HITACHI PROGRAMMABLE CONTROLLER

MICRO-EH

BASIC UNIT (64 points type)

APPLICATION MANUAL

NJI-465 (X)

WARNING

To ensure that the equipment described by this manual. As well as all equipment connected to and used with it, operate satisfactorily and safely, all applicable local and national codes that apply to installing and operating the equipment must be followed. Since codes can vary geographically and can change with time, it is the user's responsibility to determine which standard and codes apply, and to comply with them.

FAILURE TO COMPLY WITH APPLICABLE CODES AND STANDARDS CAN RESULT IN DAMAGE TO EQUIPMENT AND / OR SERIOUS INJURY TO PERSONNEL.
INSTALL EMERGENCY POWER STOP SWITCH WHICH OPERATES INDEPENDENTLY OF THE PROGRAMMABLE CONTROLLER TO PROTECT THE EQUIPMENT AND / OR PERSONNEL IN CASE OF THE CONTROLLER MALFUNCTION.

Personnel who are to install and operate the equipment should carefully study this manual and any others referred to by it prior to installation and / or operation of the equipment. Hitachi, Ltd. constantly strives to improve its products, and the equipment and the manual(s) that describe it may be different from those already in your possession.

If you have any questions regarding the installation and operation of the equipment, or if more information is desired, contact your local Authorized Distributor or Hitachi, Ltd.

IMPORTANT

THIS EQUIPMENT GENERATES, USES, AND CAN RADIATE RADIO FREQUENCY ENERGY AND, IF NOT INSTALLED AND USED IN ACCORDANCE WITH THE INSTRUCTION MANUAL, MAY CAUSE INTERFERENCE TO RADIO COMMUNICATIONS. AS TEMPORARILY PERMITTED BY REGULATION, IT HAS NOT BEEN TESTED FOR COMPLIANCE WITH THE LIMITS FOR CLASS A COMPUTING DEVICES PURSUANT TO SUBPART J OF PART 15 OF FCC RULES, WHICH ARE DESIGNED TO PROVIDE REASONABLE PROTECTION AGAINST SUCH INTERFERENCE.

OPERATION OF THIS EQUIPMENT IN A RESIDENTIAL AREA IS LIKELY TO CAUSE INTERFERENCE IN WHICH CASE THE USER, AT HIS OWN EXPENSE, WILL BE REQUIRED TO TAKE WHATEVER MEASURES MAY BE REQUIRED TO CORRECT THE INTERFERENCE.

LIMITED WARRANTY AND IMITATION OF LIABILITY

Hitachi, Ltd. (Hitachi) warrants to the original purchaser that the programmable controller (PLC) manufactured by Hitachi is free from defects in material and workmanship under normal use and service. The obligation of Hitachi under this warranty shall be limited to the repair or exchange of any part or parts which may prove defective under normal use and service within eighteen (18) months from the date of manufacture or twelve (12) months from the date of installation by the original purchaser which ever occurs first, such defect to be disclosed to the satisfaction of Hitachi after examination by Hitachi of the allegedly defective part or parts. This warranty in expressly in lieu of all other warranties expressed or implied including the warranties of merchantability and fitness for use and of all other obligations or liabilities and Hitachi neither assumes, nor authorizes any other person to assume for Hitachi, any other liability in connection with the sale of this PLC. This warranty shall not apply to this PLC or any part hereof which has been subject to accident, negligence, alteration, abuse, or misuse. Hitachi makes no warranty whatsoever in respect to accessories or parts not supplied by Hitachi. The term "original purchaser", as used in this warranty, shall be deemed to mean that person for whom the PLC in originally installed.

In no event, whether as a result of breach of contract, warranty, tort (including negligence) or otherwise, shall Hitachi or its suppliers be liable for any special, consequential, incidental or penal damages Including, but not limited to, loss of profit or revenues, loss of use of the products or any associated equipment, damage to associated equipment, cost of capital, cost of substitute products, facilities, services or replacement power, down time costs, or claims of original purchaser’s customers for such damages.

To obtain warranty service, return the product to your distributor, or send it with a description of the problem, proof of purchase, post paid, insured, and in a suitable package to:

Quality Assurance Dep.
Hitachi Industrial Equipment Systems Co., Ltd.
46-1, Ooaza-Tomioka Nakajo-machi
Kitakanbara-gun, Niigata-ken
959-2608 JAPAN

Hitachi, Ltd. provides customer assistance in varied technical areas. Since Hitachi does not posses full access to data concerning all of the uses and applications of customer‘s products, responsibility is assumed by Hitachi neither for customer product design nor for any infringements of patents or rights of others which may result from Hitachi assistance.

The specifications and descriptions contained in this manual were accurate at the time they were approved for printing. Since Hitachi, Ltd. Incorporated constantly strives to improve all its products, we reserve the right to make changes to equipment and/or manuals at any time without notice and without incurring any obligation other than as noted in this manual.

Hitachi, Ltd. assumes no responsibility for errors that may appear in this manual.

As the product works with user program and Hitachi, Ltd. cannot test all combination of user program components, it is assumed that a bug or bugs may happen unintentionally. If it is happened: please inform the fact to Hitachi, Ltd. or its representative. Hitachi will try to find the reason as much as possible and inform the countermeasure when obtained.

Nevertheless Hitachi, Ltd. intends to make products with enough reliability, the product has possibility to be damaged at any time. Therefore personnel who are to install and operate the equipment has to prepare with the counter-measure such as power off switch can be operated independently of the controller. Otherwise, it can result in damage to equipment and/or serious injury to personnel.

Safety Precautions

Read this manual and attached documents thoroughly before installing and operating this unit, and performing maintenance or inspection of this unit in order to use the unit correctly. Be sure to use this unit after acquiring adequate knowledge of the unit, all safety information, and all precautionary information. Also, be sure to deliver this manual to the person in charge of maintenance.
Safety caution items are classified as “Danger” and “Caution” in this document.

DANGER : Cases in which, if handled incorrectly, a dangerous situation may occur, resulting in possible death or severe injury.

CAUTION : Cases in which, if handled incorrectly, a dangerous situation may occur, resulting in possible minor to medium injury to the body, or only mechanical failure.

However, depending on the situation, items marked with **CAUTION** may result in major accidents.

Both of these items contain important safety information, so be sure to follow them closely.

Icons for prohibited items and required items are shown below:

: Indicates a prohibited item (item that cannot be performed). For example, when open flames are prohibited, is shown.

: Indicates a required item (item that must be performed). For example, when grounding must be performed, is shown.

1. Installation

⚠ CAUTION
• Use this product in an environment as described in the catalogue 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 an electric shock, fire or malfunction. • Installation this product according to the instructions in this manual. If installation is not performed correctly, it may result in falling, malfunction, or an operational error of the unit. • Never allow foreign objects such as wire chips to enter the unit. They may cause a fire, malfunction, or failure.

2. Wiring

REQUIRED
• Always perform grounding (FE terminal). If grounding is not performed, there is a risk of an electric shock or malfunction.

⚠ CAUTION
• Connect a power supply that meets the rating. If a power supply that does not meet the rating is connected, it may result in a fire. • Any wiring operation should only be performed by a qualified technician. If wiring is performed incorrectly, it may result in a fire, failure, or electric shock.

3. Precautions When Using the Unit

⚠DANGER
• Never touch the terminals while the power is on. There is a risk of an electric shock. • Configure the emergency stop circuit, interlock circuit and other related circuits external to the programmable controller (referred to as the PLC in this document). Otherwise, a failure in the PLC may damage the equipment or result in a serious accident. Never interlock the unit with the external load via the relay drive power supply of the relay output module.

⚠ CAUTION
• Before performing program change, forced output, run, stop and other operations while the unit is in operation, be sure to check the validity of the applicable operation and safety. An operation error may damage the equipment or result in a serious accident. • Be sure to power on the unit according to the designated power-on sequence. Otherwise, an erroneous operation may damage the equipment or result in a serious accident.

4. Maintenance

⚠ DANGER
• Never connect the ⊕ and ⊖ of the battery in reverse. Also, never charge, disassemble, heat, place in fire, or short circuit the battery. There is a risk of an explosion or fire.

PROHIBITED
• Never disassemble or modify the unit. These actions may result in a fire, malfunction, or failure.

⚠ CAUTION
• Be sure to turn off the power supply before removing or attaching the module/unit. Otherwise, it may result in an electric shock, malfunction, or failure.

Table of Contents

Chapter 1 Introduction

Thank you for using the Hitachi MICRO-EH Programmable Controller series (hereinafter called PLC).
This manual describes how to use the MICRO-EH 64 points type basic unit (hereinafter called MICRO64). Please refer to the MICRO-EH application manual (NJI-349*) about common contents with MICRO-EH series other than description in this book.
The MICRO-EH application manual has the following contents.

Table 1.1 Contents of application manual

Chapter		Contents
Chapter 1	Features	About the features of MICRO-EH series.
Chapter 2	System overview	The example of a system overview of MICRO-EH series
Chapter 3	Function and Performance Specifications	About various specifications (general specification, functional specification etc.)
Chapter 4	Product lineup and wiring	The name and function of each part of a unit.
Chapter 5	Instruction Specifications	The function of various ladder commands, the example of programming
Chapter 6	I/O Specifications	About an external I/O number and an internal output number
Chapter 7	Programming	About programming device and the programming method
Chapter 8	High speed counter, PWM/Pulse train output and Analogue I/O	The setting method and directions of High speed counter / PWM, Pulse output.
Chapter 9	PLC Operation	About the processing method of a program. (From an operation start to under operation)
Chapter 10	PLC Installation, Mounting, Wiring	About installation of MICRO-EH, and wiring
Chapter 11	Communication Specifications	The specification of a communication port, the setting method, etc.
Chapter 12	Error Code List and Special Internal Outputs	About error code details and the special internal outputs.
Chapter 13	Troubleshooting	The management flow at the time of trouble generating
Chapter 14	Operation Examples	An easy example explains even from creation of a program to transmission and operation.
Chapter 15	Daily and Periodic Inspections	About the item checked every day or periodically

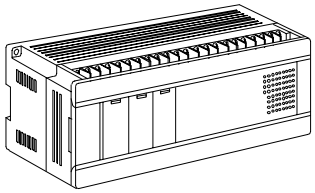
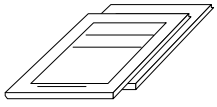
1.1 Before use

Great care has been taken in the manufacture of this product, but it is advised that the following points are checked immediately after purchase.

1. Is the model the same one that you ordered?
2. Is not the product damaged?
3. Is not any of the accessories listed in table 1.2 missing?

Contact your dealer in the event of any defects being discovered.

Table 1.2 List of accessories supplied with the MICRO64

No.	Products name	Model name	Outlook	Q'ty	Remarks
1	MICRO-64	EH-A64DR EH-D64DR EH-D64DT EH-D64DTPS		1	
2	Instruction manual	NJI-465(X)		1	

1.2 Features

MICRO64 is all-in-one compact type PLC which has the following features in addition to existing MICRO-EH series (10, 14, 23, and 28-point type).

■ Increase in I/O points

MICRO64 has 40 inputs and 24 outputs. The number of I/O points is expandable to 176 points with 4 expansion units.

■ Increase in programming memory and data memory (WR)

Program capacity is extended to 16k steps, and data memory capacity is extended to 32k words, which enables MICRO64 to support middle range applications.

■ New FUN commands

53 kinds of FUN commands and one application command are added. The added FUN commands are a data conversion command, a floating point arithmetic, etc. (they are the command currently supported by EH-150 series.)

■ 32 bits counter

The counter of MICRO64 can support up to 100kHz(single phase) or 60kHz (2-phase) pulses. The 16-bit counter is extended to the 32-bit counter.

■ Pulse train output

A pulse output with an output frequency of 65kHz is possible for MICRO64. Moreover, the number of output pulses can be set up by 32 bits. (32bit pulse is supported by software ver. 1.01 or later.)

■ PWM output

A pwm output with an output frequency of 65kHz is possible for MICRO64.

■ Compatibility with current MICRO-EH series

The command system of MICRO64 does not change with current MICRO-EH. Ladder program for the current MICRO-EH works on MICRO64 also. In addition, it is possible to connect existing expansion unit.

■ Selectable option boards

A function is expandable by attaching an option board in a basic unit. The following option boards will be released.

- RS-422/485 communication board
  … RS-422/485 Interface. It can be used as an programming port or a general-purpose port.
  10 bits analog inputs (2ch) are attached.
- RS-232C communication board
  … RS-232C Interface. It can be used as an programming port or a general-purpose port.
  10 bits analog inputs (2ch) are attached.
- Memory board

  … It can be used for backup of a user program etc.

Caution
Since above option boards have not been released yet, the first version of MICRO64 may not support all the option boards.

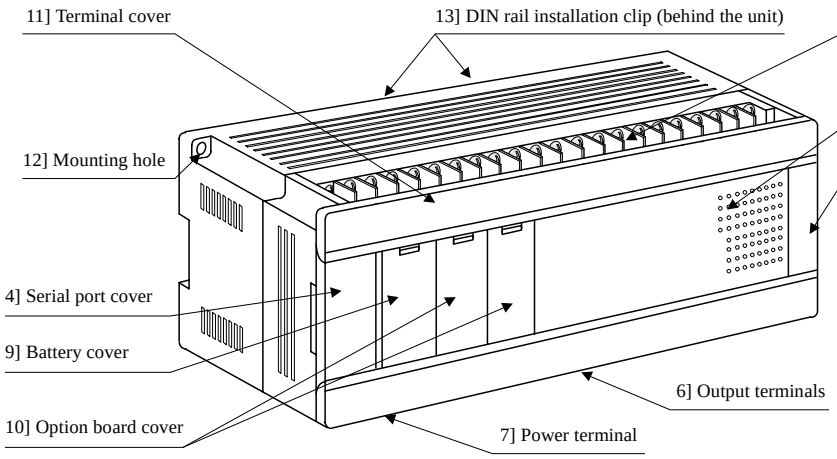
■ LED indication for FLASH memory writing of user program

If a power supply is turned off during FLASH memory writing, "user memory error (error code 31)" may occur at the next time of a power supply ON.
In the current MICRO-EH, it was monitored in special internal output(R7EF). In MICRO64, this can be visually checked in OK LED.

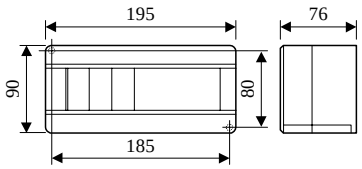
Chapter 2 MICRO64 Unit

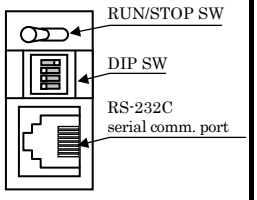
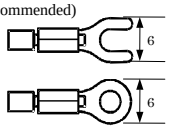
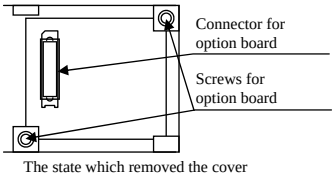
2.1 Name and function of each part

Type	EH-A64DR, EH-D64DR, EH-D64DT, EH-D64DTPS	
Weight	EH-A64DR : 0.72 kg	EH-D64DR : 0.64 kg
	EH-D64DT : 0.72 kg	EH-D64DTPS : 0.64 kg



Dimensions (mm)

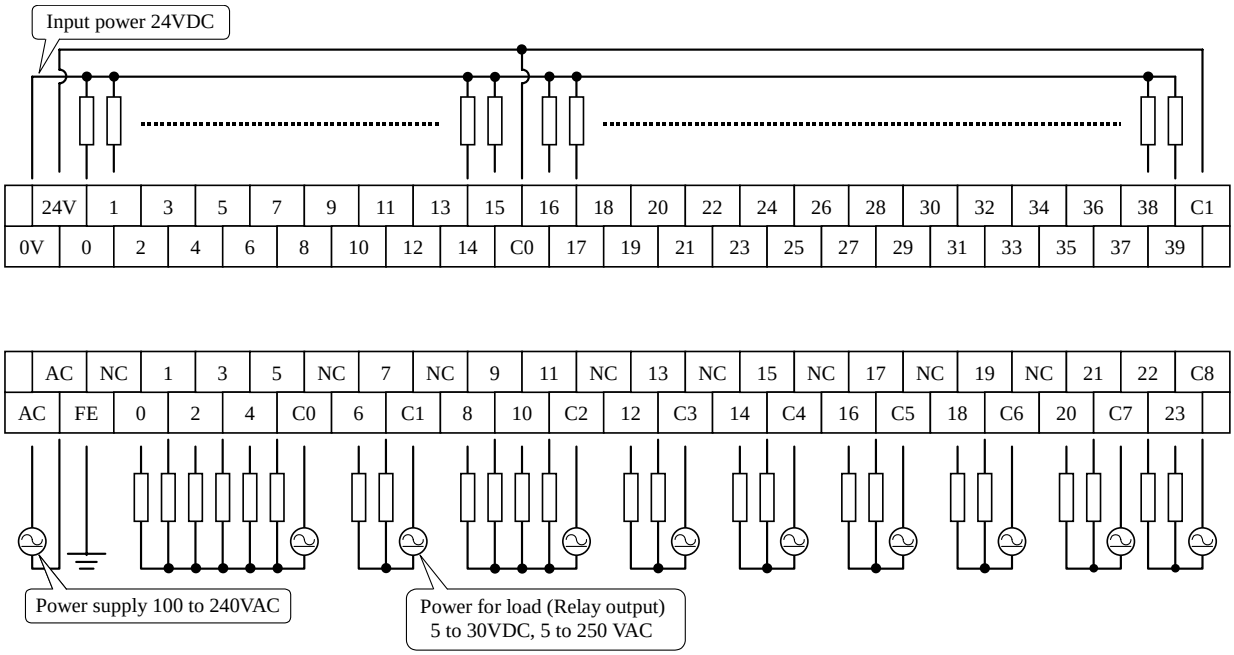


No.	Item	Detailed explanation
1]	POW LED	Lighting when the power is supplied.
2]	OK LED	Lighting at normal operation. (The 64 pts. type displays under FLASH memory backup in OK LED. Please refer to "Chapter 3 Programming" for details.)
3]	RUN LED	Lighting at RUN status.
4]	Serial port cover	Cover for the connector for connecting peripheral units, the RUN switch and DIP switch. When the cover is opened, the RUN switch, RS-232C serial port 1 (PORT 1) and DIP switch can be used. The communication specification is set to port 1. 
5]	Input terminals	Terminals for wiring the external input units. Recommended terminals are shown in the figure to the right. One piece of AWG14 to AWG22 (2.1 to 0.36 mm^2) or two pieces of AWG16 to AWG22 (1.3 to 0.36 mm^2) per terminal may be wired. (Recommended)  (Make sure that the terminals will not disengage due to loose screws.)
6]	Output terminals	Terminals for connecting the external load. The wiring specification is the same as for the input terminals.
7]	Power terminal	Terminal for connecting the power supply. The wiring specification is the same as for the input terminals.
8]	Expansion cover	Cover for the expansion connector
9]	Battery cover	Cover for the backup battery storage unit.
10]	Option board cover	Cover for the option board attachment part. In case of attach the option board, this cover is removed.  The state which removed the cover
11]	Terminal cover	Cover for terminals
12]	Mounting hole	Used when installing the PLC with screws
13]	DIN rail installation clip	Used when installing the PLC on a DIN rail

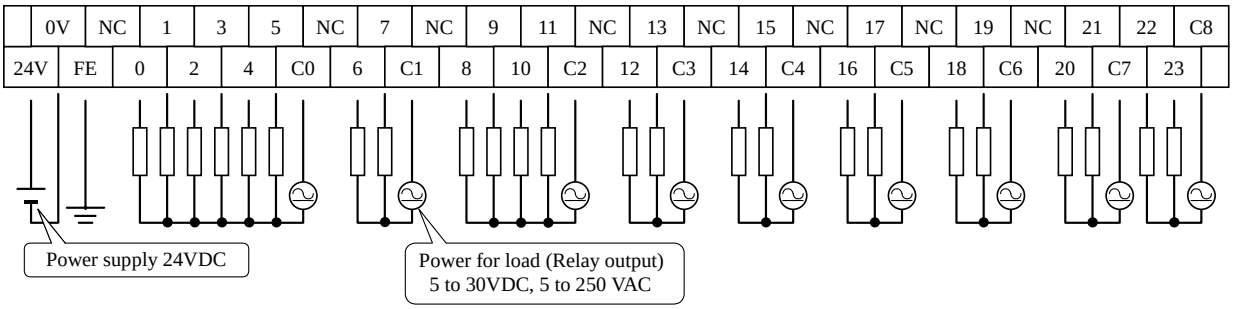
■ Terminal layout and wiring

EH-A64DR (AC power type)

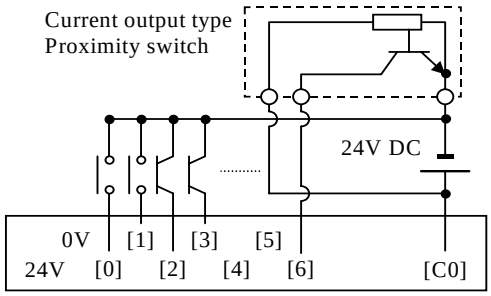
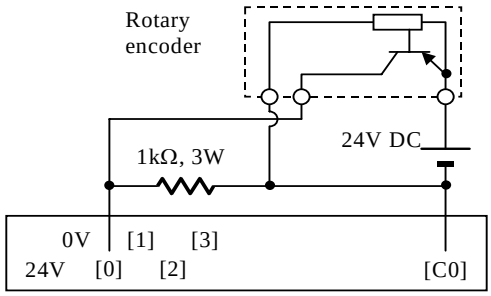
* For the DC input, both sink and source type are available. It is possible to reverse the polarity of 24VDC.



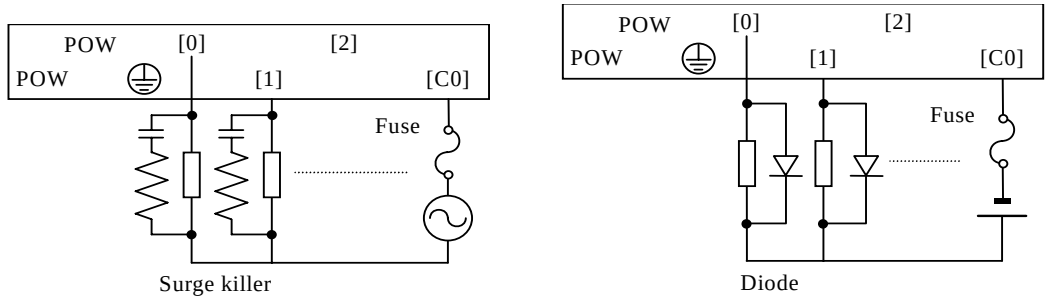
EH-D64DR (DC power type) (Input wiring is same as EH-A64DR)



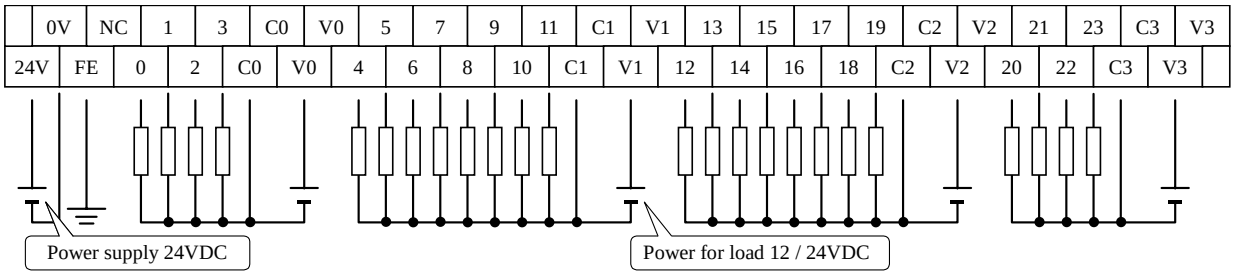
Wiring to the input terminals

Item	DC input	DC input (High Speed Counter)
External wiring		 < Note > In case the maximum count speed is more than 30kHz in 2-phase count or 60kHz in single phase, additional resister is needed as shown in diagram.

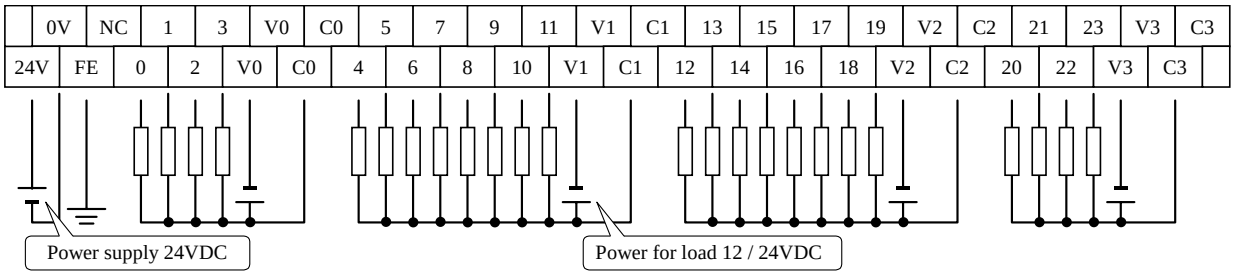
Wiring to the output terminals

Item	Relay output (EH-*64DR)
External wiring	

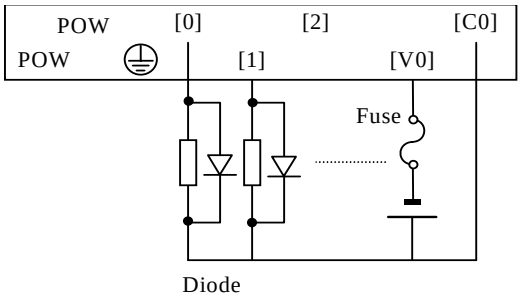
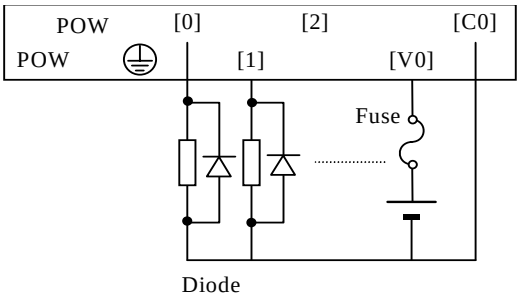
EH-D64DTPS (DC power type) (Input wiring is same as EH-A64DR)



EH-D64DT (DC power type) (Input wiring is same as EH-A64DR)



Wiring to the output terminals

Item	Transistor output (sink type) (EH-*64DT)	Transistor output (source type) (EH-*64DTPS)
External wiring		

2.2 General Specifications

<table>
<tr><td>Item</td><td colspan="5">Specification</td></tr>
<tr><td>Power supply type</td><td colspan="4">AC</td><td colspan="2">DC</td></tr>
<tr><td>Power voltage</td><td colspan="4">100/110/120 V AC (50/60 Hz),
200/220/240 V AC (50/60 Hz)</td><td colspan="2">24 V DC</td></tr>
<tr><td>Power voltage fluctuation range</td><td colspan="4">85 to 264 V AC wide range</td><td colspan="2">19.2 to 30 V DC</td></tr>
<tr><td rowspan="3">Current consumption</td><td colspan="2">100 V AC</td><td colspan="2">264 V AC</td><td>Normal</td><td>Rush</td></tr>
<tr><td>Normal</td><td>Rush</td><td>Normal</td><td>Rush</td><td rowspan="2">0.5 A</td><td rowspan="2">0.6 A</td></tr>
<tr><td>0.4 A</td><td>15 A</td><td>0.2 A</td><td>40 A</td></tr>
<tr><td>Allowable momentary power failure</td><td colspan="4">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</td><td colspan="2">19.2 to 30 V DC:
For a momentary power failure of less than 10 ms, operation continues</td></tr>
<tr><td>Operating ambient temp.</td><td colspan="6">0 to 55 °C</td></tr>
<tr><td>Storage ambient temp.</td><td colspan="6">-10 to 75 °C</td></tr>
<tr><td>Operating ambient humidity</td><td colspan="6">5 to 95 % RH (no condensation)</td></tr>
<tr><td>Storage ambient humidity</td><td colspan="6">5 to 95 % RH (no condensation)</td></tr>
<tr><td>Vibration proof</td><td colspan="6">Conforms to JIS C 0911</td></tr>
<tr><td>Noise resistance</td><td colspan="6">O 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 our measuring method.)
O Based on NEMA ICS 3-304
O Static noise: 3,000 V at metal exposed area
O Conforms with EN50081-2 and EN50082-2</td></tr>
<tr><td>Supported standards</td><td colspan="6">Conforms with UL, CE markings and C-TICK</td></tr>
<tr><td>Insulation resistance</td><td colspan="6">20 MΩ or more between the AC external terminal and the protection earth (PE) terminal (based on 500 V DC megger)</td></tr>
<tr><td>Dielectric withstand voltage</td><td colspan="6">1,500 V AC for one minute between the AC external terminal and the protection earth (PE) terminal</td></tr>
<tr><td>Grounding</td><td colspan="6">Class D dedicated grounding (grounded by a power supply module)</td></tr>
<tr><td>Environment used</td><td colspan="6">No corrosive gases and no excessive dirt</td></tr>
<tr><td>Structure</td><td colspan="6">Attached on an open wall</td></tr>
<tr><td>Cooling</td><td colspan="6">Natural air cooling</td></tr>
</table>

2.3 Performance Specifications

Spec.	Item			64 pts. type	[Reference] 28 pts. type
Control Spec.	CPU			32-bit RISC processor	
	Processing system			Stored program cyclic system	
	Processing Speed	Basic		0.9 μs / instruction	
		Application		Several 10 μs / instruction	
	User program memory			**16 k**steps max. (FLASH memory)	3 ksteps max. (FLASH memory)
Operation Spec.	Ladder	Basic		39 types such as ⊣⊢ ⊣/⊢ ⊣⊢ ⊣/⊢ —○⊣	
		Arithmetic Application		**132** types such as arithmetic, application, control, FUN, etc.	78 types such as arithmetic, application, control, FUN, etc.
I/O processing Spec.	External I/O	I/O processing system		Refresh processing	
		Max. number of points		**176** pts.	140 pts.
	Internal output	Bit		1,984 pts. (R0 to R7BF)	
		Word		**32,768** words (WR0 to **WR7FFF**)	4,096 words (WR0 to WRFFF)
		Special	Bit	64 pts. (R7C0 to R7FF)	
			Word	512 words (WRF000 to WRF1FF)	
		Bit/Word shared		16,384 pts. 1,024 words (M0 to M3FFF, WM0 to WM3FF)	
	Timer / counter	Number of points		512 pts. (TD+CU) However, TD is up to 256 pts. *1	
		Timer set value		0 to 65,535, timer base 0.01 s, 0.1 s, 1 s (64 pts. are maximum for 0.01 s *2)	
		Counter set value		1 to 65,535 times	
	Edge detection			512 pts. (DIF0 to DIF511:decimal) + 512 pts. (DFN0 to DFN511:decimal)	
Peripheral equipment	Program system			Command language, ladder program	
	Peripheral unit			Programming software (LADDER EDITOR DOS version / Windows® version, Pro-H) Command language programmer, portable graphic programmer cannot be used.	

*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.

2.4 Power Supply for Sensor

The 24 V terminal at the input terminal part can supply current to external equipment.
If this terminal is used as the power supply for the input part of this unit, the remaining can be used as power supply for the sensors.
The following current (I) can be supplied as power supply for the sensors.

EH-A64DR / EH-D64DR (64pts type basic unit)

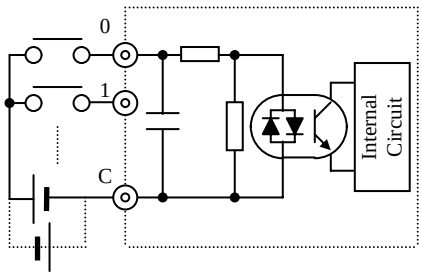
I = 435 mA – (5 mA* × number of input points that are turned on at the same time)

– (5 mA × number of output points that are turned on at the same time)

* : X0, X2, X4, X6 - 10mA.

2.5 Input specifications

Item		Specification		Internal Circuit
		X0, X2, X4, X6	Except the following	
Input voltage		24V DC		
Allowable input voltage range		0 to 30V DC		
Input impedance		Approximately 2.7 kΩ	Approximately 4.7 kΩ	
Input current		8 mA typical	4.8 mA typical	
Operating voltage	ON voltage	18 VDC (min) / 4.5mA (max)	18 VDC (min) / 3.3mA (max)	
	OFF voltage	5 VDC (min) / 1.8mA (max)	5 VDC (max) / 1.6mA (max)	
Input lag	OFF ➔ ON	2 to 20 ms (user setup is possible.) *		
	ON ➔ OFF	2 to 20 ms (user setup is possible.) *		
Number of input points		40 pts. (Refer to "2.1 terminal arrangement and wiring")		
Number of common points		2 pts. (Refer to "2.1 terminal arrangement and wiring")		
Polarity		None		
Insulation system		Photocoupler insulation		
Input display		LED (Green)		
External connection		Removable type screw terminal block (M3)		



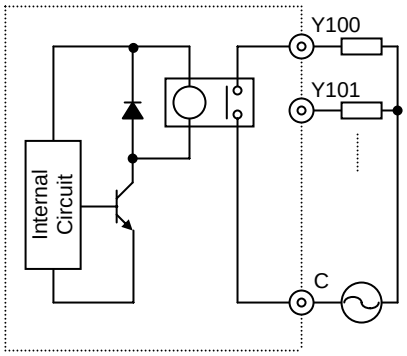
- The digital filter of MICRO64 is 2 - 20ms (WRF07F setting values 4-40). If 0-3 are set up, it will become a setup for 2ms.
- There is 2ms delay by hardware. If set up the filter time at 2ms, actual delay is from 2ms to 4ms.

