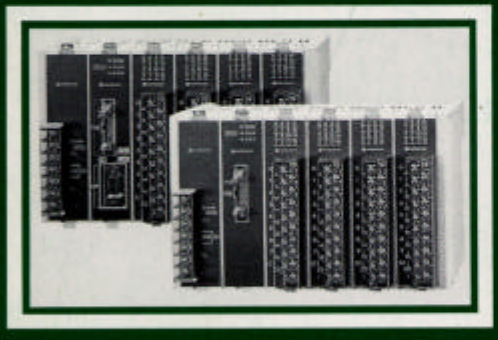
HITACHI

OPERATION MANUAL

EM-II SERIES



HITACHI PROGRAMMABLE CONTROL SYSTEM

NJI-0540(X)A

USING THIS MANUAL

Introduction

This manual describes the EM-II 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 - Principle of PC
- Chapter 3 - Input/Output and Numbers.
- Chapter 4 - Programming
- Chapter 5 - Peripheral equipment and Operation procedures
- Chapter 6 - Installation
- Chapter 7 - Maintenance
- Chapter 8 - Usage of word Input/Output modules

Additional Information

For more information on the Hitachi product line refer to these publications:

- Hitachi programmable controller EB/EM-II series protocol manual
- Graphic programmer PGM-GPE operation manual

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

6. INSTALLATION

7. MAINTENANCE

8. USAGE OF WORD INPUT/ OUTPUT MODULES

Figure 1-1 shows the system configuration of EM-II series.

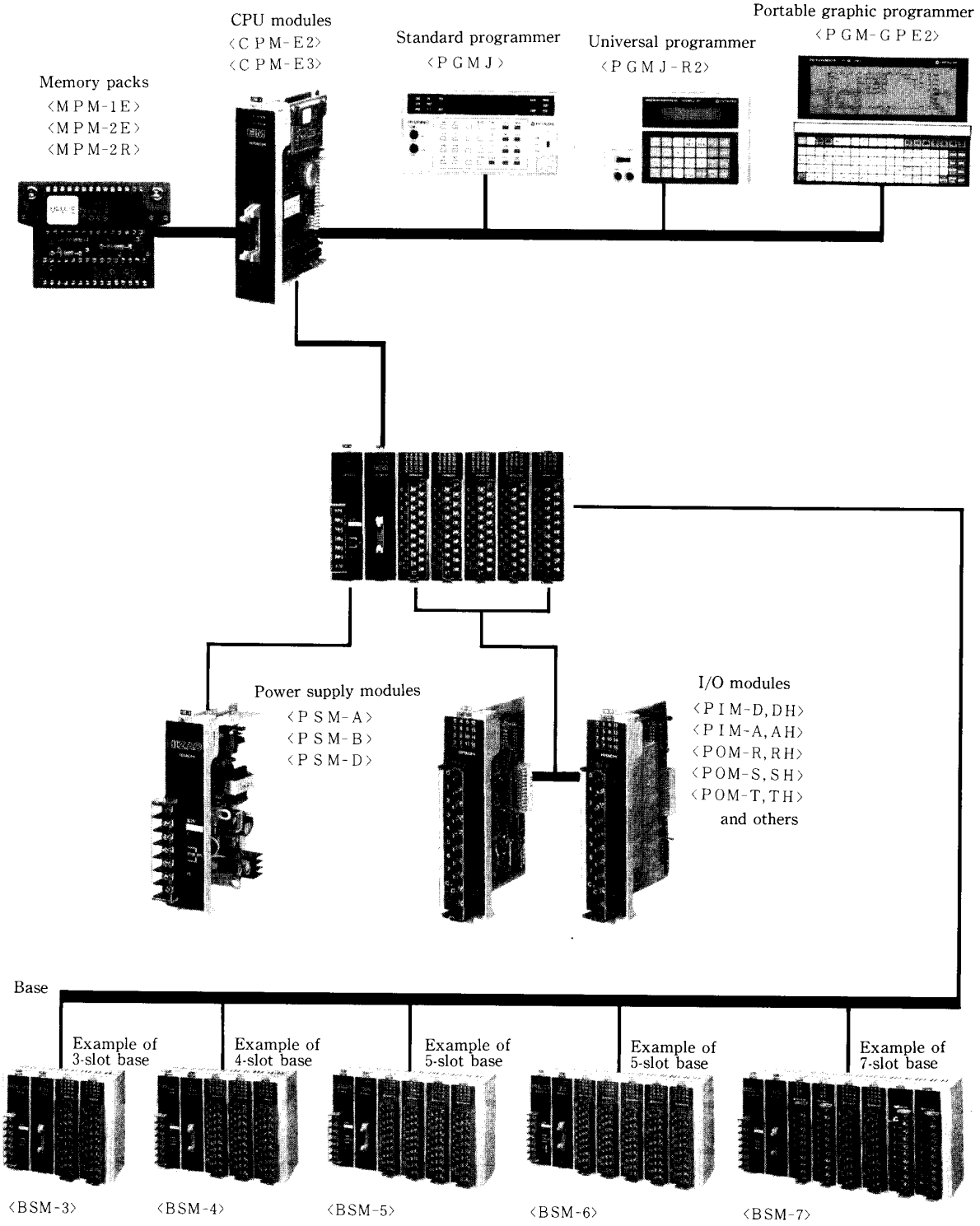


Fig. 1-1 System Configuration

[Explanation]

1. **CPU Module**

The CPU module comes in two types; CPM-E2 and CPM-E3. These modules have an upper compatibility with the preceding CPU module CPM-E. Functional differences between CPM-E and CPM-2/E3 are listed in Table 1-1.

Table 1-1 Differences from CPM-E

Type / Item	EM	EM-II	
	CPM-E	CPM-E2	CPM-E3
Processing speed	Average 5 μs/basic instruction	1.5 μs/basic instruction	1.5μs/basic instruction
Memory capacity	1949 words	3997 words	3997 words
No. of usable application instructions	44 instructions	89 instructions	89 instructions
RS-232C	Unavailable	Unavailable	Built in
Clock function	Unavailable	Unavailable	Built in
I/O link and remote I/O	Possible	Possible	Possible

2. The existing I/O modules, power supply modules, special modules, bases and memory packs are usable for CPM-E2 and CPM-E3 with no modification. The memory packs MPM-2E and MPM-2R can be used as 4K-word memories.

3. **Peripherals**

Peripherals include the standard programmer (PGMJ), universal programmer (PGMJ-R2), and portable graphic programmer (PGM-GPE2). These peripherals are all commonly usable for E, EM, EM-II and EB series. And they are capable of programming by use of the personal computer programming software (E-LADDER).

NOTE

Although the PGMJ, PGMJ-R and PBM-GPE in your possession are usable for CPM-E2 and CPM-E3, restrictions are imposed as listed in Table 1-2.

Table 1-2 Compatibility of Peripherals

Item	PGMJ	PGMJ-R		PGMJ-R2	PGM-GPE	PGM-GPE2	E-LADDER	
		Up to V:4	V:5				V:4	V:5
Programming in up to 2K words	○	○	○	○	○	○	○	○
Programming in up to 4K words	○	○	○	○	×	○	×	○
Programming in instructions compatible with EM	○	○	○	○	○	○	○	○
Printout accoding to instructions compatible with EM	–	○	○	○	○	○	○	○
Programming in new EM instructions	○	○	○	○	×	○	×	○
Printout according to new EM instructions	–	×	○	○	×	○	×	○
Decimal/hexadecimal monitor *1	○	×	○	○	×	○	×	○
CMT function in up to 2K words	○	○	○	○	○	○	–	–
CMT function in up to 4K words	○	×	×	○	×	○	–	–
ROM writer function in up to 2K words	–	○	○	○	○	○	–	–
ROM words function in up to 4K words	–	×	×	○	×	○	–	–
Time point of enhancement	–	–	Jun., '89	May, '90	–	May, '90	–	Near future

*1 Unless decimal monitoring is possible, error code in syntax check cannot be observed.

Table 1-3 Module List (1/2)

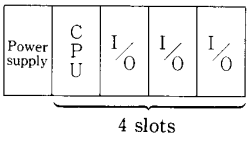
Item		Model Name	Specification	Remarks
CPU module		CPM-E2	Standart type	Completely interchangeable with CPM-E
		CPM-E3	With RS-232C interface and clock function	
Memory pack		MPM-1E	925-word EEPROM	Used as 1K-word memory
		MPM-2E	3997-word EEPROM	Used as 4K-word memory
		MPM-2R	3997-word EEPROM	
Power supply module		PSM-A	Line voltage 110/220 V AC	
		PSM-B	Line voltage 110/220 V AC, with increased output capacity	
		PSM-D	Line voltage 24 V DC	
Base		BSM-3	3 slots (Example) 	A 4-slot base modules except for the power supply.
		BSM-4	4 slots	
		BSM-5	5 slots	
		BSM-6	6 slots	
		BSM-7	7 slots	
Input module	8 input points	PIM-D	24 V DC	
		PIM-A	110/220 V AC	
		PIM-DP	24 V DC (common terminal ⊖)	
	16 input points	PIM-DH	24 V DC	
		PIM-DW	24 V DC [Removable terminal block]	
		PIM-AH	110/220 AC	
		PIM-AW	110/220 V AC [Removable terminal block]	
		PIM-DPH	24 V DC (common terminal ⊖)	
Output module	8 output points	POM-R	Relay output	
		POM-RC	Relay output, independent contacts	
		POM-S	Triac output	
		POM-T	Transistor output	
		ROM-TP	Transistor output (common terminal ⊕)	
	16 output points	POM-RH	Relay output	
		POM-RW	Relay output [removable terminal block]	
		POM-SH	Triac output	
		POM-SW	Triac output [removable terminal block]	
		POM-TH	Transistor output	
		POM-TW	Transistor output [removable terminal block]	
		ROM-TPH	Transistor output (common terminal ⊕)	
Mixed input/output	16 I/O points	PHM-DT	DC input 8 points, transistor output 8 points	
	32 I/O points	PHM-TT	TTL input 16 points, TTL output 16 points (via connector)	

Table 1-3 Module List (2/2)

Item	Model Name	Specification	Remarks
Analog module	AGM-I	Current analog input 8 points	
	AGM-O	Current analog output 4 points	
	AGM-OD	Current analog output 2 points	
	AGM-IV	Voltage analog input 8 points	
	AGM-OV	Voltage analog output 4 points	
	AGM-ODV	Voltage analog output 2 points	
Counter module	CTM	Up/down-counter, max. 10 kHz	
Remote I/O	RIOM-TM	Remote master station	Twisted pair cables
	RIOM-TL	Remote slave station	
I/O link	IOLM-T	I/0 link	
Expansion cable	CNM-01	Cable for connecting expansion unit (0.1m)	Ribbon cable
	CNM-06	Cable for connecting expansion unit (0.6m)	
	CNEB-06	Cable for connecting expansion unit (0.6m)	Round cable
Cover	CVM	Cover for empty (unused) slot	
Programmer mounting seat	PAM-E	For mounting programmer on wall	

Table 1-4 Peripherals

Item	Model Name	Specification	Remarks
Portable graphic programmer	PGM-GPE2	Liquid crystal type graphic programmer	
Standard programmer	PGMJ	With audio cassette interface	
Universal programmer	PGMJ-R2	With audio cassette interface, ROM writer function and RS-232C serial port	
Software package for personal computer input	E-LADDER (IBM)	Software package for IBM 5150/5160	

Figure 1-2 shows the name of each external part in case of a 5-slot base.

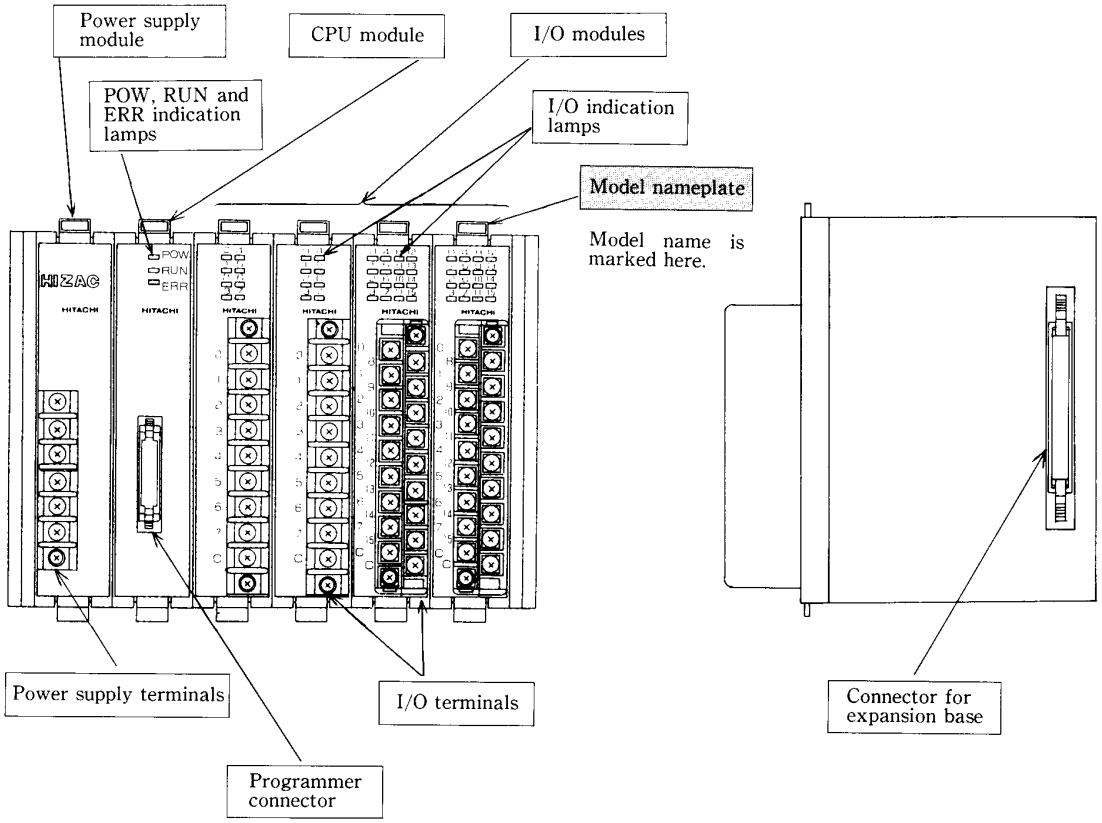


Fig. 1-2 Name of Earch External Part

Each of 3, 4, 6, and 7-slot bases has a different number of slots than the 5-slot base. The base is commonly used for basic and expanded configurations. For expansion, an I/O module is to be mounted in the CPU module slot of expansion unit.

(1) **Basic specifications**

Basic specifications are listed in Table 1-5.

Table 1-5 Basic Specifications

Item			CPM-E2	CPM-E3
Control specifications	Control system		Stored program cyclic processing	
	Processing speed		1.5μs/basic instruction	
	Program capacity		925 words with EEPROM (MPM-1E)	
			3,997 words with EEPROM (MPM-2E)	
			3,997 words with EPROM (MPM-2R)	
Processing function	Basic instruction		12 kinds (ORG, STR, AND, OR, STR, AND STR, OUT, etc.)	
	Application instruction		20 kinds (edge detection, step, master control, jump, etc.)	
	Arithmetic instruction		69 kinds (word load, word out, arithmetic calculations, comparison, etc.)	
Input/output processing specifications	Input/output allocation		Free location	
	No. of external input/output points		Max. 160 points with 16 I/O modules	
			Max. 320 points with 32 I/O modules (PHM-TT)	
	No. of internal output points	Non-retentive at power failure	256 points	
		Retentive at power failure	256 points (NOTE 1)	
		Special function	12 points + 4 words	
	Timer/counter	Counting system	Addition	
		No. of points	96 points	
		Time base	0.01, 0.1, 1 sec	
		Preset value	4 digits (max. 10 points), 3 digits (NOTE 2)	
	Kind of external input		24 V DC, 110/220 V AC, analog	
	Kind of external output		Relay, transistor, triac, analog	
	Operation control input		Programmable (a single input in a input module specifiable)	
	RUN contact output		Programmable (a single output in a output module specifiable)	
Peripheral function	Peripherals		PGMJ, PGMJ-R2, PGM-GPE2	
	Monitor function		Bit monitor and word monitor	
Communication function	Personal computer link		Via PGMJ-R2	Direct hookup to personal computer (RS-232C built in)
	I/O link		I/O link module (IOLM-T)	
	Remote I/O		Remote I/O module (RIOM-TM, TL)	
Clock function				Calendar clock built in
Self-diagnosis function			Watchdog timer, sum check, undefined instruction check	

NOTES: 1 The internal output retentive at power failure and the current value of timer/counter are backed up with a capacitor. Backup is possible for 2 weeks (at 25°C). When using the calender clock, the number of internal output points is reduced to 240 in case of the CPM-E3.

2 10 points of T/C 0 to 9 are presettable in 4 digits.

(2) General specifications

Table 1-6 General Specifications

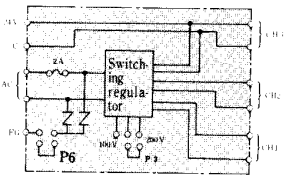
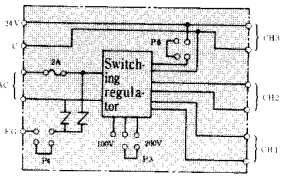
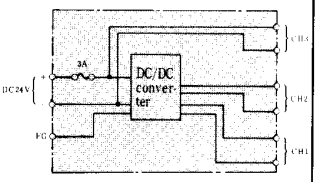
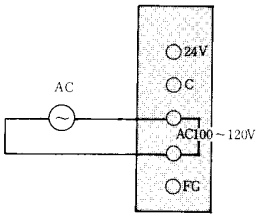
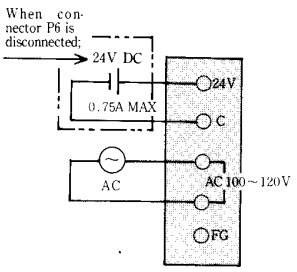
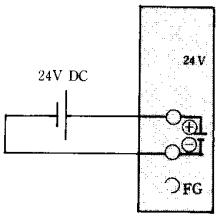
Item	Specifications
Dielectric strength	1,500 V AC for 1 min. between input/output terminals (including power terminal) and ground terminal (NOTE 1)
Insulation resistance	20 MΩ or more for 1 min between input/output terminals (including power terminal) and ground terminal when measured with 500 V DC megger (NOTE 1)
Operating temperature	0 to 55℃
Storage temperature	− 10 to 75℃
Operating humidity	20 to 90%(non-condensing)
Strage humidity	10 to 90%(non-condensing)
Vibration resistance	Conforms to JIS C 0911 IIB, 3rd class on condition that vibration with frequency 10 to 55 Hz and amplitude 0.5 mm is applied for 2 hours in each of X, Y and Z directions
Shock resistance	Conforms to JIS C 0912on condition that shock of 10G is applied twice in each of X, Y and Z directions
Noise resistance	Noise voltage 1,500 Vp-p, pulse width 1 μs (Measurement by Hitachi method with noise simulator)
Environment	Must be free from corrosive gas and dust.
Altitude	2,000 m or less
Grounding	100 Ω max.

