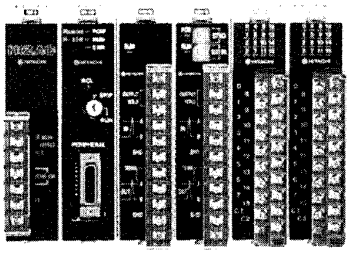
HITACHI

OPERATION MANUAL

H-200 SERIES



NJI 055(X)-2

USING THIS MANUAL

Introduction

This manual describes the H-200 Series Programmable Controller. This manual tells how to install, program, operate, and maintain your programmable controller.

Formore information on the HITACHI product line refer to the publications listed under additional information.

Manual Contents

- Chapter 1 - Configuration and Specifications.
- Chapter 2 - Compatibility
- Chapter 3 - Processing system
- Chapter 4 - Input/Output Numbers
- Chapter 5 - Programming
- Chapter 6 - Example of simple operation
- Chapter 7 - Installation
- Chapter 8 - Dubug and operation
- Chapter 9 - Maintenance
- Chapter 10 - Communication modules
- Chapter 11 - Advanced function modules

Signs used through the manual except noted

○ : Apllicable
× : Not applicable
△ : Partially applicable
— : Unapplicable

WARNING

To ensure that the equipment described by this manual. As well as all equipment connected to and used with it, operate satisfactorily and safety, 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 EMARGENCY 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 other referred to by it prior to installation and/or operation of the equipment. Hitachi Industrial Equipment Systems Co., 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 Industrial Equipment Systems Co., 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 ROULES, WHICH ARE DESIGNED TO PROVIDE PEASONABLE 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 Industrial Equipment Systems Co., Ltd. (Hitachi) warrants to the original purchaser that the programmable logic 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, alternation, 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 or 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 Dept.
Hitachi Industrial Equipment Systems Co., Ltd.
46-1 Ooaza-Tomioka Nakajo-machi
Kitakanbara-gun, Niigata-ken
959-2608 JAPAN

The Information and/or drawing set forth in this document and all right in and to inventions disclosed herein and patent which might be granted thereon disclosing or employing and the materials, methods, techniques or apparatus described herein are the exclusive property of Hitachi Industrial Equipment Systems Co., Ltd .

No copies of the information or drawings shall be made without the prior constant of Hitachi Industrial Equipment Systems Co., Ltd.

Hitachi Industrial Equipment Systems Co., Ltd. provides customer assistance in varied technical areas. Since Hitachi does not possess 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 infringement 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 Industrial Equipment Systems Co., Ltd. Incorporated constantly strives to improve all its products, we reserve the right to make changes to equipment and/or manual at any time without notice and without incurring any obligation other than as noted in this manual.

Hitachi Industrial Equipment Systems Co., Ltd. assumes no responsibility for errors that may appear in this manual.

As the product works with user program, and Hitachi Industrial Equipment Systems Co., 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 Industrial Equipment Systems Co., Ltd. or its representative. Hitachi will try to find the reason as much as possible and inform the countermeasure when obtained.

Nevertheless Hitachi Industrial Equipment Systems Co., 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 have to prepare with the countermeasure 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.

TABLE OF CONTENTS

1. CONFIGURATION AND SPECIFICATIONS

1. 1 System Configuration

Figure 1-1-1 shows system configuration

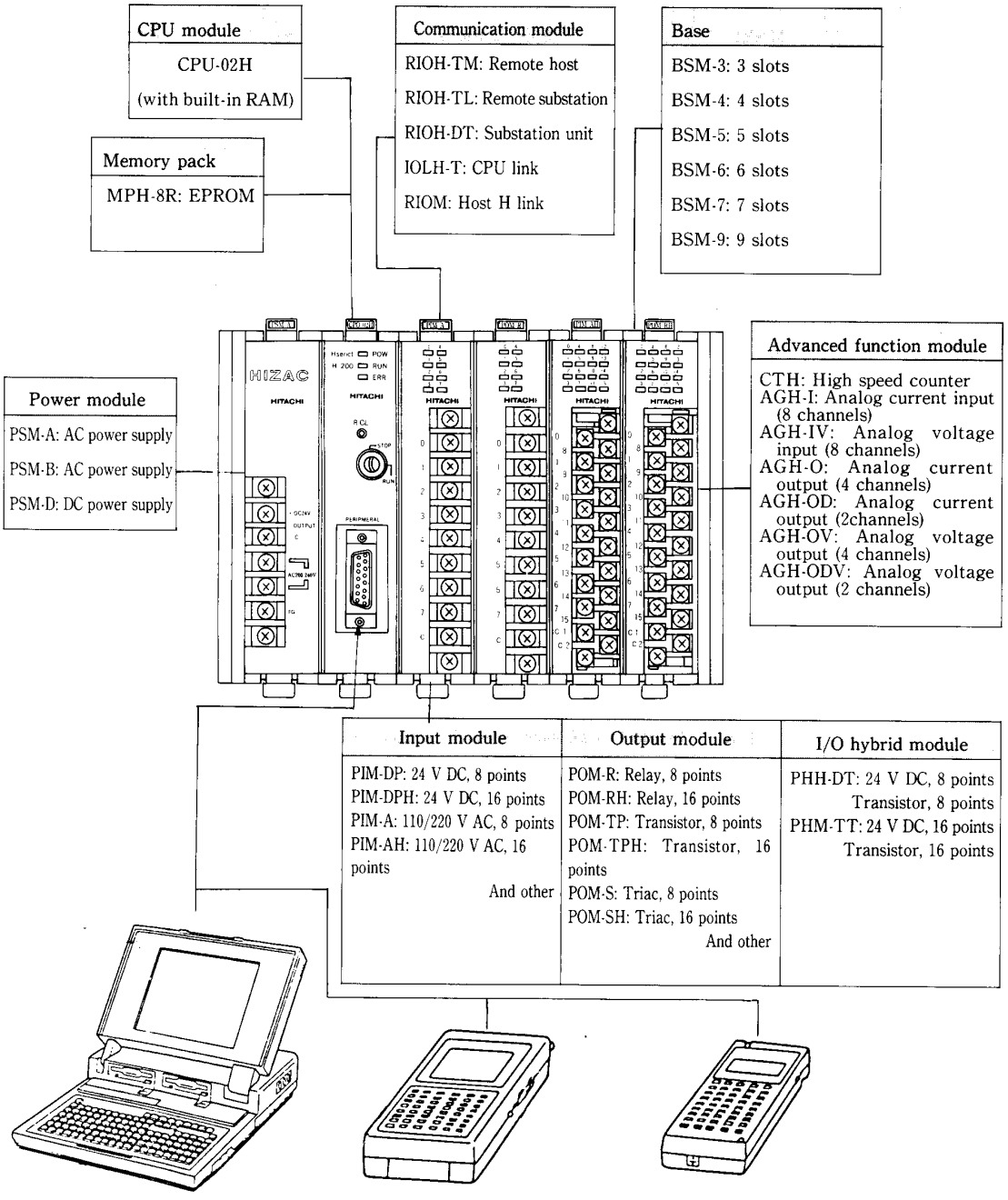


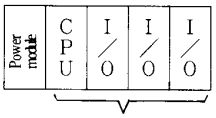
Fig. 1-1-1 System Configuration

1. 2 Component devices

(1) Basic component

Table 1-2-1 lists the modules.

Table 1-2-1 List of Modules (1/2)

	Item		Model	Specifications	No. of points used	Remarks
☆	CPU	H-200	CPU-02H	I/O 256 points max. (when using 16-point I/O)	—	
	Memory	RAM	——	Program capacity 7.6k steps (built-in)	—	Built in CPU module
☆		EPROM	MPH-8R	Program capacity 7.6k steps	—	
	Base		BSM-3	3 slots (Example)  4 slots 4-slot base mounts 4 modules besides power module.	—	○ Base is commonly used for basic and expanded configurations. ○ For 11 to 16 slots, the BSM-9 is requered as the base of basic unit (may be called "basic base" hereafter).
			BSM-4	4 slots		
			BSM-5	5 slots		
			BSM-6	6 slots		
			BSM-7	7 slots		
			BSM-9	9 slots		
	Power module		PSM-A	Power supply voltage 110/220 V AC	—	
			PSM-B	Power supply voltage 110/220 V AC, with increased output capacity		
			PSM-D	Power supply voltage 24 V DC		
	Input module	AC input module	PIM-A	8-point 110/220 V AC input	16	
			PIM-AH	16-point 110/220 V AC input	16	
			PIM-AW	16-point 110/220 V AC input [detachable terminal block]	16	
		DC input module (negative logic)	PIM-D	8-point 24 V DC input	16	
			PIM-DH	16-point 24 V DC input	16	
			PIM-DW	16-point 24 V DC input [detachable terminal block]	16	
		DC input module (positive logic)	PIM-DP	8-point 24 V DC input, source type (Note 2)	16	
			PIM-DPH	16-point 24 V DC input, source type (Note 2)	16	
			PIM-DPW	16-point 24 V DC input, source type [detachable terminal block]	16	
	Output module	Relay output module	POM-R	8-point relay output	16	
			POM-RC	8-point relay output, independent contact	16	
			POM-RH	16-point relay output	16	
			POM-RW	16-point relay output [detachable terminal block]	16	
		Triac output module	POM-S	8-point triac output	16	
			POM-SH	16-point triac output	16	
			POM-SW	16-point triac output [detachable terminal block]	16	
		Transistor output module (negative logic)	POM-T	8-point transistor output	16	
			POM-TH	16-point transistor output	16	
			POM-TW	16-point transistor output [detachable terminal block]	16	

Notes:
1. The modules marked ☆ are exclusive for H-200.
   Other modules are commonly used with EM.
2. The source type corresponds to positive logic input.

Table 1-2-1 List of Modules (2/2)

	Item		Model	Specifications	No. of points used	Remarks
	Output module	Transistor output module (positive logic)	P O M-T P	8-point transistor output, source type (Note 2)	16	
			P O M-T P H	16-point transistor output, source type (Note 2)	16	
			P O M-T P W	16-point transistor output, source type [detachable terminal block]	16	
☆	I/O mixed module		P H H-D T	DC input 8 points, transistor output 8 points	32	
			P H M-T T	TTL input 16 points, TTL output 16 points	32	Connector connection
	Communication module	CPU link	R I O M	Host link (coaxial cable)	32	Connection with high-order model of H series
☆			I O L H-T	Parallel link (twisted pair cables)	—	CPU link between H-200 PLCs
☆		Remote I/O	R I O H-T M	Remote host (twisted pair cables)	128	Remote I/O between H-200 PLCs
☆			R I O H-T L	Remote substation (twisted pair cables)	Variable	
☆			R I O H-D T	Substation unit, I/O 32 points	32	
☆	Advanced function module	High speed counter	C T H	High speed counter input, 10 kHz max.	32 6W	
☆		Analog input module	A G H-I	8-channel, 4 to 20 mA, 8-bit currrent input	8W	
☆			A G H-I V	8-channel, 0 to 10 V, 8-bit voltage input	8W	
☆		Analog output module	A G H-O	4-channel, 4 to 20 mA, 8-bit current output	8W	
☆			A G H-O D	2-channel, 4 to 20 mA, 8-bit current output	8W	
☆			A G H-O V	4-channel, 0 to 10 V, 8 bit voltage output	8W	
☆			A G H-O D V	2-channel, 0 to 10 V, 8 bit voltage output	8W	

Notes:
1. The modules marked ☆ are exclusive for H-200.
   Other modules are commonly used with H-100 M/EM.
2. The source type corresponds to positive logic input.

(2) Peripherals

Table 1-2-2 lists peripherals.

Table 1-2-2 List of peripherals

Item		Model	Specifications	Remarks
Portable graphic programmer	Programmer mainframe	P G M-G P H	With EL backlight Audio CMT I/F function	
	Option I/F	P G M I F1H	ROM writer function Printer I/F function	
Instruction word programmer		P G M-C H H	Display with LED backlight Audio CMT I/F incorporated	

Note: These peripherals are commonly usable for H-300, 700 and 2000.

(3) Connecting cable

Table 1-2-3 lists the connecting cables.

Table 1-2-3 Connecting Cables

Application	Model	Specification	Remarks
Expansion signal cable (between basic base and expansion base)	C N M-06	0.6 m long	Exclusive for H-200
Cable for portable graphic programmer and instruction word programmer	P G C B02H	2 m long between CPU and programmer	Commonly usable for H-300, 700 and 2000
	P G C B05H	5 m long between CPU and programmer	
Cable for graphic input device	G P C B02H	2 m long between CPU and graphic input device	
	G P C B05H	5 m long between CPU and graphic input device	
	G P C B15H	15 m long between CPU and graphic input device	
	C B P G B	2 m long between graphic input device and printer	

(4) Other

Model	Item	Remarks
L I B A T-H	Lithium battery	Commonly usable for H-300, 700 and 2000
C V M	Empty slot cover	Exclusive for H-200

1. 3 Names of Unit Components

Fig. 1-3-1 shows the name of each H-200 part, exemplifying the 5-slot base.

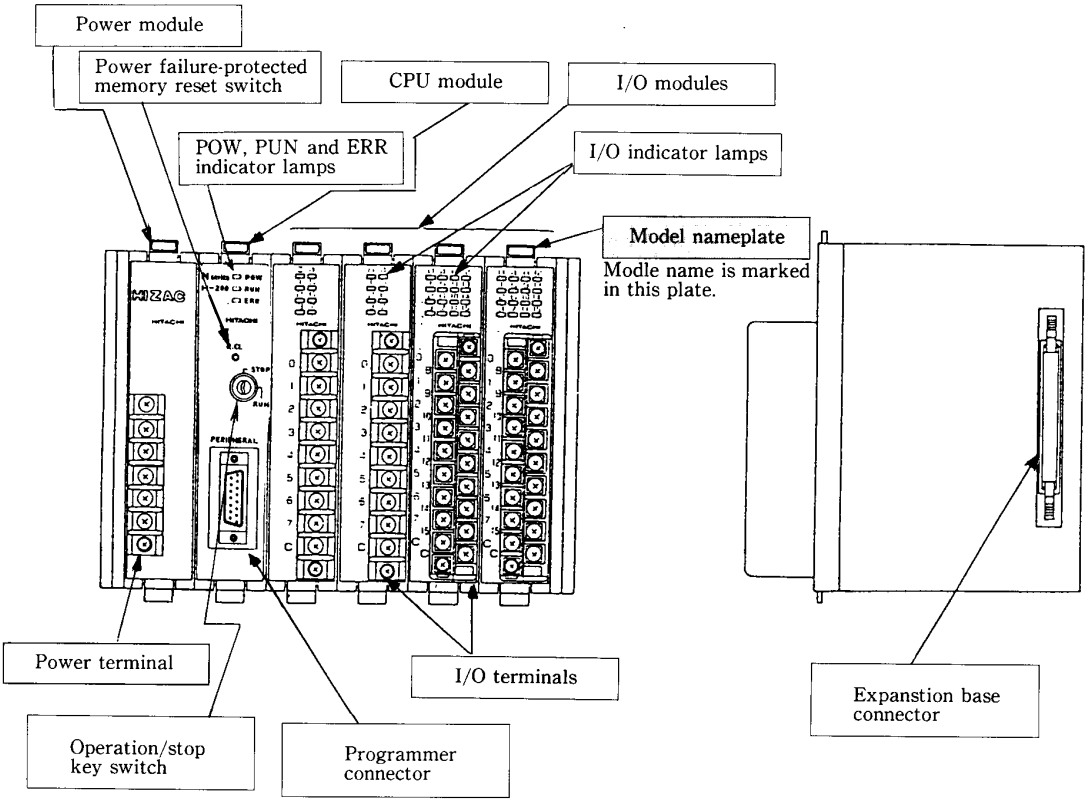


Fig. 1.3.1 Name of Each Part

The number of slots is different among the 3-slot, 4-slot, 6-slot, 7-slot and 9-slot bases. Base is commonly usable between basic and expansion units. For expansion, an I/O module is to be mounted in the slot for CPU module.

1. 4 General Specifications

Table 1-4-1 lists the common specifications among the modules used for H-200 system.

Table 1-4-1 General Specification

Item	Specifications
Operating temperature	0 to 55℃
Storage temperature	−10 to 75℃ (Retention of the memory contents is assured only within the operating temperature range.)
Operating humidity	20 to 90% RH (non-condensing)
Storage humidity	10 to 90% RH (non-condensing)
Allowable instantaneus power failure time	20 ms
Vibration resistance	Frequency 16.7 Hz, multi-amplitude 3 mm in X, Y and Z directions
Noise resistance	○1,500 Vp-p noise voltage in 100ns and 1μs noise pulse width (by using a noise simulator) ○Based on NEMA ICS2-230-42 to 45 (except for input) ○Static noise 3,000 V applied to exposed metal
Insulation resistance	20 MΩ or more between external AC terminal and FG (frame ground) terminal (by using a 500 V DC megger) (Note 1)
Dielectric strength	1,500 V AC for 1 minute between external AC terminal and FG terminal (Note 1)
Grounding	100 Ω max. (exclusive grounding required)
Atmosphere	Must be free from corrosive gases such as ammonia, hydrogen sulfide and sulfur dioxide, from excessive dust.
Structure	Open wall mounting (in only one direction)
Cooling	Natural air-cooling

Notes:

1. A varistor for suppressing lightning surge is connected to the power supply terminal of power module. Therefore, the connector inside the power module must be separated when testing insulation resistance or dielectric strength of the power supply terminal. Otherwise, the varistor might be damaged.
2. For peripherals, refer to their manuals (including instruction manuals).

1. 5 Basic Specifications

Table 1-5-1 lists the basic specifications.

Table 1-5-1 Basic Specifications

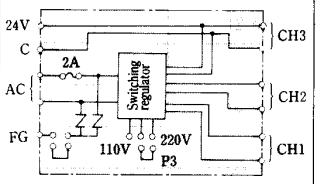
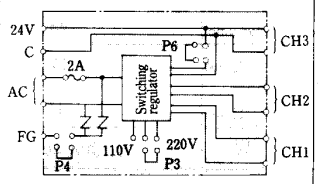
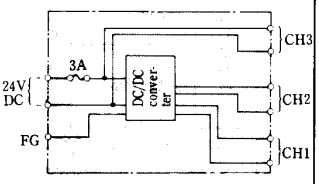
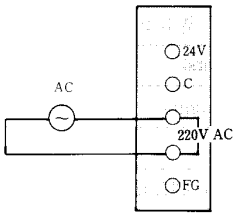
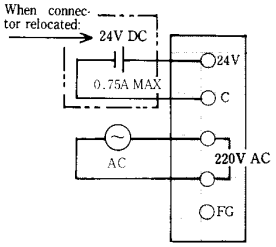
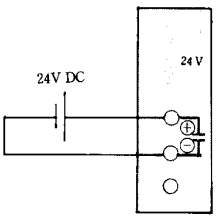
Item				H-200	H-300 (reference)
Control specifications	Max. slots usable			16 (Note 1)	9
	No. of I/O points	8-point I/O module		Up to 128	—
		16-point I/O module		Up to 256	—
		32-point I/O module		Up to 512 (PHM-TT)	Up to 288
		64-point I/O module		—	Up to 576
	Instruction word/ladder diagram	Processing system		Stored program cyclic processing	
		Processing speed	Sequence instructions	1.5 μs/instruction	1.1 to 2.5 μs/instruction
			Arithmetic and application instructions	Several ten μs to several thousand μs/instruction	Several μs to several thousand μs/instruction
		User program memory		Up to 7.6k steps (in both RAM and EPROM)	Up to 7.6K steps (in both RAM and EPROM)
		Instruction word	Sequence instructions	13 kinds	13 kinds
			Arithmetic and application instructions	44 kinds (Note 2)	69 kinds
		Ladder diagram	Sequence instructions	17 kinds	17 kinds
			Arithmetic and application instructions	44 kinds (Note 2)	69 kinds
	HI-FLOW			— (not supported)	Possible
I/O processing specifications	I/O processing system			Direct processing	
	Internal output	Bit		1,984 points (R0 ~ R7BF)	
		Word		1 k words (WR0 ~ WR3FF)	
		Special	Bit	64 points (R7C0 ~ R7FF)	
			Word	64 words (WRF000 ~ WRF03F)	
		CPU link		128 points × 2 (256 points)	1,024 words/loop
		Remote I/O		128 points/host × 4 (512 points)	512 points/32 words × 4 boards
		Bits/words shared		4,092 points/256 words	16,384 points/1,024 words
	Timer, conuter	No. of points		512 points (TD + CU) (0 to 255 points for timer)	
		Timer setup value		0 to 65, 535 sec with time base of 0.01, 0.1 sec and 1 sec	
		Counter setup value		1 to 65, 535 counts	
	Edge detection			Rising 128 points, falling 128 points	Rising 512 points, falling 512 points
	Calendar clock			Year, month, day, day of week, hour, minute and second	—
Peripheral functions	Programming system			By instruction words and ladder diagram	By instruction words, ladder diagram, HI-FLOW and BASIC
	Peripheral devices			Instruntion word programmer, portable graphic programmer	

Notes:
1. The maximum slots usable are only 10 when using the base for EM. For 11 to 16 slots, the basic base must be BSM-9.
2. Signed instructions, and MOV, COPY, FOR to NEXT, RSRV and FREE instructions are unavailable. Besides, double-word calculation is impossible.

1. 6 Specifications of Power Module

Table 1-6-1 lists the specifications of power module.

Table 1-6-1 Specifications of Power Module

Item \ Model		PSM-A	PSM-B	PSM-D
Power supply voltage	Rated voltage	100 V/110 V/120 V AC, 200 V/220 V/240 V AC (110 V AC and 220 V AC switchable with connentor)		24 V DC
	Permissible fluctuation	85 V~132 V AC, 170 V~264 V AC		19.2~30 V DC
Frequency	Rated frequency	50/60Hz		—
	Permissible fluctuation	47~63Hz		—
Input current		0.6 A or less		1.6 A or less
Output current	CH1 (5 V)	1 A (for CPU and programmer)	1.7 A (for CPU and programmer)	1 A (for CPU and programmer)
	CH2 (24V)	0.3 A (for output module)	0.5 A (for output module)	0.3 A (for output module)
	CH3 (24V)	0.45A (for input module)	0.25A (for input module)	1 A (for input module)
Circuit diagram				
External wiring				

(1) Power module receives AC or DC power and supplies the determined power to the CPU, programmer and I/O modules. Output is composed of 3 channels; CH1 (5 V) for CPU and programmer, CH2 (24 V) for output module and CH3 (24 V) for input module.
The maximum output current in each channel is specified in the table above. The current consumption of each module is marked in the specifications following. So configure a system so that total current consumption in each channel does not exceed the maximum current output.

(2) The PSM-A and PSM-B select either 110 V AC or 220 V AC by means of a connector.
Since they have been factory-set at 220 V AC, the connecton need be changed over to 110 V when necessary, and then attach the furnished voltage nameplate.

(3) The PSM-A and PSM-B incorporate a varistor for suppressing lightning surge. Therefore, the connector must be separated before testing dielectric strength or insulation resistance.
Otherwise, the varistor might be damaged.

(4) CH3 is also used by sensor. Confirm that total current consumption does not exceed 0.45 A with the PSM-A, and 0.25 A with the PSM-B.

Point

The PSM-B allows power supply from an external power unit to CH3 through relocation of the connector.
Utilize this feature in case CH3 current is inadequate because there are many input modules to be connected.

(5) Table 1-6-2 lists the current consumptions of CPU module and programmers.

Table 1-6-2 Current consumptions of CPU and programmers

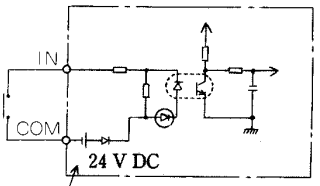
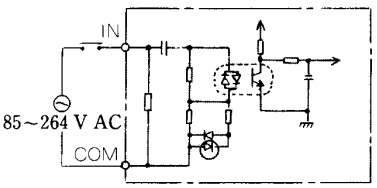
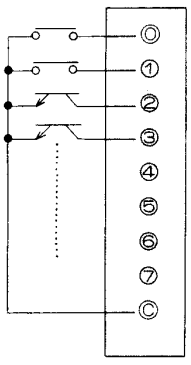
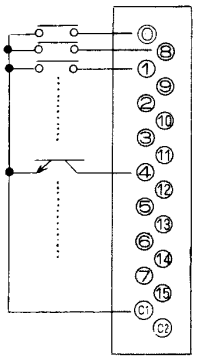
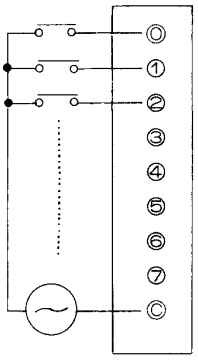
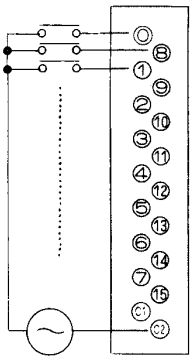
Item	Model	CH1 (5V)	CH2 (24V)	CH3 (24V)
CPU module	CPU-02H	150 mA	5 mA	0 mA
Portable graphic programmer	PGM-GPH	400 mA	0 mA	0 mA
Option I/F	PGMIF 1H	700 mA	0 mA	0 mA
Instruction word programmer	PGM-CHH	450 mA	0 mA	0 mA

1. 7 Specifications of I/O Modules

(1) Specifications of Input Module

Table 1-7-1 lists the specifications of input module.

Table 1-7-1 Specifications of Input Module

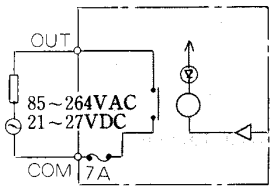
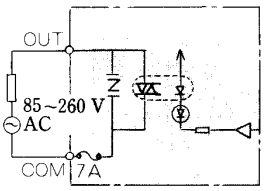
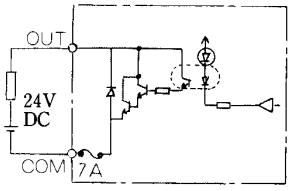
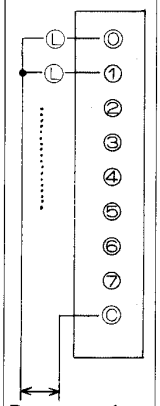
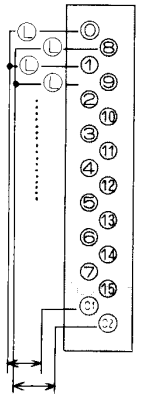
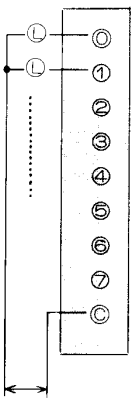
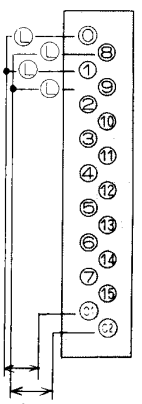
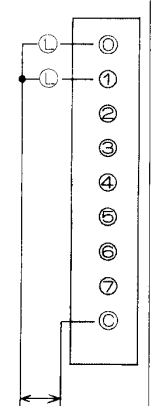
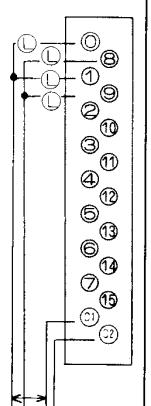
Item \ Model		PIM-D	PIM-DH, DW	PIM-A	PIM-AH, AW
Input specification		DC input		AC input	
Nominal voltage		24 V DC		110V/220 V AC	
Input voltage		21.6 DC~26 V DC		85~264 V AC, 50/60 Hz	
Input current		9 mA (when input and common terminals are shortcircuted)		7 mA (110VAC, 50Hz)	
Operational specification	ON	19 V or more (resistance 300 Ω or less)		85 V AC or more	
	OFF	7 V or less (resistance 200 kΩ or more)		30 V AC or less	
Input delay time	ON→OFF	4 ms or less		16 ms or less	
	OFF→ON	4 ms or less		16 ms or less	
No. of input points		8 inputs/module	16 inputs/module	8 inputs/module	16 inputs/module
Common input connection		8 inputs/common tenminal		8 inputs/common terminal	
Polarity		Common terminal (−)		—	
Isolation method		Photocoupler		Photocoupler	
Current consumption (average) (Note 1)	CH 1	0.5 mA + (No. of input ON points) × 0.5 mA		1 mA	
	CH 2	0 mA		0 mA	
	CH 3	(No. of input ON points) × 9 mA		0 mA	
Circuit diagram		 Power supply incorporated			
External wiring			 ○C1 and C2 are internally connected.		 ○C1 and C2 are internally connected.

(Note 1) The power consumption of each module is entered above. The total current consumption of each channel must not exceed the maximum output current of the power module. For output current capacity of the power module, refer to "1.6 Specifications of Power Module."

(2) Specifications of Output Module.

Table 1-7-2 lists the specifications of output module.

Table 1-7-2 Specifications of Output Module

Item \ Model	POM-R	POM-RH, RW	POM-S	POM-SH, SW	POM-T	POM-TH, TW
Output specification	Relay output		Triac output		Transistor output	
Nominal voltage	110 V/220 V AC		110/220 V AC		24 V DC	
Output voltage	85 ~ 264 V AC		85 ~ 264 V AC		5 ~ 27 V DC	
Max. load current: 1 circuit	2 A		1 A		0.5 A	
Max. load current: 8 circuits	4 A		4 A		2.5 A(Note)	
Min. load current	10 mA (5V DC)		50 mA		10 mA (24 V DC)	
Max. leakage cuttent	--		1 mA (220 V AC)		0.1 mA (24 V DC)	
Max. rush cuttent	6 A(100 ms)		20 A(20 ms)		3 A(20 ms)	
Max. output delay time: ON→OFF	10 ms		11 ms		1 ms	
Max. output delay time: OFF→ON	10 ms		11 ms		1 ms	
No. of output points	8 points	16 points	8 points	16 points	8 points	16 points
Common output connection	8 points/common terminal		8 points/common terminal		8 points/common terminal	
Polarity	—		—		Common terminal (−)	
Isolation method	Relay		Photocoupler		Photocoupler	
Current consumption (average): CH1	0.2 mA + (No. of output ON points) × 0.2 mA		0.3 mA + (No. of output ON points) × 0.2 mA		0.2 mA + (No. of output ON points) × 0.2 mA	
Current consumption (average): CH2	(No. of output ON points) × 10 mA		(No. of output ON points) × 6.5 mA		(No. of output ON points) × 6.5 mA	
Current consumption (average): CH3	0 mA		0 mA		0 mA	
Circuit diagram						
External wiring						

Note: As four-element transistor devices are used, max. load current is limited in blocks of terminals No. 0 to 3, 4 to 7, 8 to 11 and 12 to 15. Transistor output must be 1.25 A/4 circuits at maximum.

(3) Specifications of positive logic (source type) I/O modules

Tables 1-7-3 and 1-7-4 list the specifications of source type output (PNP transistor) module and source type input module.

Table 1-7-3 Specifications of Positive Logic Input

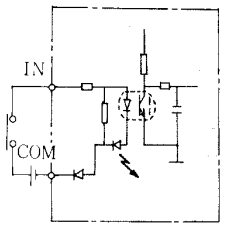
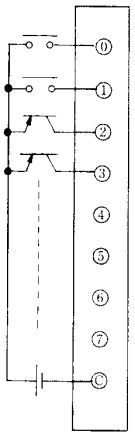
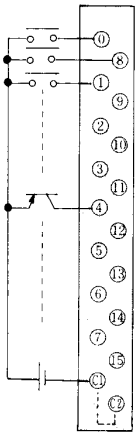
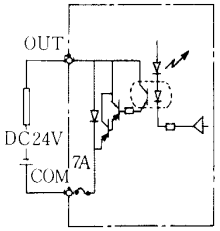
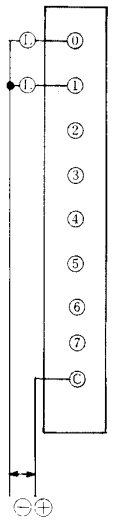
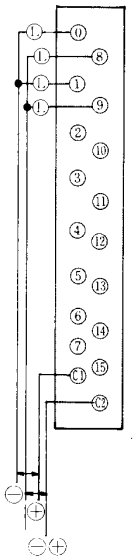
Item \ Model		PIM-DP	PIM-DPH, DPW
Input specification		DC input	
Nominal voltage		24 V DC	
Input voltage		21.6 ~ 26 V DC	
Input current		Approx. 9 mA/24 V DC (impedance approx. 2.7 kΩ)	
Operating voltage	ON	19 V or more (resistance 300 Ω or less)	
	OFF	7 V or less (resistance 200 kΩ or more)	
Max. input delay time	ON→OFF	4 m sec	
	OFF→ON	4 m sec	
No. of input points		8 points	16 points
Common input connection		8 points/common terminal	
Polarity		Common terminal (−)	
Isolation method		Photocoupler	
Current consumption (average)	CH1	0.5mA + (No. of input ON points) × 0.5mA	
	CH2	0 mA	
	CH3	0 mA	
Circuit diagram			
External wiring			 C1 and C2 are internally connected.

Table 1-7-4 Specifications of Positive Logic Output

Item \ Model		POM-TP	POM-TPH, TPW
Output specification		Transistor output	
Nominal voltage		24 V DC	
Output voltage		3 ~ 26 V DC	
Max. load current	1 circuit	0.5 A	
	4 circuits	1.25 A	
	8 circuits	—	
Min. load current		10 mA (24 V DC)	
Max. leakage current		0.1 mA (24 V DC)	
Max. rush current		3 A (20 m sec)	
Max. output delay time	ON→OFF	1 m sec	
	OFF→ON	1 m sec	
No. of output points		8 points	16points
Common output connection		8 points/common terminal	
Polarity		Common terminal (+)	
Isolation method		Photocoupler	
Current consumption (average)	CH1	0.2mA + (No. of output ON points) × 0.2mA	
	CH2	(No. of output ON points) × 0.65mA	
	CH3	0 mA	
Circuit diagram			
External wiring		 DC power supply	 DC power supply

(4) Specifications of hybrid modules

Tables 1-7-5 and 1-7-6 list the specifications of I/O mixed module and TTL I/O mixed module, respectively.

Table 1-7-5 Specifications of I/O Mixed Module

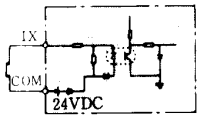
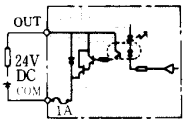
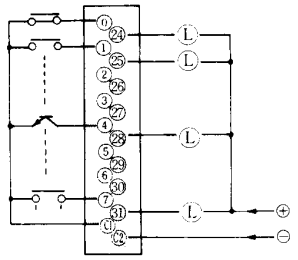
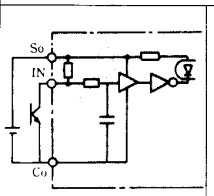
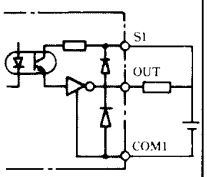
Item \ Model		PHH-DT	
I/O specification		DC input	Transistor output
Nominal voltage		24 V DC	24 V DC
Permissible voltage range		21.6 ~ 26 V DC	5 ~ 27 V DC
Input current		9 mA	—
Operational specification	ON	ON resistance 300 Ω max.	—
	OFF	OFF resistance 200 kΩ min.	—
Max. load current	1 circuit	—	0.5 A
	4 circuits	—	1.25 A
	8 circuits	—	2.5 A
Max. leakage current		—	0.1mA (24 V DC)
Max. rush current		—	3 A (20 m s)
Max. delay time	ON→OFF	4 ms	1 ms
	OFF→ON	4 ms	1 ms
No. of I/O points		8 points (0 to 7)	8 points (8 to 15)
Common terminal		8 I/O points/common terminal	8 I/O points/common terminal
Polarity		Common terminal ⊖	Common terminal ⊖
Isolation method		Photocoupler	Photocoupler
Current consumption (average)	CH 1	(No. of ON inputs) × 9 mA + (No. of ON outputs) × 8 mA	
	CH 2	0 mA	0 mA
	CH 3	(No. of ON inputs) × 9 mA	0 mA
Circuit diagram			
External wiring			

Table 1-7-6 Specifications of TTL I/O Mixed Module

Item \ Model		PHM-TT	
I/O specification		TTL input	TTL output (open collector)
I/O voltage		4 V ~ 27 V DC	4 V ~ 27 V DC
Input current		6 mA (5 V DC)	—
Input voltage	ON	1.5 V max. (5 V DC)	—
	OFF	3.5 V min. (5 V DC)	—
Max. load current		—	20mA / point
Max. leakage current		—	50μ A
Max. delay time	ON→OFF	1 m s	1 m s
	OFF→ON	1 m s	1 m s
No. of I/O points		16 points/module	16 points/module
Common terminal		16 points/common terminal	8 points/common terminal
Polarity		Common ··· ⊖	Common ··· ⊖
Isolation method		Photocoupler	Photocoupler
I/O indication		None	None
Current consumption (average)	CH 1	(No. of ON outputs) × 5 mA + 30mA	
	CH 2	0 mA	
	CH 3	0 mA	
Circuit diagram			
External wiring		Pin layout of 40-pin flat cable	
Connector for external wiring		Exclusive connector (made by Hirose Denki) Socket: HIF3C-40D-2.45C	

Pin layout of 40-pin flat cable

PinNo.	Signal	PinNo.	Signal	PinNo.	Signal	PinNo.	Signal
1	COM0	21	NC	2	COM1	22	COM2
3	S 0	23	NC	4	S 1	24	S 2
5	IN0	25	IN8	6	OUT16	26	OUT24
7	1	27	9	8	17	28	25
9	2	29	10	10	18	30	26
11	3	31	12	12	19	32	27
13	4	33	13	14	20	34	28
15	5	35	13	16	21	36	29
17	6	37	14	18	22	38	30
19	7	39	15	20	23	40	31

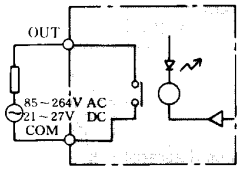
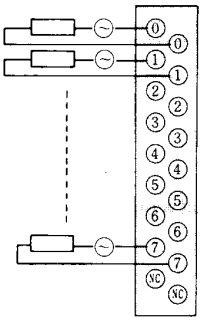
Connector pin	Cable dia.
HIF3-2226SC	AWG22 to 26
HIF3-2428SC	AWG24 to 28

Be sure to use a connector with gold coating.
An exclusive solderless fastening tool is required.

(5) Specifications of Independent Contact Relay Output Module

Table 1-7-7 lists the specifications of independent contact relay output module.

Table 1-7-7 Specifications of Independent Contact Relay Output Module

Item \ Model		P O M-R C
Output specification		Relay output
Nominal voltage		110 V/220 V AC
Output voltage		85 ~ 264 V AC
Max. load current	1 circuit	2 A
Max. rush current		6 A (100ms)
Max. output delay time	O N→O F F	4 ms
	O F F→O N	5 ms
No. of output points		8 points
Common output connection		1 point/common terminal
Isolation method		Relay
Current consmption (average)	C H 1	0.2mA+(No. of ON outputs)×0.2mA
	C H 2	(No. of ON outputs)×10mA
	C H 3	0 mA
Circuit diagram		
External wiring		

(6) Specifications of counter module

Table 1-7-8 lists the specifications of counter module.

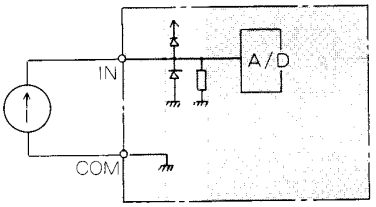
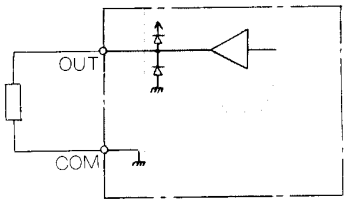
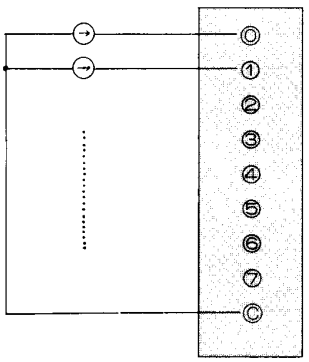
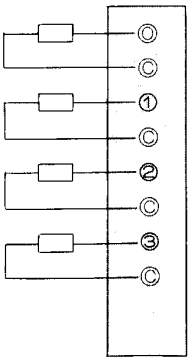
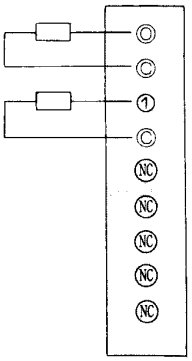
Table 1-7-8 Specification of Counter Module

	Item		Specifications
Input specifications	Count pulse frequency		Max. 10 k Hz
	Input pulse voltage level	O N	0 ~ 2 V
		O F F	5 ~ 12V
	Count pulse width		Min. 20 μsec
	Marker pulse width		Min. 20 μsec
	Input impedance		Approx. 10 k Ω
	Isolation method		Photocoupler
	No. of pulse input points		3 points (A, B and M)
	Polarity		Common ···⊖
	2-phase input pulse	Count-up (addition)	A ON OFF B 90° Phase B is 90° delayed from phase A.
		Count-down (subtraction)	A B 90° Phase B is 90° advanced to phase A.
	Power supply for external input device		12 V DC + 10%, 50 mA (external supply possible)
Output specifications	Output voltage		10 ~ 27 V DC
	Load current		Max. 0.5 A/circuit, max. 1.25 A/4 circuits
	Output system		Transistor (open collector)
	Min. load current		1 mA
	Output delay time	O N→O F F	Max. 1 msec
		O F F→O N	Max. 1 msec
	Voltage drop at ON		Max. 1.5V (0.5A)
	Isolation method		Photocoupler
	No. of output points		4 points (OUT0, OUT1, OUT2 and OUT3)
	Leakage current		Max. 0.1mA
	Polarity		Common ···⊖
	External power supply for signal output		10 to 27 DC, 50 mA (externally supplied to module)
Count range			0 ~ 9999 / 0 ~ 65535
Count system			○ Dual-phase pulse count (up or down) ○ Single phase forward/reverse count (Dual or single phase switch-selectable)
Output			○ 1 point per one set value (open collector) ○ Output held when set value = counter value Switch ○ Output issued when set value is less than selectable counter value
Marker			1 point (The counter value is reset by this signal.)
Operation display			Output and pulse input display
Register			○ Count register ○ Set value CU0, CU1, CU2 and CU3 registers ○ Status/control register
Functions			○ Count value preset ○ Count value read ○ Set value write ○ Set value read ○ Status read Phase-A pulse ON/OFF status Phase-B pulse ON/OFF status Marker ON/OFF status Set value = count value (latch) Set value < count value Overflow flag Underflow flag
Current consumption		C H 1	200mA Max.
		C H 2	0 mA
		C H 3	160 mA max. when 50 mA is supplied to external input device (sensor)
			110 mA max. without current supply to external input device

(7) Specifications of analog current module

Table 1-7-9 lists the specifications of analog current module.

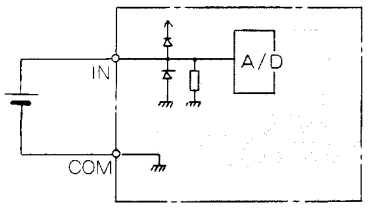
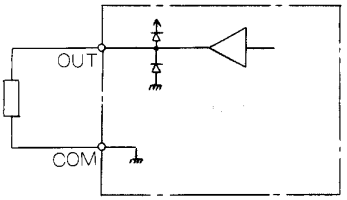
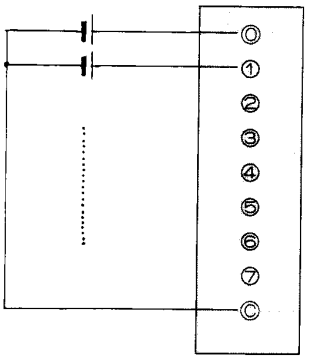
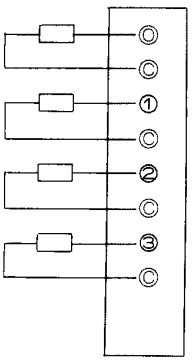
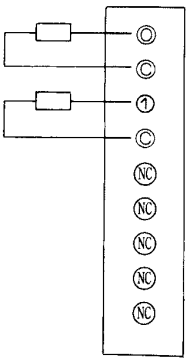
Table 1-7-9 Specifications of Analog Current Module

Item \ Model		AGH-I	AGH-O	AGH-OD
I/O specification		Analog current input	Analog current output	
Current range		4~20mA	4~20mA	
Input impedance		220Ω		
Load impedance		—	0~500Ω	
Resolution		8 bits	8 bits	
Conversion time		1 ms	1 ms	
Overall accuracy		±(1%+1 bit)	±1%	
No. of points		8 points	4 points	2 points
Isolation method		Photocoupler (not isolated from DC input)	Photocoupler (not isolated from DC input)	
Isolation between inputs		Not provided	Not provided .	
Current consumption (average)	CH 1	25mA	50mA	50mA
	CH 2	0 mA	0 mA	0 mA
	CH 3	60mA	250mA	140mA
Circuit diagram				
External wiring				

(8) Specifications of analog voltage module

Table 1-7-10 lists the specifications of analog voltage module.

Table 1-7-10 Specifications of Analog Voltage Module

Item \ Model	AGH-IV	AGH-OV	AGH-ODV
I/O specification	Analog voltage input	Analog voltage output	
Voltage range	0 ~ 10 V DC	0 ~ 10 V DC	
Input impedance	100K Ω	—	
Load impedance	—	10K Ω Min.	
Resolution	8 bits	8 bits	
Conversion time	1 ms	1 ms	
Overall accuracy	±(1%+1 bit)	±1%	
No. of points	8 points	4 points	2 points
Isolation method	Photocoupler (not isolated from DC input)	Photocoupler (not isolated from DC input)	
Isolation between inputs	Not provided	Not provided	
Current consumption (average) CH 1	25mA	50mA	30mA
Current consumption (average) CH 2	0 mA	0 mA	0 mA
Current consumption (average) CH 3	60mA	140mA	70mA
Circuit diagram			
External wiring			

Notes:
1. Unused terminals should be connected to C(common) terminal.

2. COMPATIBILITY

This section describes compatibility between H-200 and other models (H-300/700/2000) of H series.

2. 1 Outline

Fig. 2-1 outlines the result of comparison between H-200 and H-300/700/2000. Four distinct features can be pointed out as shown in the figure.

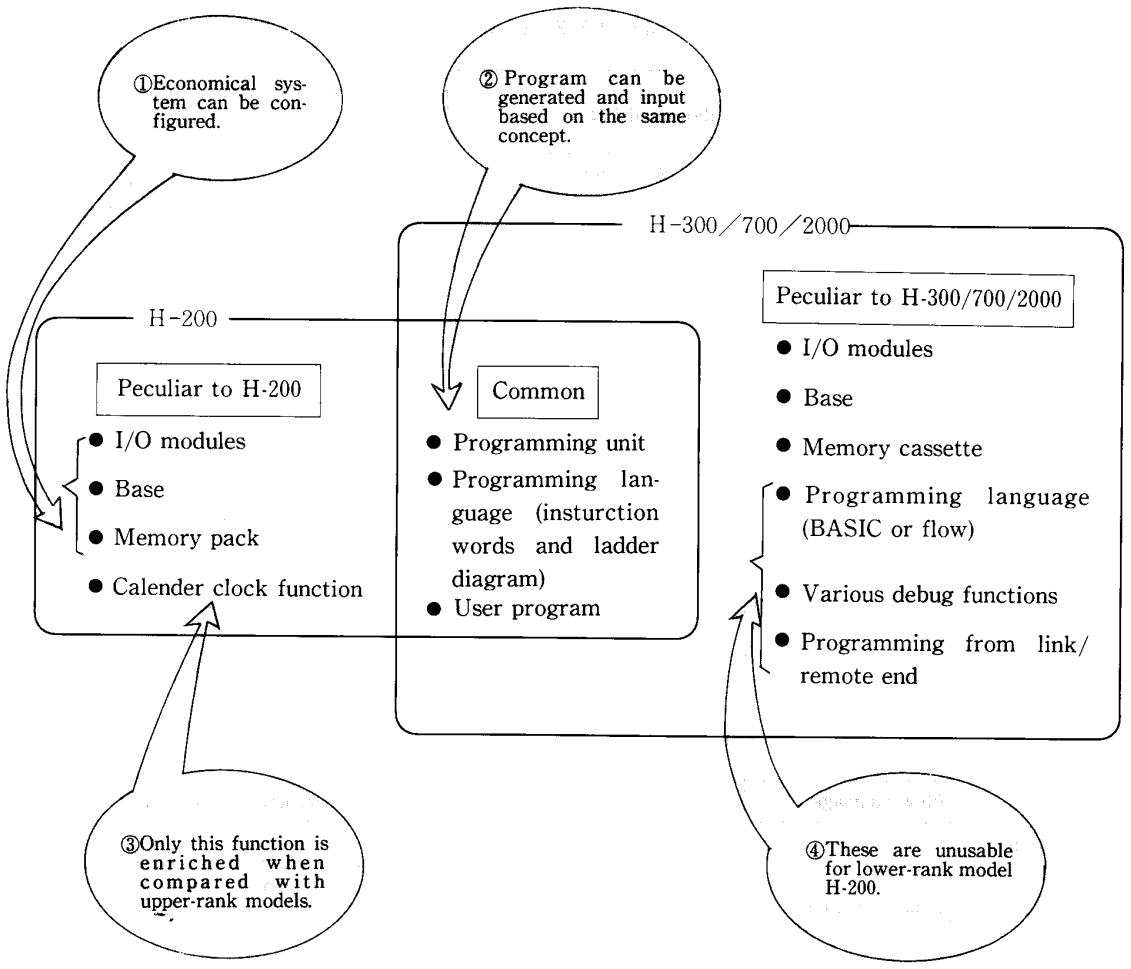


Fig. 2-1 Comparison of H-200 and Upper-Rank Models (H-300/700/and 2000)

2. 2 Unusable Functions

H-200 is incapable of using the I/O categories and instructions listed in the table below unlike the upper-rank models H-300, 700 and 2000. Therefore, take "Substitutive measure" in the table. If the measure cannot be taken, then upper-rank model should be used.

Table 2-1 List of Unusable Functions

No.	Unusable item	Name	Description	Substitutive measure (reference)
1	I/O categories	Doubleword	DX, DY, DL, DR, DM and constant	Expand word by using C (carry).
		Special timer/counter	WDT, MS, TMR, RCU	Combine multiple timers/couters or manipulate current value.
		Array variable	Subscripted I/O	Realize transmission, distibution, extraction, etc. by using multiple assignment statements or comparisons.
2	Insturction	Signed relational box	S=, S<>, S<, S<=	Calculate sign and absolute value separately according to ABS instruction, or calculate by use of two's complement according to NEG instruction.
		Signed comparison expression	S=, S<>, S<, S<=	
		Signed multiplication and division	S*, S/	
		Sign addition and extension	SGET, EXT	
		Transmission excluding XCG	MOV, COPY, WSHR, WSHL, WBSR, WBSL	Use multiple assignment statements.
		7-segment conversion	SEG	Use comparison and assignment statement.
		Square root	SQR	Use approximate expression.
		FIFO	FIFIT, FIFWR, FIFRD	Save and manage number of stored words and read words in internal output.
		BASIC-related control instructions	RSRV, FREE, STAR	Unnecessary because BASIC is unusable
		FOR-NEXT loop	FOR, NEXT	Make loop by using conditional jump insruction.
3	Programming language	BASIC	HI-BASIC	Connect a personal computer and make processing with it.
		Flow language	HI-FLOW	Program in instruction words and ladder.
4	Function	Debug function	Simulation	Operate H-200 with load (machine) power supply turned off or cabling (common cabling) disconnected from load.
			Forced debug execution	Debug without resorting to such function
		Programmer connection at link/remote end	Change of monitor/program from link/remote end	Connect programmer to CPU.
		Installation of advanced function module at remote end	Installation of analog/counter module at remote substation	Install CPU at remote end, load program and communicate with CPU link module (IOLH-T).
		I/O assignment copy	Automatic generation of I/O assignment table	Generate I/O assignment table by key entry.

2. 3 Difference in instruction action

(1) Nesting of subroutine

Table 2-2 Difference in Subroutine Nesting

H-200	H-300/700/2000
Nesting up to 1 level	Up to 6 levels

The Model H-200 allows only 1 level of nesting. So a subroutine cannot call another subroutine.

3. PROCESSING SYSTEM

3. 1 I/O Processing System

The H-200 employs the direct I/O processing system. Differences between the refresh processing system and direct processing system are summarized here.

Refresh Processing System

Before computational processing, the ON/OFF status of external input is fetched in the image memory. Even if the status of external input changes during computational processing, the input status in the image memory remains unchanged. Change in the external input is to be fetched at the input processing of the next scan. Along with computational processing, each status of external output, internal output, etc. changes sequentially on the image memory.

After computational processing, the ON/OFF status of external output on the image memory is outoput to the output circuit.

Therefore, if input XO turns ON immediately after input processing, it is fetched at the next input processing (point a in the figure below) and output Y100 turns ON at the output processing (point b). This signifies that a time period for 2 scans at maximum is taken between input and output.

Direct Processing System

At every computation, the ON/OFF status of external input is fetched and the ON/OFF status of external output is output to the output circuit.

On execution of an instruction (at point c), the ON status of input XO is fetched and output Y100 turns ON. Therfore, a time period for only 1 scan is needed from input to output.

As compared above, the direct system can be said to provide a better response than the refresh system.

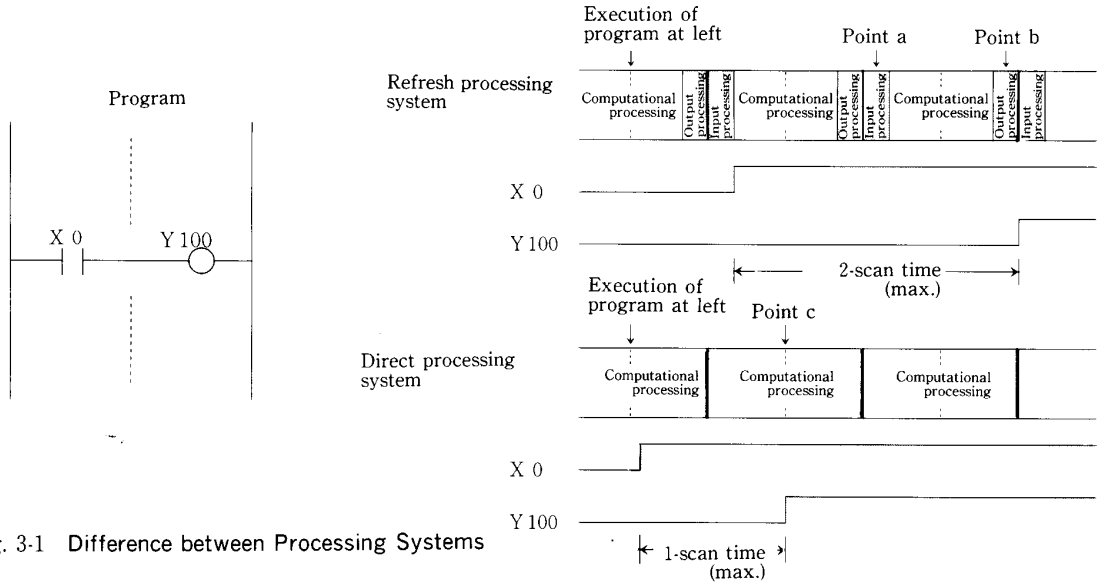
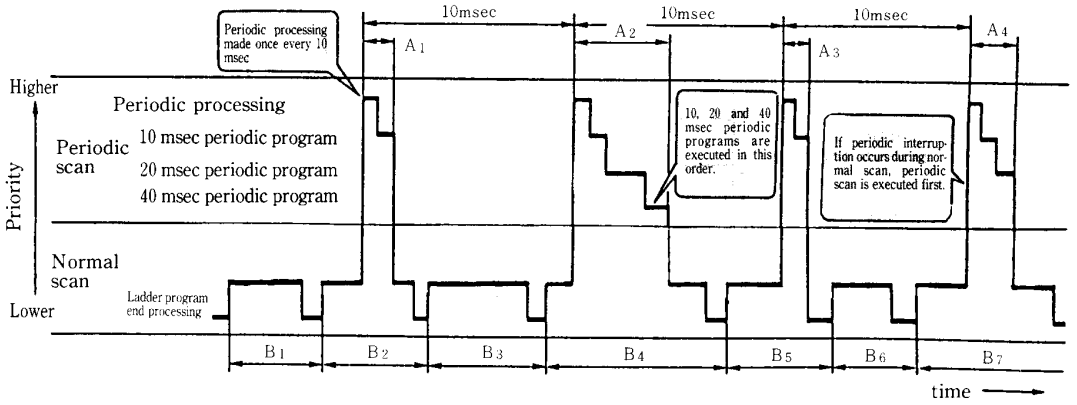


Fig. 3-1 Difference between Processing Systems

3. 2 Scan and Interruption

The CPU module of H-200 performs two kinds of scan; normal scan and periodic scan. The execution sequence and contents of each scan are illustrated in Fig. 3-2.



An: Periodic scan time, Bn: Normal scan time

[Point]
Scan means that program is executed from the start point to the end sequentially.