■ High speed counter

Item		Single	2-phase
Choices for counter input channels		X0, X2, X4, X6	Use X0 and X2 in pair / Use X4 and X6 in pair
Input voltage	ON	18 V	
	OFF	5 V	
Width of count pulse		10 μs	17 μs
Maximum count frequency		100 kHz	60 kHz
Count register		16 bits / 32 bits (depend on operation mode)	
Coincidence output		Possible (or assigned as standard output)	
ON / OFF preset		Possible (or assigned as standard output)	
Upper / lower limit setting		Impossible (16 bits counter : ring counter … 0 to 65,535) (32 bits counter : ring counter … 0 to 4,294,967,295)	
Pre-load / Strobe		Possible (or assigned as standard input)	

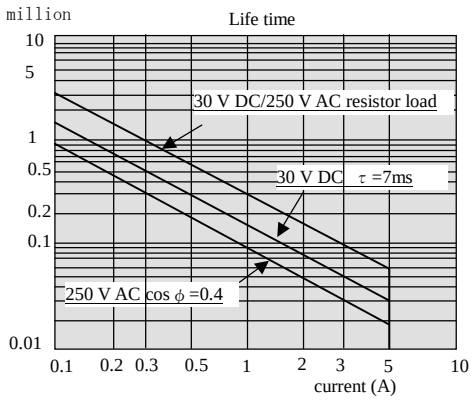
2.6 Output specifications

(1) Relay output

Item		Specification	Internal Circuit
Rated load voltage		5 to 250V AC, 5 to 30V DC	
Minimum switching current		1 mA	
Maximum load current	1 circuit	2A (24V DC, 240V AC)	
	1 common	5A	
Output response time	OFF ➔ ON	15 ms (max)	
	ON ➔ OFF	15 ms (max)	
Number of output points		24 pts. (Refer to "2.1 terminal arrangement and wiring")	
Number of common points		9 pts. (Refer to "2.1 terminal arrangement and wiring")	
Surge removal circuit		None	
Fuse		None	
Insulation system		Relay insulation	
Output display		LED (Green)	
External connection		Removable type screw terminal block (M3)	
Externally supplied power (For driving relays)		Not used	
Contact life*1		20,000,000 times (mechanical) 200,000 times (electrical : 2A)	
Insulation		1500V or more (external - internal) 500V or more (external - external)	

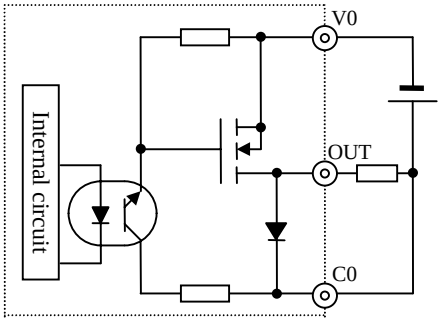
*1 : Please refer to the following figure.

■ Life of relay contacts



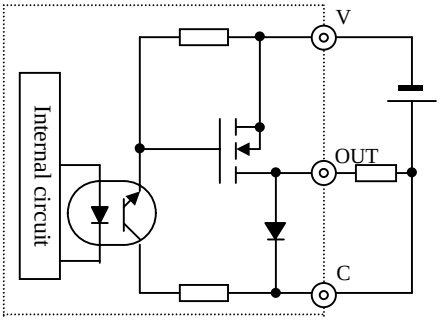
Since the lifetime of relay contact is in inverse proportion to squared current, be aware that interrupting rush current or directly driving the condenser load will drastically reduce the life of the relay.
If switching frequency is very high, transistor output is recommended to use.

(2) DC output … LCDC-low Current (Y100 - Y103)

Item		Specification	Circuit diagram
Output specification		Transistor output	Sink type [EH-D64T] (Y100-Y103) 
Rated load voltage		24/12 V DC (+10 %, -15 %)	
Minimum switching current		1 mA	
Leak current		0.1 mA (max)	
Maximum load current	1 circuit	0.5 A 24 V DC / 0.3 A 12 V DC	
	1 common	2.0 A	
Output response time	OFF ➔ ON	1 μs (max) 24 V DC 0.2A	
	ON ➔ OFF	1 μs (max) 24 V DC 0.2A	
Number of output points		4 pts. (Refer to “2.1 terminal arrangement and wiring”)	
Number of common *1		1 pts. (Refer to “2.1 terminal arrangement and wiring”)	
Surge removing circuit		None	
Fuse		None	
Insulation system		Photocoupler insulation	
Output display		LED (green)	
External connection		Removable type screw terminal block (M3)	
Externally supplied power *2		12 to 30 V DC	
Insulation		1500 V or more (external-internal) 500 V or more (external-external)	
Output voltage drop		0.3 V DC (max)	

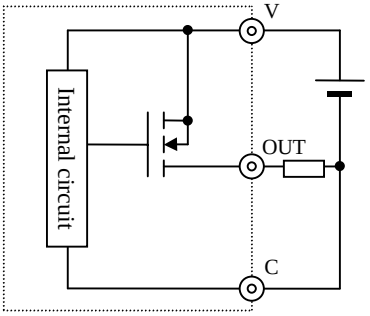
*1: V and C terminals are separated each output terminal. Refer to “2.1 terminal arrangement and wiring” for further information.
*2: It is necessary to supply 12 to 30 V DC between the V and C terminals externally.

(3) DC output … LCDC-low Current (Y104 - Y123)

Item		Specification	Circuit diagram
Output specification		Transistor output	Source type [EH-D64T] (Y104-Y123) 
Rated load voltage		24/12 V DC (+10 %, -15 %)	
Minimum switching current		1 mA	
Leak current		0.1 mA (max)	
Maximum load current	1 circuit	0.5 A 24 V DC / 0.3 A 12 V DC	
	1 common	2.0 A	
Output response time	OFF ➔ ON	1 μs (max) 24 V DC 0.2A	
	ON ➔ OFF	1 μs (max) 24 V DC 0.2A	
Number of output points		4 pts. (Refer to “2.1 terminal arrangement and wiring”)	
Number of common *1		1 pts. (Refer to “2.1 terminal arrangement and wiring”)	
Surge removing circuit		None	
Fuse		None	
Insulation system		Photocoupler insulation	
Output display		LED (green)	
External connection		Removable type screw terminal block (M3)	
Externally supplied power *2		12 to 30 V DC	
Insulation		1500 V or more (external-internal) 500 V or more (external-external)	
Output voltage drop		0.3 V DC (max)	

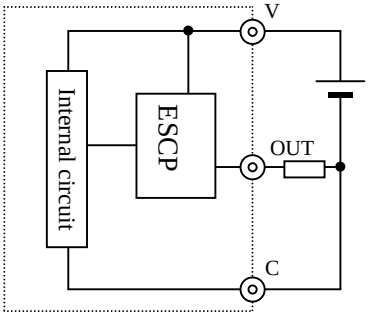
*1: V and C terminals are separated each output terminal. Refer to “2.1 terminal arrangement and wiring” for further information.
*2: It is necessary to supply 12 to 30 V DC between the V and C terminals externally.

(4) DC output … LCDC-low Current (Y100 - Y103)

Item		Specification	Circuit diagram
Output specification		Transistor output	Source type [EH-D64TPS] (Y100-Y103) 
Rated load voltage		24/12 V DC (+10 %, -15 %)	
Minimum switching current		1 mA	
Leak current		0.1 mA (max)	
Maximum load current	1 circuit	0.5 A 24 V DC / 0.3 A 12 V DC	
	1 common	2.0 A	
Output response time	OFF ➔ ON	1 μs (max) 24 V DC 0.2A	
	ON ➔ OFF	1 μs (max) 24 V DC 0.2A	
Number of output points		4 pts. (Refer to “2.1 terminal arrangement and wiring”)	
Number of common *1		1 pts. (Refer to “2.1 terminal arrangement and wiring”)	
Surge removing circuit		None	
Fuse		None	
Insulation system		Photocoupler insulation	
Output display		LED (green)	
External connection		Removable type screw terminal block (M3)	
Externally supplied power *2		12 to 30 V DC	
Insulation		1500 V or more (external-internal) 500 V or more (external-external)	
Output voltage drop		0.3 V DC (max)	

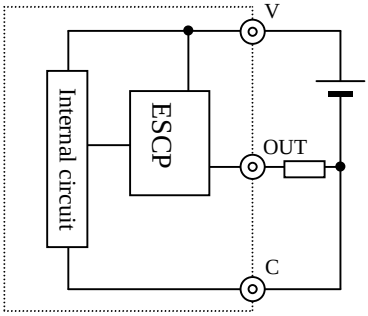
*1: V and C terminals are separated each output terminal. Refer to “2.1 terminal arrangement and wiring” for further information.
*2: It is necessary to supply 12 to 30 V DC between the V and C terminals externally.

(5) DC output (ESCP type) … LCDC-Low Current (Y104-Y119)

Item		Specification	Circuit diagram
Output specification		Transistor output	Source type [EH-D64TPS] (Y104-Y119) 
Rated load voltage		24/12 V DC (+10 %, -15 %)	
Minimum switching current		10 mA	
Leak current		0.1 mA (max)	
Maximum load current	1 circuit	0.7 A	
	1 common	2.8 A	
Output response time	OFF ➔ ON	1 ms (max) 24 V DC	
	ON ➔ OFF	1 ms (max) 24 V DC	
Number of output points		16 pts. (Refer to “2.1 terminal arrangement and wiring”)	
Number of common *1		2 pts. (Refer to “2.1 terminal arrangement and wiring”)	
Surge removing circuit		None	
Fuse		None	
Insulation system		Photocoupler insulation	
Output display		LED (green)	
External connection		Removable type screw terminal block (M3)	
Externally supplied power *2		12 to 30 V DC	
Insulation		1500 V or more (external-internal) 500 V or more (external-external)	
Output voltage drop		0.3 V DC (max)	

*1: V and C terminals are separated each output terminal. Refer to “2.1 terminal arrangement and wiring” for further information.
*2: It is necessary to supply 12 to 30 V DC between the V and C terminals externally.

(6) DC output (ESCP type) … HCDC-High Current (Y120-Y123)

Item		Specification	Circuit diagram
Output specification		Transistor output	Source type [EH-D64TPS] (Y120-Y123) 
Rated load voltage		24/12 V DC (+10 %, -15 %)	
Minimum switching current		10 mA	
Leak current		0.1 mA (max)	
Maximum load current	1 circuit	1.0 A	
	1 common	3.0 A	
Output response time	OFF ➔ ON	1 ms (max) 24 V DC	
	ON ➔ OFF	1 ms (max) 24 V DC	
Number of output points		4 pts. (Refer to “2.1 terminal arrangement and wiring”)	
Number of common *1		1 pts. (Refer to “2.1 terminal arrangement and wiring”)	
Surge removing circuit		None	
Fuse		None	
Insulation system		Photocoupler insulation	
Output display		LED (green)	
External connection		Removable type screw terminal block (M3)	
Externally supplied power *2		12 to 30 V DC	
Insulation		1500 V or more (external-internal) 500 V or more (external-external)	
Output voltage drop		0.3 V DC (max)	

*1: V and C terminals are separated each output terminal. Refer to “2.1 terminal arrangement and wiring” for further information.
*2: It is necessary to supply 12 to 30 V DC between the V and C terminals externally.

■ Pulse train output / PWM output

Item	64 pts. type Transistor output
Available outputs	Y100-Y103 (optional)
Load voltage	12 / 24 V
Minimum load current	1 mA
PWM max. output frequency	65,535 Hz
Pulse train max. output frequency	65,535 Hz

* : Please do not use a relay output type as a pulse output.

Chapter 3 Programming

3.1 Memory size and Memory assignment

Table 3.1 lists the programming specifications for the MICRO64.

Table 3.1 Programming specifications

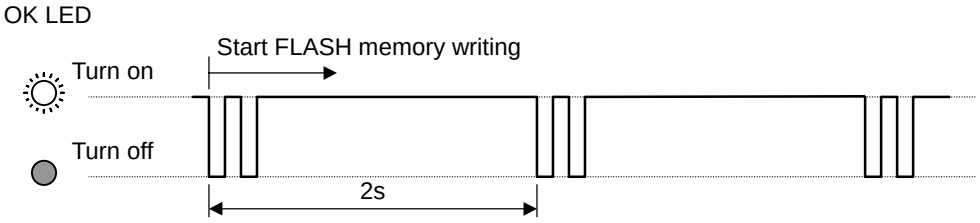
No.	ITEM		64 pts type	[Reference] 28 pts type
1	Program size		16k steps	3 k steps (3,072 steps)
2	Memory assignment		RAM-16H	RAM-04H
3	Instruction size		32 bits / 1step	
4	Memory specification	SRAM	Backup with optional battery.	
		FLASH	Backup without battery.	
5	Program language		H-series ladder/instruction language	
6	Program creation		Created with H-series programming devices	
7	Program modification	in STOP status	Possible by programming software.	
		in RUN status	Possible (Online change in RUN) by programming software. (except for control commands.)*1 (While online change in RUN, PLC operation momentarily stops.).	
7	Off line CPU type		H-302*2	H-302 or MICROEH

*1 : Refer to the peripheral unit manual for details.

*2 : If the off-line CPU type is set as "MICROEH" in LADDER EDITOR for Windows ® before Ver.3.05, it becomes impossible to choose RAM-16H. In this case, the off-line CPU type should choose H-302.

Caution

The MICRO-EH series backup user programs in the FLASH memory.
In order to shorten the program transfer time, user program is transferred once to the operation execution memory (SRAM), and transfer operation is completed seen from programming software. Then backup copying to FLASH memory starts afterwards. Do not turn off the power to the PLC within approximately two minutes after program downloading. If the power is turned off within two minutes, a user memory error (31H) may occur. Note that the transfer completion to the FLASH memory can be confirmed by the special internal output (R7EF).
In MICRO64, this can be visually checked in OK LED. While FLASH memory is being written, OK LED blinks as follows.



3.2 I/O assignment

The I/O assignment and the I/O address of each unit are shown below.

table 3.2 I/O assignment and I/O address of each unit

Unit		Assignment	64 pts type	[Reference] 28 pts type
Basic	Digital	Slot 0 : X48	X0-39	X0-15
		Slot 1 : Y32	Y100-123	Y100-111
		Slot 2 : Empty	Empty16	Empty16
Exp.1	Digital	Unit 1 / Slot 0 : B1/1	X1000-1007 / 1015 (14 / 28 pts)	
			Y1016-1021 / 1027 (14 / 28 pts)	
	Analog	Unit 1 / Slot 0 : FUN0	WX101-104 (WX100 is used by the system.)	
			WY106-107 (WY105 is used by the system.)	
Exp.2	Digital	Unit 2 / Slot 0 : B1/1	X2000-2007 / 2015 (14 / 28 pts)	
			Y2016-2021 / 2027 (14 / 28 pts)	
	Analog	Unit 2 / Slot 0 : FUN0	WX201-204 (WX200 is used by the system.)	
			WY206-207 (WY205 is used by the system.)	
Exp.3	Digital	Unit 3 / Slot 0 : B1/1	X3000-3007 / 3015 (14 / 28 pts)	
			Y3016-3021 / 3027 (14 / 28 pts)	
	Analog	Unit 3 / Slot 0 : FUN0	WX301-304 (WX300 is used by the system.)	
			WY306-307 (WY305 is used by the system.)	
Exp.4	Digital	Unit 4 / Slot 0 : B1/1	X4000-4007 / 4015 (14 / 28 pts)	
			Y4016-4021 / 4027 (14 / 28 pts)	
	Analog	Unit 4 / Slot 0 : FUN0	WX401-404 (WX400 is used by the system.)	
			WY406-407 (WY405 is used by the system.)	

3.3 Internal output, Edge, Timer

The capacity of an internal output and the number of edge, timers is shown below.

Table 3.3 List of Internal output, Edge, Timer

Function		Symbol	Size		Name	64 pts type	Ref. 28 pts type
						Number of points	Number of points
Internal I/O	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	32,768 words	4,096 words
		DR	D	16	Double word internal output		
		WR	W	16	Word special internal output	512 words	
		DR	D	16	Double word special internal output		
	Sharing of bit / word	M	B	16	Bit internal output	16,384 points	
		WM	W	16	Word internal output	1,024 words	
		DM	D	16	Double internal output		
Others	Edge detection	DIF	B	10	Leading edge	512 words	
		DFN	B	10	Trailing edge	512 words	
	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 + Counter Total 512 points* (Timer is to 256 pts)	Timer + Counter Total 256 points*
		SS	B	10	Single shot timer		
		CU	B	10	Up counter		
		CTU	B	10	Up-down counter up input		
		CTD	B	10	Up-down counter down input		
		CL	B	10	Clear progress value		

* The same timer counter number cannot be used more than once.

Chapter 4 Special I/O

4.1 Introduction

Standard I/O of MICRO-EH can be used as counter input, interruption input, pulse output and a PWM output. In order to use those functions, "operation mode" must be configured at first. In addition to existing mode for the current MICRO-EH, MICRO64 has new mode of 32-bit counter.

This chapter describes this new additional mode only. (Please refer to a MICRO-EH application manual about other operation modes.)

4.2 Setting of Special I/O

The procedure to switch from standard I/O to either counter input or pulse output is shown below.

[Step 1] Setting of each parameter

1) Set operation mode No. to WRF070. (MICRO64 addition mode: H20 to 23)

➔ Please refer to "4.3 Operation mode" about operation mode.

2) Set the function of each I/O to WRF071.

➔ Please refer to "4.4 Function setting of I/O terminal" about function of I/O terminal.

3) Set parameters or conditions to WRF1B0 - WRF1C7.

➔ Please refer to "(1) Parameter setting"of each function about detail of condition.

[Step 2] Enable configuration

Set R7F5 to high to enable above configuration.

[Step 3] Control of special I/O

If no error is found in Step2, configuration is completed. Special I/O function is available on user program.

➔ Please refer to "(4) Errors in mode setting" of each function about detail of setting errors.

[Step 4] Save configuration parameters

If necessary, set R7F6 to high to save configuration parameters in FLASH memory. Once parameters are saved in FLASH memory, above configuration is not necessary in the next power ON time.

4.3 Operation mode

In operation modes 20 – 23, each I/O is divided into 4 groups as below, and configured per every group. Both single phase counters and 2-phase counters can be used as 32-bit counter.

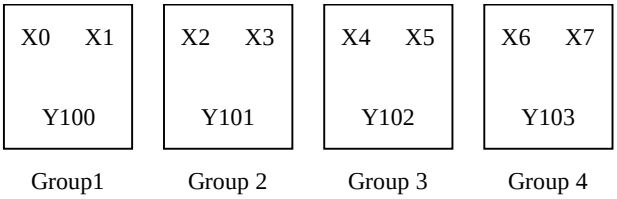


Figure 4.1 Overview of special I/O group

Table 4.1 Special I/O operation mode

Mode No. (WRF070)	Input			Output	
	Single-phase counter	2-phase counter	Interrupt	Pulse	PWM
20 H	4 ch	0 ch	4 ch	4 ch	4 ch
21 H	2 ch	1 ch	2 ch	3 ch	3 ch
22 H	2 ch	1 ch	2 ch	3 ch	3 ch
23 H	0 ch	2 ch	0 ch	2 ch	2 ch

* Channel number shown in above table is the maximum number. Channel number that can be used decreases by combination of I/O function.

Example) 2ch. of 2-phase counter : WRF070 ➔ H0023

4.4 Function setting of special I/O

Each I/O function is configured in WRF071 for every group.

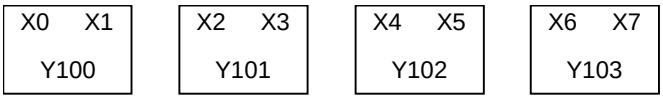
WRF071 is divided to 4 groups, and every 4 bits are assigned to every group.

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRF071 :	Group 1				Group 2				Group 3				Group 4			
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4.2 Special internal output for an I/O functional detailed setup

■ Mode 20

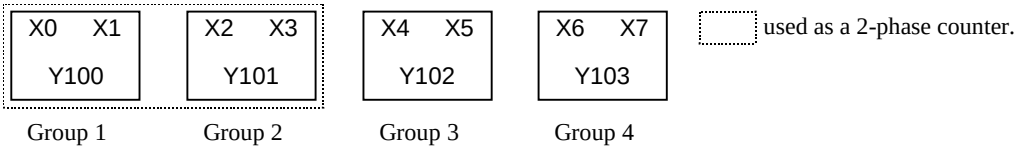
Groups 1-4 choose a function from special I/O(A).



■ Mode 21

Groups 1 choose a function from special I/O(B). Groups 2 choose a function from special I/O(C).

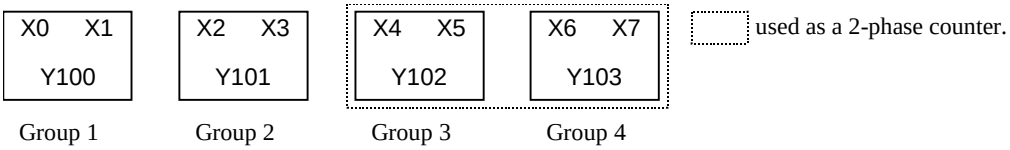
Groups 3,4 choose a function from special I/O(A).



■ Mode 22

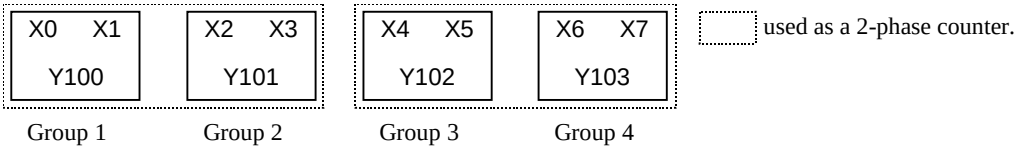
Groups 1,2 choose a function from special I/O(A).

Groups 3 choose a function from special I/O(B). Groups 4 choose a function from special I/O(C).



■ Mode 23

Groups 1,3 choose a function from special I/O(B). Groups 2,4 choose a function from special I/O(C).



Refer to the table (Table 4.2 to 4.4) for the setting value of special I/O(A)(B)(C). It inputs into WRF071 combining the setting value of a table. Refer to the next page for Tables 4.2-4.4.

< Note >

Even if the software of Ver.0100 sets up PWM or pulse output in the modes 20-23, it does not operate.

Table 4.2 The function which can be set up, and its setting value in mode 20 - 22

Setting Value	Xn	Xn+1	Ym
0 H	Standard input	Standard input	Standard output
1 H			PWM output "n"
2 H			Pulse output "n"
3 H		Interrupt input	Standard output
4 H			PWM output "n"
5 H			Pulse output "n"
6 H	Counter input "n"	Standard input	Standard output
7 H			Counter output
8 H		Pre-load input "n"	Standard output
9 H			Counter output
A H		Pre-strobe input "n"	Standard output
B H			Counter output
Except the above	Standard input	Standard input	Standard output

n : Group No.

Table 4.3 Function and setting value of group 1,3 in mode 21 - 23

Setting Value	Xn	Xn+1	Ym
0 H	Counter xA	Starndard input	Standard output
1 H			Counter output
2 H		Pre-load input x	Standard output
3 H			Counter output
4 H		Pre-strobe input x	Standard output
5 H			Counter output
Except the above	Counter xA	Starndard input	Standard output

Table 4.4 Function and setting value of group 2,4 in mode 21 - 23

Setting Value	Xn+2	Xn+3	Ym+1
0 H	Counter xB	Counter xZ	Standard output
1 H			PWM output
2 H			Pulse output
3 H		Standard input	Standard output
4 H			PWM output
5 H			Pulse output
Except the above	Counter xB	Counter xZ	Standard output

■ Setting example 1 (Mode 20)

Group	Function			Table	Value
1	X0 : Standard input	X1 : Standard input	Y100 : Standard output	4.2	➔ 0H
2	X2 : Counter input 2	X3 : Pre-load input 2	Y101 : Standard output	4.2	➔ 8H
3	X4 : Counter input 3	X5 : Standard input	Y102 : Coincidence output	4.2	➔ 7H
4	X6 : Standard input	X7 : Interrupt input	Y103 : Pulse output	4.2	➔ 5H

WRF071 ➔ 0875H

■ Setting example 1 (Mode 21)

Group	Function			Table	Value
1	X0 : Counter 1A	X1 : Pre-strobe input	Y100 : Standard output	4.3	➔ 4H
2	X2 : Counter 1B	X3 : Counter input 1Z	Y101 : Standard output	4.4	➔ 0H
3	X4 : Standard input	X5 : Standard input	Y102 Pulse output	4.2	➔ 2H
4	X6 : Standard input	X7 : Interrupt input	Y103 PWM output	4.2	➔ 4H

WRF071 ➔ 4024H

4.5 High Speed Counter (HSC)

(1) Parameter setting

■ Setting of on-preset

If counter output is used, set counter value that counter output is turned on (the on-preset value). Possible range is from 0 to FFFFFFFFH (0 to 4,294,967,295). If the on-preset value is set as same value as the off-preset value, the counter will not perform any counting operation.

On-preset value of Counter 1 :	WRF1B1 (High word)	WRF1B0 (Low word)
On-preset value of Counter 2 :	WRF1B3 (High word)	WRF1B2 (Low word)
On-preset value of Counter 3 :	WRF1B5 (High word)	WRF1B4 (Low word)
On-preset value of Counter 4 :	WRF1B7 (High word)	WRF1B6 (Low word)

Figure 4.3 Special internal outputs for setting the on-preset values

When counter is not configured, the above special internal outputs are used for other purpose.

■ Setting of off-preset

If counter output is used, set counter value that counter output is turned off (the off-preset value). Possible range is from 0 to FFFFFFFFH (0 to 4,294,967,295). If the off-preset value is set as same value as the on-preset value, the counter will not perform any counting operation.

Off-preset value of Counter 1 :	WRF1B9 (High word)	WRF1B8 (Low word)
Off-preset value of Counter 2 :	WRF1BB (High word)	WRF1BA (Low word)
Off-preset value of Counter 3 :	WRF1BD (High word)	WRF1BC (Low word)
Off-preset value of Counter 4 :	WRF1BF (High word)	WRF1BE (Low word)

Figure 4.4 Special internal outputs for setting the off-preset values

When counter is not configured, the above special internal outputs are used for other purpose.

■ Setting of counter pre-load

If pre-load value is used, set pre-load value. Possible range is from 0 to FFFFFFFFH (0 to 4,294,967,295).

Pre-load value of Counter 1 :	WRF1C1 (High word)	WRF1C0 (Low word)
Pre-load value of Counter 2 :	WRF1C3 (High word)	WRF1C2 (Low word)
Pre-load value of Counter 3 :	WRF1C5 (High word)	WRF1C4 (Low word)
Pre-load value of Counter 4 :	WRF1C7 (High word)	WRF1C6 (Low word)

Figure 4.5 Special internal outputs for setting the pre-load values

When counter is not configured, the above special internal outputs are used for other purpose.

(2) Errors in mode setting

If the on-preset and off-preset values are the same, and flag (R7F5) is activated, error bit shown below will be on, and counter does not work. In addition, the setting error flag (R7F7) turns on.

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRF057 :	a	Not used							b	c	d	e	f	g	h	i

Figure 4.6 Special internal output for setting error indication

Bit	Description of error	Related I/O
a	(Total pulse frequency error)	Y100 to Y103
b	(Pulse 4 frequency error)	Y103
c	(Pulse 3 frequency error)	Y102
d	(Pulse 2 frequency error)	Y101
e	(Pulse 1 frequency error)	Y100
f	Counter 4 preset value error	X6
g	Counter 3 preset value error	X4
h	Counter 2 preset value error	X2
i	Counter 1 preset value error	X0

(3) Control of the counter input by the ladder program

Operation of a counter input is controllable by the ladder program with a FUN command. Moreover, each parameter can be changed.

FUN140	HSC operation control	Start / stop
FUN141	Counter output control	Enable / disable counter output
FUN142	Up / down count setting	Up counter / down counter
FUN143	Write counter value	Write current counter value
FUN144	Read counter value	Read current counter value
FUN145	Clear counter value	Clear counter value
FUN146	Change preset value	Change preset value

* Please refer to "Chapter 8 Additional commands" in the end of this book about the details of the FUN command.

(4) Notes at the time of counter input use

If the pulse of the frequency exceeding specification is inputted, a counter may incorrect-count. When MICRO64 watches a counter value periodically and a counter value changes a lot, it displays that errors occurred on special internal output WRF06A.

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRF06A :	Not used												a	b	c	d

Figure 4.7 Special internal output For an incorrect count display in counter

Bit	Description of abnormality	Related terminal
a	Counter 4 counting error	X6
b	Counter 3 counting error	X4
c	Counter 2 counting error	X2
d	Counter 1 counting error	X0

* The above error flag is cleared by setting error clear bit (R7EC) manually or in user program.

4.6 PWM output

(1) Parameter setting

■ Setting of output frequency

The output frequency (Hz) of a PWM output is set up. The values which can be set up are 0-FFFFH (0-65,535).

*Please be sure to set H0000 to High-WORD.

Output frequency of PWM output 1 :	WRF1B1(Not used H0000)	WRF1B0 (Output frequency)
Output frequency of PWM output 2 :	WRF1B3(Not used H0000)	WRF1B2 (Output frequency)
Output frequency of PWM output 3 :	WRF1B5(Not used H0000)	WRF1B4 (Output frequency)
Output frequency of PWM output 4 :	WRF1B7(Not used H0000)	WRF1B6 (Output frequency)

Figure 4.8 Special Internal output for an output frequency setup

The above-mentioned special internal output is used as a parameter of another purpose by setup of those other than a PWM output.

■ Setting of ON-duty

ON-duty (The rate of ON time: %) of a PWM output is set up. The values which can be set up are 0-64H (0-100). If the value more than 64H (100) is set up, it will operate by 100.

ON-duty of PWM output 1 :	WRF1B9 (Not used H0000)	WRF1B8 (ON-duty)
ON-duty of PWM output 2 :	WRF1BB (Not used H0000)	WRF1BA (ON-duty)
ON-duty of PWM output 3 :	WRF1BD (Not used H0000)	WRF1BC (ON-duty)
ON-duty of PWM output 4 :	WRF1BF (Not used H0000)	WRF1BE (ON-duty)

Figure 4.9 Special Internal output for an ON-duty setup

The above-mentioned special internal output is used as a parameter of another purpose by setup of those other than a PWM output.

(2) Errors in mode setting

PWM output does not have the abnormalities in a parameter.

When output frequency is set as 0Hz, a system sets output frequency as 10Hz.

(3) Control of the PWM output by the ladder program

Operation of a PWM output is controllable by FUN command. Moreover, each parameter can be changed.

FUN147 PWM operation control A start/stop of a PWM output are executed.

FUN148 Frequency/ON-duty changes The parameter of the specified PWM output is changed.

* The FUN command about a PWM output is not to change / addition. For details, please refer to a MICRO-EH application manual.

4.7 Pulse train output

In operation modes 20 – 23, the output pulse-number can be set up by 32 bits(1~4,294,967,295).

Moreover, a maximum output frequency is 65,535Hz.

(1) Parameter setting

■ Setting of output frequency

Output frequency is set as the pulse output to be used. The values which can be set up are 0-FFFFH(0-65,535).

*Please be sure to set H0000 to high word in operation modes 20 – 23.

Output frequency of Pulse output 1 :	WRF1B1(Not used H0000)	WRF1B0 (Output frequency)
Output frequency of Pulse output 2 :	WRF1B3(Not used H0000)	WRF1B2 (Output frequency)
Output frequency of Pulse output 3 :	WRF1B5(Not used H0000)	WRF1B4 (Output frequency)
Output frequency of Pulse output 4 :	WRF1B7(Not used H0000)	WRF1B6 (Output frequency)

Figure 4.10 Special Internal output for an Output frequency setup

The above-mentioned special internal output is used as a parameter of another purpose by setup of those other than a pulse train output.

■ Setting of Pulse output

Output pulse-number is set as the pulse output to be used. The values which can be set up are 0-FFFFFFFFH(0-4,294,967,295).

Output pulse-number of Pulse output 1 :	WRF1C1 (high data)	WRF1C0 (low data)
Output pulse-number of Pulse output 2 :	WRF1C3 (high data)	WRF1C2 (low data)
Output pulse-number of Pulse output 3 :	WRF1C5 (high data)	WRF1C4 (low data)
Output pulse-number of Pulse output 4 :	WRF1C7 (high data)	WRF1C6 (low data)

Figure 4.11 Special Internal output for an Pulse output setup

The above-mentioned special internal output is used as a parameter of another purpose by setup of those other than a pulse train output.

(2) Errors in mode setting

Pulse output does not have the abnormalities in a parameter.

When output frequency is set as 0Hz, a system sets output frequency as 10Hz..

(3) Control of the pulse output by the ladder program

Operation of a pulse output is controllable by FUN command. Moreover, each parameter can be changed.

FUN149	Pulse output control	Pulse output control
FUN150	Pulse frequency setting changes	Pulse frequency output setting changes
FUN151	Pulse output with acceleration/deceleration	Frequency is changed by a start and stop of a pulse output.
FUN153	Pulse output with sequence parameter change	The frequency of a pulse output is changed arbitrarily. *

Please refer to "Chapter 8 Additional commands" in the end of this book about the details of the FUN command.

(4) Notes at the time of pulse output use

A pulse output requires load for system processing. Therefore, while outputting the pulse, command processing time is extended 1.4 times at the maximum. (It is large effect, so that output frequency is high.)

Example) 4ch All pulse outputs are outputted by 65kHz. Scan time 20ms ➔ 28ms

Chapter 5 Communication port

MICRO64 has one RS-232C port. This port can be used as a dedicated port or a general-purpose port. In addition, it has modem control function which communicates from a remote place through a modem.

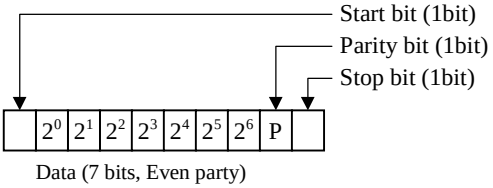
5.1 Dedicated port

The specification of communication port is shown in table 5.1.

The communication port can be connected with the peripheral unit that supports a H-Protocol. (Portable diagram programming tool and instruction language programming tool cannot be used.) By connecting this port with a peripheral unit, created user programs can be transferred, user programs stored in the CPU can be read/verified, and the CPU operating status can be monitored. In addition, remote monitoring system can be built up by HMI ,etc.

Modem function is available in this port also. Please refer to the application manual of MICRO-EH for further information.

Table 5.1 Communication port specification

<table>
<tr><th>Item</th><th colspan="4">Specification</th></tr>
<tr><td rowspan="7">Transmission speed</td><td colspan="3">When peripheral units are connected</td><td>Modem mode</td></tr>
<tr><td colspan="3">4800 bps, 9600 bps, 19.2 kbps,
38.4 kbps</td><td>2400 bps, 4800 bps, 9600 bps, 19.2 kbps,
38.4 kbps, 57.6 kbps</td></tr>
<tr><td>SW1</td><td>SW3</td><td>Transmission speed setting</td><td rowspan="6">Set the transmission speed when connected via modem in the special internal output (WRF01A).</td></tr>
<tr><td>ON</td><td>ON</td><td>38.4 kbps</td></tr>
<tr><td>ON</td><td>OFF</td><td>19.2 kbps</td></tr>
<tr><td>OFF</td><td>ON</td><td>9600 bps</td></tr>
<tr><td>OFF</td><td>OFF</td><td>4800 bps</td></tr>
<tr><td>Communication system</td><td colspan="4">Half duplex</td></tr>
<tr><td>Synchronization system</td><td colspan="4">Start-stop synchronization</td></tr>
<tr><td>Startup system</td><td colspan="4">One-sided startup using the host side command</td></tr>
<tr><td>Transmission system</td><td colspan="4">Serial transmission (bit serial transmission)</td></tr>
<tr><td>Transmission code</td><td colspan="4">ASCII</td></tr>
<tr><td>Transmission code configuration</td><td colspan="4"> 
</td></tr>
<tr><td>Transmission code outgoing sequence</td><td colspan="4">Sent out from the lowest bit in character units</td></tr>
<tr><td>Error control</td><td colspan="4">Vertical parity check, checksum, overrun check, framing check</td></tr>
<tr><td>Transmission unit</td><td colspan="4">Message unit (variable length)</td></tr>
<tr><td>Maximum message length</td><td colspan="4">503 bytes (including control characters)</td></tr>
<tr><td>Interface</td><td colspan="4">Conforms to RS-232C (maximum cable length: 15 m)</td></tr>
<tr><td>Control procedure</td><td colspan="4">H-series dedicated procedure (H-Protocol)
Standard procedure (transmission control procedure 1), Simplified procedure (transmission control procedure 2)</td></tr>
<tr><td>Connector used</td><td colspan="4">CPU side: 8-pin modular connector (RJ-45)</td></tr>
</table>

■ Note

- Portable diagram programming tool and instruction language programming tool cannot be used.
- Please note that if DIP switch 1 is set to On, +12V is output from pin 4.
- If the negative acknowledge command (NAK) is sent from the host using the transmission control procedure 1 or 2, wait at least 10 ms before sending the next text.
- Specify a value of 20 ms or higher for the response TM of the H-protocol. (When the response TM is set to 0, the default value of 20 ms will be used.)