NOTE: 1 A varistor for suppressing lightning surge is connected to the power supply terminal. Therefore, the connector P3 in the power supply module must be separated when testing dielectiric strength or insulation resistance of the power supply terminal.

(3) Specifications of power supply modules

The specifications of each power supply module are listed in Table 1-7.

Table 1-7 Specifications of Power Supply Modules

Item \ Model		PSM-A	PSM-B	PSM-D
Line voltage	Rated voltage	100 V/110/120 V AC, 200/220/240 V AC (110 V AC system and 220 V AC system switchable with connector P3)		24 V DC
	Allowable fluctuation	85~132 V AC, 170~264 V AC		19.2~30 V DC
Frequency	Rated frequency	50/60 Hz		–
	Allowable fluctuation	47~63 Hz		–
Input current		0.6 A or less		1.6 A or less
Output current	CH1 (5 V)	1 A (for CPU, Programmer)	1.7 A (for CPU, Programmer)	1 A (for CUP, programmer)
	CH2 (24 V)	0.3 A (for output module)	0.5 A (for output module)	0.3 A (for output module)
	CH3 (24 V)	0.45 A (for input module)	0.25 A (for input module)	1 A (for input module)
Circuit diagram				
External wiring				

[Explanation]

1. Each power supply module receives an AC or DC primary power supply and outputs the determined system power supply to the CPU, programmer and input/output modules. Its output consists of 3 channels; CH1 (5 V) for programmer, CH2 (24V) for output module and CH3 (24 V) for input module. The maximum output current is restricted as shown is the above table. The current consumption of each module is determined in the specifications below. The system must be configured so that the total current consumption does not exceed the maximum output current of the power supply module.
   - The average current consumption of the CPU module is 110 mA (via CH1), while that of the programmer is 260 mA (via CH1). For current consumption of other modules, refer to each table of specifications.

2. The PSM-A and PSM-B select the 110 or 220 V AC system by means of connector P3.
   These modules have been factory-set to the 220 V AC system. For 110 V system, switch over the connector to the 110 V side and attach the furnished voltage nameplate.

3. The PSM-A and PSM-B incorporate a varistor for protection against lightning surge. Therefore, the internal connector (P6 on PSM-A, P4 on PSM-B) must be separated before a dielectric strength or insulation resistance test. Otherwise, the varistor might be broken.

4. CH3 is also used for power supply to the sensor. Total output current in this channel must be limited to 0.45 A max. with the PMS-A, and to 0.25 A max. with the PSM-B.

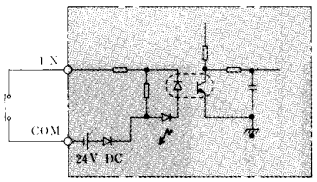
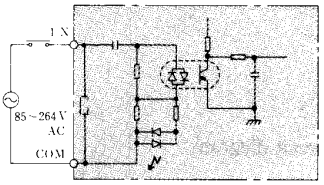
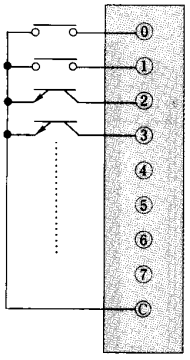
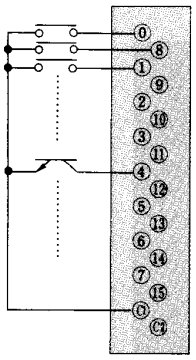
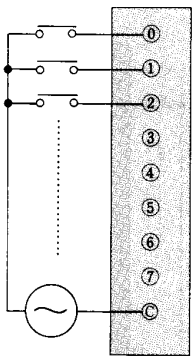
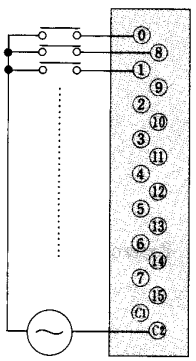
NOTE

The PSM-B allows its channel 3 to receive power from an external switching power supply when the internal connector P6 is separated. Utilize this method in case the CH3 current is inadequate because there are many input modules connected.

(4) Specifications of input modules

The specifications of each input module are listed in Table 1-8.

Table 1-8 Specifications of Input Modules

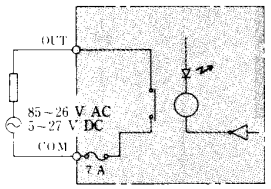
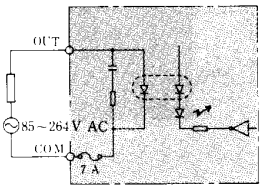
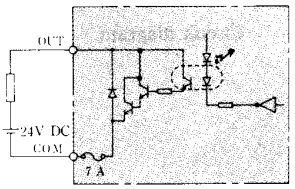
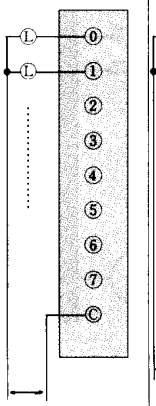
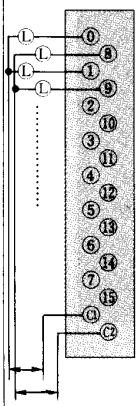
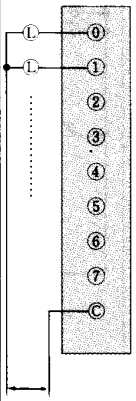
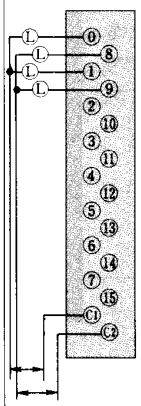
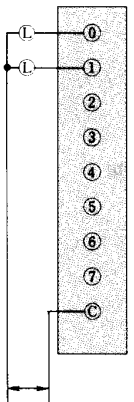
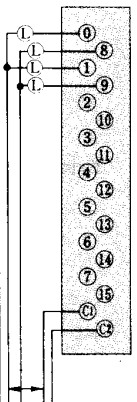
Item \ Model		P I M − D	P I M−D H	P I M − A	P I M−A H
Input specification		DC input		AC input	
Nominal voltage		24V DC		AC110V／220V	
Input voltage		21.6～26V DC		85～264V AC, 50／60Hz	
Input current		9 mA (when input and common terminals are short-circuited)		7 mA (AC110V, 50Hz)	
Operational specification	O N	19 V DC or more (resistance 300 Ω or less)		85 V AC or more	
	OFF	7 V DC or less (resistance 200 kΩ or more)		30 V AC or less	
Input delay time	O N→OFF	4 m s or less		16m s or less	
	OFF→O N	4 m s or less		16m s or less	
No. of input points		8 points/module	16 points/module	8 points/module	16 points/module
Common input connection		8 inputs/common terminal		8 inputs/common terminal	
Polarity		Common terminal ⊖		—	
Isolation method		Photocoupler		Photocoupler	
Current consumption (average) (NOTE)	C H 1	0.5 mA + (no. of input ON points) × 0.5 mA		1 mA	
	C H 2	0 mA		0 mA	
	C H 3	(No. of input ON points) × 9 mA		0 mA	
Circuit diagram					
External wiring			 ○ C1 and C2 are internally connected.		 ○ C1 and C2 are internally connected.

NOTE. This represents the power consumption of each module. The total current consumption of each channel must not exceed the maximum load current of the power supply module.

(5) Specifications of output modules

The specifications of each output module are listed in Table 1-9.

Table 1-9 Specifications of Output Modules

Item \ Model	POM-R	POM-RH	POM-S	POM-SH	POM-T	POM-TH
Output specification	Relay output		Triac output		Transistor output	
Nominal voltage	110/220 V AC, 24 V DC		110/220 V AC		24 V CD	
Output voltage	85~264 V AC, 5~27 V DC		85~264 V AC		5~27 V DC	
Max. load current — 1 circuit	2 A		1 A		0.5A	
Max. load current — 4 circuits	2 A		2 A		1.25A (Note)	
Max. load current — 8 circuits	4 A		4 A		2.5A (Note)	
Min. load current	10mA (5 V DC)		50mA		10mA (5 V DC)	
Max. leakage current	—		3mA (220 V AC)		0.1mA (24 V DC)	
Max. rush current	6 A (100m s)		20A (20ms)		3 A (20ms)	
Max. output delay time — ON→OFF	10ms		11ms		1 ms	
Max. output delay time — OFF→ON	10ms		11ms		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) — CH 1	0.2mA + (no. of output ON points) × 0.2mA		0.2mA + (no. of output ON points) × 0.2mA		0.2mA + (no. of output ON points) × 0.2mA	
Current consumption (average) — CH 2	(No. of output ON points) × 10mA		(No. of output ON points) × 10mA		(No. of output ON points) × 10mA	
Current consumption (average) — CH 3	0 mA		0 mA		0 mA	
Circuit diagram						
External wiring						

Note: Since four-element transistor devices are used, max. load currents are limited for each group of terminals No. 0 to 3, 4 to 7, 8 to 11 and 12 to 15. Operation is unallowable beyond the maximum load current.

(6) Specifications of source type input/output modules

Table 1-10 Source Type Input Module

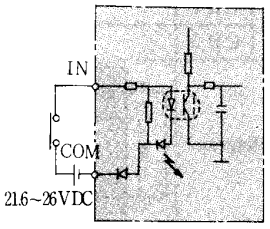
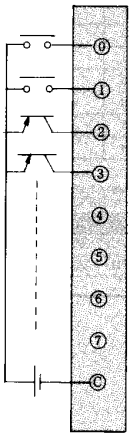
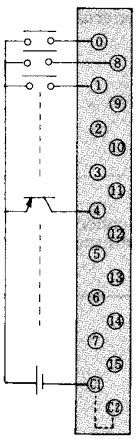
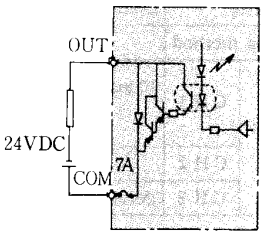
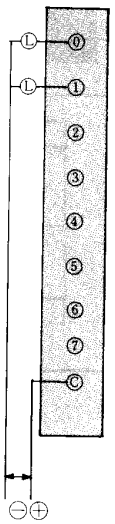
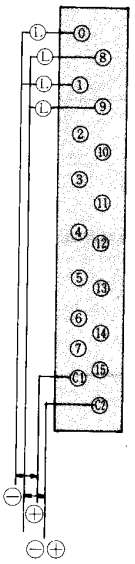
Item \ Model		PIM-DP	PIM-DPH
Input specification		DC input	
Nominal voltage		24 V DC	
Input voltage		21.6~26 V DC	
Input current		Approx. 9mA/24 V DC (inpedance approx. 2.7 kΩ)	
Operating voltage	ON	19 V DC or more (resistance 300 Ω or less)	
	OFF	7 V DC or less (resistance 200 Ω or more)	
Max. input delay time	ON→OFF	4 msec	
	OFF→ON	4 msec	
No. of input points		8 points	16 points
Common input connection		8 points/common terminal	
Polarity		Common terminal ⊖	
Isolation method		Photocoupler	
Circuit diagram			
Extenal wiring			 C1 and C2 internally connected

Table 1-11 Source Type Output Module

Item \ Model		POM-TP	POM-TPH
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		10mA (24 V DC)	
Max. leakage current		0.1mA (24 V DC)	
Max. ruch current		3A (20msec)	
Max. output delay time	ON→OFF	1msec	
	OFF→ON	1msec	
No. of output points			
Common output connection		8 points/common terminal	
Polariy		Common terminal ⊕ (source type)	
Isolation method		Photocoupler	
Circuit diagram			
External wiring		 DC power supply	 DC power supply

(7) **Specifications of hybrid modules**

Table 1-12 and 1-13 list the specifications of I/O hybrid module and TTL I/O hybrid module, respectively.

Table 1-12 I/O Hybrid Module

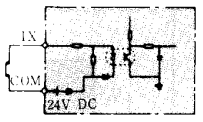
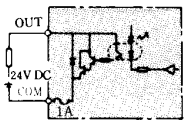
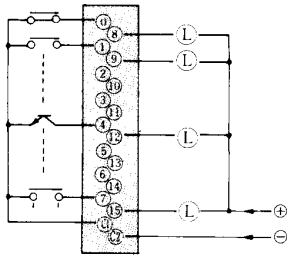
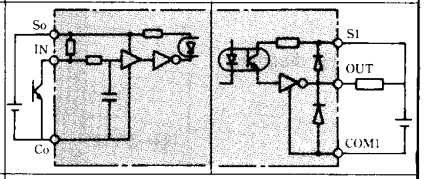
Item \ Model		PHM-DT	
I/O specification		DC input	Transistor output
Nominal voltage		24 V DC	24 V DC
Permissible voltage range		21.6~26 V DC	5~27 V DC
Input current		9 mA	—
Operational specification	ON	Resistance 300 Ω or less	—
	OFF	Resistance 200 kΩ or more	—
Max. load current	1 circuit	—	0.5A
	4 circuits	—	1.25A
	8 circuits	—	2.5A
Max. leakage current		—	0.1mA (24 V DC)
Max. rush current		—	3 A (20ms)
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 connection		8 points/common terminal	8 points/common terminal
Polarity		Common terminal ⊖	Common terminal ⊖
Isolation method		Photocoupler	Photocoupler
Current consumption (average)	C H 1	10 mA + (no. of input ON points) × 9 mA + (no. of output ON points) × 8 mA	
	C H 2	0 mA	0 mA
	C H 3	(No. of input ON points) × 9mA	0 mA
Circuit diagram			
External wiring			

Table 1-13 TTL I/O Hybrid Module

Item \ Model		PHM-TT	
I/O specification		TTL input	TTL output (open collector)
I/O voltage		4~27VDC	4~27VDC
Input current		6mA (5 V DC)	—
Input voltage	ON	1.5 V DC or less (5 V DC)	—
	OFF	3.5 V DC or more (5 V DC)	—
Max. load current		—	20 mA/point
Max. leakage current		—	50μ A
Max. delay time	ON→OFF	1 ms	1 ms
	OFF→ON	1 ms	1 ms
No. of I/O points		16 points/module	16 points/module
Common connection		16 points/common terminal	8 points/common terminal
Polarity		Common terminal ⊖	Common terminal ⊖
Isolation method		Photocoupler	Photocoupler
I/O indication		None	None
Current consumption (average)	C H 1	(No. of output ON points) × 5 mA + 30 mA	
	C H 2	0 mA	
	C H 3	0 mA	
Circuit diagram			
External wiring		Pin layout of 40-pin flat cable connector	
Connector for external wiring		Exclusive connector (made by Hirose Denki) Socket: HIF3C-40D-2. 54C	

Pin No.	Signal	Pin No.	Signal	Pin No.	Signal	Pin No.	Signal
1	COM0	21	NC	2	COM1	22	COM2
3	S0	23	NC	4	S1	24	S2
5	IN0	25	IN8	6	OUT0	26	OUT8
7	1	27	9	8	1	28	9
9	2	29	10	10	2	30	10
11	3	31	11	12	3	32	11
13	4	33	12	14	4	34	12
15	5	35	13	16	5	36	13
17	6	37	14	18	6	38	14
19	7	39	15	20	7	40	15

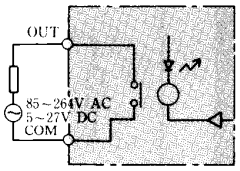
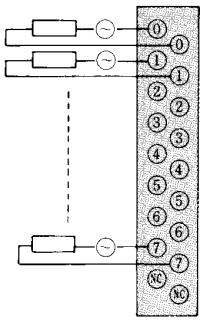
Connector pin	Cable dia.
HIF3-2226SC	AWG22 to 26
HIF3-2428SC	AWG24 to 28

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

(8) **Specifications of independent contact relay output module**

The specifications of independent contact rely output module are listed in Table 1-14.

Table 1-14 Specifications of Independent Contact Relay Output Module

Item \ Model		POM-RC
Output specification		Relay output
Nominal voltage		100/200 V AC, 24 V DC
Output voltage		85~264 V AC, 5~27 V DC
Max. load current	1 circuit	2 A
Max. rush current		6 A (100ms)
Max. output delay time	ON→OFF	4 ms
	OFF→ON	5 ms
No. of output points		8 points
Common output connection		1 point/common terminal
Isolation method		Relay
Current consumption (average)	CH1	0.2 mA + (no. of outpit ON points) × 0.2 mA
	CH2	(No. of output ON points) × 10 mA
	CH3	0 mA
Circuit diagram		
External wiring		

(9) **Specifications of counter module**

The specifications of the counter module are listed in Table 1-15.