Fig. 3-2 Operational Interrelation between Scans

A time period taken for each scan is termed a scan time. In Fig. 3-2, suppose that the periodic scan time indicated by A2 is longer than 10 msec (interval of periodic processing). In this case, the next periodic scan starts before the end of the first periodic scan processing. This phenomenon is called a periodic scan congestion error. With regard to normal scan, the number of periodic processings is counted within a time period from one normal scan to the next normal scan. In case the time determined by the periodic processing count X periodic processing interval (10 msec) exceeds the specified time (100 msec if unspecified), a normal scan congestion error occurs. If a congestion error occurs on either periodic scan or normal scan, the CPU module stops program execution (operation stops).
Next, the relation between user program and each scan is shown in Fig. 3-3.

[Point]

- ○ Time period for congestion error check of normal scan: 100 msec (specifiable within 10 to 2550 msec with programming device)
- ○ Time period for congestion error check of periodic scan equals the interval of the shortest periodic scan under use. (An error occurs if the total execution time of all periodic scans exceeds the interval of the shortest periodic scan.)

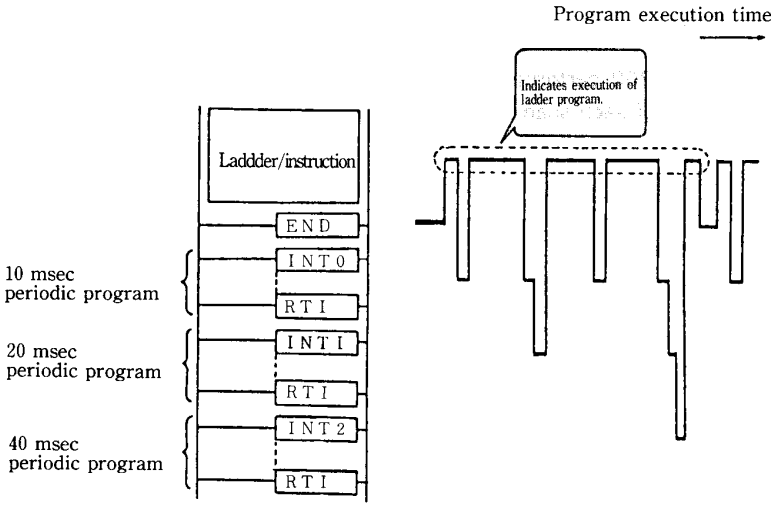
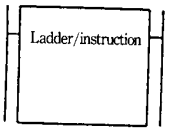
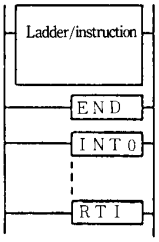


Fig. 3-3 Relation between User Program and Each Scan

The configuration of user program memory shown in Fig. 3-3 represents the case where ladder, instruction and periodic program are all used. An actual program can be composed by using only the necessary elements.



(a) Ladder/instruction alone



(b) Ladder/insturction + periodic program

〔**Point**〕
Diverse program configurations can be realized through various combinations.

Fig. 3-4 Example of Different Program Configurations

4. INPUT AND OUTPUT NUMBERS

This section covers the composition of input/output numbers, usable range, external input/output assignment method and others.

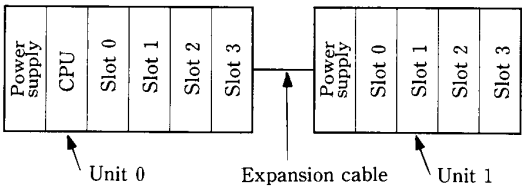
4.1 I/O Number Assignment List

Table 4-1 List of External I/O Numbers

Large category	Intermediate category	Small category	Code	Usable range of H-200	Possible input range of Programming device(Note 1)
External input/ output	External input	Bit	X [0][u][s][b][b]	u: Unit No. (0 to 2) s: Slot No. (0 to 7) bb: Bit No. (0 to 95), decimal Wl: Word No. (0 to 7) Memory protection at power failure impossible	u: Unit No. (0 to 5) s: Slot No. (0 to F), hexadecimal bb: Bit No. (0 to 95), decimal Wl: Word No. (0 to 7) Memory protection at power failure impossible
		Word	W X [0][u][s][Wl]		
	External output	Bit	Y [0][u][s][b][b]		
		Word	W Y [0][u][s][Wl]		
Remote external input/ output	External input	Bit	X [r][St][s][b][b]	r: Remote master station No. (1 to 4) St: Remote slave station No. (0 to 7) s: Slot No. (0 to 7) bb: Bit No. (00 to 15), decimal Wl: Word No. (0 to 1) Memory protection at power failure impossible	r: Remote master station No. (1 to 4) St: Remote slave station No. (1 to 9) s: Slot No. (0 to F), hexadecimal bb: Bit No. (00 to 95), decimal Wl: Word No. (0 to 7) Memory protection at power failure impossible
		Word	W X [r][St][s][Wl]		
	External output	Bit	Y [r][St][s][b][b]		
		Word	W Y [r][St][s][Wl]		

Notes:

1. The possible input range of programming device can be restricted within the usable range of H-200 by registering the I/O assignment table correctly so as to meet the H-200.
2. Doubleword (DX, DY) is unusable.
3. Unit and slot numbers are exemplified below.



CAUTION:
Correct operation is impossible if registration of I/O assignment table is wrong.

- Basic base accommodates unit 0 and the first expansion base houses unit 1.
- In the basic base, slots are numbered 0, 1, 2 and on starting from the one at the right of CPU.
- In the expansion base, slots are numbered 0, 1, 2 and on starting from the one at the right of power module.

4. Unlike the method in Note 3, slot numbers can be assigned sequentially from the slot next to CPU across unit numbers. This method is employed for H-100M.

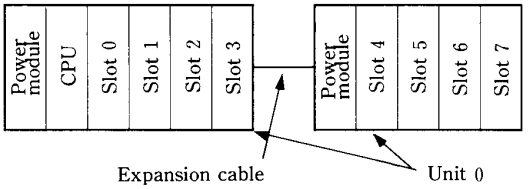


Table 4-2 List of Internal Output and Timer/Counter Numbers

Large category	Intermediate category	Small category	Usable range of H-200	Possible input range of programming device	Remarks
CPU link area	1st CPU link area	Bit	L 0 ~ 7 F	L 0 ~ 3 F F F	○ Memory not protected at power failure (hexadecimal) ○ Not cleared when CPU turns from STOP to RUN.
		word	W L 0 ~ 7	W L 0 ~ 3 F F	
	2nd CPU link area	Bit	L10000~1007 F	L10000~13 F F F	
		word	W L1000~1007	W L1000~13 F F	
Internal output	Bit/word shared	Bit	M 0 ~ F F F	M 0 ~ 3 F F F	Memory protected at power failure (hexadecimal)
		word	WM 0 ~ F F	WM 0 ~ 3 F F	
	Bit/word independent	Bit	R 0 ~ 7 B F		
		word	W R 0 ~ 3 F F		
	Special internal output	Bit	R 7 C 0 ~ 7 F F		○ Provision of memory protection determined for No. (hexadecimal)
		word	W R F000~ F 1 F F		
Edge detection	Rising edge detection	Bit	D I F 0 ~127	D I F 0 ~511	○ Memoey protected at power failure ○ Number duplication allowable
	Falling edge detection		D F N 0 ~127	D F N 0 ~511	
Master control	Master control set	Bit	M C S 0 ~49		○ Decimal ○ Number duplication allowable
	Mster control reset		M C R 0 ~49		
Timer/ counter	On-delay timer	Bit	T D 0 ~255		○ Memory protected at power failure (decimal) ○ Each usable range of timer and counter is restricted within 0 to 63 when time base is 0. 01 s. ○ Number duplication unallowable ○ Up/down counters(CTU, CTD, CT) are used in combination. ○ Total of timers and counters must not exceed 512 points.
	Single-shot timer		S S 0 ~255		
	Up counter		C U 0 ~511		
	Up/down counter up coil		C T U 0 ~511		
	Up/down counter down coil		C T D 0 ~511		
	Up/down counter contact		C T 0 ~511		
	Current value clear		C L 0 ~511		
	Timer/counter current value	word	T C 0 ~511		
Constant value	Decimal	word	0 ~65,535		
	Hex-decimal	word	H 0 ~ H F F F F		
	Bit	Bit	0, 1		

Notes:

1. Doubleword is unusable.
2. If an input or output number outside the usable range of H-200 is used in a program, it is automatically converted as follows.

Bit I/O No.→ L13 F F F , D I F→D I F511

Word I/O No.→W L13 F F , D F N→D F N511

Doubleword I/O No.→ D L13 F E

4.2 Memory-Protectable I/O

Memory protection range of R, WR, WM, TD, DIF and DFN at power failure can be set or canceled. Note, however, that the range of memory protection at power failuer need be specified from a peripheral device to the CPU preliminary. For operating procedure, refer to the manual of each programming device.
On start of RUN, the CPU clears the data memory area for the above I/O to 0 (zero). But the range specified for memory protection at power failure is not cleared, or retains the previous value.
Even if power supply to the CPU is cut off, the data within the range specified for memory protection at power failure is retained because of backup by the battery built in the CPU.
Memory-protectable I/Os and their protection ranges are listed below.

I/O classification	Settable range
R	R 0 ~ R 7 B F
WR	WR 0 ~ WR 3 F F
WM	WM 0 ~ WM F F
TD (Note 2)	TD 0 ~ TD511
DIF	DIF 0 ~ DIF127
DFN	DFN 0 ~ DFN127

Notes:

1. Error does nor occur even if setting has been made beyond the settable range. However, the data outside the range is always handled as 0 (zero).
2. TD includes the following I/O's.
   Timers ····· TD, SS
   Counters ····· CU, CTU, CTD, CT

4.3 Special Internal Outputs

Special internal outputs are internal outputs having a special meaning. These outputs are provided for reporting the system status to the user program and for allowing the user program to control the system status. (Provision for memory protection of special internal outputs at power failure is determined depending on their numbers.)
The area below is assigned to the special internal outputs. Utilize these outputs after understanding their specifications in clear distinction from general internal outputs.

- Word special internal output area:
  WRF000 to WRF1FF (512 words)
- Bit special internal output area:
  R7C0 to R7FF (64 bits)

List of Special Internal Outputs

Bit Special Internal Outputs (64 points)

	Number
Continuation despite congestion error (normal)	R 7 C 0
Continuation despite congestion error (periodic)	1
Continuation despite congestion error (interrupt)	2
(Undefined)	3
	4
	5
	6
Mid-RUN change enable	7
Major error	8
(Undefined)	9
Memory abnormal(user, PI/O)	A
(Undefined)	B
Memory oversize (user)	C
(Undefined)	D
	E
	R 7 C F
	R 7 D 0
Congestion error (nermal)	1
Congestion error (Periodic)	2
Congestion error (interrupt)	3
Syntax/assemble error	4
(Undefined)	5
	6
	7
	8
Battery error	9
(Undefined)	A
Self-diagnosis error	B
(Undefined)	C
	D
	E
	R 7 D F
Key switch position (STOP)	R 7 E 0
(Undefined)	1
Key switch position (RUN)	2
1 scan ON after RUN	3
Always ON	4
0.02 sec clock	5
0.1 sec clock	6
1.0 sec clock	7
Occupancy declaration	8
RUN inhibitstatus	9
Mid-RUN change start	A
LED (error) clear	B
Error special internal output clear	C
(Undefined)	D
	E
	R 7 E F
Carry	R 7 F 0
Overflow	1
Shift data	2
Computation error	3
Data error	4
(Undefined)	5
	6
	7
Time reading requeat	8
Time setting request	9
±30 sec adjust	A
Time setting error	B
(Undefined)	R 7 F C
	⌇
	R 7 F F

Word special internal outputs (512 words)

	Number
Self-diagnosis error code	W R F 0 0 0
Details of syntax/assemble error	1
(Undefined)	2
	3
	4
	5
	6
	7
Error circuit No.	8
(Undefined)	9
	A
Year (always readadle)	B
Month and day (always readadle)	C
Day of week (always readadle)	D
Hour and minute (always readadle)	E
Second (always readadle)	W R F 0 0 F
Max. cycle time	W R F 0 1 0
Current cycle time	1 1
Min cycle time	1 2
CPU status	1 3
Word internal output	1 4
Computation error code capacity	1 5
Expansion register for computation	1 6
	1 7
(Undefined)	1 8
	1 9
	1 A
Year (readout, setting)	1 B
Month and day (readout, setting)	1 C
Day of week (readout, setting)	1 D
Hour and minute (readout, setting)	1 E
Second (readout, setting)	1 F
	W R F 0 2 0
(Undefined)	⌇
	W R F 1 F F

Table 4-3 Details of Bit Special Internal Outputs (1/3)

No.	Name	Contents	Details
R 7 C 0	Continuation despite congestion error (normal scan)	0: Stops RUN at error. 1: Continues RUN despite error.	Specifies stop or continuation of RUN if congestion error of normal scan occurs.
R 7 C 1	Continuation despite congestion error (periodic scan)	0: Stops RUN at error. 1: Continues RUN despite error.	Specifies stop or continuation of RUN if congestion error of periodic scan occurs.
R 7 C 2	Continuation despite congestion error (interrupt scan)	0: Stops RUN at error. 1: Continues RUN despite error.	Specifies stop or continuation of RUN if congestion error of interrupt scan occurs.
R 7 C 3 ⌇ R 7 C 6	(Undefined)		
R 7 C 7	Mid-RUN change enable	0: Inhibits change during RUN. 1: Enables change during RUN.	Specifies whether or not user program is changeable during RUN.
R 7 C 8	Major error	0: Indicates absence of major error. 1: Indicates occurrence of major error.	Indicates whether or not control microcomputer is abnormal.
R 7 C 9	(Undefined)		
R 7 C A	Memory abnormal	0: Indicates absence of major error. 1: Indicates occurrence of major error.	Indicates whether or not user memory is abnormal.
R 7 C B	(Undefined)		
R 7 C C	Memory oversize	0: Normal 1: Abnormal	Indicates whether or not user program memory size set by parameter is exceeded.
R 7 C D ⌇ R 7 D 0	(Undefined)		
R 7 D 1	Congestion error (normal scan)	0: Normal 1: Abnormal	Indicates whether or not normal scan exceeds parameter-specified execution time.
R 7 D 2	Continuation despite congestion error (periodic scan)	0: Normal 1: Abnormal	Indicates whether or not periodic scan has been completed within the determined period.
R 7 D 3	Continuation despite congestion error (interrupt scan)	0: Normal 1: Abnormal	Indicates whether or not interruption of the same factor has occurred during interrupt scan.
R 7 D 4	Syntax/assemble error	0: Normal 1: Abnormal	Indicates whether or not user program has syntax error. (Detailed information is output in WRFOO1.)
R 7 D 5 ⌇ R 7 D 8	(Undefined)		

Table 4-3 Details of Bit special internal Outputs (2/3)

No.	Name	Contents	Details
R 7 D 9	Battery error	0: Normal 1: Abnormal	Indicates whether or not battery charge in CPU module in low.
R 7 D A	(Undefined)		
R 7 D B	Self-diagnosis error	0: Normal 1: Abnormal	Indicates the result of self-diagnosis. (Detailed information is output in WRF000.)
R 7 D C 〃 R 7 D F	(Undefined)		
R 7 E 0	Mode selector key switch position (STOP)	0: Key position RUN 1: Key position STOP	Either R7E0 or R7E2 turns ON.
R 7 E 1	(Undefined)		
R 7 E 2	Mode selector key switch position (RUN)	0: Key position STOP 1: Key position RUN	Either R7E0 or R7E2 turns ON.
R 7 E 3	1 scan ON after RUN	0: From 2 scans after RUN 1: 1 scan after RUN	ON / OFF, 1 scan, A, A: RUN start point
R 7 E 4	Always ON	0: No 1: Yes	
R 7 E 5	0.02 sec clock	0: 0.01 sec 1: 0.01 sec	ON / OFF, A, 0.01 sec, 0.01 sec, A: RUN start point
R 7 E 6	0.1 sec clock	0: 0.05 sec 1: 0.05 sec	ON / OFF, A, 0.05 sec, 0.05 sec, A: RUN start point
R 7 E 7	1.0 sec clock	0: 0.5 sec 1: 0.5 sec	ON / OFF, A, 0.5 sec, 0.5 sec, A: RUN start point
R 7 E 8	Occupancy declaration	0: Not occupied 1: Occupied	Occupancy indicates CPU is under communication with peripheral device, etc.
R 7 E 9	RUN inhibit status	0: Enables RUN. 1: Inhibits RUN.	Indicates whether or not RUN is inhibited.
R 7 E A	Mid-RUN change start	0: Other than under change during RUN 1: Under change during RUN	Indicates whether or not RUN is at pause (output is held) due to change amid RUN.
R 7 E B	LED (error) clear	Cleared at 1	
R 7 E C	Error special internal output clear	Cleared at 1	
R 7 E D 〃 R 7 E F	(Undefined)		

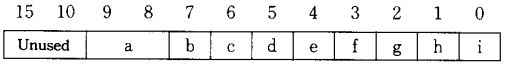
Table 4-3 Details of Bit Special internal Outputs (3/3)

No.	Name	Contents	Details
R 7 F 0	Carry	0: Carry OFF 1: Carry ON	Carry flag used in arithmetic instruction
R 7 F 1	Overflow	0: Without overflow 1: With overflow	Overflow flag in arithmetic instruction
R 7 F 2	Shift data	0: Shift data "0" 1: Shift data "1"	Shift data used in shift instruction, etc.
R 7 F 3	Computation error	0: Normal 1: Abnormal	Indicates whether or not instruction error has been detected during execution of arithmetic instruction. (Detailed information is output in WRF015.)
R 7 F 4	Data error	0: Normal 1: Abnormal	Indicates whether or not data error has been detected during execution of arithmetic instruction.
R 7 F 5 〜 R 7 F 7	(Undefined)		
R 7 F 8	Time readout request	Time read out at ⎍⎾	Year, month/day, day of week, hour/minute and second are read out in WRF01B to WRF01F when ⎍⎾ is set. After information has been read out, this request is reset by system.
R 7 F 9	Time setting request	Time set at ⎍⎾	The values in WRF01B to WRF01F are set to calendar clock when ⎍⎾ is set. After information has been set, this request is reset by system.
R 7 F A	±30 sec adjust	30 sec adjusted at ⎍⎾	Clock is adjusted by 30 sec when ⎍⎾ is set. After adjustment, this output is reset by system.
R 7 F B	Time setting error	0: Normal 1: Abnormal	Indicates whether or not set data has error upon time setting.
R 7 F C 〜 R 7 F F	(Undefined)		

Note: Output in an undefined number is neglected. And input from an undefined number causes input of 0 or previously output data.

4

Table 4-4 Details of Word Special Internal Outputs (1/2)

No.	Name	Stored data	Description
WR F000	Self-diagnosis error code	Self-diagnosis error No.	Stores the number of error detected by CPU in binary code. (Refer to 9.2 to Section 9.)
WR F001	Detail of syntax/assemble	Syntax/assemble error No. error	Stores syntax/assemble error in user program in binary code. (Refer to 9.3 of Section 9.)
WR F002 ⌇ WR F007	(Undefined)		
WR F008	Error circuit No.	No. of circuit with error	Stores the number of circuit which has undefined instruction, abnormal I/O No. and syntax/assemble error.
WR F009 ⌇ WR F00A	(Undefined)		
WR F00B	Year (always)	Current year on calendar clock	Always sets 4-digit year in BCD code.
WR F00C	Month/day (always)	Current month and day on calendar clock	Always sets month and day in BCD code
WR F00D	Day of week (always)	Current day of week on calendar clock	Always sets day of week in BCD code. (Note 1)
WR F00E	Hour/minute (always)	Current hour and minute on calendar clock	Always sets hour (24-hour base) and minute in BCD code.
WR F00F	Second (always)	Current second on calendar clock	Always sets second in BCD code. (Note 2)
WR F010	Cycle time (max.)	Max. execution time of normal scan	In 10 msec steps
WR F010	Cycle time (current)	Execution time of normal scan	In 10 msec steps
WR F012	Cycle time (min.)	Min. execution time of normal scan	In 10 msec steps The min. execution time is set to 65535, and then to proper value after RUN.
WR F013	CPU status	Operational status of CPU  a = CPU type~01=H-2000, 10=H-700, 11=H-300 The CPU type of H-200 is the same as H-300. b = battery error~1=detected, 0=not detected c, d, e, g, h = unused f = error~1=detected, 0=not detected i = operation~1=RUN, 0=STOP	

Notes:

1. 0: Sunday, 1: Monday, 2: Tuesday, 3: Wednesday, 4: Thursday, 5: Friday, 6: Saturday
   Upper 3 digits are always set at "000".
2. Upper 2 digits are always set at "00".

Table 4-4 Details of Word Special Internal Outputs (2/2)

No.	Name	Stored data	Description
WR F014	Word internal output	Number of words in word capacity internal output	Any of 0400, 4400 and C400
WR F015	Computation error code	Computation error No.	Stores the number of error which occurred during execution of arithmetic instruction. (Refer to 9. 4 of Section 9.)
WR F016 WR F017	Expansion register for	Remainder at execution of computation arithmetic instruction	32-bit computation: High order~F017 Low order~F016 16-bit computation: F016 used
WR F018 ⌇ WR F01A	(Undefined)		
WR F01B	Year (readout, setting)	4-digit year	Reads 4-digit year from calendar clock when R7F8 is at _┌‾. Sets 4-digit year to calendar clock when R7F9 is at _┌‾.
WR F01C	Month/day (readout, setting)	Month/day	Reads month and day from calendar clock when R7F8 is at _┌‾. Sets month and day to calendar clock when R7F9 is at _┌‾.
WR F01D	Day of week (readout, setting)	Day of week (Note 1)	Reads the day of week from calendar clock when R7F8 is at _┌‾. Sets the day of week to calendar clock when R7F9 is at _┌‾.
WR F01E	Hour/minute (readout, setting)	Hour (24-hour base) and minute	Reads hour (24-hour base) and minute from calendar clock when R7F8 is at _┌‾. Sets hour (24-hour base) and minute to calendar clock when R7F9 is at _┌‾.
WR F01F	Second (readout, setting)	Second (Note 2)	Reads second from calendar clock when R7F8 is at _┌‾. Sets second to calendar clock when R7F9 is at _┌‾.
WR F020 ⌇ WR F1FF	(Undefined)		

Notes:

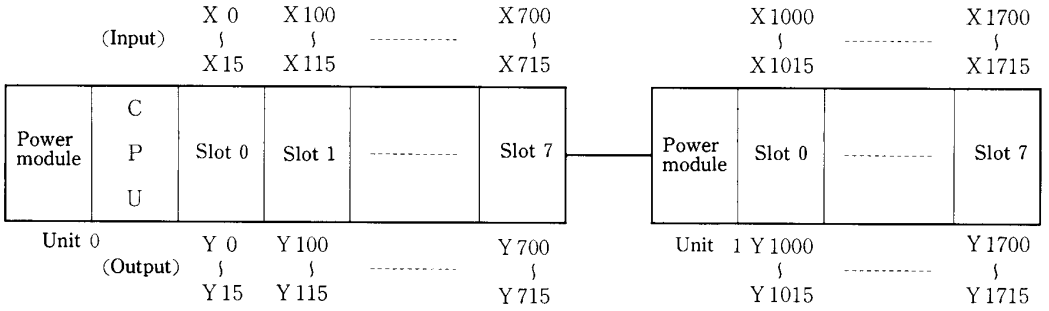
1. 0: Sunday, 1: Monday, 2: Tuesday, 3: Wednesday, 4: Thursday, 5: Friday, 6: Saturday
   Upper 3 digits are always set at "000".
2. Upper 2 digits are always set at "00".
3. Output in an undefined number is neglected. And input from an undefined number causes input of 0 or previously output data.

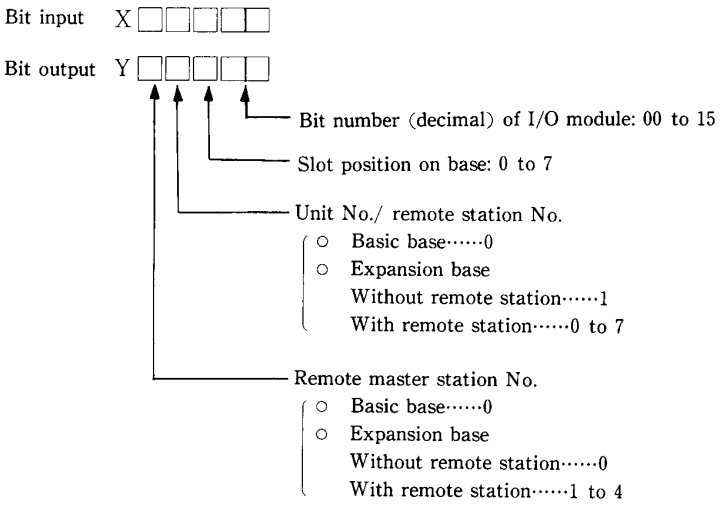
4.4 Assignment of External Inputs and Outputs

Assignment of external inputs and outputs is determined depending on I/O classification, discrimination among basic, expansion and remote I/O, mounting slot position, and bit and word numbers in module.

(1) Assignment of bit module

I/O assignment of bit module is shown in the figure below. The figure shows that input is numbered X0 to X15 when 16-point input is assigned to slot 0 and that output is numbered Y100 to Y115 when output is assigned to slot 1.





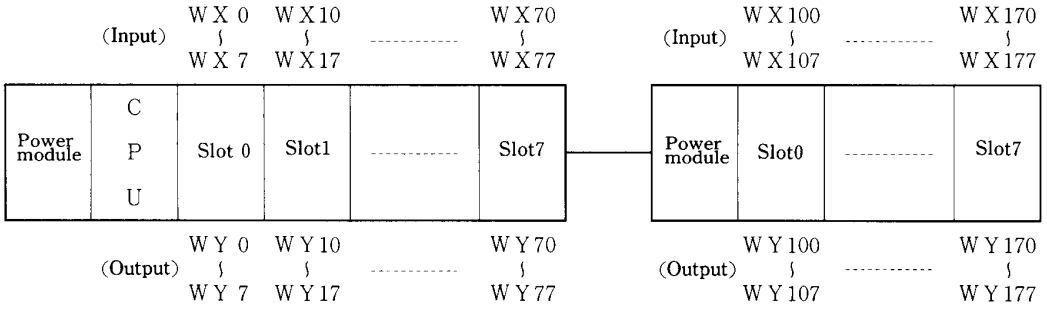
CAUTION

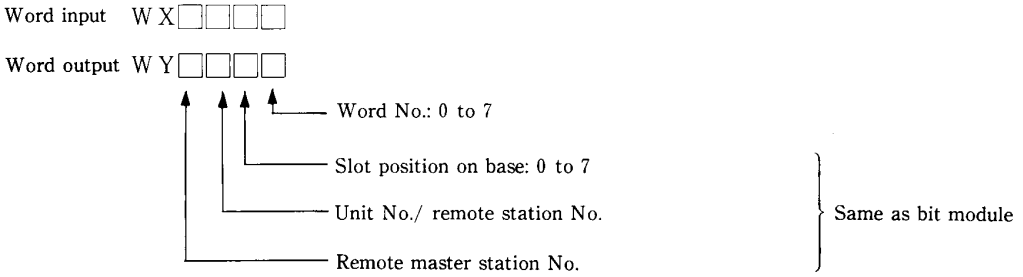
I/O mixed modules are assigned as follows.

Model	Input No.	Output No.
PHH-DT	Input 8 points X0 to X7	Output 8 points Y24 to Y31
PHM-TT	Input 16 points X0 to X15	Output 16 points Y16 to Y31

(2) Assignment of word module

Assignment of word (16 bits) module is shown in the figure below. The figure shows that input is numbered WX0 to WX7 when the word input module is assigned to slot 0 and that output is numbered WY10 to WY17 when output is assigned to slot 1.

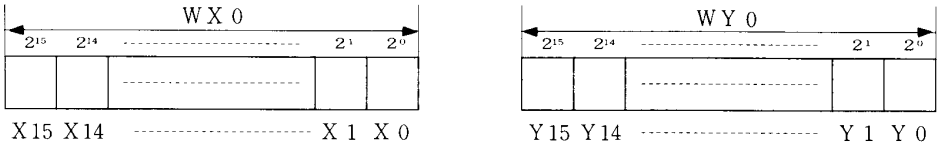




○ The word I/O modules of H-200 are listed in the table below.
The word I/O modules are mountable to the basic and expansion bases, but they are unusable at the remote end.

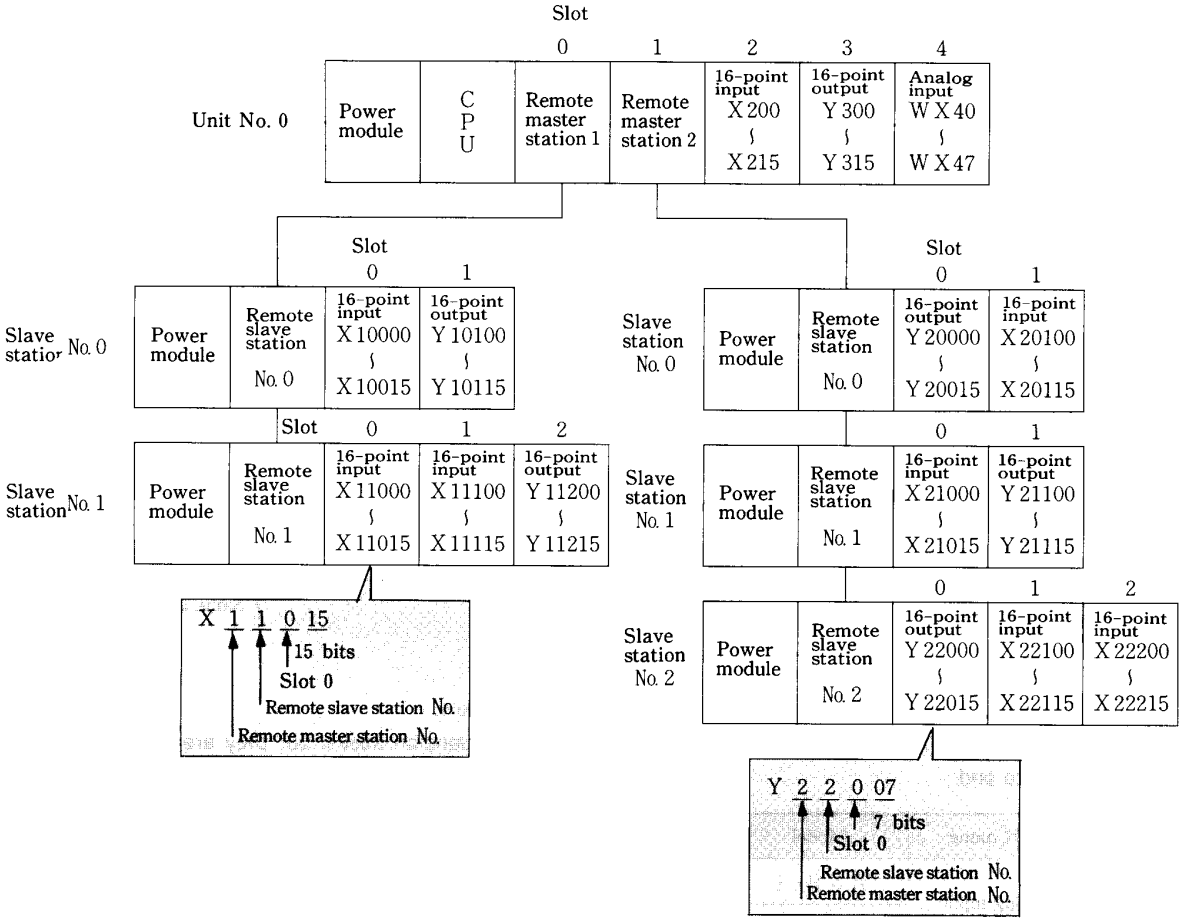
Part name	Model	Input (WX)	Output (WY)	Specifications
Analog input	AGH-I	WX0 ~ WX7	——	8 channels, 4 to 20 mA, 8 bits
	AGH-IV			8 channels, 0 to 10 V, 8 bits
Analog output	AGH-O	——	WY0 ~ WY3	4 channels, 4 to 20 mA, 8 bits
	AGH-OV			4 channels, 0 to 10 V, 8 bits
	AGH-OD	——	WY0, WY1	2 channels, 4 to 20 mA, 8 bits
	AGH-ODV			2 channels, 0 to 10 V, 8 bits
Counter	CTH	WX0, WX1	WY2 ~ WY7	2 phases, 10 kHz

○ WX and WY are also used for collective input and output of 16-bit data from/to a bit module.
The relation between bit and word is shown below.



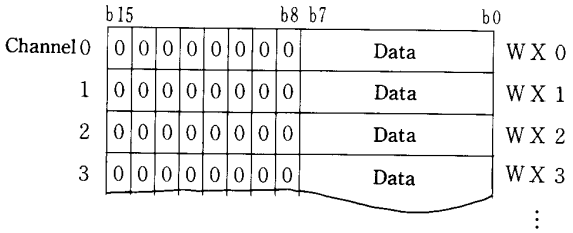
4

4.5 Example of Remote I/O and Analog Module Assignment



Assignment is exemplified for use of 2 remote master stations.

- Remote master station occupies 1 slot width as in a general module.
- Up to 4 remote master stations (master station Nos. 1 to 4)can be mounted.
- Since the analog input module is mounted in the slot No. 4 of basic base (unit No. 0), input numbers WX40 to WX47 are assigned. Analog input module has an 8-bit configuration, and the upper 8 bits of register are all 0 (zero). Register configuration is shown below.



5. PROGRAMMING

5.1 Classification of Instructions

The instructions used for H-200 are classifiable as shown in Table 5-1.

Table 5-1 Classification of Instructions

No.	Classification	Description	Kind
1	Basic instruction	Sequence	21
		Timer/counter	6
		Relational box	4
2	Arithmetic instruction	Assignment statement	1
		4-rule expression	8
		Logical expression	3
		Comparison expression	4
3	Application instruction	Bit operation	3
		Shift/rotate	8
		Transfer	1
		Negation	3
		Conversion	4
		Application (bit count, exchange, unite, separation)	4
4	Control instruction	END, CEND, JMP, LBL, CJMP CAL, SB, RTS, INT, RTI	10

5.2 Instruction List

How to read the instruction list is described here.

Table 5-2 Instruction List

Class	No.	Ladder symbol	Instruction symbol	Instruction name	Contents of processing	Size		Type	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C	Step	Remarks
Bit operation	1	BSET (d, n)		Bit Set	d : Word n : 0 ~ 15	Word	d n	Word d WY WR WL WM TC	•	•	•	•	•	3	
	2	BRES (d, n)		Bit Reset	d : Word n : 0 ~ 15	Word	d n	n WX WY WR WL WM TC Constants	•	•	•	•	•	3	
	3	BTS (d, n)		Bit Test	d : Word n : 0 ~ 15	Word	d n		•	•	•	•	↕	3	

① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩

(1) Class ····· Indicates the classification of instruction by processing.

(2) No. ····· Indicates the serial number of each instruction.

(3) Ladder symbol ····· Indicates the format used to generate a program with ladder.
Instruction symbol ····· Indicates the format used to generate a program with instructions.

(a) Instructions come in two kinds; one is the basic instructions which have symbols different from ladder symbols, and the other is arithmetic, application, control and special instructions whose symbols are the same as ladder symbols.

(b) Symbols d, s and n in the parens "( )" denote the I/O to be processed or the typo of numeric values.
d (destination): The contents of I/O are changed after processing.
s (source): The contents of I/O are referred to. They remain the same.
n (number): Indicates size, bit position or label number.

(4) Instruction name ····· Indicates the name of each instruction.

(5) Contents of processing ····· Outlines the processing to be made by each instruction.

(6) Size ····· Indicates the I/O size usable for each instruction.

(a) Bit:
1 constant

(b) Word:
1 constant

(7) Type ····· Indicates the type of I/O usable for each instruction.

(8) Condition codes

C ····· Carry (R7F0 of special internal output) Represents the contents of carry by addition, borrow by subtraction, or shift.

V ····· Overflow (R7F1 of special internal output)
Represents that overflow occurred as a result of signed data operation regarding data as a two's complement.
Example: The range of signed word data is from −32768 to 32767.

```
WR0001 = -32768
WR0002 = 1
WR0000 = WR0000 - WR0002
```

Because WR0000 exceeds the range of −32768, it causes an overflow. For details, refer to "binary addition" and "binary subtraction."

SD ····· Shift data (R7F2 of special internal output)
The contents of SD are shifted by the SHR or SHL instruction.

ERR ····· Error (R7F3 of special internal output)
"1" is set if an error occurs during the execution of a control or special instruction.
The error code is also set in WRF015. Unless an error is detected, the previous status is retained.

DER ····· Data error (R7F4 of special internal output)
This is set to "1" to represent a data error if the I/O number exceeds the limit or abnormal BCD data is set. Set to "0" with on data error detected.

↕ ····· Changes according to the result of computation.

● ····· Holds the previous status.

① ····· This is set to 1 if an error is detected in the result of computation. Otherwise, the previous status is held.

(9) Step ····· Indicates the number of instruction steps.
The number of steps may differ between the same instructions depending on the size or type.

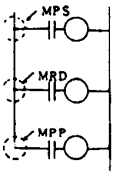
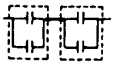
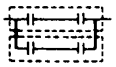
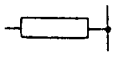
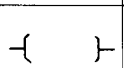
(10) Remarks ····· Remarks about each instruction

Instruction are listed below.

(1) Basic instructions

Class	No.	Ladder symbol	Instruction symbol	Instruction name	Contents of processing	Size	Type	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C	Step	Remarks
Sequence instruction	1		LD	Start Logical Operation	Denotes the start of contact "a" operation of relay function type X, Y, R, L, M, TD, SS, CU, CT, DIF or DFN.	Bit	X OUS bb X RStS bb Y OUS bb Y RStS bb Decimal R0~R7FF L0~L7F L10000~L1007F M0~MFFF TD0~TD255 SS0~SS255 CU0~CU511 CT0~CT511	•	•	•	•	•	1	
	2		LDI	Start Logical NOT operation	Denotes the start of contact "b" operation of relay function type X, Y, R, L, M, TD, SS, CU, CT, DIF or DFN.	Bit								
	3		AND	AND	Denotes contact "a" series connection of relay function type X, Y, R, L, M, TD, SS, CU, CT, DIF or DFN.	Bit								
	4		ANI	NAND	Denotes contact "b" series connection of relay function type X, Y, R, L, M, TD, SS, CU, CT, DIF or DFN.	Bit								
	5		OR	OR	Denotes contact "a" parallel connection of relay function type X, Y, R, L, M, TD, SS, CU, CT, DIF or DFN.	Bit								
	6		ORI	NOR	Denotes contact "b" parallel connection of relay function type X, Y, R, L, M, TD, SS, CU, CT, DIF or DFN.	Bit	DIF0~DIF127 DFN0~DFN127	•	•	•	•	•	2	
	7		NOT	Logical NOT	Inverts the result of computation.	Bit	—	•	•	•	•	•	2	
	8	DIF	AND DIF	Rising Edge Detection	Denotes the detection of signal rise (↑).	Bit	DIF0 ~DIN127 Decimal	•	•	•	•	•	3	NO. duplication impossible
		DIF	OR DIF										4	
	9	DFN	AND DFN	Falling Edge Detection	Denotes the detection of signal fall (↓).	Bit	DFN0 ~DFN127 Decimal	•	•	•	•	•	3	NO. duplication impossible
		DFN	OR DFN										4	
	10		OUT	Device Output	Denotes the output coil of relay function type Y, R, L, M, TD, SS, CU, CTU, CTD or CI.	Bit	Decimal Y OUS bb Y RStS bb R0~R7FF L0~L7F L10000~L1007F M0~MFFF TD0~TD255 SS0~SS255 CU0~CU511 CTU0~CTU511 CTD0~CTD511 CL0~CL511	•	•	•	•	•	1 —	
	11	SET	SET	Device Set	Denotes set output of relay function type Y, R, L or M.	Bit	Decimal Y OUS bb Y RStS bb R0~R7FF L0~L7F L10000~L1007F M0~MFFF	•	•	•	•	•	1	
	12	RES	RST	Device Reset	Denotes reset output of relay function type Y, R, L or M.									
	13	MCS	MCS	Master Control Start	Denotes the set operation of master control.	Bit	MCS0 ~MCS49	•	•	•	•	•	3	No duplication possible
	14	MCR	MCR	Master Control Reset	Denotes the reset operation of master control.	Bit	MCR0 ~MCR49	•	•	•	•	•	2	No duplication possible

5

Class	No.	Ladder symbol	Instruction symbol	Instruction name	Contents of processing	Size	Type	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C	Step	Remarks
Sequence instruction	15		MPS	Push Computation Result	Stores the computation result just before MPS instruction.			•	•	•	•	•	0	
	16		MRD	Read Computation Result	Reads the computation result stored by the MPS instruction and continues computation according the next instruction.									
	17		MPP	Pull Computation Result	Reads the Computation result stored by the MPS instruction and continues computation according to the next instruction. Clears the computation result stored by the MPS command.									
	18		ANB	Logical Block Series Connection	Denotes series connection of two logical blocks.			•	•	•	•	•	0	
	19		ORB	Logical Block Parallel Connection	Denotes parallel connection of two logical blocks.								1	
	20		[ ]	Processing Box Start and End	Programs an arithmetic, application, special or control instruction between processing box start "[" and end "]".			•	•	•	•	•	3	
	21		[ ]	Relational Box Start and End	Programs a relational box instruction between relational box start "[" and "]".			•	•	•	•	•	—	
Timer/counter	1		OUT TD	On-delay timer	Denotes on-delay timer operation.		TD0~TD255 Possible within 0 to 63 at 0.01 sec base	•	•	•	•	•	5	
	2		OUT SS	Single shot	Denotes single shot operation.		SS0~SS255 Possible within 0 to 63 at 0.01 sec base	•	•	•	•	•	5	
	3		OUT CU	Counter	Denotes counter operation.		CU0~CU511	•	•	•	•	•	5	
	4		OUT CTU	Up/down Counter Up	Denotes up/down counter up operation. Use with the up/down counter down in a pair.		CTU0~ ~CTU511	•	•	•	•	•	5	
	5		OUT CTD	Up/down Counter Down	Denotes up/down counter down operation. Use with the up/down counter up in a pair.		CTD0 ~CTD511	•	•	•	•	•	3	
	6		OUT CL	Counter Clear	Denotes counter clear (CU, CTU or CTD).		CL0~CL511	•	•	•	•	•	1	

Basic Instructions

Class	No.	Ladder symbol	Instruction symbol	Instruction name	Contents of processing	Size		Type	R7F4 DER	R7F3 ERR	R7F2 S	D	R7F1 V	R7F0 C	Step	Remarks
Relational box	1	S1 = S2	LD(S1=S2)	= Relational box	Conducted when S1=S2 Not conducted when S1≠S2	Word	S1	Word S1, S2 WX, WY WR, WL WM, TC constant	•	•	•		•	•	5	(Note 1)
			AND(S1=S2)				S2								6	
			OR(S1=S2)													
	2	S1 <> S2	LD (S1<>S2)	<> Relational box	Conducted when S1≠S2 Not conducted when S1=S2	Word	S1		•	•	•		•	•	5	(Note 1)
			AND (S1<>S2)				S2								6	
			OR (S1<>S2)													
	3	S1 < S2	LD(S1<S2)	< Relational box	Conducted when S1<S2 Not conducted when S1≧S2	Word	S1		•	•	•		•	•	5	(Note 1)
			AND(S1<S2)				S2								6	
			OR(S1<S2)													
	4	S1 <= S2	LD (S1<=S2)	<= Relational box	Conducted when S1≦S2 Not conducted when S1>S2	Word	S1		•	•	•		•	•	5	(Note 1)
			AND (S1<=S2)				S2								6	
			OR (S1<=S2)													

(Note 1) The word size LD (S1S2) and AND (S1S2) have five steps, and OR (S1S2) haz six steps.

(2) Arithmetic instructions

Class	No.	Ladder symbol	Instruction symbol	Instruction name	Contents of processing	Size		Type	R7F4 DER	R7F3 ERR	R7F2 S	D	R7F1 V	R7F0 C	Step	Remarks
Assignment statement	1	d = S		Assignment Statement	Assignment d←S	Bit	d	Y, R, L, M	↕	•	•		•	•	3	
							S	X, Y, R, L, M constant								
						Word	d	WY, WR, WL, WM, TC	↕	•	•		•	•		
							S	WX, WY, WR, WL WM, TC, constant								
4-rule expression	1	d = S1 + S2		Binary Addition	d←S1 + S2	Word	d S1 S2	Word d WY, WR, WL, WM S1 S2 WX, WY, WR, WL, WM, TC, constant	•	•	•		↕	↕	4	
	2	d = S1 B+ S2		BCD Addition	d←S1 + S2	Word	d S1 S2		↕	•	•		•	↕	4	
	3	d = S1 − S2		Binary Subtraction	d←S1 − S2	Word	d S1 S2		•	•	•		↕	↕	4	
	4	d = S1 B− S2		BCD Subtraction	d←S1 − S2	Word	d S1 S2		↕	•	•		•	↕	4	
	5	d = S1 * S2		Binary Multiplication	d←S1 × S2	Word	d S1 S2		•	•	•		•	•	4	
	6	d = S1 B* S2		BCD Multiplication	d←S1 × S2	Word	d S1 S2		↕	•	•		•	•	4	
	7	d = S1 / S2		Binary Division	d←S1 ÷ S2 WR F016←remainder	Word	d S1 S2		↕	•	•		•	•	4	
	8	d = S1 B/ S2		BCD Division		Word	d S1 S2								4	

Constants for BCD computation must all be entered in hexadecimal number.

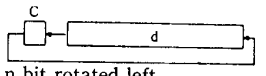
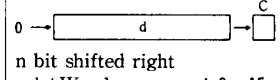
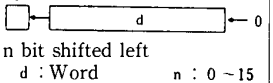
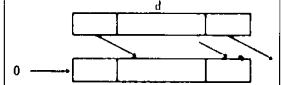
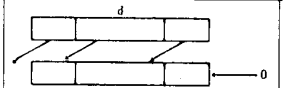
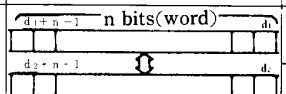
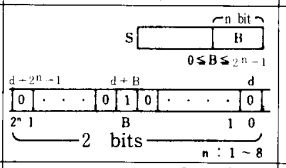
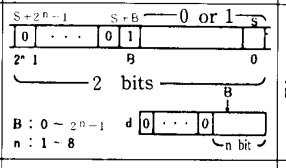
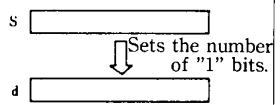
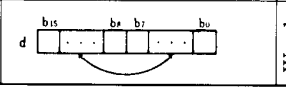
5

Arithmetic Instructions

Class	No.	Ladder symbol / Instruction symbol	Instruction name	Contents of processing	Size	Type	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C	Step	Remarks
Logical expression	1	d=S1 OR S2	OR	d←S1+S2	Bit	Bit d Y,R,L,M S1 S2 X,Y,R,L,M	•	•	•	•	•	4	
					Word							4	
	2	d=S1 AND S2	AND	d←S1 · S2	Bit	Word d WY,WR,WL,WM, TC S1 S2 WX,WY,WR,WL, WM,TC,constant	•	•	•	•	•	4	
					Word							4	
	3	d=S1 XOR S2	Exclusive OR	d←S1 ⊕ S2	Bit		•	•	•	•	•	4	
					Word							4	
Comparison Expression	1	d=S1==S2	= Comparison Expression	If S1=S2, then d←1 else d←0	Word	Word d Y,R,L,M S1 S2 WX,WY,WR,WL, WM,TC, constant	•	•	•	•	•	4	
	2	d=S1< >S2	≠ Comparison Expression	If S1≠S2, then d←1 else d←0	Word		•	•	•	•	•	4	
	3	d=S1 < S2	< Comparison Expression	If S1<S2, then d←1 else d←0	Word		•	•	•	•	•	4	
	4	d=S1<=S2	≦ Comparison Expression	If S1≦S2, then d←1 else d←0	Word		•	•	•	•	•	4	

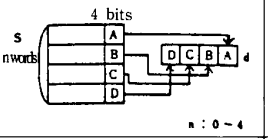
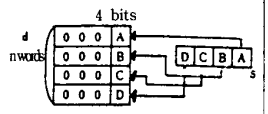
(3) Application instructions

Class	No.	Ladder symbol / Instruction symbol	Instruction name	Contents of processing	Size	Type	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C	Step	Remarks
Bit operation	1	B S E T (d , n)	Bit Set	d : Word n : 0 ~15	Word	Word d WY,WR,WL,WM, TC n WX,WY,WR,WL,WM WM,TC,constant	•	•	•	•	•	3	
	2	B R E S (d , n)	Bit Reset	d : Word n : 0 ~15	Word		•	•	•	•	•	3	
	3	B T S (d , n)	Bit Test	d : Word n : 0 ~15	Word		•	•	•	•	↕	3	
Shift/rotate	1	S H R (d , n)	Shift Right	n bit shifted right d : Word n : 0 ~15	Word		•	•	•	•	↕	3	
	2	S H L (d , n)	Shift Left	n bit shifted left d : Word n : 0 ~15	Word		•	•	•	•	↕	3	
	3	R O R (d , n)	Rotate Right	n bit rotated right d : Word n : 0 ~15	Word		•	•	•	•	↕	3	

Class	No.	Ladder symbol / Instruction symbol	Instruction name	Contents of processing	Size		Type	R/F4 DER	R/F3 ERR	R/F2 SD	R/F1 V	R/F0 C	Step	Remarks
Shift/rotate	4	ROL (d, n)	Rotate Left	 n bit rotated left d : Word n : 0 ~ 15	Word	d	Word d WY,WR,WL,WM,TC n WX,WY,WR,WL,WM TC, constant	•	•	•	•	↕	3	
	5	LSR (d, n)	Logical Shift Right	 n bit shifted right d : Word n : 0 ~ 15	Word	d n		•	•	•	•	↕	3	
	6	LSL (d, n)	Logical Shift Left	 n bit shifted left d : Word n : 0 ~ 15	Word	d n		•	•	•	•	↕	3	
	7	BSR (d, n)	BCD Shift Right	 BCD n digit shifted right d : Word n : 0~3	Word	d n		•	•	•	•	•	3	
	8	BSL (d, n)	BCD Shift Left	 BCD n digit shifted left d : Word n : 0~3	Word	d n		•	•	•	•	•	3	
Transfer	1	XCG (d1, d2, n)	Exchange	 n : 0 ~ 255	Bit	d1 d2	R,L,M	↕	•	•	•	•	4	
						n	WX,WY,WR,WL,WM TC, constant							
					Word	d1 d2	WR,WL,WM						4	
						n	WX,WY,WR,WL, WM,TC, constant							
Negation	1	NOT (d)	Not	$d \leftarrow \overline{d}$	Bit	d	Y,R,L,M	•	•	•	•	•	2	
					Word	d	WY,WR,WL,WM						2	
	2	NEG (d)	Negate	d ← − d	Word	d	WY,WR,WL,WM	•	•	•	•	•	2	
	3	ABS (d, S)	Absolute	d ← | S |, c ← S sign Positive : 1 Negative : 0	Word	d	WY,WR,WL,WM	•	•	•	•	↕	3	
						S	WX,WY,WR,WL, WM,TC, constant							
Conversion	1	BCD (d, S)	BCD	Converts the contents of S into BCD and sets in d. If binary data S≥10000, DER=1 If S≤9999, DER=0	Word	d S	Word d WY,WR,WL,WM S WX,WY,WR,WL, WM,TC, constant	↕	•	•	•	•	3	
	2	BIN (d, S)	BINARY	Converts the contents of S into binary and sets in d. If BCD data is abnormal, DER=1. If BCD data is normal, DER=0.	Word	d S		↕	•	•	•	•	3	
	3	DECO (d, S, n)	Decode	 n : 1 ~ 8	Bit	d	R,L,M	↕	•	•	•	•	4	
						S	WX,WY,WR,WL, WM,TC, constant							
						n	Constants 1 to 8							
	4	ENCO (d, S, n)	Encode	 B : 0 ~ 2n−1 n : 1 ~ 8	Bit	d	WY,WR,WL,WM	↕	•	•	•	↕	4	
						S	R,L,M							
						n	Constants 1 to 8							
Application instruction	1	BCU (d, S)	Bit Count	 Sets the number of "1" bits.	Word	d	WY,WR,WL,WM	•	•	•	•	•	3	
						S	WX,WY,WR,WL, WM,TC, constant							
	2	SWAP (d)	Swap		Word	d	WY,WR,WL,WM	•	•	•	•	•	2	

5

Application Instructions

Class	No.	Ladder symbol / Instruction symbol	Instruction name	Contents of processing	Size		Type	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C	Step	Remarks
Application instructions	3	UNIT (d, S, n)	Unit		Word	d	WY,WR,WL,WM	↕	•	•	•	•	4	
						S	WR,WL,WM							
						n	Constants 0 to 4							
	4	DIST (d, S, n)	Distribute		Word	d	WR,WL,WM	↕	•	•	•	•	4	
						S	WX,WY,WR,WL, WM,TC,constant							
						n	Constants 0 to 4							

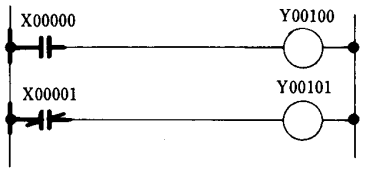
(4) Control instructions

Class	No.	Ladder symbol / Instruction symbol	Instruction name	Contents of processing	Size		Type	R7F4 DER	R7F3 ERR	R7F2 SD	R7F1 V	R7F0 C	Step	Remarks
Control instructions	1	END	END	Indicates the end of normal scan. Returns to the beginning of normal scan and executes the normal scan.				•	•	•	•	•	1	
	2	CEND(S)	Conditional End	Returns to the beginning of normal scan and executes it when S is 1, or executes the next instruction when S is 0.	Bit	S	X,Y,R,L,M	•	•	•	•	•	2	
	3	JMP n	Jump	Jumps to LBLn of the same code number.	Word	n	constant 0~255	•	①	•	•	•	2	
	4	CJMP n(S)	Conditional Jump	Jumps to LBLn if S=1, and does not jump if S=0.	Word	n	constant 0~255	•	①	•	•	•	3	
						s	X,Y,R,L,M							
	5	LBL n	Label	Indicates the jump destination of JMP and CJMP instructions of the same code number n.	Word	S	constant0~255	•	•	•	•	•	1	
	6	CAL n	Call	Calls the subroutine SBn of the same code number.	Word	n	constant0~99	•	①	•	•	•	2	
	7	SB n	Subroutine	Indicates the start of the subroutine of the code number n.	Word	n	constant0~99	•	•	•	•	•	1	
	8	RTS	Return Subroutine	Returns from subroutine.				•	•	•	•	•	1	
	9	INT n	Interrupt	Indicates the start of interrupt scan of code number n.	Word	n	constant0~31	•	•	•	•	•	1	
	10	RTI	Return Interrupt	Returns from interrupt scan.				•	•	•	•	•	1	

5.3 Basic Instructions

Each basic instruction is detailed below.

(1) Start Logical Operation (⊣⊢ , ⊣/⊢ /LD, LDI)

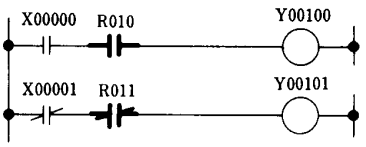


LD X00000
OUT Y00100
LDI X00001
OUT Y00101

Circuit operation:

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

(2) Series Connection of Contact (⊣⊢ , ⊣/⊢ /AND, ANI)



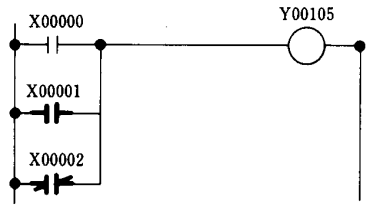
LD X00000
AND R010
OUT Y00100
LDI X00001
ANI R011
OUT Y00101

Note: The edge detections DIF and DFN detailed later can be specified to AND, but cannot be specified to ANI.

Circuit operation:

- Output Y00100 is on only when both input X00000 and R010 are on.
- Output Y00101 is on only when both input X00001 and R011 are off.

(3) Parallel Connection of Contact (⊣⊢ , ⊣/⊢ /OR, ORI)

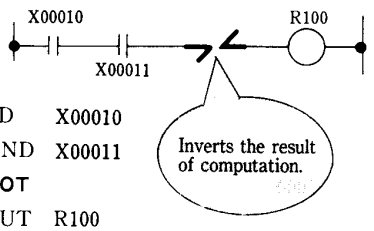


LD X00000
OR X00001
ORI X00002
OUT Y00105

Circuit operation:

- Output Y00105 is on when either input X00000 or X00001 is on, or when X00002 is off.

(4) Negation (→ ← /NOT)



LD X00010
AND X00011
NOT
OUT R100

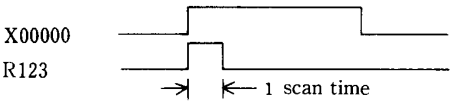
Circuit operation:

- Output R100 is off when both input X00010 and X00011 are on. Otherwise, output R100 is on.