(1) Port settings

Port can be set when the DR signal of port is off. The setting becomes valid when the DR signal is turned on.

1] Setting the DIP switches

Remove the serial port cover on the front case and set the DIP switches according to the below table.

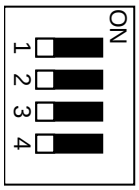


Table 5.2 Setting of DIP SW

SW No.	1	2	3	4	Setting	Remarks
DIPSW	ON	OFF	ON	OFF	38.4 kbps	
	ON	OFF	OFF	OFF	19.2 kbps	Default
	OFF	OFF	ON	OFF	9600 bps	
	OFF	OFF	OFF	OFF	4800 bps	
	OFF	ON	OFF	OFF	Connection via modem	

(do not set SW4 to ON; it is fixed to OFF.)

2] Setting the special internal output

If necessary, set the transmission control procedure and transmission speed in case of modem mode in special internal output WRF01A.

Values in this special internal output is stored in the FLASH memory by setting various setting write request (R7F6) On. Once stored in the FLASH memory, it is not necessary to make the setting again when the power supply is turned on next time.

> Note
> If transmission control procedure 2 is set for port 1 and the special internal output setting is stored in the FLASH memory by R7F6, port 1 starts up with transmission control procedure 2 when the power is turned on next time. Thus, note that the peripheral units that only support transmission control procedure 1 will not be connected.

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRF01A :	a	Not used		b					Not used							
Initial value :	0			0	0	0	0	0								

Figure 5.1 Special internal output for setting port

Area	Setting Value	Content		Remarks	
a	0	Transmission control procedure 1			H0***
	1	Transmission control procedure 2			H8***
b	0	Transmission speed	4800 bps	Setting of bits 8 to 12	00000 (H*0**)
	1	when connecting via modem	9600 bps		00001 (H*1**)
	2		19.2 kbps		00010 (H*2**)
	3		38.4 kbps		00011 (H*3**)
	4		57.6 kbps		00100 (H*4**)
	5		2400 bps		00101 (H*5**)
	Other than above		4800 bps		

(2) Port hardware

The circuit diagram of port and the signal list are shown in Figure 5.2 and Table 5.3 respectively.

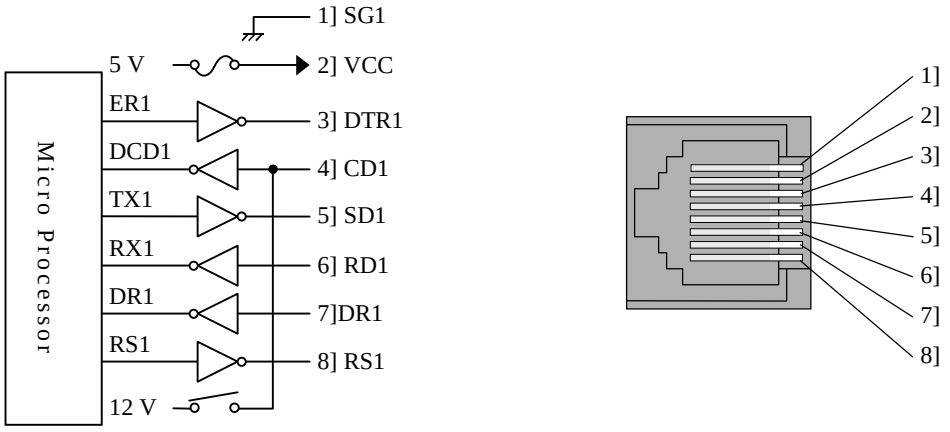


Figure 5.2 Circuit diagram and pin numbers for port

Table 5.3 List of port 1 signals

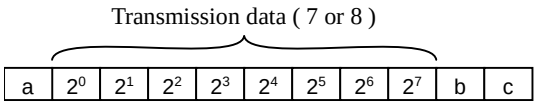
Pin No.	Signal abbreviation	Direction		Meaning
		CPU	HOST	
1]	SG1	⟷		Ground for signals
2]	VCC	→		5 V DC is supplied. (Protective fuse is connected.)
3]	DTR1(ER)	→		Communication enabled signal When this signal is high level, communication is possible.
4]	CD1(DCD)	→		12V is output when DIP switch 1 is turned On.
5]	SD1(TXD)	→		Data sent by the CPU
6]	RD1(RXD)	→		Data received by the CPU
7]	DR1(DSR)	←		Peripheral units connected signal When this signal is high level, indicates that dedicated peripherals are connected.
8]	RS1(RTS)	←		Transmission request signal When this signal is high level, indicates that the CPU can receive data.

5.2 General-purpose port

The communication port can be switched to general-purpose port by command. (General-purpose port works only in RUN status.)

General purpose port is switched by special FUN command (FUN 5) in user program. Communication on the general-purpose port is operated by communication command (TRNS 0) in user program.

Table 5.4 Communication port specifications (general-purpose port)

Item	Specification
Transmission speed	Specifies by TRNS 0 / RECV 0 : 4800 bps, 9600 bps, 19.2 kbps, 38.4 kbps, 57.6 kbps
Communication system	Half duplex
Synchronization system	Start-stop synchronization
Transmission system	Serial transmission (bit serial transmission)
Transmission code configuration	Specifies by TRNS 0 / RECV 0  a : Start bit b : Parity bit (Even / Odd / None) c : Stop bit (1 or 2)
Error control	Vertical parity check, overrun check, framing check
Transmission format	1] Start character & Receiving data length 2] Start character & Stop character 3] Stop character 4] Receiving data length Specification by the format of 1] - 4] is possible.
Sending buffer	1,024 bytes
Receiving buffer	1,024 bytes

Note

In order to use a communication port as a general-purpose port (TRNS 0 / RECV 0 is performed), it is necessary to execute FUN 5 (general-purpose port change command) first.

Please refer to a MICRO-EH application manual about the details of TRNS 0 / RECV 0 / FUN 5.

Chapter 6 Special internal output

6.1 Special internal output (bit)

New added or changed special internal output (bit) for MICRO64 is shown in the following table.

* The other special internal output is the same as existing MICRO-EH.

Table 6.1 Special internal output (Bit) list (add / change)

No.	Name	Meaning	Description	Setting condition	Resetting condition
R7CA	Retentive area error	0: Normal 1: Error	When retentive area is undefined status, this bit is activated.	Set by the system	Cleared by user
R7CB	Processor error	0: Normal 1: Error	When micro processor is in error, this bit is activated.		
R7D8	Clock error	0: Normal 1: Error	When clock IC is in error, this bit is activated.		
R7DF	Option board error	0: Supported 1: Not supported	When unsupported option board is mounted, this bit is activated.		

■Reference Special internal output (bit) list

No.	Name
R7C0	Ignore scan time error (normal scan)
R7C1	Ignore scan time error (periodic scan)
R7C2	Ignore scan time error (interrupt scan)
R7C3	Undefined
R7C4	Undefined
R7C5	Undefined
R7C6	Undefined
R7C7	Online change in RUN allowed
R7C8	Serious error flag
R7C9	Microcomputer error
R7CA	**User memory error**
R7CB	**Processor error**
R7CC	Memory size over
R7CD	I/O configuration error
R7CE	Undefined
R7CF	Undefined
R7D0	Undefined
R7D1	Scan time error (normal scan)
R7D2	Scan time error (periodic scan)
R7D3	Scan time error (interrupt scan)
R7D4	Grammar/assemble error
R7D5	Blown fuse detection
R7D6	Undefined
R7D7	Undefined
R7D8	**Clock IC error**
R7D9	Battery error
R7DA	Undefined
R7DB	Self-diagnostic error
R7DC	Output selection at stop
R7DD	Undefined
R7DE	Undefined
R7DF	**Option board error**

No.	Name
R7E0	Key switch location (STOP)
R7E1	Undefined
R7E2	Key switch location (RUN)
R7E3	1 scan ON after RUN
R7E4	Always ON
R7E5	0.02 second clock
R7E6	0.1 second clock
R7E7	1.0 second clock
R7E8	Occupied flag
R7E9	RUN prohibited
R7EA	Executing a online change in RUN
R7EB	Power off memory
R7EC	Clear error special internal output
R7ED	Undefined
R7EE	Battery error display selection
R7EF	Backup memory writing execution flag
R7F0	Carry flag (CY)
R7F1	Overflow flag (V)
R7F2	Shift data (SD)
R7F3	Operation error (ERR)
R7F4	Data error (DER)
R7F5	PI/O function setting flag
R7F6	Individual setting write request
R7F7	PI/O function setting error
R7F8	Calendar, clock read request
R7F9	Calendar, clock setting request
R7FA	Clock ± 30 second adjustment request
R7FB	Calendar and clock set data error
R7FC	Output control 1
R7FD	Output control 2
R7FE	Output control 3
R7FF	Output control 4

6.2 Special internal output (word)

The special internal output (word) added or changed from MICRO64 is shown in the following table.

* About the special internal output of except the following table, it is the same.

Table 6.2 Special internal output (Word) list (add / change)

<table>
<tr><th>No.</th><th>Name</th><th colspan="2">Meaning / Description</th><th>Setting condition</th><th>Resetting condition</th></tr>
<tr><td>WRF061</td><td>Memory board
Write-protect setting</td><td colspan="2">The memory board (option board) is set up write-protected.
<table><tr><th>Setting</th><th>Value (set by user)</th><th>Display after setting (set by system)</th></tr><tr><td>Write-protected</td><td>H8001</td><td>H0001</td></tr><tr><td>Write-protected cancel</td><td>H8000</td><td>H0000</td></tr></table></td><td>Set by user</td><td>Reset by user</td></tr>
<tr><td>WRF062</td><td>Memory board
Status</td><td colspan="2">The state of a memory board (option board) is displayed.
15 14 13 12 11 8 7 0
a | b | c | d | Not used | Error code
a : 1 - Under writing to memory board [write]
b : 1 - Write failure to a memory board [write]
c : Not used
d : 1 - Read failure from a memory board [Read]
* Please refer to Chapter 9 about an error code.</td><td>Set by the system</td><td>—</td></tr>
<tr><td>WRF06A</td><td>HSC count failure
Display</td><td colspan="2">The bit which corresponds if an incorrect count occurs in a counter input turns on.
15 8 7 4 3 2 1 0
Not used. | a | b | c | d
a : 1 Counter No.1 incorrect count occurred
b : 1 Counter No.2 incorrect count occurred
c : 1 Counter No.3 incorrect count occurred
d : 1 Counter No.4 incorrect count occurred</td><td>Turned on by the system</td><td>Turned off by user</td></tr>
<tr><td>WRF06F</td><td>Phase coefficient mode</td><td colspan="2">15 8 7 0
Phase coefficient mode (Ch3) | Phase coefficient mode (Ch1)
00 : Mode 1
01 : Mode 2
02 : Mode 3
03 : Mode 4</td><td rowspan="4">Turned on by user</td><td rowspan="4">Turned off by user</td></tr>
<tr><td>WRF1B0
~
WRF1B7</td><td>Output frequency,
On-preset value
(32bit operation mode)</td><td colspan="2">HSC : On-preset value (0 to 4,294,967,295)
Pulse output : Output frequency (Hz)
PWM output : Not used.</td></tr>
<tr><td>WRF1B8
~
WRF1BF</td><td>On duty,
On-preset value
(32bit operation mode)</td><td colspan="2">HSC : Off-preset value (0 to 4,294,967,295)
Pulse output : Not used.
PWM output : ON duty (%, 0 to 100)</td></tr>
<tr><td>WRF1C0
~
WRF1C7</td><td>Pre-load value,
Pulse output value
(32bit operation mode)</td><td colspan="2">HSC : Pre-load value (0 to 4,294,967,295)
Pulse output : Number of pulse (0 to 4,294,967,295)
PWM output : Not used.</td></tr>
</table>

■Reference Special internal output (word) list

No.	Name
WRF000	Self-diagnosis error code
WRF001	Syntax/Assembler error details
WRF002	I/O verify mismatch details
WRF003 ~ F00A	Undefined
WRF00B	Calendar and clock present value
WRF00C	(4 digit BCD)
WRF00D	
WRF00E	
WRF00F	
WRF010	Scan time (maximum value)
WRF011	Scan time (present value)
WRF012	Scan time (minimum value)
WRF013	CPU status
WRF014	Word internal output capacity
WRF015	Operation error code
WRF016	Division remainder register (lower)
WRF017	Division remainder register (upper)
WRF018	Undefined
WRF019	Undefined
WRF01A	Communication port 1 Setting
WRF01B	Read and set values
WRF01C	for calendar and clock
WRF01D	(4 digit BCD)
WRF01E	
WRF01F	
WRF020 ~ F03B	Undefined
WRF03C	Dedicated port 1 Modem timeout time
WRF03D	Dedicated port 2 Communication settings
WRF03E	Potentiometer input 1
WRF03F	Potentiometer input 2
WRF040 ~ F042	Occupied member registration area 1
WRF043 ~ F045	Occupied member registration area 2
WRF046 ~ F048	Occupied member registration area 3
WRF049 ~ F04B	Occupied member registration area 4
WRF04C ~ F04F	Undefined

No.	Name
WRF050	System use area
WRF051	System use area
WRF052	Undefined
WRF053	Undefined
WRF054	Power on timer
WRF055	Power on timer
WRF056	Strobe complete flag
WRF057	Detailed information of counter setting errors
WRF058	PI/O function individual setting request 1
WRF059	PI/O function individual setting request 2
WRF05A	PI/O function individual setting request 3
WRF05B	PI/O function individual setting request 4
WRF05D ~ F060	Undefined
WRF061	**Memory board write-protect setting**
WRF062	**Memory board status**
WRF063 ~ F069	Undefined
WRF06A	**HSC count failure display**
WRF06B	Pulse and PWM output auto correction setting
WRF06C	Potentiometer CH1
WRF06D	Potentiometer CH2
WRF06E	Analog input type selection
WRF06F	**Phase coefficient mode**
WRF070	I/O operation mode
WRF071	I/O detailed function settings
WRF072 ~ F075	Output frequency, On-preset value
WRF076 ~ F079	On-duty value, Off-preset value
WRF07A ~ F07D	Pre-load value, Pulse output value
WRF07E	Input edge
WRF07F	Input filtering time
WRF080 ~ F1AF	Undefined
WRF1B0 ~ F1B7	**Output frequency, On-preset value (32bit operation mode)**
WRF1B8 ~ F1BF	**On-duty, On-preset value (32bit operation mode)**
WRF1C0 ~ F1CF	**Pre-load value, Pulse output value (32bit operation mode)**

Chapter 7 Error code

The error code added by MICRO64 is shown in the following table.

Table 7.1 Additional error code details

Error Code	Error name [detection timing]	Classification	Description	RUN LED	OK LED	Operation	Related special internal output	
							Bit	Word
2B	Processor error [when power is turned on]	Serious error	The abnormalities of the processor for I/O control were detected.	◐	◐	Stops	R7CB	-
5E	Option board error [Always checking]	Warning	Unsupported option board is mounted.	-	○	Runs	R7DF	-
75	Memory board error [when power is turned on]	Warning	Data failure in memory board.	-	○	Runs	-	WRF062
76	Power failure memory area error [when power is turned on]	Warning	The area specified to be power failure memory is unfixed by the low battery.	-	◕	Runs	R7CA	-

○ : ON ● : OFF ◕ : Flashing (1 s ON, 1 s OFF) ◑ : Flashing (500 ms ON, 500 ms OFF)

◴ : Flashing (250 ms ON, 250 ms OFF)

- : Depends on the CPU's operating state. The RUN LED is lit while the CPU is in operation; the RUN LED is unlit while the CPU is not in operation.

■ Error code list

Table 7.2 Error code list (1/2)

Error Code	Error name [detection timing]	Classification	Description
11	System ROM error [when power is turned on]	Fatal error	The system ROM has a checksum error or cannot be read Error in built-in ROM/FLASH)
12	System RAM error [when power is turned on]	Fatal error	The system RAM cannot be read and/or written properly
13	Micro computer error [always checking]	Fatal error	Address error interrupt, undefined instruction interrupt occurred in the micro computer
1F	System program error [always checking]	Fatal error	System program in FLASH memory has a checksum error
23	Undefined instruction [when starting RUN]	Serious error	Error is detected when an attempt is made to execute a user program instruction that cannot be decoded (undefined instruction)
27	Data memory error [when power ON, when initializing CPU]	Serious error	Data memory cannot be read/written properly.
31	User memory error [when power is turned on, when RUN starts, during RUN]	Serious error	A checksum error is detected in user memory.
33	User memory size error [when RUN starts]	Serious error	User program capacity set by the parameter is other than 280 HEX.
34	Grammar/assemble error [when RUN starts, when changing during RUN]	Serious error	There is a grammatical error in the user program.
41	I/O information verification error [always checking]	Minor error	I/O assignment information and actual loading of module do not match
44	Overload error (normal scan) [during END processing]	Minor error	Execution time for normal scan exceeded the overload check time set by the parameter.
45	Overload error (periodical scan) [periodical processing]	Minor error	Execution time for periodical scan exceeded the execution period.
46	Overload error (interrupt scan) [during interrupt processing]	Minor error	An interrupt of the same cause occurred during interrupt scan
5F	Backup memory error [when program writing is executed, when PI/O function setting is requested]	Warning	Data cannot be written to the backup memory.

Table 7.3 Error code list (2/2)

Error Code	Error name [detection timing]	Classification	Description
61	Port 1 transmission error (parity) [when transmitting]	Warning	A parity error was detected during transmission.
62	Port 1 transmission error (framing/overrun) [when transmitting]	Warning	A framing error or overrun error was detected during transmission.
63	Port 1 transmission error (time out) [when transmitting]	Warning	A time out error was detected during transmission.
64	Port 1 transmission error (protocol error) [when transmitting]	Warning	A protocol (transmission procedure) error was detected during transmission.
65	Port 1 transmission error (BCC error) [when transmitting]	Warning	A checksum error was detected during transmission.
67	Port 2 transmission error (parity) [when transmitting]	Warning	A parity error was detected during transmission.
68	Port 2 transmission error (framing/overrun) [when transmitting]	Warning	A framing error or overrun error was detected during transmission.
69	Port 2 transmission error (time out) [when transmitting]	Warning	A time out error was detected during transmission.
6A	Port 2 transmission error (protocol error) [when transmitting]	Warning	A protocol (transmission procedure) error was detected during transmission.
6B	Port 2 transmission error (BCC error) [when transmitting]	Warning	A checksum error was detected during transmission.
71	Battery error (data memory) [always checking]	Warning	• Battery voltage dropped below the specified value • Battery not installed
94	Port 1 No modem response [when modem is connected]	Warning	There is no response with the AT command.

Chapter 8 Additional commands

One application command and 53 FUN commands have been added to MICRO64. In addition, since the counter input and number of output pulse is extended to 32-bit, the counter input control and pulse output control command is applied to 32-bit.
This chapter describes the specification of a command added / changed.

8.1 Additional command list

(1) Application command

Table 8.1 Additional command list (Application command)

No.	Ladder symbol	Command name	Process descriptions
1	ADRIO(d, s)	I/O address conversion	Stores the actual address of the I/O designated by s in d.

(2) FUN command

Table 8.2 Additional command list (FUN command) 1/2

No.	Ladder symbol	Command name	Process descriptions
1	FUN 0(s) [PIDIT(s)]	PID operation initialization	Initializes the area for PID operation.
2	FUN 1(s) [PIDOP(s)]	PID operation execution control	Performs control for PID operation execution.
3	FUN 2(s) [PIDCL(s)]	PID operation calculation	Executes PID operation.
4	FUN 4 (s) [IFR (s)]	Process stepping	Performs the process stepping processing.
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.
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.
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.
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.
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.
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.
11	FUN22 (s)	Check code calculation	Check code for sending serial communication message is calculated and created.
12	FUN23 (s)	Check code verifying	Check code for receiving serial communication message is verified.
13	FUN 30 (s) [BINDA (s)]	BIN → ASCII conversion (16 bits)	Converts 16-bit unsigned binary data to a decimal ASCII code, then stores it.
14	FUN 31 (s) [DBINDA (s)]	BIN → ASCII conversion (32 bits)	Converts 32-bit unsigned binary data to a decimal ASCII code, then stores it.
15	FUN 32 (s) [BINHA (s)]	BIN → ASCII conversion (16 bits)	Converts 16-bit unsigned binary data to an ASCII code, then stores it.
16	FUN 33 (s) [DBINHA (s)]	BIN → ASCII conversion (32 bits)	Converts 32-bit unsigned binary data to an ASCII code, then stores it.
17	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.
18	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.
19	FUN 36 (s) [DABIN (s)]	ASCII → BIN conversion (16 bits)	Converts unsigned BCD 5-digit data to an ASCII code, then stores it.
20	FUN 37 (s) [DDABIN (s)]	ASCII → BIN conversion (32 bits)	Converts signed BCD 10-digit data to an ASCII code, then stores it.
21	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.
22	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.
23	FUN 40 (s) [DABCD (s)]	ASCII → BIN conversion (16 bits)	Converts a 4-digit ASCII code to 4-digit BCD data, then stores it.
24	FUN 41 (s) [DDABCD (s)]	ASCII → BIN conversion (32 bits)	Converts a 8-digit ASCII code to 8-digit BCD data, then stores it.
25	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.
26	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.
27	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.
28	FUN 45 (s) [SCMP (s)]	Compare character strings	Compares the designated character stings (up to NULL), then stores the comparison result.
29	FUN 46 (s) [WTOB (s)]	Word → byte conversion	Divides 16-bit word data, converts it to 8-bit byte data, then stores it.
30	FUN 47 (s) [BTOW (s)]	Byte → word conversion	Divides 8-bit byte data, merges it into 16-bit word data, then stores it.
31	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).
32	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).

*[ ] indicates the display when the LADDER EDITOR is used.

Table 8.3 Additional command list (FUN command) 2/2

No.	Ladder symbol	Command name	Process descriptions
33	FUN 100(s) [INTW(s)]	Floating point operation (Real number to integer)	Real number to integer (Word) conversion.
34	FUN 101(s) [INTD(s)]	Floating point operation (Real number to integer)	Real number to integer (Double word) conversion.
35	FUN 102(s) [FLOAT(s)]	Floating point operation (Integer to real number)	Integer (word) to real number conversion.
36	FUN 103(s) [FLOATD(s)]	Floating point operation (Integer to real number)	Integer (Double word) to real number conversion.
37	FUN 104(s) [FADD(s)]	Floating point operation (Addition)	The addition of the real number.
38	FUN 105(s) [FSUB(s)]	Floating point operation (Subtraction)	The subtraction of the real number.
39	FUN 106(s) [FMUL(s)]	Floating point operation (Multiplication)	The multiplication of the real number.
40	FUN 107(s) [FDIV(s)]	Floating point operation (Division)	The division of the real number.
41	FUN 108(s) [FRAD(s)]	Floating point operation (Radian conversion)	Angle to radian conversion.
42	FUN 109(s) [FDEG(s)]	Floating point operation (Angle conversion)	Radian to angle conversion.
43	FUN 110(s) [FSIN(s)]	Floating point operation (SIN)	Calculates the SIN of the floating point number.
44	FUN 111(s) [FCOS(s)]	Floating point operation (COS)	Calculates the COS of the floating point number.
45	FUN 112(s) [FTAN(s)]	Floating point operation (TAN)	Calculates the TAN of the floating point number.
46	FUN 113(s) [FASIN(s)]	Floating point operation (ARC SIN)	Calculates the ARC SIN of the floating point number.
47	FUN 114(s) [FACOS(s)]	Floating point operation (ARC COS)	Calculates the ARC COS of the floating point number.
48	FUN 115(s) [FATAN(s)]	Floating point operation (ARC TAN)	Calculates the ARC TAN of the floating point number.
49	FUN 116(s) [FSQR(s)]	Floating point operation (Square root)	Calculates the square root of the floating point number.
50	FUN 117(s) [FEXP(s)]	Floating point operation (Exponent)	Calculates the exponent of the floating point number.
51	FUN 118(s) [FLOG(s)]	Floating point operation (Logarithm)	Calculates the logarithm of the floating point number.
52	FUN 119(s)	Floating point operation (Common logarithm)	Calculates the common logarithm of the floating point number.
53	FUN 153(s)	Pulse output with sequence parameter change	Pulse output according to the parameter beforehand registered into the table.

*[ ] indicates the display when the LADDER EDITOR is used.

: Supported by software ver. 1.01 or later

8.2 Changed command list

Table 8.4 Changed command list

No.	Ladder symbol	Command name	Process descriptions
1	FUN 143 (s)	HSC Counter value rewrite	The count value of the specified counter is rewritten.
2	FUN 144 (s)	HSC Counter value re	The present value of the specified counter is read.
3	FUN 146 (s)	HSC Preset value change	The preset value of the specified counter is changed.
4	FUN 150 (s)	Pulse frequency output setting changes	The frequency / number of output pulse of the specified counter is changed.
5	FUN 151 (s)	Pulse output with acceleration / deceleration	A pulse is outputted increasing / decreasing frequency.

: Changed by software ver. 1.01 or later

8.3 Command specifications

Please refer to the command specification from the following page about the details of a command added or changed.

Name	I/O address conversion

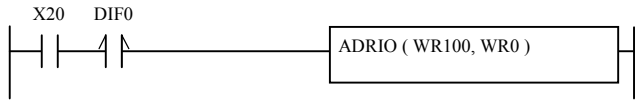
Ladder format	Condition code					Processing time (μs)		Remark
ADRIO (d, s)	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	Upper case: B Lower case: W
	DER	ERR	SD	V	C	26.5	←	
	●	●	●	●	●			
Command format	Number of steps							
ADRIO (d, s)	Condition			Steps				
	–			3				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
d	Conversion address						○	○						
s	I/O to be converted	○	○	○		○	○	○						

Function

Obtains the actual address of the I/O designated by s, and sets the result in d.

Program example




```
LD      X200
AND     DIF0
[
ADRIO ( WR100, WR0 )
]
```

Program description

Upon X00020 rise, the actual address of WR0000 (H3C00) is set in WR0100.
After command execution, WR0100 becomes H3C00.

Name	PID Initialization														
Ladder format		Condition code					Processing time (μs)		Remark						
FUN 0 (s) * [PIDIT (s)]		R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max							
		DER	ERR	SD	V	C	4,115	6,502							
		●	●	●	●	●									
Command format		Number of steps													
FUN 0 (s) * [PIDIT (s)]		Condition				Steps									
		—				3									
Usable I/O		Bit				Word				Double word			Constant	Other	
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, 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.

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.

* [ ] indicates the display when the LADDER EDITOR is used.

Name	PID operation control														
Ladder format		Condition code					Processing time (μs)		Remark						
FUN 1 (s) * [PIDOP (s)]		R7F4	R7F3	R7F2	R7F1	R7F0	Ave.	Max.							
		DER	ERR	SD	V	C	118	195							
		●	●	●	●	●									
Command format		Number of steps													
FUN 1 (s) * [PIDOP (s)]		Condition				Steps									
		—				3									
Usable I/O		Bit				Word				Double word			Constant	Other	
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, 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.

Name	PID calculation process							
Ladder format	Condition code					Processing time (μs)		Remark
FUN 2 (s) * [PIDCL (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave.	Max.	
	DER	ERR	SD	V	C	147	←	
	●	●	●	●	●			
Command format	Number of steps							
FUN 2 (s) * [PIDCL (s)]	Condition			Steps				
	—			3				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, 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.
- 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.

Cautionary notes

- Set all of the head number of WR of the word table for each PID loop of the FUN 2 (s).
- 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.

(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)]

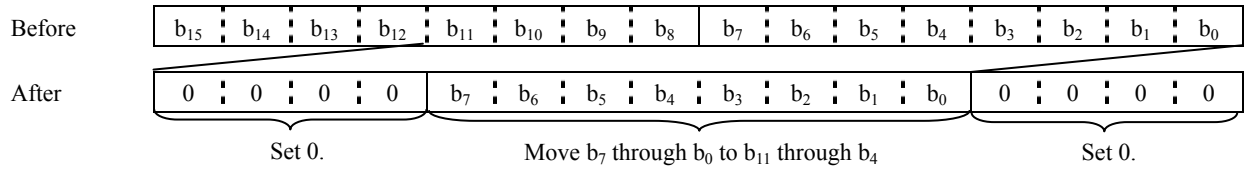
Address	Contents	Specifications	Notes	Remarks
yyyy	ADRIO (yyyy, zzzz) zzzz is the header number of the bit internal output.	Sets the header address of the bit table.	Uses 16 bits per loop. Set the actual address of the header number using the ADRIO command so the last suffix of the bit internal output 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 *3 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 *4 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]

*3 Refer to the following page (set value bit pattern) for details.

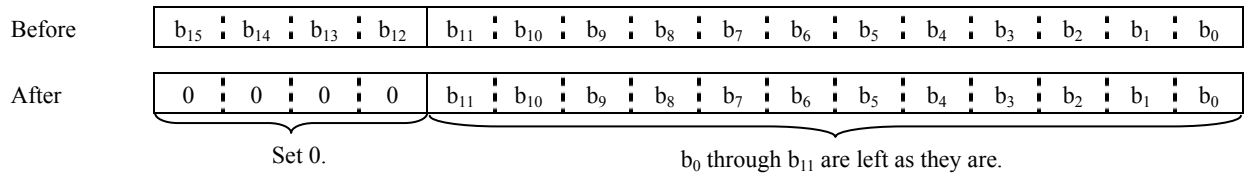
*4 Refer to the following page (output value bit pattern) for details.

■ Set value bit pattern

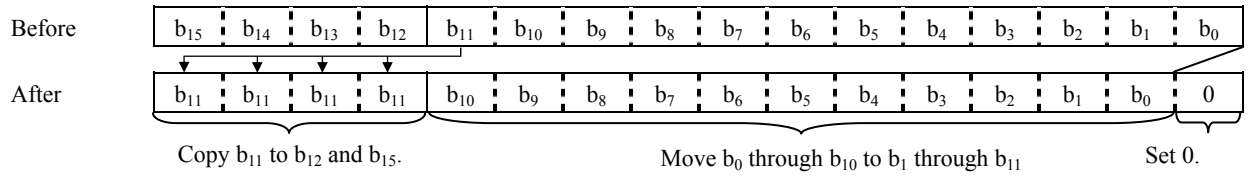
H0001 : 8-bit ➔ 16-bit



H0002 : 12-bit unsigned ➔ 16-bit



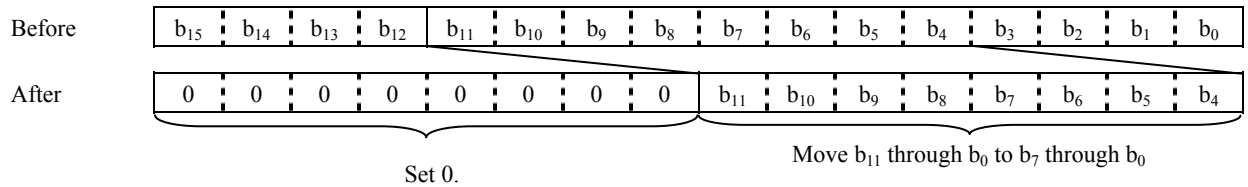
H0003 : 12-bit signed ➔ expand the sign to 16-bit



H0004 : Do not convert

■ Output value bit pattern

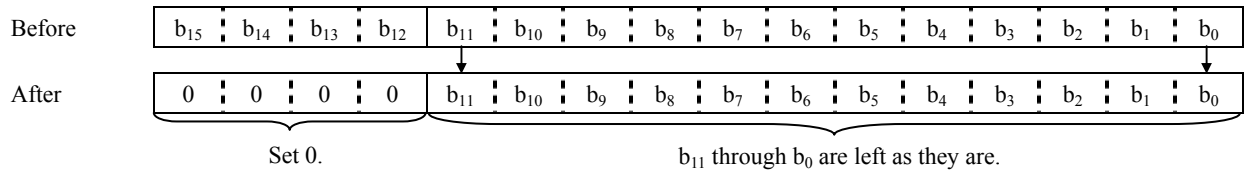
H0001 : 16-bit ➔ 8-bit



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.

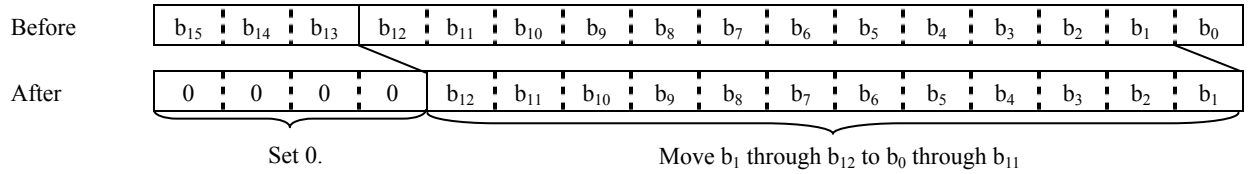
H0002 : 16-bit ➔ 12-bit



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.