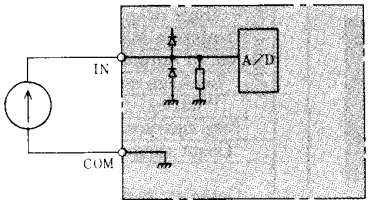
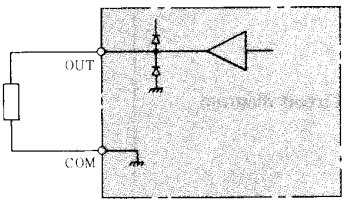
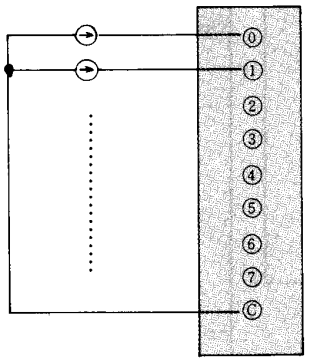
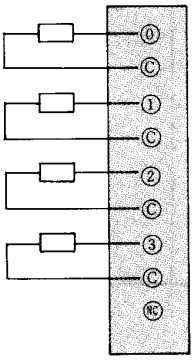
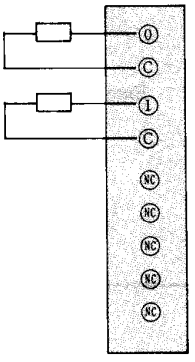
Table 1-15 Specifications of Counter Module

Item			Specifications
Input specifications	Count puls frequency		MAX. 10kHz
	Input pulse voltage level	ON	0~2 V DC
		OFF	5~12 V DC
	Count pulse width		MIN. 20μ sec
	Marker pulse width		MIN. 20μ sec
	Input impedance		Approx.10 kΩ
	Isolation method		Photocoupler
	No. of pulse input points		3 points (A, B, M)
	Polarity		Common terminal ⊖
	2-phase imput pulse	Count-up (addition)	A, B: ON/OFF, 90° B delayed 90° from A
		Count-down (subtraction)	A, B: 90° B advanced 90°from A
	Power supply for external input device		12 V DC ± 10%, 50 mA (can be supplied to external device)
Output specifications	Output method		DC10~27V
	Load current		Max. 0.5 A/circuit, max. 1.25 A/4 circuits
	Output method		Transistor (open collector)
	Min. load current		1 mA
	Output delay time	ON→OFF	MAX. 1 msec
		OFF→ON	MAX. 1 msec
	Voltage drop at ON		MAX. 1.5V (0.5A)
	Isolation method		Photocoupler
	No. of output points		4 points (OUT, OUT1, OUT2, OUT3)
	Leakage current		MAX 0.1mA
	Polarity		Common ⊖
	Power supply input for output		10 to 27 V DC, 50 mA (external supply to module)
Count range			0~9999
Counting method			○2-phase pulse counting (up/down) ○Single-phase pulse and inverted pulse counting (Selectable between single phase and 2phases)
Output			○1 point/1preset value (open collector) ○Output held when preset value = count value selectable ○Output when present value < count value selectable
Marker			1 point (direct resetting of count value)
Operational indication			Output and pulse input imdicated
Register			○Count register ○Preset value (CU0, CU1, CU2, CU3) register
Functions			○Count value reset ○Preset value read ○Preset value write ○Status read A-phase pulse ON/OFF status B-phase pulse ON/OFF status Marker ON/OFF status Preset value = count value (latch) Preset value < count value Overflow Flag Underflow flag
Noise resistance			Noise voltage 500 Vp-p when measured by our company method with noise simulator
Insulation resistance			20 MΩ or more between external terminal and ground terminal (FG)
Dielectric strength			500 V DC for 1 min bitween external terminal and ground terminal
Vibration resistance			Conforms to JIS C0911 HB, 3rd class on condition that vibration with frequincy 16.7Hzand amplirude 3mm is applied in each of X,Y and Z directions.
Shock resistance			Conforms to JIS C0912 on condition that shock of X, Y and z directions
Operating temperature			0~55℃
Operating humidity			30 to 90% RH (non-condensing)
Storage temperature			−10~65℃
Environment			Must be free from excessive corrosive gas, salinity and iron powder.
Current consumption	CH1		200mA MAX.
	CH2		0 mA
	CH3		160 mA max. when supplying about 50 mA to external imput device (sensor)
			110 mA max. without currint supply to external input device (sensor)

(10) Specifications of analog current modules

The specifications of each analog current module are listed in Table 1-16.

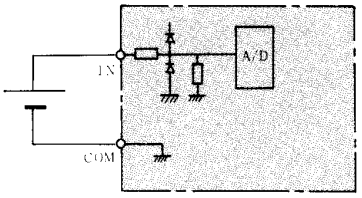
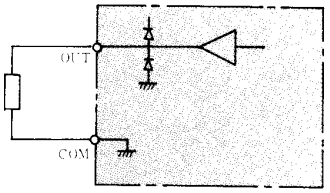
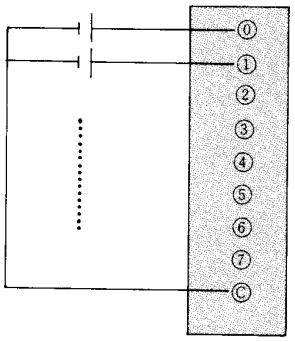
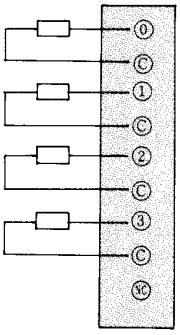
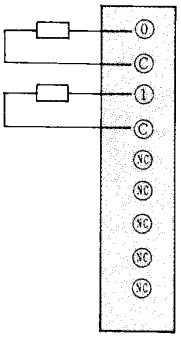
Table 1-16 Specifications of Analog Current Modules

Item \ Model		AGM-I	AGM-O	AGM-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 poionts		8 points	4 points	2 points
Isolation method		Photocoupler (not isolated from DC input)	Photocoupler (not isolated from DC input)	
Isoration between imputs		Not provided	Not provided	
Current consumption (average)	CH 1	25mA	50mA	50mA
	CH 2	0mA	0mA	0mA
	CH 3	60mA	250mA	140mA
Circuit diagfam				
External wiring				

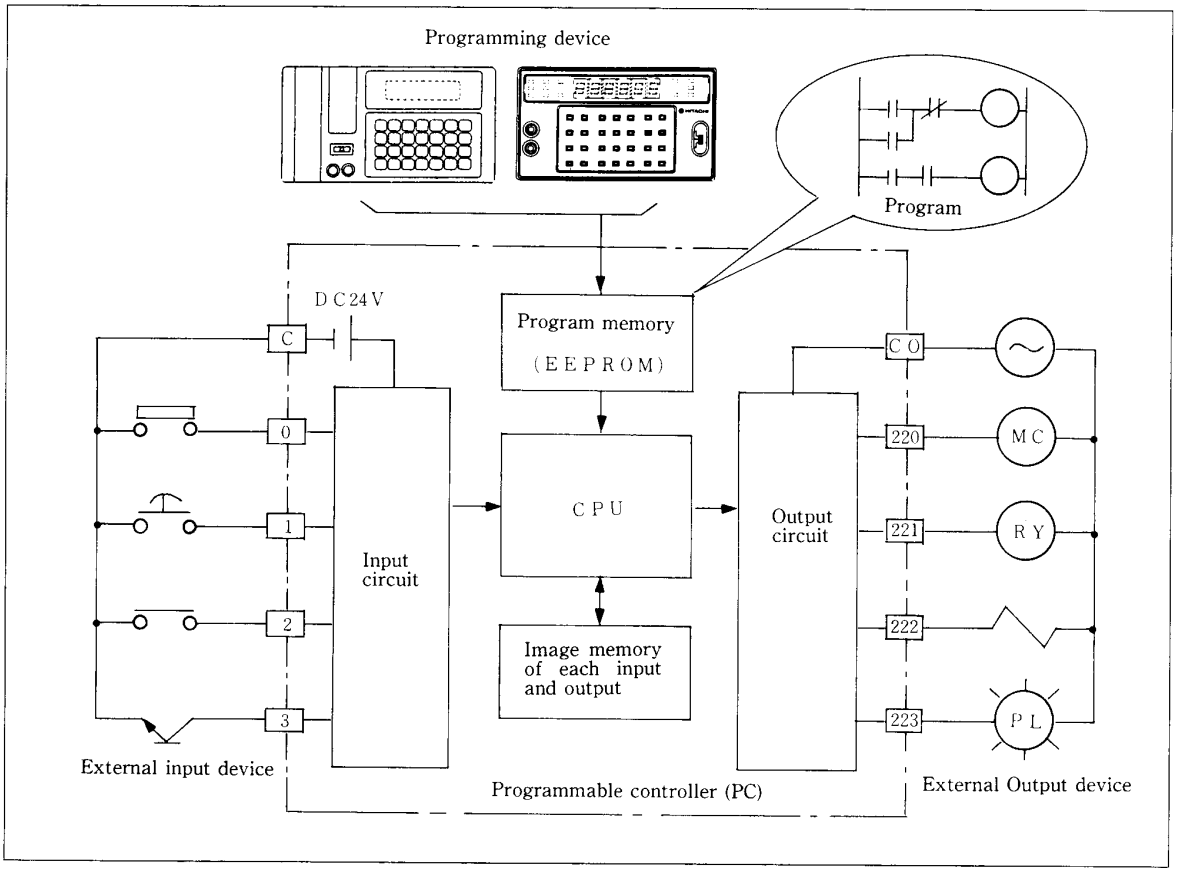
(11) Specifications of analog voltage modules

The specifications of each analog voltage module are listed in Table 1-17.

Table 1-17 Specifications of Analog Voltage Modules

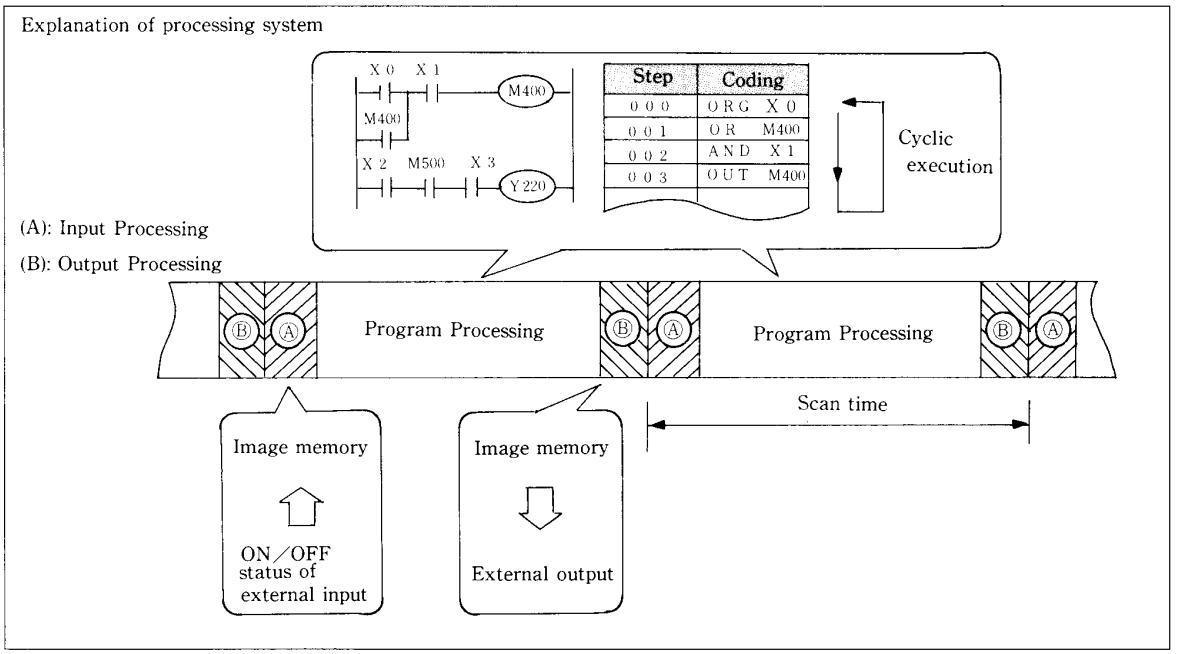
Item \ Model		AGM-IV	AGM-OV	AGM-ODV
I/O specification		Voltag range	Analog voltage output	
Voltage range		0~10 V DC	0~10 V DC	
Input impedance		100 kΩ	–	
Load impedance		–	10 kΩ or more	
Resolution		8 bits	8 bits	
Conversion time		1 ms	1 ms	
Overall accuracy		1% + 1 bit	1 %	
No. of points		8 points	4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

PRINCIPLE OF PC



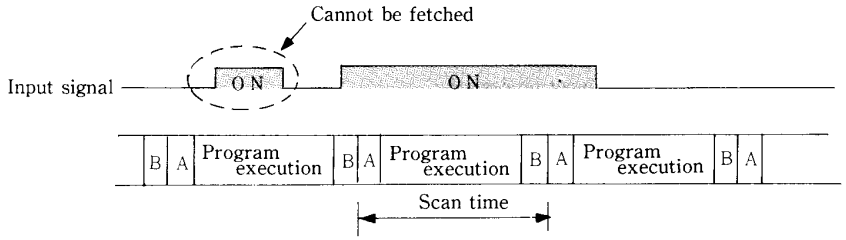
[Explanation]

1. **The programmable controller (PC) consists of CPU, program memory, image memory of each input and output, input circuit, output circuit and power supply.**
   (1) The CPU is composed of a microprocessor which excutes logic and arithmetic operations, and the system software which controls PC itself.
   (2) The program memory is used to store a user-defined sequence program (ladder diagram). Program is to be generated by using the exclusive programming device or personnal computer.
   In the E/EM series, EEPROM is used for program, so a stored program will not be lost after the PC power supply is turned off. The program can be modified easily if necessary.
   (3) The image memory of each component contains data including ON/OFF status of input/output and current value of timer/counter. These data change along with program execution.
   (4) The input circuit composes an interface to the external input devices (such as pushbutton switches, limit switches and proximity switches). It is electrically isolated by photocouplers.
   (5) The output circuit composes an interface to the external output devices (such as electromagnetic contactors, valves and lamps).



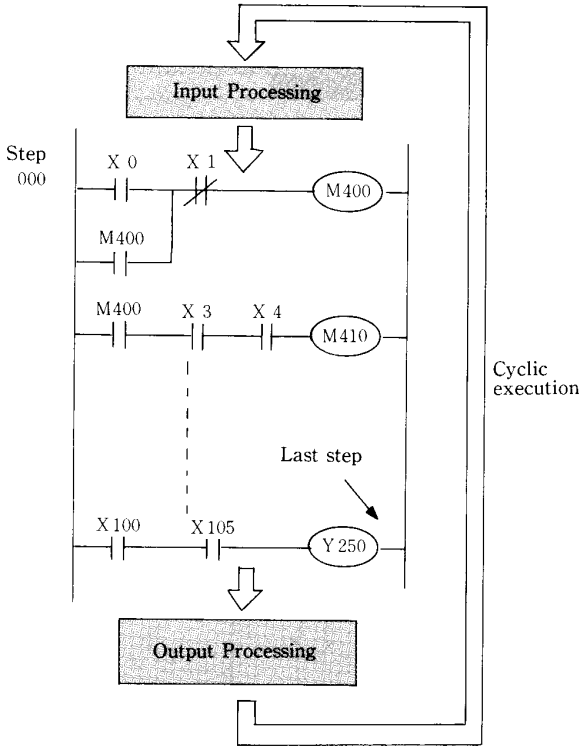
[**Explanation**]

1. **Scan time**
   The PC sequentially executes the written program (stored program) from its first step to the last step, then returns to the first step again and repeats the operation (cyclic execution). The duration of a single cycle of this operation is called the scan time.
2. **Input operation**
   The ON/OFF status of external input is fetched in the image memory. Even if the ON/OFF status of external input changes during program execution, the input status in the image memory remains unchanged. The status change can be read only during input processing for the next scan. So an input signal can be fetched only when its duration is longer than the time for a single scan. For fetching an input signal with a shorter duration than above, external interruption input or refresh instruction is usable.



3. **Program execution**
   A program runs sequentially from its start step (step 0000) to the last step according to the written instructions. The status of external output, internal output, etc. changes sequentially on the image memory along with program processing.

[Example]



4. Output processing

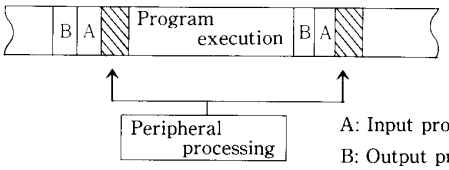
The ON/OFF status of external output on the image memory is sent to the output circuit.

5. I/O batch processing

Reading the status of all external input signals at the beginning of a scan and outputting the resulting signals to an external device at the end of this scan is called I/O batch precessing. (Some PC's use direct processing in which the external inputs are read sequentially and the result is output to the external device also sequentially.)

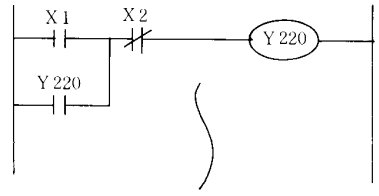
I/O batch processing does not cause a change in the ON/OFF status of external input and output during a scan. This makes the timing check on a program easy. Therefore, this system is widely used on small-scale PC's. The EM-II employs this system.

6. Peripheral processing



Peripheral processing (communication) with programming device, etc.) is to be made for only 1 ms during the time between program execution and input processing.

A: Input processing
B: Output processing



Code list

Step	Instruction	
	Instruction word	I/O No.
000	ORG	X1
001	OR	Y220
002	AND NOT	X2
003	OUT	Y220

X, Y and M representing I/0 classification cannot be keyed in. Key in numeral (s) alone.

Instruction = Instruction word + I/O No.

Instruction
- 1-word instruction = 1 step ··· Basic instruction, etc.
- 2-word instruction, etc. = 2 steps ··· Arithmetic instruction, etc.

2

[Explanation]

1. Instruction

(1) An instruction is a combination of an instruction word (basic instruction, application instruction or arithmetic instruction) and the I/O number (external input, external output, internal output, timer, counter, constant or the like). Some instruction words do not require an I/O number.

AND ↑ Instruction word

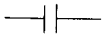
M990 ↑ I/O No.

(2) A single word occupies one step.
There are two kinds of instructions; one-word (16-bit) instruction and two-word (32-bit) instruction. Because the capacity of standard EM-II program memory is 3,997 words,up to 3,997 one-word instructions are programmable.

2. I/O number

A code representing the I/0 classification is prefixed to each I/0 number.