(5) Rising edge detection (AND DIF, OR DIF)

LD X00000
AND DIF 0 ..... Note: Cannot be specified to ANI.
OUT R123

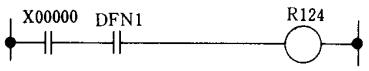
Time chart:



Circuit operation:

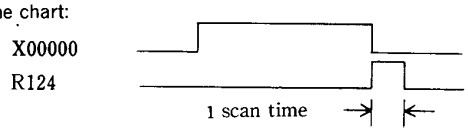
- Output R123 is on for a period of 1 scan time when input X00000 turns from off to on.

(6) Falling Edge Detection (AND DFN, OR DFN)



LD X00000
AND DFN1 ..... Note: Cannot be specified to ANI.
OUT R124

Time chart:

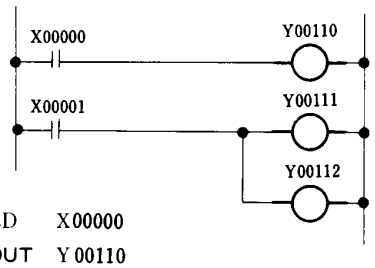


Circuit operation:

- Output R124 is on for a period of 1 scan time when input X00000 turns from on to off.

5

(7) Output (—○—/OUT)



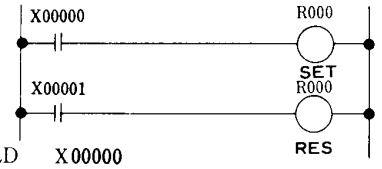
LD X00000
OUT Y00110
LD X00001
OUT Y00111
OUT Y00112

Circuit operation:

○ Output Y00110 is on when input X00000 is on.

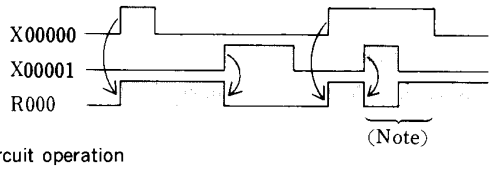
○ Outputs Y00111 and Y00112 are on when input X00001 is on.

(8) Set and Reset (SET, RES)



LD X00000
SET R000
LD X00001
RES R000

Time chart:



Circuit operation

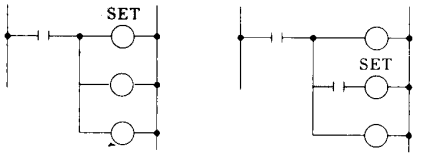
○ Output R000 is on when input X00000 is on.

○ Output R000 is off when input X00001 is on.

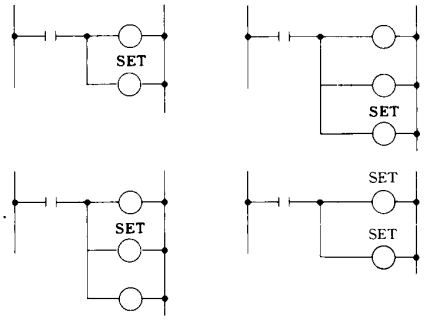
○ When the conditions of SET and RES are established at the same time as indicated by (Note) above, the one programmed later is executed first.

Restriction:

○ When using a set or reset coil in parallel with other coils, program must be written so that the coil is located at the uppermost stage or that a contact is provided just before the coil.



NG examples



(9) Master Control (MCS, MCR)

Master Control Set (MCS) and Reset (MCR) instructions are used in such circuit as shown below.

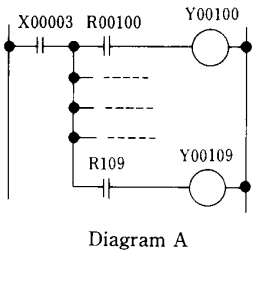


Diagram A

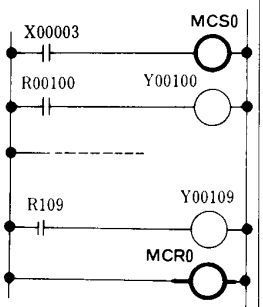
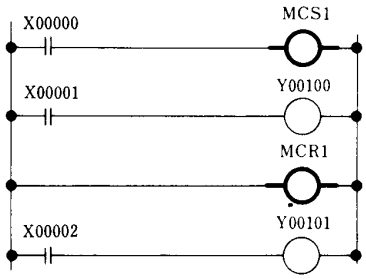


Diagram B

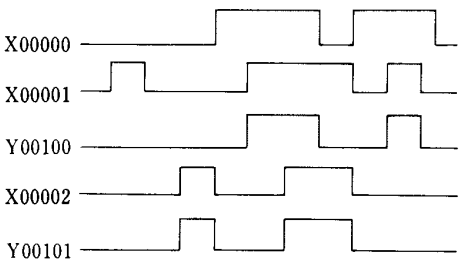



```
LD    X00000
MCS1
LD    X00001
OUT   Y00100
MCR1
LD    X00002
OUT   Y00101
```

Circuit operation:

- ○ Output Y00100 turns on/off according to input X00001 when input X00000 is on.
- ○ Output Y00100 is always off irrespective of input X00001 when input X00000 is off.

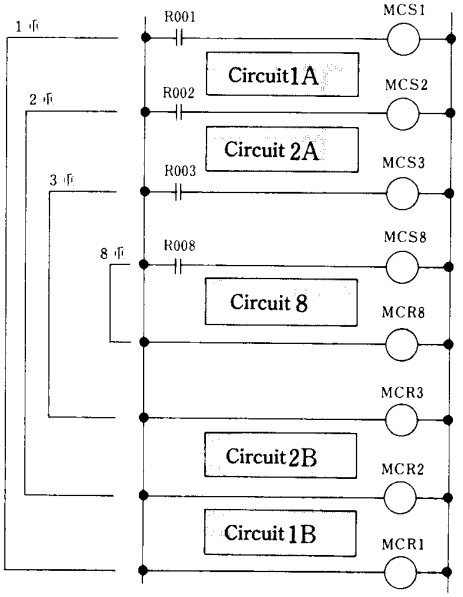
Time chart:



Notes:

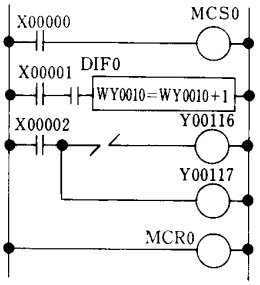
1. Master control can be programmed to have up to 8 loops. The same master control number can be assigned to MCS and MCR in pairs for easy programming.

Master control instruction is numbered 0 to 49. But the usable number of master control instructions is not restricted and the same number can be used doubly.

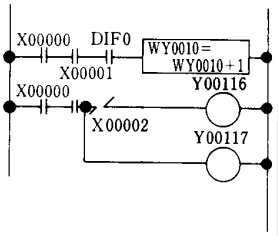


2. The following diagrams show the difference in operation between when the master control instruction is used (Circuit A) and when it is not used (Circuit B) in the program containing edge detection and negation.

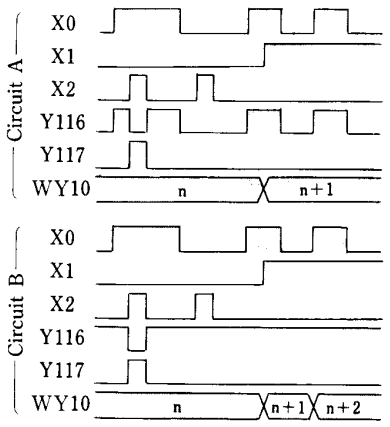
Circuit A



Circuit B

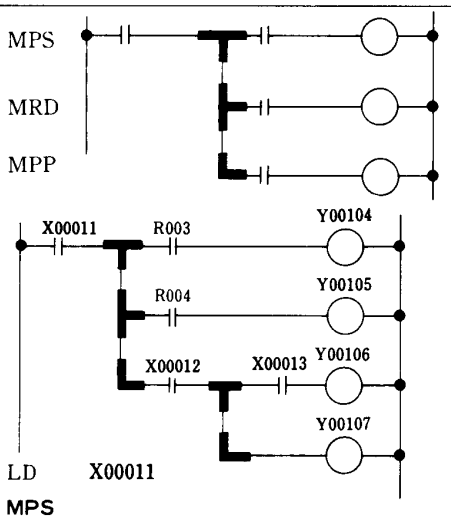


Time chart:



5

(10) Store, Read or Clear Computation Result



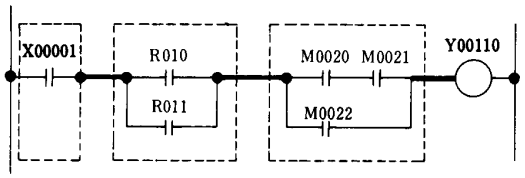

```
LD    X00011
MPS
AND   R003
OUT   Y00104
MRD
AND   R004
OUT   Y00105
MPP
AND   X00012
MPS
AND   X00013
OUT   Y00106
MPP
OUT   Y00107
```

○ MPS stores the result of computation just before this instruction.

○ MRD reads the result of computation stored by the MPS instruction and continues computation according to the next instruction.

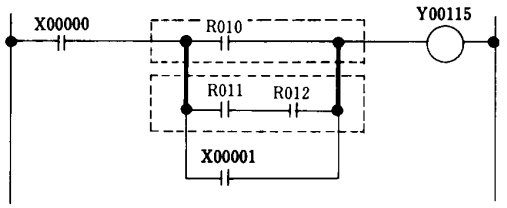
○ MPP reads the result of computation stored by the MPS instruction and continues computation according to the next instruction. The result of computation stored by the MPS instruction is cleared on execution of the MPP instruction.

(11) Logical Block Series Connection (ANB)



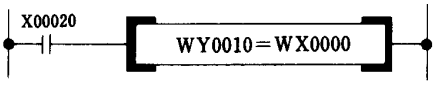

```
LD    X00001
LD    R010
OR    R011
ANB
LD    M0020
AND   M0021
OR    M0022
ANB
OUT   Y00110
```

(12) Logical Block Parallel Connection (ORB)




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

(13) Processing Box Start and End ([,])

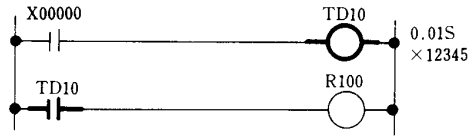



```
LD      X00020
[
WY0010=WX0000
]
```

Circuit operation:

○ Computation in the processing box is carried out when input X00020 is on.

(14) On-Delay Timer (TDn ts)

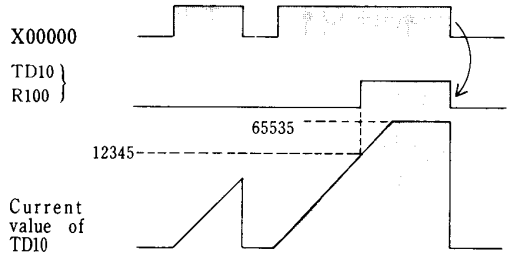


LD X00000
OUT TD10 0.01S 12345
LD TD10
OUT R100

Circuit operation:

○ TD10 and then R100 turn on when the current value exceeds the set time value (123.45 sec) after input X00000 was on.

Time chart:



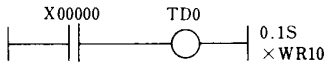
Change of current time value:

○ If current time value is changed by an arithmetic instruction or other, the new value is in effect immediately.

Change of set time value during run operation:

○ When the set time value of timer is changed during run operation, the new time base and set time value are in effect immediately.

○ To change the set time value during program execution, specify WX, WY, WR, WL or WM for the set time value. Processing of the on-delay timer reads a new set time value at each scan.



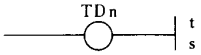
Timer number n: 0 to 255

Tine base t: 0.01 sec, 0.1 sec, 1.0 sec

Set time value s: WX, WY, WR, WL, WM, constant

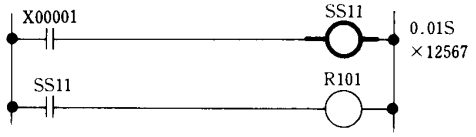
Notes:

1. The time base of 0.01 sec must be used within a timer number range of 0 to 63.



2. Up to 256 points (0 to 255) are usable in total of TD and SS. However, the same number must not be used for both TD and SS (for example, TD1 and SS1 cannot be used at a time).

(15) Single Shot (SSn ts)



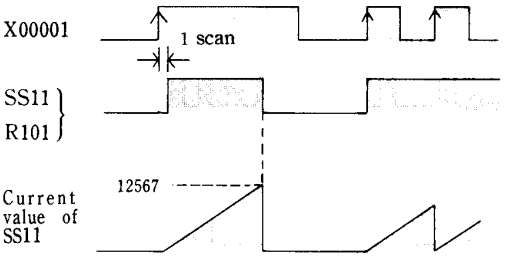
LD X00001
OUT SS11 0.01S 12567
LD SS11
OUT R101

Circuit operation:

○ Output R101 is on at 1 scan after the leading edge of input X00001 (from off to on) was detected. This output is off when current value is larger than the set time value (125.67 sec), and the current value becomes 0.

○ If input X00001 is on at the first scan during run startup, SS11 does not detect the rising edge; thus, output R101 remains off.
If input X00001 is off at the first scan and on at the second scan, then SS11 detects the rising edge.

Time chart:



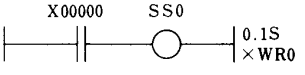
Change of current time value:

○ If current value is changed by an arithmetic instruction or other, the new value is in effect immediately.

Change of set value during run operation:

○ When the set time value of the timer is changed during run operation, the new time base and set time value are in effect immediately.

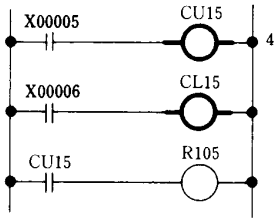
○ To change the set time value during program execution, specify WX, WY, WR, WL, or WM for the set time value.
Processing of the single shot reads a new set time value at each scan.



Note: Up to 256 points (0 to 255) are usable in total of TD and SS. However, the same number must not be used for both TD and SS (for example, TD1 and SS1 cannot be used at a time).

5

(16) Counter (CUn S)



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

Circuit operation:

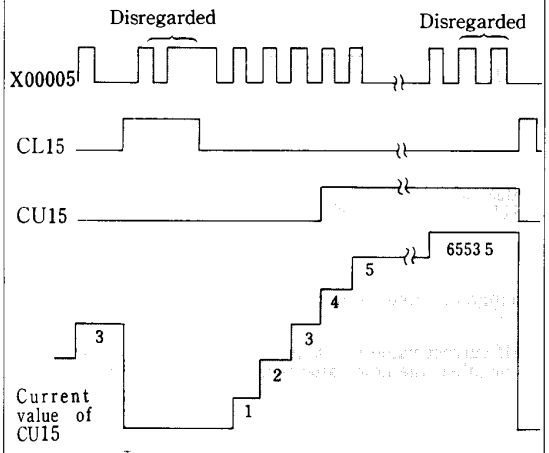
①On detection of the rising edge (from off to on) of start condition X00005 of counter CU15, the current value of this counter is incremented by 1. CU15 is on when the current value is larger than the set value(4).

②When CL15 is on, U15 is off and the current value becomes "0." If the set value is "0" at that time, current value≧set value=0. However, CU15 remains off.

③The rising edge of start condition X00005 of CU15 is disregarded while CL15 is on.

④The rising edge is not detected even if start condition X00005 is on at the first scan upon run startup.

Time chart:



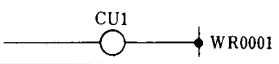
Change of current time value:

○If current value is changed by an arithmetic instruction or other, the new value is in effect immediately.

Change of set value during run operation:

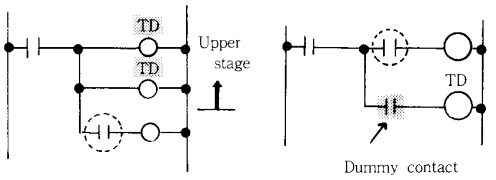
○When the set value of counter is changed during run operation, the new value is in effect immediately.

○To change the set value during program execution, specify WX, WY, WR, WL or WM for the set time value. Processing of the counter reads a new set time value at each scan.

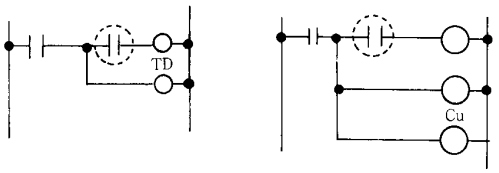


CAUTION

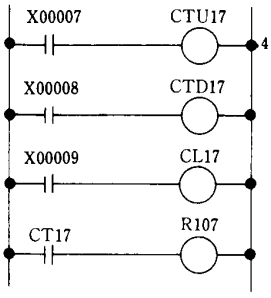
When using a timer or counter coil in pallallel with other contact, program must be written so that the coil is located at the upper stage than the contact or that a contact is provided just before the coil.



NG examples



(17) Up/Down Counter Down (CTDn S) and Up (CTUn S)




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

Circuit operation:

①When the rising edge (transition from off to on) of start condition X00007 of CTU17 is detected, the current value of TC17 is incremented by 1.

②When the rising edge (from off to on) of start condition X00008 of CTD17 is detected, the current value of TC17 is decremented by 1.

③CT17 is on when the current value is larger than the set value within 65535.CT17 is off when the current value is between 0 and "set value-1."

④The rising edge of start conditions X00007 and X00008 is disregarded while CL17 is on.

⑤The rising edge of start condition X00007 of CTU is disregarded when current value TC17 is 65535.

⑥The rising edge of start condition X00008 of CTD is disregarded when current value TC17 is 0.

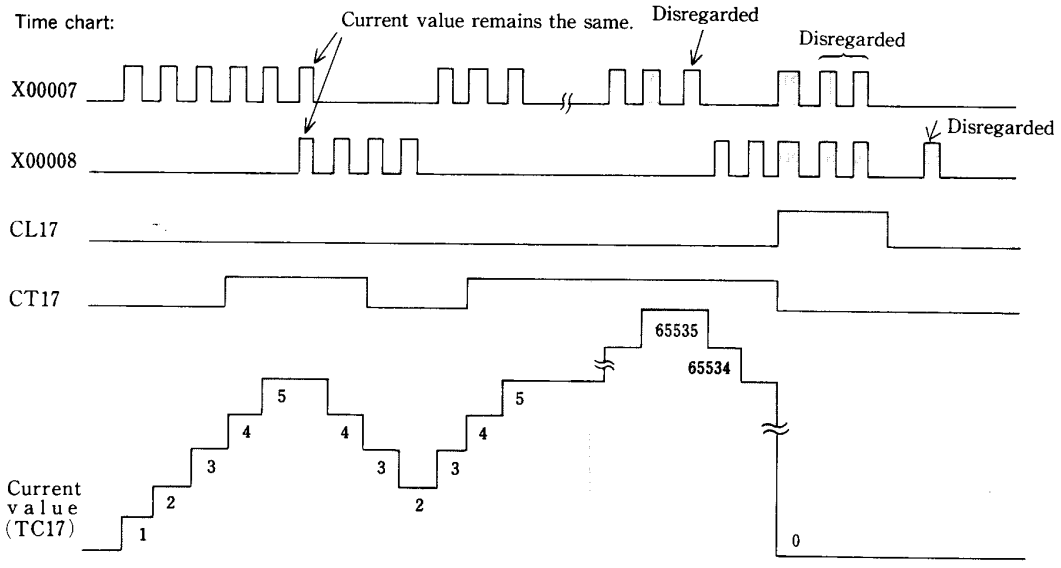
Note: When the set value is 0 and the current value TC17 is between 0 and 65535, CT17 is always on.However, when CL17 is on, CT17 is off.

Note: Timer accuracy
Timer accuracy varies with time base.

Time base	Accuracy
0.01sec	-10 msec to +1 scan time
0.1sec	-100 msec to+1 scan time
1 sec	-1 sec to+1 scan time

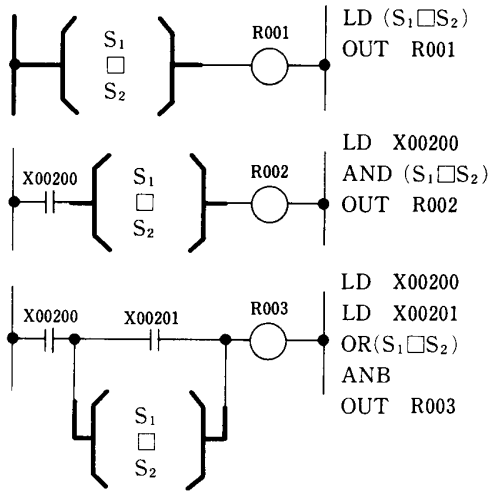
1 scan time indicated the time taken for a single cycle of scan (normal, periodic or interrupt scan) under which the timer is used.

Time chart:



5

(18) Relational box (=, < >, <, < =)



Function:

- The relational box turns on or off according to the result of comparison of S1 and S2 like "a" contact.
- The compare instruction □ specifies the binary comparisons shown below between two 16-bit words.

  =, < >, <, < =

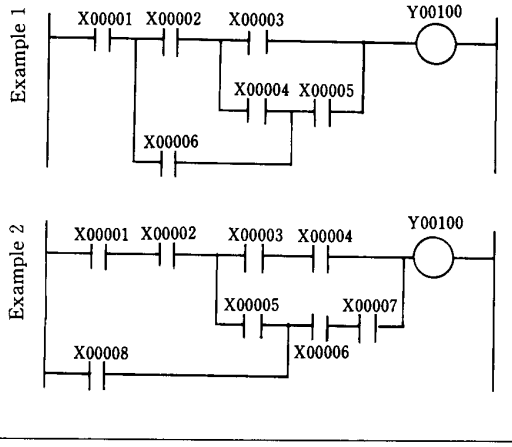
Each operation is listed below.

Instruction	Word length of S1 and S2	Comparison	Result
=	o Word (16 bits)	$S_1 = S_2$	ON
		$S_1 \neq S_2$	OFF
< >		$S_1 \neq S_2$	ON
		$S_1 = S_2$	OFF
<		$S_1 < S_2$	ON
		$S_1 \geqq S_2$	OFF
< =		$S_1 \leqq S_2$	ON
		$S_1 > S_2$	OFF

Note: Both S1 and S2 must be a word.

Reference:

Example of the program which can be represented by a ladder diagram, but cannot be expressed by instruction words.



5.4 Arithmetic Instructions

(1) Assignment Statement (d=s)

Function:

• d=s

①The contents of the right side are assigned in the left side d.

d ← s

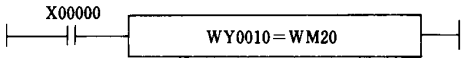
Note:

①The left and right sides must have either one of the following combinations.

d	s
Bit	Bit
Word	Word

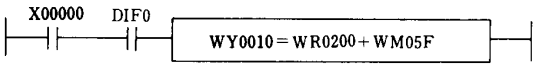
Example:

①Set the contents of WM20 in WY0010.



LD X00000
[
WY0010=WM20
]

(2) Binary Addition (d=S1+S2)



LD X00000
AND DIF0
[
WY0010=WR0200+WM05F
]

Circuit operation:

○ WR0200 and WM05F are added in binary and the result is stored in WY0010.

Function:

○ d=S1+S2

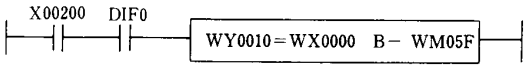
①The result of S1 and S2 addition is stored in d as binary data.

②Value 1 is stored in C(R7FO) when S1 and S2 are unsigned binary data and a carry occurs in d; otherwise, 0 is stored.

ⓐWhen d, S1 and S2 are word data, the contents of d, S1 and S2, or hexadecimal H0000 to HFFFF denote decimal 0 to 65535. If the result of S1 and S2 addition is within the range of H0000 to HFFFF, 0 is stored in C, and 1 is stored in C when HFFFF is exceeded.

5

(3) BCD Addition (d=S1 B+S2)



LD X00200
AND DIF0
[
WY0010=WX0000 B+ WM05F
]

Circuit operation:

○ WX0000 and WM05F are added in BCD and the result is stored in WY0010.

Function:

○ d=S1 B+S2

①The contents of S1 and S2 are added as BCD data and the result is stored in d.

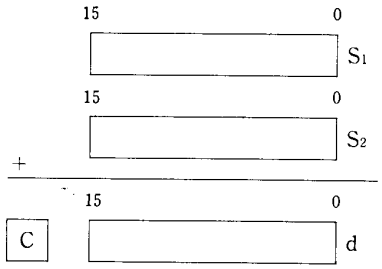
②Carry is set in C.

③When the contents of S1 or S2 are abnormal as BCD data, DER is set to 1, and computation is not performed and no output is made to d. C is held in the previous status.

④When the contents of both S1 and S2 are normal as BCD data, DER is set to 0 and computation is carried out.

(C denotes the special internal output R7F0.)
(DER is the special internal output R7F4.)

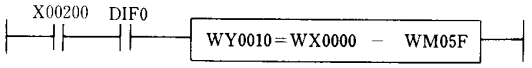
$d \leftarrow S_1 + S_2$



Combination of d, S1 and S2

d	S1	S2
Word	Word	Word

(4) Binary Subtraction (d=S1−S2)



LD X00200
AND DIF0
[
WY0010=WX0000 - WM05F
]

Circuit operation:

○ WM05F is subtracted form WX0000 in binary and the result is stored in WY0010.

Function:

○ d=S1−S2

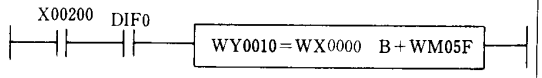
①S2 is subtracted from S1 in binary and the result is stored in d.

②Value 1 is stored in C (R7F0) when S1 and S2 are unsigned binary data and a borrow occurs in d; otherwise, 0 is stored.

ⓐWhen d, S1 and S2 are word data, the contents of d, S1 and S2, or hexadecimal H0000 to HFFFF, denote decimal 0 to 65535.

If S2 is subtracted from S1 when S1<S2, 1 is stored in C.
If S2 is subtracted from S1 When S1≧S2, 0 is stored in C.

(5) BCD Subtraction (d=S1B−S2)



LD X00200
AND DIF0
[
WX0010=WX0000 B− WM05F
]

Circuit operation:

○ WM05F is subtracted from WX0000 in BCD and the result is stored in WY0010.

Function:

○ d=S1 B−S2

①The contents of S2 are subtracted from the contents of S1 as BCD data and the result is stored in d.

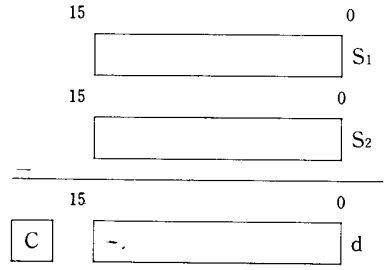
②Borrow is set in C.

③When the contents of S1 or S2 are abnormal as BCD data, DER is set to 1, and computation is not carried out and no output is made to d.C is held in the previous status.

④When the contents of both S1 and S2 are normal as BCD data, DER is set to 0 and computation is carried out.

(C denotes the special internal output R7F0.)
(DER is the special internal output R7F4.)

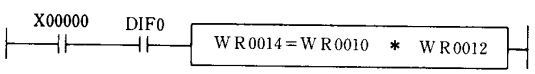
d ← S1 − S2



Combination of d, S1 and S2

d	S1	S2
Word	Word	Word

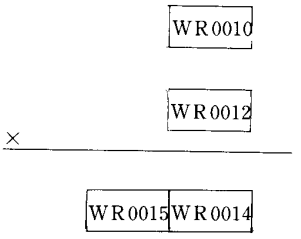
(6) Binary Multiplication (d=S1＊S2)



LD X00000
AND DIF0
[
WR0014=WR0010 * WR0012
]

Circuit operation:

○ WR0010 and WR0012 are multiplied in binary and the result is stored in WR0014 and WR0015.



CAUTION

Even when each data is small in WR0010＊WR0012 computation, the product is always stored in WR0014 and WR0015.Pay attention not to use WR0005 for other computation.

Function:

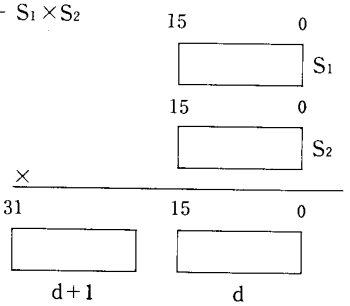
○ d=S1＊S2

①Multiplication of S1 and S2 is carried out as binary data and the result is stored in d+1 (high-order word of product) and d (low-order word of product).

②Value d+1 must not exceed the maximum value (RW3FF, WL7, WL1007, WMFF).
If exceeded, DER is set to 1 and only low-order word of the product is stored in d.
If not exceeded, DER is set to 0 and the result is stored in d and d+1.

(DER is the special internal output R7F4.)

d+1, d ← S1 × S2

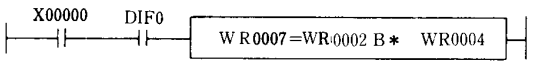


Combination of d, S1 and S2

d	S1	S2
Word	Word	Word

5

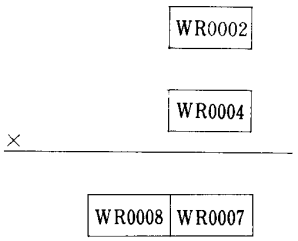
(7) BCD Multiplication (d=S1 B*S2)



LD X00000
AND DIF0
[
WR0007=WR0002 B* WR0004
]

Circuit operation:

○ WR0002 and WR0004 are multiplied in BCD and the product is stored in WR0007 and WR0008.



CAUTION

Even when each data is small in WR0002*WR0004 computation, the product is always stored in WR0007 and WR0008.Pay attention not to use WR0008 for other computation.

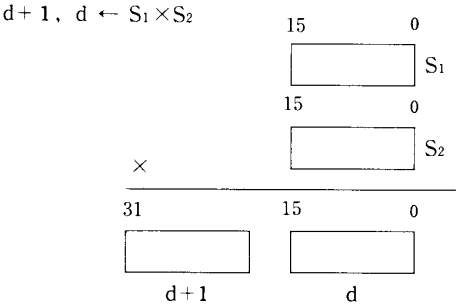
Function:

○ d=S1B*S2

①Multiplication of S1 and S2 is carried out as BCD data and the product is stored in d+1 (high-order word product) and d (low-order word of product).

②Value d+1 must not exceed the maximum value (WR3FF, WL7, WL1007, WMFF).

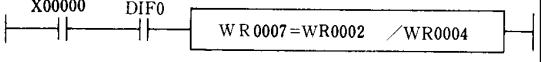
If exceeded, DER is set to 1 and only the low-order word of the product is stored in d.
If not exceeded, DER is set to 0 and the product is stored in d and d+1.
(DER denotes the special enternal output R7F4.)



Combination of d, S1 and S2

d	S1	S2
Word	Word	Word

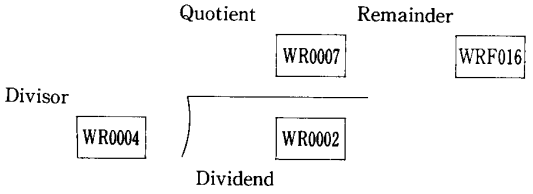
(8) Binary Division (d=S1/S2)



LD X00000
AND DIF0
[
WR0007=WR0002 /WR0004
]

Circuit operation:

○ WR2 is divided by WR4 in binary, and the quotient is stored in WR0007 and the remainder is stored in WRF016.



Function:

○ d=S1/S2

①S1 is divided by S2 as binary data.The quotient is stored in d and the remainder is stored in WRF016.

②When divisor S2 is 0, DER is set to 1 and no computation is performed.(No quotient and remainder are output.) When divisor S2 is not 0, DER is set to 0 and division is carried out.

(DER is the special internal output R7F4.)

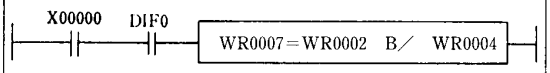
(a) Word

d←S1÷S2 (quotient)
WRF016←S1 mod S2 (remainder)

Combination of d, S1 and S2

d	S1	S2
Word	Word	Word

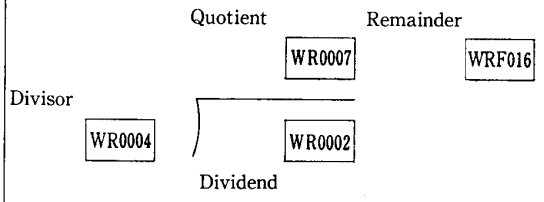
(9) BCD Division (d=S1 B/S2)



LD X00000
AND DIF0
[
WR0007=WR0002 B/ WR0004
]

Circuit operation:

○WR2 is divided by WR4 in binary, and the quotient is stored in WR7 and the remainder is stored in WRF016.



Function:

○ d=S1 B/S2

①S1 is divided by S2 as BCD data.The quotient is stored in d and the remainder is stored in WRF016.

②When divisor S2 is 0, DER is set to 1, and computation, d output and remainder output do not occur.

③When the contents of S1 or S2 are abnormal, DER is set to 1, and none of computation, d output and remainder output are made.

④When the contents of both S1 and S2 are normal, DER is set to 0, and computation is carried out.

(DER is the special internal output R7F4.)

(a) Word

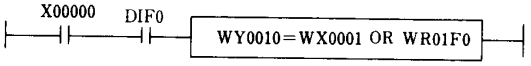
d←S1÷S2 (quotient)

WRF016←S1 mod S2 (remainder)

Combination of d, S1 and S2

d	S1	S2
Word	Word	Word

(10) OR(=S1 OR S2)



LD X00000
AND DIF0
[
WY0010=WX0001 OR WR01F0
]

Circuit operation:

○WX0001 and WR01F0 are ORed and the result is output to WY0010.

WX0001	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
WR01F0	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1
WY0010	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1

Function:

○ d=S1 OR S2

①The contents of S1 and S2 are ORed, and the result is stored in d.

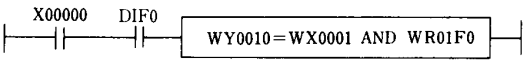
d←S1+S2

Combination of d, S1 and S2

d	S1	S2
Bit	Bit	Bit
Word	Word	Word

5

(11) AND (d=S1 and S2)



LD X00000
AND DIF0
[
WY0010=WX0001 AND WR01F0
]

Circuit operation:

○WX0001 and WR01F0 are ANDed and the result is output to WY0010.

WX0001	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
WR01F0	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1
WY0010	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Function:

○ d=S1 AND S2

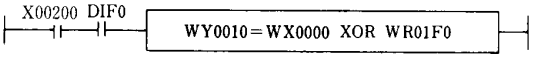
①The contents of S1 and S2 are ANDed, and the result is stored in d.

d←S1 · S2

Combination of d, S1 and S2

d	S1	S2
Bit	Bit	Bit
Word	Word	Word

(12) Exclusive OR (d=S1 XOR S2)



LD X00200
AND DIF0
[
WY0010=WX0000 XOR WR01F0
]

Circuit operation:

○WX0000 and WR01F0 are exclusively ORed and the result is output to WY0010.

WX0000	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
WR01F0	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1
WY0010	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0

Function:

○ d=S1 XOR S2

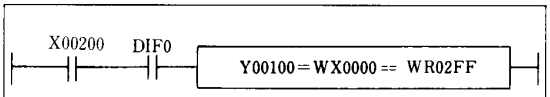
①The contents of S1 and S2 are exclusively ORed, and the result is stored in d.

d←S1+S2

Combination of d, S1 and S2

d	S1	S2
Bit	Bit	Bit
Word	Word	Word

(13) "=" comparison expression (d=S1==S2)



LD X00200
AND DIF0
[
Y00100=WX0000 == WR02FF
]

Circuit operation:

○ WX0000 and WR02FF are compared to check if they are equal, and the result is output to Y00100.

Function:

○ d=S1==S2

①The contents of S1 and S2 are compared, and 1 is set in d when S1=S2.

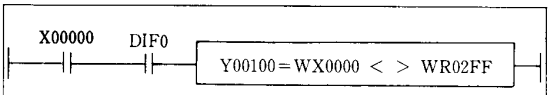
0 is set in d when S1≠S2.

IF $S_1 = S_2$ THEN d ← 1
ELSE d ← 0

Combination of d, S1 and S2

d	S1	S2
Bit	Word	Word

(14) "≠" comparison expression (d=S1<>S2)



LD X00000
AND DIF0
[
Y00100=WX0000 < > WR02FF
]

Circuit operation:

○ WX0000 and WR02FF are compared to check if they are not equal, and the result is output to Y00100.

Function:

○ D=S1<>S2

①The contents of S1 and S2 are compard, and 1 is set in d when S1≠S2.

When S1=S2, 0 is set in d.

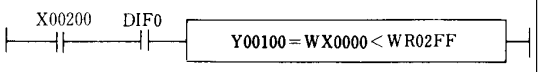
IF $S_1 = S_2$ THEN d ← 1
ELSE d ← 0

Combination of d, S1 and S2

d	S1	S2
Bit	Word	Word

5

(15) "<" comparison expression (d=S1<S2)



LD X00200
AND DIF0
[
Y00100=WX0000<WR02FF
]

Circuit operation:

○WX0000 and WR02FF are compared to check if WX0000 is equal to or smaller than WR02FF, and the result is output to Y00100.

Function:

○ D=S1<S2

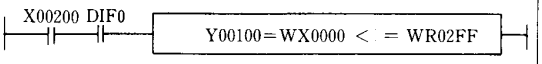
①The contents of S1 and S2 are compared, and 1 is set in d when S1<S2.
When S1≧S2, 0 is set in d.

IF $S_1 < S_2$ THEN d ← 1
ELSE d ← 0

Combination of d, S1 and S2

d	S1	S2
Bit	Word	Word

(16) "≦" comparison expression (d=S1<=S2)



LD X00200
AND DIF0
[
Y00100=WX0000 < = WR02FE
]

Circuit operation:

○WX0000 and WR02FF are compared to check if WX0000 is equal to or smaller than WR02FF, and the result is output to Y00100.

Function:

○ d=S1<=S2

①The contents of S1 and S2 are compared, and 1 is set in d when S1≦S2.

When S1>S2, 0 is set in d.

IF $S_1 \leqq S_2$ THEN d ← 1
ELSE d ← 0

Combination of d, S1 and S2

d	S1	S2
Bit	Word	Word

5.5 Application Instructions

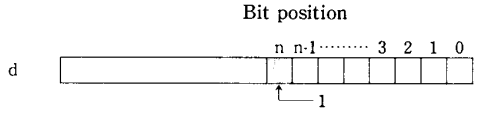
(1) Bit Set (BSET (d, n))

Function:

○ BSET (d, n)

①Value 1 is set in bit n of d.

②The contents of other bits are saved.

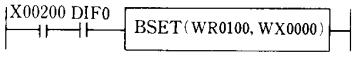


Combination of d and n

d \ n	Word (WX, WY, WR, WL, WM, TC)	Word (constant)
Word (WY, WR, WL, TC)	Bit position is specified by contents of low-order 4 bits of n (0 to 15).	Values 0 to 15 are specifiable for n (bit position).

Example:

Set 1 in the bit specified by WX000 of WR0100.




```
LD      X00200
AND     DIF0
[
BSET (WR0100, WX0000)
]
```

When WR0100 is H0000:

(a)When WX0000 is HFFF0, the low-order 4 bits of WX0000 are 0000, which is 0 in decimal. Because 1 is set in bit 0, WR0100, is H0001.

(b)When WX0000 is HFFF3, the low-order 4 bits of WX0000 are 0011, which is 3 in decimal. Because 1 is set in bit 3, WR0100 is H0008.

(c)When WX0000 is H0006, the low-order 4 bits of WX0000 are 0110, which is 6 in decimal. Because 1 is set in bit 6, WR0100 is H0040.

(d)When WX0000 is HF0C8, the low-order 4 bits of WX0000 are 1000, which is 8 in decimal. Because 1 is set in bit 8, WR0100 is H0100.

(e)When WX0000 is HFFFF, the low-order 4 bits of WX0000 are 1111, which is 15 in decimal. Because 1 is set in bit 15, WR0100 is H80000.

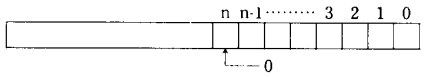
(2) Bit Reset (BRES (d, n))

Function:

○ BRES (d, n)

①Value 0 is set in bit n of d.

②The contents of other bits are saved.

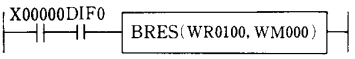


Combination of d and n

d \ n	Word (WX, WY, WR,WL, WM, TC)	Word (constant)
Word (WY, WR, WL, WM, TC)	Bit position is specified by contents of low-order 4 bits of n (0 to 15).	Values 0 to 15 are specifiable for n (bit position).

Example:

Set 0 in the bit specified by WM000 of WR0100.




```
LD      X00000
AND     DIF0
[
BRES (WR0100, WM000)
]
```

When WR0100 is HFFFF:

(a)When WM000 is HFFF0, the low-order 4 bits of WM000 are 0000, which is 0 in decimal.Because 0 is set in bit 0, WR0100 is HFFFE.

(b)When WM000 is HFFF3, the low-order 4 bits of WM000 are 0011, which is 3 in decimal.Because 0 is set in bit 3, WR0100 is HFFF7.

(c)When WM000 is H0006, the low-order 4 bits of WM000 are 0110, which is 6 in decimal.Because 0 is set in bit 6, WR0100 is HFFBF.

(d)When WM000 is HF0C8, the low-order 4 bits of WM000 are 1000, which is 8 in decimal.Because 0 is set in bit 8, WR0100 is HFEFF.

(e)When WM000 is HFFFF, the low-order 4 bits of WM000 are 1111, which is 15 in decimal.Because 0 is set in bit 15, WR0100 is H7FFF.

5

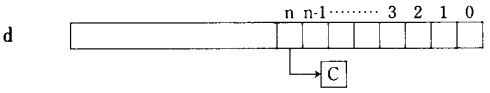
(3) Bit Test (BTS (d, n))

Function:

○ BTS (d, n)

①The contents of bit n of d are checked. When bit n is 1, C is set to 1, and it is set to 0 when bit n is 0.

②The contents of d are saved.
(C is the special internal output R7F0.)

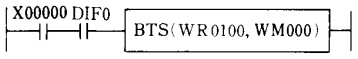


Combination of d and n

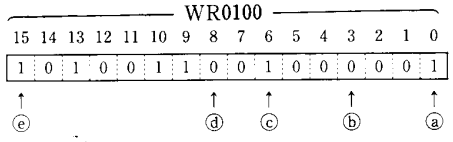
d \ n	Word (WX, WY, WR, WL, WM, TC)	Word (constant)
Word (WY, WR, WL, WM, TC)	Bit position is specified by contents of low-order 4 bits of n (0 to 15).	Values 0 to 15 are specifiable for n (bit position).

Example:

Check bit value 0 or 1 of WR0100.




```
LD      X00000
AND     DIF0
[
BTS(WR0100, WM000)
]
```



When WR0100 is HA641:

ⓐWhen WM000 is H0000, the lower-order 4 bits are 0000, which is 0 in decimal.Value 1, the contents of bit 0, is set in C.

ⓑWhen WM000 is HFFF3, the lower-order 4 bits are 0011, which is 3 in decimal.Value 0, the contents of bit 3, is set in C.

ⓒWhen WM000 is H0006, the lower-order 4 bits are 0110, which is 6 in decimal.Value 1, the contents of bit 6, is set in C.

ⓓWhen WM000 is HF0C8, the lower-order 4 bits are 1000, which is 8 in decimal.Value 0, the contents of bit 8, is set in C.

ⓔWhen WM000 is H00FF, the lower-order 4 bits are 1111, which is 15 in decimal.Value 1, the contents of bit 15, is set in C.

(4) Shift Right (SHR (d, n))

Function:

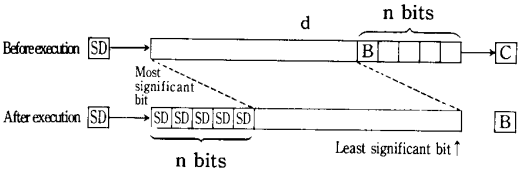
○ SHR (d, n)

①The contents of d (H0000 to HFFFF) are shifted n bits to the right.

②The contents of SD (0 or 1) are stored for n bits from the most significant bit.

③The contents of n bits (0 or 1) form the least significant bit are stored in C.

④When n (shift amount) is 0, no shift is performed.C retains the previous status.
(C is the special internal output R7F0, and SD is the special internal output R7F2.)

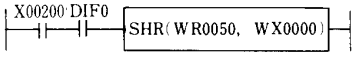


Combination of d and n

d \ n	Word (WX, WY, WR, WL, WM, TC)	Word (constant)
Word (WY, WR, WL, WM, TC)	The contents of low-order 4 bits of n (0 to 15) are used as the shift amount.	Values 0 to 15 are specifiable for n (shift amount).

Example:

Shift WR0050 to the right for the bits specified by WX0000.




```
LD      X00200
AND     DIF0
[
SHR (WR0050, WX0000)
]
```

ⓐWhen WR0050 is H1234, WX0000 is HFFF3 and SD is 1, the low-order 4 bits of WX0000 are 0011 (3 in decimal).
The contents are shifted 3 bits to the right. Thus, WR0050 becomes HE246 and C becomes 1.

ⓑWhen WR0050 is H1234, WX0000 is H0006 and SD is 1, the low-order 4 bits of WX0000 are 0110 (6 in decimal).
The contents are shifted 6 bits to the right. Thus, WR0050 becomes HFC48 and C becomes 1.

ⓒWhen WR0050 is H1234, WX0000 is HF0C8 and SD is 0, the low-order 4 bits of WX0000 are 1000 (8 in decimal).
The contents are shifted 8 bits to the right. Thus, WR0050 becomes H0012 and C becomes 0.

ⓓWhen WR0050 is H1234, WX0000 is HFFFF and SD is 1, the low-order 4 bits of WX0000 are 1111 (15 in decimal).
The contents are shifted 15 bits to the right. Thus, WR0050 becomes HFFFE and C becomes 0.

(5) Shift Left (SHL (d, n))

Function:

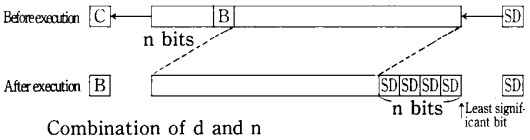
○ SHL (d, n)

①The contents of d (H0000 to HFFFF) are shifted n bits to the left.

②The contents of SD (0 or 1) are stored for n bits from the least significant bit.

③The contents of n bits (0 or 1) form the most significant bit are stored in C.

④When n (shift amount) is 0, no shift is performed. C retains the previous status.
(C is the special internal output R7F0, and SD is the special internal output R7F2.)

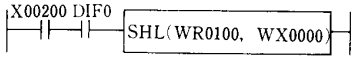


Combination of d and n

d \ n	Word (WX, WY, WR, WL, WM, TC)	Word (constant)
Word	The contents of low-order 4 bits of n (0 to 15) are used as the shift amount.	Values 0 to 15 are specifiable for n (shift amount).

Example:

Shift WR0100 to the left for the bits specified by WX0000.




```
LD    X00200
AND   DIF0
[
SHL (WR0100, WX0000)
]
```

ⓐWhen WR0100 is H1234, WX0000 is HFFF2 and SD is 1, the low-order 4 bits of WX0000 are 0010 (2 in decimal).
The contents are shifted 2 bits to the left. Thus, WR0100 becomes H48D3 and C becomes 0.

ⓑWhen WR0100 is H1234, WX0000 is H00F9 and SD is 0, the low-order 4 bits of WX0000 are 1001 (9 in decimal).
The contents are shifted 9 bits to the left. Thus, WR0100 becomes H48D3 and C becomes 0.

ⓒWhen WR0100 is H1234, WX0000 is H000B and SD is 1, the low-order 4 bits of WX0000 are 1011 (11 in decimal).
The contents are shifted 11 bits to the left. Thus, WR0100 becomes HA7FF and C becomes 1.

ⓓWhen WR0100 is H1234, WX0000 is H00FF and SD is 1, the low-order 4 bits of WX0000 are 1111 (15 in decimal).
The contents are shifted 15 bits to the left. Thus, WR0100 becomes H7FFF and C becomes 0.

(6) Rotate Right (ROR (d, n))

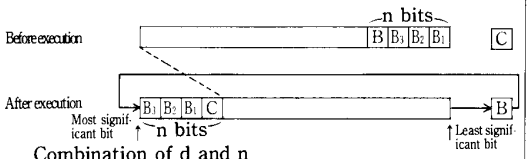
Function:

○ ROR (d, n)

①The contents of d (H0000 to HFFFF) are rotated n bits to the right.

②The contents of C (0 or 1) are stored in the most significant bit and the contents of the least significant bit (0 or 1) are stored in C.This processing is repeated n times.

③When n (shift amount) is 0, rotation is not performed. C retains the previous status.
(C is the special internal output R7F0.)

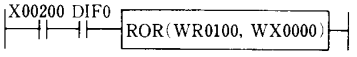


Combination of d and n

d \ n	Word (WX, WY, WR, WL, WM, TC)	Word (constant)
Word	The contents of low-order 4 bits of n (0 to 15) are used as the shift amount.	Values 0 to 15 are specifiable for n (shift amount).

Example:

Rotate WR0100 to the right for the bits specified by WX0000.




```
LD    X00200
AND   DIF0
[
ROR (WR0100, WX0000)
]
```

ⓐWhen WR0100 is H5678, WX0000 is HFFF1 and C is 1, the low-order 4 bits of WX0000 are 0001 (1 in decimal).
The contents are rotated 1 bit to the right. Thus, WR0100 becomes HAB3C and C becomes 0.

ⓑWhen WR0100 is H5678, WX0000 is H00F5 and C is 1, the low-order 4 bits of WX0000 are 0101 (5 in decimal).
The contents are rotated 5 bits to the right. Thus, WR0100 becomes H8AB3 and C becomes 1.

ⓒWhen WR0100 is H5678, WX0000 is H000A and C is 0, the low-order 4 bits of WX0000 are 1010(10 in decimal).
The contents are rotated 10 bits to the right. Thus, WR0100 becomes H3C15 and C becomes 1.

ⓓWhen WR0100 is H5678, WX0000 is H00FF and C is 1, the low-order 4 bits of WX0000 are 1111(15 in decimal).
The contents are rotated 15 bits to the right. Thus, WR0100 becomes H59E2 and C becomes 1.

5

(7) Rotate Left (ROL (d, n))

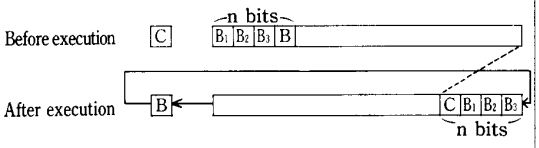
Function:

○ ROL (d, n)

①The contents of d are rotated n bits to the left.

②The contents of C are stored in the least significant bit and the contents of the most significant bit are stored in C.This processing is repeated n times.

③When n (shift amount) is 0, rotation is not performed.
C retains the previous status.
(C is the special internal output R7F0.)

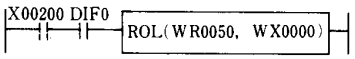


Combination of d and n

d \ n	Word (WX, WY, WR, WL, WM, TC)	Word (constant)
Word	The contents of low-order 4 bits of n (0 to 15) are used as the shift amount.	Values 0 to 15 are specifiable for n (shift amount).

Example:

Rotate WR0050 to the left for the bits specified by WX0000.




```
LD    X00200
AND   DIF0
[
ROL (WR0050, WX0000)
]
```

ⓐWhen WR0050 is H1234, WX0000 is H0003 and C is 1, the low-order 4 bits of WX0000 are 0011 (3 in decimal). The contents are rotated 3 bits to the left. Thus, WR0050 becomes H91A4 and C becomes 0.

ⓑWhen WR0050 is H1234, WX0000 is HFFF8 and C is 1, the low-order 4 bits of WX0000 are 1000 (8 in decimal).
The contents are rotated 8 bits to the left. Thus, WR0050 becomes H8123 and C becomes 0.

ⓒWhen WR0050 is H1234, WX0000 is H000D and C is 1, the low-order 4 bits of WX0000 are 1101(13 in decimal). The contents are rotated 13 bits to the left. Thus, WR0050 becomes H8123 and C becomes 0.

ⓓWhen WR0050 is H1234, WX0000 is H00FF and C is 0, the low-order 4 bits of WX0000 are 1111(15 in decimal). The contents are rotated 15 bits to the left. Thus, WR0050 becomes H048D and C becomes 0.

(8) Logical Shift Right (LSR (d. n))

Function:

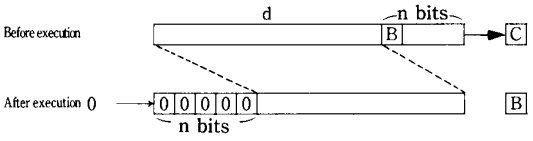
○ LSR (d, n)

①The contents of d are shifted n bits to the right.

②The n bits form the most significant bit are all set to zero.

③C contains the contents of the nth bit (0) from the least significant bit.

④When a (shift amount) is 0, shift is not performed.
C retains the previous status.
(C is the special internal output R7F0.)

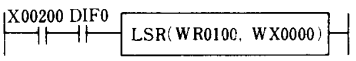


Combination of d and n

d \ n	Word (WX, WY, WR, WL, WM, TC)	Word (constant)
Word	The contents of low-order 4 bits of n (0 to 15) are used as the shift amount.	Values 0 to 15 are specifiable for n (shift amount).

Example:

Shift WR0100 to the right for the bits specified by WX0000.




```
LD    X00200
AND DIF0
[
LSR (WR0100, WX0000)
]
```

ⓐWhen WR0100 is H5678 and WX0000 is HFFF2, the low-order 4 bits of WX0000 are 0010 (2 in decimal). The contents are shifted 2 bits to the right. Thus, WR0100 becomes H159E and C becomes 0.

ⓑWhen WR0100 is H5678 and WX0000 is H0006, the low-order 4 bits of WX0000 are 0110 (6 in decimal). The contents are shifted 6 bits to the right. Thus, WR0100 becomes H0159 and C becomes 1.

ⓒWhen WR0100 is H5678 and WX0000 is H000B, the low-order 4 bits of WX0000 are 1011 (11 in decimal). The contents are shifted 11 bits to the right. Thus, WR0100 becomes H000A and C becomes 1.

ⓓWhen WR0100 is H5678 and WX0000 is H00FF, the low-order 4 bits of WX0000 are 1111 (15 in decimal). The contents are shifted 15 bits to the right. Thus, WR0100 becomes H0000 and C becomes 1.

(9) Logical shift Left (LSL (d, n))

Function:

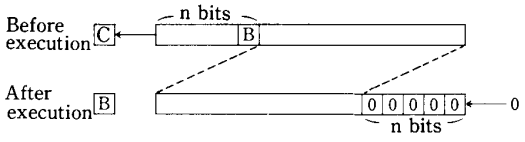
○ LSL (d, n)

①The contents of d are shifted n bits to the left.

②The n bits from the least significant bit are all set to zero.

③C contains the contents of the nth bit from the most significant bit.

④When n (shift amount) is 0, shift is not performed. C retains the previous status.
(C is the special internal output R7F0.)

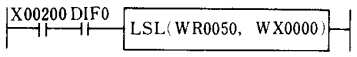


Combination of d and n

d \ n	Word (WX, WY, WR, WL, WM, TC)	Word (constant)
Word	The contents of low-order 4 bits of n (0 to 15) are used as the shift amout.	Values 0 to 15 are specifiable for n (shift amount).

Example:

Shift WR0050 to the left for the bits specified by WX0000.




```
LD    X00200
AND   DIF0
[
LSL (WR0050, WX0000)
]
```

ⓐWhen WR0050 is H1234 and WX0000 is HFFF3, the low-order 4 bits of WX0000 are 0011 (3 in decimal). The contents are shifted 3 bits to the left. Thus, WR0050 becomes H91A0 and C becomes 0.

ⓑWhen WR0050 is H1234 and WX0000 is H0007, the low-order 4 bits of WX0000 are 0111 (7 in decimal). The contents are shifted 7 bits to the left. Thus, WR0050 becomes H1A00 and C becomes 1.

ⓒWhen WR0050 is H1234 and WX0000 is H000C, the low-order 4 bits of WX0000 are 1100 (12 in decimal). The contents are shifted 12 bits to the left. Thus, WR0050 becomes H4000 and C becomes 1.

ⓓWhen WR0050 is H1234 and WX0000 is HFFFF, the low-order 4 bits of WX0000 are 1111 (15 in decimal). The contents are shifted 15 bits to the left. Thus, WR0050 becomes H0000 and C becomes 0.

(10) BCD Shift Right (BSR (d, n))

Function:

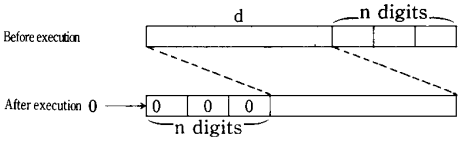
○ BSR (d, n)

①The contents of d are shifted n digits to the right. (One digit consists of four bits.)

②The n digits from the most significant bit are all set to zero.

③The n digits from the least significant bit are truncated.

④When n (shift amount) is 0, shift is not performed.



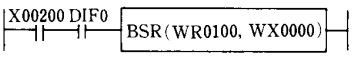
One digit consists of 4 bits.