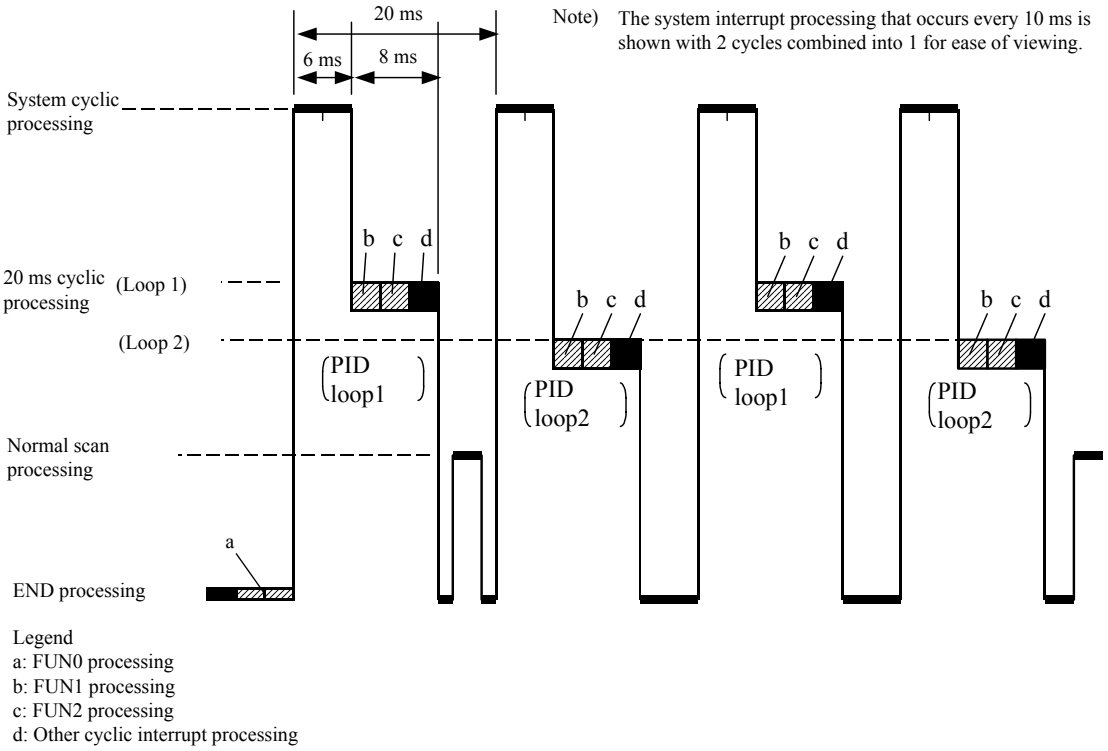
H0004 : Do not convert

(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)



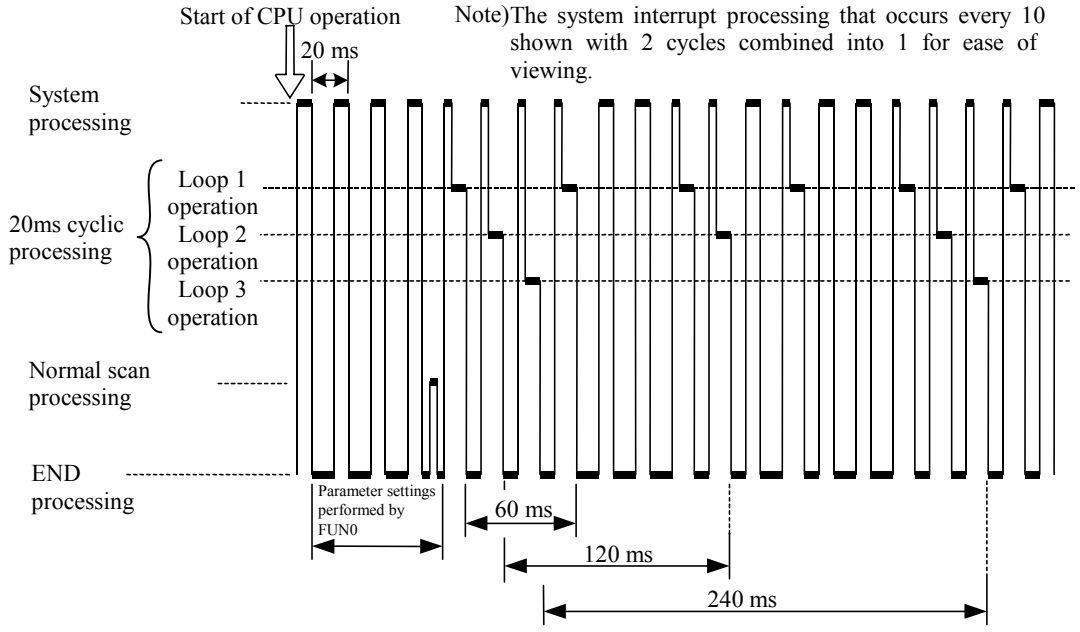
PID Operation Execution Control (2 loops)

(Example 2) Using three loops set as follows:

Loop1: TZ = 3 (× 20 ms)

Loop2: TZ = 6 (× 20 ms)

Loop3: TZ = 12 (× 20 ms)

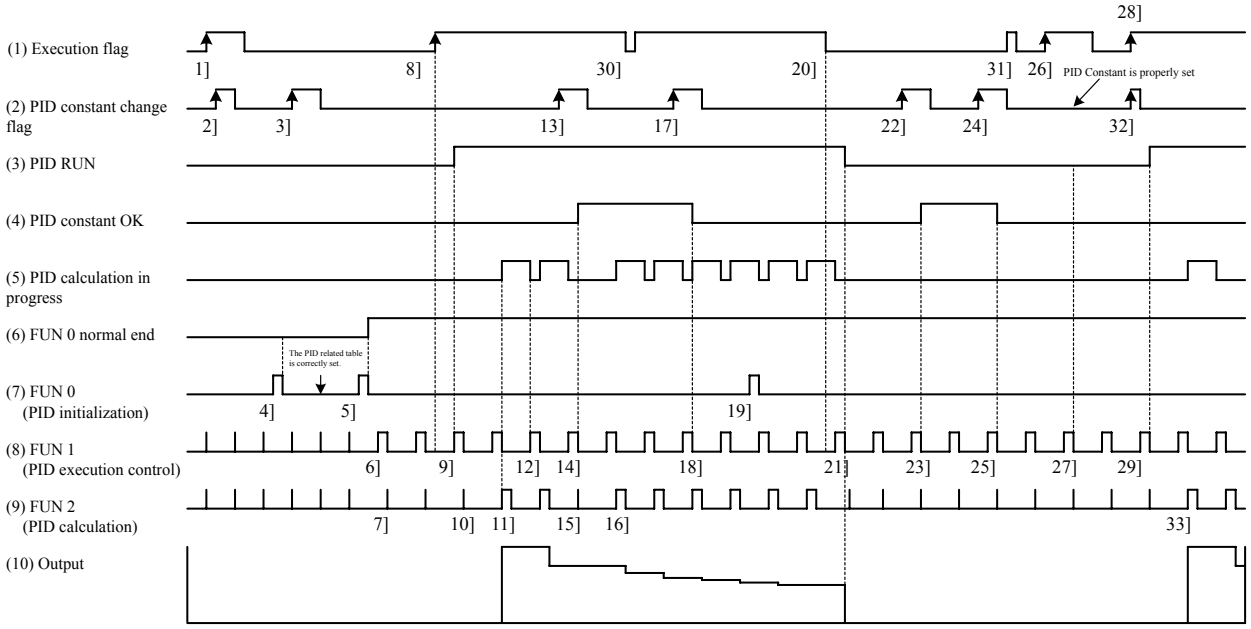


PID Operation Execution Control (3 loops)

(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.

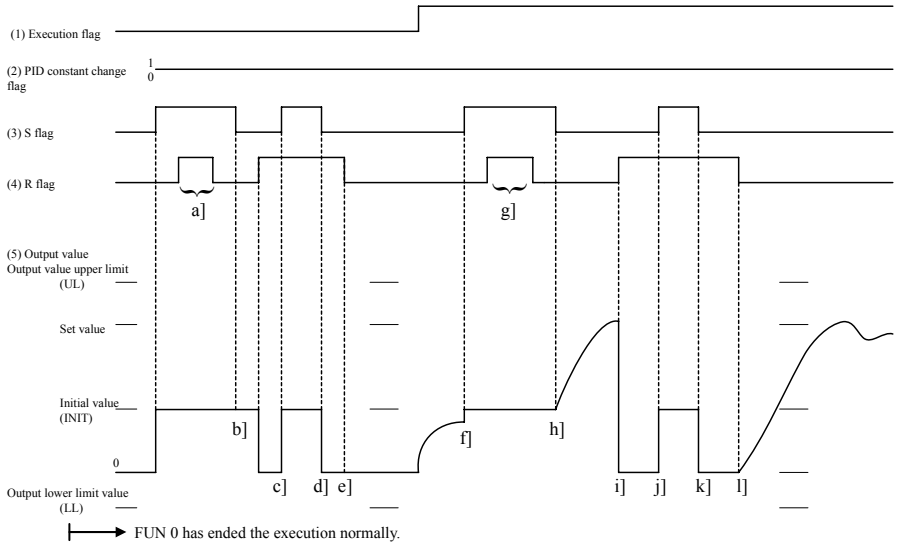
28] Since 29] FUN 1 detected turning on of both the execution flag and the 32] PID constant change flag simultaneously, turning on of the 32] PID constant change flag will be ignored. 29] FUN 1 checks the PID constant, and since it is normal, the PIDRUN flag will be turned on. PID calculation will be 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).

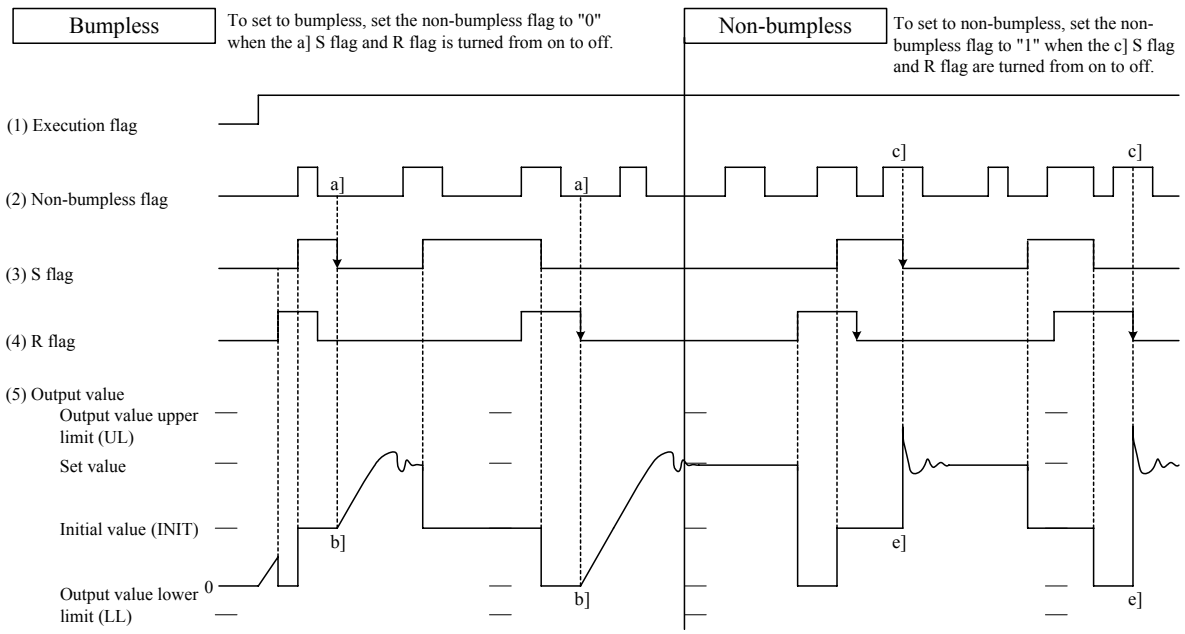
S flag…..Sets the output value to the initial value.
R flag…..Sets the output value to 0.



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

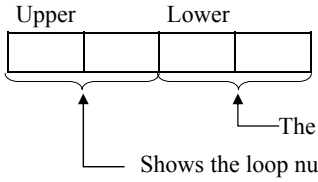


b] When the S flag and R flag turn from on to off, the output value will continuously change to move toward the set value.

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.



(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 to 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 to 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 to 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 ××0D). 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.	

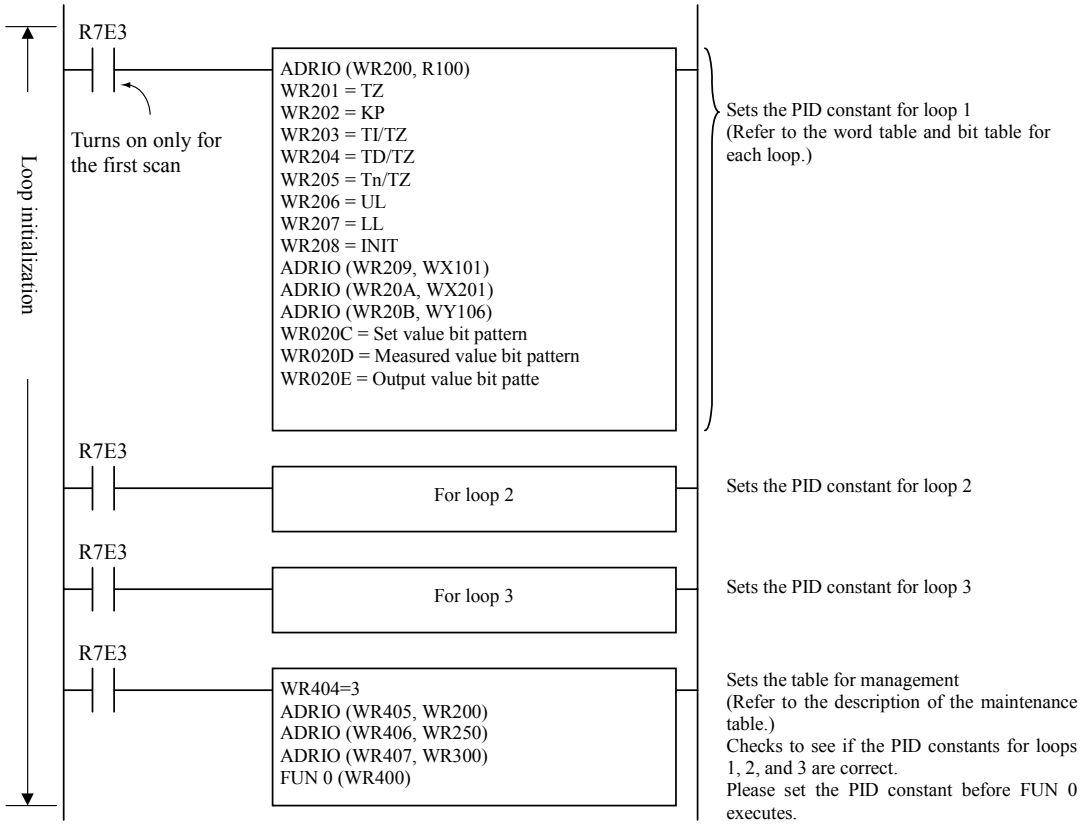
(c) Error code 2

Error code	Contents and cause	Corrective action	Remarks
0040			(Reserv)
××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 to 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 to 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 to 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.

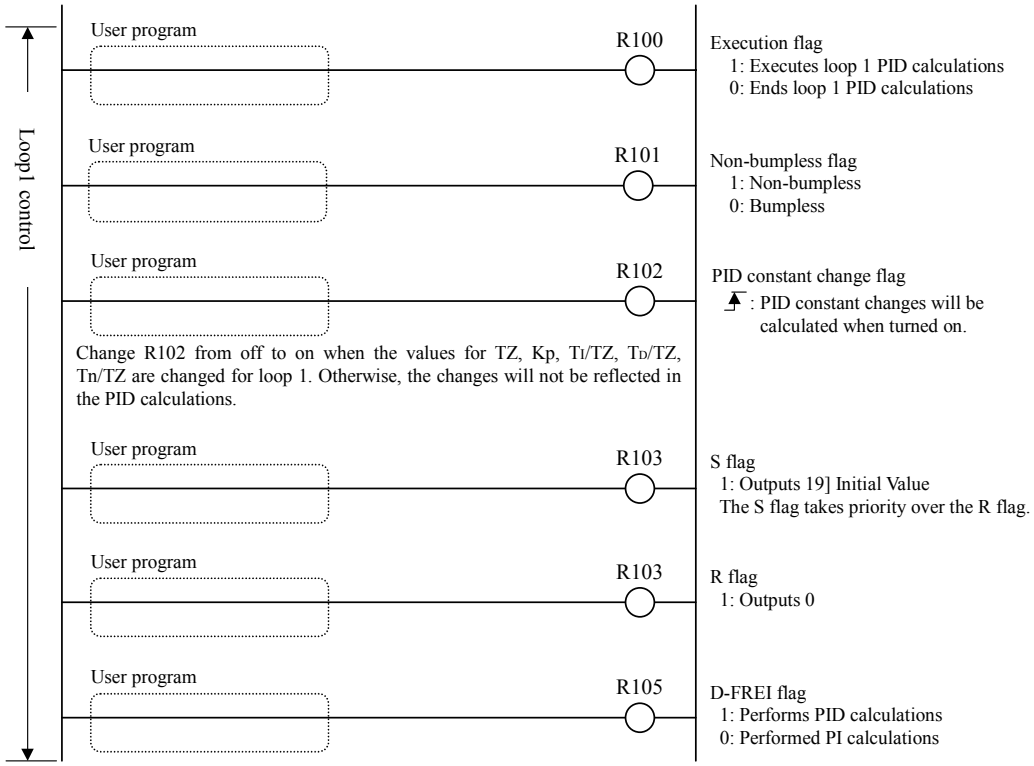
(5) Program example

This program is an example comprised of three loops. This program also rewrites the PID constant every time the CPU starts a RUN process.

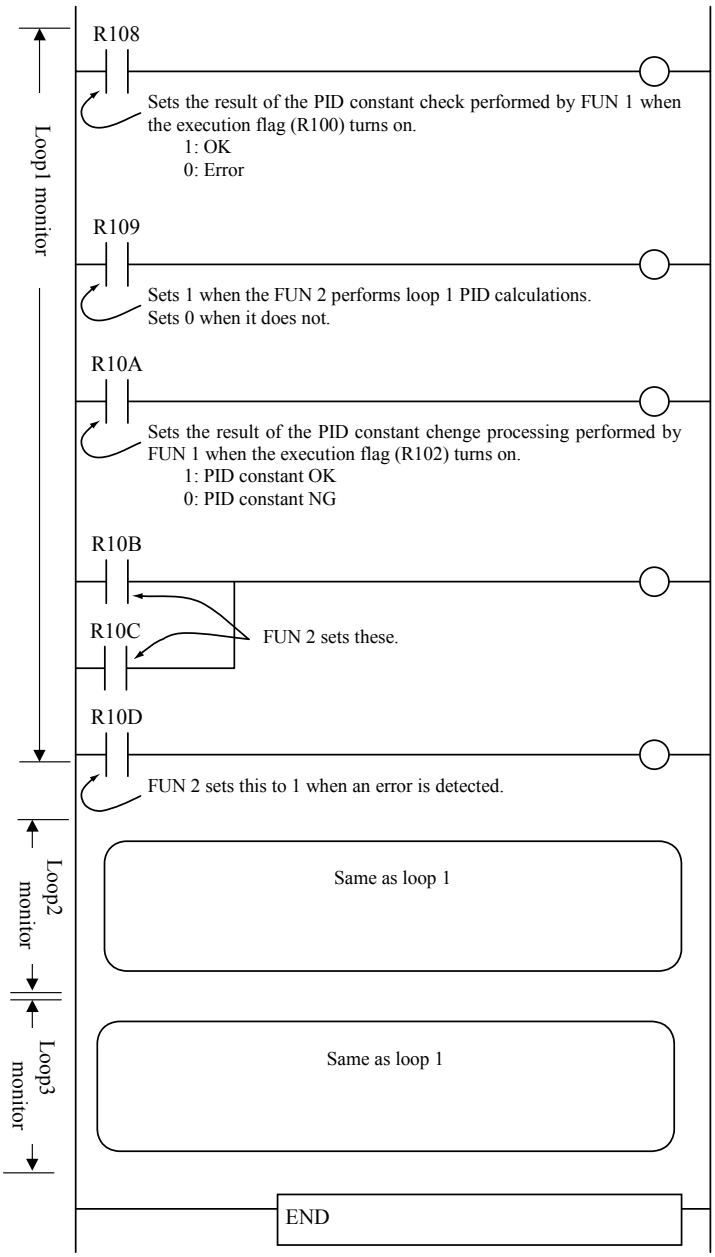
■ Loop Initialization



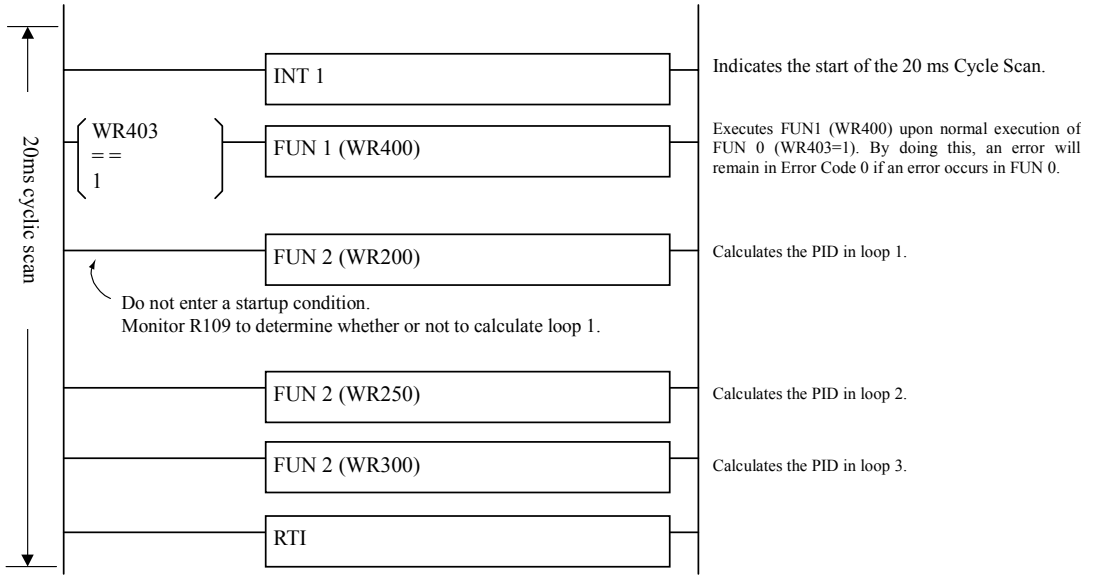
■ Loop control



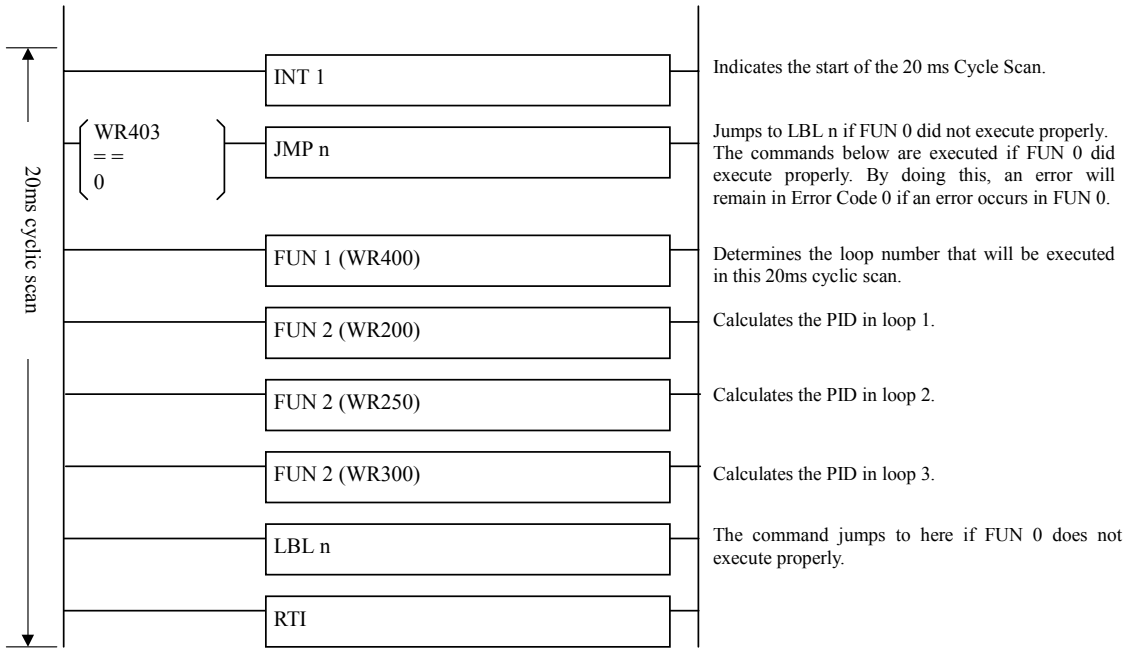
■ Loop monitor



■ 20ms cyclic scan



The program on this page can also be as shown below.



Name	Process stepping							
Ladder format	Condition code					Processing time (μs)		Remark
FUN 4 (s) * [IFR (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	602	←	
	↕	●	●	●	●			
Command format	Number of steps							
FUN 4 (s) * [IFR (s)]	Condition			Steps				
	—			3				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument							○						s uses up to s+3.

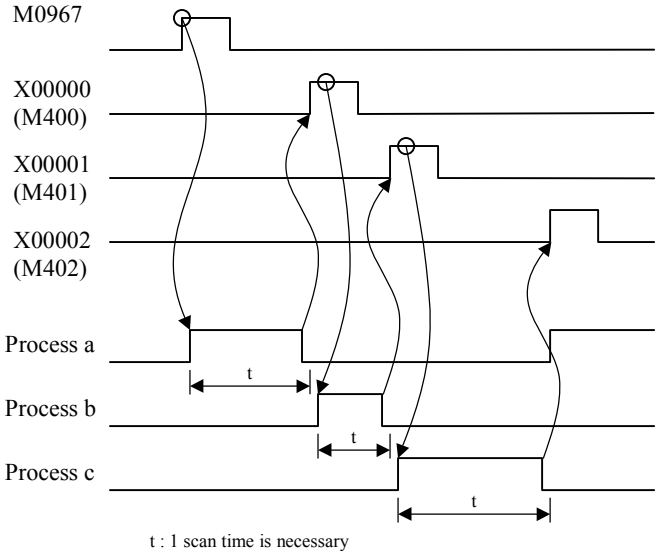
Function

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.

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.



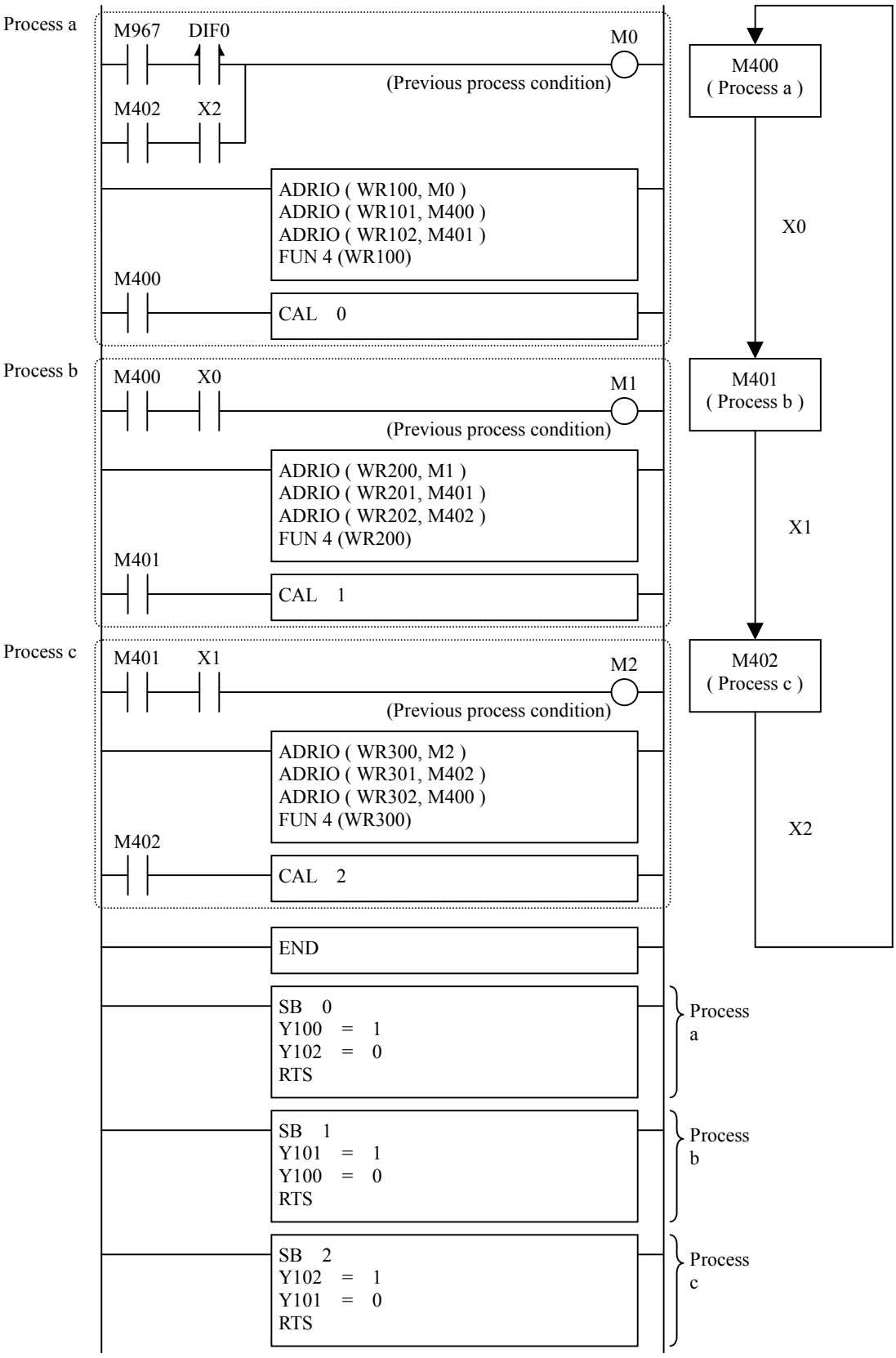
t : 1 scan time is necessary

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.

* [ ] indicates the display when the LADDER EDITOR is used.

Name	Process stepping

Program example



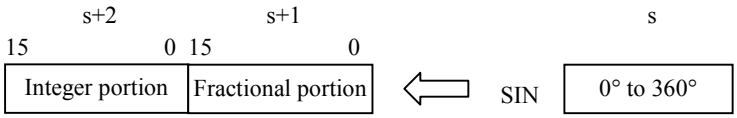
Name	SIN function

Ladder format	Condition code					Processing time (μs)		Remark
FUN 10 (s) * [SIN (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	81	←	
	↕	●	●	●	●			

Command format	Number of steps	
FUN 10 (s) * [SIN (s)]	Condition	Steps
	—	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument							○						s uses up to s+2.

Function

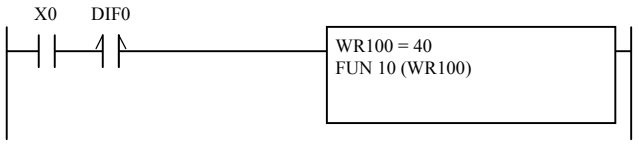


- Calculates the SIN value using the unsigned binary value designated using s as the argument, and sets the integer and fractional portions of the result in s+2 and s+1, respectively.
- The SIN value is indicated in a binary value, and negative values are indicated in two's complements.
- If the calculation is performed normally, DER is equal to "0".
- The fractional data is the value obtained by multiplying the actual value by 65,535.

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    X00000
AND DIF0
[
WR0100    = 40
FUN  10 ( WR0100 )
]
```

Program description

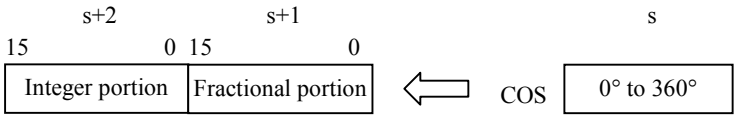
- An angle of 40° is set in WR0100.
- SIN operation is performed at the leading edge of X00100, and the fractional portion of the result is set in WR0101 and the whole number portion is set in WR0102 as binary values.
  Execution results: WR0102=H0000, WR0101=HA48E, WR0100=H0028

* [ ] indicates the display when the LADDER EDITOR is used.

Name	COS function							
Ladder format	Condition code					Processing time (μs)		Remark
FUN 11 (s) * [COS (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	84	←	
	↕	●	●	●	●			
Command format	Number of steps							
FUN 11 (s) * [COS (s)]	Condition			Steps				
	—			3				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument							○						s uses up to s+2.

Function

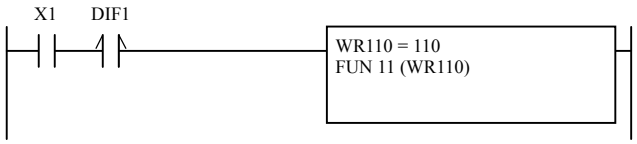


- Calculates the COS value using the unsigned binary value designated by s as the argument, and sets the integer and fractional portions of the result in s+2 and s+1, respectively.
- The COS value is indicated in a binary value, and negative values are indicated in two's complements.
- If the calculation is performed normally, DER is equal to "0".
- The fractional data is the value obtained by multiplying the actual value by 65,535.

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    X00001
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 X00001, 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

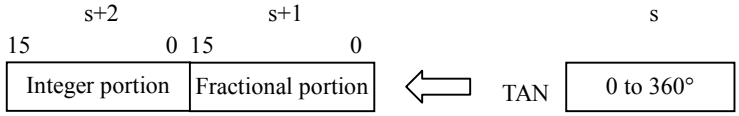
* [ ] indicates the display when the LADDER EDITOR is used.

Name	TAN function

Ladder format	Condition code					Processing time (μs)		Remark
FUN 12 (s) * [TAN (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	84	←	
	↕	●	●	●	●			
Command format	Number of steps							
FUN 12 (s) * [TAN (s)]	Condition			Steps				
	—			3				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument							○						s uses up to s+2.

Function

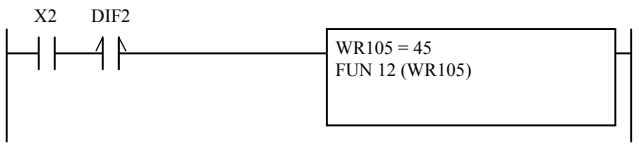


- Calculates the TAN value using the unsigned binary value designated by s as the argument, and sets the integer and fractional portions of the result in s+2 and s+1, respectively.
- The TAN value is indicated in a binary value, and negative values are indicated in two's complements.
- If the calculation is performed normally, DER is equal to "0."
- The fractional data is the value obtained by multiplying the actual value by 65,535.

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   X00002
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 X00002, 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

* [ ] indicates the display when the LADDER EDITOR is used.

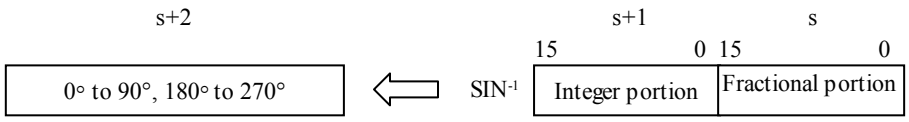
Name	ARC SIN function

Ladder format	Condition code					Processing time (μs)		Remark
FUN 13 (s) * [ASIN (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	160	←	
	↕	●	●	●	●			

Command format	Number of steps	
FUN 13 (s) * [ASIN (s)]	Condition	Steps
	—	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument (fractional portion)							○						s uses up to s+2.
s+1	Argument (integer portion)							○						

Function

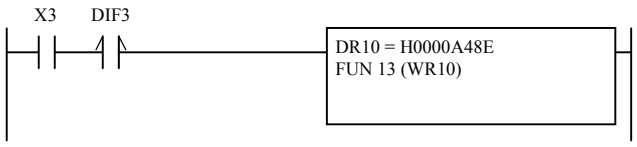


- Calculates the SIN^{-1} value using the unsigned binary value designated by s (fractional portion) and s+1 (integer portion) as the argument, and outputs s+2.
- The SIN^{-1} value is described in degrees in the range of 0° to 90° and 180° to 270°.
- If the calculation is completed normally, DER is equal to "0."
- The fractional data is the value obtained by multiplying the actual value by 65,535.

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   X00003
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 X00003, and the result is set in WR0012 as a binary value.
  Execution results: WR0012=H0028, WR0011=H0000, WR0010=HA48E

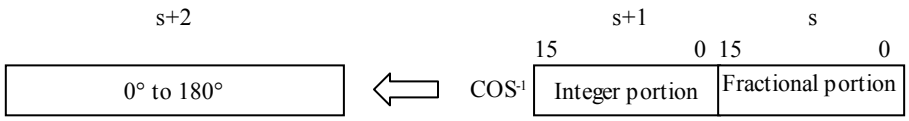
* [ ] indicates the display when the LADDER EDITOR is used.

Name	ARC COS function

Ladder format	Condition code					Processing time (μs)		Remark
FUN 14 (s) * [ACOS (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	163	←	
	↕	●	●	●	●			
Command format	Number of steps							
FUN 14 (s) * [ACOS (s)]	Condition			Steps				
	—			3				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument (fractional portion)							○						s uses up to s+2.
s+1	Argument (integer portion)							○						

Function

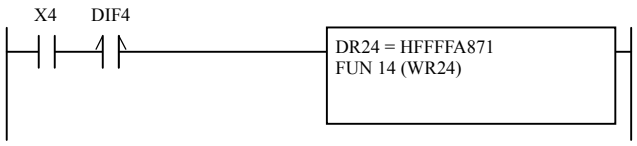


- Calculates the COS^{-1} value using the unsigned binary value designated by s (fractional portion) and s+1 (integer portion) as the argument, and outputs s+2.
- The COS^{-1} value is described in degrees in the range of 0° to 180°.
- If the calculation is completed normally, DER is equal to "0."
- The fractional data is the value obtained by multiplying the actual value by 65,535.

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    X00004
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 X00004, and the result is set in WR0026 as a binary value.
  Execution results: WR0026=H006E, WR0025=HFFFF, WR0024=HA871

* [ ] indicates the display when the LADDER EDITOR is used.

Name	ARC TAN function

Ladder format	Condition code					Processing time (μs)		Remark
FUN 15 (s) * [ATAN (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	116	←	
	↕	●	●	●	●			

Command format	Number of steps	
FUN 15 (s) * [ATAN (s)]	Condition	Steps
	—	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument (fractional portion)							○						s uses up to s+2.
s+1	Argument (integer portion)							○						

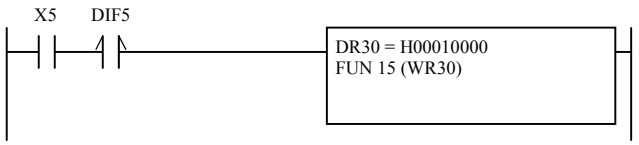
Function

- 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.

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   X00005
AND  DIF5
[
DR30   = H00010000
FUN  15 ( WR30 )
]
```

Program description

- Set data in DR0030 (WR0030, WR0031).
- TAN^{-1} operation is performs at the leading edge of X00005, and the result is set in WR0032 as a binary value.
  Execution results: WR0032=H002D, WR0031=H0001, WR0030=H0000

* [ ] indicates the display when the LADDER EDITOR is used.