I/0 classification, I/O No.: X 100



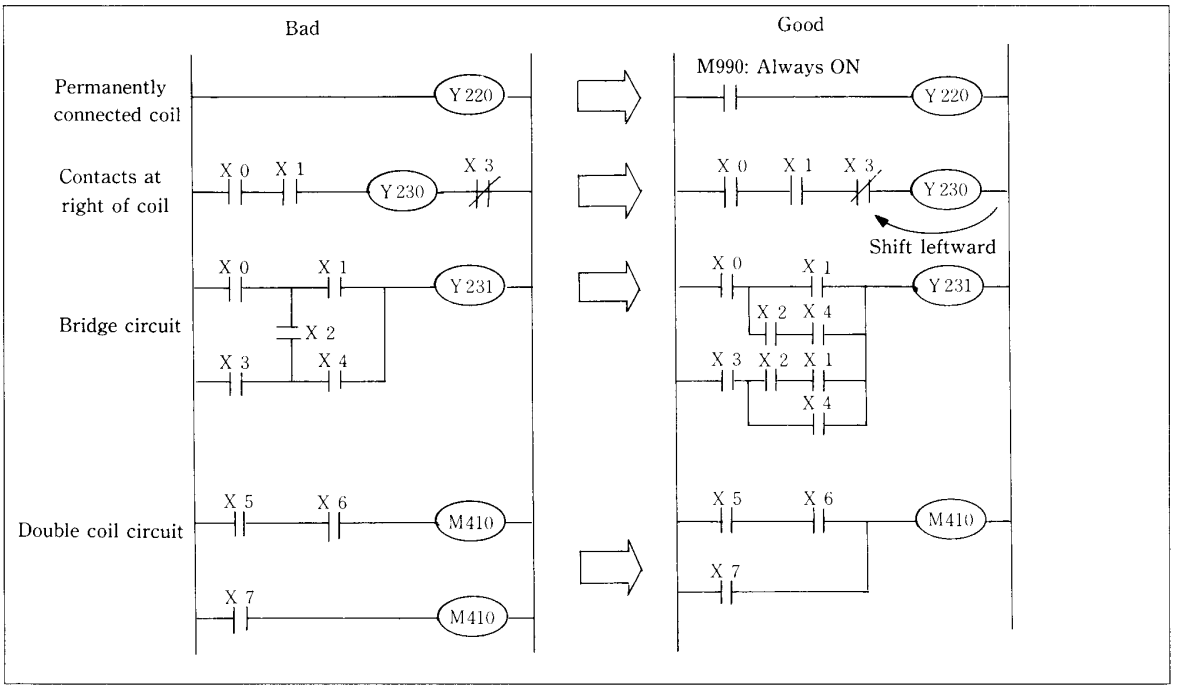
[I/O classification code]

X: External input
Y: External output
M: Internal output
} These can be identified by I/0 number, so they need not be keyed in when using a programmer.

T/C: Timer, counter

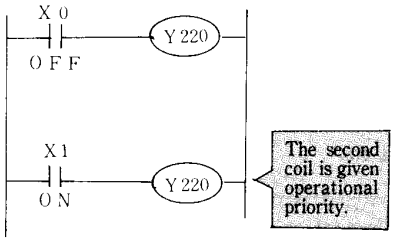
No code: Constant, number of jump, etc.

The I/O number is determined by the assignment table (described later) so that numbers used for X, Y and M are not used twice, and the I/O classification (X, Y and M) need not be keyed in when using a programmer. However, X, Y and M are written in this manual so that the reader can easily recognize the I/O classification in the sequence program.



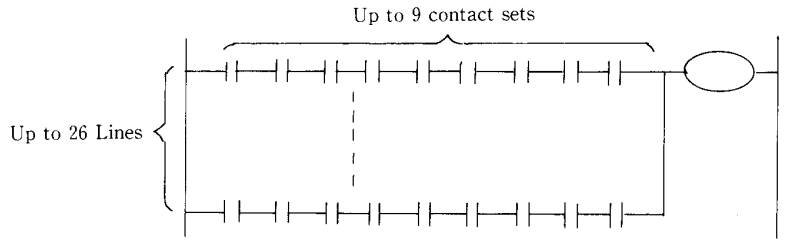
[**Explanation**]

1. **Permanently connected coil**
   An output coil cannot be connected directly to the left bus. It must be connected via the contacts of special internal output (M990) which are always closed.
2. **Contacts at right of coil**
   Although the contacts of thermal relay are connected at the right of output coil in the relay sequence, it is unallowable in the PC sequence. In case such a connection is required, the contacts must be connected at the left of the coil.
3. **Bridge circuit**
   Vertical disposition of any contacts cannot be programmed. So connect the contacts in the horizontal direction.
4. **Double coil**
   Do not use the same output coil more than once, otherwise a double coil error (E. display) will be detected during the syntax check. However, operation will continue even if a double coil error occurs, and the output signal of the second coil will be used in the subsequent steps.



NOTE
Coil following the FUN02 (IF) or FUN;3 (IFR) is not treated as a double coil error.

5. Restrictions on number of serial and parallel contacts



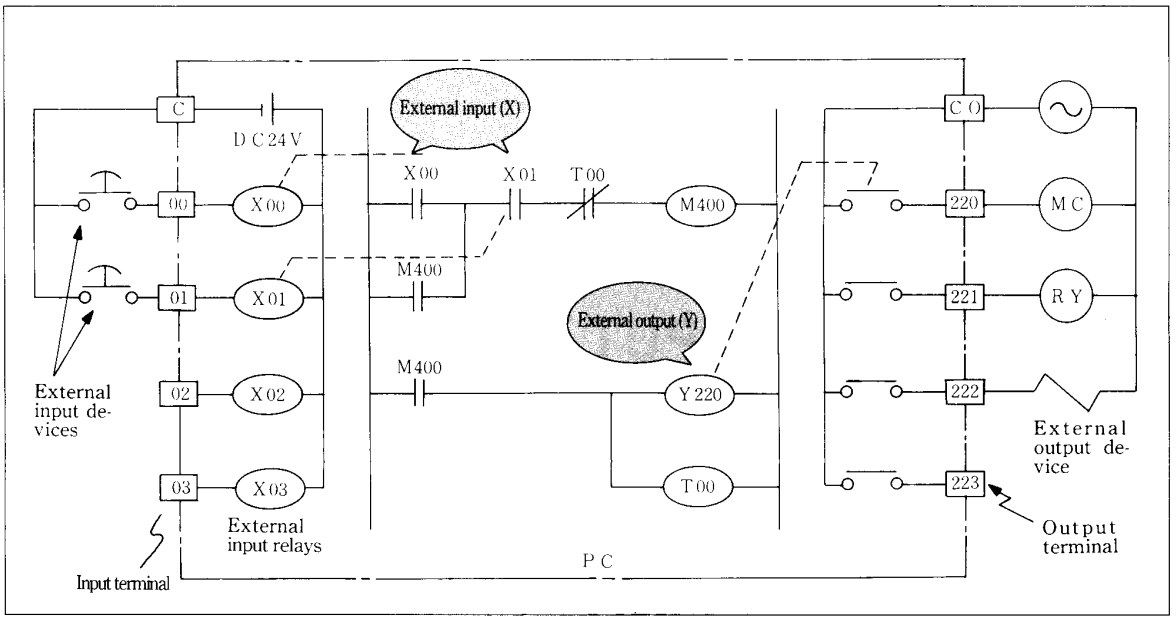
(1) For entering a program with the portable graphic programmer (PGM-GPE2), the number of contact sets is restricted to 9 on each of 26 lines at maximum.
(2) For printing out data using the universal programmer (PRGMJ-R2), the number of contact sets is restricted to 8 on each of 26 lines at maximum.
(3) Although there is no restriction imposed in either vertical or horizontal direction when using the standard programmer (PGMJ) or universal programmer (PGMJ-R2), it is recommended to avoid using contacts beyond 8 sets on each line and beyond 26 lines in consideration of (1) and (2) above.

Q What is the difference between the PC and relay panel?

A The PC is more compact, has higher performance and more flexibility and is easier to operate than a relay panel.

Item \ System		Relay system		PC control
Function	△	Complicated control is enabled by using many relays.	○	Control can respond to any complication through programming.
Modification of control data	×	Impossible except by rewiring.	○	Possible freely through program modification.
Reliability	△	No problem in normal use. However, poor contact may occur and the servise life is limited.	○	Highly reliable because semiconductors are used in key components.
Universality	×	Complete device serves for only one purpose.	○	Usable for any control through programming.
System expandability	△	Difficult because modification is required.	○	Freely expandable within capacity.
Ease of maintenance	△	Periodic maintenance and replacement of service parts are required.	○	Repair is possible inside each unit.
Necessary technical understanding	○	Popular, widely known, simple and easy to understand	△	Programming software rules must be learned.
Equipment size	△	Usually large	○	Remains compact for even complicated and sophisticated control
Design and manufacturing periods	×	Many drawings must be prepared, and a long time is needed for arranging parts and performing assembly test.	○	Design is easy even for complicated control. Manufacture can be completed in a shorter time period. Hardware is usable for general purposes (ready-made products).

Sign: ○ Very good △ Good × Poor



[Explanation]

1. **External input (X)**

   Input sensors, such as limit switches, pushbutton switches, proximity switches and photoelectric switches are external input devices of the PC. They are connected to the input terminals of the PC and drive the external input relay (X) in the PC.

   (The ON/OFF status of each external input device is fetched in the image memory.)

   External input relay is referred to as an external input (X) hereinafter. The external input (X) has many normally open contacts ("a" contacts) and normally closed contacts ("b" contacts). They are used for generating the sequence in the PC.

2. **External output (Y)**

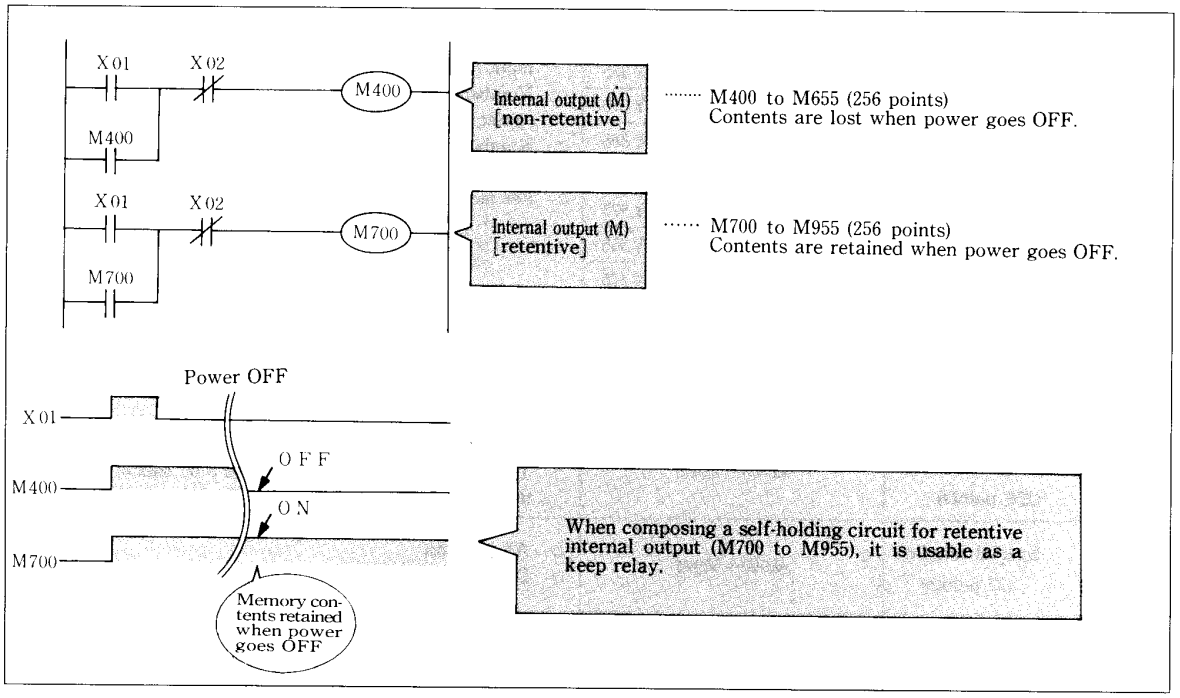
   Electromagnetic contactors, valves, indicator lamps, etc. , are external output devices of the PC. These devices are connected to the PC output terminals and driven via the contacts of external output relays in the PC. External output relay is referred to as an external output (Y) hereinafter.

   The external output (Y) also has many normally open contacts ("a" contacts)and normally closed contacts ("b") contacts). They are used for generating the sequence in the PC.

3. **I/O number assignment**

   I/O numbers are assigned according to the slot position of base (BSM-3 to 7). When mounting an input module in slot 0 in the example below, input numbers X00 to X15 are assigned, and output numbers Y220 to 235 are assigned when mounting an output module in slot 1 in the same example.

Input [X]	Power supply	CPU	X00 ~ X15	X20 ~ X35	X40 ~ X55	X60 ~ X75	X80 ~ X95	Power supply	X100 ~ X115	X120 ~ X135	X140 ~ X155	X160 ~ X175	X180 ~ X195
			Slot 0	Slot 1	Slot 2	Slot 3	Slot 4		Slot 5	Slot 6	Slot 7	Slot 8	Slot 9
Output [Y]	Power supply	CPU	Y200 ~ Y215	Y220 ~ Y235	Y240 ~ Y255	Y260 ~ Y275	Y280 ~ Y295	Power supply	Y300 ~ Y315	Y320 ~ Y335	Y340 ~ Y355	Y360 ~ Y375	Y380 ~ Y395



3

[**Explanation**]

1. The internal output (M) is equivalent to an auxiliary relay in relay sequence. It has many normally open contacts ("a" contacts) and norlally closed contacts ("b" contacts). They are used for generating an internal PC sequence.
2. There are two kinds of internal outputs (M) ;
   non-retentive (memory cleared to zero because of status change from power OFF to ON, stop to run or run to stop) and retentive (memory not cleared to zero regardless of status change from power OFF to ON, stop to run or run to stop). This is discriminated in I/O number.
3. When composing a self-holding circuit for retentive internal output (M700 to M955), it is usable as a keep relay.
4. Internal outputs with special function (M960 through M991)
   There are special internal outputs which function as a clock or a flag for a failure. Table 3-3 details the functions of each special intemnal output.

Table 3-1 shows how the external input (X), external output (Y), internal output (M) and timer/counter (T/C)) are assigned.

Table 3-1 Assignment of I/O Numbers

Classification		Number		Remarks
	Slot	Input module	Output module	
External input (160 points) or external output (160 points)	0	X 0~X 15	Y200~Y215	○ Decimal numbers ○ Numbers 0 to 15 are assigned when mounting a 16-point input module in slot 0. ○ Numbers 200 to 215 are assigned when mounting a 16-point output module in slot 0. ○ Numbers 8 to 15 are omitted when mounting an 8-point module. ○ For assignment of special modules, refer to Section 8.
	1	X 20~X 35	Y220~Y235	
	2	X 40~X 55	Y240~Y255	
	3	X 60~X 75	Y260~Y275	
	4	X 80~X 95	Y280~Y295	
	5	X100~X115	Y300~Y315	
	6	X120~X135	Y320~Y335	
	7	X140~X155	Y340~Y355	
	8	X160~X175	Y360~Y375	
	9	X180~X195	Y380~Y395	
Internal output: Non-retentive memory at power failure (256 points)		M400~M655		○ Decimal numbers ○ Each number has a data capacity of 8 bits. M400 b7 b6 b5 b4 b3 b2 b1 b0 M401 b7 b6 b5 b4 b3 b2 b1 b0 ○ The bit handling instruction determines ON/OFF status of b7. ○ The word handling instruction handles 8-bit data of M400 and that of M401, 16 bits in total, when No. 400 is designated.
Internal output: Retentive memory at power failure (256 points)		M700~M955		
Internal output: Special function (32 points)		M960~M991		○ All bit data ○ Detailed in Table 3-3.
Timer and counter (96 points in total)	Coil contacts	T/C 0~T/C 95		○ Decimal numbers ○ Timer and counter share the same number. ○ Up-timer and up-counter, respectively ○ 100 is added to timer/counter number (2-digit) for representing a current value, and 200 is added for indicating preset value. ○ States of coil and contacts are shown by bit data. ○ Current value and preset value are of 16 bit data.
	Current value	T/C 100~T/C 195		
	Preset value	T/C 200~T/C 295		

Table 3-2 lists each range of constent and argument used in instrucctions such as AJMP and MODE.