Combinatuion of d and n

d \ n	Word (WX, WY WR, WL, WM, TC)	Word (constant)
Word	Contents of low-order 2 bits of n (0 to 3) are used as shift amount.	Values 0 to 3 are specifiable for n (shift amount).

Example:

Shift WR0100 to the right for the number of digits specified by WX0000.




```
LD    X00200
AND   DIF0
[
BSR (WR0100, WX0000)
]
```

ⓐWhen WR0100 is H5678 and WX0000 is H0000, the low-order 2 bits of WX0000 are 00 (0 in decimal). Because shift amount is 0, shift does not occur.

ⓑWhen WR0100 is H5678 and WX0000 is HFFF1, the low-order 2 bits of WX0000 are 01 (1 in decimal). The contents are shifted 1 digit to the right. Thus, WR0100 becomes H0567.

ⓒWhen WR0100 is H5678 and WX0000 is H0002, the low-order 2 bits of WX0000 are 10 (2 in decimal). The contents are shifted 2 digits to the right. Thus, WR0100 becomes H0056.

ⓓWhen WR0100 is H5678 and WX0000 is HFFFF, the low-order 2 bits of WX0000 are 11 (3 in decimal). The contents are shifted 3 digits to the right. Thus, WR0100 becomes H0005.

5

(11) BCD Shift Left (BSL (d, n))

Function:

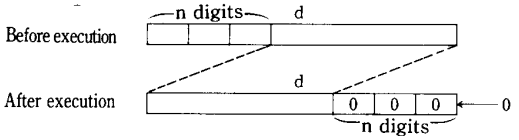
○ BSL (d, n)

①The contents of d are shifted n digits to the left. (One digit consists of four bits.)

②The n digits from the least significant bit are all set to zero.

③The n digits from the most significant bit are truncated.

④When n (shift amount) is 0, shift is not performed.



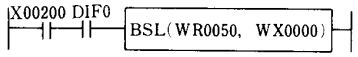
One digit consists of 4 bits.

Combination of d and n

d \ n	Word (WX, WY WR, WL, WM, TC)	Word (constant)
Word	Contents of low-order 2 bits of n (0 to 3) are used as shift amount.	Values 0 to 3 are specifiable for n (shift amount).

Example:

Shift WR0050 to the left for the number of digits specified by WX0000.




```
LD      X00200
AND     DIF0
[
BSL (WR0050, WX0000)
]
```

ⓐWhen WR0050 is H1234 and WX0000 is HFFFC, the low-order 2 bits of WX0000 are 00 (0 in decimal). Because shift amount is 0, shift does not occur.

ⓑWhen WR050 is H1234 and WX0000 is H0001, the low-order 2 bits of WX0000 are 01 (1 in decimal). The contents are shifted 1 digit to the left. Thus, WR0050 becomes H2340.

ⓒWhen WR0050 is H1234 and WX0000 is H0002, the low-order 2 bits of WX0000 are 10 (2 in decimal). The contents are shifted 2 digits to the left. Thus, WR0050 becomes H3400.

ⓓWhen WR0050 is H1234 and WX0000 is HFFFF, the low-order 2 bits of WX0000 are 11 (3 in decimal). The contents are shifted 3 digits to the left. Thus, WR0050 becomes H4000.

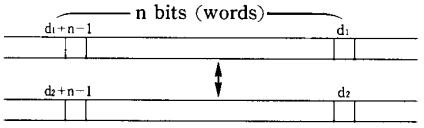
(12) Exchange (XCG (d1, d2, n))

Functeon:

○ XCG (d1, d2, n)

①The contents of n bits (words) from dl to dl + n - 1 and the contents of from d2 to d2 + n - 1 are exchanged. When n (block width) is 0, dats is not exchanged. DER is set to 0.

②The value dl + n - 1 or d2 + n - 1 must not exceed the last bit (word) (R7FF, L0007F, L0007F, MFFF, WRF1FF, WL0007, WL1007 or WMFF).
If exceeded, DER is set to 1 and the smaller block width between the last bit (word) - dl + 1 and the last bit (word) - d2 + 1 is used to exchange data blocks.
If not exceeded, DER is set to 0 and data blocks are exchanged as specified.



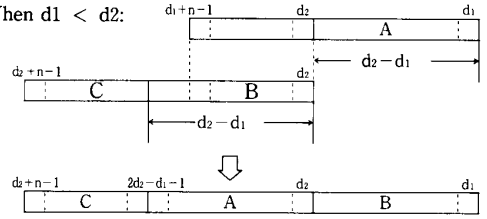
Combination of dl, d2, and n

d1,d2 \ n	Word (WX, WY, WR, WL, WM, TC)	Word (constant)
Both bits or words	The contents of the low-order 8 bits of n (0 to 255) are used as block width.	Values 0 to 255 are specifiable for n (block width).

Array variables are not specifiable for d, s, n.

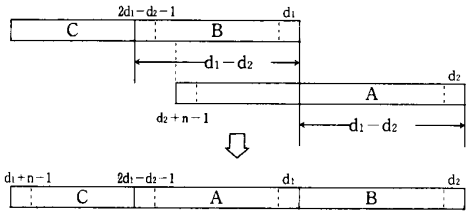
③When overlapping occurs between the block from dl to dl + n - 1 and the block from d2 to d2 + n - 1, DER is set to 1 and these blocks are exchanged as shown below.

ⓐWhen dl < d2:



○ Block from dl to d2 - 1 and that from d2 to 2d2 - dl to 1 are exchanged.

ⓑWhen dl > d2:



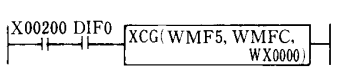
○ Block from d2 to dl - 1 ans that from dl to 2dl - d2 - 1 are exchanged.

ⓒWhen dl = d2:

○ Block exchange is not carried out because the block from dl to dl + n - 1 is the same as the blok from d2 to d2 + n - 1.

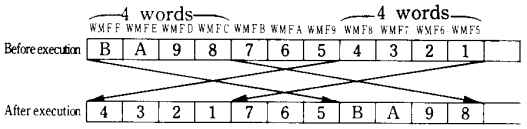
Example:

①Exchange the WX0000-specified words from WMF5 and those from WMFC.



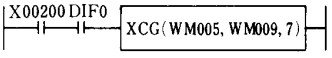

```
LD    X00200
AND   DIF0
[
XCG(WMF5, WMFC
    WX0000)
]
```

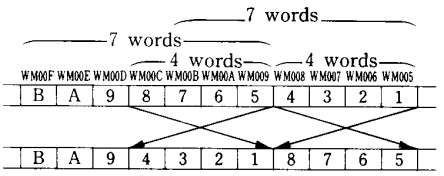
ⓐWhen WX0000 is HFF08:



The low-order 8 bits of WX0000 are 00001000 (8 in decimal). So block width is 8 words. However, because there are only 4 words from WMFC to WMFF, DER is set to 1.

②Exchage of 7 words from WM005 ans those from WM009





The block exchenge width is 7 words. However, because words from WM009 to WM00B overlap betweem the two blocks, exchange is made between two 4-word blocks in actuality.

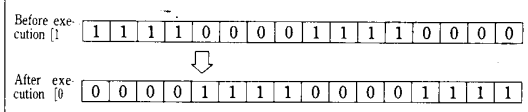
(13) Not (NOT (d))

Function:

○ NOT ⓓ

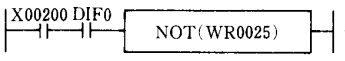
①Bit setting of d is inverted.

$d \leftarrow \bar{d}$



Example:

Invert the contents of WR0025.




```
LD    X00200
AND   DIF0
[
NOT  (WR0025)
]
```

ⓐWhen WR0025 is H1234, WR0025 has the value HEDCB.

ⓑWhen WR0025 is HFFFF, WR0025 has the value H0000.

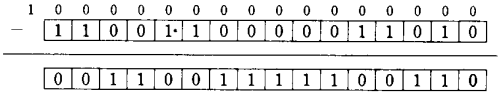
(14) Negate (NEG ⓓ)

Function:

○ NEG ⓓ

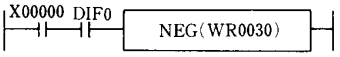
①Two's complement of d (words) is computed. H10000 −(contents of d) is computed and the result is output to d.

$d \leftarrow -d$



Example:

Compute two's complement of WR0030.




```
LD    X00000
AND   DIF0
[
NEG  (WR0030)
]
```

ⓐWhen WR0030 is H5678, WR0030 has the value HA988.

ⓑWhen WR0030 is HFFFF, WR0030 has the value H0001.

5

(15) Absolute (ABS (d, s))

Function:

○ ABS (d, s)

①When s is positive or zero, the contents of s are stored in d. Value 0 is stored in C (carry).

②When s is negative, the two's complement of s is stored in d. Value 1 is stored in C. See Item 14.

③When s is a word, a decimal number between 0 and 32767 corresponds to a hexadecimal number between H0000 and H7FFF. A decimal number between -32768 and -1 corresponds to a hexadecimal number between H8000 and HFFFF.
C is the special internal output R7F0.

d ← | s |

c ← Sign of s

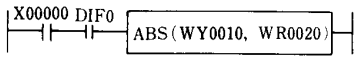
{ Positive or zero: 0
{ Negative: 1

Combination of d and s

d	s
Word	Word

Example

Store the absolute value of WR0020 in WY0010.




```
LD    X00000
AND   DIF0
[
ABS (WY0010, WR0020)
]
```

ⓐWhen WR0020 in H8000, H8000 is set in WY0010 and 1 is set in C.

ⓑWhen WR0020 is H0000, H0000 is set in WY0010, and 0 is set in C.

ⓒWhen WR0020 is H7FFF, H7FFF is set in WY0010, snd 0 is set in C.

(16) Binary-BCD Conversion (BCD (d, s))

Function:

○ BCD (d, s)

①The contents of s are converted from binary to BCD, and the result is stored in d.

② When s is a word:

○ H0000 ≦ S ≦ H270F

DER ← 0 Terminated normally

○ H270F < S

DER ← 1 Binary data error

③The contents of d are left intact when a binary data error occurs.
DER is the special internal output R7F4.

s

Before convresion [BINARY]

⇩ d

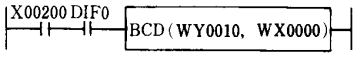
After converssion [BCD]

Combination of d and s

d	s
Word	Word

Example:

Convert the binary data input from WX0000 to BCD data and store the result in WY0010.




```
LD    X00200
AND   DIF0
[
BCD (WY0010, WX0000)
]
```

ⓐWhen WX0000 in WY1759.
H1759 is 5977 in decimal. So H5977 (BCD) is stored in WY0010.

ⓑWhen WX0000 is H2707:
H2707 is 9991 in decimal. So H9991 (BCD) is stored in WY0010.

(17) BCD-binary Conversion (BIN (d, s))

Function:

○ BIN (d, s)

①The contents of s are converted from BCD to binary and the result is stored in d.

②When a digit in s contains a hexadecimal number (one digit consisting of four bits) between A and F:
DER ← 1 BCD data error

③When all digits of s contain hexadecimal number 0 - 9:
DER ← 0 Normal

④The contents of d are left untact when a BCD data error occurs.

s

Before conversion [BCD]

⇩ d

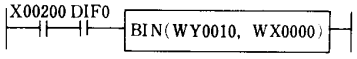
After conversion [BINARY]

Combination of d and s

d	s
Word	Word

Example:

Convert BCD data input from WX0000 to binary and store the result in WY0010.




```
LD    X00200
AND   DIF0
[
BIN (WY0010, WX0000)
]
```

ⓐWhen WX0000 is H1000:

H03E8 is stored in WY0010.

DER is 0.

ⓑWhen WX0000 is H9999:

H270F is stored in WY0010.

DER is 0.

(18) Decode (DECO (d, s, n)

Function:

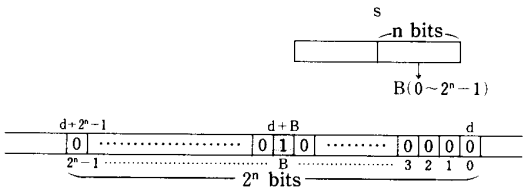
○ DECO (d, s, n)

①Consider B as the contents of the low-order n bits of s.

②Bit d + B is set to 1.

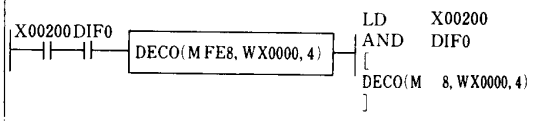
③The bits from d to d + B - 1 and the bits from d + B + 1 to d + 2^n - 1 (total of 2^n - 1 bit) are set to 0.

④d + 2^n - 1 must not extend beyond the following addresses; R7FF, L0007F, L0007F and MFFF. If it extends beyond them, DER is set to 1 and the data is decoded within the bit address limits.
DER is set to 0 when it does not exceed this limit.

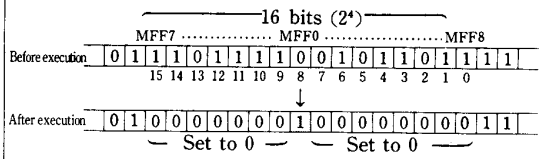


Example:

①Decode the contents of WX0000 within 16-bit width from MFE8.



ⓐWhen WX0000 is H0008 (8 in decimal):



DER is set to 0.

5

(19) Encode (ENCO (d, s, n))

Function:

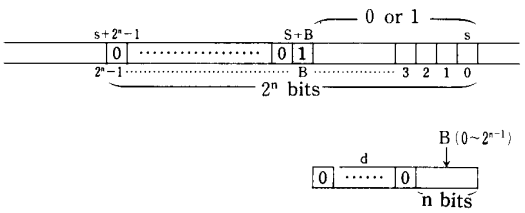
○ ENCO (d, s, n)

①The bit position B where 1 is set withtin 2^n bits from s to s + 2^n - 1 is set in d.
The high-order bits (16 - n) of d are set to 0.

②If all bits from s to s + 2^n - 1 are 0, 0 is stored in d and C is set to L.
Otherwise, C is set to 0.

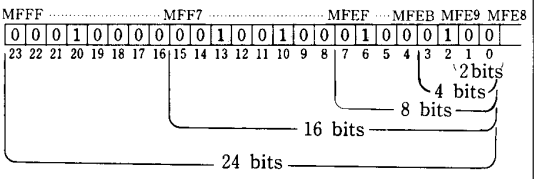
③If two or more bits are 1 within a range from s to s + 2^n - 1, the highest bit position is encoded.

④s + 2^n - 1 must not extend beyond the following bit addresses: R7FF, L0007F, L0007F and MFFF.
If it extends beyond them, DER is set to 1 and the bits from s to the last bit are encoded.
DER is set to 0 and encoding is carried out normally unless this limit is exceeded.



Example:

Store the value encoded from MFE8 in WY0010.



ⓐ X00200 DIF0 — ENCO(WY0010, MFE8, 1)

```
LD    X00200
AND   DIF0
[
ENCO(WY0010,MFE8,1
```

No bit between MFE8 and MFE9 has the value 1. So, 0 is stored in WY0010. C is set to 1 and DER is set to 0.

ⓑ X00200 DIF0 — ENCO(WY0010, MFE8, 2)

H0002 is stored in WY0010.
C is set to 0 and DER is set to 0.

ⓒ X00200 DIF0 — ENCO(WY0010, MFE8, 3)

H0006 is stored in WY0010.
C is set to 0 and DER is set to 0.

ⓓ X00200 DIF0 — ENCO(WY0010, MFE8, 4)

H000D is stored in WY0010.
C is set to 0 and DER is set to 0.

X00200 DIF0 — ENCO(WY0010, MFE8, 5)

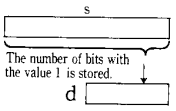
H0014 is stored in WY0010.
C is set to 0 and DER is set to 1.

(20) Bit Count (BCU (d, s))

Function:

○ BCU (d, s)

①The number of bits which are 1 in s is stored in d.



Combination of d and s

d	s
Word	Word

Example:

Count the number of bits with the value 1 in WX0000 and store the result in WM020.

X00200 DIF0 — BCU(WM020, WX0000)

```
LD    X00200
AND   DIF0
[
BCU(WM020, WX0000)
]
```

ⓐWhen WX0000 in HF557:

WX0000 1 1 1 1 0 1 0 1 0 1 0 1 0 1 1 1

H000B is stored in WM020.

(21) Swap (SWAP (d))

Function:

○ SWAP (d)

①The low-order 8 bits and the high-order 8 bits of d are swapped.

d [Low-order 8 bits][High-order 8 bits]

Example:

Swap the bits in WY0010.

X00200 DIF0 — SWAP(WY0010)

```
LD    X00200
AND   DIF0
[
SWAP(WY0010)
]
```

ⓐWhen WY0010 is H1234:

WY0010 becomes H3412.

(22) Unite (UNIT (d, s, n))

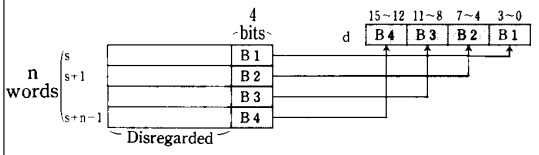
Function:

○ UNIT (d, s, n)

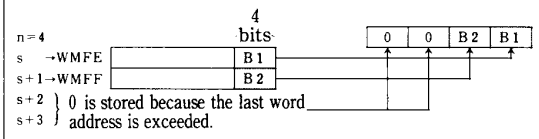
①The sets of low-order 4 bits (B1, B2, B3 and B4) of n words (0 to 4 words) from s are stored in each 4-bit unit of d, starting with the least significant 4 bits of d.

②If n is 0 to 3, the high-order 4 x (4 - n) bits of d are filled with zero.

③Data in s through s + n - 1 remains unaffected by the UNIT instruction.



④If s + n - 1 exceeds the last word address (WRF1FF, WL0007, WL1007 or WMFF), DER is set to 1. And computation is carried out as follows.

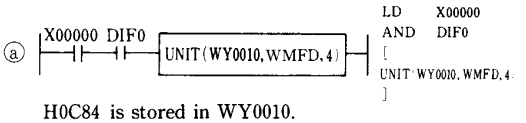


(DER is the special internal output R7F4.)

Example:

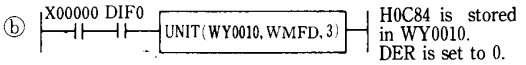
Unite WMFD through WMFF and store the values in WY0010.

WMFD	H1234
WMFE	H5678
WMFF	H9ABC

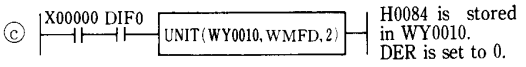


H0C84 is stored in WY0010.

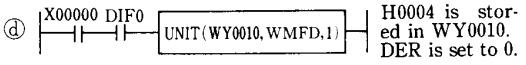
DER is set to 1.



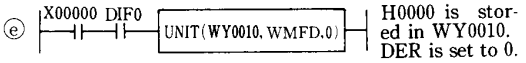
H0C84 is stored in WY0010.
DER is set to 0.



H0084 is stored in WY0010.
DER is set to 0.



H0004 is stored in WY0010.
DER is set to 0.



H0000 is stored in WY0010.
DER is set to 0.

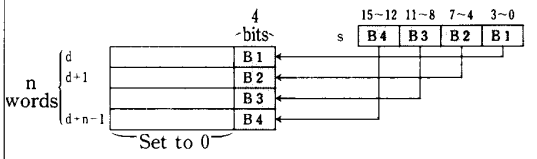
(23) Distribute (DIST (d, s, n))

Function:

○ DIST (d, s, n)

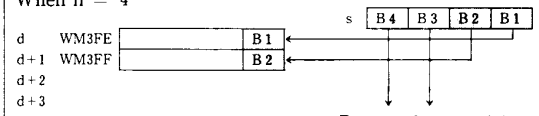
①Each block of 4-bit data (B1, B2, B3 and B4) in s is copied into the low-order 4 bits of words from d to d + n - 1.

②The high-order 12 bits of words from d to d + n - 1 are all set to zero.



③If d + n - 1 exceeds the last word address (WRF1FF, WL0007, WL1007 or WMFF), DER is set to 1. And operation is performed as follows.

When n = 4

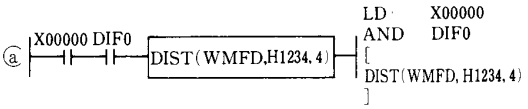


Because d + 2 and d + 3 exceed the last word addresses, B4 and B3 are not copied.

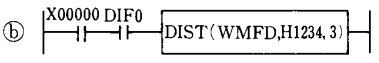
④When n is 0, no operation is performed.
(DER is the special internal output R7F4.)

Example:

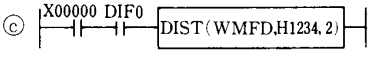
Distribute H1234 to each word starting from WMFD.



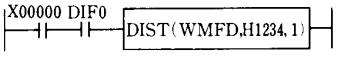
H0004 is copied to WMFD, H0003 is copied to WMFE and H0002 is copied to WMFF. DER is set to 1.



H0004 is copied to WMFD, H0003 is copied to WMFE and H0002 is copied to WMFF.
DER is set to 0.



H0004 is copied to WMFD and H0003 is copied to WMFE. DER is set to 0.



H0004 is copied to WMFD.
DER is set to 0.

5

5.6 Control Instructions

(1) Ending Normal Scan (END)

Function:

○ END

①Designates end of normal scan.

②Causes return to the beginning of a program for execution of normal scan.

③Normal scan covers a range from the beginning of a program to the END instruction.

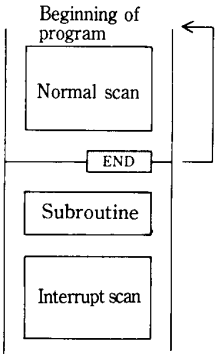
④In case an error is found in the END instruction processing before operation, any of the error codes below is set in the word special internal output WRF001.

H10 END undefined

H22 END area error

H32 END start condition error

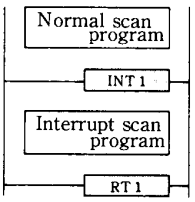
(Error code)



Cautions:

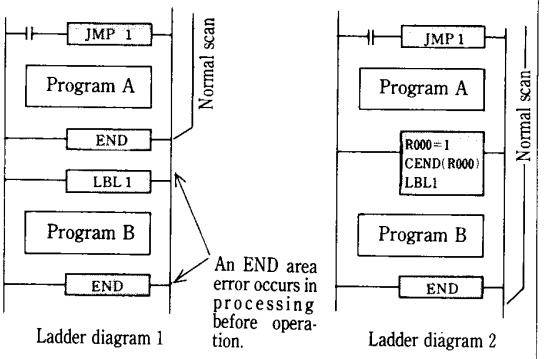
①Where a subroutine or interrupt scan is not used in the program, this instruction is optional.
If the instruction is not placed, the normal scan covers the whole range from the beginning to end of the program.

②Where subroutine or interrupt scan is used in the program, be sure to place this instruction at the end of the normal scan area.



END undefined error is caused in processing before operation.

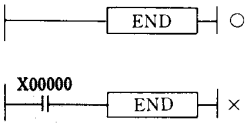
③Use this instruction only once in the program.



(1)In this case, normal scan is terminated at the first END instruction so that an END area error is caused in processing before operation.

(2)Correct Ladder 1 as shown in Ladder 2.

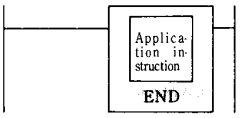
④Unconditional start is required when this instruction is used.



In this case, END start condition error is caused.

(1)In case X00000 is OFF, END instruction is not executed and instead a next instruction is executed. Therefore, correct operation is not made.

(2)In case of unconditional start, the following can also be programmed.

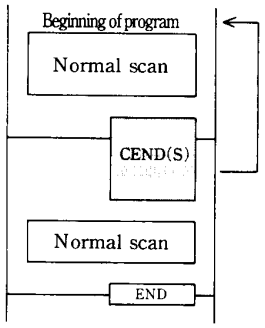


(2) Conditional End of Scan (CEND (s))

Function:

○ CEND (s)

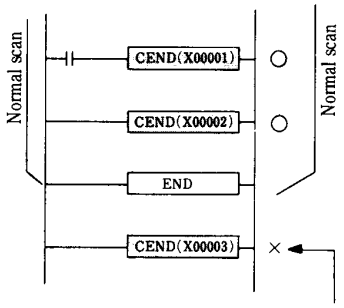
①To cause return to the beginning of a program for execution of normal scan if s is ON.
To cause execution of an instruction following instruction if s is OFF.



②In case an error is found in the CEND instruction in processing before operation, the error code below is set in the word special internal output WRF001.

H23 END area error

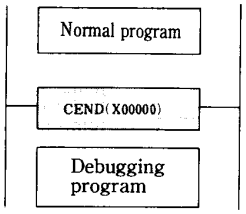
③This instruction can be used only in normal scan area. It is usable as many times as desired.



(1)In the above case, a CEND area error occurs in processing before operation since a CEND instruction is placed outside the normal scan area.

Example:

Place a debugging program after the normal program and run the debugging program for debugging if system is abnormal.



(3) Unconditional Jump (JMP n)

Function:

○ JUP n

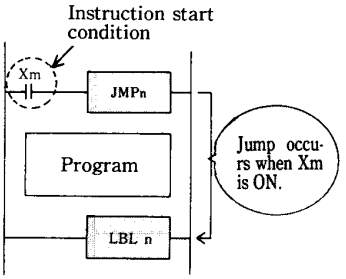
①To cause a jump from this instruction to LBL n of the same code number when the condition of this instruction is satisfied (ON).

②To cause a program after this instruction to be executed when the condition is not satisfied.

③At execution of this instruction, the error codes below may occur.

	(Error code)
(1) LBL undefined error	H0015
(2) Off-area jump error	H0040

In case an error occurs, "1" is set at ERR, an error code is set at WRF015 and the program following this instruction is executed.
If no error occurs, ERR and WRF015 hold the previous state.



● n stands for a code number from 0 to 255.

(ERR is the special internal output R7F3.)

5

(4) Conditional Jump (CJMP n (s))

Function:

○ CJMP n (s)

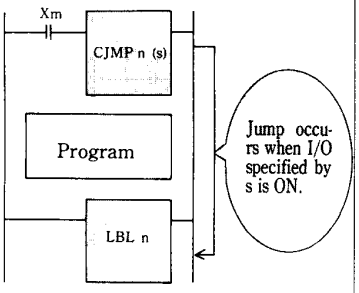
①To cause a jump from this instruction to LBL n of the same code number when bit I/O specified by s is ON.

②To cause program after this instruction to be executed when the bit I/O is OFF.

③On execution of this instruction, the errors below may occur.

	(Error code)
(1) LBL undefined error	H0015
(2) Off-area jump error	H0040

In case either error occurs, "1" is set at ERR and the relevant error code is set in WRF015, then the program after this instruction is executed.
If no error occurs, ERR and WRF015 hold the previous state.

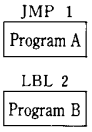


● "n" stands for a code number within 0 to 255.

(ERR is the special internal output R7F3.)

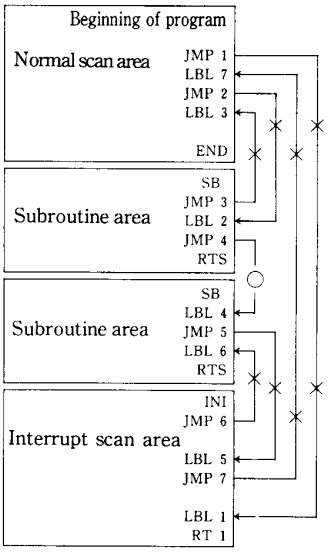
* Syntax of JMP and CJMP Instructions

①A JMP instruction requires an LBL n of the same code number n.



● Execution of JMP1 without LBL1 causes an LBL undefined error.
JMP1 does not have any effect and processing is made according to the next program A.

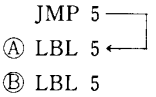
②Jump is impossible beyond the area in which the JMP instruction is placed.



● If a JMP1 instruction is executed in this case, an "off-area jump error" is caused since LBL1 is not within normal scan area.
JMP1 instruction will bring about no effect and the next instruction will be executed.

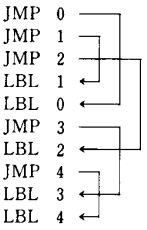
● Same processing made for JMP2 to JMP7.

③LBL of the same code number n as the one of JMP instruction must not be used doubly.

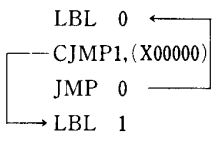


● In processing before operation, an LBL double definition error is caused since both LABEL instructions (A) and (B) have the same code number 5.

④ JMP instructions can be nested.



⑤A JMP instruction can cause a backward jump.

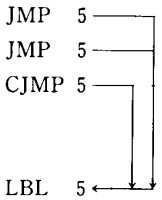


● JPM0 causes backward jump to LBL0.

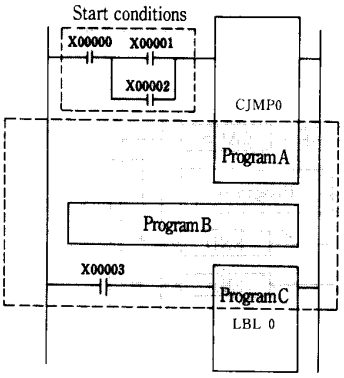
● When input X00000 is ON, jump out of loop from LBL0 to JMP0 occurs by jumping from CJMP1, (X00000) to LBL1.

● In the absence of an instruction such as CJMP1, (X00000) that causes jump out of loop, the loop between LBL0 and JMP0 is repeated infinitely.

⑥JMP instructions of the same code number may be used at multiple locations.



⑦Start conditions can be programmed for a JMP instruction.

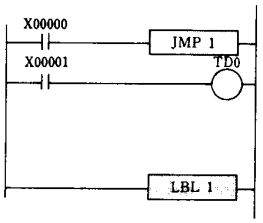


● If a jump is made from CJMP0 to LBL0, programs A, B and C are not executed.

⑧Syntax ① to ⑦ must also be followed for CJMP instruction.

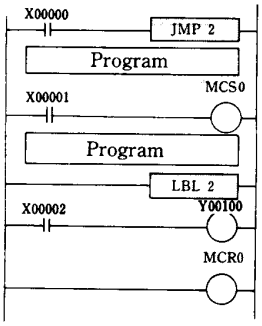
* Cautions on JMP and CJMP Instructions

①When a jump is made from a JMP instruction to an LBL, the state of each I/O between JMP and LBL is held as it is, but the current value of timer is updated.



● If X00000 gose ON and jump from JMP1 to LBL1 occurs after X00001 goes ON, the current value of TD0 is updated. But if X00000 is held ON, TD0 will not go ON even if its current value exceeds 100.

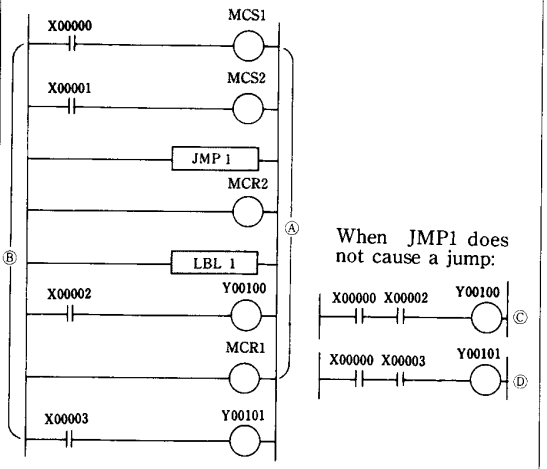
②The following operation takes place if JMP, MCS and MCR instructions are used in combination. So use care when programming.



● When JMP2 does not cause a jump:
When both X00001 and X00002 are ON, Y00100 is turned ON.

● When JMP2 causes a jump:
So long as X00000 is ON, Y00100 is turned ON of OFF according to ON/OFF status of X00002, regardless of whether X00001 is ON or OFF.

③ Jump out from between MCS and MCR



● Circuit (A) is equivalent to circuit (C) regardless of wheter JMP1 causes or does not cause a jump.

● Circuit (B) is equivalent to circuit (D) when JMP1 causes a jump.

(1)When X00000 or X00001 is OFF:

● JMP1 does not cause a jump so that processing is the same as when JMP1 instruction is absent.

(2)When both X00000 and X00001 are ON:

● JMP1 causes a jump, but because both X00000 and X00001 are ON, circuits (A) and (C) are the same. Also circuits (B) and (D) are the same. As a result, processing is the same as when JMP1 instruction is absent.

5

(5) Label (LBL n)

Function:

○ LBL n

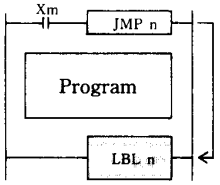
①To cause a jump to LBL n of the same code number n when the condition (Xn) for the JMP n instruction is satisfied (ON).

②LBL n designates the jump destination of a JMP or CJMP instruction and LBL n itself causes no processing.

③In case an error is found in the LBL instruction at processing before operation, the error code below is set in the word special internal output WRF001.

H0001 LBL double definition

(Error code)



● "n" stands for a code number within 0 to 255.

(6) Subroutine Call (CAL n)

Function:

○ CAL n

①When start condition Xm is turned ON, the subroutine with the same code number n as this instruction is called.

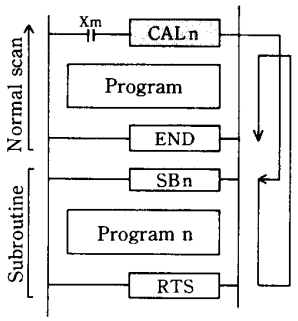
②On execution of this instruction, the errors below might occur.

(Error code)

(1) SB undefined error H0013

(2) Nesting error H0041

In case either error occurs, "1" is set at ERR, the relevant error code is set in WRF015 and the program after this instruction is executed.
If no error occurs, ERR and WRF015 hold the previous state.



● For n, designate a code number within 0 to 99.

● ERR stands for the special internal output R7F3.

● A subroutine cannot call another subroutine. That is, nesting is possible at only one level.

(7) Subroutine Start (SB n)

Function:

○ SB n

①This instruction declares the start of a subroutine at its beginning.

②The code number of this instruction becomes the subroutine name.

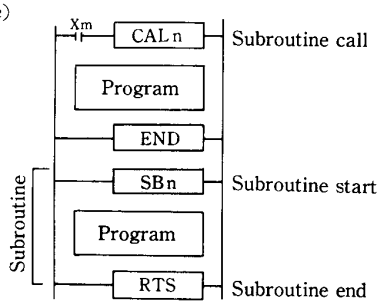
③Subroutine can be called by designation the same code number n as of SB n to corresponding CAL n.

④When the subroutine is called, the subroutine start instruction itself does not have any effect and execution starts from the next instruction.

⑤In case an error is found in the SB instruction at processing before operation, the error code below is set in the word special internal output WRF001.

H0004 SB double definition

(Error code)



● For n, designate a code number within 0 to 99.

(8) Subroutine End (RTS)

Function:

○ RTS

①After execution of this instruction, control is transferred to the processing following the instruction which called the subroutine.

②This instruction must be programmed at the end of the subroutine.

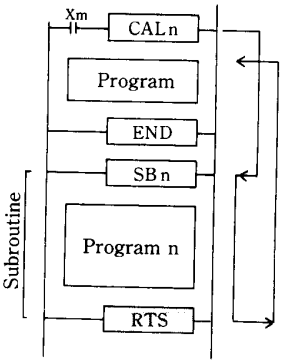
③If an error is found in the RTS instruction at processing before operation, any of the error codes below is set in the word special internal output WRF001.

H0011 RTS undefined error

H0020 RTS area error

H0030 RTS start condition error

(Error code)

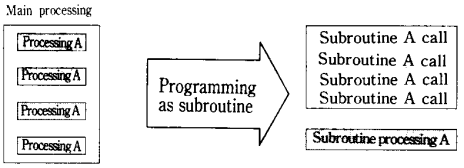


5

* Subroutine

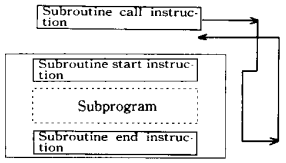
① How to use subroutine

A single subroutine can be used for processing which occurs may times in a given program.
Then, you can call the subroutine program whenever you need it.



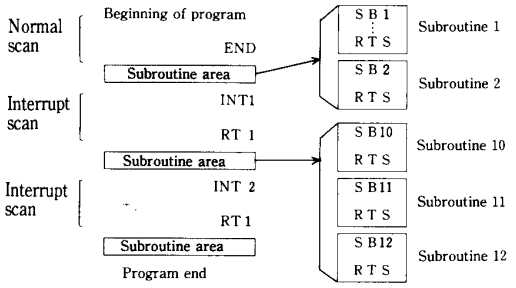
② Composition of subroutine

A subroutine consists of an instruction to call it (subroutine call instruction) and the subroutine itself.
The subroutine itself is composed of the subroutine start instruction, target subprogram and subroutine end instruction.

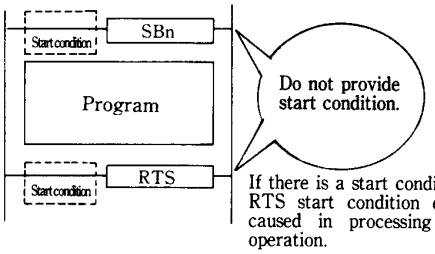


* Syntax of Subroutine

①A subroutine can be placed between a normal scan and an interrupt scan, between interrupt scans or after a final interrupt scan.

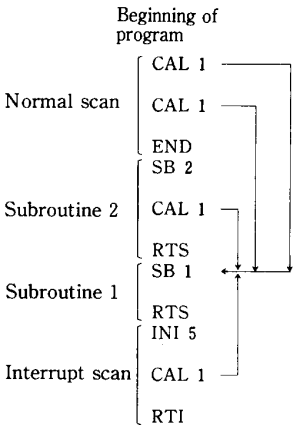


②Program subroutine start (SB n) instruction and subroutine end (RTS) instruction without start conditions.

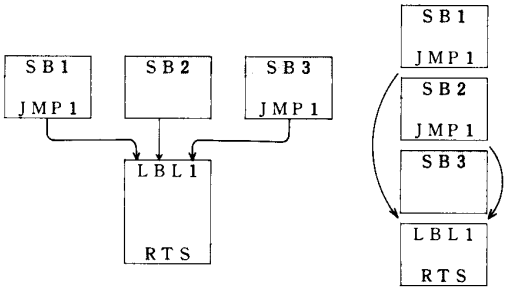


If there is a start condition, an RTS start condition error is caused in processing before operation.

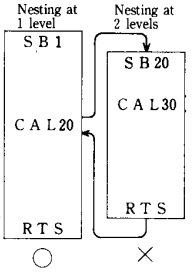
③The same subroutine can be called from a normal scan, interrupt scan or subroutine.



④A subroutine with two or more entry points and with one exit point can also be programmed.



⑤Subroutine can be nested at only one level.



(9) Starting Interrupt Scan (INT n)

Function:

○ INTn

①This instruction declares the start of an interrupt scan at its beginning point.

②The code number of this instruction becomes the interrupt scan number.

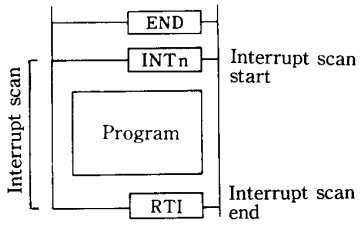
③On execution of interrupt scan, the INT n instruction itself does not bring about any effect and execution starts from the next instruction.

④In case an error is found in the INT instruction at processing before operation, either error code below is set in the word special internal output WRF001.

(Error code)

H0005 INT double defined error

H0014 INT undefined error



- For n, designate a code number within 0 to 2.
- Start condition for interrupt scan is specifiable by code number n.

n	Name	Start condition
0	10 msec periodic program	Start is automatically made every 10 msec for execution.
1	20 msec periodic program	Start is automatically made every 20 msec for execution.
2	40 msec periodic program	Start is automatically made every 40 msec for execution.

5

(10) Ending Interrupt Scan (RTI)

Function:

○ RTI

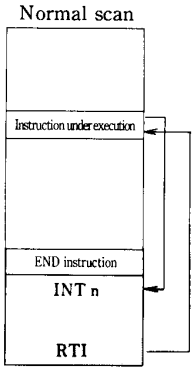
①Interrupt scan n is executed upon interruption while an instruction is executed. Execution of an RTI instruction at the end of an interrupt scan causes control to be transferred to the processing following the interrupted instruction.

②Be sure to place this instruction at the end of an interrupt scan.

③In case an error is found in the INT instruction in the processing before operation, any of the error codes below is set in the word special internal output WRF001.

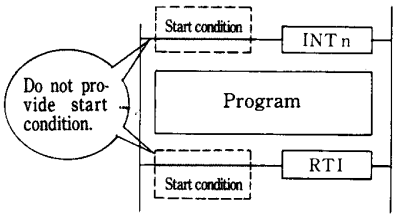
(Error code)

H0012	RTI undefined error
H0021	RTI area error
H0031	RTI start condition error

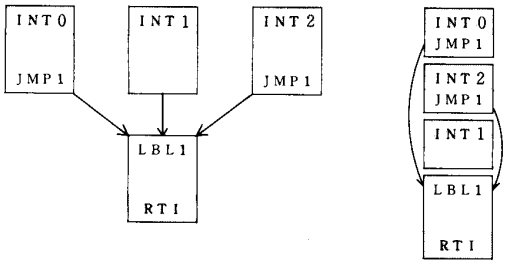


* Syntax of Interrupt Scan

①Program interrupt scan start (INT n) instruction and interrupt scan end (RTI) instruction without start condition.



②An interrupt scan with two or more entry points and one exit point can also be programmed.



5.7 Calendar Clock

(1) How to read

The current year, month, day, day of week, hour, minute and second can directly be read out via the special internal outputs (WRF00B to WEF00F). Time of the internal clock has been factory-set. So the clock is usable immediately after turning on power supply.

WRF00B Year [1990] ····· ▷ Year can always be read out in 4-digit BCD code.

WRF00C Month and day [1123] ····· ▷ Month and day can always be read out in BCD code.

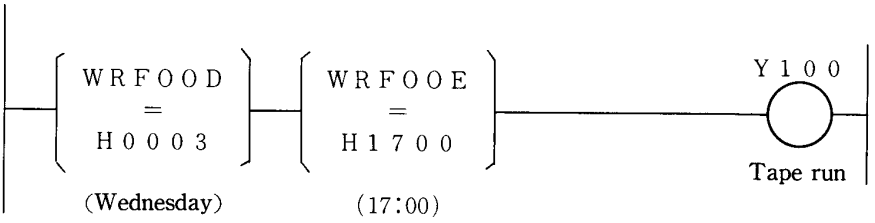
WRF00D Day of week [0005] ····· ▷ Day of week can always be read out in BCD code.
(0: Sunday, 1: Monday, ····· 6: Saturday)

WRF00E Hour and minute [1345] ····· ▷ Hour (24-hour base) and minute can always be read out in BCD code.

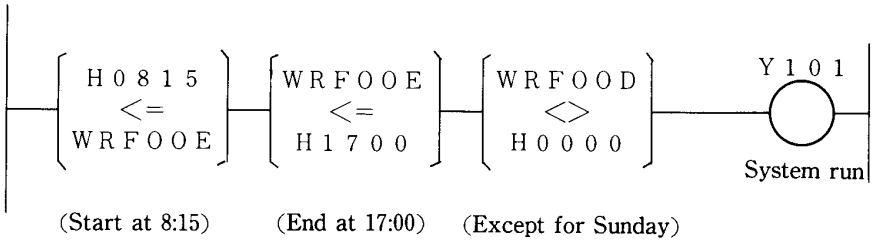
WRF00F Second [0059] ····· ▷ Second can always be read out in BCD code.

Exemplified below is real-time control in response to day of week or time point by using relational boxes.

(Example 1) Announcement of "regular work end time report" and broadcasting of music for 1 minute at 17:00 on Wednesday



(Exzmple 2) Running of a certain system from 8:15 to 17: 00 except for Sunday



Each time value in the above special internal outputs (WRF00B to ERF00F) is to be updated between scans.

That is, time value does not change during a scan.

Therefore, We need not mind a carry from second to minute, etc.

Besides WRF00B to WRF00F from Which current time is always readable, the special internal outputs (WRF01B to WRF01F) have been prepared for reading out the exact time point of the leading edge (⎍↑⎍) of the special internal output R7F8 which requests readout.

The special internal outputs (WRF01B to WRF01F) hold the current readout time point unless the next leading edge of R7F8 is detected.

(2) Clock adjustment

Although the calendar clock has been factory-adjusted, its accuracy will be thermally affected as shown below.

{ +30 sec to -60 sec/month (at 0 to 35°C)
{ +30 sec to -130 sec/month (at 0 to 45°C) (+: Advance, −: Delay)

Error is apt to increase as temperature rises or falls extremely. Therefore, time reading must be adjusted because it will be deviated from the exact point after long use. The Model H-200 incorporates the convenient special internal output (R7FA) for ±30 sec adjustment.

R7FA □ ····· ±30 sec adjustment (second matching)

By turning this special internal output from 0 to 1, second matching is carried out at the exact moment of rising (⌐↑) edge as follows.

{ When current sec reading is within 00 to 29, it is returned to 00.
{ When current sec reading is within 30 to 59, it is carried to 1 minute.

(3) Clock setting

The current year, month, day, day of week, hour, minute and second of the calendar clock has been factory-matched with the standard time in Japan. The clock can be set again so as to match the local time of the customer's country. For this purpose, implement the procedures below.

① Connect a programming device to the CPU, and set the current year, month, day, day of week. hour, minute and second to the special internal outputs (WRF01B to WRF01F).

WRF01B	Year [1990]	Set the current year in 4-digit BCD code.
WRF01C	Month and day [0524]	Set month and day in BCD code.
WRF01D	Day of week [0004]	Set day of week in BCD code. (0: Sunday, 1: Monday and so on)
WRF01E	Hour and minute [1400]	Set hour (24-hour base) and minute in BCD code.
WRF01F	Second [0000]	Set second in BCD code.

② While watching a clock, wait until the above time point is reached. Exactly at that monent, turn from 0 to 1 the special internal output (R7F9) which requests time setting. At the exact moment of the rising (⌐↑) edge of this special internal output (R7F9), the Model H-200 fetches the values set in WRF01B through WRF01F into the built-in calendar clock and starts it from the new time setting.

R7F9 □ ····· Requests time setting.

When turning this special internal output from 0 to 1, the values in WRF01B through WRF01F are set to the calendar clock the moment the rising edge (⌐↑) is detected, and the clock starts from the new time point.

Fig. 5.1 illustrates operations of the calendar clock in the above (1) through (3).

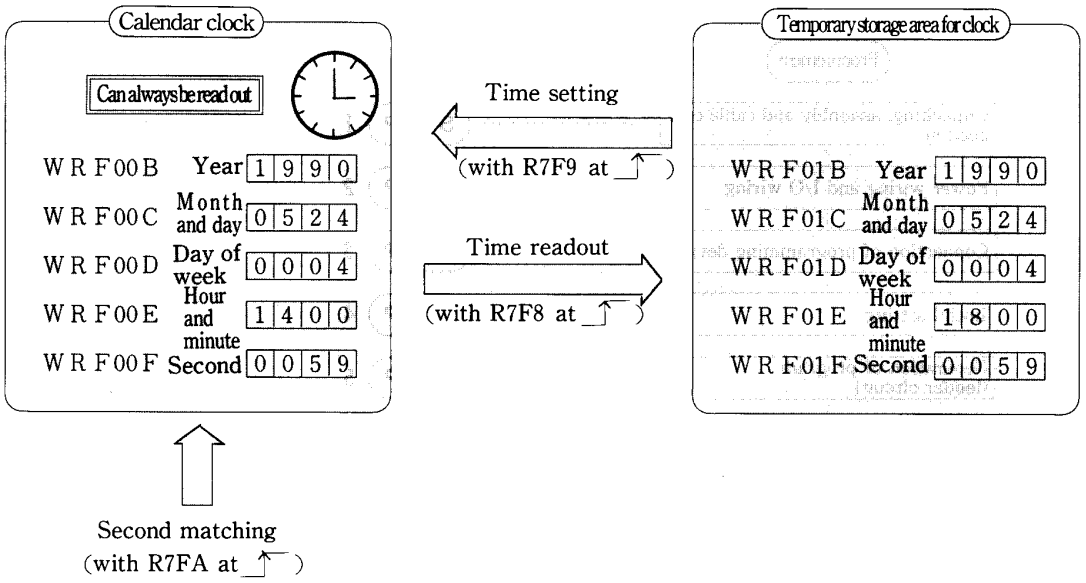


Fig. 5-1 Operation of Calendar Clock

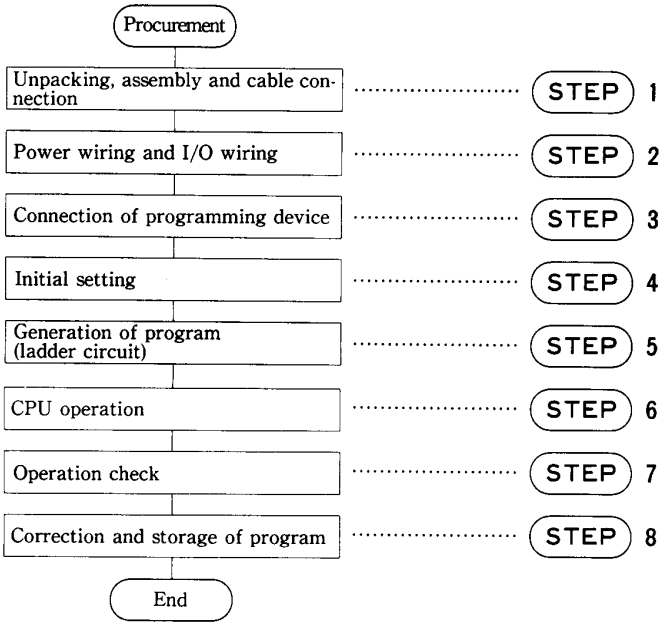
5

6. EXAMPLE OF SIMPLE OPERATION

This section describes the minimum contents to be implemented and concrete examples of operation in the process from unpacking to operation check with a simple relay ladder program in order to enable the users to understand the basic handling of the Model H-200.

6.1 Procedures for Operation Check

Implement the following procedures for checking the operation of H-200.



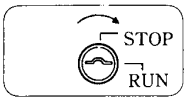
6.2 Minimum Contents to be Implemented and Key Points of Operation

Table 6-2-1 lists the minimum contents to be implemented and the key points of operation in each (STEP).

Table 6-2-1 Key Points of Operation (1/2)

(STEP)	Item	Contents to be implemented and key points of operation	Check
1	Unpacking	(1) Comparison of ordered item and delivered one. (2) Check if accessories are furnished. (3) Check for any damage.	
	Assembly	(1) Install power, CPU and I/O modules in basic/expansion bases. (2) Lithium battery has been factory-installed in CPU module.	
	Cable connection	(1) Connect the basic base and expansion base with the connecting cable. (2) Connect the CPU module and programming device.	

Table 6-2-1 Key Points of Operation (2/2)

STEP	Item	Contents to be implemented and key points of operation	Check
2	Power wiring	(1) Wire 220 V AC to the power module. (2) For 110 V AC, turn to 110 V the selection connector in the power module.	
	I/O module wiring (for receiving external input or activating external load)	Make I/O wiring with reference to Section 1 ("1" and "7") of this maural.	
3	Connection of programmer	Hook up programmer with CPU.	
4	Initial setting	(1) Initialize the CPU. (2) Trun CPU setting/CPU type setting to H-300, and memory type setting to RAM-08H. (3) Perform I/O assignment (copy function is unavailable).	
5	Generation of program (ladder circuit)	(1) Select EDIT for "programming mode." (2) Select WRITE NEXT. (3) → Input circuit element. → [ENT] → [STA] Generate circuit by repeating this sequence. Circuit generation ([STA] registers circuit in the memory of graphic input device.)	
6	CPU opration		
7	Operation check	(1) Check I/O status with the I/O LED display. (2) Check program operation by activating the monitor function. (3) Check if external load operates normally.	
8	Correction and storage of program	(1) Select the CHANGE function on the EDIT screen. (2) Correct program. (3) Save the program onto ROM or CMT.	

6.3 Concrete Example of Operation

A concrete example of programmable controller operation is introduced here when using the graphic input device (GPCL), portable graphic programmer (PGM-GPH) and instruction word programmer (PGM-CHH).

6.3.1 With Portable Graphic Programmer (PGM-GPH)

STEP 1 Unpacking, assembly and cable connection

Unpack the shipping crate, and check if the delivered product exactly matches the ordered one. Assemble the H-200 CPU unit, I/O units and PGM-GPH, and cable them according to the figure below.

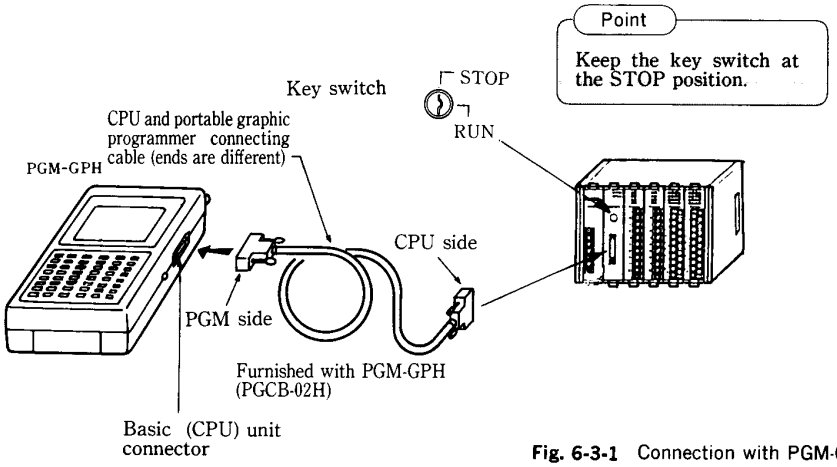
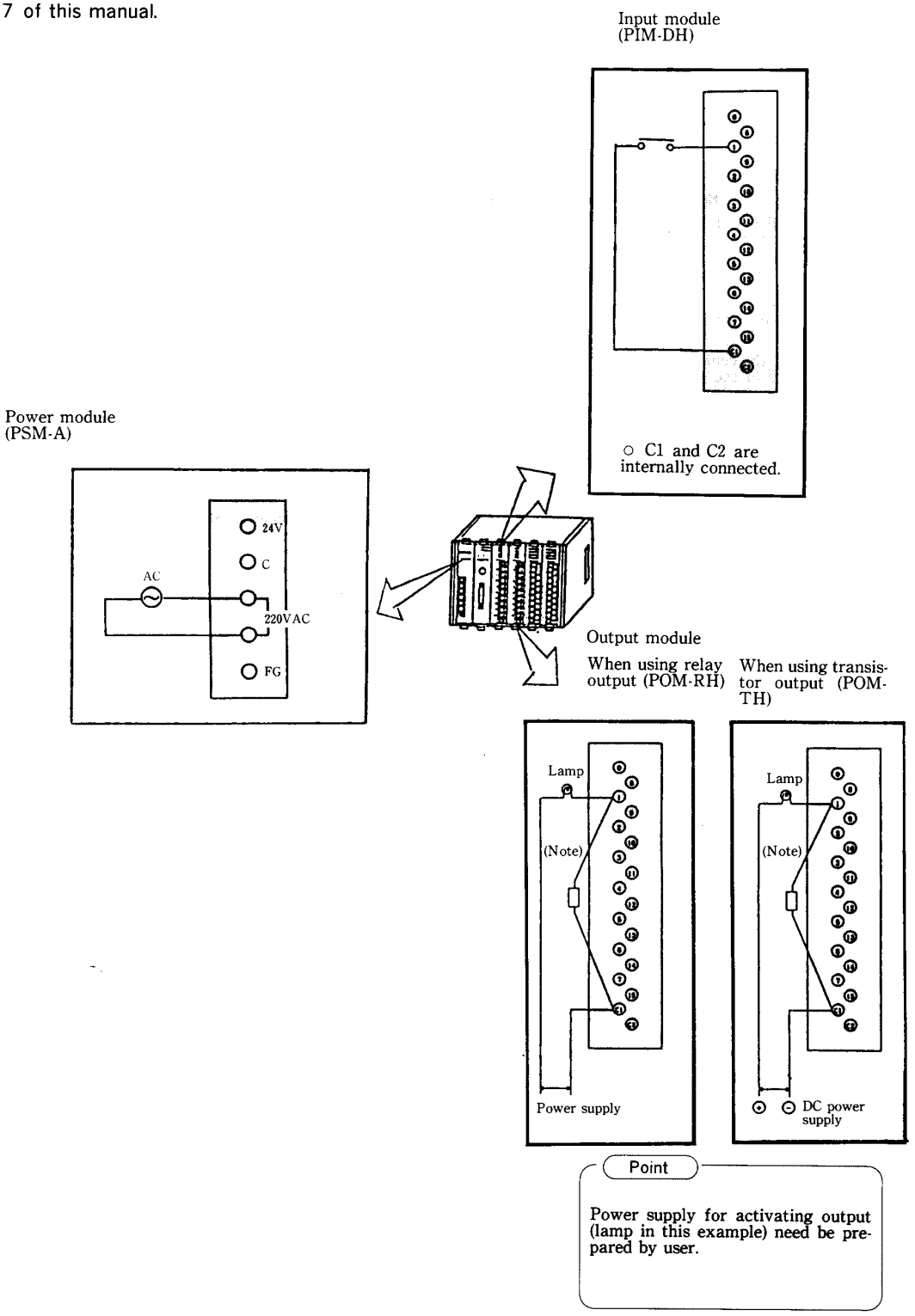


Fig. 6-3-1 Connection with PGM-GPH

STEP 2 Power wiring and I/O wiring

Make wiring as shown below for operating the flicker circuit in STEP 5.
For details, refer to 1.6 "Specifications of Power module," 1.7 "Specifications of I/O Modules" and Section 7 of this manual.