Name	Check code calculation

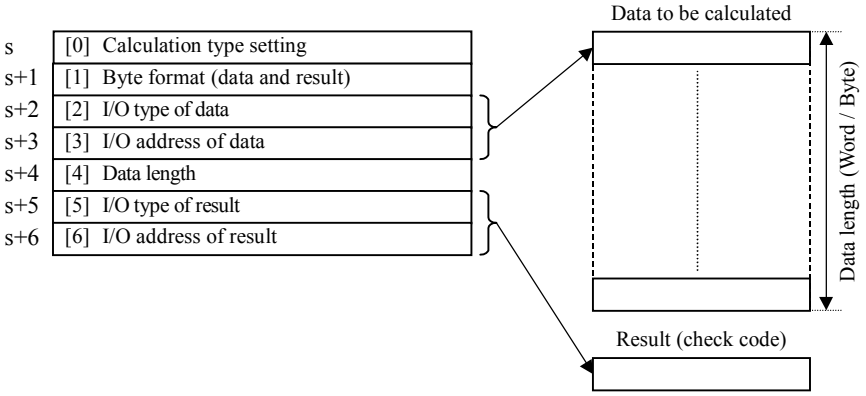
Ladder format	Condition code					Processing time (μs)	Remark
FUN 22 (s)	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	
	DER	ERR	SD	V	C	1.6 n + 458.5 (n : Data length)	
	↕	●	●	●	●		

Command format	Number of steps	
FUN 22 (s)	Condition	Steps
	—	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Starting I/O							○						s uses up 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".



[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)		

* [ ] indicates the display when the LADDER EDITOR is used.

Name	Check code calculation

Function

[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)		(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)	

Word type

H : High byte

L : Low byte

Wn :

Wn_H	Wn_L

<High byte>

Calculation starting byte

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	…

<Low byte>

Result storingposition

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

[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

Name	Check code calculation

Program example

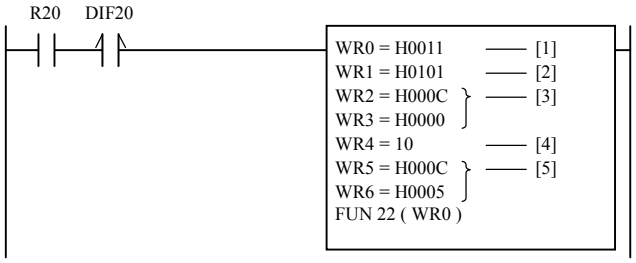
< Sent data frame > Check code = XOR for each byte and ASCII conversion

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 >




```
LD   R020
AND  DIF20
[
WR0  =  H0011
WR1  =  H0101
WR2  =  H000C
WR3  =  H0000
WR4  =  10
WR5  =  H000C
WR6  =  H0005
FUN 22 ( WR0 )
]
```

Program description

At a rising edge of R20, A check code is calculated and it stores in an internal output (WM5, WM6).

[1] Calculation type setting (Byte, ASCII, normal) : H0011

[2] Calculation starts from L-byte

Data storing from L-byte : (H0101)

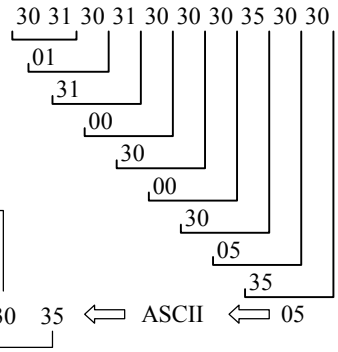
[3] Data address : WM0 (H000C, H0000)

[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



Name	Check code verifying

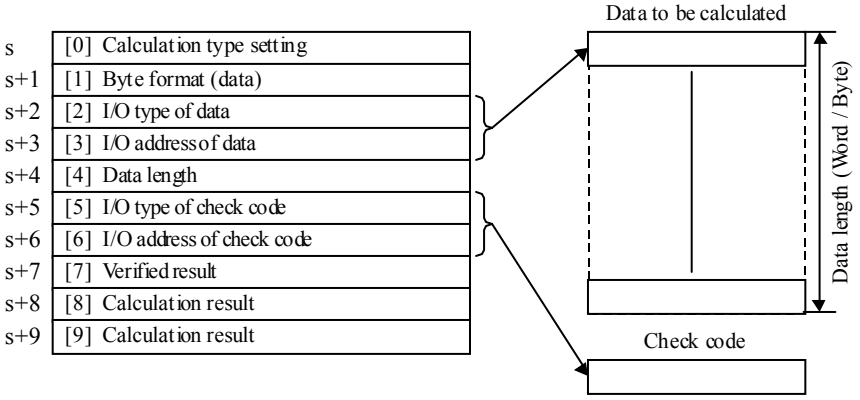
Ladder format	Condition code					Processing time (μs)	Remark
FUN 23 (s)	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	
	DER	ERR	SD	V	C	1.6 n + 474.7 (n : Data length)	
	↕	●	●	●	●		

Command format	Number of steps	
FUN 23 (s)	Condition	Steps
	—	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Starting I/O							○						s uses up 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".



[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)		

* [ ] indicates the display when the LADDER EDITOR is used.

Name	Check code verifying

Function

[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

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	…

<Low byte>

Check code starting byte

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

[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.

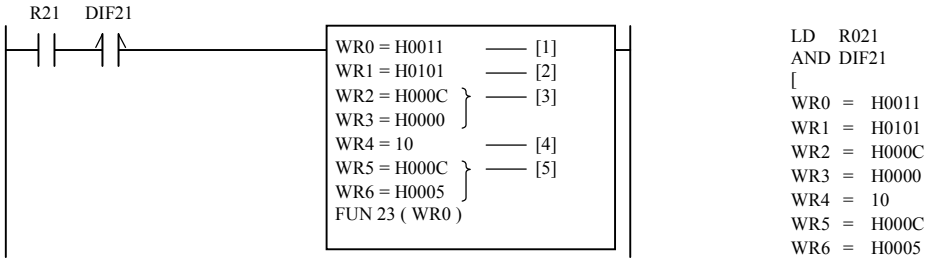
Name	Check code verifying

Program example

<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 >




```
LD   R021
AND  DIF21
[
WR0  =  H0011
WR1  =  H0101
WR2  =  H000C
WR3  =  H0000
WR4  =  10
WR5  =  H000C
WR6  =  H0005
FUN 23 ( WR0 )
]
```

Program description

At a rising edge of R21, A check code is calculated and it value is compared with verify data.
A result is stored in s+7.

[1] Calculation type setting (Byte, ASCII, normal) :H0003

[2] Verification starts from L-byte

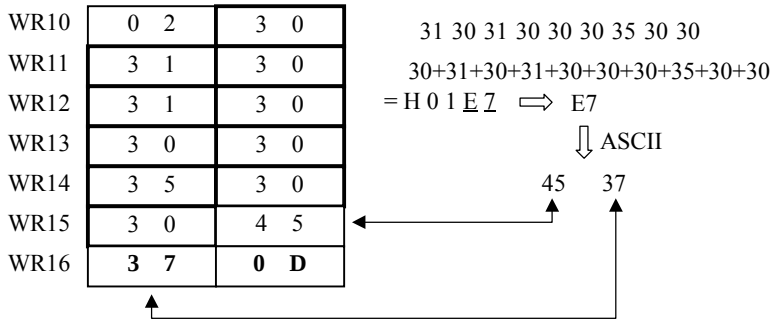
Check code starts from L-byte : H0101

[3] Data address: WR100 (H000A, H0100)

[4] Data length : 10 bytes

[5] Check code address : WR105 (H000A, H00105)

<Result>



Verifying OK (WM7=H8000) as right value. (WM7=H80FF in case of wrong value)

Name	Conversion from 16-bit unsigned binary to decimal ASCII data (BINARY TO DECIMAL ASCII)

Ladder format	Condition code					Processing time (μs)		Remark
FUN 30 (s) * [BINDA (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	309	←	
	↕	●	●	●	●			

Command format	Number of steps	
FUN 30 (s) * [BINDA (s)]	Condition	Steps
	—	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument (conversion data)							○						s uses up to s+3.

Function

16-bit unsigned binary data

s: 0 to 65535

Decimal ASCII data

	15 — 8	7 — 0
s+1	10^4	10^3
s+2	10^2	10^1
s+3	10^0	NULL

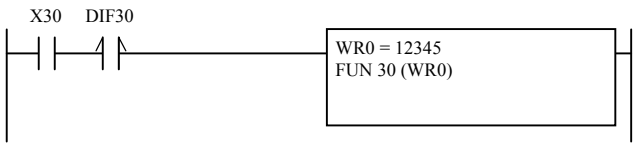
10^n: ASCII code in the 10^n place

- 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."

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   X00030
AND  DIF30
[
WR0   = 12345
FUN  30 ( WR0 )
]
```

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

* [ ] indicates the display when the LADDER EDITOR is used.

Name	Conversion from 32-bit signed binary to decimal ASCII data (DOUBLE BINARY TO DECIMAL ASCII)

Ladder format	Condition code					Processing time (μs)		Remark
FUN 31 (s) * [DBINDA (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	471	←	
	↕	●	●	●	●			
Command format	Number of steps							
FUN 31 (s) * [DBINDA (s)]	Condition			Steps				
	—			3				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument (lower)							○						-2,147,483,648 to
s+1	Argument (higher)							○						2,147,483,647

Function

32-bit signed binary data

s	Lower 16-bit
s+1	Higher 16-bit

Decimal ASCII data

	15 — 8	7 — 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

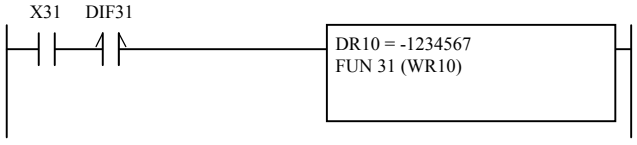
Sign Plus : H20 (space)
Minus: H2D ("-")
10^n: ASCII code in the 10^n place

- 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."

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   X00031
AND  DIF31
[
DR10   = -1234567
FUN  31 ( WR10 )
]
```

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

* [ ] indicates the display when the LADDER EDITOR is used.

Name	Conversion from 16-bit binary to hexadecimal ASCII data (BINARY TO HEXA ASCII)

Ladder format	Condition code					Processing time (μs)		Remark
FUN 32 (s) * [BINHA (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	311	←	
	↕	●	●	●	●			
Command format	Number of steps							
FUN 32 (s) * [BINHA (s)]	Condition			Steps				
	—			3				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument (conversion data)							○						s uses up to s+3

Function

16-bit unsigned binary data

s | 0 to HFFFF ⇨

Hexadecimal ASCII data

	15 — 8	7 — 0
s+1	16^3	16^2
s+2	16^1	16^0
s+3	NULL	

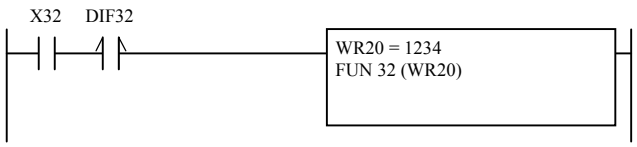
16^n: ASCII code in the 16^n place

- 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.”

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   X00032
AND  DIF2
[
WR0   = 1234
FUN  32 ( WR20 )
]
```

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

* [ ] indicates the display when the LADDER EDITOR is used.

Name	Conversion from 32-bit binary to hexadecimal ASCII data (DOUBLE BINARY TO HEXA ASCII)

Ladder format	Condition code					Processing time (μs)		Remark
FUN 33 (s) * [DBINHA (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	377	←	
	↕	●	●	●	●			

Command format	Number of steps	
FUN 33 (s) * [DBINHA (s)]	Condition	Steps
	—	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, 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	

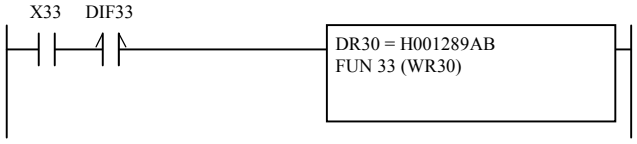
16^n: ASCII code in the 16^n place

- 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."

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    X00033
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

* [ ] indicates the display when the LADDER EDITOR is used.

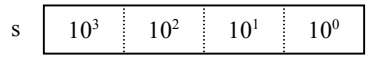
Name	Conversion from 16-bit BCD to decimal ASCII data (BCD TO DECIMAL ASCII)

Ladder format	Condition code					Processing time (μs)		Remark
FUN 34 (s) * [BCDDA (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	267	←	
	↕	●	●	●	●			
Command format	Number of steps							
FUN 34 (s) * [BCDDA (s)]	Condition			Steps				
	—			3				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument (conversion data)							○						s uses up to s+3

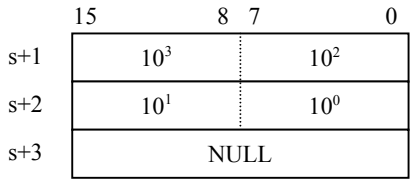
Function

16-bit BCD data



10^n: BCD code in the 10^n place

Decimal ASCII data



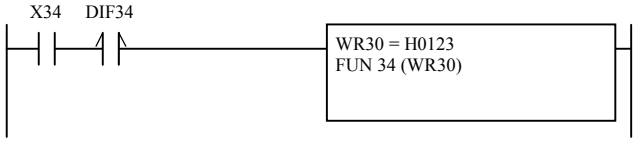
10^m: ASCII code in the 10^m place

- 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."

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   X00034
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

* [ ] indicates the display when the LADDER EDITOR is used.

Name	Conversion from 32-bit BCD to decimal ASCII data (DOUBLE BCD TO DECIMAL ASCII)

Ladder format	Condition code					Processing time (μs)		Remark
FUN 35 (s) * [DBCDDA (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	385	←	
	↕	●	●	●	●			
Command format	Number of steps							
FUN 35 (s) * [DBCDDA (s)]	Condition			Steps				
	—			3				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument (lower)							○						s is BCD data.
s+1	Argument (higher)							○						s uses up to s+6

Function

32-bit BCD data

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	

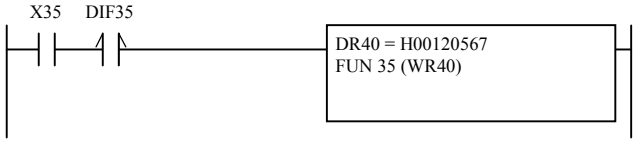
10^m: ASCII code in the 10^m place

- 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."

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   X00035
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

* [ ] indicates the display when the LADDER EDITOR is used.

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
FUN 36 (s) * [DABIN (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	185	←	
	↕	●	●	●	●			

Command format	Number of steps	
FUN 36 (s) * [DABIN (s)]	Condition	Steps
	—	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, 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 → 16-bit binary data

	15 — 8	7 — 0
s	10^4	10^3
s+1	10^2	10^1
s+2	10^0	H00

⇒ s+3: 0 to 65,535

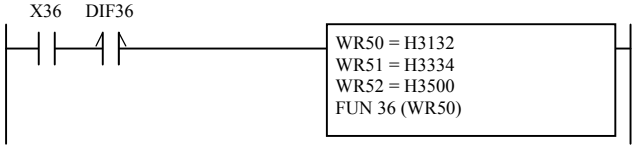
10^n: ASCII code in the 10^n place

- 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."

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   X00036
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)

* [ ] indicates the display when the LADDER EDITOR is used.

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
FUN 37 (s) * [DDABIN (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	249	←	
	↕	●	●	●	●			
Command format	Number of steps							
FUN 37 (s) * [DDABIN (S)]	Condition			Steps				
	—			3				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument (ASCII)							○						Sign is H20 or H2D, and other digits are combinations of H00, H20, and H30 to H39. s uses up to s + 7
~	~							~						
s+2	Argument (ASCII)							○						

Function

Signed decimal ASCII data

	15 — 8	7 — 0
s	Sign	10^9
s+1	10^8	10^9
s+2	10^6	10^5
s+3	10^4	10^3
s+4	10^2	10^1
s+5	10^0	H00

Sign Plus : H20(space)
Minus : H2D("-")
10^n: ASCII code in the 10^n place

32-bit signed binary data

s+6	Lower 16-bit
s+7	Higher 16-bit

- 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.

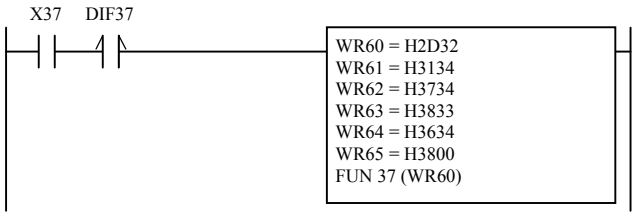
Cautionary notes

- 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.

* [ ] indicates the display when the LADDER EDITOR is used.

Name	Conversion from 10-digit signed decimal ASCII to 32-bit binary data (DOUBLE DECIMAL ASCII TO BINARY)

Program example




```
LD    X00037
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)

Name	Conversion from 4-digit hexadecimal ASCII to 16-bit binary data (HEXA ASCII TO BINARY)

Ladder format	Condition code					Processing time (μs)		Remark
FUN 38 (s) * [HABIN (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	154	←	
	↕	●	●	●	●			

Command format	Number of steps	
FUN 38 (s) * [HABIN (s)]	Condition	Steps
	—	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument (higher ASCII)							○						Combination of H00, H20, H30 to H39 and H41 to 46 s uses up to s + 2
s+2	Argument (lower ASCII)							○						

Function

Hexadecimal ASCII data

	15 — 8	7 — 0
s	16^3	16^2
s+1	16^1	16^0

⇨ s+2: 0 to HFFFF (16-bit binary data)

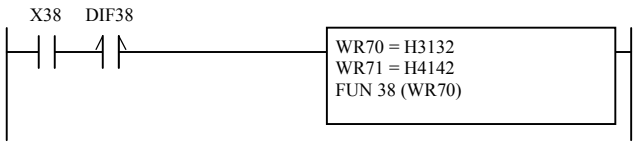
16^n: ASCII code in the 16^n place

- 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."

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   X00038
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

* [ ] indicates the display when the LADDER EDITOR is used.

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
FUN 39 (s) * [DHABIN (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	230	←	
	↕	●	●	●	●			
Command format	Number of steps							
FUN 39 (s) * [DHABIN (s)]	Condition			Steps				
	—			3				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument (ASCII data)							○						Combination of H00, H20, H30 to H39 and H41 to 45 s uses up to s + 5
~	~							~						
s+5	Argument (ASCII data)							○						

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

⇨

32-bit binary data

s+4	Lower 16-bit
s+5	Higher 16-bit

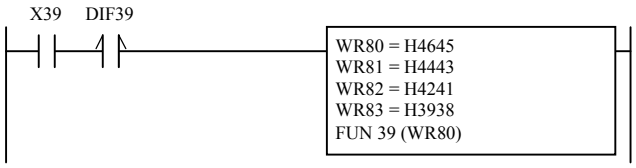
16^n: ASCII code in the 16^n place

- 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."

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   X00039
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

* [ ] indicates the display when the LADDER EDITOR is used.

Name	Conversion from 4-digit decimal ASCII to 16-bit BCD data (DECIMAL ASCII TO BCD)

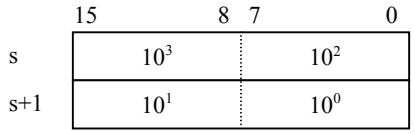
Ladder format	Condition code					Processing time (μs)		Remark
FUN 40 (s) * [DABCD (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	154	←	
	↕	●	●	●	●			

Command format	Number of steps	
FUN 40 (s) * [DABCD (s)]	Condition	Steps
	—	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument (ASCII data)							○						Combination of H00, H20 and H30 toH39 s uses up to s + 2
s+1	Argument (ASCII data)							○						

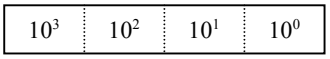
Function

Decimal ASCII data



10^m: ASCII code in the 10^m place

16-bit unsigned BCD data



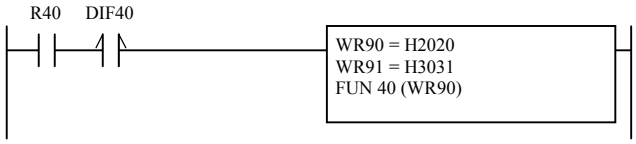
10^n: BCD code in the 10^n place

- 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" .

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   R0040
AND  DIF40
[
WR90 = H2020
WR91 = H3031
FUN 40 (WR90)
]
```

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

* [ ] indicates the display when the LADDER EDITOR is used.

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
FUN 41 (s) * [DDABCD (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	232	←	
	↕	●	●	●	●			
Command format	Number of steps							
FUN 41 (s) * [DDABCD (s)]	Condition			Steps				
	—			3				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument (ASCII data)							○						Combination of H00, H20 and H30 to H39 s uses up to s + 5
~	~							~						
s+3	Argument (ASCII data)							○						

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

10^m: ASCII code in the 10^m place

32-bit BCD data

s+4	10^3	10^2	10^1	10^0
s+5	10^7	10^6	10^5	10^4

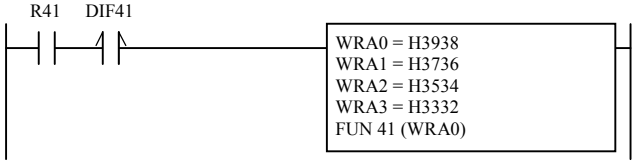
10^n: BCD code in the 10^n place

- 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."

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   R0041
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

* [ ] indicates the display when the LADDER EDITOR is used.

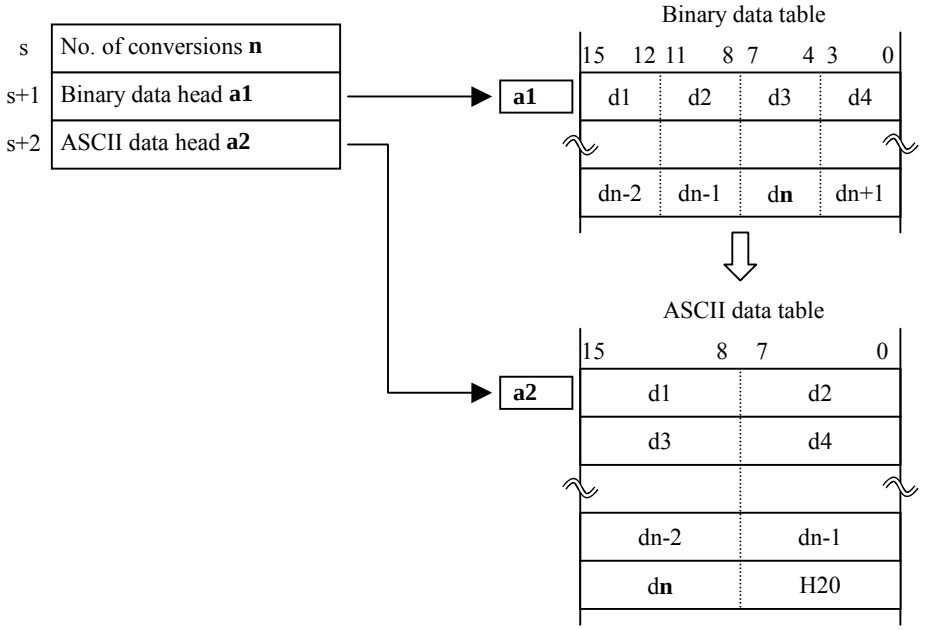
Name	Conversion from hexadecimal binary to hexadecimal ASCII data (BINARY TO ASCII)

Ladder format	Condition code					Processing time (μs)	Remark
FUN 42 (s) * [ASC (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	
	DER	ERR	SD	V	C	5.8 n + 273.9 (n : Number of conversion)	
	↕	●	●	●	●		

Command format	Number of steps	
FUN 42 (s) * [ASC (s)]	Condition	Steps
	—	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, 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."

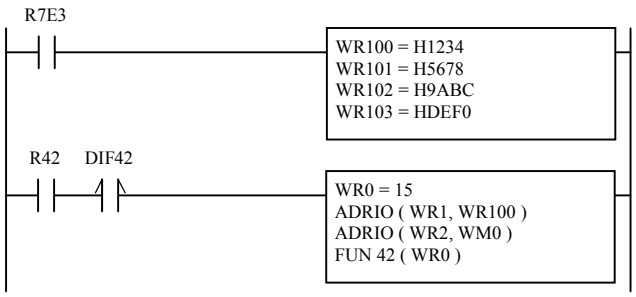
Cautionary notes

- 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.

* [ ] indicates the display when the LADDER EDITOR is used.

Name	Conversion from hexadecimal binary to hexadecimal ASCII data (BINARY TO ASCII)

Program example




```
LD    R7E3
[
WR0100 = H1234
WR0101 = H5678
WR0102 = H9ABC
WR0103 = HDEF0
]
LD    R0042
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 X00042, 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.)

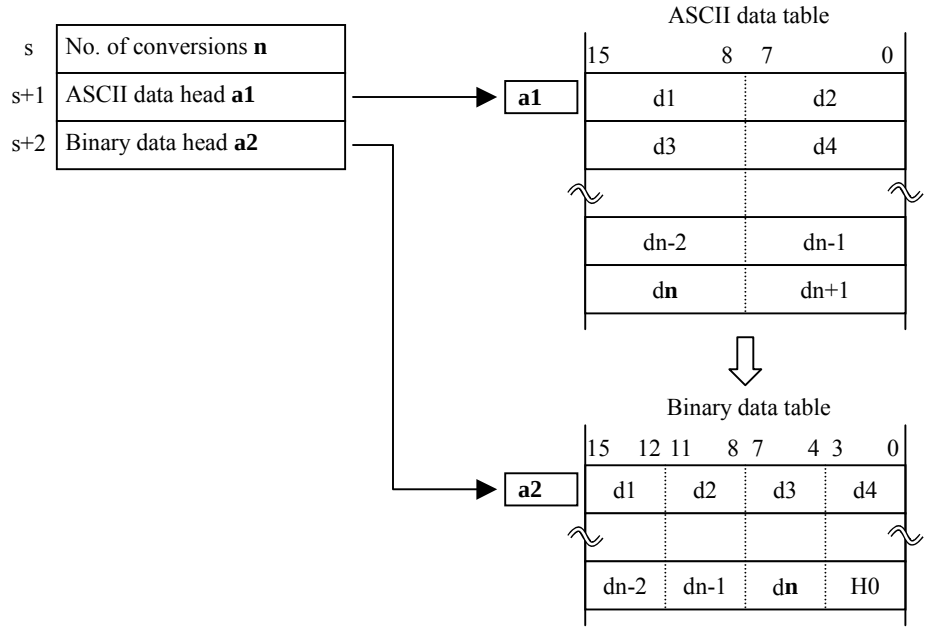
Name	Conversion from hexadecimal ASCII to hexadecimal binary data (ASCII TO BINARY)

Ladder format	Condition code					Processing time (μs)	Remark
FUN 43 (s) * [HEX (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	
	DER	ERR	SD	V	C	21.1 n + 271.8 (n : Number of conversion)	
	↕	●	●	●	●		

Command format	Number of steps	
FUN 43 (s) * [HEX (s)]	Condition	Steps
	—	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, 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."

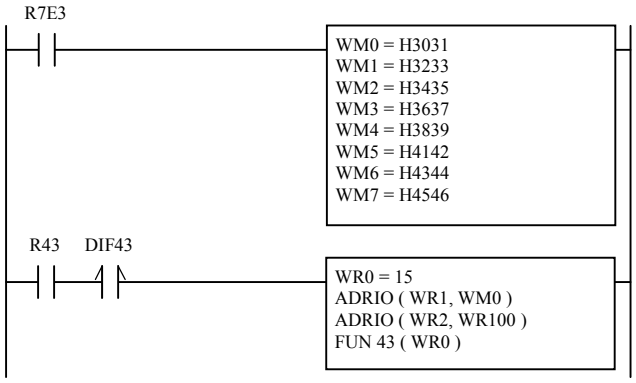
Cautionary notes

- 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.

* [ ] indicates the display when the LADDER EDITOR is used.

Name	Conversion from hexadecimal ASCII to hexadecimal binary data (ASCII TO BINARY)

Program example




```
LD    R7E3
[
WM000 = H3031
WM001 = H3233
WM002 = H3435
WM003 = H3637
WM004 = H3839
WM005 = H4142
WM006 = H4344
WM007 = H4546
]
LD    R0043
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 X00043, 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

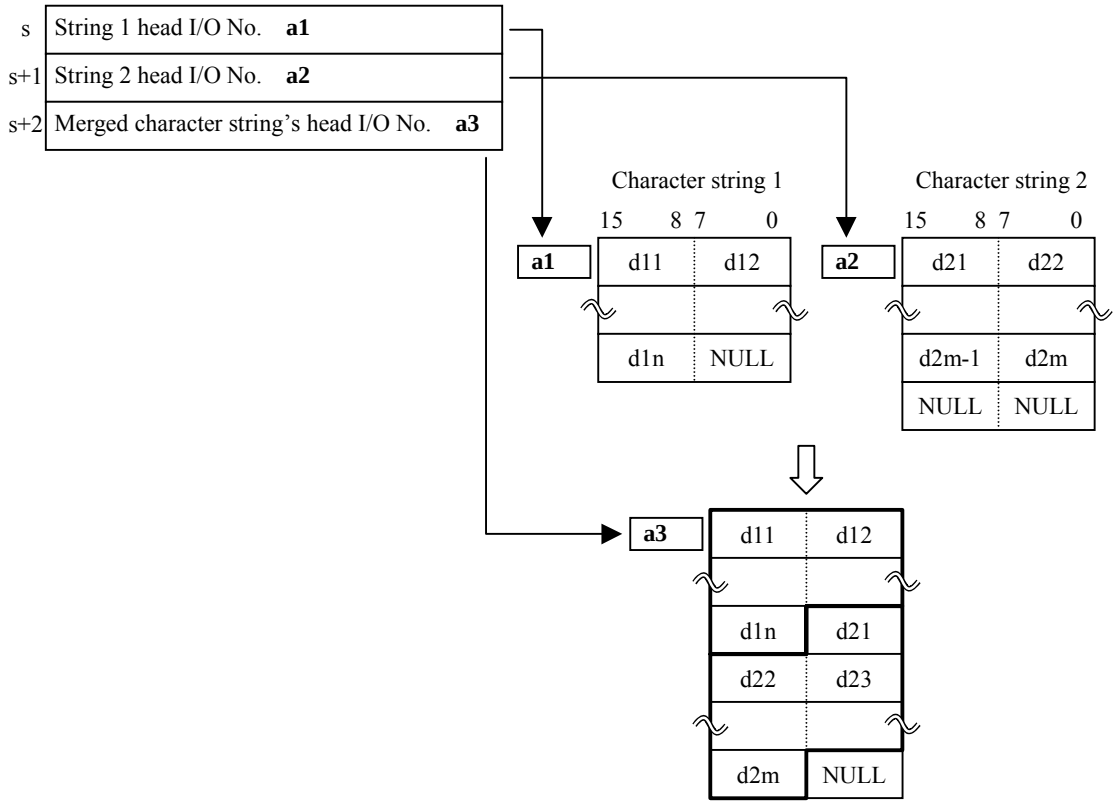
Name	Merge strings

Ladder format	Condition code					Processing time (μs)	Remark
FUN 44 (s) * [SADD (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	
	DER	ERR	SD	V	C	18.0 n + 401.9 (n : Number of merge bytes)	
	↕	●	●	●	●		

Command format	Number of steps	
FUN 44 (s) * [SADD (s)]	Condition	Steps
	—	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
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."

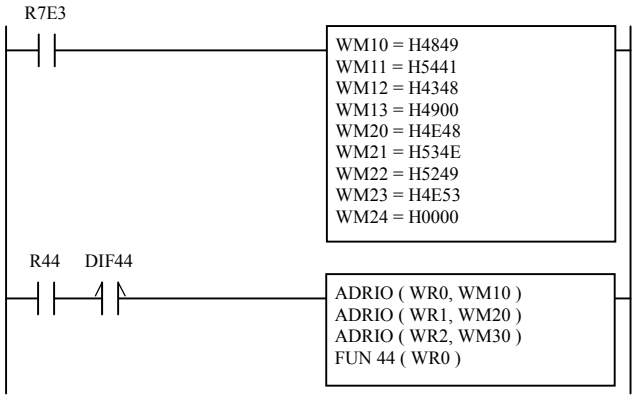
* [ ] indicates the display when the LADDER EDITOR is used.

Name	Merge strings

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
WM011 = H5441
WM012 = H4348
WM013 = H4900
WM020 = H4E48
WM021 = H534E
WM022 = H5249
WM023 = H4E53
WM024 = H0000
]
LD    R044
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 R044, 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

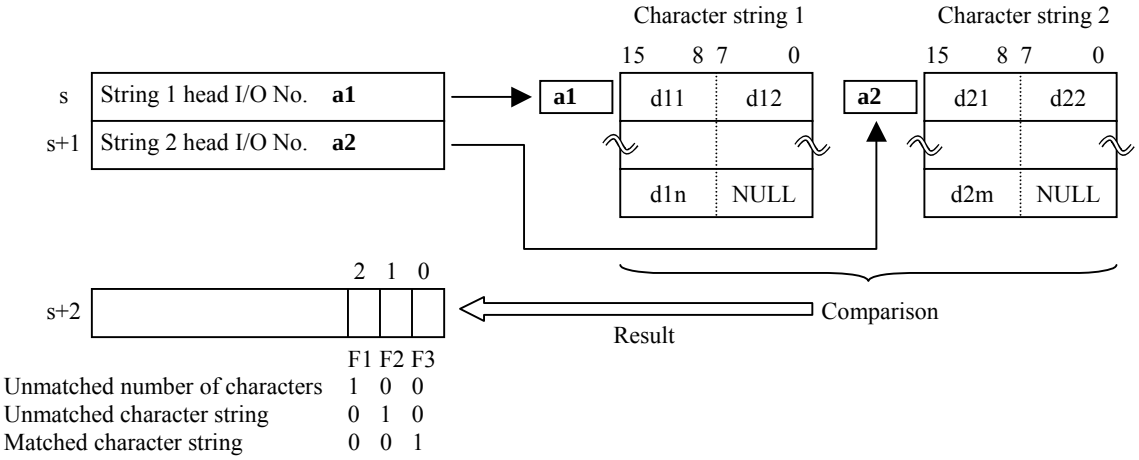
Name	Compare character strings

Ladder format	Condition code					Processing time (μs)	Remark
FUN 45 (s) * [SCMP (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	
	DER	ERR	SD	V	C	12.7 n + 324.5 (n : Number of compare bytes)	
	↕	●	●	●	●		

Command format	Number of steps	
FUN 45 (s) * [SCMP (s)]	Condition	Steps
	—	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, 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



- 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."

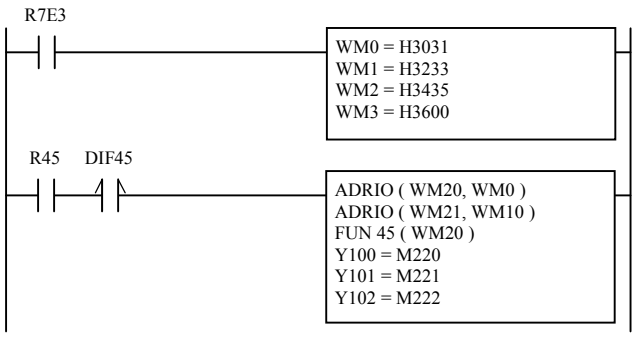
Cautionary notes

- 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.

* [ ] indicates the display when the LADDER EDITOR is used.

Name	Compare character strings

Program example




```
LD     R7E3
[
WM000 = H3031
WM001 = H3233
WM002 = H3435
WM003 = H3600
]
LD     R0045
AND DIF45
[
ADRIO ( WM020, WM000 )
ADRIO ( WM021, WM010 )
FUN 45 ( WM020 )
Y00100 = M0220
Y00101 = M0221
Y00102 = 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.