Table 3-2 Each Range of Constant and Argument

Classification		Range	Remarks
Constant	Word constant	0000H~9999H	The hexadecimal code H is not suffixed at the time of program entry. (Example) FUNO. 1234 (1234H→AR)
		0 ~FFFF	This constant is designated in a decimal number because the programmer does not have keys A to F which are indispensable for hexadecimal designation. Entry is possible in up to 3 digits. Effective range of decimal constant: 0 to 999 (Example) FUN51 427 (AR+1ABH → AR) (Decimal 427=hexadecimal 1ABH)
	Byte constant	00 ~FF	This constant is also designated in a decimal number because the programmer does not have keys A to F which are indispensable for hexadecimal designation. Effective range of decimal constant: 0 to 255 (Example) FUN50 255 (FFH → ARL) (Decimal 255=hexadecimal FFH)
	No. of bits	0~255	Used for FUN72 and FUN73. (Example) FUN72 5 (AR is masked by 5 bits from the left.)
Argument		0~63	Used as an argument of FUN08 (AJMP), FUN09 (AJEND), FUN42 (CALL), FUN43 (SB), FUN93 (INT) and FUN97 (MODE). (Example) FUN08 63 (AJMP63)

Table 3-3 Function of Special Internal Output (1/2)

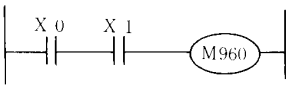
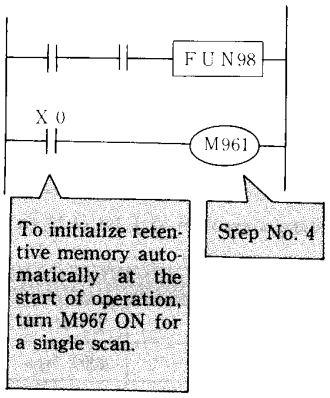
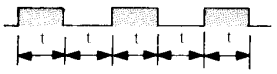
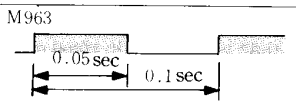
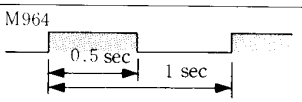
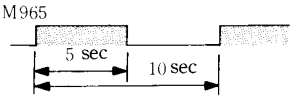
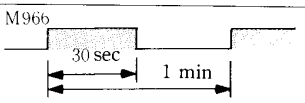
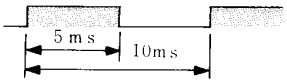
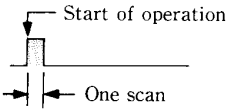
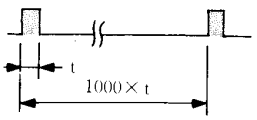
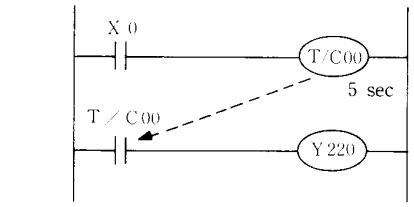
No.	Function	Description
M960	All outputs OFF	When M960 is switched ON by the program, all external output signals go OFF except for the RUN contacts.  ○ Suppose that an error program is written. (The X0 and X1 are not closed simultaneously during normal operation.) As a result, M 960 is switched ON. In this status, the PC judges that there is a system error and it switches all output signals OFF. However, program operation does not stop. ○ Eliminate the cause of the error and turn on power supply again.
M961	Initializing retentive memory	 ○ In the sysytem shown in the figure, retentive memory is or is not initialized depending on whether X0 is ON or OFF at the start of operation. XO: ON……Retentive memory is initialized when power is switched ON. XO: OFF……Retentive memory is not initialized when power is switched ON. ○ Retentive memory is initialized only at the start of operation. During operation, it is not initialized even if M961 is switched ON. ○ M961 coil operates only when it is written in step 4. It is invalid when it is weitten in any other step.
M962	Cyclic oscillation	 t: Period of one scan (scan time) Signal goed ON/OFF alternately for each scan.
M963	0.1 sec clock	 
M964	1 sec clock	
M965	10 sec clock	 
M966	1 min clock	
M969	10 ms clock	
M967	ON for a single scan after start of operation	 To initialize all volatile memories at the start of operation, use M967 in combination with M961. To initialize memory individually, use M967 alone.
M968	1000-scan cycle	 t: Scan time ON once every 1,000 scans. Used for measuring scan time.

Table 3-3 Function of Special Internal Output (2/2)

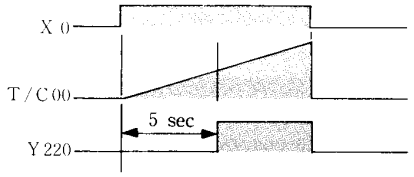
<table>
<tr><th>No.</th><th>Function</th><th>Description</th></tr>
<tr><td>WM970</td><td>System error factor</td><td>If system error occurs (when ERR lamp comes on), an error code within 0 to 65535 is displayed. Details are given number system Error codes in section 7. The code cannot be cleared by turning on power supply again.</td></tr>
<tr><td>WM972</td><td>Program counter at occurrence of system error</td><td>If system error, occurs, count on the program counter of microprocessor is displayed.</td></tr>
<tr><td>WM974</td><td>Designation of read address at occurrence of system error</td><td rowspan="2">If system error occurs, data at the address designated by WM 974 is presented in M976.</td></tr>
<tr><td>M976</td><td>Data readout at occurrence of system error</td></tr>
<tr><td>M977</td><td>Registration of system ROM sum</td><td rowspan="2">System ROM sum appears in WM978 only when this I/O is ON upon turning on power supply.</td></tr>
<tr><td>WM978</td><td>System ROM sum</td></tr>
<tr><td>M989</td><td>System attribute</td><td>Systm attribute appears in each bit of b7 to b5. Bits b4 to b0 are undefined.
<table>
<tr><th>Bit</th><th>At 0</th><th>At 1</th></tr>
<tr><td>b7</td><td>CPM-E2</td><td>CPM-E3</td></tr>
<tr><td>b6</td><td>9600 bps</td><td>4800 bps</td></tr>
<tr><td>b5</td><td>RUN instruction valid</td><td>RUN instruction invalid</td></tr>
</table>
Used for hardware check</td></tr>
<tr><td>WM980</td><td>Syntax error factor</td><td>If syntax error is detected in the check specified by a peripheral or in the check before start of operation, an error code within 0 to 65535 is displayed.
The code cannot be cleared by turning on power supply again.</td></tr>
<tr><td>WM982</td><td>Scan time</td><td>The latest scan time is indicated in steps of 10 ms, though the first scan is shown as 65535 ms.
Indication contains an error of ±10 ms.
Unit is millisecond (ms). (Indicated as 0, 10, 20, ms……)</td></tr>
<tr><td>WM984</td><td>Max. scan time</td><td>Of scan times after the start of operation, the maximum time is displayed in steps of 10 ms, though the first scan is shown as 0 ms.
Indication contains an error of ±10 ms.
Unit is millisecond (ms). (Indicated as 0, 10, 20, ms……)</td></tr>
<tr><td>M990</td><td>Normally ON</td><td>Always ON irrespective of run/stop status.</td></tr>
<tr><td>M991</td><td>ON during run</td><td>ON during run and OFF during stop</td></tr>
</table>

M986 through M988 are for functional expansion and unused (undefined) by the system.

Timers: T/C00 to T/C95 (96 points shared with counter)



Coding		Remarks
ORG	X 0	(1) Write a decimal point (.) before the preset value. (2) Write the I/O classification code "T/C."
OUT	T/C00.5	
ORG	T/C00	
OUT	Y220	



Time lapse is fetched in incremental mode.

Timer contacts close when the current time value reaches the preset value.

[Explanation]

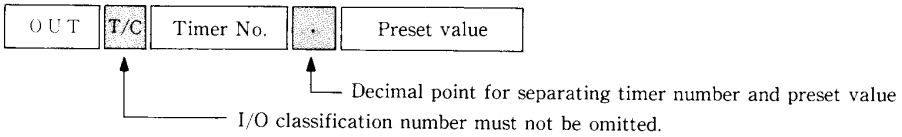
1. Kinds of timer

(1) On-delay timers are used. In the above sequence, the timer coil T/C00 is excited when input X0 turns ON. After 5sec, the timer contacts close. There are many timers with "a" and "b" contacts. They are used for generating a sequence in the PC.

(2) The same data area is shared by timers and counters, a total of 96 points (T/C00 throuh T/C95). A number used for a counter cannot be used for a timer.

2. Key input of timer

For specifying a timer coil using the programmer, enter the timer number (1 or 2 digits), a decimal point (.) as a separator and the preset value in this order.



3. Time base

The timers have two time bases: 0.01 and 0.1 sec. Time base is automatically selected according to the key-in method.

Time base	Key-in method	Preset value range
0.1sec	OUT T/C Timer No. · 6 3 · 5 → 63.5sec	T/C 0~9…0.1~999.9sec T/C10~95… {0.1~99.9sec / 1~999sec}
	OUT T/C Timer No. · 7 7 0 → 770.0sec	
0.01sec	OUT T/C Timer No. · 0 · 5 5 → 0.55sec	T/C 0~95…0.01~9.99 (settable only in 3 digits)

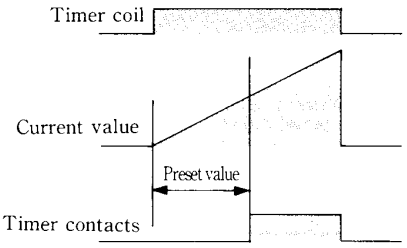
4. Preset value

Up to 10 timers/counters (T/C0 to T/C9) can be set using 4 digits (except for the timer adopting 0.01 sec time base which must be set using 3 digits).

Up to 86 timers/counters (T/C10 to T/C95) can be set using 3 digits.

5. Current value

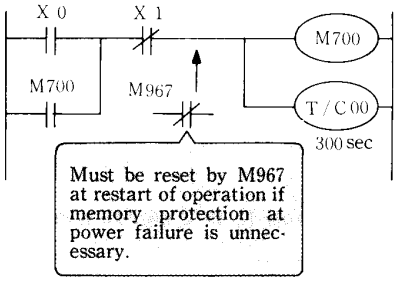
Each timer operates in the incremental mode. It starts timing when the timer coil is energized. When the current valuue reaches the present value, the timer contacts close. When the timer coil is deenergized, the current value is reset to 0.

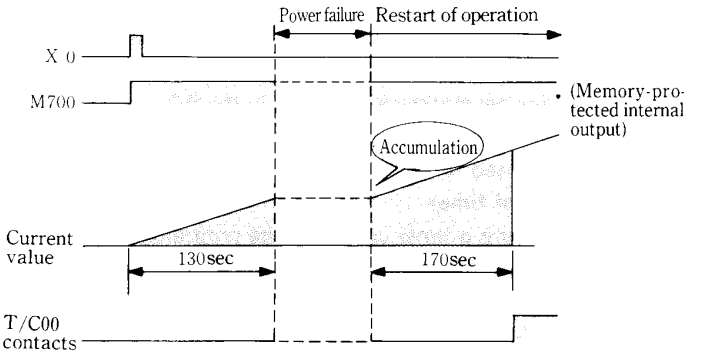


NOTE

The current value of each timer is retained in memory even if power failure occurs or when power supply is turned off. When combining the timer with the retentive internal output, an accumulation timer can be composed.

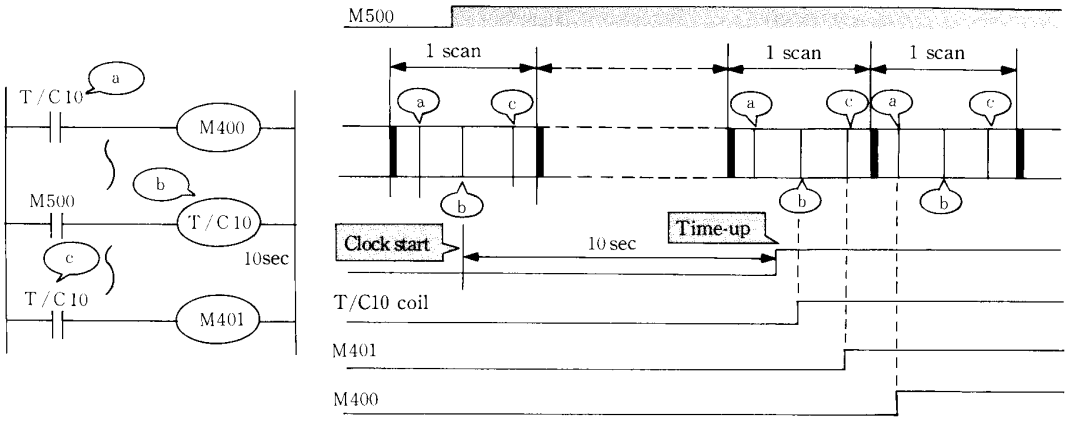
[Example]





6. Contacts operation timing and accurancy

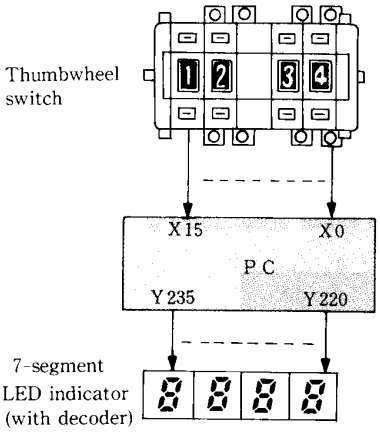
[Example]



The clock starts when the timer coil is energized (time point (b)). When the coil instruction is executed after time-up the output contacts close.

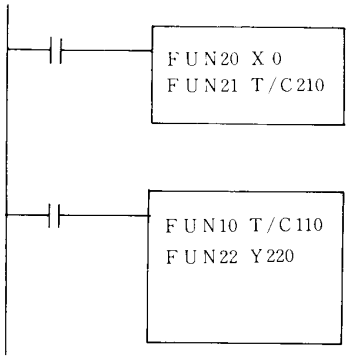
Condition	Timer starts by other than external input signal		Timer starts by external input signal.		Total timer accuracy
	Timer contacts (a) before coil	Timer contacts (b) after coil	Timer contacts (a) before coil	Timer contacts (b) after coil	
Timer accuracy	+2 scans	+1 scan	Input fethch delay (4 ms single scan filter) +2 scans	Same as left + 1 scan	Preset time +2 scans - time base (0.01 sec or 0.1 sec)

7. Handling timer preset value and current value in arithmetic instructions in application



The preset value of a timer can be changed by using the thumbwheel switch, and the current value of a timer can be read on the 7-segment LED indicator. An example of program is shown below. The table below lists the number assignment when using the timer preset value and current value in the arithmetic operation.

Segment	Assignment No.	Remarks
Current value	T/C100~T/C195	Add 100 to timer coils T/C00 to T/C95.
Preset value	T/C200~T/C295	Add 200 to timer coils T/C00 to T/C95.



X0 through X15(thumbwheel switch data) → AR

AR → timer T/C preset value

(T/C 210 programmed for preset value)

} The timer preset value can be changed by using the thumbwheel switch.

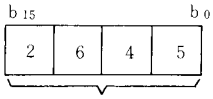
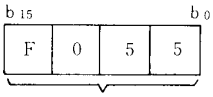
Current value of T/C10 timer → AR

(T/C110 programmed for current value)

AR → Y220~Y235

} The timer current value is read on the indicator.

The timer preset value and current value are data to be processed in blocks of 16 bits as shown below.

Segment	Kind of timer	Data to be processed by arithmetic instruction
Preset value and current value	0.1 sec timer	 ····· BCD 4 digits The least significant digit represents 0.1 sec order. Indicates 264.5 sec.
	0.01 sec timer	 ····· BCD 3 digits The most significant digit stands for "F" (0.01 sec timer). The least significant dight represents 0.01 sec order. Indicates 0.55 sec.

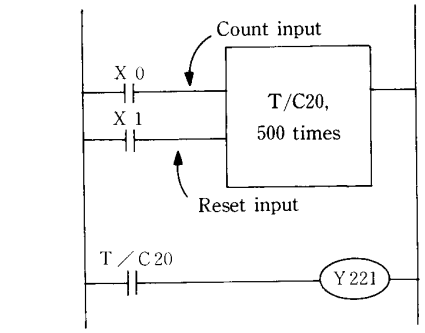
Q What is EEPROM?

A The memory device (EEPROM) of the E series does not require a battery. So it is easy to maintain.

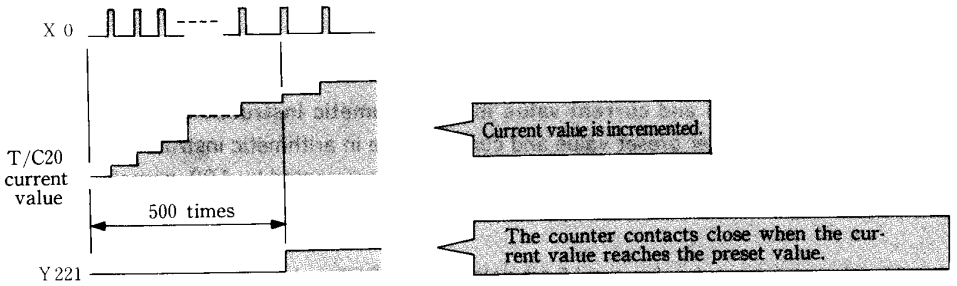
The EEPROM does not require battery backup for program. Hence, program will not be lost because of the end of useful life or abnormal discharge of a battery, and there is no need for tiresome battery replacement. Despite being a ROM, the EEPROM allows a program to be written and erased electrically like a RAM without using a ROM writer or UV eraser.
Compare the EEPROM with the already popular RAM and EPROM for an easier understanding.

Kind of memory	Program write	Program erase	Program protection reliability	Program store
EEPROM	Can be written electrically	Can be erased electrically	Intermediate	Battery unnecessary
EPROM	ROM writer necessary	UV eraser required	High	Battery unnecessary
RAM	Can be written electrically	Can be erased electrically	Low	Battery necessary

Counter: T/C00 through T/C95(selectable to function as timer or counter, total of 96)

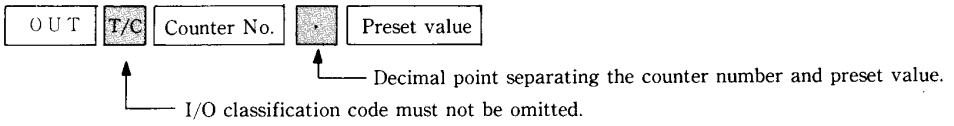


Code		Remarks
O R G	X 0	(1)The reset input is given in the form of STR instruction. (2)Place a decimal point(.)before the preset value. (3)Use the I/O claasification code "T/C."
S T R	X 1	
O U T	T/C20.500	
O R G	T/C20	
O U T	Y221	



[Explanation]

1. **Kind of counter**
   (1) An up-counter is used. In the above sequence, the counter T/C20 counts ON/OFF cycles of input X0. When the count reaches 500, the counter contacts close.
   The counters can provided with any number of "a" and "b" contacts.
   They are used for generating sequences in PC.
   (2) Timers and counters share the same data area. There are 96 timers/counters in total (T/C00 through T/C95).
   Once a T/C number is assigned to a timer, it cannot be reused for a counter.
   (3) When the reset input turns ON, the counter is reset and the current to 0.
2. **Counter key input**
   (1) Program the count input and reset input in this order. Reset input must be programmed by an STR instruction.
   (2) A counter preset value can be entered in the same way as for a timer.



3. Preset value

Up to 10 timers/counters (T/C0 to T/C9) can be set using 4 digits.

Up to 86 timers/counters (T/C10 to T/C95) can be set using 3 digits.

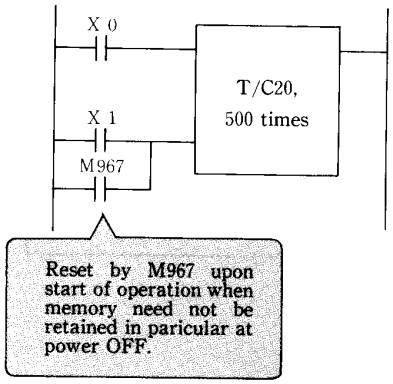
4. Current value

The current value of each counter is incremented by 1 (one) whenever the count input turns from OFF to ON. The counter contacts close when the current value reaches the preset value.