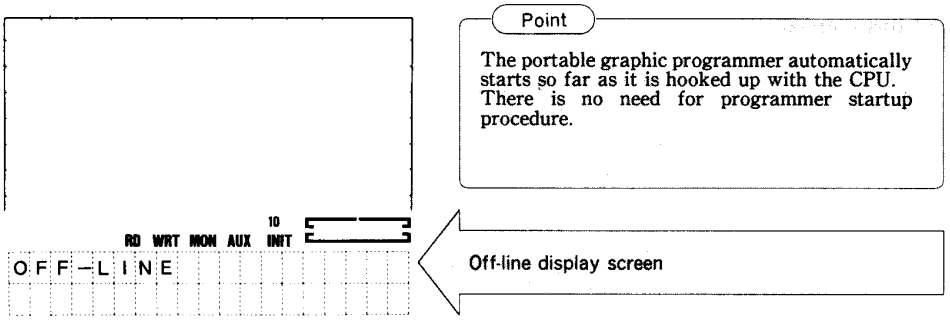


(Note) When using a lamp, provide a bleeder resistor for preventig rush current.

Fig. 6-3-2 Example of Wiring

STEP 3 Connection of programming device (PGM-GPH)

At about 12 sec after power is supplied from the CPU, a beep is emitted and the screen below appears.



Point

The portable graphic programmer automatically starts so far as it is hooked up with the CPU. There is no need for programmer startup procedure.

Off-line display screen

Point

Utilize the contrast adjusting variable resistor at the left of the programmer if the screen is hard to see or if no display appears.

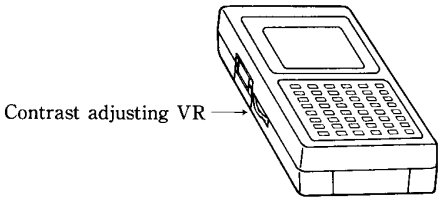
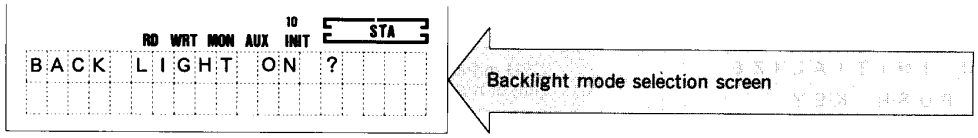


Fig. 6-3-3 Location of Contrast Adjusting VR

STEP 4 Initial setting

GRS

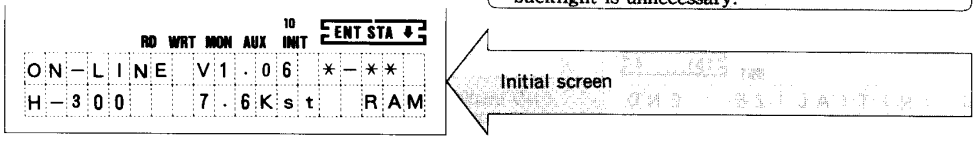


Display the initial screen with the backlight lit.

STA

Point

Press CLR (or key other than STA) when backlight is unnecessary.



Caution

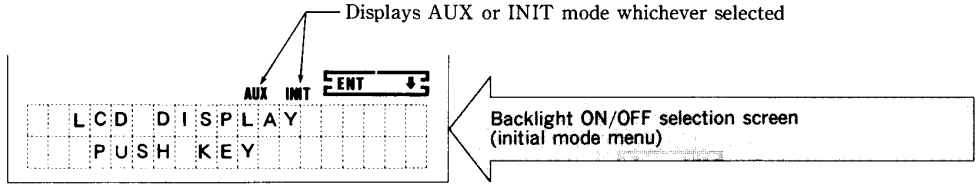
The above initial screen appears nomally. If error display appears, the relevant error must be eliminated with reference to the appendix "Error List" of the portable graphic programmer manual or the instruction manual.

(1) CPU initialize

The memory of CPU can be initialized. This is required after procuring a new CPU or for generating a new program.

Select the initial mode.

A/I (The AUX and INIT modes are selected alternately whenever pressing the A/I key.)



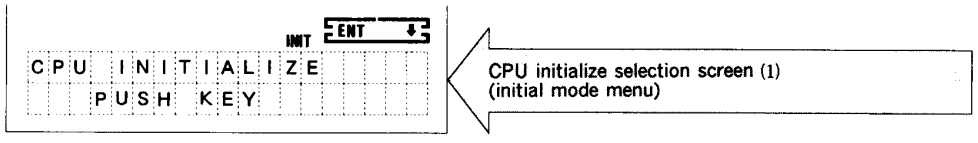
The back light ON/OFF selection screen of initial mode menu is displayed. Change this screen to the CPU initialize selection screen.

↓

Press 13 times

Point

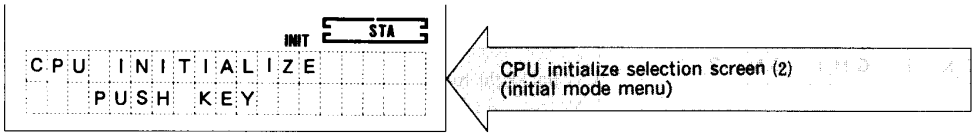
Use ↑ and ↓ keys for changing over the function (for CPU initialization, etc.) in the initial mode. For details, refer to the portable graphic programmer manual or Section 7 "INITIAL SETTING FUNCTION" of the instruction manual.



6

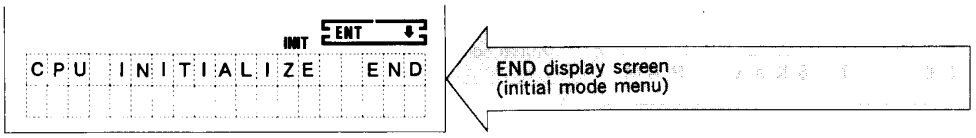
Select "CPU initialize."

ENT



Procedure is terminated when END display appears after CPU INITIALIZE WAIT is displayed for about 7 to 8 sec.

STA



(2) **I/O assignment**

Inputs and outputs are to be assigned so as to match system configuration. Prepare an assignment table in response to the mounted status of each module connected to the CPU unit.
Call the I/O assignment selection screen of initial mode menu.

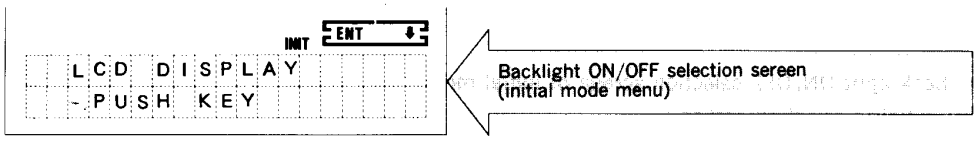
Point

If this I/O assignment operation is neglected after CPU initialization in the above (1) program input is impossible.

Select the initial mode again.

A/I

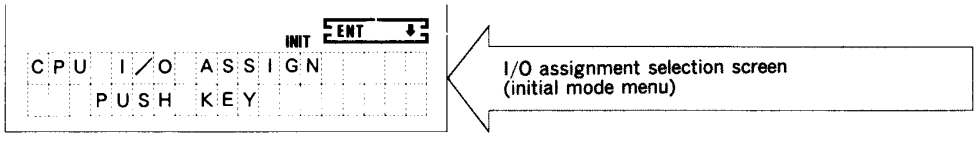
Press twice



The backlight ON/OFF selection screen of initial mode menu is displayed. Change this screen to the I/O assignment selection screen.

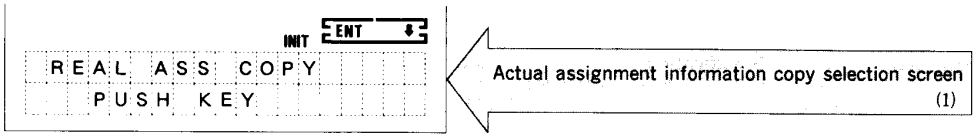
↓

Press 9 times



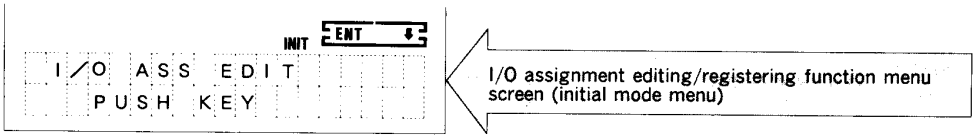
Select the I/O assignment function.

ENT



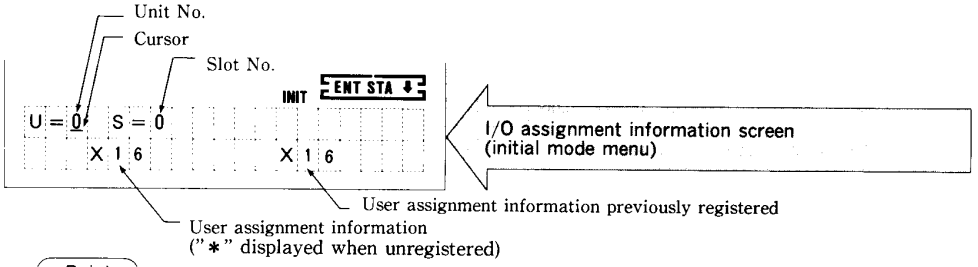
The H-200 does not have a copying function. Therefore, I/O assignment must be edited (by using EDIT function) and registered in the CPU. Display the I/O assignment editing/registering menu.
(Unless editing operation is carried out, assignment information is not set in the CPU.)

↓



Select the I/O assignment editing/registering function.

ENT



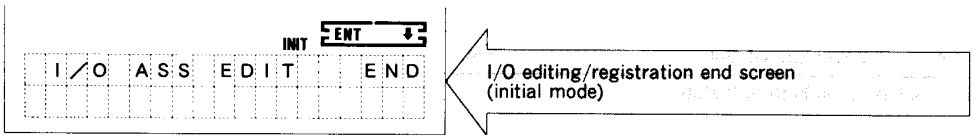
Point

The previously registered assignment information can be known by changing the unit No. or slot No. by using keys ←→↑↓.

Execute I/O editing and registration.

ENT STA

I/O ASS CHECK WAIT appears momentarily and ASSEMBLE WAIT is displayed for 7 to 8 sec. Then I/O ASS EDIT END is displayed, which indicates the end of I/O assignment.



Implement initial setting procedures other than mentioned above when required.
For details of initial setting, refer to "8.3 Initial Setting" of Section 8 in this manual.

6

Since the H-200 does not have a function for copying the I/O assignment table, it must be set according to the table below prior to program input.

Table 6-3-1 Assignment Code List

I/O module Classification		Specification	Model	I/O assignment code	No. of slots occupied	Istallation at remote end
Standard input	Input 8 points	24 V DC	PIM-D	X16	1	○
		110/220 V AC	PIM-A	X16	1	○
		24 V DC (source type)	PIM-DP	X16	1	○
	Input 16 points	24 V DC	PIM-DH	X16	1	○
		110/220 V AC	PIM-AH	X16	1	○
		24 V DC (source type)	PIM-DPH	X16	1	○
Standard output	Output 8 points	Relay output	POM-R	Y16	1	○
		Triac output	POM-S	Y16	1	○
		Transistor output	POM-T	Y16	1	○
		Transistor output (source type)	POM-TP	Y16	1	○
		Relay output (independent contact)	POM-RC	Y16	1	○
	Output 16 points contact)	Relay output	POM-RH	Y16	1	○
		Triac output	POM-SH	Y16	1	○
		Transistor output	POM-TH	Y16	1	○
		Transistor output (source type)	POM-TPH	Y16	1	○
Hybrid I/O	I/O 16 points I/O 32 points	DC input 8 points, transistor output 8 points	PHH-DT	B1/1	1	○
		TTL input 16 points, TTL output 16 points	PHM-TT	B1/1	1	○
Analog I/O	Input	Current 4 to 20 mA, 8 points	AGH-I	WX8W	1	×
		Voltage 0 to 10 V, 8 points	AGH-IV	WX8W	1	×
	Output	Current 4 to 20 mA, 4 points	AGH-O	WY8W	1	×
		Current 4 to 20 mA, 2 points	AGH-OD	WY8W	1	×
		Voltage 0 to 10 V, 4 points	AGH-OV	WY8W	1	×
		Voltage 0 to 10 V, 2 points	AGH-ODV	WY8W	1	×
Communi-cation	Link	Host link, coaxial cable	RIOM	B1/1 (Note)	2	○
		Parallel link, twisted pair cables	IOLH-T	Link	1	×
	Remote	Master station, twisted pair cables	RIOH-TM	Remote	1	×
		Slave station, twisted pair cables	RIOH-TL	Setting unnecessary	1	○
		Slave station unit, I/O 32 points	RIOH-DT	B1/1	1 (unit type)	○
Special	Counter	2-phase, 10 kHz, 16 bits	CTH	FUN3	1	×
Unused slot		Cover unused slot	CVM	DUM16	1	○

(Note) For RIOH, set B1/1 on both of 2 slots.

○ : possible
× : Impossible

On Unit No. and Slot No.

When using a programming device for I/O assignment, unit and slot numbers must be defined as shown in Fig. 6-3-4.

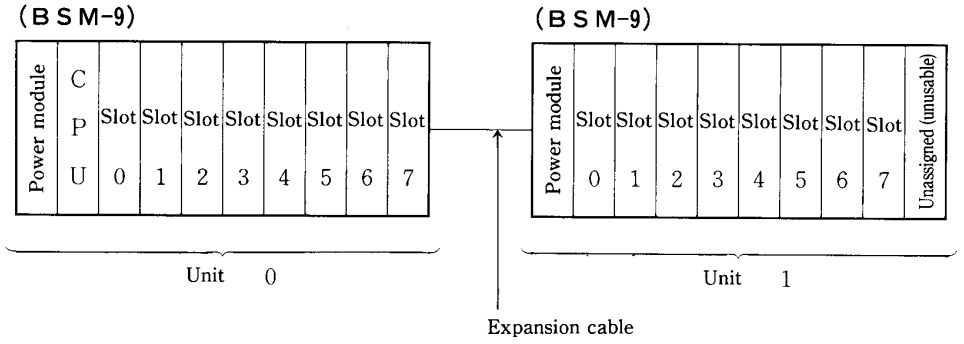


Fig. 6-3-4 Example of Unit and Slot No. Definition

6

STEP 5 Generation of program

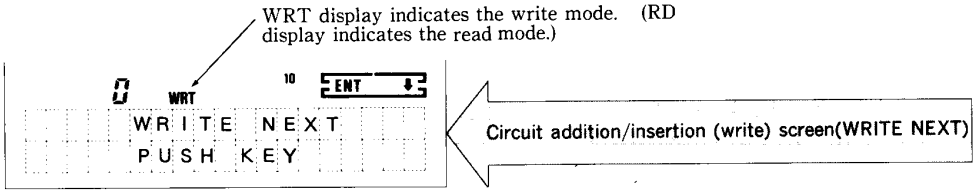
Select write mode.

R/W

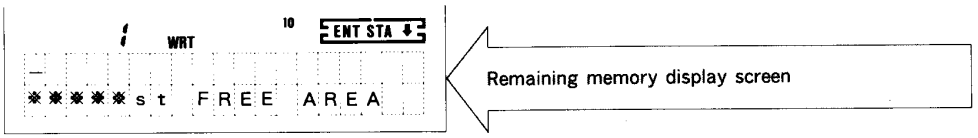
Press twice

Point

Whenever pressing R/W, read mode (RD) and write mode (WRT) are displayed alternately.



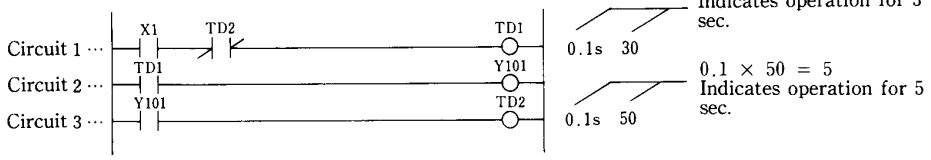
ENT



Example of Cricuit Input

Exemplified here is input of the flicker circuit below by using the portable graphic programmer.

Example of circuit: The flicker circuit below is to be input.



Circuit operation

At 3 sec after the external input X1 turns ON, the output Y101 turns on for 5 sec. Then, the output Y101 repeats a sequence of 3-sec OFF and 5-sec ON. Various flicker circuits can be generated by altering each set value of timer inputs TD1 and TD2.

Input of circuit 1: After the remaining memory display screen, make entry as follows.

Input "a" contact X1.

⊣⊢ X 1 ENT

Input "b" contact TD2.

⊣/⊢ TD 2 ENT

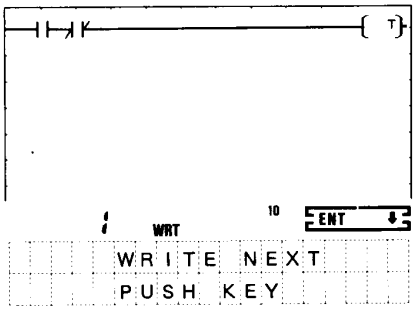
Input on-delay timer TD1.

-○- TD 1 ENT 0.1s 3 0 ENT

Store the circuit in the memory of CPU

STA

The screen at right is displayed.



Input of circuit 2:

ENT

Input "a" contact TD1.

⊣⊢ TD 1 ENT

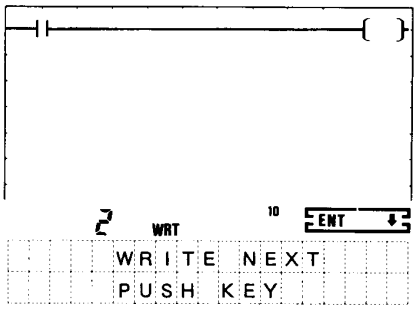
Input coil Y101.

-○- Y 1 0 1 ENT

Store the circuit in the memory of CPU

STA

Screen at right is displayed.



Input of circuit 3

ENT

Input "a" contact Y101.

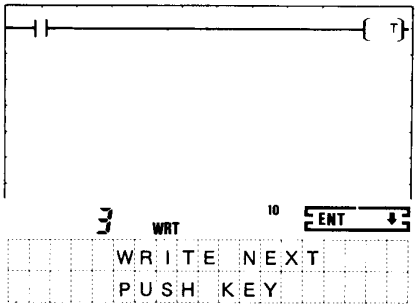
⊣⊢ Y 1 0 1 ENT

Input on-delay timer TD2.

-○- TD 2 ENT 0.1s 5 0 ENT

Store circuit in the memory of CPU.

STA



Point

When pressing STA program is written in the memory of CPU.

STEP 7 CPU operation

(1) Turn the key switch of CPU unit to the RUN position.
(2) Turn ON the switch connected to the input X 1.

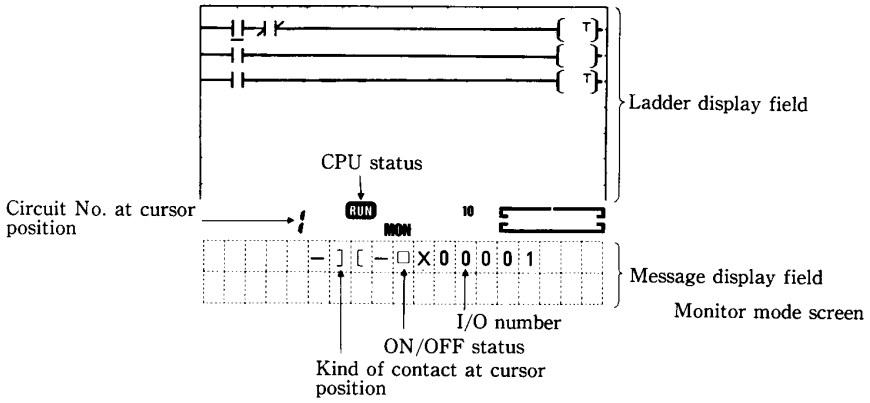
At 3 sec after X 1 turns ON, output (lamp) turns ON for 5 sec.
Then, the output (lamp) repeats a sequence of 3-sec OFF and 5-sec ON.

6

STEP 8 Operation check

CPU operation can be monitored with the portable graphic programmer. Monitor is roughly classifiable into I/O monitor and circuit monitor. A simple method of circuit monitoring is introduced here.
Select the monitor mode.

MON



ON/OFF status of contact and coil is displayed as shown in Table 6-3-2.

Table 6-3-2 ON/OFF Status of Contact and Coil

Item \ Kind		Status display		
		"a" contact input	"b" contact input	Output (Coil)
Conducting (ON)	Ladder display field	⊣ ■ ⊢	⊣/ ■ ⊢/	—(▌ ▐)—
	Message display field	–] [–■	–> <–■	– ( ) ■
Non-conducting (OFF)	Ladder display field	⊣ ⊢	⊣/ ⊢/	—( )—
	Message display field	–] [–□	–> <–□	– ( ) □

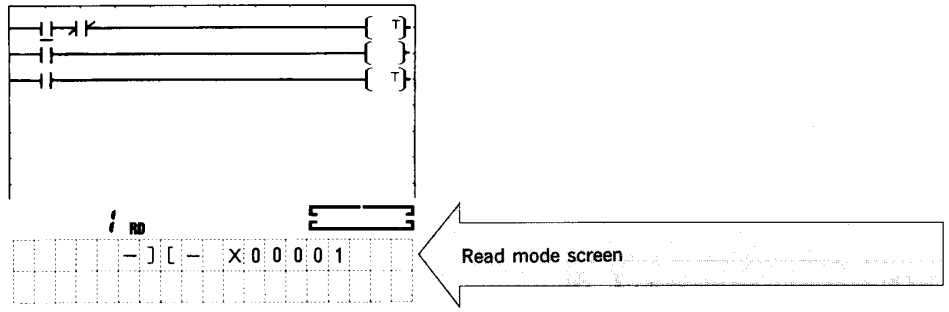
STEP 9 Correction and storage of program

Midway correction of erroneous circuit input (before pressing STA key) and that after pressing this key are instructed here.

(1) **Midway correction of circuit input (before pressing STA key)**
Move the cursor to the symbol to be corrected and delete it by CLR key. then make a correct entry.

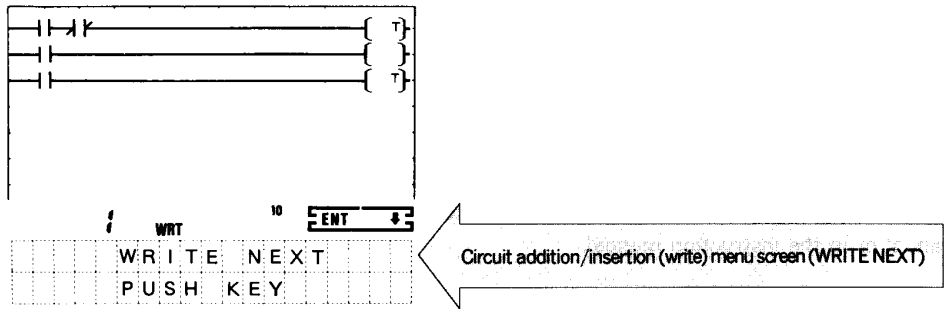
(2) **Correction of circuit after pressing STA key**
Select the read mode and read out a circuit to be corrected.

R/W 1 STA (Correction of circuit No. 1)
↑—Circuit No.



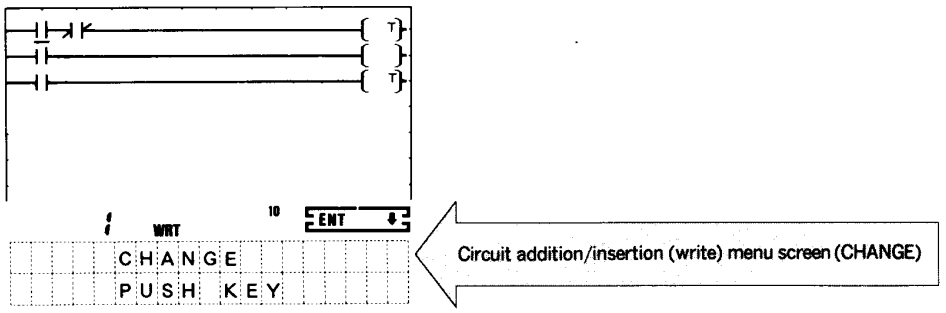
Select the write mode.

R/W



Switch over the WRITE NEXT screen to the CHANGE screen.

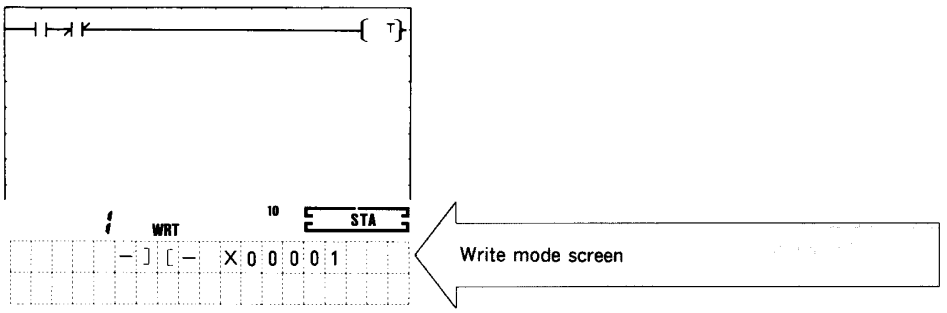
↓



6

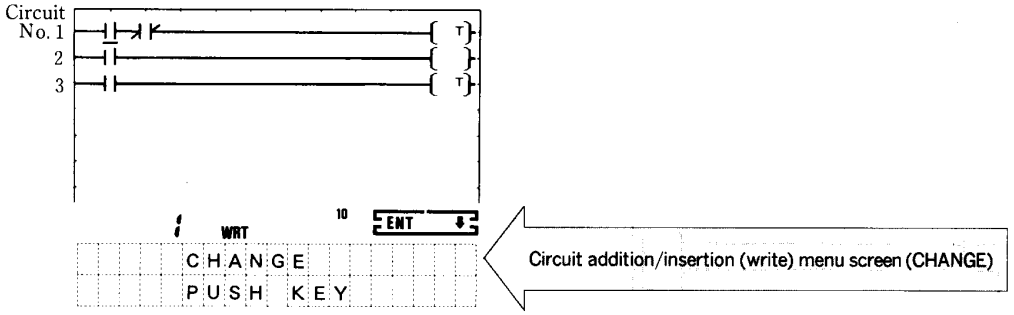
Select CHANGE.

ENT



Move the cursor to the symbol to be corrected and make a correct entry there. Then press STA key to write the circuit in the CPU memory.
The circuit addition/insertion (write) menu (CHANGE) is displeyed.

STA



(3) **Storage of program**

After correcting a program, save it onto the ROM, CMT, etc. and keep it in custody.
For detailed procedures, refer to 6.6 "Audio cassette (CMT) I/F" in the portable graphic programmer manual or in the instruction manual.

6.3.2 With Instruction Word Programmer (PGM-CHH)

STEP 1 Unpacking, assembly and cable connection
STEP 2 Power wiring and I/O wiring
STEP 3 Connection of programming device
STEP 4 Initial setting
STEP 5 Generation of program

} Same as with the portable graphic programmer (PGM-GPH)

Select the write mode.

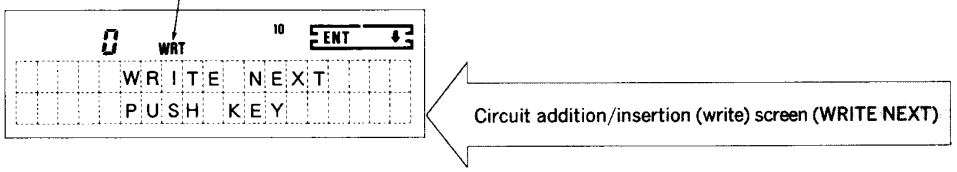
R/W

Press twice

Point

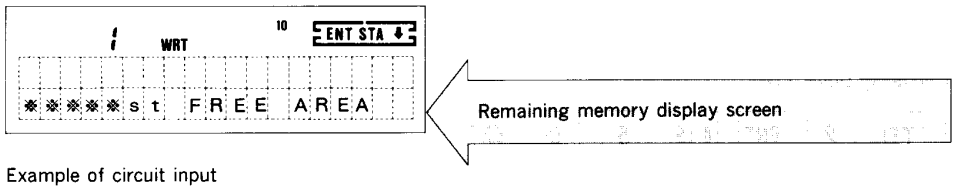
Whenever pressing R/W, the read mode (RD) and write mode (WRT) are displayed alternately.

WRT display indicates the write mode.
(RD display indicates the read mode.)



Circuit addition/insertion (write) screen (WRITE NEXT)

ENT

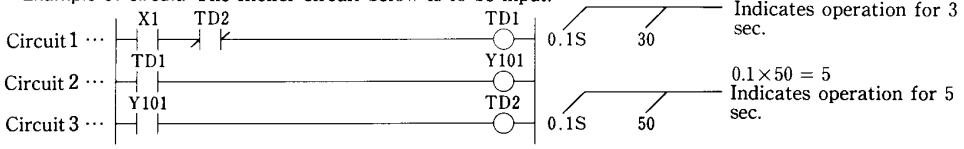


Remaining memory display screen

Example of circuit input

Input of the flicker circuit below with the instruction word programmer is exemplified here.

Example of circuit: The flicker circuit below is to be input.



0.1×30 = 3
Indicates operation for 3 sec.

0.1×50 = 5
Indicates operation for 5 sec.

Circuit operation: Refer to circuit operation in STEP 5 of 6. 3. 1.

Input of circuit 1: After the remaining memory display screen, make entry as follows.

Input "a" contact X1.

LD X 1 ENT

Input "b" contact TD2.

ANI TD 2 ENT

Input on-delay timer TD1.

OUT TD 1 ENT 0.1S 3 0 ENT

Store the circuit in the memory of CPU.

STA

The screen at right is displayed.

Input of circuit 2

ENT

Input "a" contact TD1.

LD TD 1 ENT

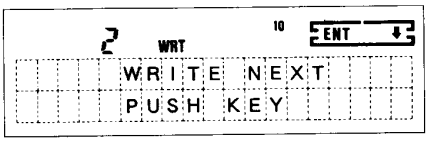
Input coil Y101.

OUT Y 1 0 1 ENT

Store the circuit in the memory of CPU.

STA

The screen at right is displayed.



Input of circuit 3

ENT

Input "a" contact Y101.

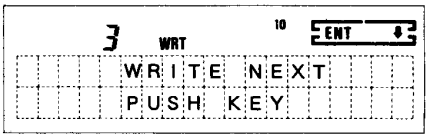
LD Y 1 0 1 ENT

Input on-delay timer TD2.

OUT TD 2 ENT 0.1S 5 0 ENT

Store the circuit in the memory of CPU.

STA



Point

When pressing STA, program is written in the CPU memory.

STEP 7 CPU operation

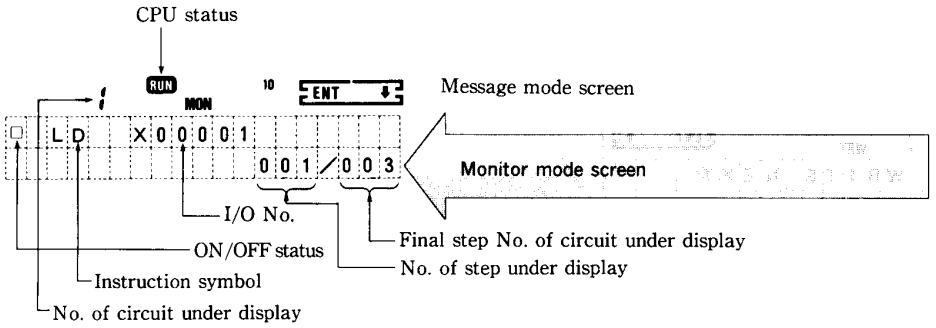
(1) Turn the key switch of CPU module to the RUN position.
(2) Turn ON the switch connected to the input X 1.

At 3 sec after X 1 turns ON, output (lamp) turns ON for 5 sec.
Then, the output repeats a sequence of 3-sec OFF and 5-sec ON.

STEP 8 Operation check

CPU operation can be monitored with the instruction word programmer. Monitor comes in two categories; I/O monitor and circuit monitor.
A simple method of circuit monitoring is introduced here.
Select the monitor mode.

MON



ON/OFF status of contact and coil is displeyed as shown in Table 6-3-3.

Table 6-3-3 ON/OFF Status of Contact Coil

Item \ Kind	Status display		
	"a" contact input	"b" contact input	Output
Conducting (ON)	■ LD R0 ■ AND R0 ■ OR R0	■ LDI R0 ■ ANI R0 ■ ORI R0	■ OUT R0
Non-conducting (OFF)	□ LD R0 □ AND R0 □ OR R0	□ LDI R0 □ ANI R0 □ ORI R0	□ OUT R0

STEP 9 Correction and storage of program

Midway correction of erroneous input of circuit (before pressing STA key) and that after pressing this key are explained here.

(1) **Midway correction of circuit input (before pressing STA key)**
Read an instruction to be corrected and delete it by CLR key, then make correct input.

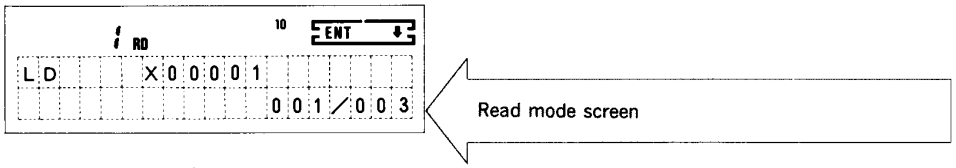
(2) **Correction of circuit after pressing STA key**

6

Select the read mode and read a circuit to be corrected.

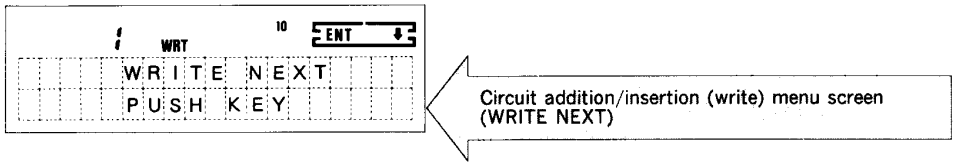
R/W 1 STA (Correction of circuit No. 1)

Circuit No.



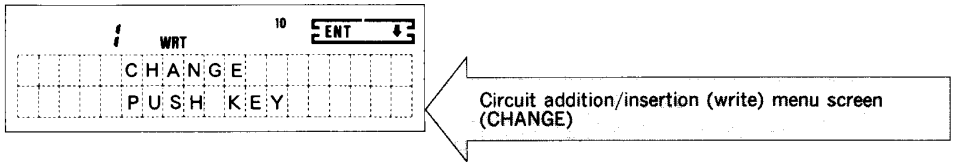
Select the write mode.

R/W



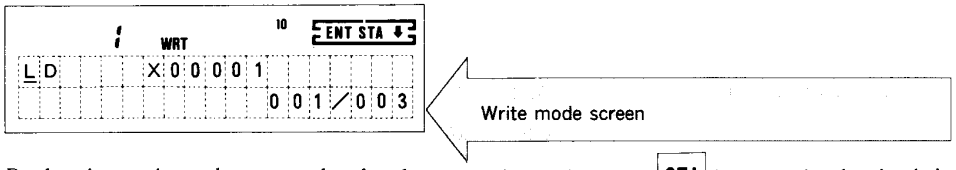
Switch over display from the WRITE NEXT screen to the CHANGE screen.

↓



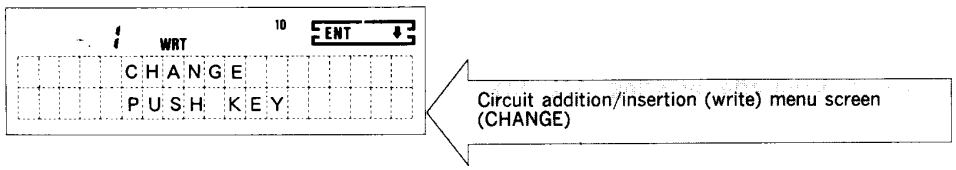
Select CHANGE.

ENT



Read an instruction to be corrected and make correct input, then press STA key to write the circuit in the CPU memory. The circuit addition/insertion (write) menu screen(CHANGE) is displayed.

STA



(3) Storage of program

After correcting a program, save it into CMT, etc. and keep the medium in custody. For detailed procedure, refer to 6.6 "Audio cassette (CMT) I/F" in the instruction word programmer manual or in the instruction manual.

7. INSTALLATION

7.1 Mounting

Figs. 7-1 and 7-2 show the external dimensions of H-200 and the drilling locations for mounting it, respectively.

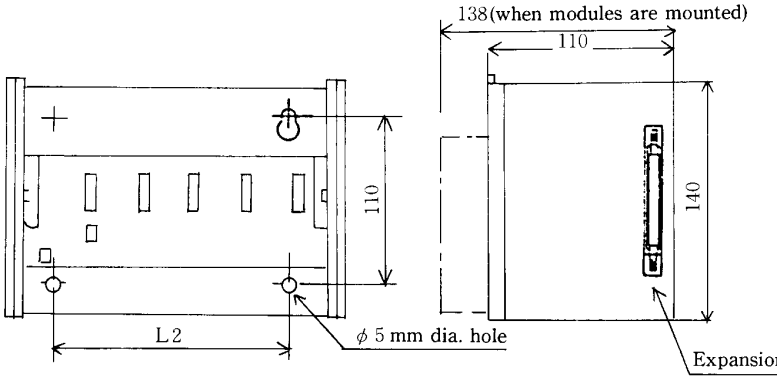


Table of Dimensions

Base	L1 (external dimension)	L2 (mounting dimension)
3 slots	160	80
4 slots	195	120
5 slots	230	160
6 slots	265	200
7 slots	300	240
9 slots	370	310

Fig. 7-1 External Dimensions

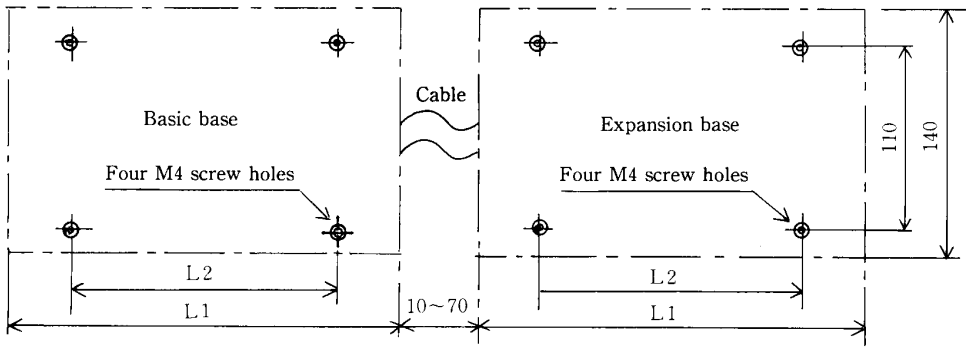
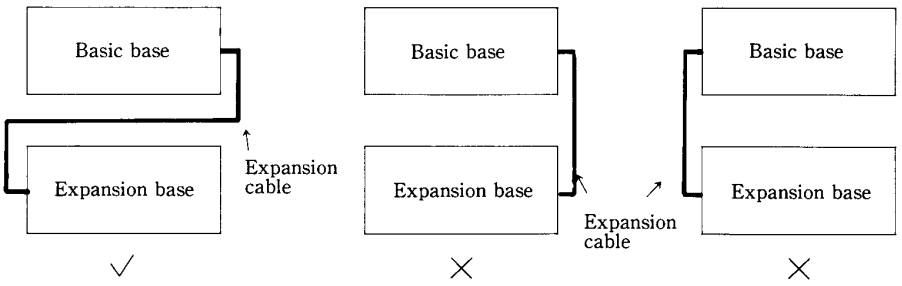


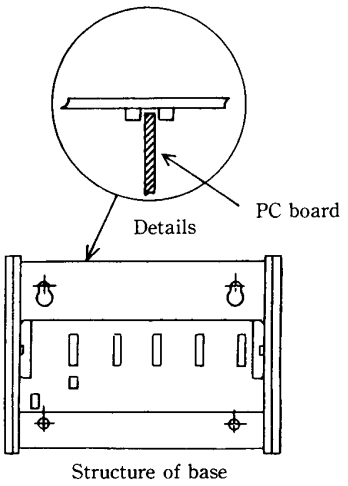
Fig. 7-2 Drilling Locations for Mounting

(1) Each base alone with no module accommodated must be mounted to the control panel.
(2) An expansion unit must, as a rule, be mounted at the right side of the basic unit. Provide a space of 10 to 70 mm between these units when using the 10 cm expansion cable ⟨CNM-01⟩ .
(3) Pay attention to the cable connecting direction when mounting the basic and expansion bases vertically with the 60 cm expansion calle ⟨CNM-06⟩. If connection is wrong, the instrument won't operate. Furthermore, modules might be damaged.



(4) **Mounting of module**

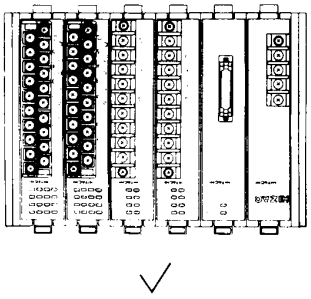
At the top and bottom, each base has a groove as shown in the figure at right. Push in the PC board of each module, matching it with the top and bottom grooves.

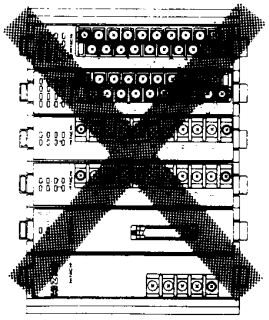


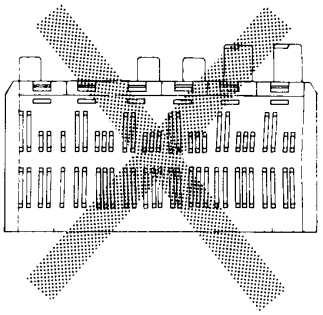
Structure of base

(5) **Mounting direction**

Each unit is mountable upside down. However, it must not be laid on its side or turned inside out.







(6) **Installation interval**

① Provide a space of 50 mm or more at the top and bottom of each unit for facilitating ventilation and maintenance. Also secure a free space of 10 mm or more at the left and right for ventilation.

② During installation, pay strict attention not to let fragments from drilling or wiring fall into the programmable controller.

③ Avoid installation right above an equipment which radiates much heat (such as a heater, transformer or large-capacity resistor).

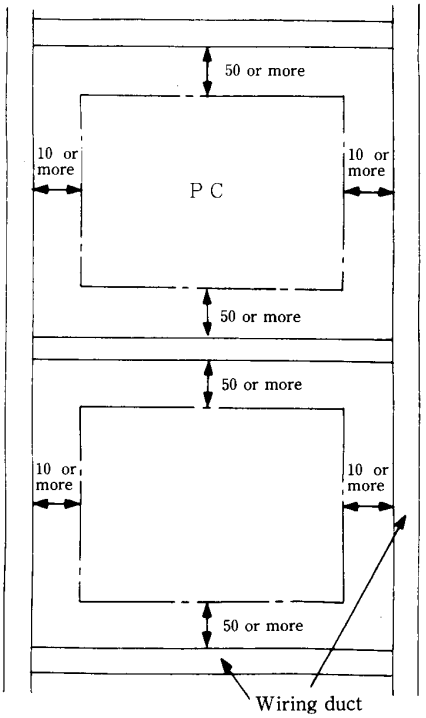
④ Secure a distance of 200 mm or more from a high tension cable (3,000 V min.) or power cable.

(7) **Installation environment**

① Avoid locations which receive direct sunlight, or which are subjected to condensation or exposed to wind and rain.

② Installation is unallowable where dust, oil smoke, conductive dust or corrosive gas is excessive.

③ Do not install the programmable controller at locations where vibration or shock will be directly applied.



7.2 Power Wiring

Power wiring of H-200 is exemplified in Fig. 7-3.

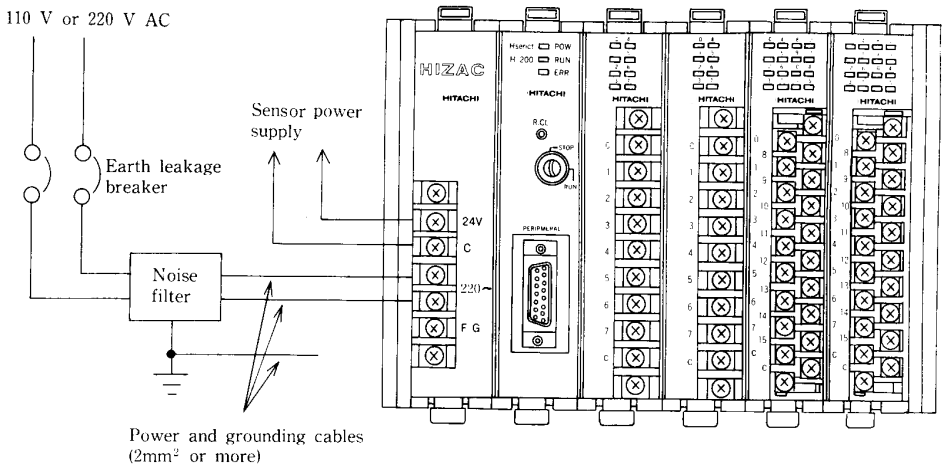
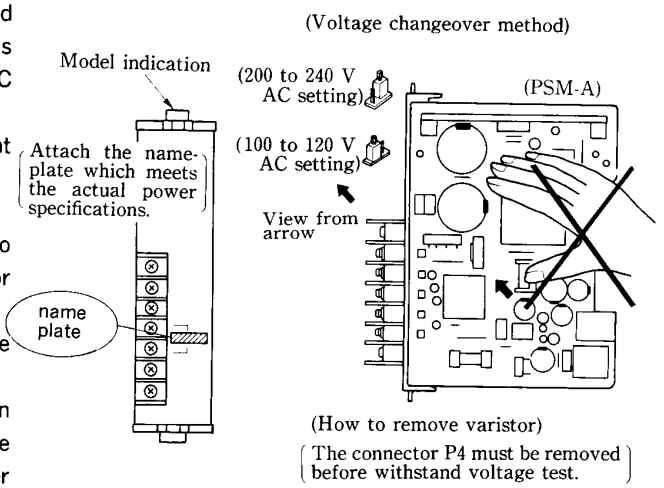


Fig. 7-3 Example of Power Wiring

(1) Line voltage
This instrument can operate on either 110 V or 220 V AC system. Although the instrument has been set for 220 V AC as a standard (factory setting), setting can be changed as shown in the figure at right when 110 V AC power supply is required.



(2) Use a power cable of 2 mm² or more to prevent occurrence of voltage drop.

(3) Grounding
Connect the grounding terminal (FG terminal) to make 100 Ω or less using a cable of 2 mm² or more.
① Grounding can be shared between the instrument panel and relay panel.
② Common grounding must be avoided with an equipment which might induce a large noise as a high-frequency furnace, large-scale power panel (beyond a few kW), thyristor converter and electric welding machine.
③ Use of a noise filter is recommended when power supply environment is undesirable.

(4) Insulation resistance and dielectric strength tests
A varistor (470 V class) is built in the power module for suppressing lightning surge. Be sure to disconnect the varistor before performing an insulation resistance or dielectric strength test. Otherwise, the power module might be damaged.

(5) Use the PSM-D when line voltage is 24 V DC.

7.3 I/O Wiring

Fig. 7-4 shows each external wiring method for DC input and AC input.

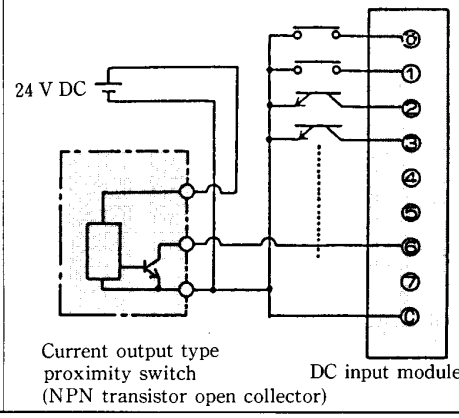
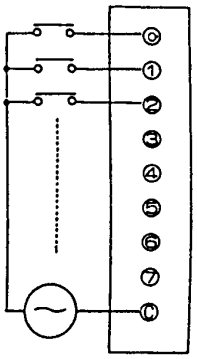
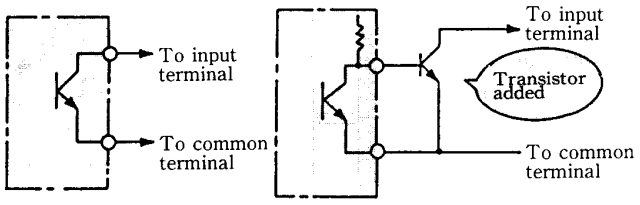
Item	DC input (PIM-D, DH, etc.)	AC input (PIM-A, AH, etc.)
External wiring		

Fig. 7-4 Input Wiring

(1) **DC input module wiring**

① The H-200 incorporates the power supply (24 V DC) for external inputs. When each input terminal (X0, X1...) is short-circuited with the common terminal (C), input is turned on. As a result, a current of about 10 mA flows from the PC to the external input contact.

② Sensors such as proximity switch and photoelectric switch are directly connectable when they are of current output type (NPN transistor open collector output). Sensors of voltage output type must be connected to the input terminal via a transistor.



Current output type
✓ (directly connectable)

Voltage output type
× (requiring addition of transistor)

③ Although the instrument is sensitive to an input current within 4 to 6 mA, secure 7 mA or more for reliable ON operation and 1 mA or less for reliable OFF operation.

Caution: When connecting a 2-wire type proximity switch or an LED display-equipped limit switch, check the input impedance and select one within the above current specifications.

④ As a sensor power supply, 24 V DC of the power module 〈PSM-A〉 can be used. Its current value I is represented by:

I = CH3 capacity of 〈PSM-A〉 (450 mA) − CH3 current to be consumed by I/O modules

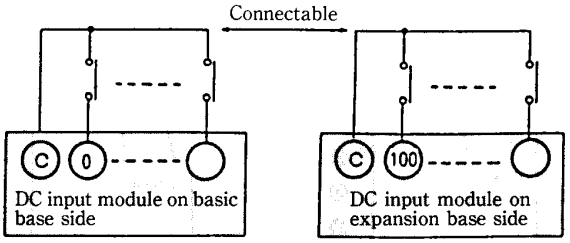
⑤ When installing a 24 V DC switching regulator as a sensor power supply, connect the negative pole of power supply and the common terminal of DC input module. (See the above DC input wiring.)

⑥ Connection of common terminal between DC input modules.

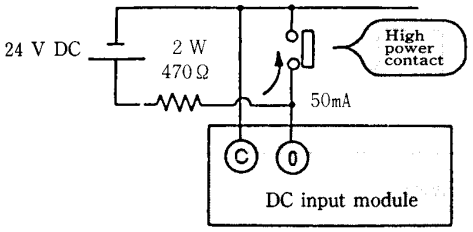
The common terminals need not be connected between the DC input modules in the same base. (They are connected with the mother board in the base.)

It is recommended to separate the common terminal of DC input between the basic base and expansion base.

However, connection of the common terminals, if unavoidable, does not present a problem.



⑦ Prevention of poor contact of high power contact



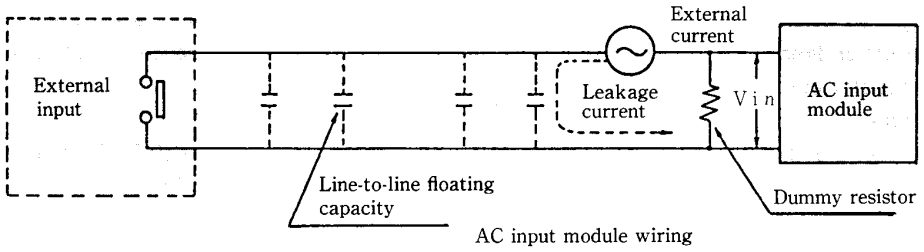
When an external contact is closed, a current of about 9 mA flows through it. Therefore, use a contact which does not incur poor contact at that current level. If you must employ a high power contact, an adequate current must be supplied to the contact via a resistor as shown at left in order to prevent poor contact.

⑧ Lenghth of input wiring

Input wiring must be 30 m max. If wiring beyond 30 m is inevitable, the input wire and output wire must be separated completely. Even in this case, wiring length must not exceed 100 m.

(2) **AC input module wiring**

① When using an AC input module, a voltage may appear at the input terminal though no signal is fed. This phenomenon is apt to occur as wiring length increases.



AC input module wiring

This phenomenon occurs for the following reason.

Even if the contact of external input is open, a voltage is applied to the input terminal because of a leakage current due to the floating capacity between lines. When the voltage rises beyond the maxicum OFF voltage, the input module comes into the same status as when it receives an input signal. Therefore, the module operates. To prevent this, reduce the terminal voltage due to capacitance to 1/2 or less of the maximum OFF voltage by diminishing the impedance of input module with a dummy resistor connected in parallel with the input terminal.

7.4 Output Wiring

Fig. 7-5 shows examples of relay output, transistor output and triac output wirings.

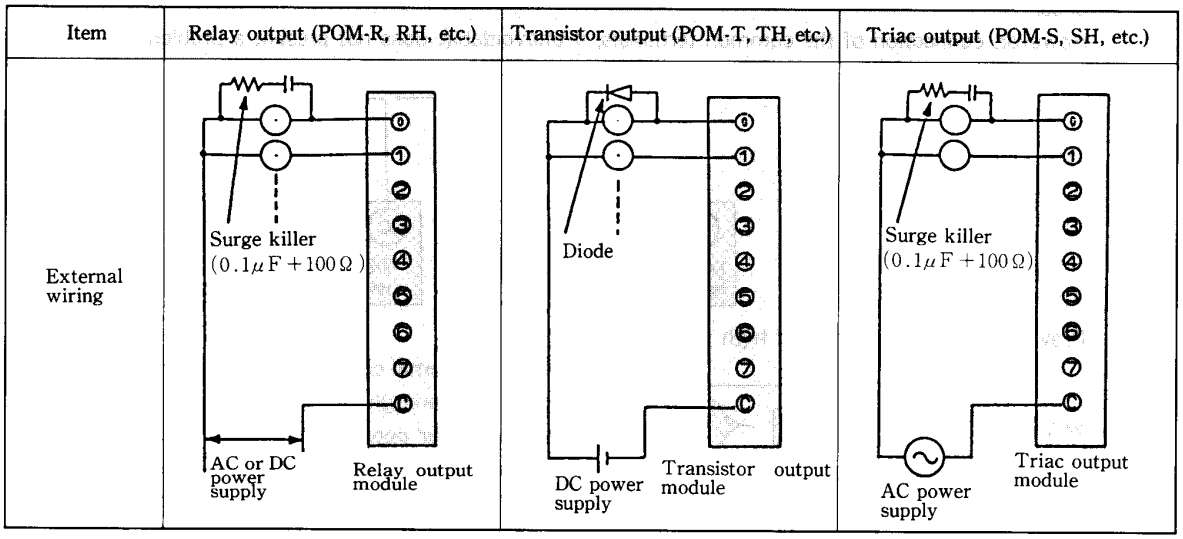


Fig. 7-5 Output Wiring

(1) **Relay output module wiring**

① **Service life of relay contact**

○ Relay contact has a service life of 200,000 or more operations under 120 V AC and 2 A resistive load, and 1 million or more operstions when load is applied from an electromagnetic contactor (Hitachi H10C: 45 VA at ON operation and inductive load 9 VA after ON operation).

○ The service life is inversely proportional to the square of current (life quadruples when current is reduced to half). So attention must be paid since the life is significantly shortened when breaking a rush overcurrent or directly driving a capacitor load. In case the circuit need be opened/closed frequently, use of a transistor output module or triac output module is recommended.

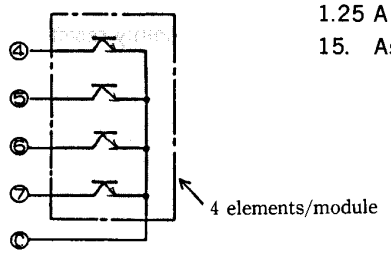
② **Surge killer**

For an inductive load beyond a coil capacity of 10 VA, connect a surge killer (0.1 μF capacitor + 100 Ω resistor) in parallel with load. Note that a flywheel diode must be connected to DC load.

(2) **Transistor output module wiring**

① Used to control a DC load. Although a protection circuit is incorporated against surge which might cause inductive load, it is recommended to suppress occurrence of surge by connecting a flywheel diode (with current 1 A and peak inverse voltage 250 V as a standard) in parallel with inductive load as shown in the figure above.