Name	Conversion from word units to byte units (CONVERSION WORDS TO BYTES)

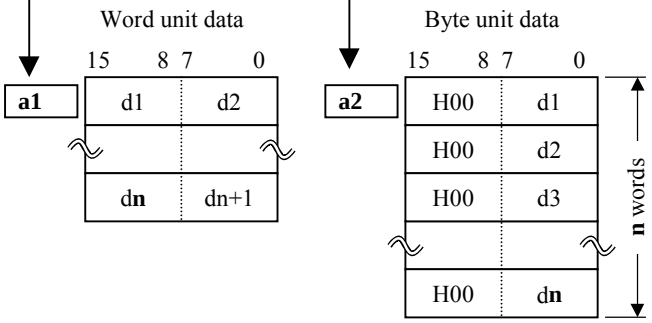
Ladder format	Condition code					Processing time (μs)	Remark
FUN 46 (s) * [WTOB (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	
	DER	ERR	SD	V	C	4.6 n + 248.6 (n : Number of converted bytes)	
	↕	●	●	●	●		

Command format	Number of steps	
FUN 46 (s) * [WTOB (s)]	Condition	Steps
	—	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, 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

s	Word-unit data head I/O No. **a1**
s+1	Converted byte-unit data head I/O No. **a2**
s+2	No. of converted bytes **n**



- 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."

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.

* [ ] indicates the display when the LADDER EDITOR is used.

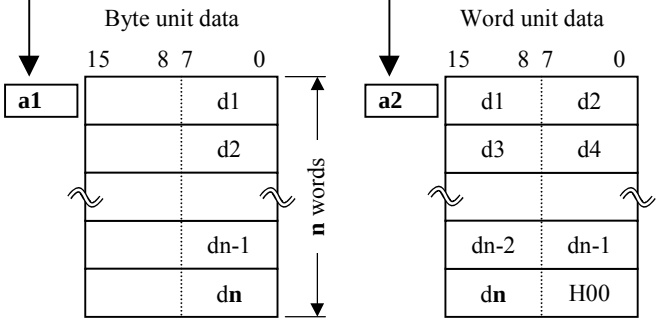
Name	Conversion from byte units to word units (CONVERSION BYTES TO WORDS)

Ladder format	Condition code					Processing time (μs)	Remark
FUN 47 (s) * [BTOW (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	
	DER	ERR	SD	V	C	3.5 n + 252.5 (n : Number of converted bytes)	
	↕	●	●	●	●		
Command format	Number of steps						
FUN 47 (s) * [BTOW (s)]	Condition			Steps			
	–			3			

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Byte-unit data head I/O No.							○						Actual addresses are set in s and s + 1 s uses up to s + 2
s+1	Word-unit data head I/O No.							○						
s+2	No. of converted bytes							○						

Function

s	Byte-unit data head I/O No. **a1**
s+1	Converted word-unit data head I/O No. **a2**
s+2	No. of converted bytes **n**



- 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.

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.

* [ ] indicates the display when the LADDER EDITOR is used.

Name	Byte right shift

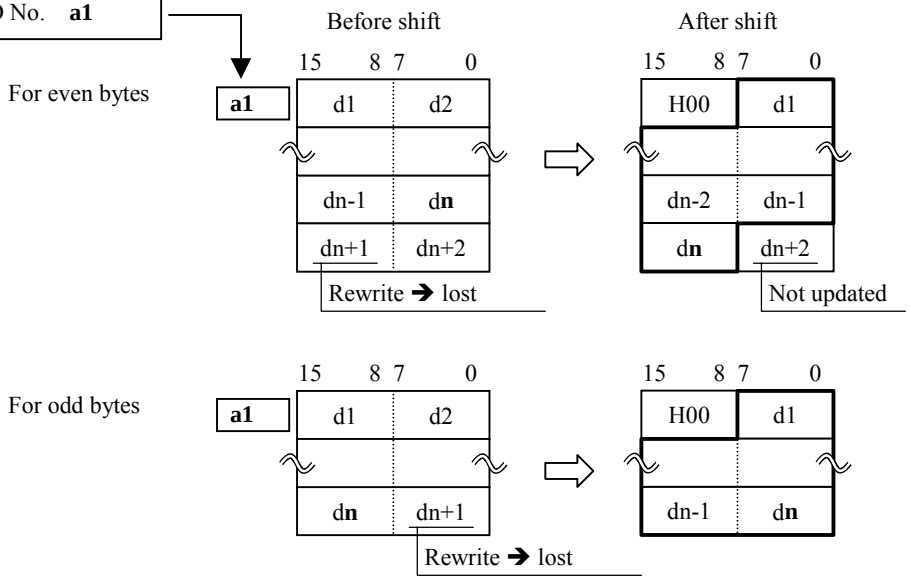
Ladder format	Condition code					Processing time (μs)	Remark
FUN 48 (s) * [BSHR (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	
	DER	ERR	SD	V	C	2.5 n + 183.5 (n : Number of shifted bytes)	
	↕	●	●	●	●		

Command format	Number of steps	
FUN 48 (s) * [BSHR (s)]	Condition	Steps
	—	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, 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

s	No. of shifted bytes **n**
s+1	Shift data head I/O No. **a1**



- 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."

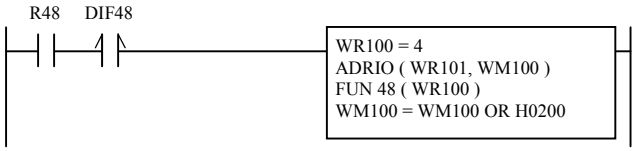
Cautionary notes

- 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.

* [ ] indicates the display when the LADDER EDITOR is used.

Name	Byte right shift

Program example




```
LD    R0048
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 "	⇨	WM100	H02	" T "
WM101	" X "	" T "		WM101	" E "	" X "
WM102	H00	H00		WM102	" T "	H00

Name	Byte left shift

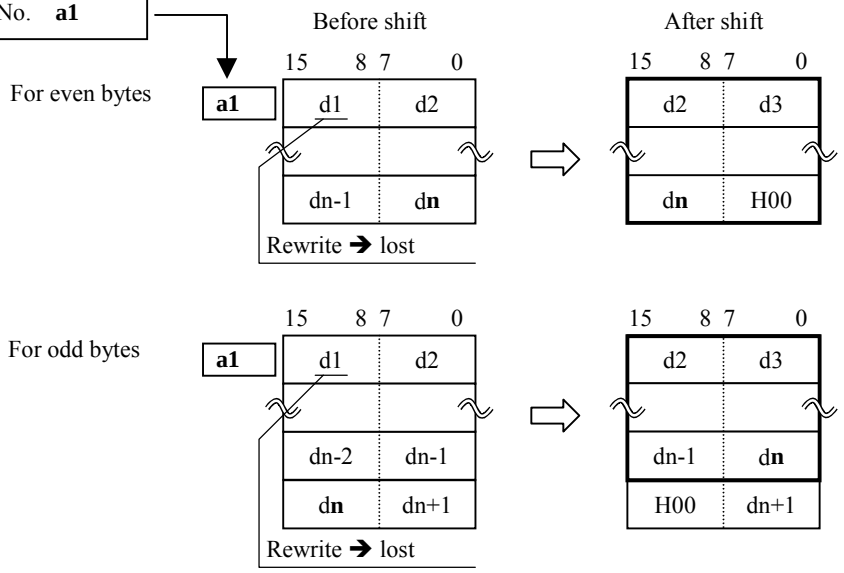
Ladder format	Condition code					Processing time (μs)	Remark
FUN 49 (s) * [BSHL (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	
	DER	ERR	SD	V	C	2.5 n + 186.3 (n : Number of shifted bytes)	
	↕	●	●	●	●		

Command format	Number of steps	
FUN 49 (s) * [BSHL (s)]	Condition	Steps
	—	3

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

s	No. of shifted bytes **n**
s+1	Shift data head I/O No. **a1**



- 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".

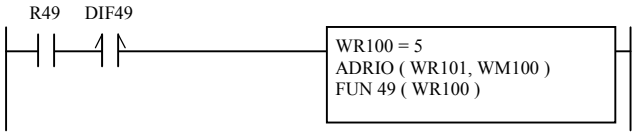
Cautionary notes

- 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.

* [ ] indicates the display when the LADDER EDITOR is used.

Name	Byte left shift

Program example




```
LD    R0049
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 "	⇨	WM100	" T "	" E "
WM101	" E "	" X "		WM101	" X "	" T "
WM102	" T "			WM102	H00	

■ 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

(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

Internal 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	$(1.00 \cdots 00)_2 \times 2^{7EH-7FH} = 0.5$

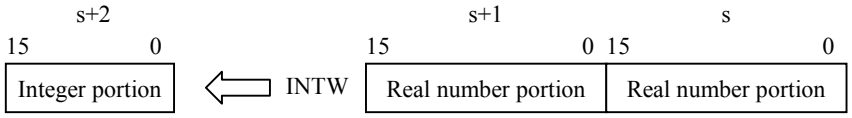
Name	Floating Point Operation (Real to Integer (Word) Conversion)

Ladder format	Condition code					Processing time (μs)		Remark
FUN 100 (s) * [INTW (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	80	←	
	↕	●	●	●	●			

Command format	Number of steps	
FUN 100 (s) * [INTW (s)]	Condition	Steps
	–	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument							○						s uses up to s+2.

Function

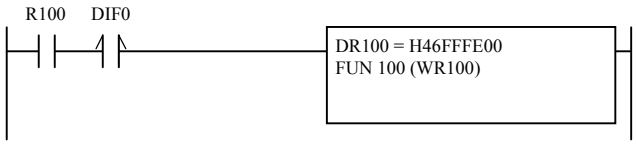


- Converts the real number specified by arguments s and s+1 to integer word data, then sets the result in s+2.
- If the calculation is completed normally, DER is equal to "0."
- The floating point format conforms to IEEE754.

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   R0100
AND  DIF0
[
DR0100 = H46FFFE00
FUN 100 (WR0100)
]
```

Program description

At a rising edge of R0100, 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

* [ ] indicates the display when the LADDER EDITOR is used.

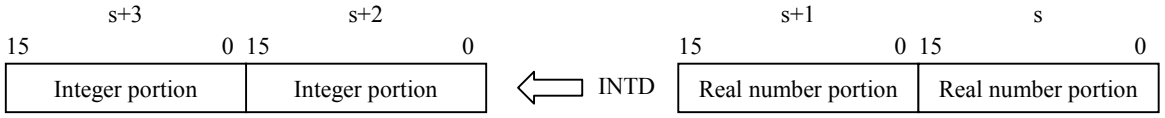
Name	Floating Point Operation (Real to Integer (Double Word) Conversion)

Ladder format	Condition code					Processing time (μs)		Remark
FUN 101 (s) * [INTD (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	96	←	
	⇕	●	●	●	●			

Command format	Number of steps	
FUN 101 (s) * [INTD (s)]	Condition	Steps
	—	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument							○						s uses up to s+3.

Function

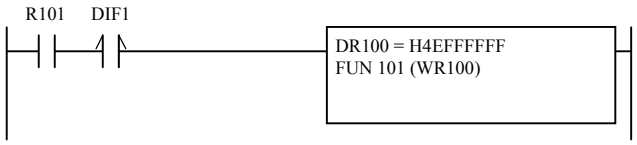


- Converts the real number specified by arguments s and s+1 to double word data, then sets the result in s+2 and s+3.
- If the calculation is completed normally, DER is equal to "0."
- The floating point format conforms to IEEE754

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   R0101
AND  DIF1
[
DR0100 = H4EFFFFFF
FUN 101 (WR0100)
]
```

Program description

At a rising edge of R0101, 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

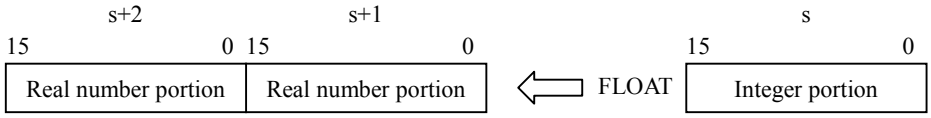
* [ ] indicates the display when the LADDER EDITOR is used.

Name	Floating Point Operation (Integer (Word) to Real Number Conversion)

Ladder format	Condition code					Processing time (μs)		Remark
FUN 102 (s) * [FLOAT (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	73	←	
	⇕	●	●	●	●			
Command format	Number of steps							
FUN 102 (s) * [FLOAT (s)]	Condition			Steps				
	—			3				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument							○						s uses up to s+2.

Function

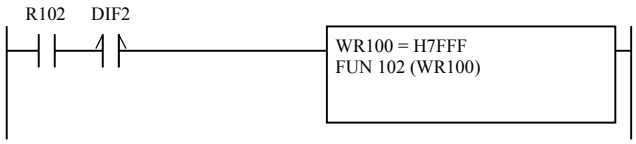


- Converts the integer word data s to a real number, then sets the result in s+1 and s+2.
- If the calculation is completed normally, DER is equal to "0."
- The floating point format conforms to IEEE754.

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   R0102
AND  DIF2
[
WR0100 = H7FFF
FUN 102 (WR0100)
]
```

Program description

At a rising edge of R0102, 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

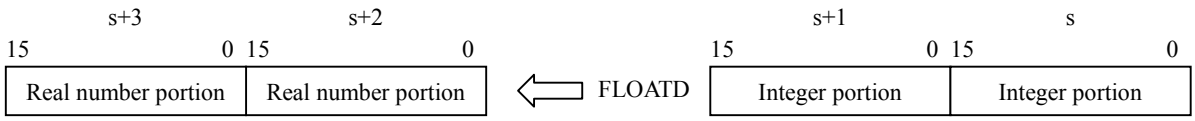
* [ ] indicates the display when the LADDER EDITOR is used.

Name	Floating Point Operation (Integer (Word) to Real Number Conversion)

Ladder format	Condition code					Processing time (μs)		Remark
FUN 103 (s) * [FLOATD (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	83	←	
	↕	●	●	●	●			
Command format	Number of steps							
FUN 103 (s) * [FLOATD (s)]	Condition			Steps				
	—			3				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument							○						s uses up to s+3.

Function

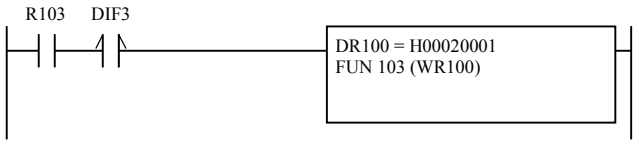


- Converts the integer double word data s and s+1 to a real number, then sets the result s+2 and s+3.
- If the calculation is completed normally, DER is equal to "0."
- The floating point format conforms to IEEE754.

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   R0103
AND  DIF3
[
DR0100 = H00020001
FUN 103 (WR0100)
]
```

Program description

At a rising edge of R0103, 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

* [ ] indicates the display when the LADDER EDITOR is used.

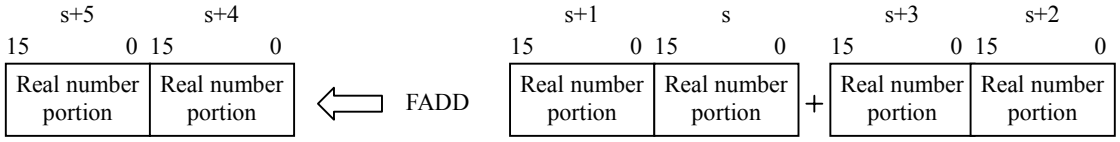
Name	Floating Point Operation (Addition)

Ladder format	Condition code					Processing time (μs)		Remark
FUN 104 (s) * [FADD (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	126	←	
	⇕	●	●	●	●			

Command format	Number of steps	
FUN 104 (s) * [FADD (s)]	Condition	Steps
	—	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument							○						s uses up to s+5.

Function

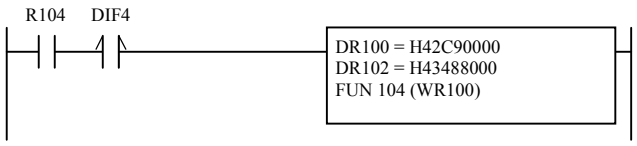


- Adds the real number (s+2, s+3) to the real number (s, s+1), then sets the result in (s+4, s+5).
- If the calculation is completed normally, DER is equal to "0."
- The floating point format conforms to IEEE754.

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   R0104
AND  DIF4
[
DR0100 = H42C90000
DR0102 = H43488000
FUN 104 (WR0100)
]
```

Program description

At a rising edge of R0104, 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

* [ ] indicates the display when the LADDER EDITOR is used.

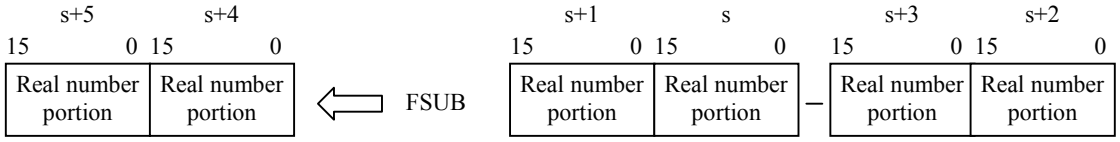
Name	Floating Point Operation (Subtraction)

Ladder format	Condition code					Processing time (μs)		Remark
FUN 105 (s) * [FSUB (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	126	←	
	↕	●	●	●	●			

Command format	Number of steps	
FUN 105 (s) * [FSUB (s)]	Condition	Steps
	–	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument							○						s uses up to s+5.

Function

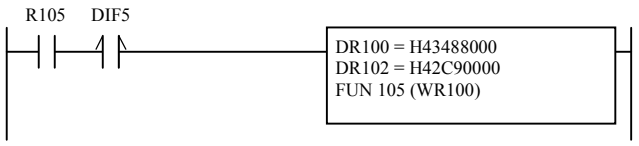


- Subtracts the real number (s+2, s+3) from the real number (s, s+1), then sets the result in (s+4, s+5).
- If the calculation is completed normally, DER is equal to "0."
- The floating point format conforms to IEEE754.

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   R0105
AND  DIF5
[
DR0100 = H43488000
DR0102 = H42C90000
FUN 105 (WR0100)
]
```

Program description

At a rising edge of R0105, 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

* [ ] indicates the display when the LADDER EDITOR is used.

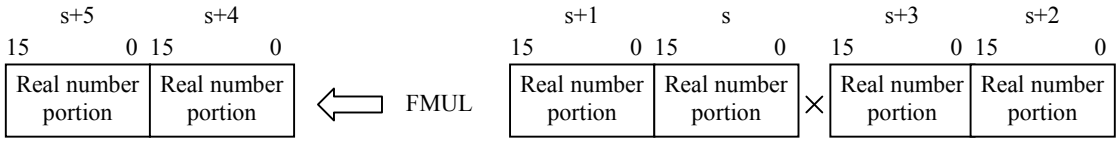
Name	Floating Point Operation (Multiplication)

Ladder format	Condition code					Processing time (μs)		Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
FUN 106 (s) * [FMUL (s)]	DER	ERR	SD	V	C	125	←	
	↕	●	●	●	●			

Command format	Number of steps	
	Condition	Steps
FUN 106 (s) * [FMUL (s)]	—	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument							○						s uses up to s+5.

Function

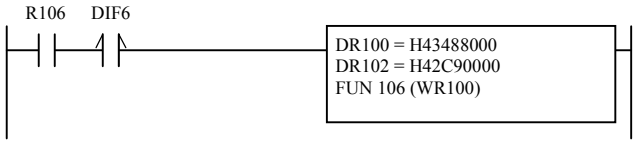


- Multiplies the real number (s, s+1) with the real number (s+2, s+3), then sets the result in (s+4, s+5).
- If the calculation is completed normally, DER is equal to "0."
- The floating point format conforms to IEEE754.

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   R0106
AND  DIF6
[
DR0100 = H43488000
DR0102 = H42C90000
FUN 106 (WR0100)
]
```

Program description

At a rising edge of R0106, 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

* [ ] indicates the display when the LADDER EDITOR is used.

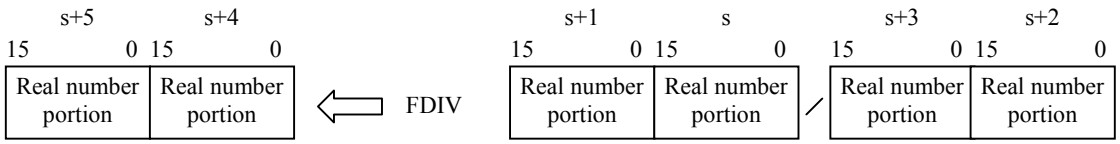
Name	Floating Point Operation (Division)

Ladder format	Condition code					Processing time (μs)		Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
FUN 107 (s) * [FDIV (s)]	DER	ERR	SD	V	C	160	←	
	↕	●	●	●	●			

Command format	Number of steps	
	Condition	Steps
FUN 107 (s) * [FDIV (s)]	—	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument							○						s uses up to s+5.

Function

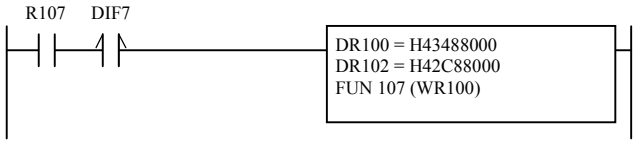


- Divides real number (s, s+1) by real number (s+2, s+3), then sets the result in (s+4, s+5).
- If the calculation is completed normally, DER is equal to "0."
- The floating point format conforms to IEEE754.

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   R0107
AND  DIF7
[
DR0100 = H43488000
DR0102 = H42C88000
FUN 107 (WR0100)
]
```

Program description

At a rising edge of R0107, 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

* [ ] indicates the display when the LADDER EDITOR is used.

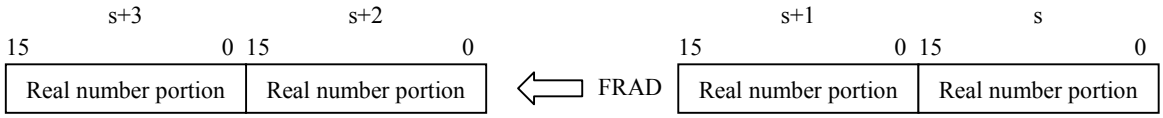
Name	Floating Point Operation (Angle to Radian Conversion)

Ladder format	Condition code					Processing time (μs)		Remark
FUN 108 (s) * [FRAD (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	110	←	
	↕	●	●	●	●			

Command format	Number of steps	
FUN 108 (s) * [FRAD (s)]	Condition	Steps
	—	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument							○						s uses up to s+3.

Function



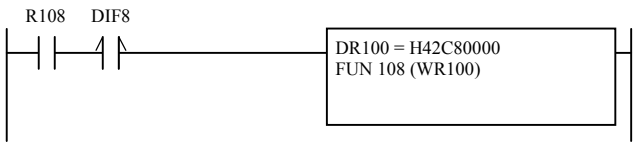
$$\text{degrees} \times \frac{\pi}{180} = \text{radian}$$

- 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.

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   R0108
AND  DIF8
[
DR0100 = H42C80000
FUN 108 (WR0100)
]
```

Program description

At a rising edge of R0108, 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

* [ ] indicates the display when the LADDER EDITOR is used.

Name	Floating Point Operation (Radian to Angle Conversion)

Ladder format	Condition code					Processing time (μs)		Remark
FUN 109 (s) * [FDEG (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	109	←	
	↕	●	●	●	●			

Command format	Number of steps	
FUN 109 (s) * [FDEG (s)]	Condition	Steps
	—	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument							○						s uses up to s+3.

Function

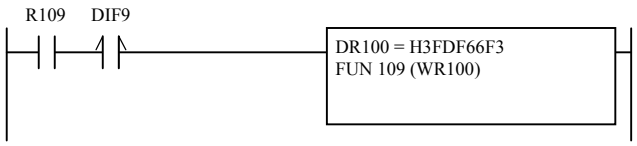
$$\text{radian} \times \frac{180}{\pi} = \text{degrees}$$

- 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.

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   R0109
AND  DIF9
[
DR0100 = H3FDF66F3
FUN 109 (WR0100)
]
```

Program description

At a rising edge of R0109, 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

* [ ] indicates the display when the LADDER EDITOR is used.

Name	Floating Point Operation (SIN)

Ladder format	Condition code					Processing time (μs)		Remark
FUN 110 (s) * [FSIN (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	381	←	
	↕	●	●	●	●			

Command format	Number of steps	
FUN 110 (s) * [FSIN (s)]	Condition	Steps
	–	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument							○						s uses up to s+3.

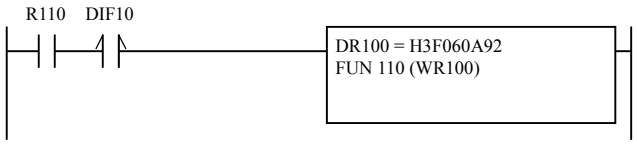
Function

- 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.

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 1.41484755040568800e+16, the sine value cannot be calculated, thus DER is set to "1."
- When the value of s, s+1 is greater than 2.98156826000000000e+08, a result is obtained but the accuracy decreases, so DER is set to "1."

Program example




```
LD   R0110
AND  DIF10
[
DR0100 = H3F060A92
FUN 110 (WR0100)
]
```

Program description

At a rising edge of R0110, 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

* [ ] indicates the display when the LADDER EDITOR is used.

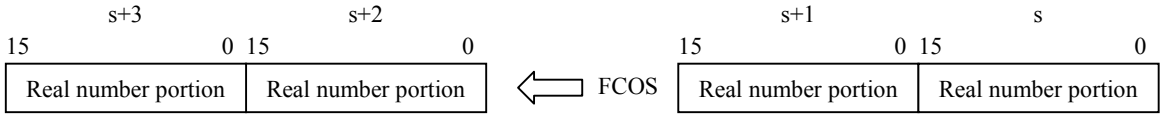
Name	Floating Point Operation (COS)

Ladder format	Condition code					Processing time (μs)		Remark
FUN 111 (s) * [FCOS (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	428	←	
	↕	●	●	●	●			

Command format	Number of steps	
FUN 111 (s) * [FCOS (s)]	Condition	Steps
	–	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument							○						s uses up to s+3.

Function

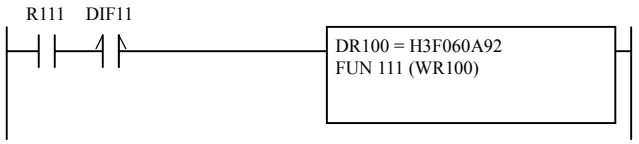


- Calculates the cosine value of the real number value in radian units specified in s and s+1 as the arguments, the sets the result in s+2 and s+3.
- If the calculation is completed normally, DER is equal to “0”.
- The floating point format conforms to IEEE754.

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 1.41484755040568800e+16, the cosine value cannot be calculated and DER is set to "1".
- When the value of s, s+1 is greater than 2.98156826000000000e+08, a result is obtained but the accuracy decreases, so DER is set to "1".

Program example




```
LD   R0111
AND DIF11
[
DR0100 = H3F060A92
FUN 111 (WR0100)
]
```

Program description

At a rising edge of R0111, 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

* [ ] indicates the display when the LADDER EDITOR is used.

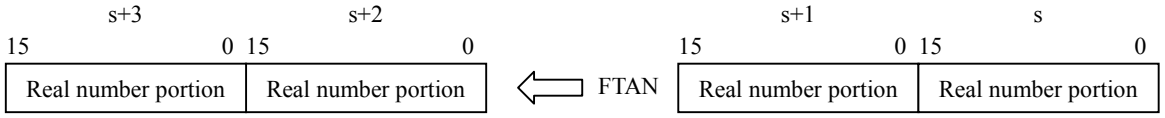
Name	Floating Point Operation (TAN)

Ladder format	Condition code					Processing time (μs)		Remark
FUN 112 (s) * [FTAN (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	411	←	
	↕	●	●	●	●			

Command format	Number of steps	
FUN 112 (s) * [FTAN (s)]	Condition	Steps
	–	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument							○						s uses up to s+3.

Function

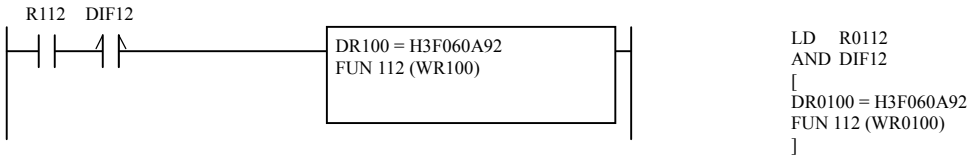


- Calculates the tangent value of the real number value in radian units specified in s and s+1 as the arguments, the sets the result in s+2 and s+3.
- If the calculation is completed normally, DER is equal to “0”.
- The floating point format conforms to IEEE754.

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 1.41484755040568800e+16/2, the tangent value cannot be calculated and DER is set to "1".
- When the value of s, s+1 is greater than 2.98156826000000000e+08/2, a result is obtained but the accuracy decreases, so DER is set to "1".

Program example




```
LD   R0112
AND  DIF12
[
DR0100 = H3F060A92
FUN 112 (WR0100)
]
```

Program description

At a rising edge of R0112, 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

* [ ] indicates the display when the LADDER EDITOR is used.

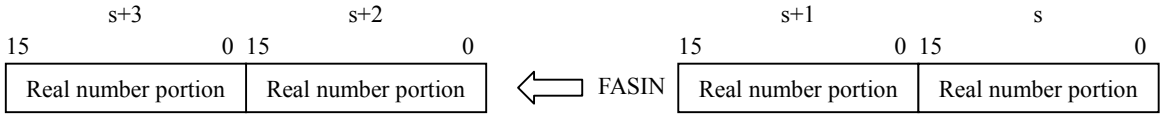
Name	Floating Point Operation (ARC SIN)

Ladder format	Condition code					Processing time (μs)		Remark
FUN 113 (s) * [FASIN (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	321	←	
	↕	●	●	●	●			

Command format	Number of steps	
FUN 113 (s) * [FASIN (s)]	Condition	Steps
	–	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument							○						s uses up to s+3.

Function

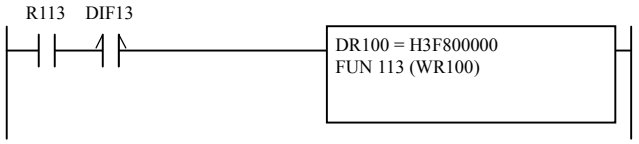


- Calculates the SIN $^{-1}$ value of the real number value specified in s and s+1 as the arguments, and sets the result in radian units in s+2 and s+3.
- If the calculation is completed normally, DER is equal to “0.”
- The floating point format conforms to IEEE754.