When the reset input turns ON, the current value is reset to 0.

The current value of the counter is retained in memory even if power in turned OFF.

[Example]

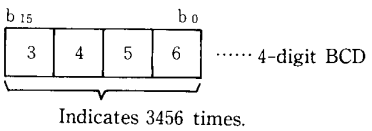


If the retentive data is unnecessary, use the special internal output M967, which turns on a single scan at start of operation. Program as shown at left.

5. Handling the counter preset value and current value in the arithmetic instructions.

When using a combination of counter preset value and current value in arithmetic instructions, the current value must be equal to the counter coil number (T/C0 throuh 95) incremented by 100, namely T/C100, to T/C195. The preset value must be equal to the coil number incremented by 200, namely T/C200 to T/C295.

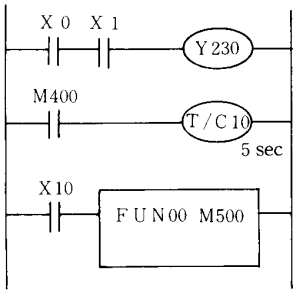
The counter preset value and current value are 16-bit data (4-digit BCD value) and processed as shown in the table below.

Item	Assingnment No.	Data to bi processed by arithmetic ininstruction
Current value	T/C100 through T/C195 (equal to counter coil numbers T/C0 to T/C95 incremented by 100)	
Preset value	T/C200 through T/C295 (equal to counter coil numbers T/C0 to T/C95 incremented by 200)	

Instruction can be in the form of bits, words and bit data handled as words.

1. Bit-type operating instruction

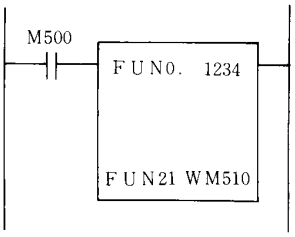
[Example]



A bit-type operating instruction affects only a single set of contracts (via coil) as shown in the figure.
The basic instructions are all bit-type instructions.

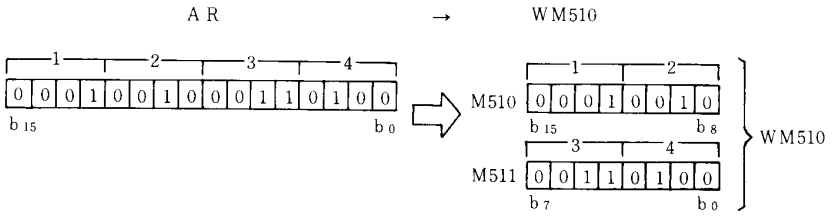
2. Word-type operating instruction

[Example]



Constant 1234H → AR

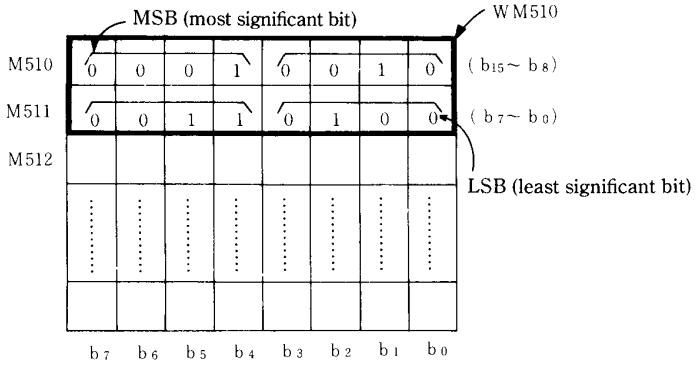
Aword-type operating insturction handles 16 bits as one word.



(1) In the above circuit, the constant 1234H is stored in the AR (arithmetic register) by "FUNO. 1234." The AR data is output to the 16 bits of M510 and M511 by FUN21 WM510."

(2) When an internal output number is specified by a word-type operating instruction, it is handled as 16-bit data in the following way. The 8-bit data of the specified internal output (M510 in the above example) is taken as b8 through b15, while that of the next internal output (M511) is taken as b0 through b7.

[**Example**]

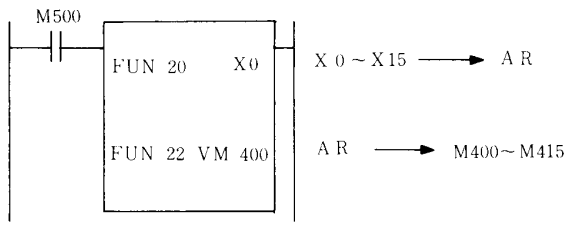


(3) The timer and counter preset values, current values and constants (0000H to 9999H) are all 16-bit data. So they are directly processed as a word when specifying their numbers by a word-type operating instruction.

3. Handling of bit-type data as a word

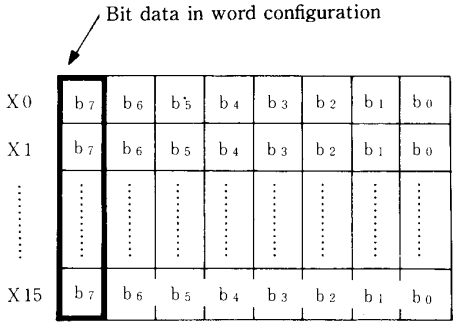
An instruction that treats 16 one-bit data (X0 to X15) as a single word is called a "word-type instruction for bit data."

[**Example**]



(1) In the above sequence, the 16-bit data of X0 through X15 is stored in the AR by the "FUNO20 X0" instruction, and the data in the AR is output to M400 to M415 by the "FUN22 M400" instruction.

(2) When external I/O or internal output number is specified by a word-type instruction for bit data, only the most significant bit (b7) of the 16 points (namely, 16 bits) starting from the specified No. (X0 and M400 in the above example) is handled as a single-word data.

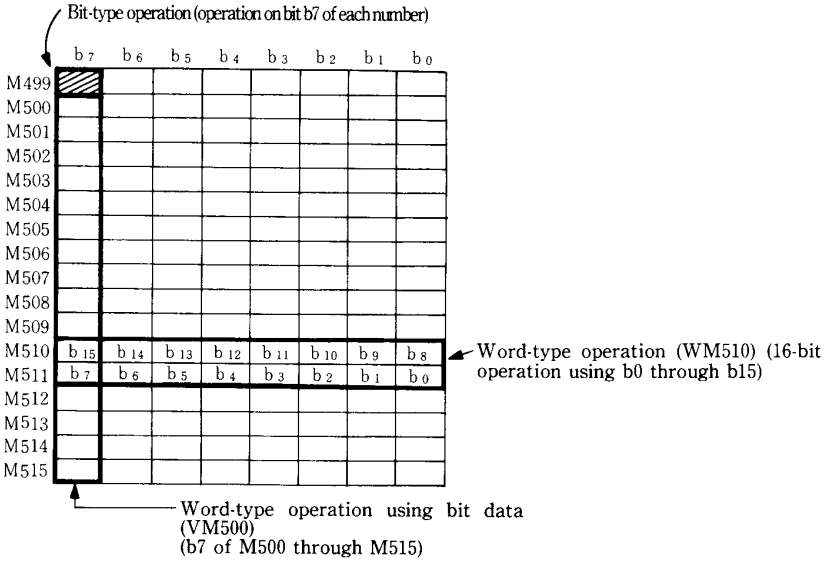


Example of external input (Word configuration remains the same in case of external output and internal output.)

(3) Word-type instructions for bit data are used for connecting the thumbwheel switch,etc. as an external input device for storing BCD data.
They are also used to output data in the AR to an external output terminal. (See the section "Handling of the timer preset value and current value by arithmetic instructions.")

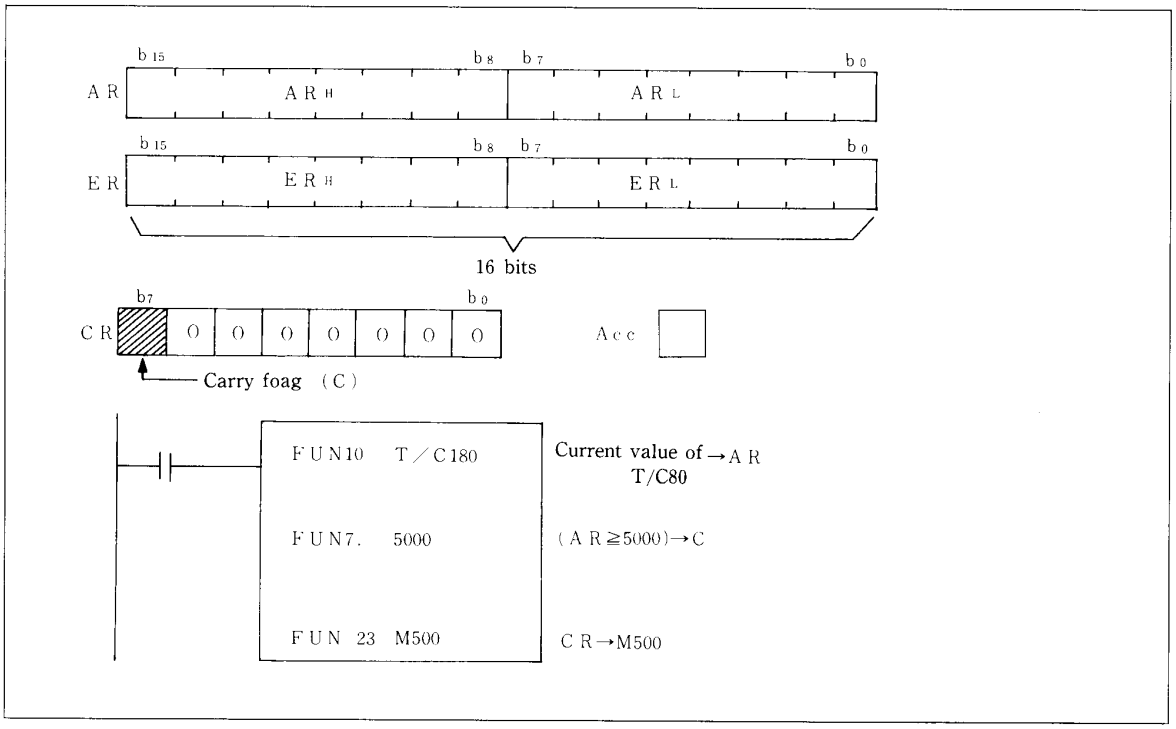
4. Summary of instructions

[Example]



(1) Internal outputs are all 8 bits long.
In word-type operation, a total of 16 bits in the specified internal output number and the next number are handled. This data is given an element code WM. WM510 in the above example consists of M510 and M511 (b0 to b15).

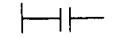
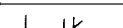
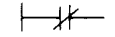
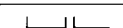
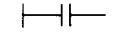
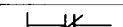
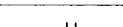
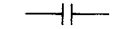
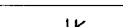
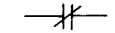
(2) In the word-type operation using bit data, the most significant bit (b7) of sixteen 8-bit data starting from the specified number is handled as a single-word data.
This word data is made up of the 16 bits in the verticcal direction. Hence it is given an element code VM.



[Explanation]

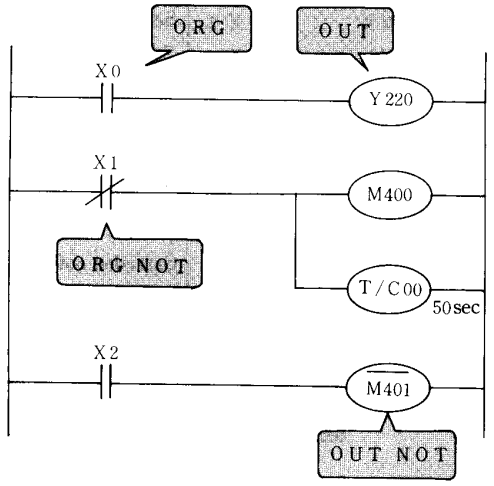
1. The registers of EM-II series come in 4 kinds below.
   (1) AR: Arithmetic Register used for instructions. It has a 16-bit configuration.
   (2) ER: Expansion Register used for storing upper word resulting from multiplication and remainder of division. It has a 16-bit cofiguration.
   (3) CR: Carry Register. Carry flag (C) turns to "1," for example when the condition for comparison is satisfied. Bits b0 to b6 are always "0."
   (4) Acc: 1-bit register which automatically changes along with execution of a basic instruction such as ORG or AND.
2. Data in the AR, ER and CR are cleared every time scan starts and they change in response to the processing of arithmetic instructions.

Table 4-1 Basic Instructions

Instruction	Symbol	Function	Coponent	No. of words	Change in register				Reference page
					AR	ER	C	Acc	
ORG		Connection of normally open contacts ("a" contacts) to bus	X, Y, M, T/C 0~95	1	●	●	●	↕	45
ORG NOT		Connection of normally closed contacts ("b" contacts) to bus		1	●	●	●	↕	45
STR		Start of branching normally open contacts ("a" contacts)	X, Y, M	1	●	●	●	↕	48
STR NOT		Start open branching normally closed contacts ("b" contacts)	T/C 0~T/C95	1	●	●	●	↕	48
AND		Serial connection of normally open contacts ("a" contacts)	X, Y, M	1	●	●	●	↕	46
AND NOT		Serial connection of normally closed contacts ("b" contacts)	T/C 0~T/C95	1	●	●	●	↕	46
OR		Parallel connection of normally open contacts ("a" contacts)	X, Y, M	1	●	●	●	↕	47
OR NOT		Parallel connection of normally closed contacts ("b" contacts)	T/C 0~T/C95	1	●	●	●	↕	47
AND STR		Serial Connection of logic block	None	1	●	●	●	↕	48
OR STR		Parallel connection of logic block		1	●	●	●	↕	48
OUT		Output of calculation result	Y, M T/C 0~T/C95 (with preset value)	1	●	●	●	●	45
OUT NOT		Inverted output of calculation result	Y, M	1	●	●	●	●	45

●: Register remains unchanged.
↕: Register changes.

Instruction	Symbol	Meaning	Function	Component	No. of words	Change in register AR	ER	C	Acc
ORG	⊢⊣⊢	Origin	Connection of normally open contacts ("a" contacts) to bus	X, Y, M, T/C0~95	1	·	·	·	↕
ORG NOT	⊢⊬⊢	Inverted origin	Connection of normally closed contacts ("b" contacts) to bus		1	·	·	·	↕
OUT	—○⊣	Output	Output of calculation result	Y, M T/C0~T/C95 (with preset value)	1	·	·	·	·
OUT NOT	—○⊣	Inverted output	Inverted output of calculation result	Y, M	1	·	·	·	·



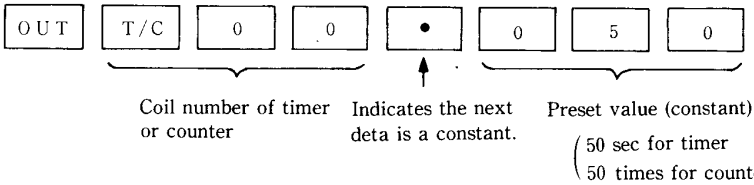
Code	Remarks
ORG X 0 OUT Y 220	Output in connection with bus
ORG NOT X 1 OUT M 400 OUT T/C00 . 050	 Tmier (coil) Tmier (preset value)
ORG X 2 OUT NOT M 401	Inverted output

Element codes X, Y and M need not be keyed in.

4.1

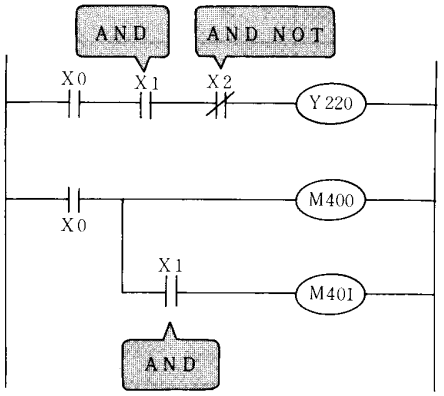
[Explanation]

1. The ORG and ORG NOT instructions are used for the contact next to the bus (at the head of circuit).
2. The OUT instruction drives each coil of external output (Y), internal output (M), timer (T) and counter (C). This instruction is not used for external input (X). The OUT NOT instruction is used for inverted output.
3. More than one OUT instruction (multiple outputs) can be used in parallel.
4. A preset value (constant) is required after an OUT instruction for a timer or counter coil.



After OUT instruction, the element number of timer/counter coil (T/C00 through T/C95), period ". " for indicating a constant and preset value must be entered in this order. This occupies a single step.

Instruction	Symbol	Meaning	Function	Component	No. of words	Change in register AR	ER	C	Acc
AND	⊣⊢	And	Serial connection of normally open contacts ("a" contacts)	X, Y, M	1	·	·	·	↕
AND NOT	⊣/⊢	Inverted and	Serial connection of normally closed contacts ("b" contacts)	T/C 0~T/C95	1	·	·	·	↕



Code		Remarks
ORG	X 0	
AND	X 1	Serial contacts
AND NOT	X 2	Serial contacts
OUT	Y 220	
ORG	X 0	
OUT	M 400	
AND	X 1	Serial contacts
OUT	M401	Cascaded output

Element codes X, Y and M need not be keyed in.

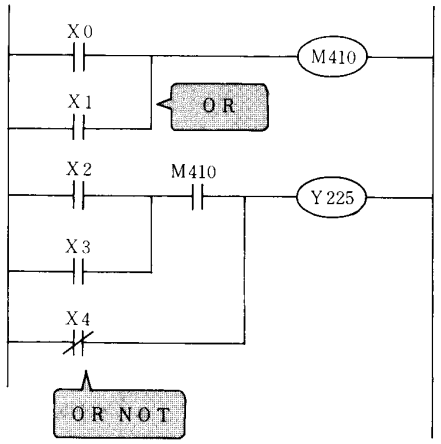
[Explanation]

1. The AND and AND NOT instructions are used for connecting a single set of contacts in series to the existing circuit.
2. Driving another coil via a contact set after OUT instruction is called a cascaded output (M400 and M401 in the figure above). Cascaded output can be repeated any number of times.

NOTE

It is recommended not to use more than 8 contact sets horizontally nor more than 26 lines vertically in a circuit, although the number of series contacts and cascaded outputs is not limited. This is because of functional restrictions on the portable graphic programmer (PGM-GPE2) and printer.