② The transistor used is a composite one consisting of 4 elements. So maximum current is restricted to 1.25 A for 4 circuits across terminals 0 to 3, 4 to 7, 8 to 11 and 12 to 15. Assign load within the maximum load current.



(3) **Triac output module wiring**

① Used to control AC load.

② Since a snubber circuit is built in the module for protecting the triac, a leakage current flows (3 mA with 220 V AC and 1.5 mA with 110 V AC).
When an extremely small current load or lamp load is connected, the module might turn ON in error or might not be able to turn OFF. Such a phenomenon due to leakage current can be prevented by connecting a dummy load (aforementioned 0.1 μF capacitor + 100 Ω resistor) in parallel with load.

7.5 Terminal Layout

Fig. 7-6 shows the terminal layout of power module and input/output modules.

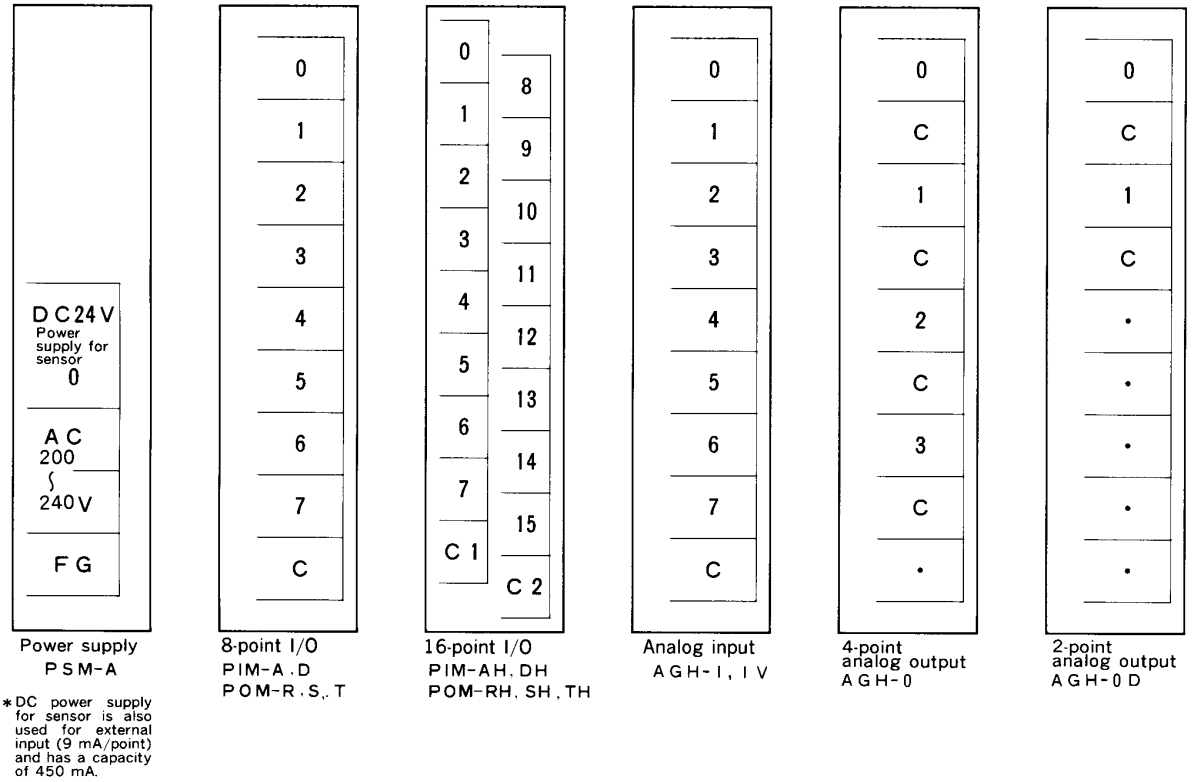
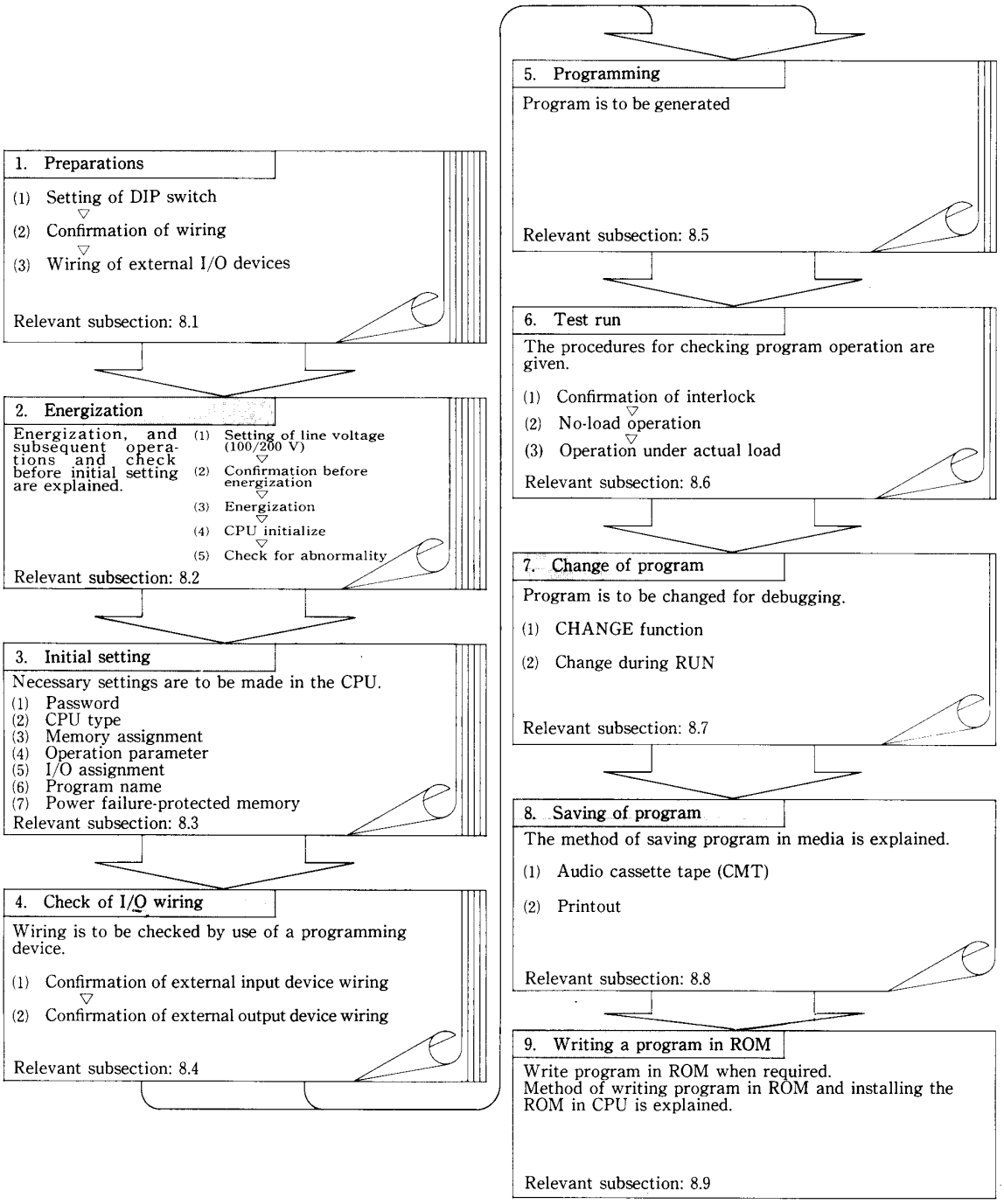


Fig. 7-6 Terminal Layout of Power Module and I/O Modules

(1) Each terminal screw has a size of M3.5. When using a solderless terminal, its outside diameter must not exceed 8 mm.
Up to 2 solderless terminals are allowed for the same terminal screw. Do not fasten three or more terminals at the same time.

8. DEBUG AND OPERATION

This section contains the procedures to be implemented till completion of test run after installation of the programmable controller as well as cautions and convenient functions.
First of all, the procedures are outlinded below.



This section details each of the above items.

8

8.1 Preparations

(1) Setting of DIP switch
▽
(2) Confirmation of wiring
▽
(3) Wiring of external I/O devices
▽
(4) Mounting of battery

(1) **Setting of DIP switch**

When connecting the CPU and a personal computer via cable, set the selectors 1 and 2 of the DIP switch (SW3) according to the baud rate, referring to Table 8-1. The selector 3 of this switch must be set at ON (factory-set at OFF) when using counter module.

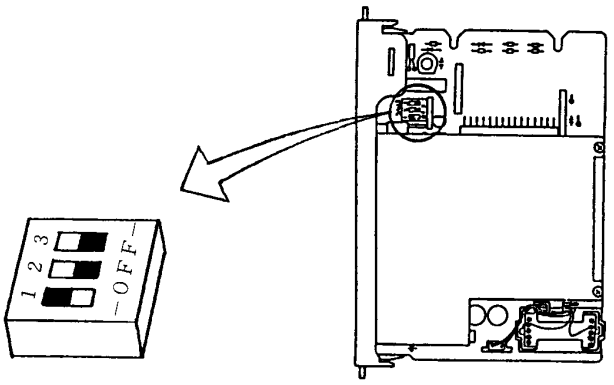


Fig. 8-1 DIP Switch (SW3)

Table 8-1 DIP Switch Settings

No.	Function	Setting			
		2400bps	4800bps	9600bps	Unusable
1	Bit rate setting for connection of personal computer	ON	ON	OFF	OFF
2		ON	OFF	ON	OFF
3	Methode of input/ output processing	ON: refresh processing. OFF: direct processing.			

(2) **Confirmation of wiring between units**

After mounting the I/O modules, connect cable between units before mounting the CPU, and basic and expansion unit power modules.

Reference
Section 7 INSTALLATION

(3) **Wiring of external input/output devices**

Connect the input module of H-200 and an input device (such as a switch). Also connect the output module and an output device (target of control).

For terminal layout, connecting method and necessary parts (electric wire, solderless terminal, etc.), refer to the explanatory diagram attached on the front of each unit and the following.

Reference
Section 1 "CONFIGURATION AND SPECIFICATIONS" ... 1.7 "Basic Specifications of I/O Modules"

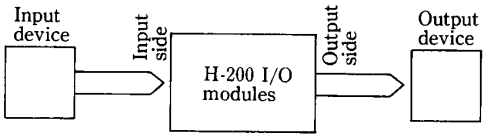


Fig. 8-2 Wiring of External I/O Devices

8.2 Energization

(1) Setting of line voltage
▽
(2) Confirmation before energization
▽
(3) Energization
▽
(4) CPU initialize
▽
(5) Check for abnormality

(1) **Setting of line voltage**

The H-200 can operate on either power supply of 110 V AC or 220 V AC whichever selected by the switching connector CN1 on the power module PC board. Select the position of the connector according to the actual power supply voltage as per Fig. 8-4. When the 110 V AC side is selected, the pasted voltage nameplate must be teared off. Note that the instrument has been factory-set to 220 V AC.

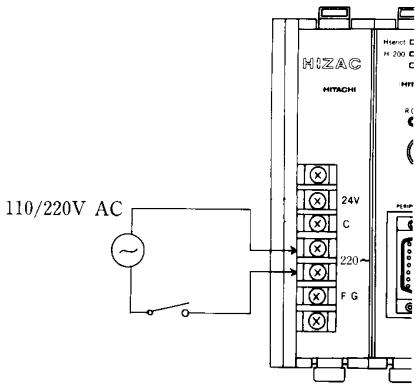


Fig. 8-3 Connection of Power Cable

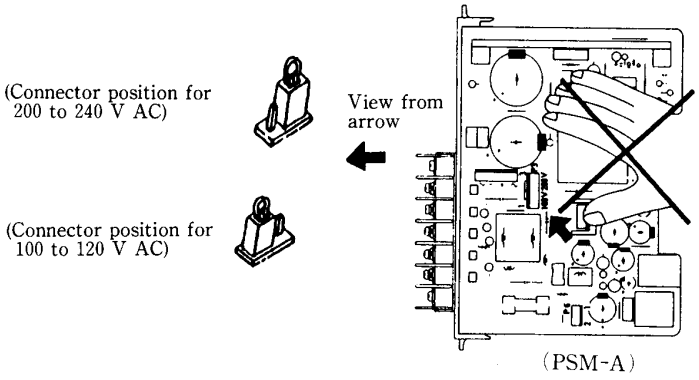
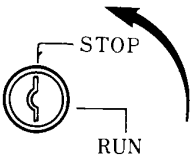


Fig. 8-4 Voltage Selecting Method



<CPU mode selector switch>

(2) **Confirmation before energization**

Confirm the items below for ensuring safety.

- ○ Turn the mode selector switch of CPU to the STOP position.
- ○ Make sure that the load power of external input/output is turned OFF.

Caution:

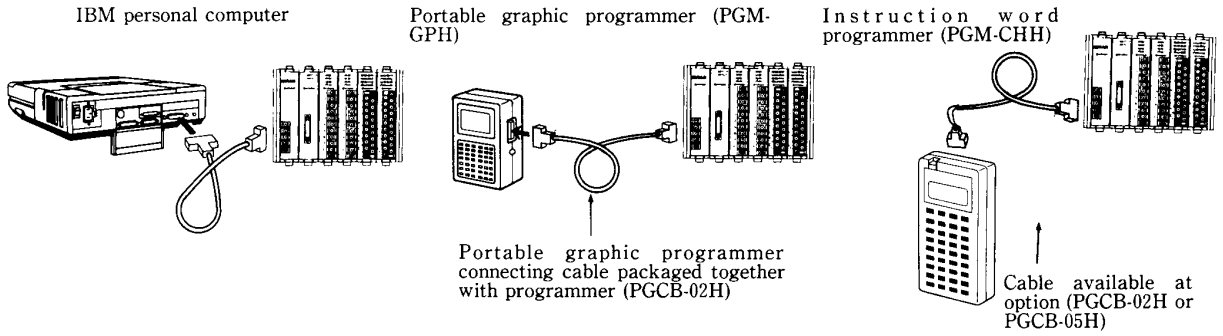
If the instrument receives 220 V under 110 V setting, it will be damaged. So confirm the above items without fail.

(3) **Energization**

In case there are both basic and expansion units, turn on each power supply.

(4) **CPU initialize**

Connect each programming device and CPU first.



Caution:

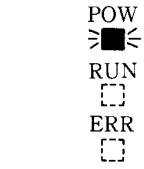
If the CPU is initialized, the contents of program memory and various settings are all cleared. This effect must be confirmed before attempting to initialize the CPU.

For details of CPU initialization, refer to the manual of each programming device or the items below of the instruction manual.

Programming device	Reference
Portable graphic programmer (PGM-GPH)	Section 7 "INITIAL SETTING FUNCTION"
Instruction word programmer (PGM-CHH)	Section 7 "INITIAL SETTING FUNCTION (manual)"

(5) **Check for abnormality**

- ○ Check the display section of CPU unit.
- ○ Make certain that the POW lamp of remote I/O is lit.



Display section of CPU unit
〈normal status〉

Display section of remote I/O
〈normal status〉

8

8.3 Initial Setting

(1) Password
(2) CPU type
(3) Memory assignment
(4) Operation parameter
(5) I/O assignment
(6) Program name
(7) Power failure-protected memory

The items shaded in Table 8-2 (2. "CPU type" and 5. "I/O assignment") must be set without fail before programming. Other items need only be set later whenever required.

Set "parameters" necessary for program generation, program transfer to CPU, etc. If the CPU is not yet initialized, CPU initialization must precede initial setting.
Contents of setting are detailed below.

Table 8-2 Initial Settings (1/2)

<table>
<tr><th>No.</th><th>Function</th><th>Description</th><th>Necessity</th></tr>
<tr><td>1</td><td>Password</td><td>○Registers a 4-digit hexadecimal password in program.
Once a password has been registered for program, the program does not allow display, change, etc. unless the correct password is input.
Password has not been factory-set.</td><td>Use this function to preserve program secrecy.</td></tr>
<tr><td>2</td><td>CPU type</td><td>○Sets the CPU name and memory capacity for programming.
For H-200 CPU, make setting as follows.
<table><tr><th>Memory</th><th>CPU type</th><th>Memory type</th></tr><tr><td>Without memory pack</td><td>H-300</td><td>RAM-08H</td></tr><tr><td>With MPH-8R</td><td>H-300</td><td>RAM-08H</td></tr></table>Note: When a ROM is provided, confirm that a correct program is written in it.</td><td>CPU type must be set whenever you conduct programming.</td></tr>
<tr><td>3</td><td>Memory assignment</td><td>○Assigns ladder program area in memory capacity and sets the range of data memory.</td><td>○Momery assignment is required when using data memory.</td></tr>
</table>

Table 8-2 Initial Settings (2/2)

4	Operation parameter	○Operation control Setting is required for controlling start/stop according to external input. (Start/stop is controlled by logical AND value with assigned external input and RUN selection of the key switch.) Unless this parameter is set, operation automatibally starts when turning the key switch to RUN. ○Congestion check time CPU can be stopped when processing becomes longer than set time. For this purpose, the maximum processing time of normal scan is to be set. In case setting is not made, the time period is automatically set to 100 msec. ○Operation mode at abnormality When this is set, operation continues despite a minor error in the CPU. Note: Avoid using this function for other than debugging.	Set operation parameter depending on user's application purpose.
5	I/O assignment	○Sets CPU-mounted I/O assignment information. ○Since the H-200 does not incorporate a function for copying the I/O assignment table, it must be set without fail referring to Table 6-3-1	I/O assignment cannot be omitted for programming.
6	Program name	○Sets program name in up to 16 alphanumeric characters. The set program name can be written in the CPU together with program. Hence, program check and management are facilitated.	Set program name for facilitating program check and management.
7	Power failure-protected memory	○Sets the range of data to be retained in a specific CPU area at CPU power OFF and RUN start. Setting is possible for R, WR, WM, TD, DIF and DFN.	Set this function when there is data which must be retained when stopping operation. Special internal output data will be retained or erased depending on its number in the event of power failure.

For details of initial setting, refer to the manual of each programming device or the following sections of the instruction manual.

Programming device	Reference
Portable graphic programmer (PGM-GPH)	Section 7 "INITIAL SETTING FUNCTION"
Instruction word programmer (PGM-CHH)	Section 7 "INITIAL SETTING FUNCTION (manual)"

8.4 Check of I/O Wiring

(1) Confirmation of external input device wiring
▽
(2) Confirmation of external output device wiring

(1) Confirmation of external input device wiring

Procedure Ascertain that the CPU is stopped.
▽
Turn on power supply to the external input device.
▽
While turning ON/OFF the external input manually, check the wiring of the external input device.

Check the wiring by confirming the ON/OFF status of external input/output devices by the method below.

Method of confirmation	Description
Using LED on I/O module	Confirm wiring by checking the lit status of LED on the I/O module, which corresponds to input.
Monitoring with programming device	Using the monitor function of programming device, specify input number to read out the input status and thereby confirm the wiring.

For details of monitor function, refer to the manual of each programming device or the sections below of the instruction manual.

Programming device	Reference
Portable graphic programmer (PGM-GPH)	Section 5 "MONITOR"
Instruction word programmer (PGM-CHH)	Section 5 "MONITOR (manual)"

(2) Confirmation of external output device wiring

Procedure Ascertain that the CPU is stopped.
▽
Turn on power supply to the external output device.
▽
Confirm interlock and safety around the external output device.
▽
Confirm wiring to the external output device.

Wiring of the external output device is checked in the following way. Using the forced output, each output of the I/O modules is turned ON/OFF independently and confirmation is to be made on whether or not the terminal output operates normally.
Note that the "forced set/reset" function must not be used for this output wiring check. Otherwise, data in the CPU might be issued to output and an unexpcted output might operate.

For details of forced output, refer to the manual of each programming device or the relevant section below of the instruction manual.

Programming device	Reference
Portable graphic programmer (PGM-GPH)	5.4 "Forced Output" of Section 5 "MONITOR"
Instruction word programmer (PGM-CHH)	5.4 "Forced Output (manual)" of Section 5 "MONITOR"

8.5 Programming

Program is directly written in the CPU by use of a portable graphic programmer or instruction word programmer.
For startup of programming device, refer to the section below of this manual.

Reference
Section 6 "SIMPLE EXAMPLE OF OPERATION"

For detailed explanation of language necessary for programming, refer to the section below of this manual.

Reference
Section 5 "PROGRAMMING"

For loading a program which has been saved in cassette tape, refer to the section below of the manual for each programming device.

<table>
<tr><td>Programming method</td><td colspan="3">Programming method and operating method
Refer to the document below.</td></tr>
<tr><td rowspan="3">Generation of new program with programming device</td><td>Programming device</td><td colspan="2">Operating method</td></tr>
<tr><td>Portable graphic programmer (PGM-GPH)</td><td colspan="2">Portable graphic programmer manual
Section 4 "Programming" and other</td></tr>
<tr><td>Instruction word programmer (PGM-CHH)</td><td colspan="2">Instruction word programmer manual
Section 4 "Programming" and other</td></tr>
<tr><td rowspan="3">Loading of program already saved in cassette into CPU</td><td>Recording medium</td><td>Progmamming device</td><td>Operating method</td></tr>
<tr><td rowspan="2">Audio cassette tape recorder (CMT)</td><td>Portable graphic programmer (PGM-GPH)</td><td>Portable graphic programmer manual
Section 6 "Cassette (CMT) I/F"</td></tr>
<tr><td>Instruction word programmer (PGM-CHH)</td><td>Instruction word programmer manual
Section 6 "CASSETTE (CMT) I/F"</td></tr>
<tr><td>Readout of program already recorded in ROM</td><td colspan="3">Refer to 8.9 "Writing a Program in ROM" of Section 8 in this manual.</td></tr>
</table>

8.6 Test Run

(1) Confirmation of interlock
▽
(2) No-Load operation
▽
(3) Operation under actual load

(1) **Confirmation of interlock**

Confirm that interlock function is activated securely should an emergency occur.

Circuits such as the emergency stop circuit, protection circuit and interlock circuit must be composed outside the programmable controller.

(2) **No-load operation**

Before actual energization of system load, turn off the load power supply and run only a program as a simulation in order to check the program operation by using the LED of I/O modules, the monitor function of programming device, etc.

This simulation must not be neglected if erroneous program operation might damage the opposite machine or entail any trouble.

The H-200 comprises a forced set/reset function which facilitates the above-mentioned program check.

Function name	Application	Contents	Programming device
Forced set/reset	○ For forced advance of control state ○ For forced change of internal/external output state	Forced setting of internal/external output	Protable graphic programmer Instruction word programmer

For details of forced set/reset function, refer to the manual of each programming device or the instruction manual.

Programming device	Reference
Portable graphic programmer (PGM-GPH)	5.3 "Forced Set/Reset (manual)" of Section 5
Instruction word programmer (PGM-CHH)	5.3 "Forced Set/Reset (manual)" of Section 5

Program operation is to be checked by the monitor function.
Monitor function is varied as follows.

No.	Function	Description	Relevant programming device
1	Circuit monitor	Searches for and displays circuit upon specification of circuit No., or I/O No. At the same time, the ON/OFF state of contact coil, etc. is displayed.	Portable graphic programmer Instruction word programmer
2	I/O monitor	Function independent of circuit monitor. Displays ON/OFF state of specified input/output (including internal output) and word contents together with circuit monitor in the message display area.	Portable graphic programmer Instruction word programmer
3	Instruction word monitor	Displays ON/OFF state of each circuit or instruction word in instruction word.	Portable graphic programmer Instruction word programmer

For details of the monitor function, refer to the manual of each programming device or the relevant section below of the instruction manual

Programming device	Reference
Portable graphic programmer (PGM-GPH)	Section 5 "MONITOR"
Instruction word programmer (PGM-CHH)	Section 5 "MONITOR"

(3) **Operation under actual load**

Operation is to be checked with power supply turned on for both external input and output.
Procedure is the same as in (2) "No-load operation" except for load activation. So refer to (2).

8.7 Program Change

(1) CHANGE function
(2) Change during RUN

(1) CHANGE function

As a result of test run, it may be judged that a program need be corrected. In this case, correct the program by using the CHANGE function of each programming device. For details of the CHANGE function, refer to the manual of each programming device or the relevant section below of the instruction manual.

Programming device	Reference
Portable graphic programmer (PGM-GPH)	4.2.2 "CHANGE" of Section 4
Instruction word programmer (PGM-CHH)	4.2.2 "CHANGE" of Section 4

(2) Change during RUN

With each programming device, a part of circuit or the set value of timer/counter can be changed during CPU run without turning off the output. For details of change during RUN, refer to the manual of each programming device or the relevant section below of the instruction manual. (This function is restricted depending on programming device.)

Programming device	Reference	Remarks
Portable graphic programmer (GPM-GPH)	5.5 "Change during RUN" of Section 5	Standard: Only the set value of timer/counter is changeable. (Note 2)
Instruction word programmer (PGM-CHH)	5.5 "Change during RUN" of Section 5	Only the set value of timer/counter is changeable.

Notes:

1. Program is changeable during CPU run only when the special internal output (R7C7) is turned ON to allow this.
2. In case of the portable graphic programmer, the function for change during RUN can be expanded by using the ROM pack (PGMPK2H) (to be purchased separately).

Listed below are functions which can be expanded.

Function		Menu display	Functional outline
Circuit generation		WRITE NEXT CHANGE WRITE FIRST (Any of the above)	Generates program of a single circuit
Circuit update	Addition/insertion	WRITE NEXT	Inserts a single circuit next to the circuit No. where cursor is set in circuit readout.
	Change	CHANGE	Changes the circuit at which cursor is located in circuit readout.
	Deletion	DELETE	Deletes the circuit with which cursor is matched in circuit readout.
	Insertion of circuit at head	WRITE FIRST	Inserts a single circuit befere current circuit No. 1.

Remember that a circut which includes any of the control instructions below cannot be changed while the CPU is running.

No.	Instruction code	Instruction name	No.	Instruction code	Instruction name
1	M C S	Master control set	6	C A L	Subroutine call
2	M C R	Master control reset	7	S B	Subroutine
3	J M P	Unconditional jump	8	R T S	Subroutine end
4	C J M P	Conditional jump	9	I N T	Start of interrupt scan
5	L B L	Label	10	R T I	End of interrupt scan
			11	E N D	End of normal scan
			12	C E N D	Conditional end of scan

For details, refer to the instruction manual of the ROM pack for portable graphic programmer.

8.8 Saving of Program

(1) Audio cassette (CMT)
(2) Printout

A program could be destroyed through misoperation.
Although this rarely occurs, it is recommended for minimizing the influence of such an event to save, verify and store the newest program periodically.

(1) **Audio cassette**

Program and various data can be saved, loaded and verified by using a commercially available audio cassette recorder (CMT).

Function	Description
Save	Copies memory contents into CMT from CPU.
Load	Copies contents of tape into CPU from CMT.
Verify	Verifies that memory contents of CPU are coincident with contents of CMT.

Connection of each programming device and audio cassette tape recorder is shown below.

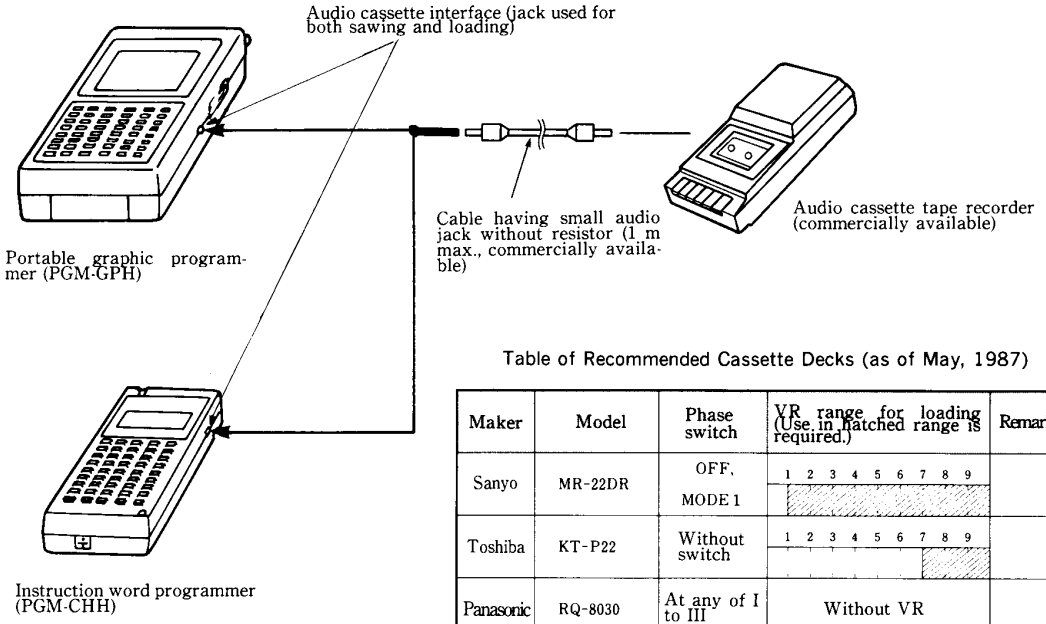


Table of Recommended Cassette Decks (as of May, 1987)

Maker	Model	Phase switch	VR range for loading (Use in hatched range is required.)	Remarks
Sanyo	MR-22DR	OFF, MODE 1	1 2 3 4 5 6 7 8 9	
Toshiba	KT-P22	Without switch	1 2 3 4 5 6 7 8 9	
Panasonic	RQ-8030	At any of I to III	Without VR	
Hitachi	TRQ-1500	OFF	1 2 3 4 5 6 7 8 9	

8K steps necessitate about 5 minutes.

For details of audio cassette, refer to the relevant section of each programming device manual.

Programming device	Reference
Portable graphic programmer (PGM-PGH)	6.6 "CMT" of Section 6
Instruction word programmer (PGM-CHH)	6.6 "CMT" of Section 6

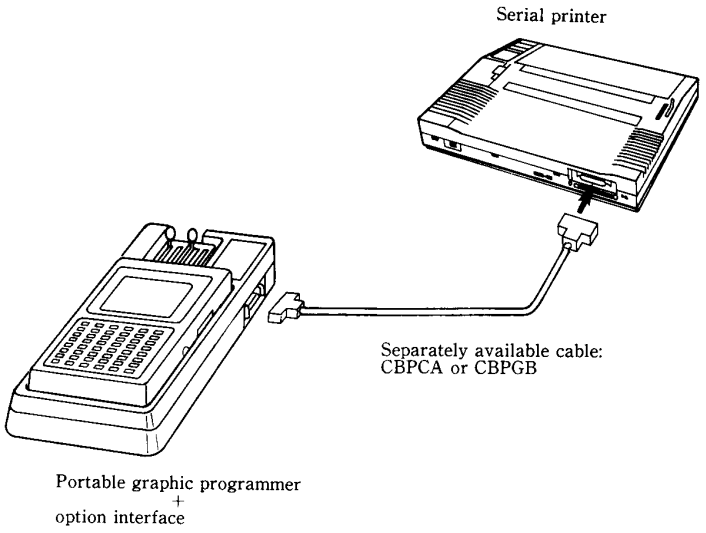
(2) **Printout**

Program and various data can be printed out by combining portable graphic programmer and the option interface.

Printout functions are listed in the table below.

Kind of printout	Selectable item	
1. Program	1. Ladder 2. Instruction word	1. Full printout
		2. Partial printout
2. Cross reference	1. Partial printout	
3. Parameter		
4. Internal output data		

○ The portable graphic programmer requires the option interface (PGMIF1H).



For details of printout, refer to the manual of each programming device or the relevant section below of the instruction manual.

Programming device	Reference
Portable graphic programmer (PGM-GPH)	2.2.4 "Connection of Printer" of Section 2
	6.8.1 "Printer" of Section 6

(Reference data 1)

When using a serial printer, reference must be made to the data below.

Table of Interface Specifications

Item No.	Item	Specification	Remarks
1	Size of ladder diagram	(9 contacts + 1 coil) × 7 lines	
2	Interface	RS−232C	Cat. No. 8145 or 8148 interface PC board (made by EPSON) is necessary.
3	Printer used	EPSON MP−80 TYPE II & III RP−80 FP−80 FP-850 ※SP−80T ※The optional tractor unit (#8304V) is required.	Use the following printer cable. Type: CBPGA (option) (for Cat. No. 8145) Type: CBPGB (option) (for Cat. No. 8145 and 8148)
4	Bit rate	4800〔BPS〕	Set with the DIP switch in the printer. (See table below.)
5	Transmission code	1 start bit, 8 data bits, 1 stop bit	

DIP switch of printer mainframe and interface PC board

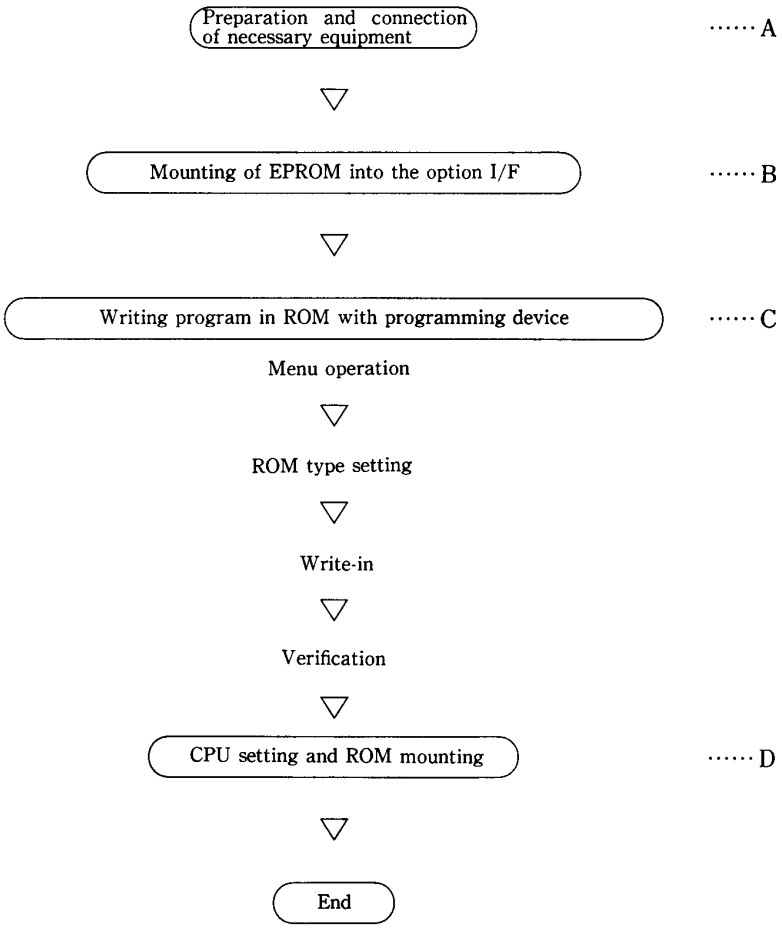
Item	DIP switch No.		MP−80 TYPEII	MP−80 TYPEIII	PR−80	FP−80	FP−850	SP−80T
Setting of DIP swich in printer mainframe	SW1	1	OFF	OFF	OFF	OFF	OFF	OFF
		2	OFF	OFF	OFF	OFF	OFF	OFF
		3	OFF	OFF	ON	OFF	OFF	OFF
		4	OFF	OFF	OFF	OFF	OFF	OFF
		5	OFF	OFF	OFF	OFF	OFF	OFF
		6	OFF	ON	ON	OFF	OFF	OFF
		7	OFF	OFF	OFF	OFF	OFF	OFF
		8	ON	ON	OFF	OFF	OFF	OFF
	SW2	1	ON	OFF	OFF	ON	OFF	OFF
		2	ON	OFF	ON	ON	OFF	ON
		3	OFF	OFF	OFF	OFF	OFF	OFF
		4	ON	OFF	OFF	OFF	OFF	OFF

Applicable interface PC board			Cat.No.8145	Cat.No.8148
Setting of DIP switch on interface PC board	SW 1	1	OFF	OFF
		2	ON	OFF
		3	OFF	OFF
		4	OFF	OFF
		5	OFF	OFF
		6	OFF	ON
		7	ON	OFF
		8	Unused (setting unnecessary)	ON
	SW 2	1	OFF	ON
		2	ON	ON
		3	OFF	ON
		4	ON	ON
		5	Not provided	OFF
		6	Not provided	OFF

8.9 Writing a Program in ROM

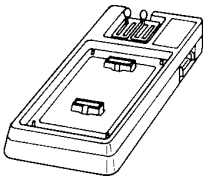
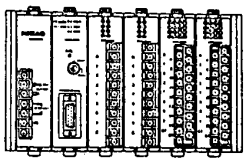
(1) **In case of EPROM**

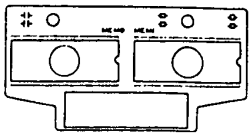
A program stored in the RAM built in the H-200 CPU can be written in a ROM by the method below.



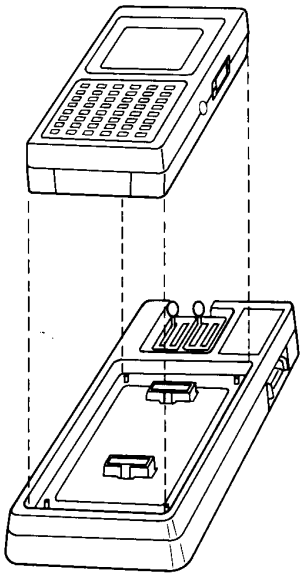
(A) **Preparation and connection of necessary equipment**

Equipments required for writing a program in ROM are shown below.

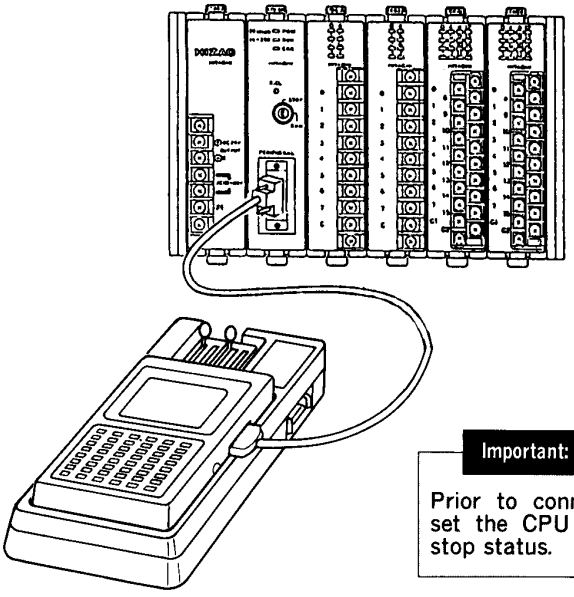
Portable graphic programmer	Option I/F for portable graphic programmer	H-200 basic unit
		
PGM-GPH	PGMIF1H	

Memory pack	PGM-GPH and CPU connecting cable
	
MPH-8R	PGCB-02H

① Install the option I/F in the portable graphic programmer (PGM-GPH).



② Connect the portable graphic programmer (PGM-GPH) and CPU module with cable.



Important:

Prior to connection, set the CPU in the stop status.

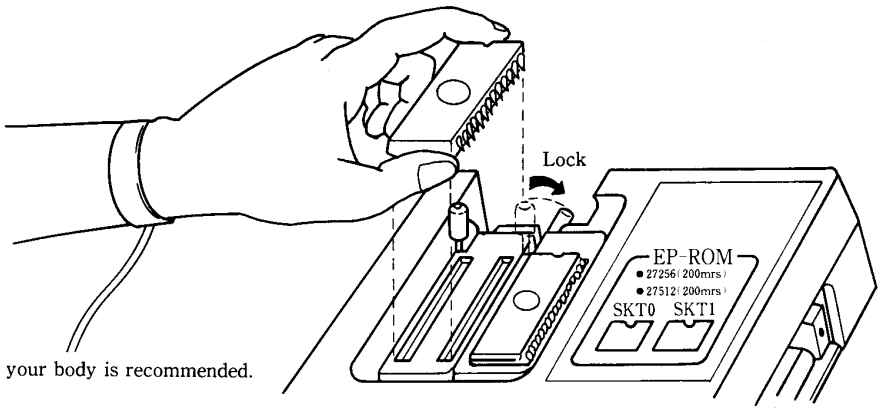
For details of the option I/F connection, refer to the manual of portable graphic programmer or the section specified below of instruction manual.

Programming device	Reference
Portable graphic programmer (PGM-GPH)	2.4.3 "Option I/F Connection" of Section 2

(B) **Mounting of EPROM into the option I/F**

Into the option I/F, mount the EPROM (ROMIC-01H) furnished with the EPROM memory pack (MPH-8R) as an accessory.

After confirming the correct orientation of pin ①, insert the EPROM into the IC socket (SKT0/SKT1) and lock the lever.



Grounding your body is recommended.

Caution:

1. Avoid plugging in or unplugging the EPROM while the ROM writer function is activated. Otherwise, the EPROM might be damaged.
2. EPROM is generally vulnerable to static electricity. Therefere, before handling EPROM, it is recommended to touch some grounding part in order to dissipate the staitic electricity charged in your body.
3. Do not touch the EPROM leads (terminal) directly with your fingers.
4. Use an EPROM whose contents have completely been erased for writing a new program. Besides, attach a light shielding label after writing.
5. In case EPROM is not intended to be used for a long time, it must be accommodated in a protective case and wrapped with aluminum foil or the like for preventing damage due to static electricity, etc.

(C) **Writing a program in ROM with portable graphic programmer**

Menu operation
Select the option box function.

▽

ROM type setting
Set ROM type 27256.

▽

Write-in
Write the contents of RAM in ROM.

▽

Verification
Verify that program is copied correctly from the RAM to ROM.

Caution:

Suppose that a program has been input by the programming device with the memory type RAM-04H specified and you attempt to write the program in ROM. In this case, error "31" will occur. To prevent this, program must be input with the memory type RAM-08H specified.

For details, refer to the manual of portable graphic programmer or the section specified below of instruction manual.

Programming device	Reference
Portable graphic programmer (PGM-GPH)	6.8.2 "ROM Writer" of Section 6

8

(D) **ROM mounting**

① Mount the EPROM written in SKTO into the IC socket (SKT0) in the memory pack, or the one written in SKT1 into the IC socket (SKT1).

Caution:

- Before mounting a ROM, touch some grounding means in order to dissipate static electricity charged in your body. Otherwise, the ROM might be damaged by the statc electricity. It is recommended to keep your body grounded during this work.

Caution:

- When mounting a ROM, carefully confirm the following.
  (1) The ROM is oriented properly (by matching the direction of the notch).
  (2) The leads of ROM are not bent.

② Install the memory pack in the CPU module.

Caution:

- When turning on power supply with the memory pack installed, the existing program in the CPU module will be lost. So confirm this in advance.
- The memory pack must be inserted properly as shown below so it won't deviate.

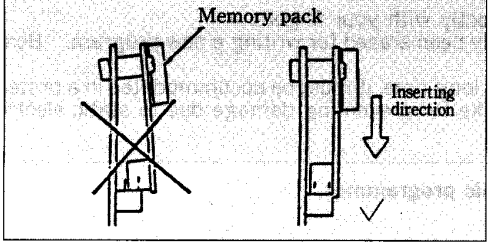
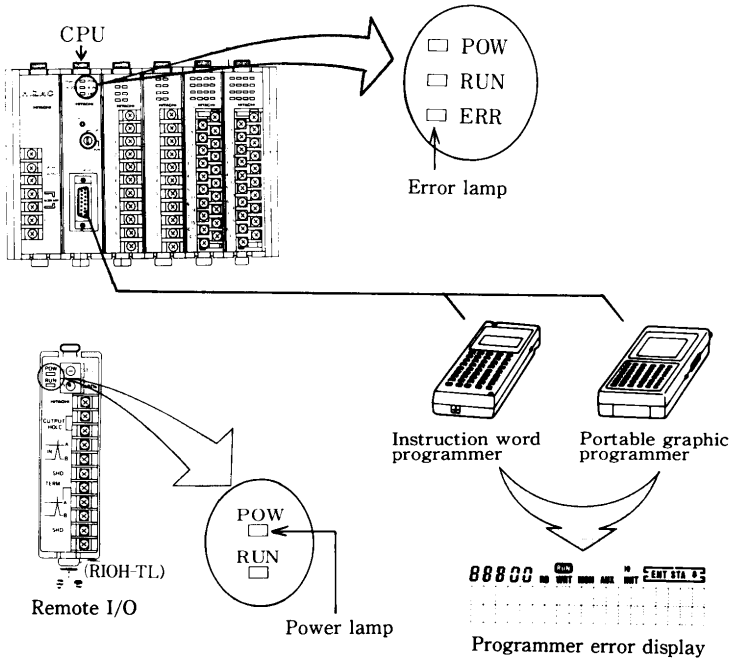


Fig. 8-5 ROM Mounting

9. ERROR DISPLAY AND COUNTERMEASURE, AND MAINTENANCE

9.1 Error Display and Countermeasure

Fig. 9-1-1 shows the location for displaying errors to be detected with each device of the H-200 system. Most errors are displayed in error codes. So take a countermeasure according to, the error code list.



Point

Even the same error code has different meanings depending on its display location.

Fig. 9-1-1 Error Display Location of H-200 System

(1) **Error display of CPU module and how to deal with error**

Error lamp display of the CPU module depends on the result of self-diagnosis by the microcomputer. The contents of display are as outlined below. For details of error code and countermeasure, refer to the self-diagnosis error code list and Table 9-2-1.

Error lamp
- Flickering: Battery error
  - Battery is not provided. Or battery requires renewal.
- Lit: After turning the key switch from RUN to STOP
  - Error lamp remains lit.
    - If the same error recurs even after turning off power supply and then turning it on again, the CPU module must be changed to a new one.
  - Error lamp goes off.
    - Localize and eliminate the error according to the self-diagnosis error code (WRF000), syntax/assemble error details (WRF001) and computation error code (WRF015). Then turn the key switch to the RUN position.

(2) **Programmer error display**

While, a programming device is operating, a double definition error, non-definition error, operation error, program-over or other error code may be displayed.

For details of error code, refer to the error code list of each programming device manual.

(3) **Error display of remote I/O (slave station)**

If the power lamp is turned off, make sure that power source is connected properly to the power module.

(4) **How to clear CPU error lamp display**

Once the error lamp of CPU module has lit, it remains lit even if the error factor is eliminated. The error lamp can be turned off by the methods below.

How to clear CPU error display

Possible in 4 ways below.

① 

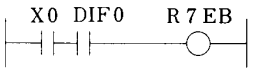
Change the position of the switch
(During operation, or in RUN status, operation need be stopped.)

② Turn on power supply again (restart by resetting).

③ With a programming device connected, set the special internal output R7EB to 1.

④ Write program preliminarily so that R7EB is set by external input.

[Example]

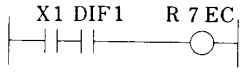


Error code is set in the special internal output area (such as WRF000). A smaller error code value corresponds to a higher error level or more serious fault.

If two or more errors concur, the smallest error code value is displayed in WRF000. For instance, if [71] (battery error) and [31] (user memory abnormal) occur concurrently, [31] is displayed. At the same error level, priority is given to the latest error.

Error special internal output is cleared by setting R7EC to 1. R7EC is settable to 1 with a programming device connected or according to external input in a program written preliminarily.

[Example]



Shown below is the range of special internal output to be cleared when setting R7EC to 1.

Bit special internal output

	Number
Major fault error	R 7 C 8
(Undefined)	9
Memory abnormal (user, PI/O)	A
(Undefined)	B
Memory over-size (user)	C
(Undefined)	D
	E
	R 7 C F
	R 7 D 0
Congestion error (normal)	1
Congestion error (periodic)	2
Congestion error (interrupt)	3
Syntax/assemble error	4
(Undefined)	5
	6
	7
	8
Battery error	9
(Undefined)	A
Self-diagnosis error	B
(Undefined)	C
	D

Word special internal output

	Number
Self-diagnosis error code	WRF000
Syntax/assemble error details	1
	2
	3
(Undefined)	4
	5
	6
	7
Error circuit No.	WRF008

Fig. 9-1-3 Area of Special Internal Output to Be Cleared by Special Internal Output R7EC

In case special internal output data cannot be cleared at all for reasons perteining to program execution, only the corresponding error flag must be cleared referring to Table 9-2-1 "Self-Diagnosis Error Code List".

Caution:

If the internal output R7DB (WRF000) of self-diagnosis error is used for CPU run stop condition as a system abnormality, R7DB might turn on (WRF000 set to H71) for even an error as slight as a warning (battery error [71]) to stop the CPU. Therefore, do not use the internal output of self-diagnosis error for the CPU run stop condition.

9.2 Self-Diagnosis Error Codes

Listed below are self-diagnosis error codes and measures to be taken. Each error code is output to WRF000 of special internal output in hex adecimal.

Table 9-2-1 Self-Diagnosis Error Code List (1/3)

Error code	Error name Check timing indicated in brackets [ ]	Classification	Contents and cause Cause indicated in parens ( )	Measure, etc.	Error lamp	Operation	Special internal output to be set	
							Bit*1	Word*2
11	System ROM error 1 [At power ON]	Major fault	Abnormality was detected in sum check of the system ROM by the microcomputer. (System ROM could not be read correctly.)	The hardware of CPU module is abnormal. In case the same error recurs after turning on power supply again, the CPU module need be renewed. [Other cause and check point] ○Check if there is a large noise source or interfering factor near the H-200 system.	Lit	Stopped		
12	System RAM error 1 [At power ON]	Major fault	Abnormality was detected in read/write check of the system RAM by the microcomputer. (System RAM could not be read/written correctly.)		Lit	Stopped		
13	Microcomputer error [Always checked]	Major fault	Abnormality was detected because the microcomputer attempted to execute an undefined instruction during execution of the system program. (System ROM could not be read correctly.)		Lit	Stopped	R 7 C 8	
23	Undefined instruction [Always checked]	Medium fault	Abnormality was detected because the microcomputer attempted to execute an undefined instruction which could not be read by the processor for user program execution. (User program or system ROM could not be read correctly.)		Lit	Stopped	R 7 C 9	
27	Data memory abnormal (At power ON When specifying all-clear operation from programming device)	Medium fault	Abnormality was detected in read/write check of data memory. (Data memory could not be read/written correctly.)		Lit	Stopped		
31	User memory abnormal (At power ON At changeover from STOP to RUN During RUN (only RAM checked) At change of parameter At all clear operation)	Medium fault	○When [RAM] is set: Abnormality was detected in sum check of the user memory. (The contents of user memory have not been correct.) ○When [ROM] is set: Abnormality was detected in sum check of the user memory in ROM. (The contents of ROM memory have not been correct.)	○ The contents of user program are destroyed. Perform initialization and transfer program again. ○This error is indicated if the battery is completely discharged or the battery is removed for a long time. ○When the ROM is used, this error may be indicated in the cases below. i) ROM chip is mounted reversely. ii) ROM memory pack is not plugged in correctly. iii) Program is not written correctly in ROM.	Lit	Stopped	R 7 CA	

*1 R7DB: Set when error is detected in CPU.

*2 WRF000: Stores in binary format an error code detected in CPU

Table 9-2-1 Self-Diagnosis Error Code List (2/3)

Error code	Error name Check timing indicated in brackets [ ]	Classification	Contents and cause Cause indicated in parens ( )	Measure, etc.	Error lamp	Operation	Special internal output to be set	
							Bit	Word
33	User memory size error [At changeover from STOP to RUN]	Medium fault	The capacity of user program set by parameter is larger than the actual capacity.	This error might be displayed when the contents of memory in the CPU module are not correct. If the same error recurs after initialization, the CPU module must be changed to a new one.	Lit	Stopped	R 7 CC	
34 *1	Syntax/assemble error [At changeover from STOP to RUN]	Medium fault	User program (ladder) contains a syntax error. Detailed contents of error are set in the special internal output WRF001.	User program contains syntex error or assemble error. Reconfirm the program and assignment table.	Lit	Stopped	R 7 D 4	WRF001
44	Congestion error (normal scan) [At execution of END processing]	Minor fault	Execution time (scan time) of normal scan was checked for each END instruction. In consequence, the parameter-set congestion check time (standard 100 msec) was exceeded. So it was detected as an abnormality.	Rewrite user program so as to shorten its scan time or change the congestion check time (See Note). Note: Operation parameter setting of CPU is changeable with programming device.	Lit	(Stopped) *2	R 7 D 1	
45	Congestion error (periodic scan) [Checked in periodic processing]	Minor fault	Execution time of periodic scan program was checked in periodic processing. As a result, the determined time was exceeded. So it was detected as an abnormality.	Rewrite periodic interruption program so as to shorten its execution time.	Lit	(Stopped) *2	R 7 D 2	
46	Congestion error (interrupt scan) [Checked at occurrence of interruption]	Minor fault	During execution of interrupt scan program, interruption of the same factor recurred. So it was detected as an abnormality.	Extend the time interval between interruption inputs.	Lit	(Stopped) *2	R 7 D 3	
61	Port transmission error*3 (parity) [At transmission]	Warning	In communication with programming device or personal computer, a parity error of received data was detected. Cable was disconnected in the online mode. { Noise is superimposed on cable. Connector cable is not connected properly. Mismatch in transmission speed, etc. }	○ Plug in the cable connector again. ○ Reconfirm wiring and signal allocation inside the connector. ○ Review setting of transmission speed, etc. ○ Eliminate noise source from the vicinity of cable. ○ Other proper measure	Turned off	Running		
62	Port transmission error*3 (framing/overrun) [At transmission]	Warning	In communication with programming device or personal computer, framing error or overrun error was detected. { Noise is superimposed on cable. Connector cable is not connected properly. Mismatch in transmission speed, etc. }		Turned off	Running		

*1 Includes the case where a program which contains any instruction unavailable with the H-200 or any I/O outside the usable range, is used.

*2 Despite this error, operation can be continued by initial setting of parameters with a programming device.

*3 Errors of codes 61 to 65 may also occur the moment the cable is disconnected or reconnected in the live status. This is due to communication in the unstable condition peculiar to disconnection/reconnection time. Since recovery will automatically be made soon, CPU operation remains unaffected. For clearing the error code in CPU module, set the special internal output R7EB to 1.

Table 9-2-1 Self-Diagnosis Error Code List (3/3)

Error code	Error name Check timing indicated in brackets [ ]	Classification	Contents and cause Cause indicated in parens ( )	Measure, etc.	Error lamp	Operation	Special internal output to be set	
							Bit	Word
63	Port transmission error* (time-out) [At transmission]	Warning	In communication with programming device or personal computer, time-out error was detected. (Noise is superimposed on cable. Connector cable is not connected properly. Processing program on the personal computer side does not agree with CPU specifications.)	○ Plug in the cable connector again. ○ Eliminate noise source from the vicinity of cable. ○ Review communication program with H-200. ○ Check the contents under actual transmission with the aid of a line monitor unit or the like. ○ Other proper measure	Turned off	Running		
64	Port transmission error* (protocol error) [At transmission]	Warning	In communication with programming device or personal computer, protocol (transmission procedure) error was detected. (Noise is superimposed on cable. Connector cable is not connected properly. Processing program on the personal computer side does not comply with protocol.)		Turned off	Running		
65	Port transmission error* (BCC error) [At transmission]	Warning	In communication with programming device or personal computer, data could not be received correctly. (Noise is superimposed on cable. Connector cable is not connected properly. Processing program on the personal computer side does not comply with protocol.)		Turned off	Running		
71	Battery error [Always]	Warning	Battery charge has become lower than the specified level. Or the battery is not mounted.	Renew the battery. (Refer to (5) of 9.5.)	Flickering	Running	R 7 D 9	

*Errors of codes 61 to 65 may also occur the moment the cable is disconnected or reconnected in the live status. This is due to communication in the unstable condition peculiar to disconnection/reconnection time. Since recovery will automatically be made soon, CPU operation remains unaffected. For clearing the error code in CPU module, set the special internal output R7EB to 1.

9.3 Details of Syntax/Assemble Errors

(1) The table below details the assemble error codes and circuit state. These error codes are set in the internal output WRF001.

Table 9-3-1 Details of Syntax/Assemble Errors (1/2)

Code	Error	Description	Corrective measure
0 1	Double LBL definition	The LBL instruction of the same number was specified more than once.	Limit the LBL instruction of the same number to only one.
0 4	Double SB definition	The SB instruction of the same number was specified more than once.	Limit the SB instruction of the same number to only one.
0 5	Double INT definition	The INT instruction of the same number was specified more than once.	Limit the INT instruction of the same number to only one.
0 F	Undefined instruction	An instruction unusable for the H-200 was specified, or an invalid instruction was found.	Delete undefined instruction.
1 0	END undefined	The INT or SB instruction is not preceded by the END instruction.	Define the END instruction before the INT or SB instruction.
1 1	RTS undefined	The RTS instruction, associated with the SB instruction, was omitted.	Define the RTS instruction after the SB instruction.
1 2	RTI undefined	The RTI instruction, associated with the INT instruction, was omitted.	Define the RTI instruction after the INT instruction.
1 3	SB undefined	The SB instruction, associated with the RTS instruction, was omitted.	Define the SB instruction before the RTS instruction.
1 4	INT undefined	The INT instruction, associated with the RTI instruction, was omitted.	Define the INT instruction before the RTI instruction.
1 6	I/O No. error.	There is a circuit containing an I/O number outside the I/O range of H-200.	Correct I/O number within the effective range or delete it.
2 0	RTS area error	The RTS instruction was defined in the normal scan area or interrupt scan area.	Define the RTS instruction in the subroutine area.
2 1	RTI area error	The RTI instruction was defined in the normal scan area or subsoutine area.	Define the RTI instruction in the interrupt scan area.

