

[L]

[M]

[N]

[O]

[P]

[R]

[S]

[T]

[U]

[W][X]

MEMO