Instruction	Symbol	Meaning	Function	Component	No. of words	Change in register AR	ER	C	Acc
OR		Or	Parallel connection of normally open contacts ("a" contacts)	X, Y, M	1	·	·	·	↕
OR NOT		Inverted or	Parallel connection of normally closed contacts ("b" contacts)	T/C 0~T/C95	1	·	·	·	↕



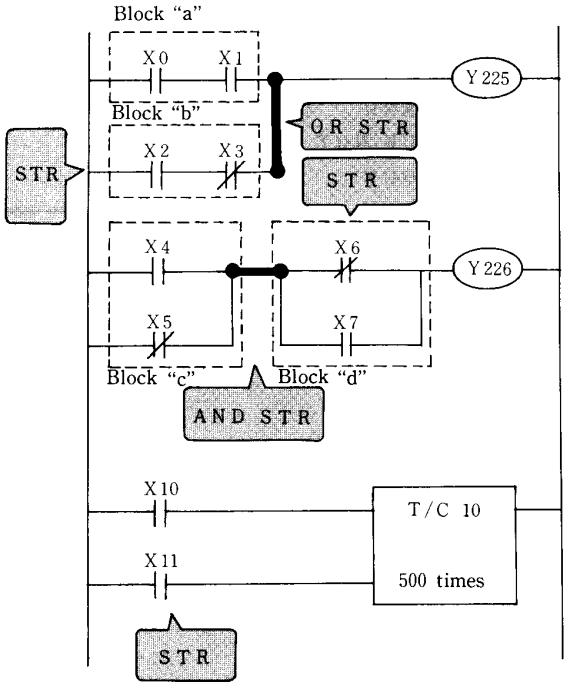
Code		Remarks
ORG	X0	Parallel connection
OR	X1	
OUT	M410	
ORG	X2	Parallel connection
OR	X3	
AND	M410	
OR NOT	X4	
OUT	Y225	

4.1

[Explanation]

The OR and OR NOT instructions establish a parallel connection of a contact set to the existing circuits. To connect a serial circuit block consisting of two or more serially connected contact sets (—┤├—┤├—) in parallel with another circuit, use the OR STR instruction explained later.

Instruction	Symbol	Meaning	Function	Component	No. of steps	Change in register AR	ER	C	Acc
STR	⊢⊣⊢	Store	Start of branching normally open contacts ("a" contacts)	X, Y, M	1	·	·	·	↕
STR NOT	⊢⊬⊢	Inverse of store	Start of branching normally closed contacts ("b" contacts)	T/C 0~T/C95	1	·	·	·	↕
AND STR		And store	Serial connection of logic block	None	1	·	·	·	↕
OR STR		Or store	Parallel connection of logic block		1	·	·	·	↕

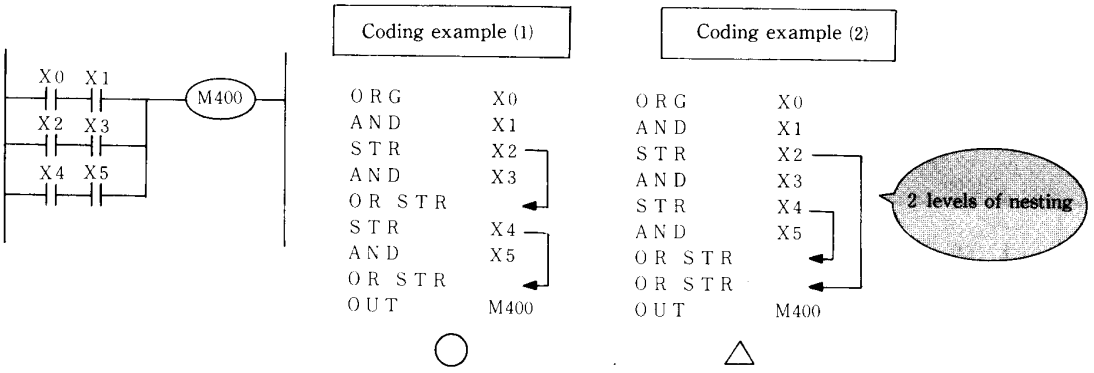


Code		Remarks
ORG	X0	a: • Block "a" is programmed.
AND	X1	
STR	X2	b: • Block "b" is programmed.
AND NOT	X3	
OR STR		a + b • Blocks "a" and "b" are combined by OR STR.
OUT	Y225	
ORG	X4	c: • Block "c" is programmed.
OR NOT	X5	
STR NOT	X6	d: • Block "d" is programmed.
OR	X7	
AND STR		c•d • Blocks "c" and "d" are combined by AND STR.
OUT	Y226	
ORG	X10	
STR	X11	
OUT T/C	10	
	500	

[Explanation]

1. A circuit with two or more contact sets connected in series is called a serial circuit block. When connecting series circuit blocks in parallel, use the STR or STR NOT instruction to begin the branch, and the OR STR instruction to end the branch.
2. A circuit with two or more contact sets connected in parallel is called a parallel circuit block. When connecting parallel circuit blocks in series,use the STR or STR NOT instruction to begin the branch,and the AND STR instruction to end the branch.

3. The circuit shown below can be programmed according to either coding example (1) or (2)



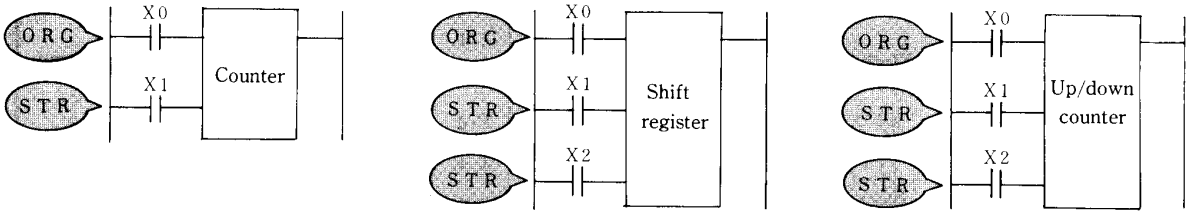
(1) Even when many parallel blocks are to be used, each circuit block is connectable to the previous one by specifying the OR STR instruction. The number of connections is not limited. (See coding example (1))

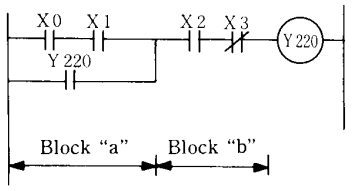
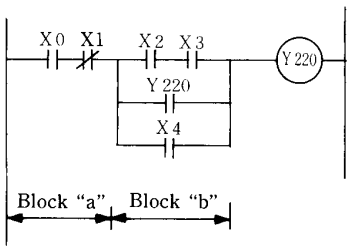
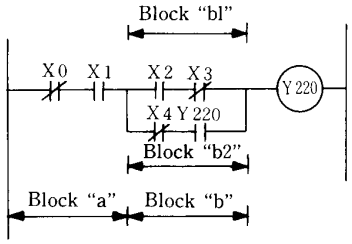
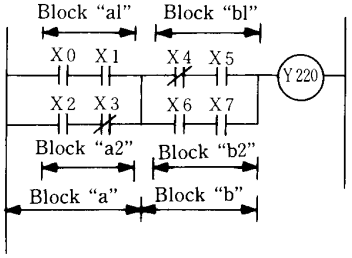
(2) The OR STR instruction can be used in the batch mode. In this case, however, the number of interations of the STR (STR NOT) instruction is limited to 7 times (up to 7 levels of nesting). (See coding example (2).)

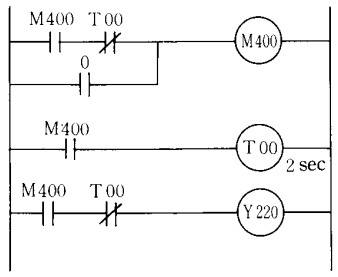
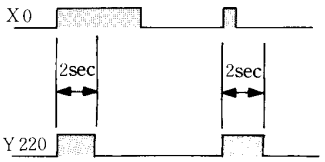
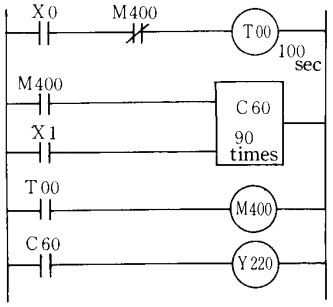
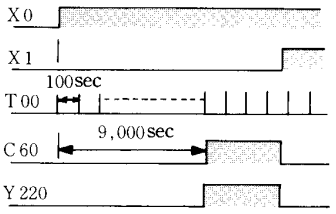
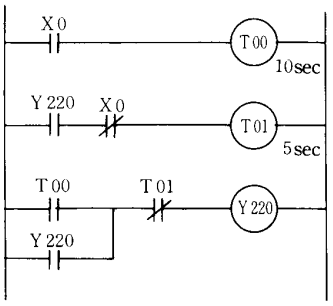
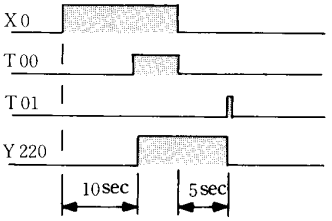
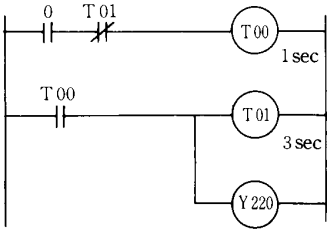
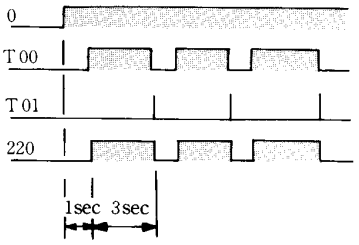
(3) The same rule applies to the AND STR instruction as well.

(4) If the STR or STR NOT is not used in correct combination with AND STR (or OR STR), it is detected as a syntax error.

4. The STR or STR NOT instruction does not correspond to the AND STR (or OR STR) instruction if the counter, up/down counter, shift register or similar circuit has two or more input conditions.



Circuit	Configuration	Program		Explanation
		Instruction code	Data	
Parallel-to-serial circuit		ORG AND OR AND AND NOT OUT	X 0 X 1 Y220 X 2 X 3 Y220	• First the parallel circuit of block "a" and then the serial circuit of block "b" are programmed.
Serial-to-parallel circuit		ORG AND NOT STR AND OR OR AND STR OUT	X 0 X 1 X 2 X 3 Y220 X 4 Y220	• The circuit is divided into blocks "a" and "b" which are programmed separately. • Blocks "a" and "b" are combined by AND STR.
		ORG NOT AND STR AND NOT STR NOT AND OR STR AND STR OUT	X 0 X 1 X 2 X 3 X 4 Y220 Y220	• Block "a" is programmed. • Block "bl" is programmed. • Block "b2" is programmed. • Blocks "bl" and "b2" are combined by OR STR. • Blocks "a" and "b" are combined by AND STR.
Serial connection of parallel circuits		ORG AND STR AND NOT OR STR STR NOT AND STR AND OR STR AND STR OUT	X 0 X 1 X 2 X 3 X 4 X 5 X 6 X 7 Y220	• First block "a1" and then block "a2" are programmed. • These blocks are combined. by OR STR. • Blocks "bl" and "b2" are programmed in the same way as above. • Blocks "a" and "b" are combined by AND STR.

Circuit		Configuration	Program		Explanation
			Instruction code	Data	
Timer/counter application circuit	One-shot circuit		ORG AND NOT T/C OR OUT ORG OUT T/C ORG AND NOT T/C OUT	M400 00 X0 M400 M400 00.002 M400 00 Y220	
	Timer and counter circuits		ORG AND NOT OUT T/C ORG STR OUT T/C ORG T/C OUT ORG T/C OUT	X0 M400 00.100 M400 X1 60.090 00 M400 60 Y220	
	ON/OFF delay circuit		ORG OUT T/C ORG AND NOT OUT T/C ORG T/C OR AND NOT T/C OUT	X0 00.010 Y220 X0 01.005 00 Y220 01 Y220	
	Flicker circuit		ORG AND NOT T/C OUT T/C ORG T/C OUT T/C OUT	X0 01 00.001 00 01.003 Y220	

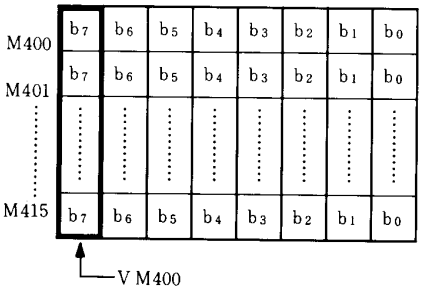
4.1

Application Instructions (I)

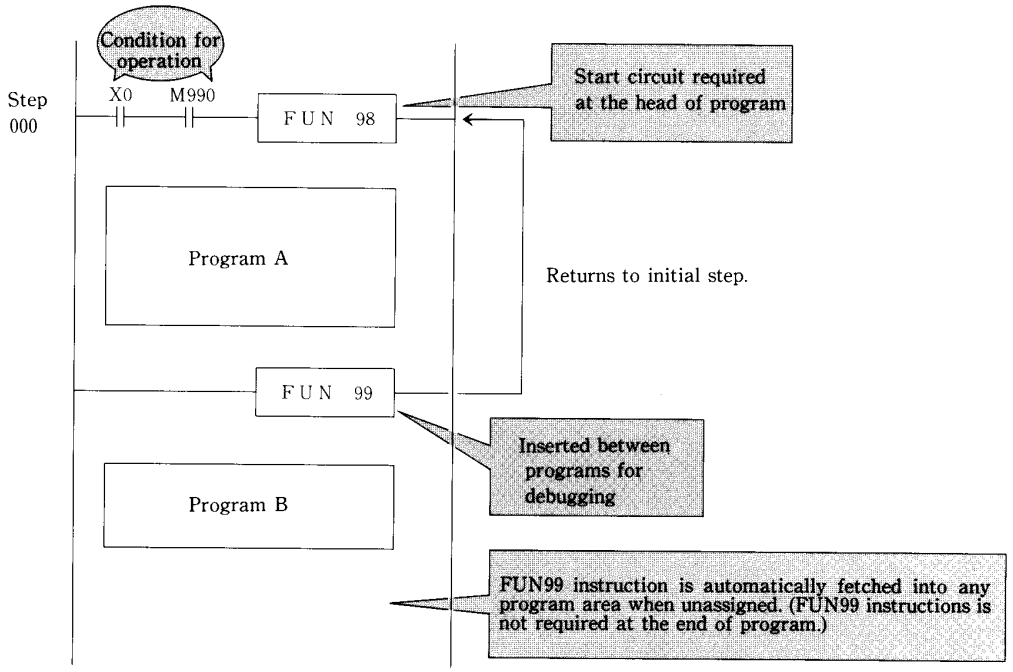
Classification	Instruction	Symbol	Name	Function	Component	No. of words	Change in register				Reference page
							AR	ER	C	Acc	
Edge	FUN00	DIF	Rising edge	Detects rising edge (_┌‾_)of signal	M	1	●	●	●	●	57
	FUN01	DFN	Trailing edge	Detects trailing edge (‾└_)of signal.	M	1	●	●	●	●	57
Step process	FUN02	IF	If	Set/reset	None	1	●	●	●	●	59
	FUN03	IFR	If reset	Step process		1	●	●	●	●	59
Master control	FUN04	MCS	Master control	Sets common serial contacts.	None	1	●	●	●	●	63
	FUN05	MCR		Releases common serial contacts.		1	●	●	●	●	63
Jump	FUN06	JMP	Jump without addressing	Skip program up to corresponding JEND.	None	1	●	●	●	●	65
	FUN07	JEND				1	●	●	●	●	65
	FUN08	AJMP	Jump with addressing	Jumps to AJEND at corresponding address number.	Address No. (O to 63)	2	●	●	●	●	65
	FUN09	AJEND				2	●	●	●	●	65
Branch	FUN28	BRANCH	Branch	Stores Acc.	None	1	●	●	●	●	68
	FUN29	RETURN	Return	Returns stored Acc.	None	1	●	●	●	↕	68
Up/down counter	FUN40	UDC	Up/down counter	Up/down counter	VM (Note)	1	●	●	●	●	67
NOP	FUN41	NOP	No operation	Nothing occurs.	None	1	●	●	●	●	71
Latch	FUN45	LATCH	Latch	Resetting priority latch	M	1	●	●	●	●	69
Shift register	FUN47	SFR	Shift register	16-bit shift register	VM (Note)	1	●	●	●	●	70
Set and reset	FUN88	SET	Set	Turns on component when Acc is at ON.	Y, M	1	●	●	●	●	58
	FUN89	RES	Reset	Turns off component when Acc is at ON.	Y, M	1	●	●	●	●	58
Start and end	FUN98	STA	Start	Operation start cntrol	None	1	●	●	●	●	55
	FUN99	END	End	Returns program to initial step.	None	1	–	–	–	–	55

●: Register remains unchanged
↕: Register changed
–: Register cleared

(Note) VM represents vertical 16 bits.
In the example below, VM is made up of 16 most significant bits of M400 throgh M415.



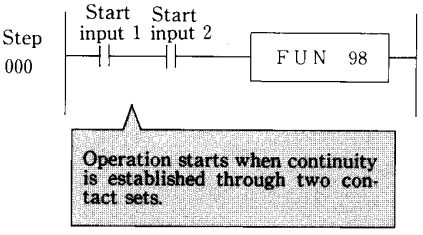
Instruction	Symbol	Name	Function	Component	No. of words	Change in register			
						AR	ER	C	Acc
FUN98	STA	Start	Operation start control	None	1	·	·	·	·
FUN99	END	End	Returns program to initial step.	None	1	–	–	–	–



[Explanation]

1. Start circuit

Operation start input is to be specified through a program. This means that start circuit must be written at the head of any program. (There is no restriction on number.)

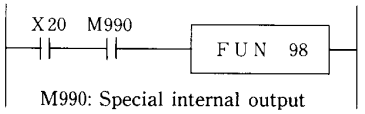


Condition for operation

Continuity is established across both start inputs 1 and 2, and program does not have an error.

(⊣/⊢ ("b" contacts) specifiable)

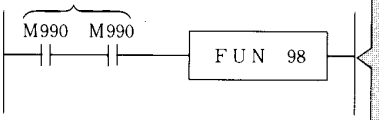
Example of start circuit



M990: Special internal output always turned on

(a) Operation starts with external input X20 at ON and stops at OFF

Always at ON



(b) Operation starts when turning on supply.