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   R0113
AND  DIF13
[
DR0100 = H3F800000
FUN 113 (WR0100)
]
```

Program description

At a rising edge of R0113, 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

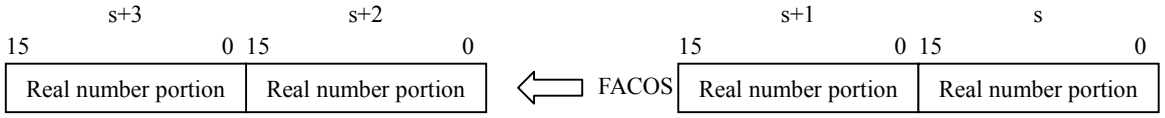
* [ ] indicates the display when the LADDER EDITOR is used.

Name	Floating Point Operation (ARC COS)

Ladder format	Condition code					Processing time (μs)		Remark
FUN 114 (s) * [FACOS (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	314	←	
	↕	●	●	●	●			
Command format	Number of steps							
FUN 114 (s) * [FACOS (s)]	Condition			Steps				
	—			3				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument							○						s uses up to s+3.

Function

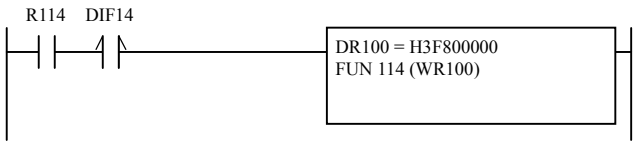


- Calculates the COS^{-1} value of the real number value specified in s and s+1 as the arguments, and sets the result in radian units in s+2 and s+3.
- If the calculation is completed normally, DER is equal to "0."
- The floating point format conforms to IEEE754.

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   R0114
AND  DIF14
[
DR0100 = H3F800000
FUN 114 (WR0100)
]
```

Program description

At a rising edge of R0114, 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

* [ ] indicates the display when the LADDER EDITOR is used.

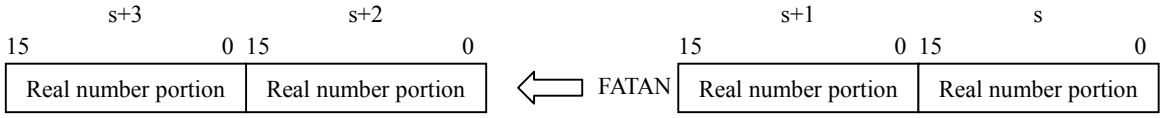
Name	Floating Point Operation (ARC TAN)

Ladder format	Condition code					Processing time (μs)		Remark
FUN 115 (s) * [FATAN (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	443	←	
	↕	●	●	●	●			

Command format	Number of steps	
FUN 115 (s) * [FATAN (s)]	Condition	Steps
	—	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument							○						s uses up to s+3.

Function

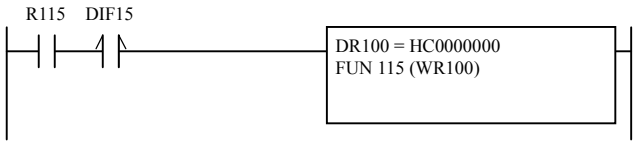


- Calculates the TAN $^{-1}$ value of the real number value specified in s and s+1 as the arguments, and sets the result in radian units in s+2 and s+3.
- If the calculation is completed normally, DER is equal to "0."
- The floating point format conforms to IEEE754.

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   R0115
AND  DIF15
[
DR0100 = HC0000000
FUN 115 (WR0100)
]
```

Program description

At a rising edge of R0115, 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

* [ ] indicates the display when the LADDER EDITOR is used.

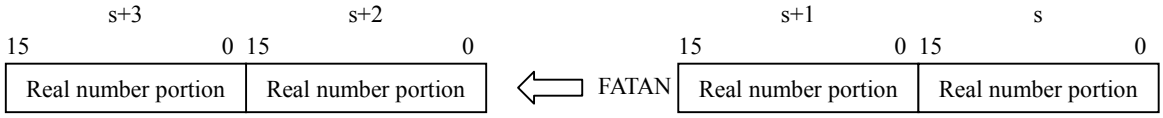
Name	Floating Point Operation (Square Root)

Ladder format	Condition code					Processing time (μs)		Remark
FUN 116 (s) * [FSQR (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	532	←	
	↕	●	●	●	●			

Command format	Number of steps	
FUN 116 (s) * [FSQR (s)]	Condition	Steps
	–	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument							○						s uses up to s+3.

Function

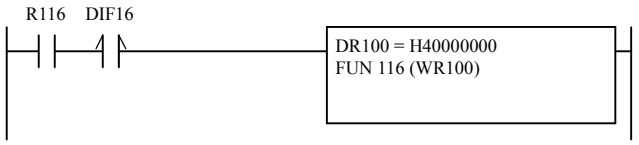


- Calculates the square root using the real number value specified in s and s+1 as the arguments, the sets the result in s+2 and s+3.
- If the calculation is completed normally, DER is equal to “0”.
- The floating point format conforms to IEEE754.

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   R0116
AND  DIF16
[
DR0100 = H40000000
FUN 116 (WR0100)
]
```

Program description

At a rising edge of R0116, 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

* [ ] indicates the display when the LADDER EDITOR is used.

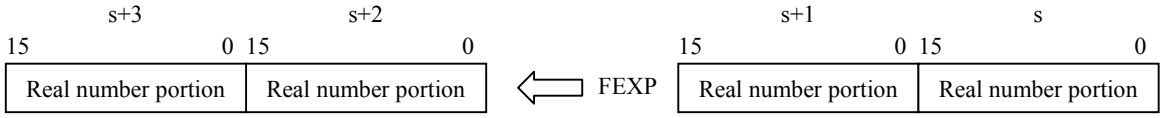
Name	Floating Point Operation (Exponent)

Ladder format	Condition code					Processing time (μs)		Remark
FUN 117 (s) * [FEXP (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	392	←	
	↕	●	●	●	●			

Command format	Number of steps	
FUN 117 (s) * [FEXP (s)]	Condition	Steps
	–	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument							○						s uses up to s+3.

Function

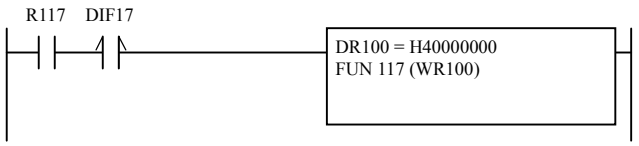


- Performs an exponent operation using the real number value specified in s and s+1 as the arguments, the sets the result in s+2 and s+3.
- An exponent operation is performed using 2.71828 as the base (e).
- If the calculation is completed normally, DER is equal to “0.”
- The floating point format conforms to IEEE754.

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   R0117
AND  DIF17
[
DR0100 = H40000000
FUN 117 (WR0100)
]
```

Program description

At a rising edge of R0117, 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

* [ ] indicates the display when the LADDER EDITOR is used.

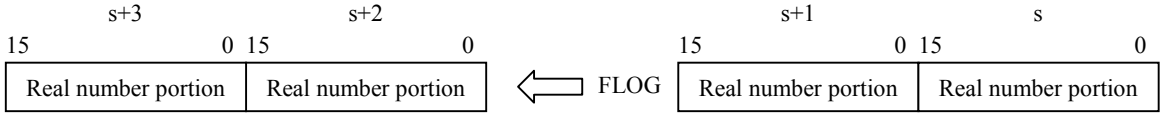
Name	Floating Point Operation (Natural Logarithm)

Ladder format	Condition code					Processing time (μs)		Remark
FUN 118 (s) * [FLOG (s)]	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
	DER	ERR	SD	V	C	289	←	
	↕	●	●	●	●			

Command format	Number of steps	
FUN 118 (s) * [FLOG (s)]	Condition	Steps
	–	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument							○						s uses up to s+3.

Function

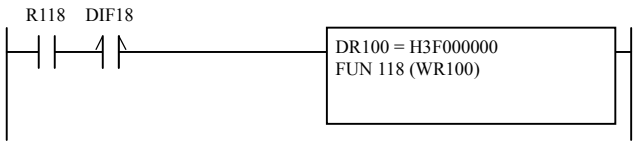


- Performs a logarithm operation for the real number value specified by arguments s and s+1 using the natural logarithm (e) as the base, then sets the result in s+2 and s+3.
- If the calculation is completed normally, DER is equal to “0.”
- The floating point format conforms to IEEE754.

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   R0118
AND  DIF18
[
DR0100 = H3F000000
FUN 118 (WR0100)
]
```

Program description

At a rising edge of R0118, 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

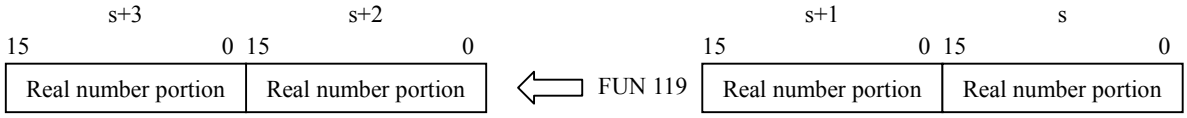
* [ ] indicates the display when the LADDER EDITOR is used.

Name	Floating Point Operation (Common logarithm)

Ladder format	Condition code					Processing time (μs)		Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
FUN 119 (s)	DER	ERR	SD	V	C	474	←	
	↕	●	●	●	●			
Command format	Number of steps							
FUN 119 (s)	Condition			Steps				
	—			3				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument							○						s uses up to s+3.

Function

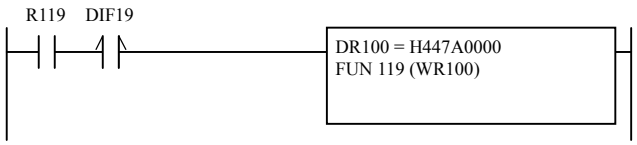


- Performs a logarithm operation for the real number value specified by arguments s and s+1 using the common logarithm (10) as the base, then sets the result in s+2 and s+3.
- If the calculation is completed normally, DER is equal to “0.”
- The floating point format conforms to IEEE754.

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   R0119
AND  DIF19
[
DR0100 = H447A0000
FUN 119 (WR0100)
]
```

Program description

At a rising edge of R0119, 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 : WR101=H447A, WR100=H0000
Operation result : WR103=H4040, WR102=H0000

* [ ] indicates the display when the LADDER EDITOR is used.

Name	High-speed Counter Current Value Replacement

Ladder format	Condition code					Processing time (μs)		Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
FUN 143 (s)	DER	ERR	SD	V	C	63.5	←	Upper case: 16-bit Lower case: 32-bit
	↕	●	●	●	●			

Command format	Number of steps	
	Condition	Steps
FUN 143 (s)	—	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument (counter number)							○						
s+1	Argument (Replacement value storage area)							○						
s+2	Argument (Replacement value storage area)							○						Only 32bit counter used.

Function

	15	8 7	0
s	Counter number	**	
s+1	Replacement value storage area		
s+2	Replacement value storage area		

Counter number: H01 to H04
**: Disable area

s+2: At the time of 32-bit counter use

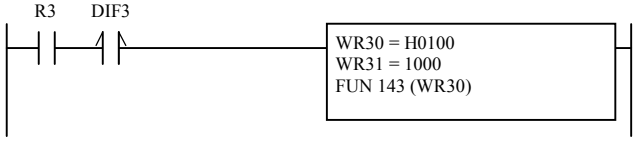
- The counter value of the specified counter number will be replaced by the data stored in the replacement value storage area.

Cautionary notes

- When using a 16-bit counter, s+2 is not used.
- If a value other than H01 to H04 is specified for the counter number, DER will be set to "1" and no processing will be performed.
- If the specified counter number is set to a function other than a corresponding external I/O counter (single-phase counter, two-phase counter), DER will be set to "1"and no processing will be performed.
- Since Counter 4 is invalid when a 10-point CPU is used, if Counter 4 is specified, DER will be set to "1" and no processing will be performed.
- If the specified counter number is unable to make an output (PI/O function setting result by R7F5), DER will be set to "1" and no processing will be performed.
- This instruction is only used to rewrite the count value. Other counter settings will not be changed and will not affect the count operation.
- If the range for S exceeds the valid range of the I/O, DER will be set to "1" and no processing will be performed.

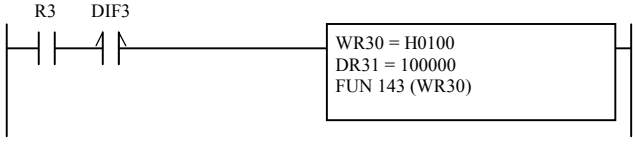
Program example

[In case of 16-bit counter]




```
LD      R3
AND     DIF3
[
WR30    = H100
WR31    = 1000
FUN   143 ( WR30 )
]
```

[In case of 32-bit counter]




```
LD      R3
AND     DIF3
[
WR30    = H100
DR31    = 100000
FUN   143 ( WR30 )
]
```

Name	High-speed Counter Current Value Replacement

Program description

[In case of 16-bit counter] Rewrite the count value of the Counter number 1 to 1000.
[In case of 32-bit counter] Rewrite the count value of the Counter number 1 to 100,000.

Name	High-speed counter current value reading

Ladder format	Condition code					Processing time (μs)		Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
FUN 144 (s)	DER	ERR	SD	V	C	64.9	←	Upper case: 16-bit Lower case: 32-bit
	↕	●	●	●	●			

Command format	Number of steps	
	Condition	Steps
FUN 144 (s)	—	3

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument (counter number)							○						
s+1	Argument (Current value storage area)							○						
s+2	Argument (Current value storage area)							○						Only 32bit counter used.

Function

	15	8	7	0
s	Counter number		**	
s+1	Current value storage area			
s+2	Current value storage area			

Counter number: H01 to H04
**: Disable area

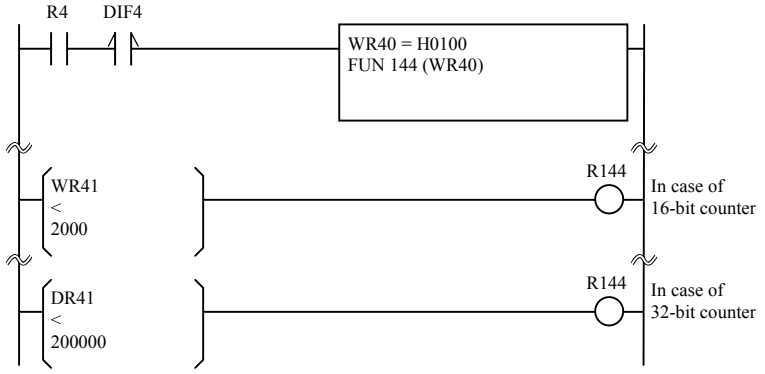
s+2: At the time of 32-bit counter use

- This function reads the count value of the specified counter number and writes it to the current value storage area.

Cautionary notes

- When using a 16-bit counter, s+2 is not used.
- If a value other than H01 to H04 is specified for the counter number, DER will be set to "1" and no processing will be performed.
- If the specified counter number is set to a function other than a corresponding external I/O counter (single-phase counter, two-phase counter), DER will be set to "1"and no processing will be performed.
- Since Counter 4 is invalid when a 10-point CPU is used, if Counter 4 is specified, DER will be set to "1" and no processing will be performed.
- If the specified counter number is unable to make an output (PI/O function setting result by R7F5), DER will be set to "1" and no processing will be performed.
- This instruction is only used to read the count value. Other counter settings will not be changed and it will not affect the count operation.
- The execution of this instruction will not change WRF07A to WRF07D (strobe area) and WRF056 (strobe complete flag).
- If the range for S exceeds the valid range of the I/O, DER will be set to "1" and no processing will be performed.

Program example




```
LD      R4
AND     DIF4
[
WR40    = H100
FUN     144 ( WR40 )
]
≈
LD      ( WR41 < 2000 )
OUT     R144
≈
LD      ( DR41 < 200000 )
OUT     R144
```

Name	High-speed counter current value reading

Program description

[In case of 16-bit counter] Load the count value of the Counter number 1 to WR41.
If the count value of the Counter number 1 is less than 2,000, R144 is turned on.

[In case of 32-bit counter] Load the count value of the Counter number 1 to DR41 (WR41, WR42).
If the count value of the Counter number 1 is less than 200,000, R144 is turned on.

Name	High-speed counter preset

Ladder format	Condition code					Processing time (μs)		Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
FUN 146 (s)	DER	ERR	SD	V	C	81.5	←	Upper case: 16-bit Lower case: 32-bit
	↕	●	●	●	●			

Command format	Number of steps			
	Condition		Steps	
FUN 146 (s)	–		3	

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument(counter number, preset specification)							○						
s+1	Argument (on-preset value)							○						
s+2	Argument (on-preset value)							○						16 bit counter : off-preset value
s+3	Argument (off-preset value)							○						16 bit counter : not used
s+4	Argument (off-preset value)							○						16 bit counter : not used

Function

[32-bit Counter]

	15 8	7 0
s	Counter number	Preset specification
s+1	on-preset value	
s+2	on-preset value	
s+3	off-preset value	
s+4	off-preset value	

Counter number : H01 to H04
Preset specification : H00 – Specification of on-preset value and off-preset value
H01 – Specification of on-preset value only
H02 – Specification of off-preset value only

[16-bit Counter]

	15 8	7 0
s	Counter number	Preset specification
s+1	on-preset value	
s+2	off-preset value	

- The on-preset value and off-preset value will be set according to the preset specifications for the specified counter number.
- The coincidence output value will remain unchanged even when coincidence output is possible.

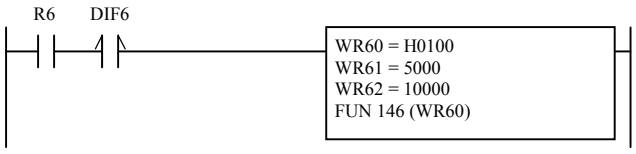
Name	High-speed counter preset

Cautionary notes

- If a value other than H01 to H04 is specified for the counter number and a value other than H00 to H02 is set for the preset specification, DER will be set to "1" and no processing will be performed.
- Since Counter 4 is invalid when a 10-point CPU is used, if Counter 4 is specified, DER will be set to "1" and no processing will be performed.
- If the specified counter number is set to a function other than a corresponding external I/O counter (single-phase counter, two-phase counter), DER will be set to "1" and no processing will be performed.
- The specified preset value will be checked using the criteria shown below. If an error occurs, DER will be set to "1" and no processing will be performed.
  If there is no error, the bit respective to the setting error detail information WRF057 will be set to "0" and releases the operation disabled status.
  1] When the preset specification is 00H
  16-bit counter : If S+1 (on-preset) and S+2 (off-preset) values are equal, and error is generated.
  32-bit counter : If S+1~S+2 (on-preset) and S+3~S+4 (off-preset) values are equal, and error is generated.
  2] When the preset specification is 01H
  16-bit counter : If S+1 (on-preset) and the off-preset value of WRF076 to WRF079 are equal, an error is generated.
  32-bit counter : If S+1~S+2 (on-preset) and the off-preset value of WRF1B8 to WRF1BF are equal, an error is generated.
  3] When the preset specification is 02H
  16-bit counter : If S+2 (off-preset) and the on-preset value of WRF072 to WRF075 are equal, an error is generated.
  32-bit counter : If S+3~S+4 (off-preset) and the on-preset value of WRF1B0 to WRF1B7 are equal, an error is generated.
  Although the 64-point type CPU does not become an error when the ON preset value / OFF preset value is in agreement by 0, even if conditions are ready, a coincidence output does not turn on.
- This instruction is used only to set the on-preset value and off-preset value. Other counter settings will not be changed and it will not affect the count operation.
- The settings made using the instruction will be reflected in the special internal output (WRF072 to WRF075 and WRF076 to WRF078 / WRF1B0 to WRF1B7 and WRF1B8 to WRF1BF). However, it is not reflected if DER becomes equal to "1."
- If the range for S exceeds the valid range of the I/O, DER will be set to "1" and no processing will be performed.

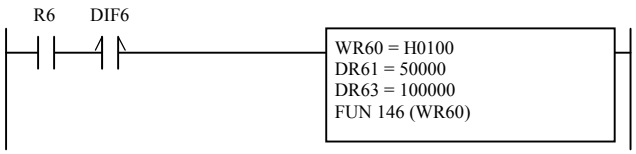
Program example

[In case of 16-bit counter]




```
LD      R6
AND     DIF6
[
WR60    = H100
WR61    = 5000
WR62    = 10000
FUN     146 ( WR60 )
]
```

[In case of 32-bit counter]




```
LD      R6
AND     DIF6
[
WR60    = H100
DR61    = 50000
DR63    = 100000
FUN     146 ( WR60 )
]
```

Program description

[In case of 16-bit counter] Sets both the on-preset value and off-preset value in the counter number 1.
Sets 5,000 for the on-preset value and 10,000 for the off-preset value.
[In case of 32-bit counter] Sets both the on-preset value and off-preset value in the counter number 1.
Sets 50,000 for the on-preset value and 100,000 for the off-preset value.

Name	Pulse frequency output setting changes

Ladder format	Condition code					Processing time (μs)		Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
FUN 150 (s)	DER	ERR	SD	V	C	132.9	←	Upper case: 16-bit Lower case: 32-bit
	↕	●	●	●	●			

Command format	Number of steps			
	Condition		Steps	
FUN 150 (s)	—		3	

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument (Pulse number)							○						
s+1	Argument (Frequency value)							○						
s+2	Argument (Number of output pulses)							○						
s+3	Argument (Number of output pulses)							○						Except mode 20-23 : not used

Function

	15	8	7	0
s	Pulse number		Change specification	
s+1	Frequency value			
s+2	Number of pulse output (Low word)			
s+3	Number of pulse output (High word)			

Pulse output number : H01 to H04
Change specification : H00: Sets the frequency value and number of pulse output,
H01: Sets the frequency value only,
H02: Sets the number of pulse output

← The modes other than mode 2x : Not used.

- Pulse output is commenced at the specified frequency. Output is stopped once the number of pulses specified have been output.
- Sets the frequency value in Hz.
  Example: To set a frequency of 10kHz, set 10000 (H2710) as internal output.
- Sets the count for the number of output pulses.
  Example: Mode 2x - To set output of 1,000,000, set 1,000,000 (HF4240) as internal output(double word).
  Except mode 2x - To set output of 60,000, set 60,000 (HEA60) as internal output(word).

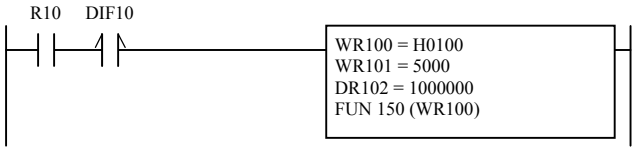
Cautionary notes

- If the pulse output number is set to a value other than H01 to H04, DER will be set to "1"and no processing will be performed.
- If the external I/O corresponding to the pulse output number is set to a function other than pulse output, DER will be set to "1"and no processing will be performed.
- The minimum frequency that can be supported is 10 kHz. If a frequency value smaller than 10 kHz is specified, it will be changed to 10 kHz internally by the system.
- In case of mode 2x : The settings by this instruction will be reflected in the special internal output (WRF1B0 to WRF1B7 and WRF1C0 to WRF1C7).
  Except above : The settings by this instruction will be reflected in the special internal output (WRF072 to WRF075 and WRF07A to WRF07D).
- If the range for S exceeds the valid range of the I/O, DER will be set to "1" and no processing will be performed.
- If the pulse output number is set to "0," pulse output will not be performed even when the pulse output start (R7FC to R7FF is set to "1" or FUN149) is set.
- If this instruction is executed for the I/O that is outputting a pulse with the acceleration/deceleration function, DER will be set to "1" and no processing will be performed.

Name	Pulse frequency output setting changes

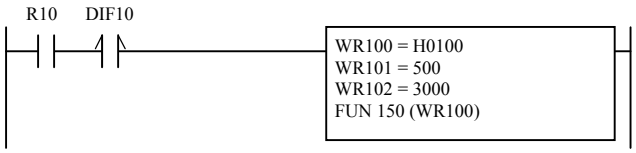
Program example

[In case of mode 2x]




```
LD      R10
AND     DIF10
[
WR100 = H0100
WR101 = 5000
DR102 = 1000000
FUN 150 (WR100)
]
```

[Except above]




```
LD      R10
AND     DIF10
[
WR100 = H0100
WR101 = 500
WR102 = 3000
FUN 150 (WR100)
]
```

Program description

[In case of mode 2x] Sets both the frequency and pulse output count of the pulse output No. 1 (Y100).
Sets 5000 (Hz) for the frequency and 1,000,000 for the number of pulse outputs.

[Except avobe] Sets both the frequency and pulse output count of the pulse output No. 1 (Y100).
Sets 500 (Hz) for the frequency and 3,000 for the number of pulse outputs.

Name	Pulse output with acceleration/deceleration

Ladder format	Condition code					Processing time (μs)		Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
FUN 151 (s)	DER	ERR	SD	V	C	1,418	←	Upper case: 16-bit Lower case: 32-bit
	↕	●	●	●	●			
Command format	Number of steps					1,324	←	(Processing time from executing command to pulse output.)
FUN 151 (s)	Condition			Steps				
	–			3				

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument (Pulse output number							○						
s+1	Argument (Total No. of output pulses)							○						
s+2	Argument (Total No. of output pulses)							○						Except mode 20-23 : Same as s+3
s+3	Argument (Maximum frequency (Hz))							○						Except mode 20-23 : Same as s+4
s+4	Argument (Initial frequency (Hz))							○						Except mode 20-23 : Same as s+5
s+5	Argument (Acceleration / deceleration time (ms))							○						Except mode 20-23 : not used

Function

[In case of mode 2x]

	15 … 8	7 … 0
s	Pulse output number	* *
s+1	Total No. of output pulses N (Low word)	
s+2	Total No. of output pulses N (High word)	
s+3	Maximum frequency F (Hz)	
s+4	Initial frequency F_0 (Hz)	
s+5	Acceleration / deceleration time T (ms)	

Pulse output No. : H01 to H04
** : Invalid area

[Except above]

	15 … 8	7 … 0
s	Pulse output number	* *
s+1	Total No. of output pulses N	
s+2	Maximum frequency F (Hz)	
s+3	Initial frequency F_0 (Hz)	
s+4	Acceleration / deceleration time T (ms)	

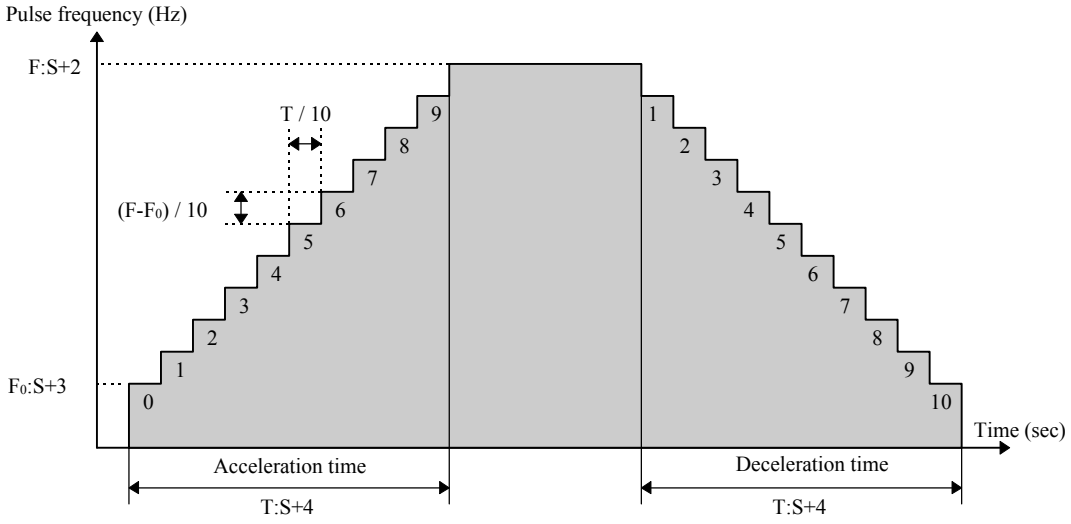
Pulse output No. : H01 to H04
** : Invalid area

- This instruction outputs pulses with the acceleration/deceleration function.
- It outputs pulses from the pulse output terminal set with the pulse output number s until the total number of output pulses set with s+1, s+2 (s+1) is reached.
- Since the output of pulses starts from the one having the frequency set with s+4 (s+3), set the parameters so that the stepping motor and other devices will not become out of tune.
- Acceleration is performed at the acceleration time set with s+5(s+4) in 10 steps until the maximum frequency set with s+3 (s+2) is reached.
- Deceleration is performed at the deceleration time set with s+5 (s+4) until the total number of output pulses set with s+2 (s+1) is reached. The ratio of frequency change for the deceleration is the same as for the acceleration.

* ( ) : In the cases of other than mode 2x

Name	Pulse output with acceleration/deceleration

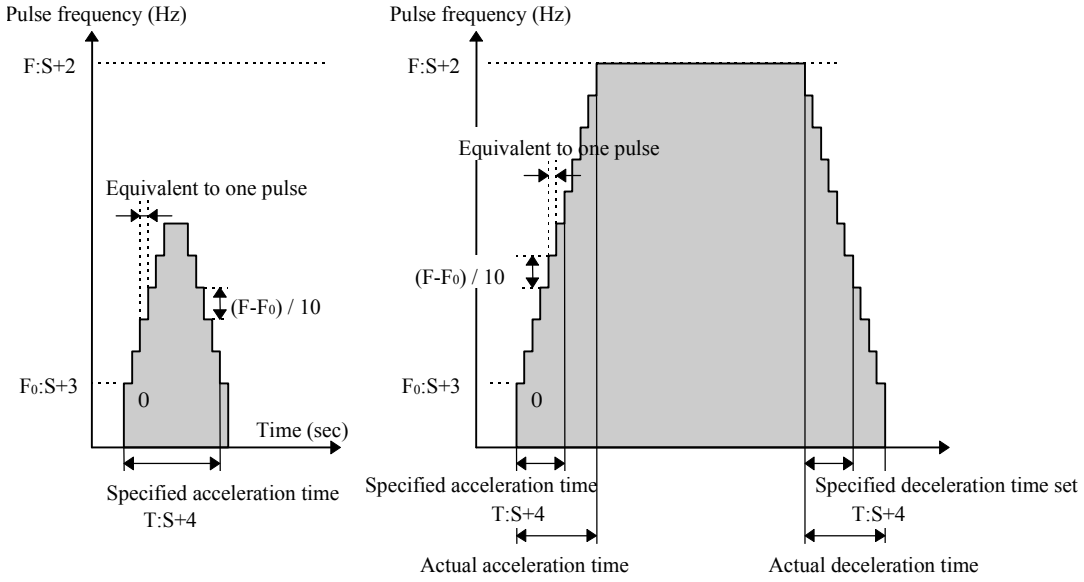
Function



Pulse output (normal setting)

Cautionary notes

- When this instruction is executed, the maximum frequency is stored in the special internal output's pulse output frequency (WRF1B0 to WRF1B7, WRF072 to WFR075), and the number of output pulses is stored in the special internal output's number of output pulses (WRF1C0 to WRF1C7, WRF07A to WRF07D) respectively.
- This instruction will not be executed if the specified pulse output is generating pulse output.
- If the output that corresponds to the specified pulse output number has not been set for pulse output, DER will be set to "1" and pulse output will not be generated.
- If the maximum frequency is larger than the initial frequency, DER will be set to "1" and pulse output will not be generated.
- If the same value is specified for the maximum frequency and initial frequency, pulses will be output for the number of pulses set with the maximum cycle without acceleration/deceleration.
- If the maximum frequency and initial frequency are set to a value smaller than 10 Hz, the specified values will be changed to 10 Hz by the system.
- If the total number of output pulses is small, deceleration will be performed without accelerating up to the maximum frequency.
  - In this case, the specified acceleration/deceleration time will not be used as the acceleration/deceleration time; it will be accelerated (or decelerated) for each pulse.
- For the acceleration/deceleration time, set a value equal to or larger than (1 / maximum frequency + 1 / initial frequency) x 5. If an acceleration/deceleration time smaller than this value is specified, the specified acceleration/deceleration will not be set.
- Acceleration and deceleration are performed in 10 steps, and at least one or more pulses are always output. Thus, if a small initial frequency value is specified, an error in the acceleration/deceleration time will become large.

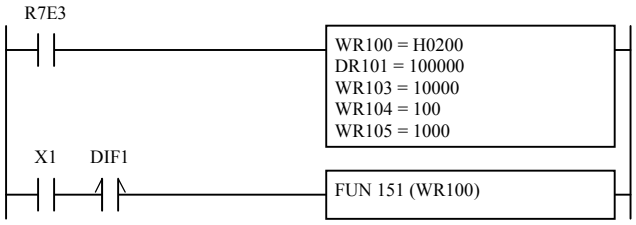


Pulse output (abnormal setting)

Name	Pulse output with acceleration/deceleration

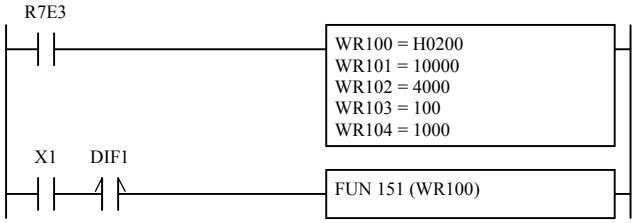
Program example

[In case of mode 2x]




```
LD      R7E3
[
WR100 = H0200
DR101 = 100000
WR103 = 10000
WR104 = 100
WR105 = 1000
]
LD      X00001
AND     DIF1
[
FUN 151 (WR100)
]
```

[Except above]