Table 9-3-1 Details of Syntax/Assemble Errors (2/2)

Code	Error	Description	Corrective measure
2 2	END area error	The END instruction was defined in the interrupt scan area or subroutine area.	Define the END instruction at the end of normal scan area.
2 3	CEND area error	The CEND instruction was defined in the interrupt scan area or subroutine area.	Define the CEND instruction in the normal scan area.
3 0	RTS startup condition error	The startup conditions were defined in the processing box containing the RTS instruction.	Delete the startup conditions from the processing box.
3 1	RTI startup condition error	The startup conditions were defined in the processing box containing the RTI instruction.	Delete the startup conditions from the processing box.
3 2	END startup condition error	The startup conditions were defined in the processing box containing the END instruction.	Delete the startup conditions from the processing box.

(2) The table below gives a typical program status at each error, which may be helpful for correcting your program if a syntax or assemble error occurs.

Table 9-3-2 Examples of Syntax and Assemble Error Programs (1/5)

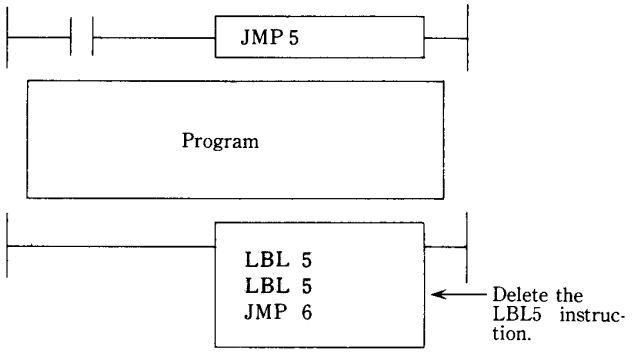
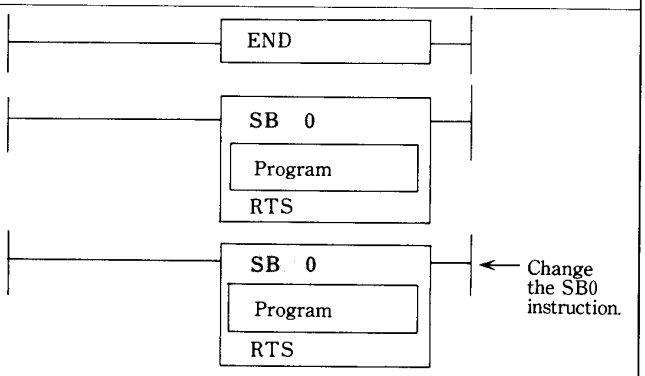
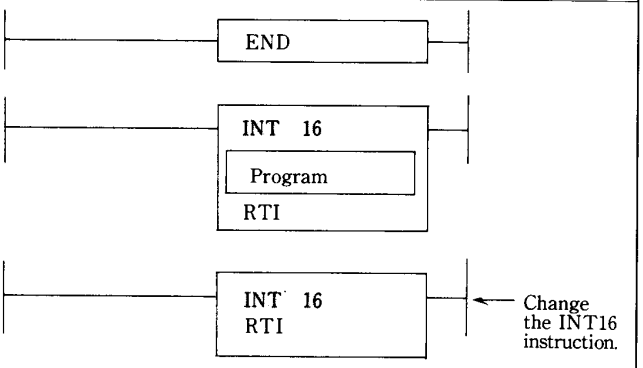
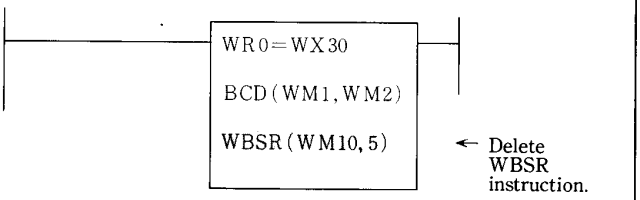
Code	Error	Example of program status
0 1	Double LBL definition	
0 4	Double SB definition	
0 5	Double INT definition	
0 F	Undefined instruction	

Table 9-3-2 Examples of Syntax and Assemble Error Programs (2/5)

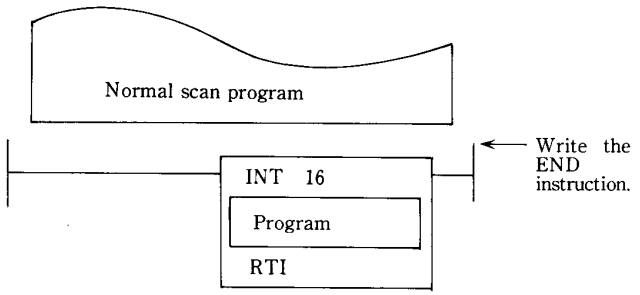
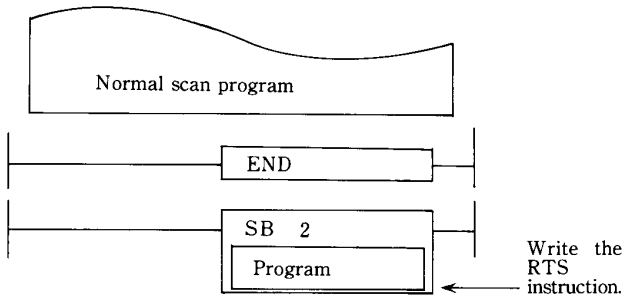
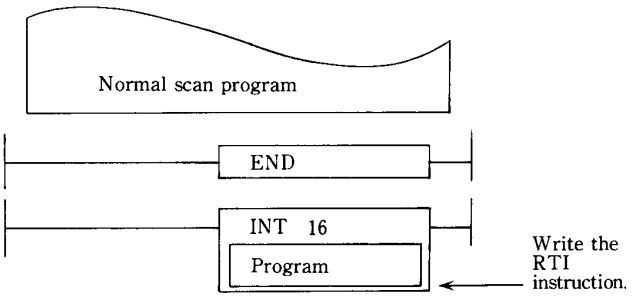
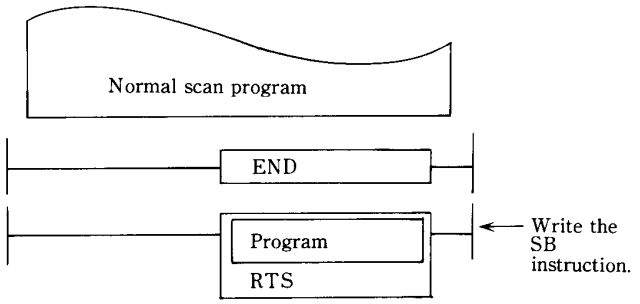
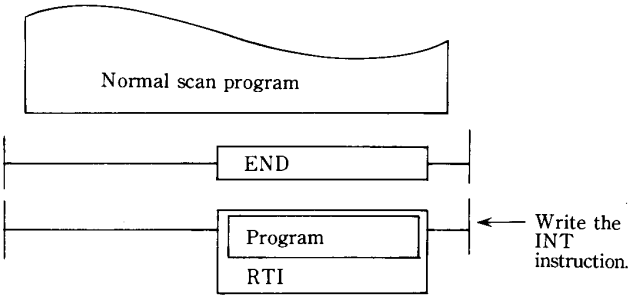
Code	Error	Example of program status
1 0	END undefined	
1 1	RTS undefined	
1 2	RTI undefined	
1 3	SB undefined	
1 4	INT undefined	

Table 9-3-2 Examples of Syntax and Assemble Error Programs (3/5)

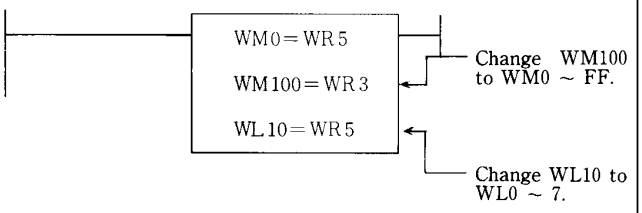
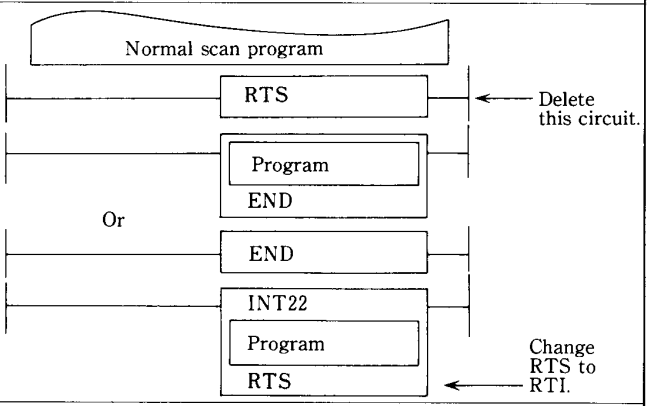
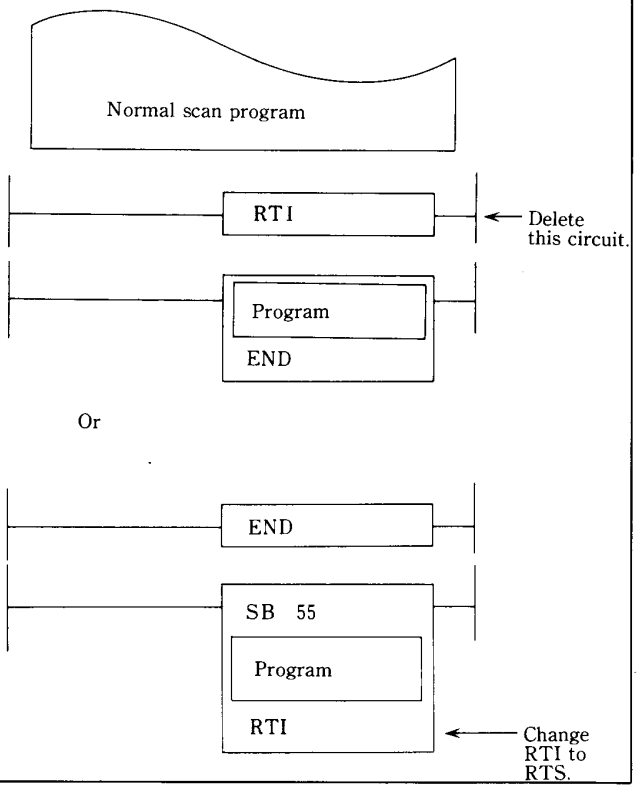
Code	Error	Example of program status
1 6	I/O No. error	
2 0	RTS area error	
2 1	RTI area error	

Table 9-3-2 Examples of Syntax and Assemble Error Programs (4/5)

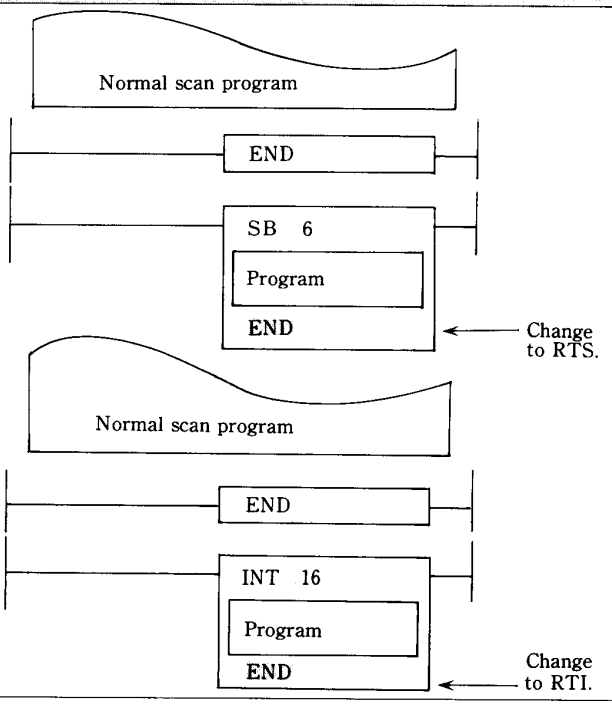
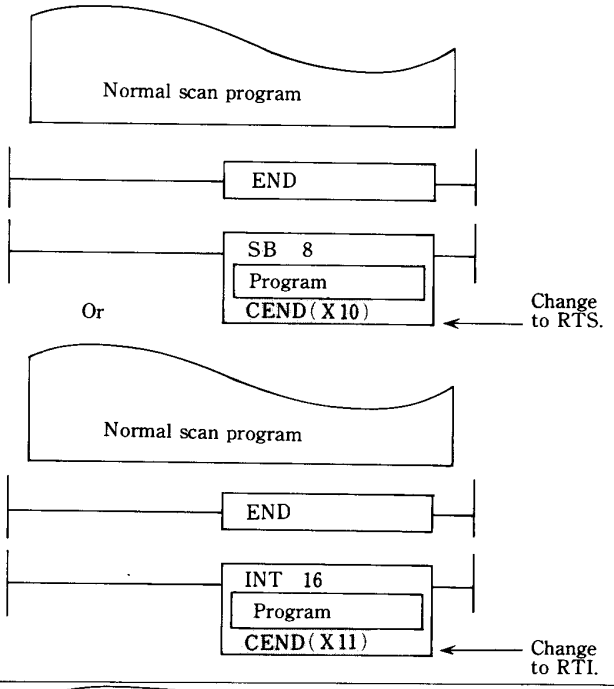
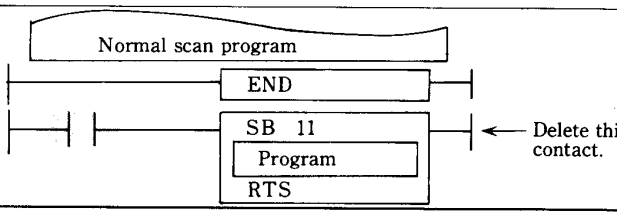
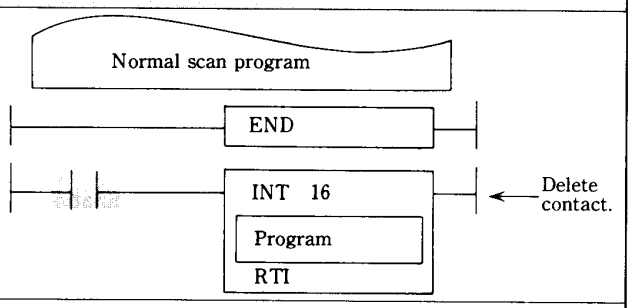
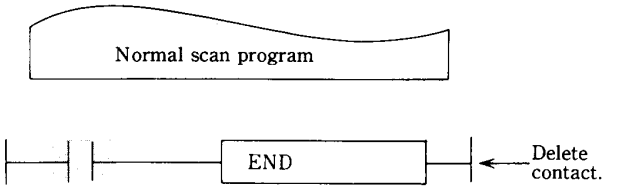
Code	Error	Example of program status
2 2	END area error	
2 3	CEND area error	
3 0	RTS startup condition error	

Table 9-3-2 Examples of Syntax and Assemble Error Programs (5/5)

Code	Error	Example of program status
3 1	RTI startup condition error	
3 2	END startup condition error	

9.4 Computation Error Codes

(1) If an error occurs during execution of control and special instructions, value 1 is set in ERR (R7F3) and an error code indicating the contents of that error is set in WRF015.

(2) Clear the contents of ERR to zero by using the program (R7F3 = 0), and clear the error code to zero by using the program (WRF015 = 0).

Table 9-4-1 Computation Error Codes (WRF015)

Error code	Error	Explanation	Error source
H0013	SB undefined	The SB n instruction of code number "n" was omitted for the CAL n instruction.	CAL
H0015	LBL undefined	The LBN n instruction of code number "n" was omitted.	JMP CJMP
H0040	LBL area error	The LBL instruction having the same code number as of the JMP (CJMP n) is not programmed in the same area.	JMP CJMP
H0041	CALL nesting level overflow	Subroutine was nested in 2 or more levels.	CAL
H0042	CAL undefined	Execution of the RTS instruction was attempted although the CAL instruction was not executed yet.	RTS

9.5 Maintenance

To use the H-200 functions under the best condition and keep system operations normal, routine and periodic checks are required. Check items are listed below.

(1) **Routine check**

In the operating status, carry out check according to the table below.

Table 9-5-1 Routine Check Items

Item	LED indication	Check method	Normal status	Possible cause
Check of CPU module indication	POW	Visual check	Lit	Abnormal power supply, etc.
	RUN	Visual check	Lit (during operation)	Congestion error, syntax error, microcomputer overrun, abnormal power supply, etc.
	ERR	Visual check	Turned off	Same as above (if lit)
				Battery error (if flickering)*
Check of remote I/O error indication	POW	Visual check	Lit	Abnormal power supply, etc.

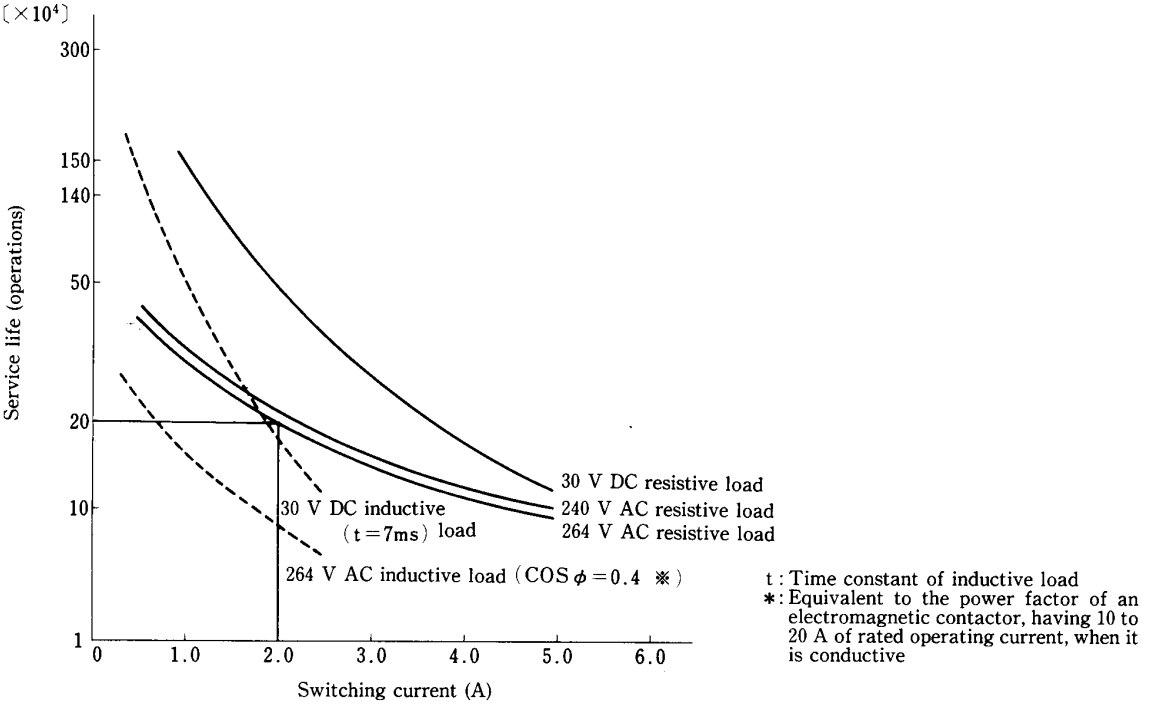
(Note*) The contents of memory might be destroyed if the old battery remains unchanged with the basic base power supply turned off beyond 1 week after the ERR lamp flickers. If the power supply is turned off for a long time, the contents of memory might have already been destroyed since this battery error cannot be detected.

(2) **Periodic check**

Turn off the power supply of external I/O circuits and check the items below half-yearly.

Table 9-5-2 Periodic Check Items

Location	Check item	Judgement standard	Remarks
Between programming device and CPU	Operation of programming device	All key switches and indicators must operate normaloly.	
Power supply	Voltage variation	85~132V/170~264V AC	Circuit tester
		19.2V~30V DC	
I/O modules	Service life of output relay	Electrical: 200 thousand operations Mechanical: 10 million operations	See Fig. 9-5-1.
	LED	Must be switched normally.	
	External power voltage	Must meet the specifications of each module.	See the specifications of each I/O module.
Battery	Check voltage and service life.	The battery must be renewed if the ERR lamp fiflickers or when the battery has been used for 3 years.	
Mounting and connection	(1) Fixing of each module (2) Fitting of each connector (3) Tightening of each screw (4) Connection of each cable	Must be free from abnormality.	Retighten the fastening screws. Plug in securely. Retighten loose screw. Visual check
Ambient conditions	(1) Temperature (2) Humidity (3) Others	0 to 55℃ 20 to 90% RH (non-condensing) Dust, foreign matter and vibration unallowable	Visual inspection
Spares	Quantity and storage of each spare	Must be free from abnormality.	Visual inspection
Program	Check of program contents	Make sure the final program contents (saved on ROM or cassette tape) match the CPU contents.	Check both master and backup media.



t : Time constant of inductive load
※: Equivalent to the power factor of an electromagnetic contactor, having 10 to 20 A of rated operating current, when it is conductive

Fig. 9-5-1 Life Curve of I/O Module Output Relay

(3) **Service Life of Each Power Module**

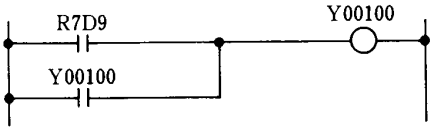
Each power module uses a number of electrolytic capacitors, which have a certim service life. It is said that the service life is reduced to half when the ambient temperature rises 10°C.

Prepare a spare power module taking into account that it has a standard lifetime of about 3 years at the rated ambient temperature (30°C). For longer life, the power module must be mounted in consideration of adequate ventilation and proper ambient temperature.

(4) **On Service Life of Battery**

Lifetime of battery	Guarantee 3 years
	Potential 10 years

- The lifetime of battery indicates the total OFF time of the power module of basic base.
- The battery has reached the end of its service life when the ERR lamp flickers.
- Battery life can also be read via the special internal output "R7D9."
  Shown below is an example of circuit which uses "R7D9."
  In the example, battery error can be output in the external output Y00100.



- "R7D9" functions as a flag which indicates battery error. This flag turns on when the battery has reached the end of its lifetime.
  The self-diagnosis error code "71" indicates that the battery is not mounted or it has reached the end of its service life.
- Be sure to renew the battery every 3 years even if it has not yet reached the end of its life.
- Even when the ROM memory is used, the battery is necessary for backing up the power failure-protected internal output data and maintaining the calendar clock function.

(5) **Procedure for renewing battery**

① Prepare a new lithium battery (LIBAT-H).

② Make sure that the newest program is saved on a cassette tape, etc. Otherwise, save the program on such a medium for the purpose of security.

③ Turn on power supply of the basic base to charge the capacitor in the CPU module.

④ Preparations have now been completed. The process up to plugging in the connector of lithium battery in ⑧ after turning off power supply to the basic base must be finished quickly, or within 1 minute.

Complete this process within 1 minute.

⑤ Turn off power supply of the basic base.

⑥ Remove the CPU module from the basic base.

⑦ Remove the old lithium battery from the battery case and unplug the connector.

⑧ Plug the connector (b) on the battery side into the connector (c) on the PC board side according to Fig. 9-5-3.
(Match polarities with the aid of the red/black indications.)

⑨ Mount a new lithium battery in the battery case, referring to Fig. 9-5-2. The lead wire must be housed in the battery case as per Fig. 9-5-4.

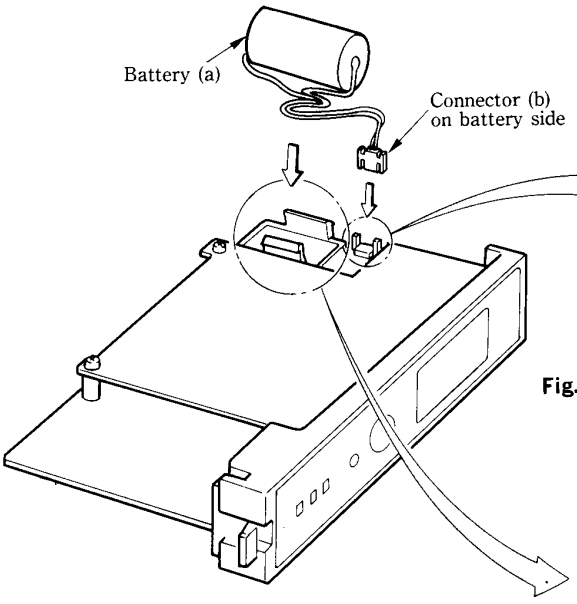


Fig. 9-5-2 Mounting of Lithium Battery

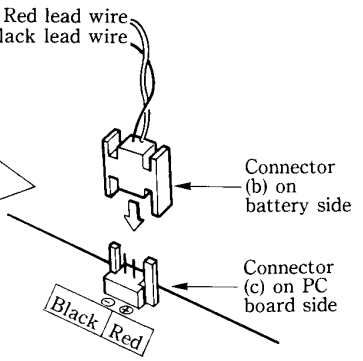
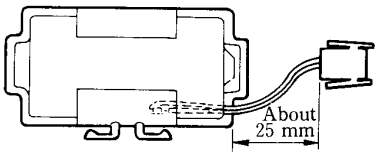


Fig. 9-5-3 Plugging in Battery Connector



House the lead wire in the battery case, leaving a portion about 25 mm long outside the case.

Fig. 9-5-4 Battery Case (top view)

10. USAGE OF COMMUNICATION MODULE

10.1 Remote I/O System

10.1.1 System Configuration

The system configuration of remote I/O is shown in Fig. 10-1-1.

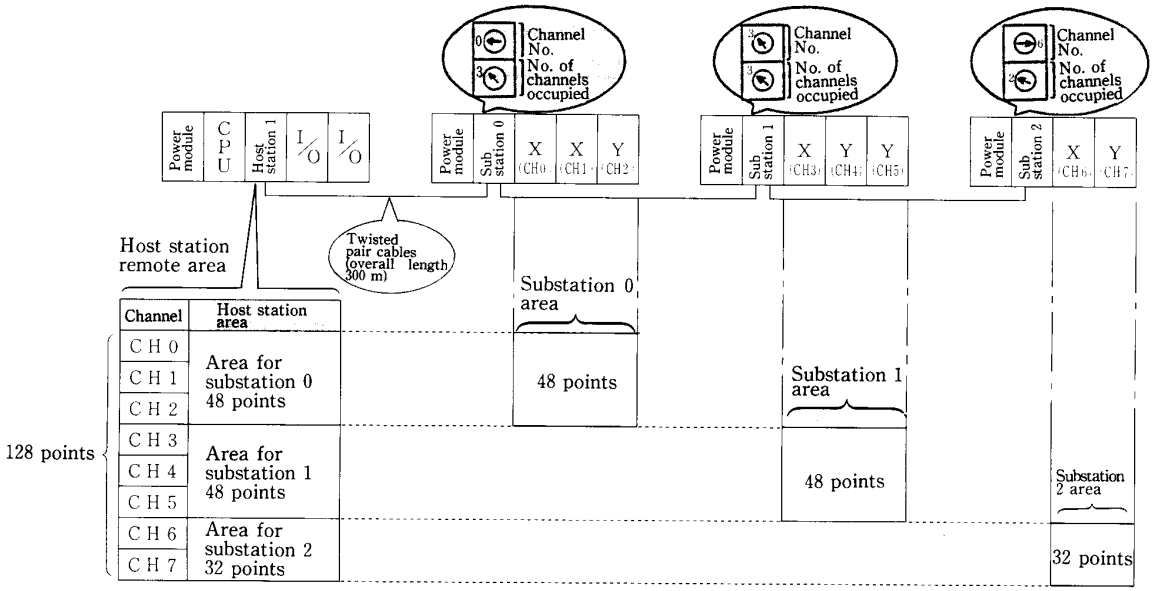


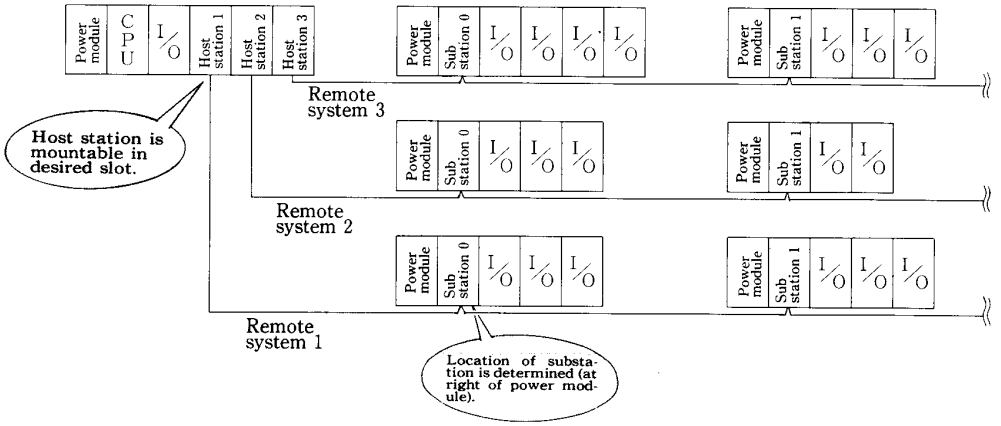
Fig. 10-1-1 System Configuration of Remote I/O

(1) Remote I/O system can be configured by connecting a remote host station [RIOH-TM] and substation [RIOH-TL] with twisted pair cables. In case the operation switch and/or indicator lamp is located away from the main control panel, I/O information is transmissible via twisted pair cables. This saves on wiring.

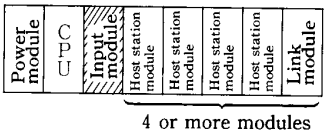
(2) The host station has a remote I/O area of 8 channels (CH0 to CH7). The I/O modules from the leading substation onward correspond to the channel number sequence in the host station area.

(3) Each channel of host station area consists of 16 bits. A single host station is capable of transmitting a total of 128 I/O points (16 points × 8 channels) to remote end.

(4) Up to 4 host stations are mountable in desired slots except for the CPU and power modules. Therefore, 4 remote I/O systems can be composed at maximum. Host stations are numbered 1, 2 and so on starting from the one closest to the CPU module.



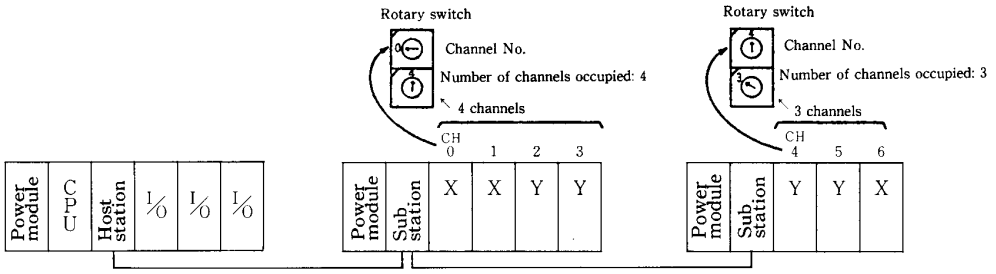
(5) Be sure to mount an input module in slot 0 next to the CPU module when the remote host station modules and CPU link modules connected to the CPU total 4 or more modules.



(6) Substation must be mounted at the right of the power module.
Up to 8 substations are connectable to a single host station.

(7) Substation side allows mounting of bit I/O modules alone.
Word I/O modules (analog and counter modules) cannot be mounted.

(8) Each substation has rotary switches which set the channel number of each channel and the number of channels occupied by the substation.



(9) Transmissible distance varies with the diameter of cable.
0.3 mm² twisted pair cable .... overall length 150 m
0.75 mm² twisted pair cable .... overall length 300 m (150 Ω termination required)

(10) The table below lists the specifications of remote host station and substation.

Item			Host station〔RIOH−TM〕	Substation〔RIOH−TL〕
General specifications	Operating temperature		0～55℃	
	Storage temperature		−20～70℃	
	Operating humidity		30～90%RH (non-condensing)	
	Current consumption	CH1(5V)	130mA	150mA
		CH2 (24V)	20mA	20mA
		CH3 (24V)	5mA	5mA
	Dimensions (mm)		35(W)×150(H)×117(D)	
	Weight (kg)		0.2	
Functional specifications	No. of connectable substations		(8 substations/host station) × 4 systems	
	No. of remote inputs/outputs		128 points × 4 systems	
	Baud rate		768kbps	
	Refresh time		About 5 ms	
	Error check		Inverted double transmission	

Transmission path	Recommended cable			Cable length		Termination resistor
	Type	Maker	Outside diameter	Between stations	Overall length	
	CO−SPEV−SB(A)−IP −0.3mm²	Hitachi Cable	About ∅5.5	150mMAX	150mMAX	Built in module(100Ω)
	CO−EV−SX−IP −0.75mm²		About ∅16	300mMAX	300mMAX	Requires external connection (150 Ω)

10.1.2 I/O Assignment and I/O Numbers

(1) For remote host station and substation, assign inputs and outputs as shown in Table 10-1-1.

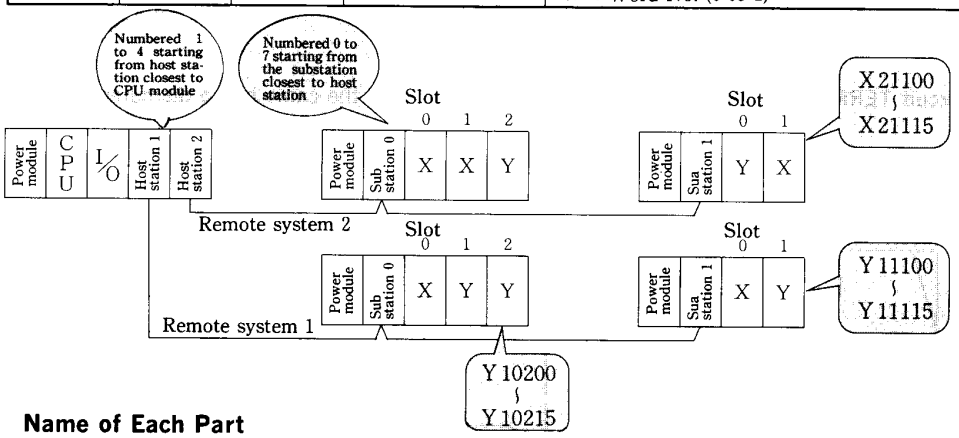
Table 10-1-1 I/O Assignment Code

Kind of module	Model	I/O assignment code
Remote host station	RIOH-TM	Remote
Remote substation	RIOH-TL	—— (setting unnecessary)

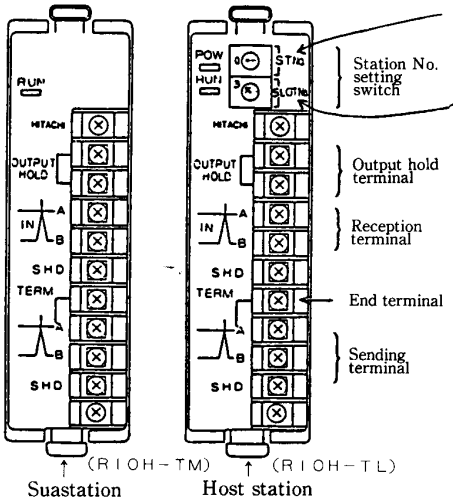
(2) Determine I/O numbers according to Table 10-1-2.

Table 10-1-2 Determination of I/O Numbers

Large category	Medium category	Small category	Code	Usable range
Remote I/O	External input	Bit	X [r][St][S][b][b]	r : Remote host station No. (1 to 4) St : Remote substation No. (0 to 7) S : Slot No. (0 to 7) bb : Bit No. (00 to 15), decimal W1 : Word No. (0 to 1)
		Word	WX [r][St][S][W1]	
	External output	Bit	Y [r][St][S][b][b]	
		Word	WY [r][St][S][W1]	



10.1.3 Name of Each Part



Sets the channel number of host station remote area in which the leading I/O module in the relevant substation is assigned.

Sets the number of I/O modules to be occupied by the relevant substation.

Output Hold Terminals

(1) **Host station (RIOH-TM)**

If a remote system is abnormal, data to be sent from the remote host station module to CPU module is determined.

- OUT HOLD terminals shorted ... Remote data before occurrence of abnormality is held.
- OUT HOLD terminals open ... All remote data is turned off (cleared to zero).

(2) **Substation (RIOH-TL)**

If a remote system is abnormal, output data of the output module on the substation module side is determined.

- OUT HOLD terminals shorted ... Output data before occurrence of abnormality is held.
- OUT HOLD terminals open ... Output data is all turned off.

10.1.4 Wiring Method

(1) Wiring method of the remote I/O modules is shown in Fig. 10-1-2.

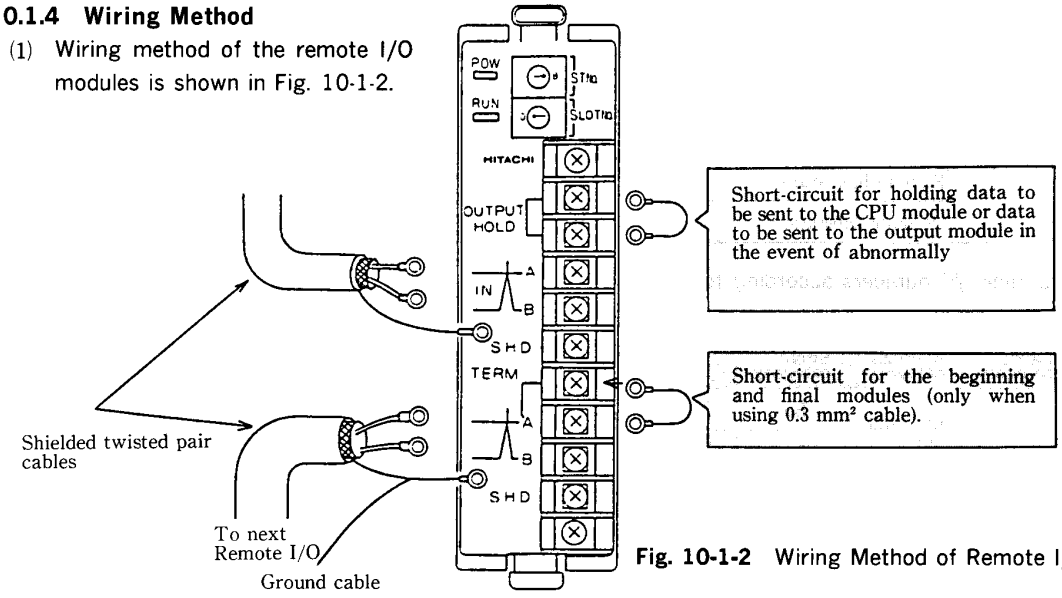
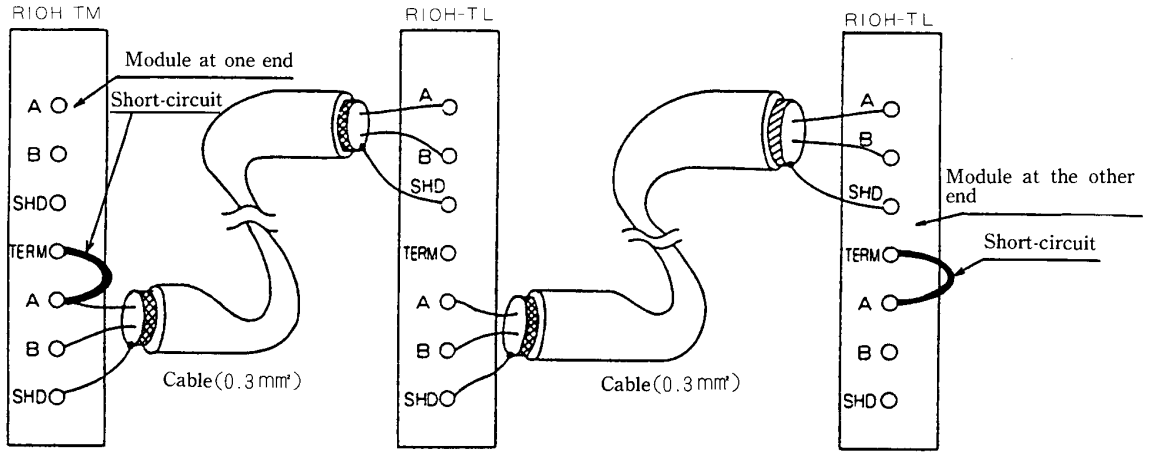
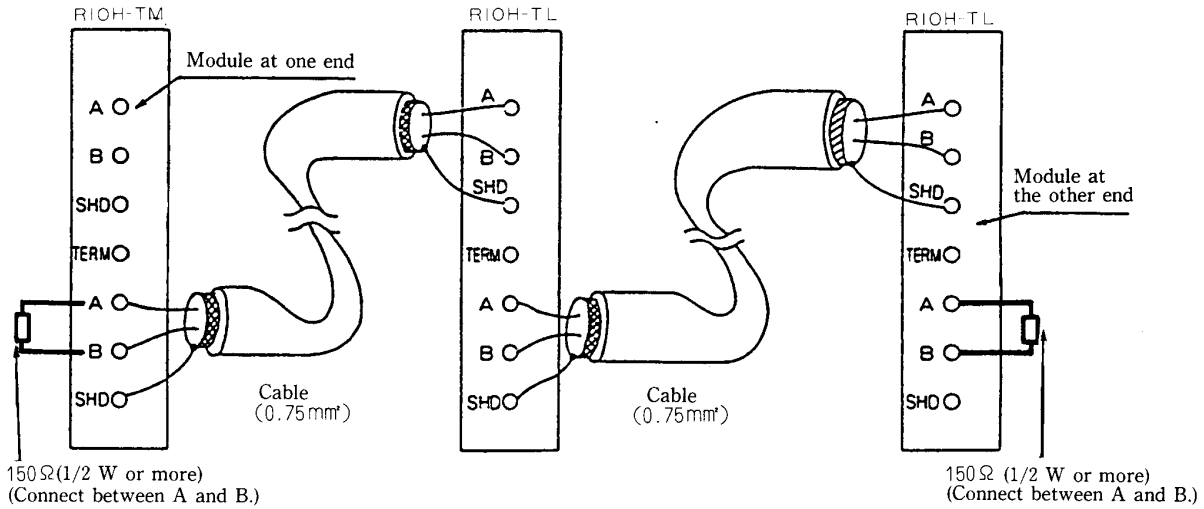


Fig. 10-1-2 Wiring Method of Remote I/O Module

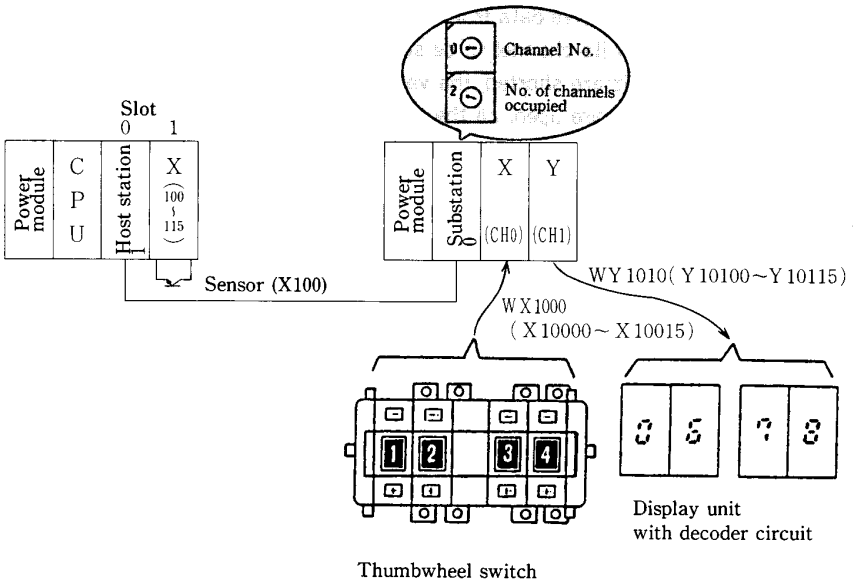
(2) Short-circuit TERM and A on the modules at both ends in case the cable has a diameter of 0.3 mm².



(3) Connect a resistor of 150 Ω (1/2 W or more) between A and B on the modules on both ends in case the cable has a diameter of 0.75 mm².



10.1.5 Example of Remote I/O Program



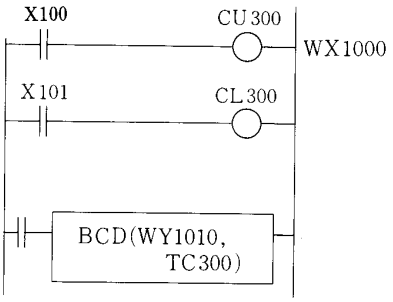
Explanation of Operation

(1) The number of ON/OFF times of the sensor X100 is counted by the counter CU300.

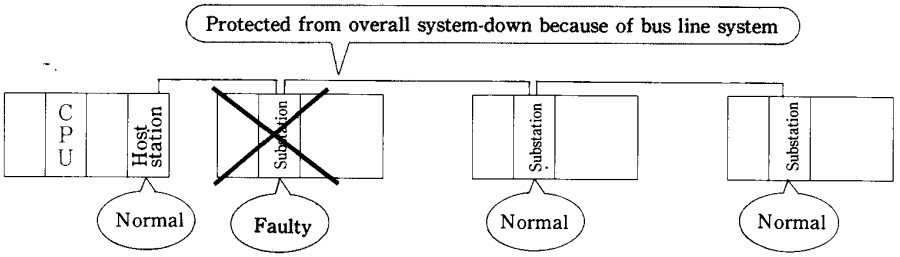
(2) The set value of counter CU300 is set by the thumbwheel switch on the substation side.

(3) The current value (TC300) of counter CU300 is indicated on the display unit on the substation side.

Programming



10.1.6 Troubleshooting



(1) **System is protected from overall inactivation even if a remote substation module becomes faulty.**
Even though one of the remote substation modules becomes faulty in the above example, the other substations maintain normal operation.
By the same principle, turning off power supply to one of the substation modules does not affect other modules.

(2) **In case host station cannot receive data:**

In case the host station does not receive data from a substation for about 500 ms, the substation is judged to be inactivated and the data of its channel to be sent to the CPU module is handled as follows.

① When the OUT HOLD terminals are shorted, the value before occurrence of abnormality is held.

② When the OUT HOLD terminals are open, all the data are turned to OFF (cleared to zero).

In case system-down occurs on all substations, the RUN lamp of host station goes off.

(3) **In case a substation cannot receive data:**

In case a substation does not receive data within about 500 ms, the RUN lamp is turned off and the data of output module on the substation side is handled as follows.

① When the OUT HOLD terminals are shorted, the value before occurrence of abnormality is held.

② When the OUT HOLD terminals are open, outputs are all turned off.

(4) **Wrong setting of rotary switch (substation)**

Upon energization, the checks below are performed. If setting is wrong, the RUN lamp flickers at intervals of 0.1 sec. In this case, correct setting and turn on power supply again.

① Check if a station number (channel number) other than 0 to 7 is set.

② Check if the set number of channels occupied is other than 1 to 8.

③ Check if the total of station numbers (channel numbers) and occupied channel numbers is other than 1 to 8.

Module reads the setting of each rotary switch only when turning on power supply.

10.2 CPU Link System

10.2.1 System Configuration

The system configuration of CPU link is shown in Fig. 10-2-1.

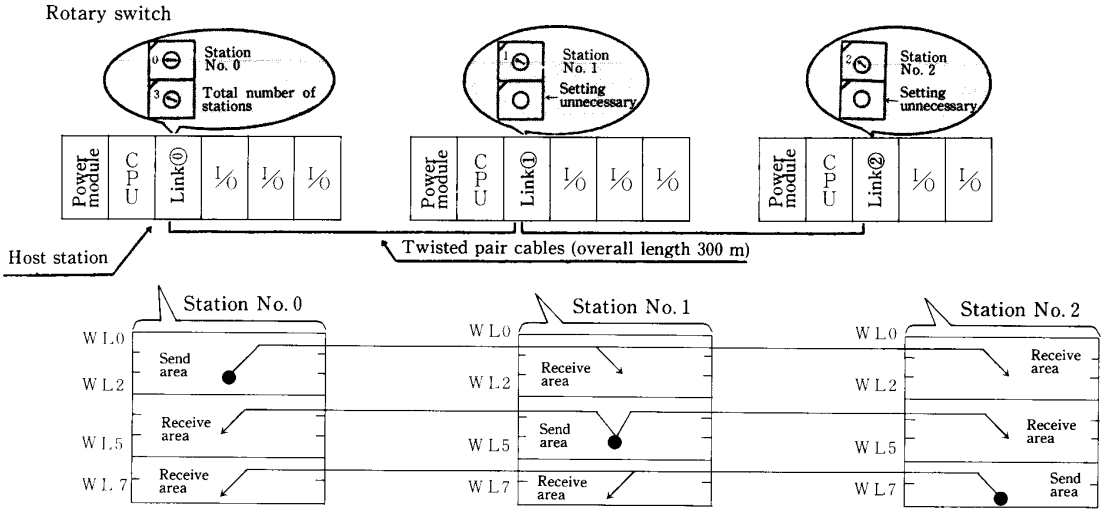
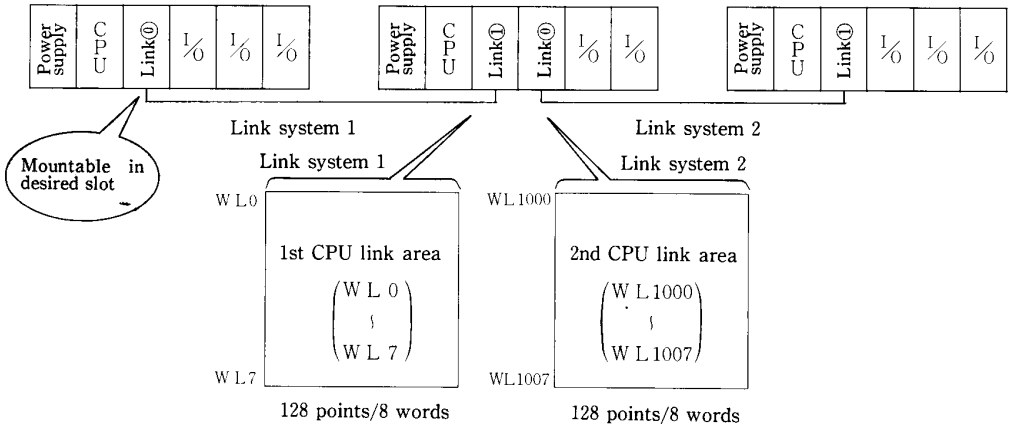


Fig. 10-2-1 System Configuration of CPU Link

(1) A CPU link system can be configured by connecting CPU link modules [IOLH-T] with twisted pair cables. Up to 8 CPU modules can be linked.

(2) Data is sent and received using the CPU link area (bit/word common area: L/WL).
There are two CPU link areas, each consisting of 128 points/8 words. Therefore, CPU link can be formed in 2 systems. One CPU link system is capable of linking 8 CPU's at maximum.



(3) Although each CPU link module in a single system is given a station number within 0 to 7 so as not to use the same number doubly, station number 0 (host station) is absolutely required. In the host station, the total number of stations connected in the same link system need be set.

(4) The CPU link area is composed of send area and receive area. In the send area, the data of own station is set. This data is sent to each link area sequentially in the same link system. This means that the CPU's in the same link system share identical data.

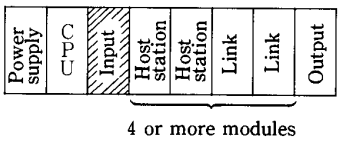
(5) Unless CPU link is used, the CPU link area is usable as a bit/word common internal output.

(6) Transmissible distance varies with the diameter of cable.
0.3 mm² twisted pair cables ... Overall length 150 m
0.75 mm² twisted pair cables ... Overall length 300 m (150 Ω termination required)

(7) The specifications of CPU link module are listed below.

Item			Specifications				
General specifications	Operating temperature		0 ~ 55℃				
	Storage temperature		−20 ~ 70℃				
	Operating humidity		30 ~ 90%RH (non-condensing)				
	Current consumption	CH1(5V)	150mA				
		CH2 (24V)	20mA				
		CH3 (24V)	5mA				
	Dimensions (mm)		35(W) × 150 (H) × 117 (D)				
	Weight (kg)		0.2				
Functional specifications	No. of connectable modules		(Max. 8 modules/system) × 2 systems				
	No. of link points		(128 points/8 words) × 2 areas				
	Baud rate		768kbps				
	Refresh time		10 ms × No. of stations (Note)				
	Error check		Inverted double transmission				
Transmission path	Recommended cable				Cable length		Termination resistor
	Type		Maker	Outside diameter	Between stations	Overall length	
	CO−SPEV−SB(A)−IP	−0.3mm²	Hitachi Cable	About ∅5.5	150mMAX	150mMAX	Built in module (100 Ω)
	CO−EV−SX−IP	−0.75mm²		About ∅16	300mMAX	300mMAX	External connection required (150 Ω)

Note:
Be sure to mount an input module in slot 0 next to the CPU module in case the total number of remote host station modules connected to CPU and the CPU link modules is 4 or more.

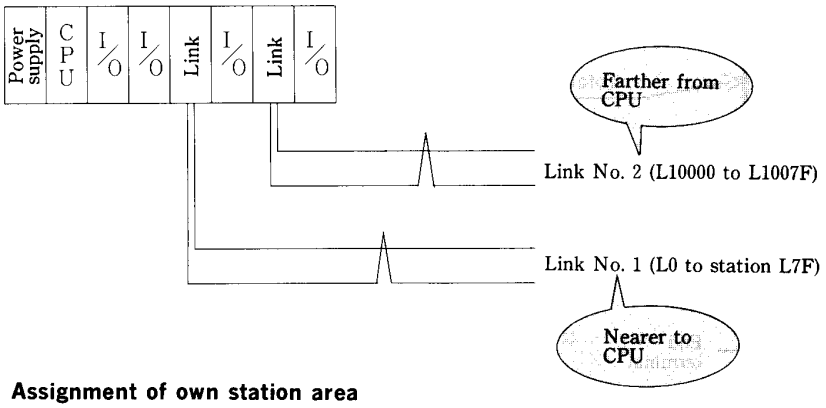


10.2.2 CPU Link Assignment

(1) **Link No. assignment**

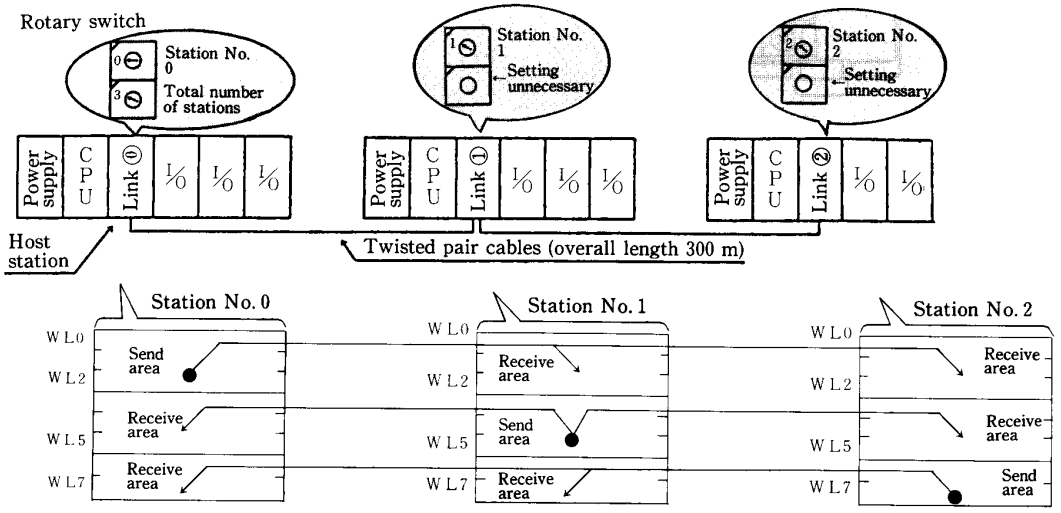
Link number assignment is not required.

The link module mounted in a slot nearer to the CPU is numbered 1 and the one mounted in a slot farther from the CPU is numbered 2.

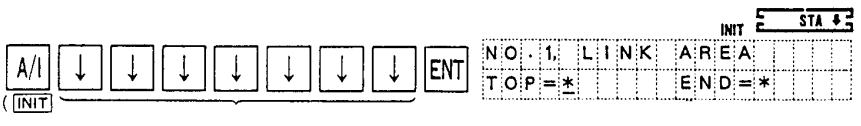


(2) **Assignment of own station area**

In the leftmost example shown in the figure below, the own station area corresponds to WL0 to WL2.



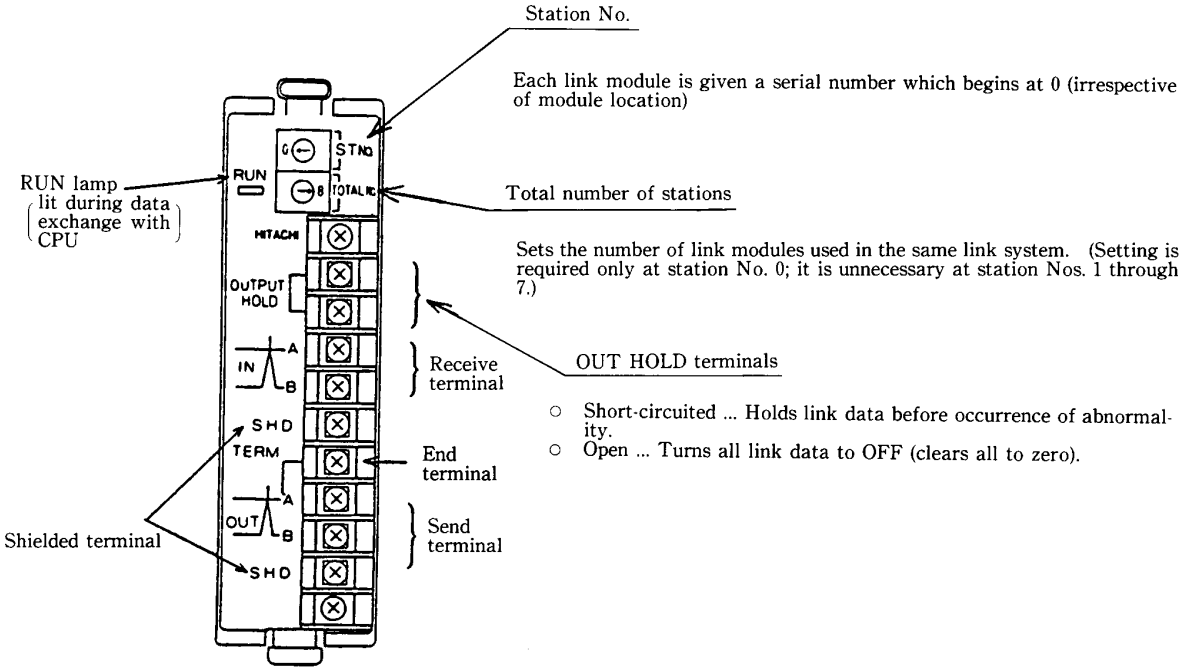
Setting must be made from the programming device by use of the CPU link parameter setting function.

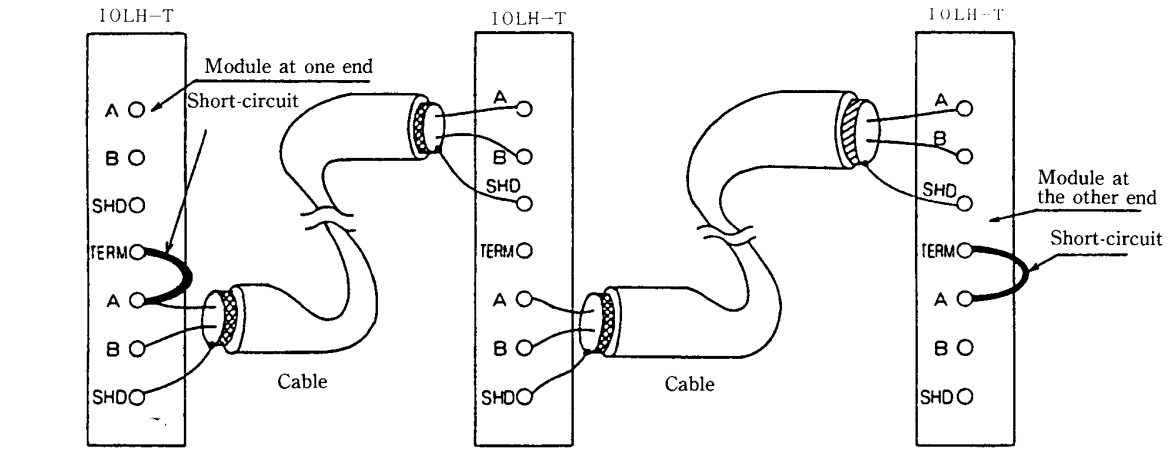


LINK AREA	TOP	END
No. 1	0	2
No. 2	3	5
No. 3	6	7

Example for Link area.

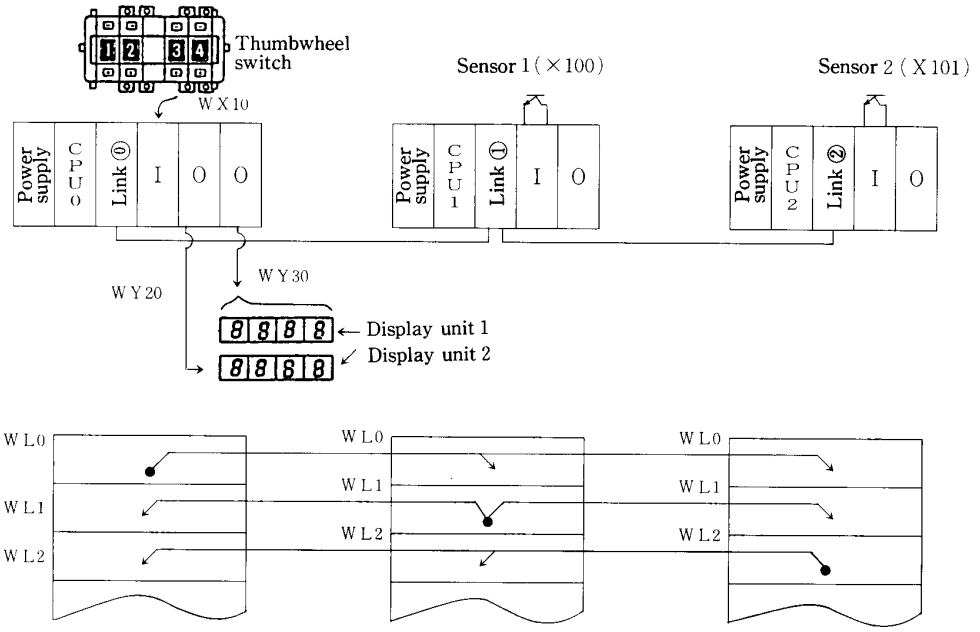
10.2.3 Name of Each Part and Wiring Method





10.2.4 Example of CPU Link Program

System configuration



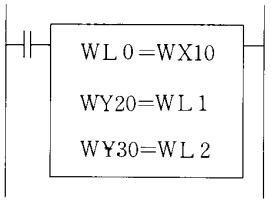
Explanation of Operation

(1) The set value of counter CU50 in each CPU module of station Nos. 1 and 2 is determined by the thumbwheel switch of station No. 0.

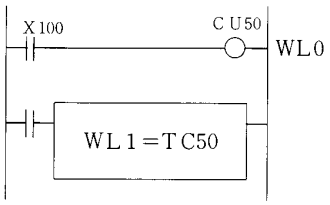
(2) ON/OFF frequency of sensors 1 and 2 is counted with counter CU50 in each CPU module of station Nos. 1 and 2.

(3) The current value of the counter of station No. 1 is indicated on the display unit 1 of station No. 0, and the current value of the counter of station No. 2 is indicated on the display unit 2 of station No. 0.

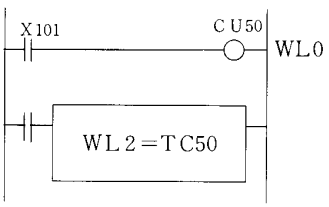
Programming



CPU 0 Program

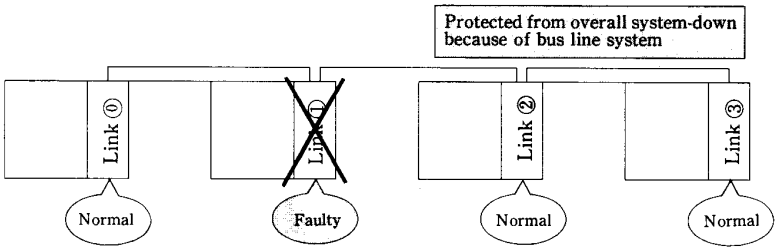


CPU 1 Program



CPU 2 Program

10.2.5 Troubleshooting



(1) **System is protected from overall inactivation even if a link module becomes faulty.**
Even though the link module of station No. 1 becomes faulty in the above example, the other stations maintain normal operation.
By the same principle, turning off power supply to station No. 1 does not affect other stations.

(2) **In case host station cannot receive data:**
In case the host station does not receive data for a time period 6 times as long as the refresh time (480 ms max.), system-down is judged to have occurred. As a result, the RUN lamp of host station is turned off and transmission is interrupted. Then, send request is issued to the substation at intervals of 9 ms. When the normal status is restored in the substation, the host station automatically restores the normal status to recompose the link system.

(3) **In case substation cannot receive data:**
In case substation does not receive data within about 500 ms, system is judged to be inactivated. In consequence, the RUN lamp of substation is turned off and reception is awaited.

(4) **Data transmission between CPU module and link module of H-200 in system-down status**
① When the OUT HOLD terminals are short-circuited, link data before occurrence of abnormality is held.
② When the OUT HOLD terminals are open, link data is all turned to OFF (cleared to zero).

(5) **Wrong setting of rotary switch**
On energization, the checks below are carried out.
If setting is wrong, the RUN lamp of link module flickers at intervals of 0.1 sec. In this case, correct setting and turn on power supply again.
① Check if a station number other than 0 to 7 is set.
② Check if the set total number of stations (only at host station) is other than 2 to 8.

(6) **Duplication of station number**
If the same station number is used doubly, transmission is not carried out. Instead, the RUN lamp of link module flashes at intervals of 0.5 sec. In this case, correct setting and turn on power supply again.

10.3 I/O Link System with Upper-Level Models (H-300/700/2000)

10.3.1 System Configuration

I/O link system can be configured with the H-series upper-level model (H-300, 700 or 2000) by mounting a coaxial type host link module [RIOM] to the H-200.

System configuration is exemplified in Fig. 10-3-1.

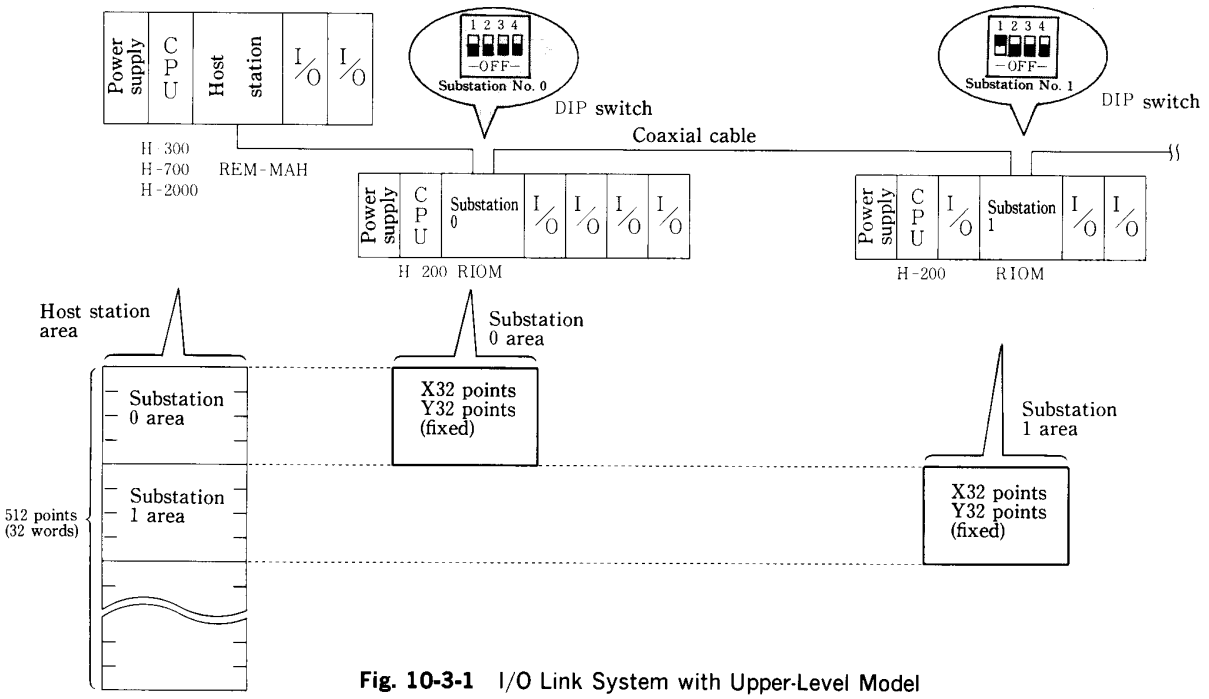


Fig. 10-3-1 I/O Link System with Upper-Level Model

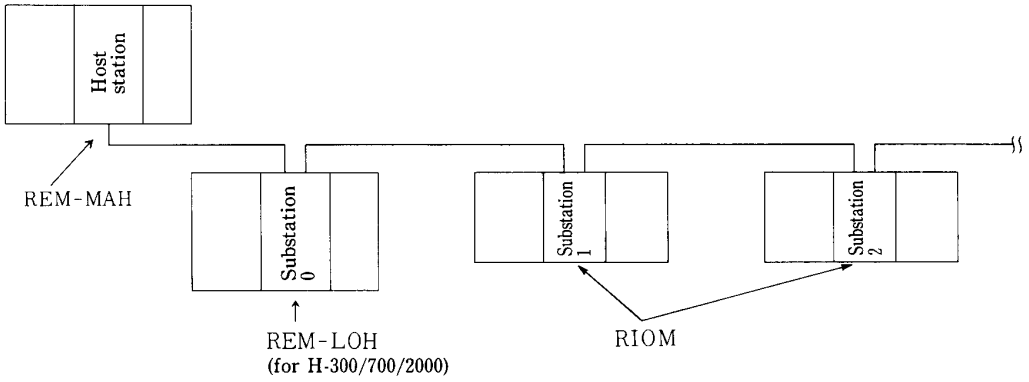
(1) Mount a remote host station [REM-MAH] to the upper-level model of H-series, and a host link module [RIOM] to the Model H-200. Then connect them with a coaxial cable. An I/O link system has now been composed. The RIOM serves as a substation.

(2) The host station has a remote I/O area of 512 points. Each RIOM occupies input and output of 32 points each.
(The number of occupied points is fixed.)
Therefore, up to 8 RIOM's are connectable to each host station.

(3) Host station is mountable in a desired slot for other than the CPU module of basic base and the power module. Substation is also mountable in a desired slot for other than CPU and power modules.

(4) Substation is provided with a DIP switch is used to set the substation number (station number). Set a number within 0 to 7 in order starting from the one closest to the host station.

(5) The RIOM is usable in parallel with the remote substation [REM-LOH] of the upper-level model.



(6) The specifications of RIOM are listed in the table below.

Specifications of RIOM

Item		Description	
General specifications	Operating temperature	0 ~ 55 ℃	
	Storage temperature	−10 ~ 75 ℃	
	Operating humidity	30%~90%RH (non-condensing)	
	Current consumption	CH1(5V)	500mA
		CH2(24V)	30mA
		CH3(24V)	0mA
	Dimensions (mm)	70mm(W)×150mm(H)×110mm(D)	
	Weight	About 400 g	
Functional specifications	No. of connectable modules	8 RIOM's/host station	
	No. of link peoints	Input 32 points, output 32 points/RIOM (fixed)	
	Baud rate	1. 5 Mbps	
	Transmission system	Half duplex serial transmission, frame synchronization	
	Meodulation system	Base band	
	Refresh time	Max. about 15 ms/512 points (for 8 substations)	
	Error check	CRC, sum check	
	Self-diagnosis	System ROM/RAM check, watch dog timer check, transmission loopback check	
Transmission path	Cable length	Between stations	500m
		Overall length	500m
	Abnormal station processing	Bypass system	
	Cable and connector used	Coaxial cable	5D-2V equivalent (shielded)
		Connector on cable side	BNC-P-5DV (made by Hirose) recommended

10.3.2 Concept of I/O Link

When RIOM is connected to host station (REM-MAH), the host station regards it as a 4-slot I/O (X16, X16,

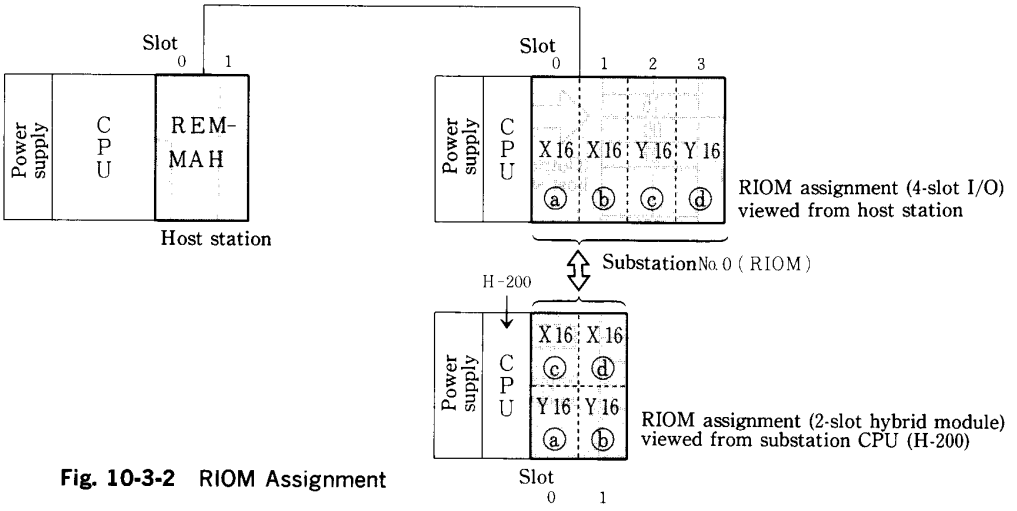
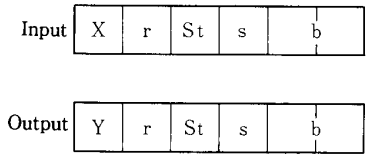


Fig. 10-3-2 RIOM Assignment

Y16 and Y16, this order fixed.) On the other hand, a substation CPU (H-200) regards RIOM as a 2-slot hybrid module (slot 0: X16, Y16, slot 1: X16, Y16). The area is linked between the host station and substation. Input (X) of the host station link area is sent from the substation (input (X) corresponds to output (Y) when viewed from the substation CPU). Output (Y) from the host station is fetched into the substation (output (Y) corresponds to input (X) when viewed from the substation CPU). Therefore, link areas correspond between the host station and substation as indicated by (a), (b), (c) and (d) in Fig. 10-3-2.

I/O numbers of the RIOM are assigned on the host station side as shown below.



r: Remote host station No. (1 to 4)
St: Remote substation No. (0 to 9)
s: Slot No. in substation (0 to 3 in system shown in the above figure)
b: Bit No. in module (00 to 15)

In the system shown in Fig. 10-3-2, RIOM numbers correspond as listed below.

Area	RIOM number viewed from host station	RIOM number viewed from substation CPU
ⓐ	X10000～X10015	Y016～Y031(WY01)
ⓑ	X10100～X10115	Y116～Y131(WY11)
ⓒ	Y10200～Y10215	X000～X015(WX00)
ⓓ	Y10300～Y10315	X100～X115(WX10)

This correspondence can be generalized as shown in Fig. 10-3-3.

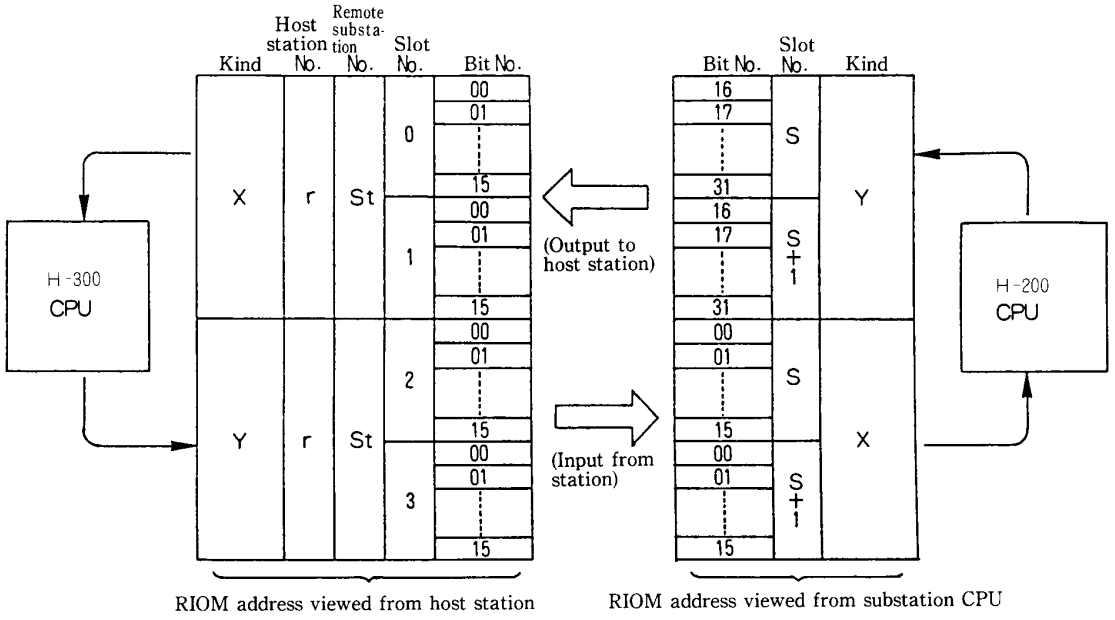


Fig. 10-3-3 Address Correspondence of RIOM

Fig. 10-3-4 exemplifies an I/O link program.

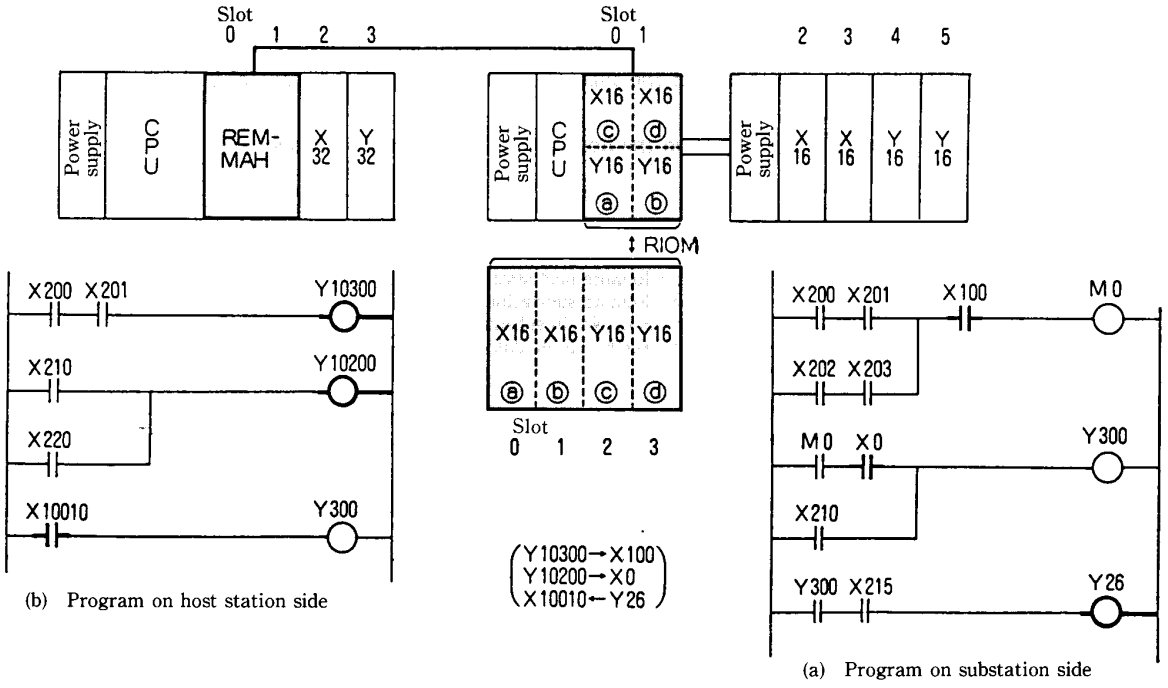


Fig. 10-3-4 Example of I/O Link Program

When host station Y10300 turns on, substation X100 turns on.
When host station Y10200 turns on, substation X0 turns on.
When substation Y26 turns on, host station X10010 turns on.

10.3.3 Mounting Method

(1) **How to connect coaxial cable**

Connection of the coaxial cable is exemplified in Fig. 10-3-5.

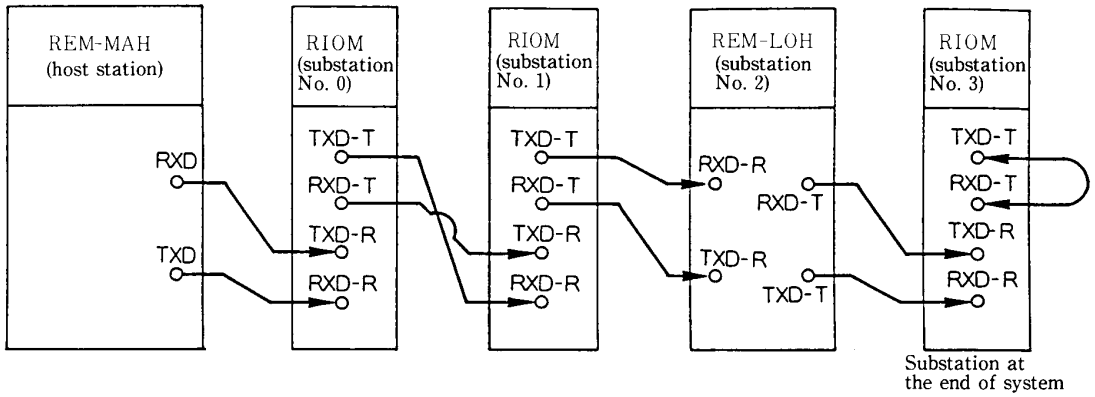


Fig. 10-3-5 Example of Coaxial Cable Connection

Connect TXD-T to RXD-R, and RXD-T to TXD-R. At the final substation, connect TXD-T and RXD-T. Fig. 10-3-6 shows the location of DIP switch and examples of switch setting.

(2) **How to set substation No.**

Substation No. is settable by the DIP switch on the 1F circuit board of RIOM.

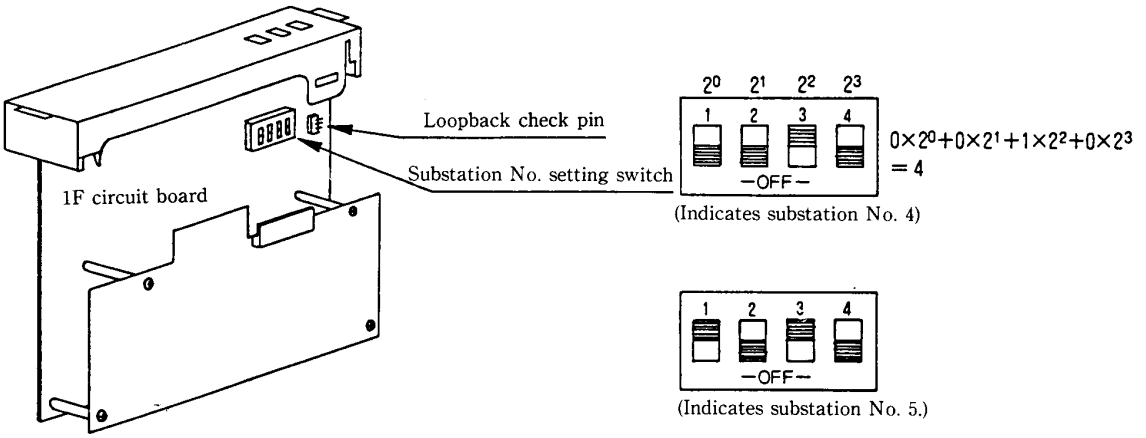


Fig. 10-3-6 Setting of Substation No.

(3) **Caution on mounting**

① **Processing of coaxial cable end**

For coaxial cable, the shielded 5D2VTXE (Fujikura Densen make or equivalent) or 5D-2VCCY (Showa Densen make or equivalent) is recommended.

For connector, the BNC-P-5DV (Hirose make or equivalent) is recommended. For processing method, refer to "Processing of coaxial cable end" in the "Manual of H-series remote I/O module (for coaxial cable)" (H-C001-1).

② **Caution on installation in control panel**

When connecting a cable provided with a straight type coaxial connector to the remote module, the cable stands out by about 70 mm from the front panel of the module as shown in Fig. 10-3-7. So take this into account when designing a control panel.

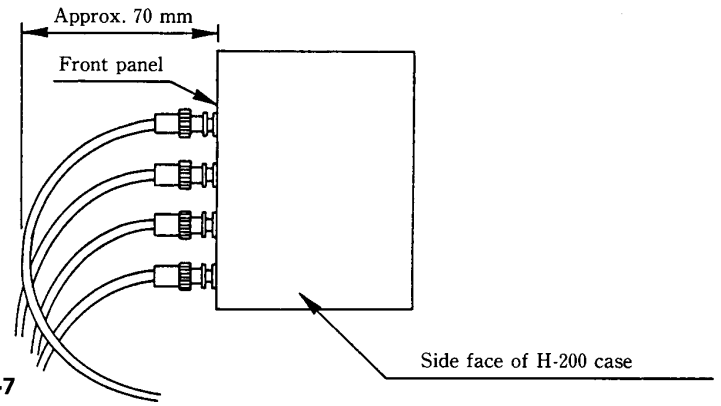


Fig. 10-3-7

③ **Caution on wiring of coaxial cable**

Separate the coaxial cable from the input/output line and power line. If the coaxial cable cannot be routed separately, then utilize a rigid metal conduit.

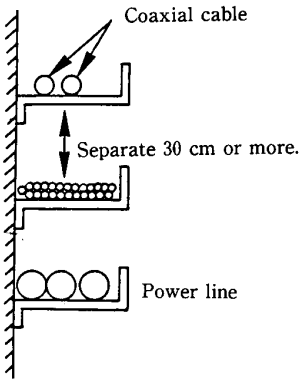


Fig. 10-3-8 Separation of Wires with Conduit or Duct

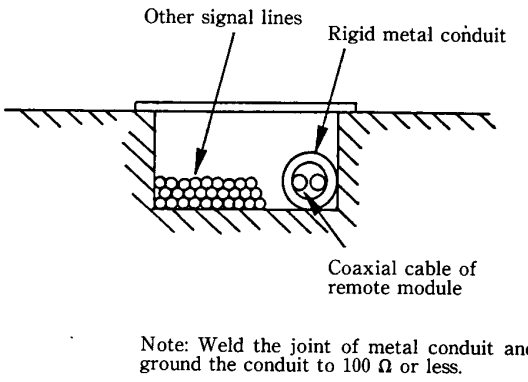


Fig. 10-3-9 Separation of Wires with Rigid Metal Conduit

10.3.4 Maintenance

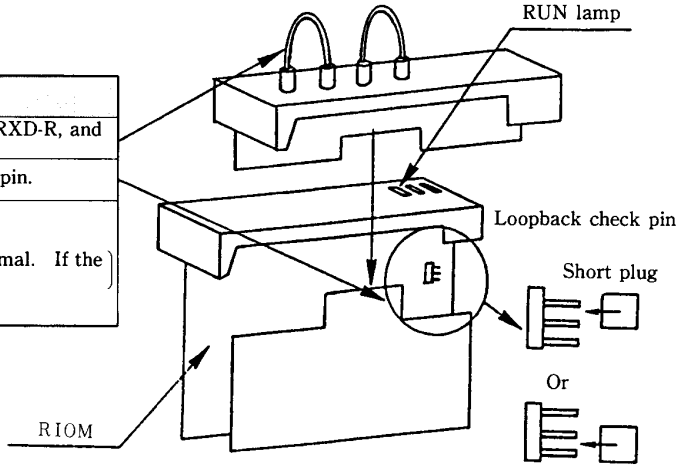
If transmission is impossible with the RUN lamp remaining off, the check below is required.

(1) **Loopback check (impossible during operation)**

① Purpose: Check whether or not an individual remote module is normal.
This check aims at an individual remote module alone indifferently to other remote modules.
Therefore, this check cannot be conducted during operation.

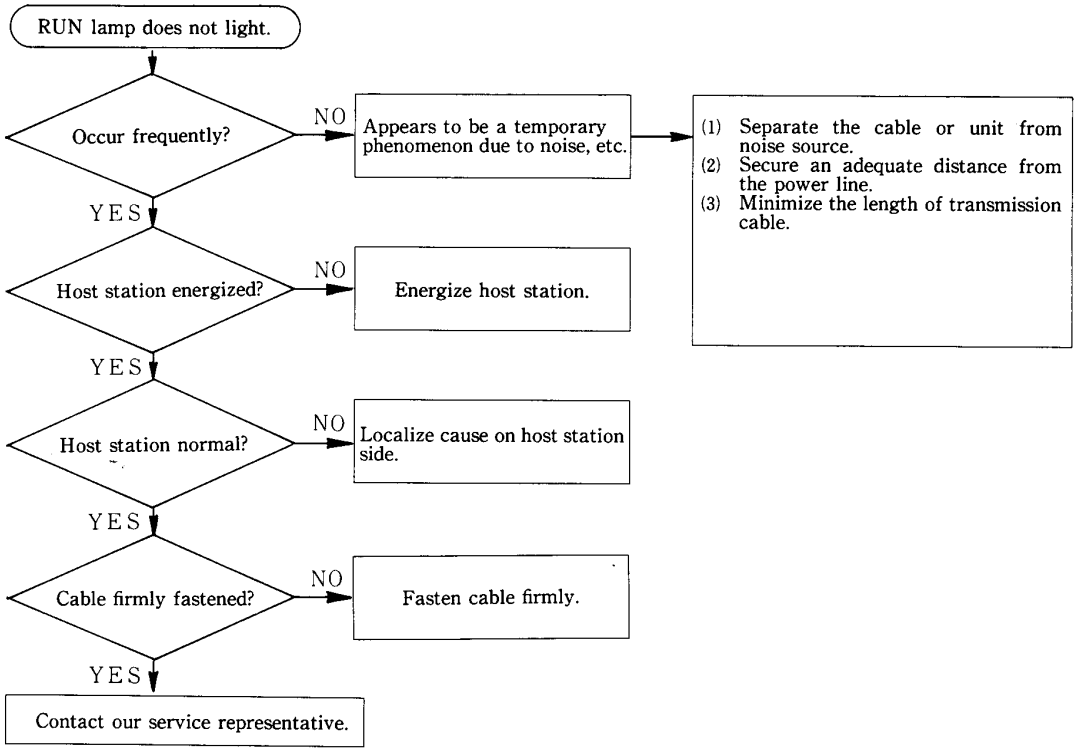
② Check method

Procedure
Connect coaxial cable (between TXD-R and RXD-R, and between TXD-T and RXD-T).
Fit short plug to the loopback check pin.
Turn on power supply. Check will start. {If the RUN lamp comes on, the module is normal. If the RUN lamp flickers, the module is defective.}



Caution: Before operation, pull out the short plug.

(2) **Troubleshooting**



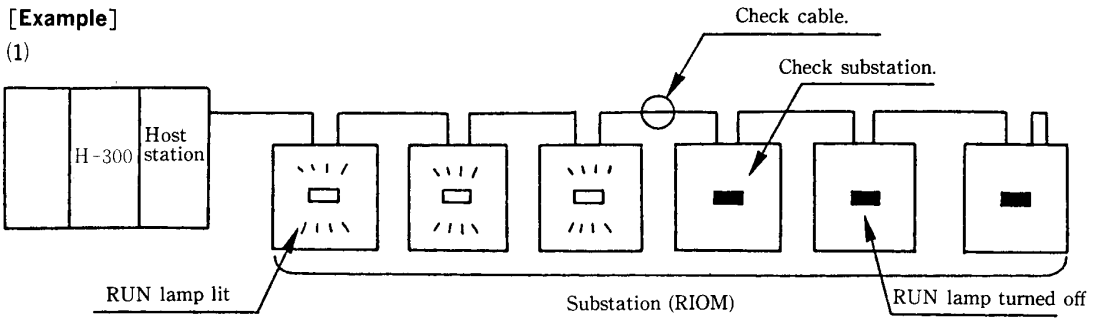
(3) **Method of localizing abnormal location**

Remote error occurs in the host station if even one of the substations or cables connected to the host station does not function. In this case, operation cannot be carried out. However, forced operation is possible when operation parameters in CPU setting on the host station side have been set so as to continue operation despite a remote error. At this time, the abnormal location can be assumed to be around a substation where the RUN lamp is turned off or input/output is disabled.

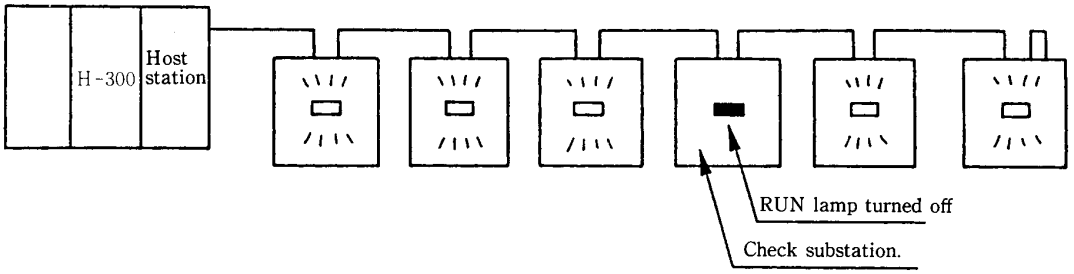
Conduct a loopback check on such a substation. If the substation is normal, then cable is defective.

[Example]

(1)



(2)

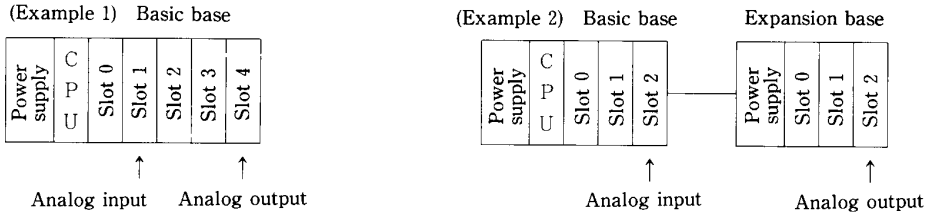


11. USAGE OF ADVENCED FUNCTION MODULES

11.1 Assignment of Analog I/O Modules

(1) **Mounting and I/O assignment**

Analog I/O modules can be installed in desired slots for other than the CPU module and power module. But they cannot be mounted at the remote end. Mounting of the analog I/O modules is exemplified below. At the time of I/O assignment, analog I/O modules are all assigned as 8-word modules. Analog input and analog output are assigned as WX8W and WY8W.



Example of Analog I/O Module Mounting

(2) **I/O numbers**

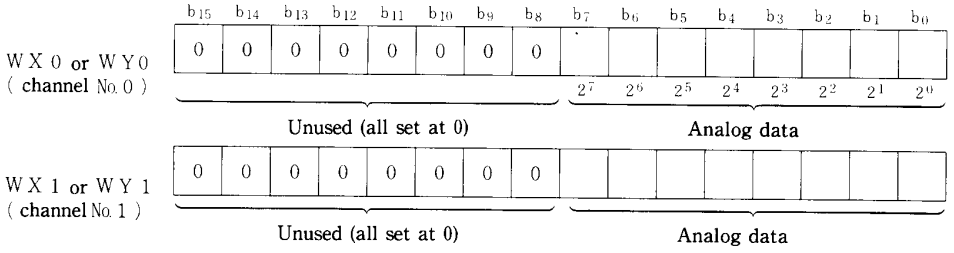
The I/O numbers of analog modules are listed in Table 11-1.

Table 11-1 I/O Numbers of Analog Module

Base	Slot	Channel	アナログ入力〔WX8W〕 AGH-I, IV	Analog output〔WY8W〕 AGH-O, OV	AGH-OD, ODV
Basic base	0	0	WX0	WY0	WY0
		1	WX1	WY1	WY1
		2	WX2	WY2	—
		3	WX3	WY3	—
		4	WX4	—	—
		5	WX5	—	—
		6	WX6	—	—
		7	WX7	—	—
	1		WX10, WX11, WX12, WX13, WX14, WX15, WX16, WX17	WY10, WY11, WY12, WY13	WY10, WY11
	⋮		⋮	⋮	⋮
Expansion base	0		WX100, WX101, WX102, WX103, WX104, WX105, WX106, WX107	WY100, WY101, WY102, WY103	WY100, WY101
	1		WX110, WX111, WX112, WX113, WX114, WX115, WX116, WX117	WY110, WY111, WY102, WY113	WY110, WY111
	⋮		⋮	⋮	⋮

(3) **Data configuration**

Analog data is set in the low-order 8 bits of 1 word (16 bits). The high-order 8 bits are all set to "0".

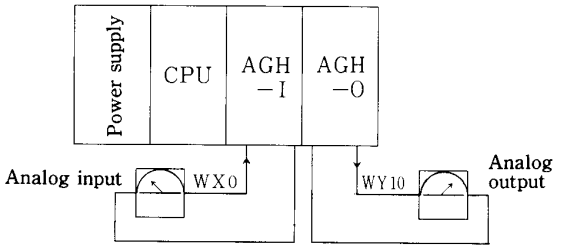


11

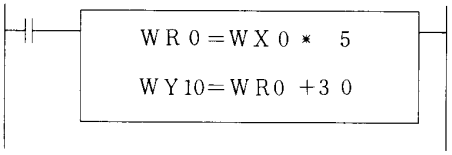
(4) **Example of program**

Computational Formula

Y = X * A + B (computational formula)
- X: Analog input (assigned to WX0)
- Y: Analog output (assigned to WY10)
- A: 5 (decimal constant)
- B: 30 (decimal constant)



Programming



WR0 ← analog input (WX0) * 5 (constant)
Analog output (WY10) ← WR0 + 30 (constant)

Analog modules have a resolution of 8 bits. They come in either current or voltage type. The relation between analog data and digital data is shown below.
Data is all BIN.

Type	A → D conversion (input)	D → A conversion (output)
Current type	4 ~ 20mA → 00 ~ FF	00 ~ FF → 4 ~ 20mA
Voltage type	0 ~ 10V → 00 ~ FF	00 ~ FF → 0 ~ 10V

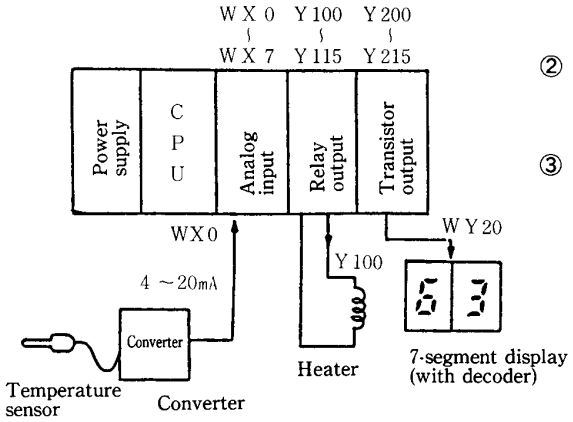
- In case BCD data is necessary, data can be converted into BCD by issuing BCD (d, s) instruction.
- In case data is to be output to an analog output module after BCD computation, data need be converted into BIN by using BIN (d, s) instruction.

(5) **The specifications of analog modules are listed in the table below.**

Item \ Model	Current type			Voltage type		
	AGH−I	AGH−O	AGH−OD	AGH−IV	AGH−OV	AGH−ODV
Type	Current input	Current output		Voltage input	Voltage output	
Range	4 ~ 20mA			0 ~ 10V		
Resolution	8 bits			8 bits		
Conversion time	1 ms			1 ms		
Overall accuracy	±(1%+1 bit)	±1%		±(1%+1 bit)	±1%	
No. of channels	8	4	2	8	4	2
Isolation means	Photocoupler			Photocoupler		
Isolation between channels	Not provided			Not provided		

(6) **Application example of analog module**

Explanation of Operation

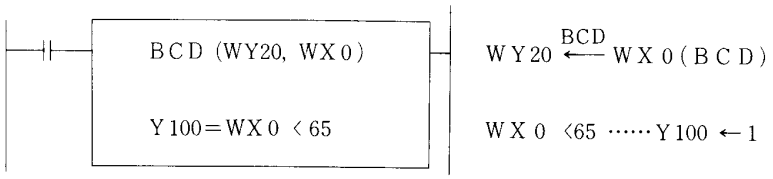


① Temperature is measured and displayed on the 7-segment display.

② If temperature is lower than 65°C, the heater turns on. At 65°C or higher, the heater turns off.

③ Assignment
Temperature data ... WX0
Heater output ... Y100
Display (2 digits) ... WY20 (Y200 to Y207)

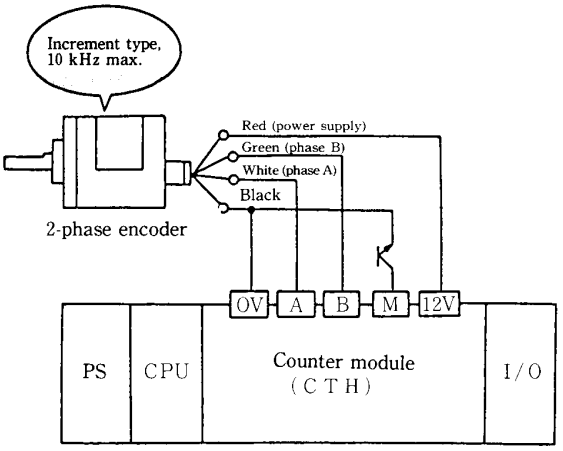
Programming



11

11.2 Counter Module

(1) Explanation of operation

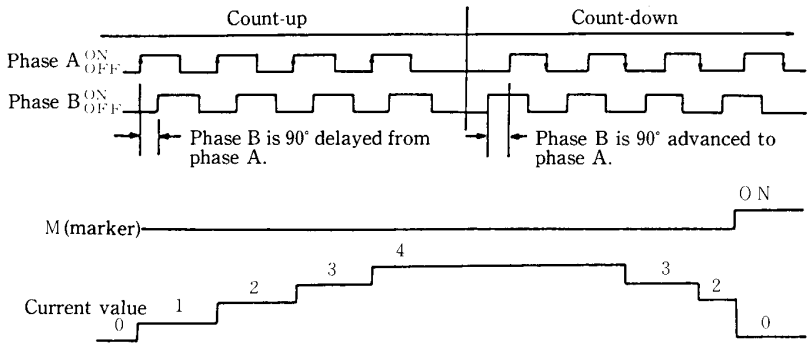


Explanation of Operation

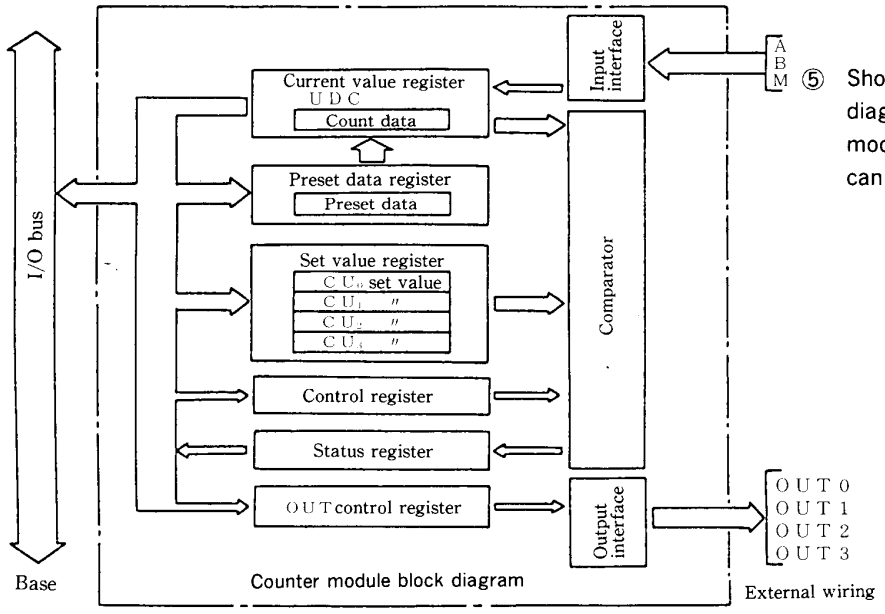
① Two-phase pulse signal (10 kHz max.) from the encoder is fetched to compare the set value and currnt value.

② Count-up or count-down operation is determined depending on the phase difference of pulse inputs of phases A and B.

③ Changeover from count-up to count-down operation and vice versa is controlled so that current value does not change by 1 pulse.



④ When M (marker) turns on, current value is reset to "0000."



⑤ Shown at left is a block diagram of the counter module. Four set values can be assigned.

(2) **Mounting and I/O assignment**

The counter module is mountable in a desired slot for other than the CPU and power modules. However, the counter module cannot be mounted at the remote end.

At the time of I/O assignment, this module is to be assigned as FUN3.

(3) **Register**

The register configuration of counter module is shown in Fig. 11-1.

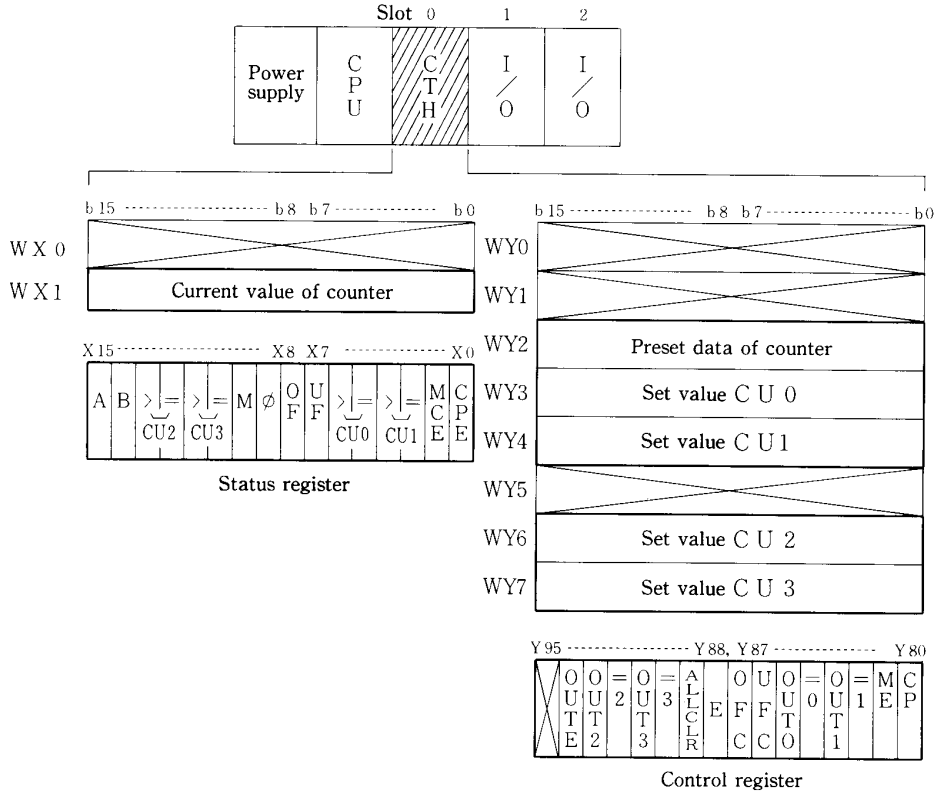


Fig. 11-1 Register Configuration

① Control register

Specifies operation of the counter module. I/O numbers of the control register are given in Y ☐ 80 to Y ☐ 95 (☐ : slot No.). The functions of control register are listed in Table 11-2.

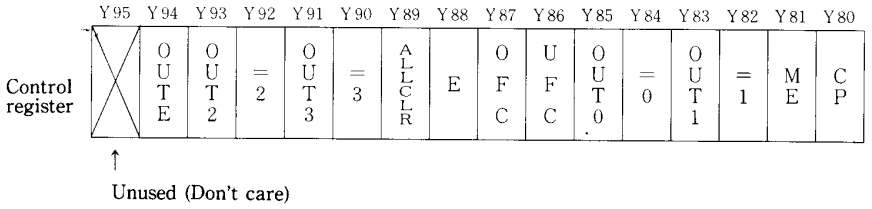
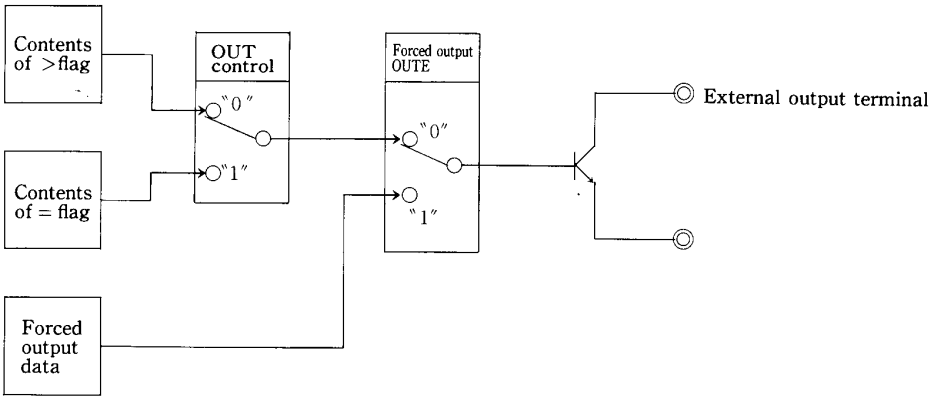


Table 11-2 Functions of Control Register

No.	Code	Flag name		Function
1	C P	Counter preset	⎍	Changes current value to preset value. (Counter does not operate while this flag is at "1".)
2	M E	Marker enable	0	Invalidates marker terminal input
			1	Validates marker terminal input. Marker terminal ON: Current value → 0
3	$\overline{=}$ 0 ~ $\overline{=}$ 3	= flag clear	0	Holds = flag.
			1	Clears = flag.
4	OUT 0 ~ OUT 3	OUT control	0	Outputs the contents of > flag to the output terminals OUT0 to OUT3.
			1	Outputs the contents of = flag to the output terminals OUT0 to OUT3.
				Each of OUT0 to OUT3 is controllable independently.
5	U F C	Underflow flag clear	0	Does not clear underflow flag (UF).
			1	Clears underflow flag (UF). (Underflow flag is not set (or does not change from 1 to UF) despite underflow when this flag is at "1.")
6	O F C	Overflow flag clear	0	Does not clear overflow flag (OF).
			1	Clears overflow flag (OF). (Overflow flag is not set (or does not change from 1 to OF) despite overflow when this flag is at "1.")
7	E	Counter enable	1	Enables counter read and write from CPU.
8	ALL CLR	All clear	⎍	Clears current value of counter and all flags.
9	OUT E	Forced output (Note 1)	1	Forcibly turns on external output terminals OUT0 to OUT3 of the counter module when setting "1" to OUT0 to OUT 3 (indifferently to the result of comparison between the current value and set value). Each of OUT0 to OUT3 is controllable independently.

Notes:

1. Forced output is possible when forced output data is set to OUT0 through OUT3 with OUTE (forced output) at "1".



2. At the time of initialization, be sure to make the following setting.
   "1" → E, ME
   ⎍ (1 scan pulse) → ALL CLR

② Preset data register

Presets current value. [Preset data] is fetched as [current value] by turning the counter preset [CP] of control register to ⎍ (1 scan pulse).

③ Set value register

Four kinds of set values can be written in CU0 to CU3.

Current value is compared with set value, and the result is output to the status register and output terminals (OUT0 to OUT3).

④ Status register

Indicates the operational status of counter module.

The result of set value and current value comparison, or overflow/underflow of current value is indicated.

The functions of status register are listed in Table 11-3.

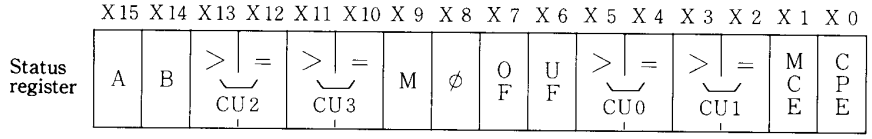


Table 11-3 Functions of Status Register

No.	Code	Flag name	Function	
1	C P E	Preset end flag	Indicates that preset data has been fetched as a current value of counter. When CP = ⎍, "1" is changed to CPE.	
2	M C E	Marker enable set end flag	When marker enable (ME) of the control register = "1," "1" is changed to MCE.	
3	= (CU0 ~ CU3)	= flag	When current value = set value, "1" is changed to = flag. = flags of CU0 to CU3 are set in response to CU0 to CU3 set values. These flags are not cleared (or held) until = flag clear of the control register is turned to "1."	
4	> (CU0 ~ CU3)	> flag	"1" is changed to > flag only when current value is larger than set value. > flags of CU0 to CU3 are set in response to CU0 to CU3 set values.	
5	U F	Underflow flag	BIN mode	Set to "1" when current value changes from 0 to FFFF.
			BCD mode	Set to "1" when current value changes from 0 to 9999.
			This flag is not cleared until underflow flag clear (UFC) is turned to "1."	
6	O F	Overflow flag	BIN mode	Set to "1" when current value changes from FFFF to 0.
			BCD mode	Set to "1" when current value changes from 9999 to 0.
			This flag is not cleared until overflow flag clear (OFC) is turned to "1."	
7	A, B, M, ϕ	Input signal flag	Indicates ON/OFF status of input signals A, B, M and from terminals.	

⑤ Current value register

Outputs current value of the counter.