```
LD      R7E3
[
WR100 = H0200
WR101 = 10000
WR102 = 4000
WR103 = 100
WR104 = 1000
]
LD      X00001
AND     DIF1
[
FUN 151 (WR100)
]
```

Program description

[In case of mode 2x] Sets the required parameters in the special internal outputs at the first scan after RUN start.
At the leading edge of X00001, pulses are output starting from Y101 using the following settings: acceleration / deceleration time of 1000 (ms), initial frequency of 100 (Hz), maximum frequency of 10,000 (Hz), and number of output pulses of 100,000 pulses.

[Except avobe] Sets the required parameters in the special internal outputs at the first scan after RUN start.
At the leading edge of X00001, pulses are output starting from Y101 using the following settings: acceleration / deceleration time of 1000 (ms), initial frequency of 100 (Hz), maximum frequency of 4000 (Hz), and number of output pulses of 10,000 pulses.

Name	Pulse output with sequence parameter change

Ladder format	Condition code					Processing time (μs)		Remark
	R7F4	R7F3	R7F2	R7F1	R7F0	Ave	Max	
FUN 153 (s)	DER	ERR	SD	V	C	169	15,095	Upper case: 16-bit Lower case: 32-bit
	↕	●	●	●	●			

Command format	Number of steps	
	Condition	Steps
FUN 153 (s)	—	3

Processing time (μs)		Remark
Ave	Max	
173	15,112	(Processing time from executing command to pulse output. The maximum time in case table number is set as 256.)

Usable I/O		Bit				Word				Double word			Constant	Other
		X	Y	R, M	TD, SS, CU, CT	WX	WY	WR, WM	TC	DX	DY	DR, DM		
s	Argument (Pulse No, Table No.)							○						
s+1	Argument (Output table No.)							○						Set by the system
s+2	Argument (Output frequency (Hz))							○						s+2 to s+5 is required by the number of tables.
s+3	Argument (Table change event)							○						
s+4	Argument (Event information)							○						
s+5	Argument (Acceleration (Event information)							○						

Function

• This command performs a pulse output according to the parameter beforehand registered into the table.

	15 — 8	7 — 0
s	Pulse No.	Number of table
s+1	Table No. (current output table)	
s+2	Table 1 : Output frequency (Hz)	
s+3	Table 1 : Table change event specification	
s+4	Table 1 : Event information (1)	
s+5	Table 1 : Event information (2)	
≈	≈	≈
s+4n+2	Table n : Output frequency (Hz)	
s+4n+3	Table n : Table change event specification	
s+4n+4	Table n : Event information (1)	
s+4n+5	Table n : Event information (2)	

Pulse No. : H01 to H04
Number of table(n) : H01 to HFF (1 to 255)
* s+1 is set by the system.

One table consists of 4 words.
Please refer to details about each parameter.

• From the pulse output terminal specified in s+0, a pulse output is performed with the parameter registered into the table.
• The numbers of tables which can be registered are H01-HFF (1-255).
• Generating of the event registered into the table switches the parameter of a pulse output to the parameter of the next table.
• Generating of the event of the last of a table suspends a pulse output.

[s+0] Pulse No, Number of table
A pulse output terminal is set to a high byte, and the number of tables is set to a low byte.

[s+1] Table No. (current output table)
Table No. in which the parameter of the pulse currently outputted is stored is displayed. (It sets by the system.)

Name	Pulse output with sequence parameter change

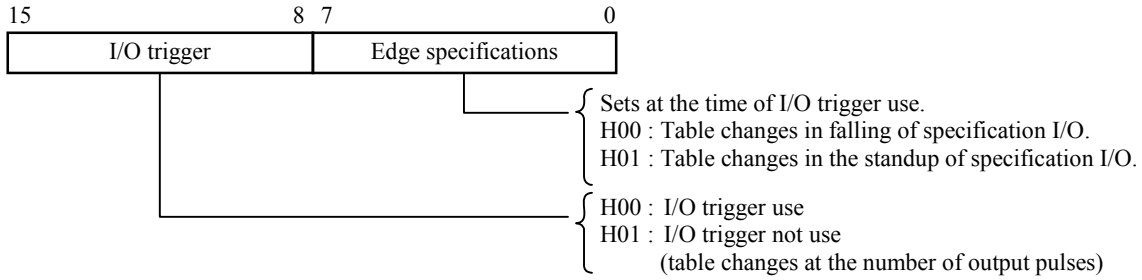
Function

[s+4n+2] Output frequency

Output frequency until the event specified by s+4n+3 occurs is set.

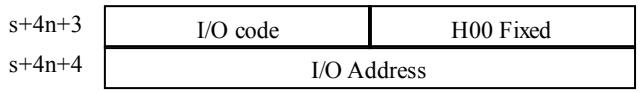
[s+4n+3] Event specification

The event which changes on the following table (or pulse output is suspended) is set.



[s+4n+3, s+4n+4] Event information

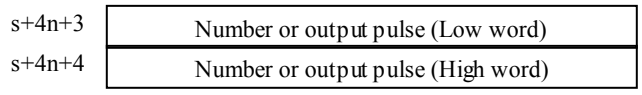
(1) I/O trigger use

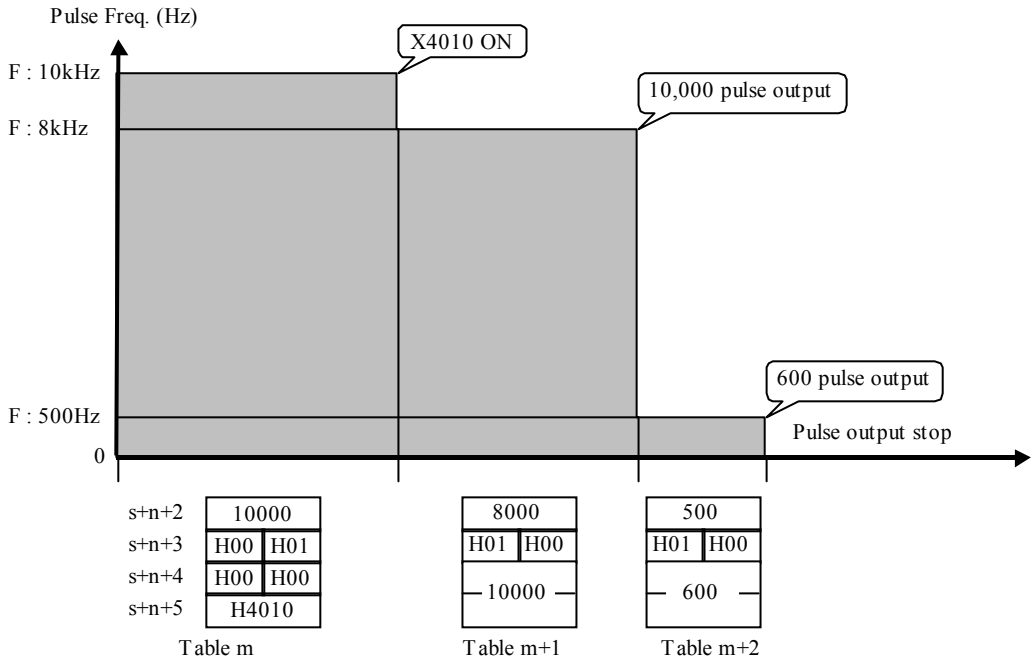


I/O which can be used	I/O code
X	H00
Y	H01
R	H02
M	H04

Ex.) When R6B0 is made into a trigger - s+4n+3 = H0200 (H02-I/O code), s+4n+4 = H06B0
When X4010 is made into a trigger - s+4n+3 = H0000 (H00-I/O code), s+4n+4 = H4010

(2) I/O trigger not use



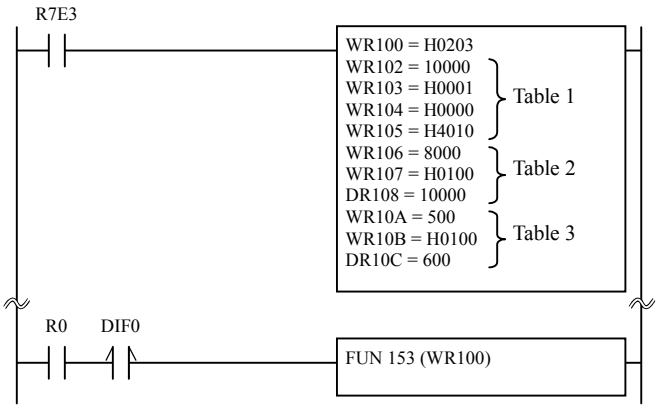


Name	Pulse output with sequence parameter change

Cautionary notes

- This instruction will not be executed if the specified pulse output is generating pulse output.
- If the output that corresponds to the specified pulse output number has not been set for pulse output, DER will be set to "1" and pulse output will not be generated.
- If the frequency are set to a value smaller than 10 Hz, the specified values will be changed to 10 Hz by the system.
- When the event which changes a table is made into an I/O trigger, the watch of "trigger I/O" is performed the constant cycle of 500 μs. Therefore, table changes are late for event generating for 500 μs(max.).

Program example




```
LD      R7E3
[
WR100 = H0203
WR102 = 10000
WR103 = H0001
WR104 = H0000
WR105 = H4010
WR106 = 8000
WR107 = H0100
DR108 = 10000
WR10A = 500
WR10B = H0100
DR10C = 600
]
LD      R0000
AND     DIF0
[
FUN 153 (WR100)
]
```

Program description

- When R0 turn on, pulse output starts with the parameter (frequency 10kHz) of a table 1.
- If the event (X4010 ON) registered into the table 1 occurs, a pulse output will change to the parameter (frequency 8kHz, number of output 10,000) of a table 2.
- If the event (the completion of output 10,000 pulse) registered into the table 2 occurs, a pulse output will change to the parameter (frequency 500Hz, number of output 600) of a table 3.
- A pulse output will be stopped if the event (the completion of output 600 pulse) registered into the table 3 occurs.

Chapter 9 Option board

MICRO64 supports optional communication or user program back up function as follows.

The function of option boards and supported software version of MICRO64 are shown in the following table.

Table 9.1 Option board list

No.	Type	Function	Supported CPU version *
1	EH-OBMEM	Backup of a user program and the special internal output for a setup of special function.	Ver.0101 ('04 / Aug. production) or later
2	EH-OB232	RS-232C serial communication port, Analog input 2ch	Ver.0101 ('04 / Aug. production) or later
3	EH-OB485	RS-422 / 485 serial communication port, Analog input 2ch	Ver.0100 ('04 / Jul. production) or later
4	EH-OBUSB	USB communication port	Ver.0101 ('04 / Aug. production) or later

* The software version of MICRO64 is stored in WRF050 and WRF051.
The software version shown in Table 9.1 is the value of WRF051.

[Notes]

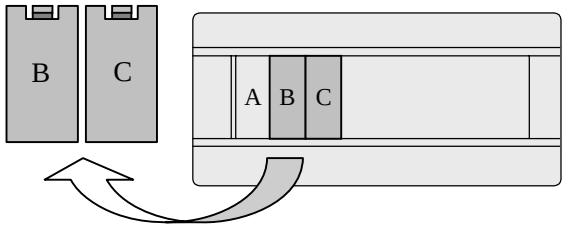
If unsupported option board is attached, error code is stored in the self-diagnostic error area (WRF000) of special internal output however, the error indication by O.K. / RUN LED is not performed. When you attach the option board and the following phenomenon occurs, please check the soft version of a basic unit.

- Communication error.
- The user program is not backed up.

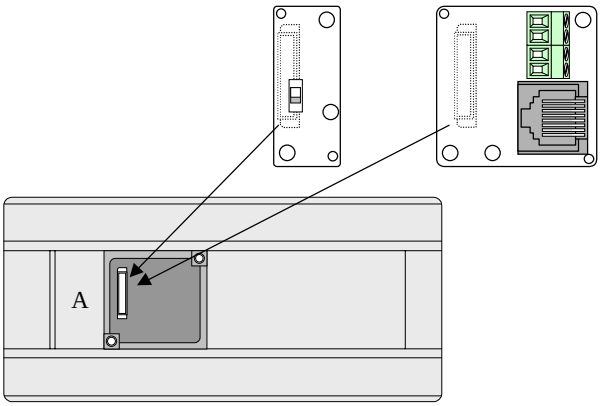
9.1 Mounting, Dismounting

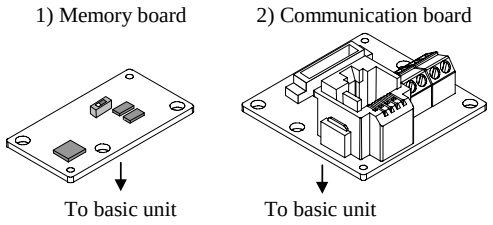
■ Mounting of option board

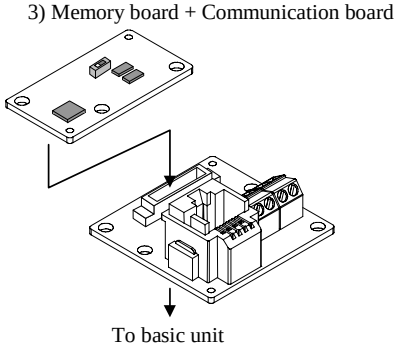
(1) Remove the cover B and C.



(2) Connect an option board as shown in this picture.

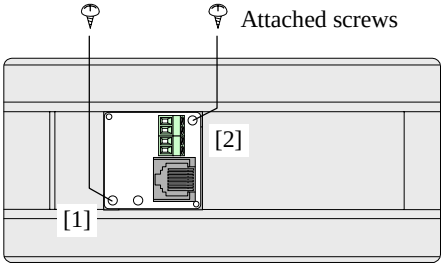






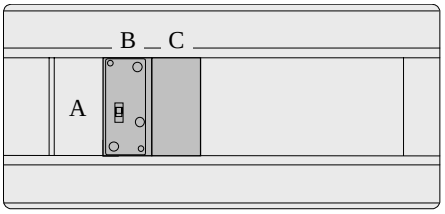
(3) Fix by attached screws.

EH-OBMEM is fixed by a screw, and other option boards are fixed by two screws.

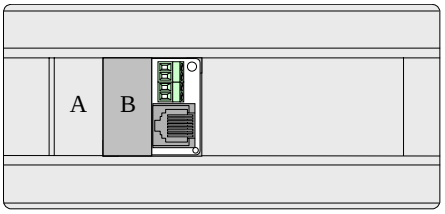


(4) Attach covers

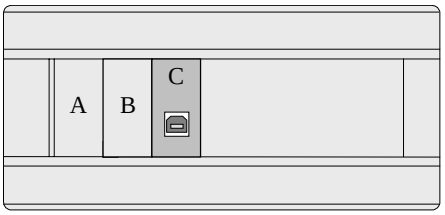
When only EH-OBMEM is installed, covers (B,C) can be attached.



In case of the other boards, only the cover B is attached.



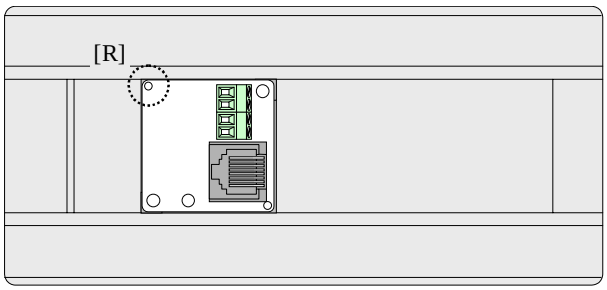
Attach the included plastic cover to C as shown below.



■ Dismounting of option board

After removing a screw, lift up the option board at the part [R] by a finger or small screw driver.

⚠ When you use a screw driver, be careful not to damage a PCB or parts.

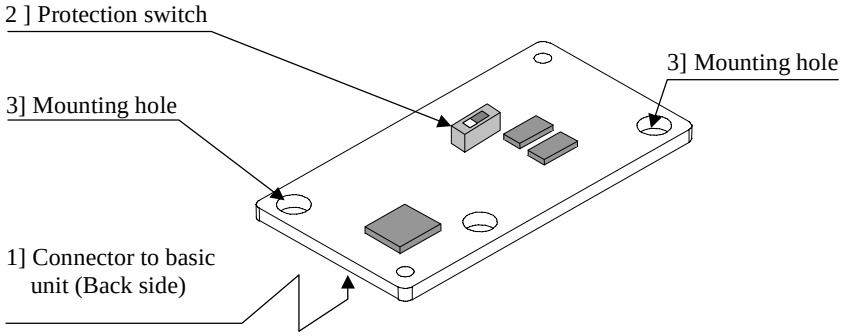


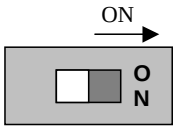
Attention on option board use

1. Mount of dismount without power supply.
2. Communication board can be attached one piece to one basic unit.
3. A communication board and a memory board can be used together however, dismount a memory board after reading / writing program because a memory board can not be fixed firmly.

9.2 Memory board

Type	EH-OBMEM
Weight	0. 01 kg



No.	Name	Details
1]	Connector to basic unit	Connector to basic unit (located at the back side)
2]	Protection switch	When the switch is on, the memory board is protected to be overwritten. 
3]	Mounting hole	Use M3 screw to fix

The function of the memory board is to save user program and data in special internal outputs. It is also possible to read out to PLC, which enables users to copy program (incl. data in special internal outputs) without programming software or peripheral devices.

[Notes]
- If the memory board is mounted or dismounted while power is activated, PLC could fail operation. Be sure to power off before attaching or detaching the memory board.
- If the power is down before writing is completed, data is not saved properly. Be sure to power off after checking if writing is completed. (Writing status is monitored in WRF062.)

(1) Writing (CPU ➔ Memory board)

- User program

  If program is downloaded from PC with memory board attached, user program is written to memory board.

- Data in special internal outputs

  Set special internal output flag “R7F6” to ON with memory board attached.

[Notes]

In case of online change in RUN, it takes 15 minutes at maximum because program processing is higher priority.

(2) Reading (Memory board ➔ CPU)

Both user program and data in special internal outputs are read out to PLC at powered up. OK LED blinks (100 ms ON / 100ms OFF) while reading. (Communication does not work while reading. CPU does not in RUN mode too.)

If read data is fault, OK LED blinks 3 times slowly (250 ms ON / 250ms OFF). Result code is stored in WRF062 also.

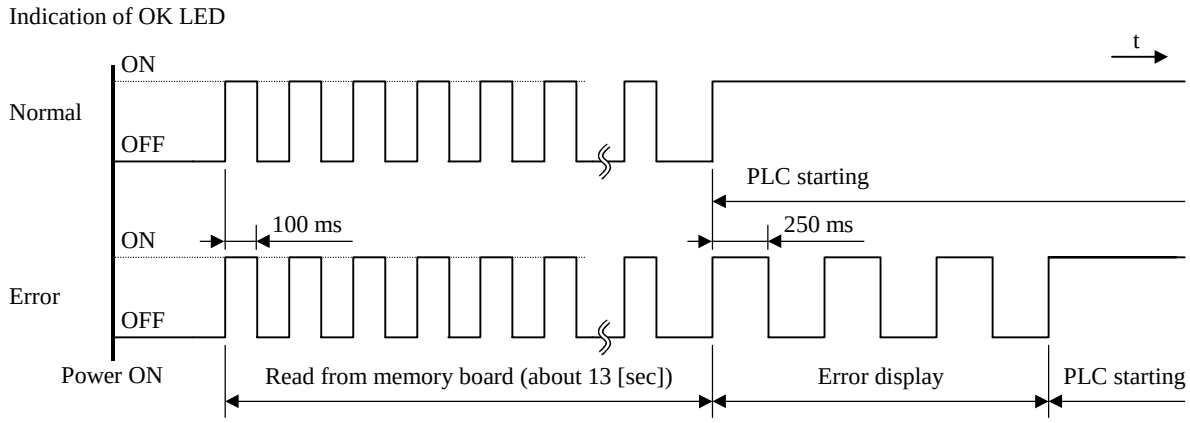


Figure 9.1 OK LED indication (In case of the memory board mount)

[Note]

If memory board is mounted, program and data in CPU are overwritten at powered up regardless of the contents or status. Be careful to use memory board to avoid deleting your program by mistake.

(3) Special internal output for memory board

3-1) WRF061 (Writing protection)

Besides protection switch, software protection is available.

Table 9.1 Setting values for writing protection

Status	WRF061	
	Set by user	Set by system
Writing protection	H8001	H0001
Cancel writing protection	H8000	H0000

3-2) WRF062 (Status information)

Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRF062 :	a	b	c	d	Not used				Error code							
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 9.2 Special internal output for setting port

Area	Status	Details
a	Memory board writing [W]	Set while memory board is being written. Reset by system at writing completed.
b	Writing error (*) [W]	Set if writing is failed.
c	User program error [R]	Set if user program read from memory board is fault.
d	Internal output values error [R]	Set if internal output read from memory board is fault.
Error Code	00 (no error)	If writing is completed properly, error code is 00.
	01 (timeout for writing)	If no response from memory board at writing, it will be timeout error.
	02 (software protected) (*)	If writing is attempted in case software protected, it will be writing error.

[W] : While writing [R] : While reading

* If hardware protection switch is enabled and writing is attempted, writing error is not detected although memory board is not actually written.

(4) The special internal output memorized on a memory board

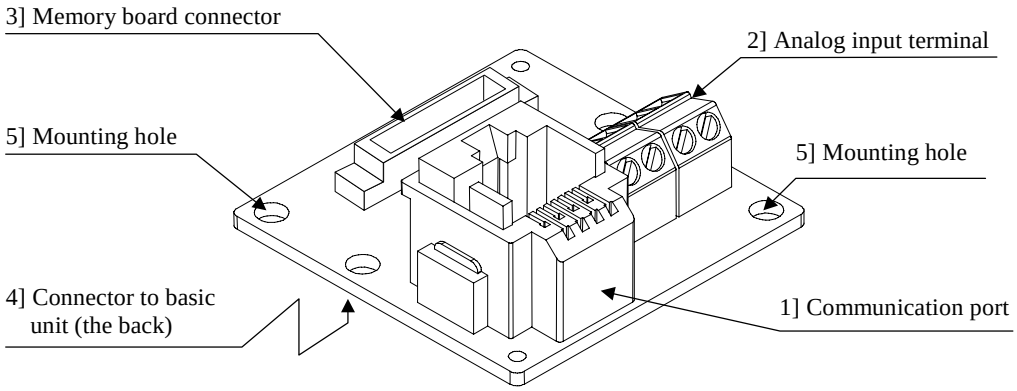
The special internal output memorized on a memory board is shown in the following table.

Table 9.2 Special internal output memorized on a memory board

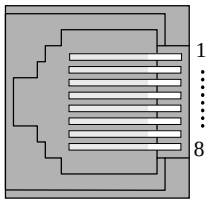
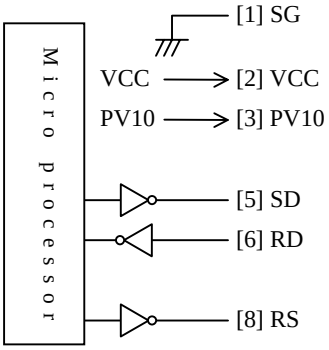
No.	Special internal output	Function
1	R7EE	Battery error display selection
2	WRF01A	Dedicated port 1 Communication settings
3	WRF03C	Dedicated port 1 Modem timeout time
4	WRF03D	Dedicated port 2 Communication settings
5	WRF06B	Pulse and PWM auto correction setting
6	WRF06C	Potentiometer 1 Filtering time
7	WRF06D	Potentiometer 2 Filtering time
8	WRF06E	Analog input type selection
9	WRF06F	Phase counting mode
10	WRF070	I/O operation mode
11	WRF071	I/O detailed function settings
12	WRF072	Gr1 On-preset value / Output frequency
13	WRF073	Gr2 On-preset value / Output frequency
14	WRF074	Gr3 On-preset value / Output frequency
15	WRF075	Gr4 On-preset value / Output frequency
16	WRF076	Gr1 Off-preset value / On-duty value
17	WRF077	Gr2 Off-preset value / On-duty value
18	WRF078	Gr3 Off-preset value / On-duty value
19	WRF079	Gr4 Off-preset value / On-duty value
20	WRF07A	Gr1 Pre-load value / Number of output pulse
21	WRF07B	Gr2 Pre-load value / Number of output pulse
22	WRF07C	Gr3 Pre-load value / Number of output pulse
23	WRF07D	Gr4 Pre-load value / Number of output pulse
24	WRF07E	Input edge
25	WRF07F	Input filtering time
26	WRF0B0	[Mode 2x] Gr1 On-preset value(Low word) / Output frequency(Low word)
27	WRF0B1	[Mode 2x] Gr1 On-preset value(High word) / Output frequency(High word)
28	WRF0B2	[Mode 2x] Gr2 On-preset value(Low word) / Output frequency(Low word)
29	WRF0B3	[Mode 2x] Gr2 On-preset value(High word) / Output frequency(High word)
30	WRF0B4	[Mode 2x] Gr3 On-preset value(Low word) / Output frequency(Low word)
31	WRF0B5	[Mode 2x] Gr3 On-preset value(High word) / Output frequency(High word)
32	WRF0B6	[Mode 2x] Gr4 On-preset value(Low word) / Output frequency(Low word)
33	WRF0B7	[Mode 2x] Gr4 On-preset value(High word) / Output frequency(High word)
34	WRF0B8	[Mode 2x] Gr1 Off-preset value(Low word) / On-duty value
35	WRF0B9	[Mode 2x] Gr1 Off-preset value(High word)
36	WRF0BA	[Mode 2x] Gr2 Off-preset value(Low word) / On-duty value
37	WRF0BB	[Mode 2x] Gr2 Off-preset value(High word)
38	WRF0BC	[Mode 2x] Gr3 Off-preset value(Low word) / On-duty value
39	WRF0BD	[Mode 2x] Gr3 Off-preset value(High word)
40	WRF0BE	[Mode 2x] Gr4 Off-preset value(Low word) / On-duty value
41	WRF0BF	[Mode 2x] Gr4 Off-preset value(High word)
42	WRF0C0	[Mode 2x] Gr1 Pre-load value(Low word) / Number of output pulse(Low word)
43	WRF0C1	[Mode 2x] Gr1 Pre-load value(High word) / Number of output pulse(High word)
44	WRF0C2	[Mode 2x] Gr2 Pre-load value(Low word) / Number of output pulse(Low word)
45	WRF0C3	[Mode 2x] Gr2 Pre-load value(High word) / Number of output pulse(High word)
46	WRF0C4	[Mode 2x] Gr3 Pre-load value(Low word) / Number of output pulse(Low word)
47	WRF0C5	[Mode 2x] Gr3 Pre-load value(High word) / Number of output pulse(High word)
48	WRF0C6	[Mode 2x] Gr4 Pre-load value(Low word) / Number of output pulse(Low word)
49	WRF0C7	[Mode 2x] Gr4 Pre-load value(High word) / Number of output pulse(High word)

9.3 RS-232C Communication board

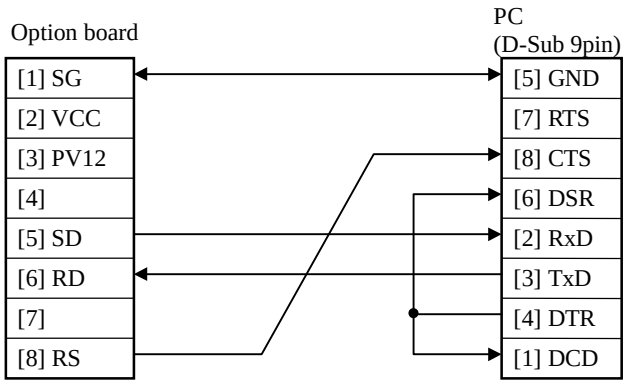
Type	EH-OB232
Weight	0.02 kg



No.	Name	Details
1]	Communication port	Communication port for programming tools or peripheral devices
2]	Analog input terminal	Input terminal for analog voltage signal Cable diameter : Single wire : 0.14 mm^2 to 1.5 mm^2 Strand wire : 0.14 mm^2 to 1.0 mm^2
3]	Memory board connector	Connector to memory board
4]	Connector to basic unit	Connector to basic unit (located at the back side)
5]	Mounting hole	Use M3 screw to fix

Terminal layout	No.	Signal	Meaning	Internal circuit
 Socket connector (Top view)	1	SG	Signal ground	
	2	VCC	5V DC output	
	3	PV10	10V DC output	
	4	N.C.	-	
	5	SD	Sent data	
	6	RD	Received data	
	7	N.C.	-	
	8	RS	Request to send	

[Cable diagram] (To RS-232C port of PC)



Standard RS-232C communication cable for the existing port on basic unit can be used with this option port too.

[Analog input]

Specification

Table. 9.3 Analog input specifications

No. of input	2 ch.
Internal output registers (ch.1 , ch. 2)	WRF03E , WRF03F
Input range	0-10V (10.24V max.)
Accuracy	±1%
Resolution	10 bits
Input impedance	100 kΩ
Isolation between channels	Not isolated
Isolation between CPU and analog signal	Not isolated

Analog input terminals are shown as below.

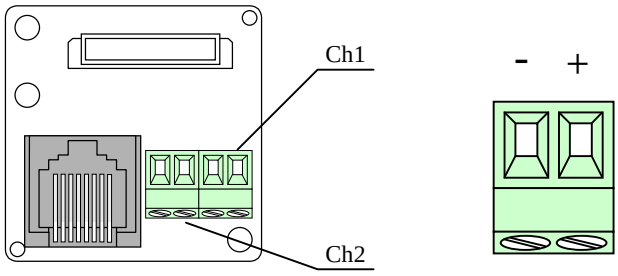


Figure. 9.3 Analog input terminals on option board

Converted analog input values are stored in internal outputs WRF03E and WRF03F (10-bit, 0 to H3FF)

WRF03E :	Analog input value for Ch.1
WRF03F :	Analog input value for Ch.2

Figure 9.4 Analog input values

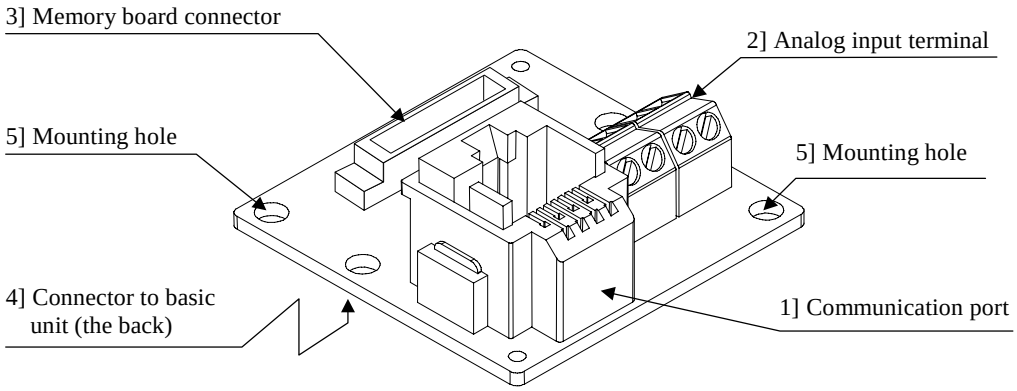
Analog input values could be unstable depending on environmental conditions. This can be reduced by setting sampling number as below. Averaged values will be stored in WRF03E and WRF03F based on sampling number. Possible sampling number is from 0 to 40 (0 to H28). If 0 is set, input values are not averaged. If 41 or larger number is set, it is regarded as 40.

WRF06C :	Sampling number for Ch.1
WRF06D :	Sampling number for Ch.2

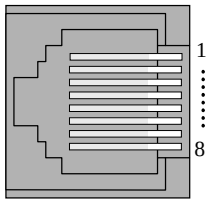
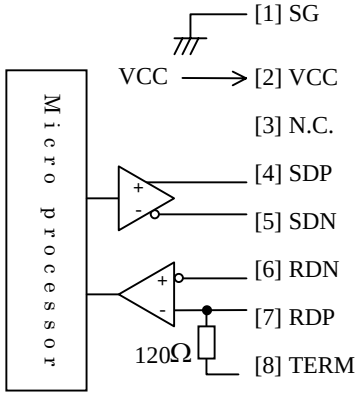
Figure. 9.5 Sampling number of analog input values

9.4 RS-422 / 485 Communication board

Type	EH-OB485
Weight	0.02 kg

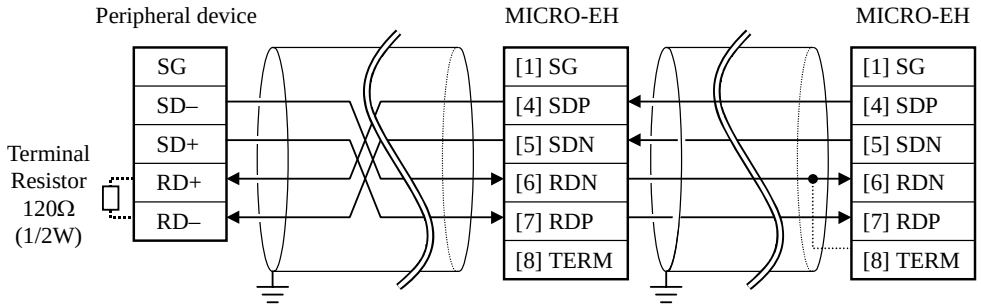


No.	Name	Details
1]	Communication port	Communication port for programming tools or peripheral devices
2]	Analog input terminal	Input terminal for analog voltage signal Cable diameter : Single wire : 0.14 mm^2 to 1.5 mm^2 Strand wire : 0.14 mm^2 to 1.0 mm^2
3]	Memory board connector	Connector to memory board
4]	Connector to basic unit	Connector to basic unit (located at the back side)
5]	Mounting hole	Use M3 screw to fix

Terminal layout	No.	Signal	Meaning	Internal circuit
 Socket connector (Top view)	1	SG	Signal ground	
	2	VCC	5V DC output	
	3	N.C.	Not used	
	4	SDP	Sent data +	
	5	SDN	Sent data -	
	6	RDN	Received data -	
	7	RDP	Received data +	
	8	TERM	Terminal resistor	

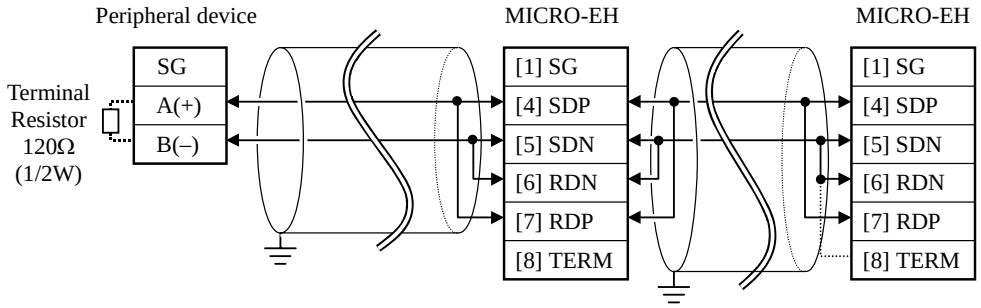
[Cable diagram]

(1) RS-422



Use a terminal resistor if necessary

(2) RS-485



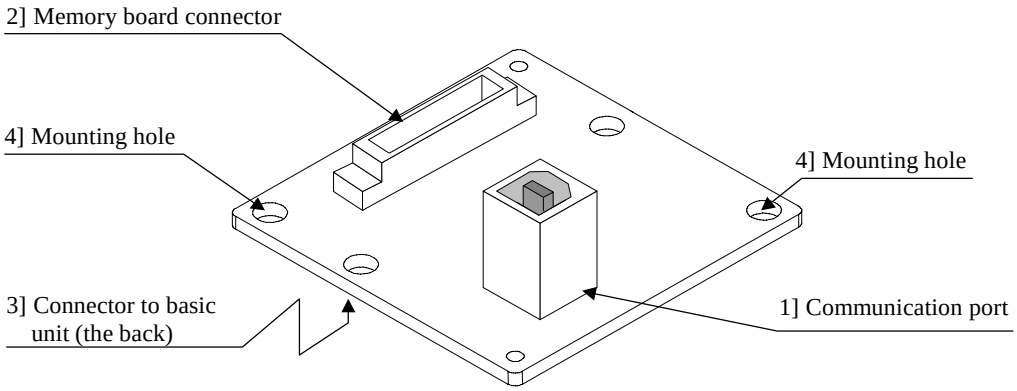
Use a terminal resistor if necessary

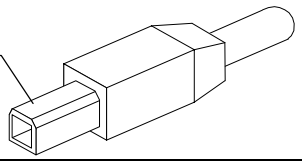
[Analog input]

Same as EH-OB232. Refer to the page of EH-OB232.

9.5 USB board

Type	EH-OBUSB
Weight	0.02 kg



No.	Name	Details
1]	Communication port USB (B-plut) connector	Communication port for programming tools or peripheral devices Connect USB B-plug. USB Cable (B plug) 
2]	Memory board connector	Connector to memory board
3]	Connector to basic unit	Connector to basic unit (located at the back side)
4]	Mounting hole	Use M3 screw to fix

Since this board is a converter from RS-232C to USB, the USB port of PC must be regarded as RS-232C port. For this reason, COM port driver is necessary for your PC. Please download the driver from following URL and install so that USB port works as serial port.

http://www.ftdichip.com/Drivers/FT232-FT245Drivers.htm

COM port number of programming software must be matched with COM port number configured in your PC.

[Note]

- USB cable is not included with EH-OBUSB.
- EH-OBUSB does not have analog input terminal. Special internal output for analog signal (WRF03E, WRF03F) will be undefined status when EH-OBUSB is installed.
- If EH-OBUSB is used in noisy enviroments, use a ferrite core with communication cable.