NOTE

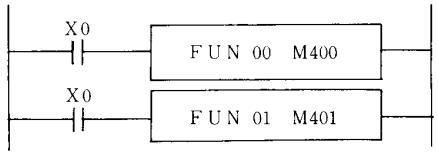
Under this configuration, operation starts as soon as the power supply is turned on. So avoid it for test run.
Such a configuration should be programmed when needed after completion of test run.

2. End

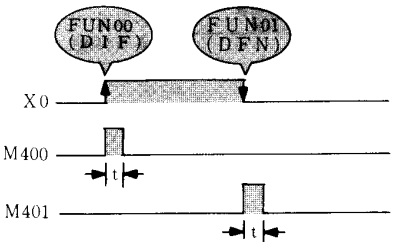
(1) The FUN99 instruction is not required usually. However, it is recommended to insert this instruction for separating programs at the time of test run since operation can be checked more easily. Program is executed from step 000 to FUN99 instruction.
Once operation has been confirmed, delete the FUN99 instruction.

(2) After completely clearing a program, all user memories are written with the FUN99 instruction (through indication is not provided).
Since the FUN99 instruction is assumed in an area not yet programmed, there is no need for writing that instruction at the end of a program.

Instruction	Symbol	Name	Function	Component	Skip size	Change in register AR	ER	C	Acc
FUN00	DIF	Rising edge	Detects rising edge (_┌‾) of signal.	M	1	·	·	·	·
FUN01	DFN	Trailing edge	Detects trailing edge (‾└_) of signal.	M	1	·	·	·	·



Code		Remarks
ORG	X0	
FUN00	M400	Detects rising edge.
ORG	X0	
FUN01	M401	Detects trailing edge.

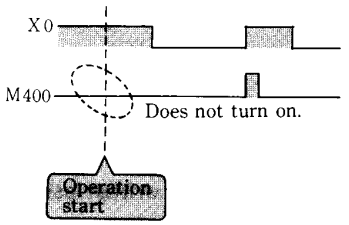


t: one (1) scan time

4.2

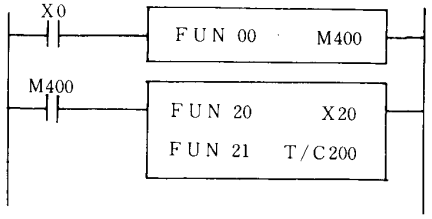
[Explanation]

1. The FUN00 (DIF) instruction is used to detect the rising edge of an input signal (status change from LOW to HIGH), and the FUN01 (DFN) instruction is used to detect the trailing edge of the signal (status change from HIGH to LOW). These instructions are programmed in combination with an internal output (M) so that the specified internal output (M) turns on only for 1 scan time when the edge is detected. Any number of FUN00 and FUN01 instructions can be used (so far as internal output permits).
2. The edge detect instructions are executed according to the input change after operation start.

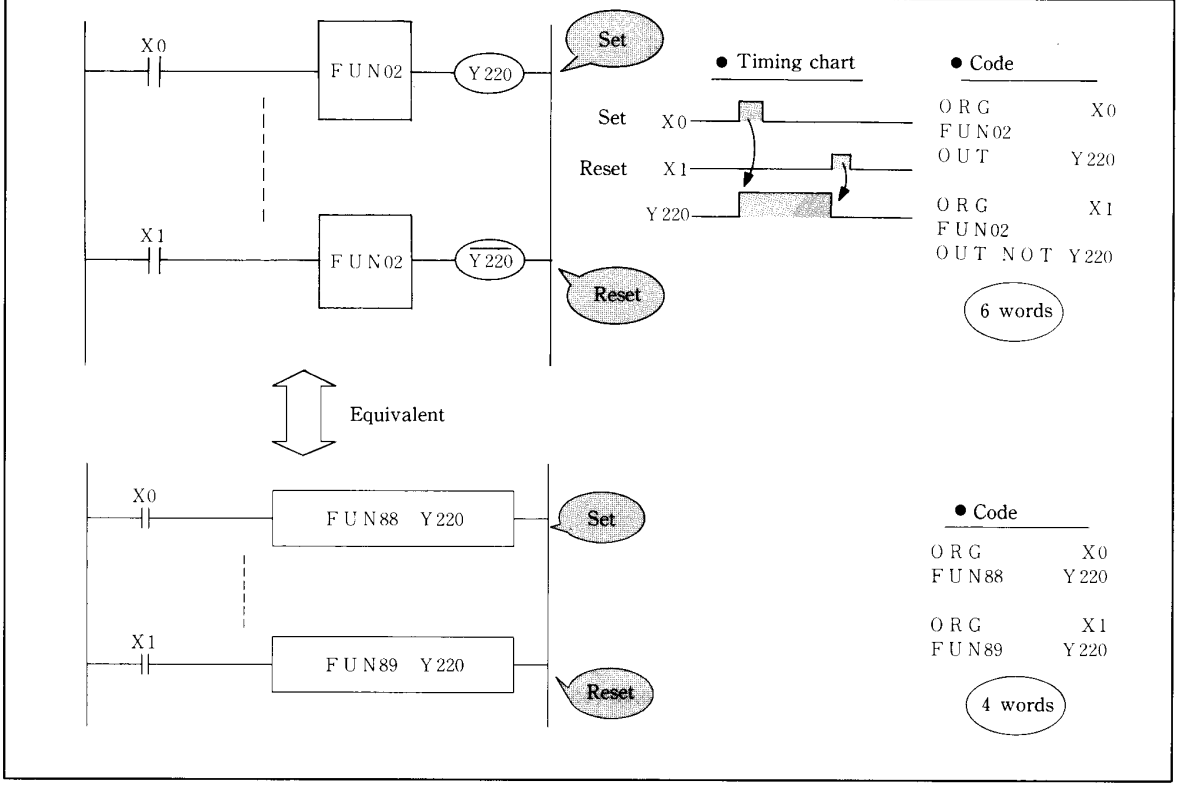


Rising edge is not detected and M400 does not turn on when X0 is already turned on at start of operation.

3. The edge detect function is effective for word LOAD, COMPARE and the like instructions, because they can be executed only when input condition changes.
(For instance, this function is used as a startup condition of arithmetic instructions.)



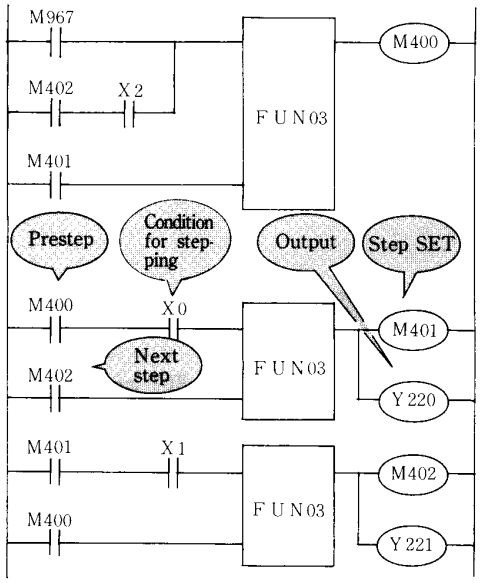
Instruction	Symbol	Name	Function	Component	No. of steps	Change in register			
						AR	ER	C	Acc
FUN02	IF	If	Set/reset	None	1	·	·	·	·
FUN88	SET	Set	Turns on component when Acc is at ON.	Y, M	1	·	·	·	·
FUN89	RES	Reset	Turns off component when Acc is at ON.	Y, M	1	·	·	·	·



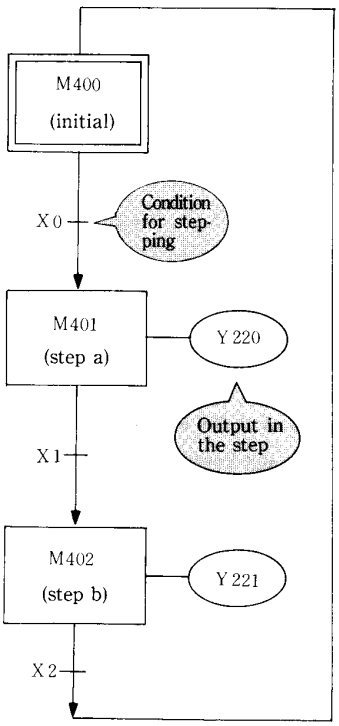
[**Explanation**]

1. Instructions FUN02 and OUT are combined and used as the SET instruction.
   Instructions FUN02 and OUT NOT are combined and used as the RESET instruction.
   - ○ ON status is held under SET input and OFF status is held under RESET input.
   - ○ Any other program may be inserted between SET coil and RESET coil. The program written last is given the highest priority.
   - ○ A keep relay can be composed when combining a FUN02 instruction with the memory-protected internal output.
2. FUN88 is the SET instruction. It provides the same function as a combination of the FUN02 and OUT instructions.
   FUN89 is the RESET instruction. It provides the same function as a combination of the FUN02 and OUT NOT instructions.
   Each of FUN88 and FUN89 instructions requires fewer words than the corresponding combination of instructions.
   If an output coil is programmed using both FUN88 and FUN89 instructions, a syntax error (double coil error E.) occurs, but operation is continuable.

Instruction	Symbol	Name	Function	Component	No. of words	Change in register			
						AR	ER	C	Acc
FUN03	IFR	If reset	Step process	None	1	·	·	·	·



● Status transition diagram



● Code

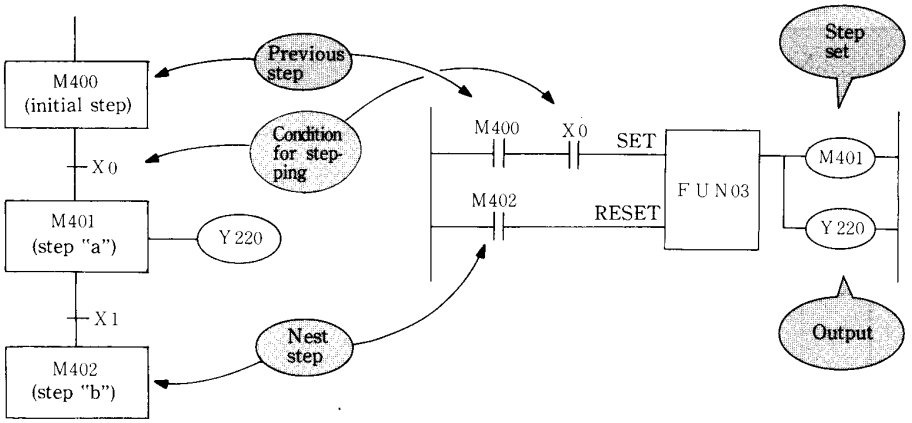
```
ORG     M967
STR     M402
AND     X2
OR STR
STR     M401
FUN 03
OUT     M400
ORG     M400
AND     X0
STR     M402
FUN 03
OUT     M401
OUT     Y220

ORG     M401
AND     X1
STR     M400
FUN 03
OUT     M402
OUT     Y221
```

4.2

[**Explanation**]

1. FUN03 is the step process (sequential control) instruction. Set input and reset input are provided. A step process program can be created in the regular format using the status transition diagram.



Status transition diagram

Explanation of operation

(1) If step condition X0 is set to ON in the initial step (M400), step "a" (M401) turns ON and Y220 is output.

(2) Y200 holds its output even when step condition X0 is set to OFF.

(3) When step condition X1 is set to ON, step "b" turns ON and Y220 is set to OFF.

(4) Even when step condition X1 is set to ON in the initial step (M400), step "b" (402) won't turn ON. All steps are executed in correct sequence.

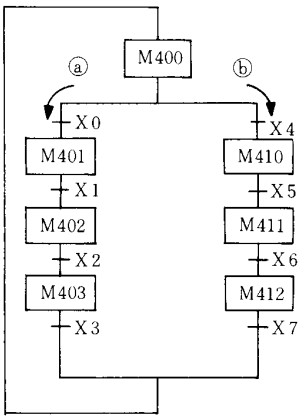
Progromming method

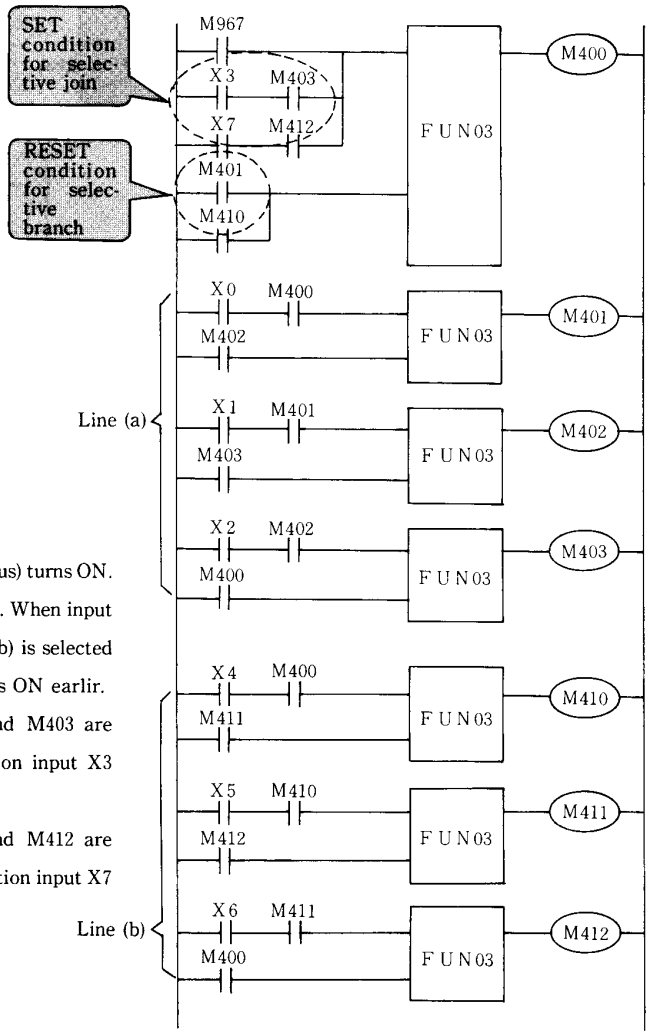
(1) To program a FUN03 SET input, the internal output (M400) specifying the previous step is ANDed with the condition for stepping (X0).

(2) For FUN03 RESET input, the internal output (M402) specifying the next step is programmed.

(3) After FUN03 the internal output (M401) specifying the current step and output (Y220) are programmed.

Selective branch and join





Explanation

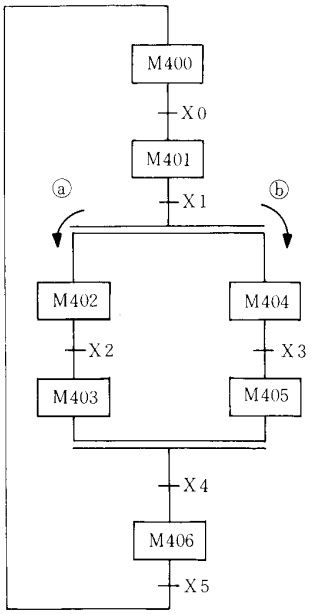
(1) Upon start lf operation,step M400 (initial status) turns ON.

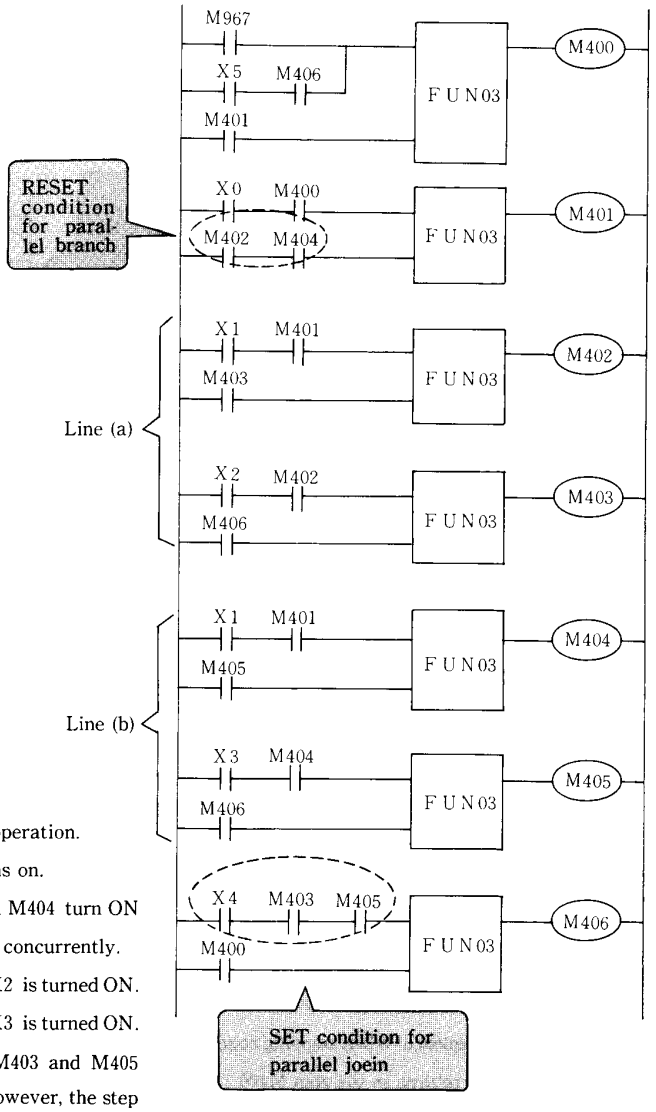
(2) When input X0 turns ON, line (a) is executed. When input X4 turns ON, line (b) is executed. Line (a) or (b) is selected according to which of inputs X0 and X4 turns ON earlir.

(3) In line (a), steps involving M401, M402 and M403 are executed in this order. When stepping condition input X3 turns ON, control returns to the step M400.

(4) In line (b), steps involving M410, M411 and M412 are executed in this sequence. When stepping condition input X7 turns ON, control returns to the step M400.

Parallel branch and join





Explanation

(1) Step M400 (intial status) is sit on start of operation.

(2) When input X0 is turned ON, step 401 turns on.

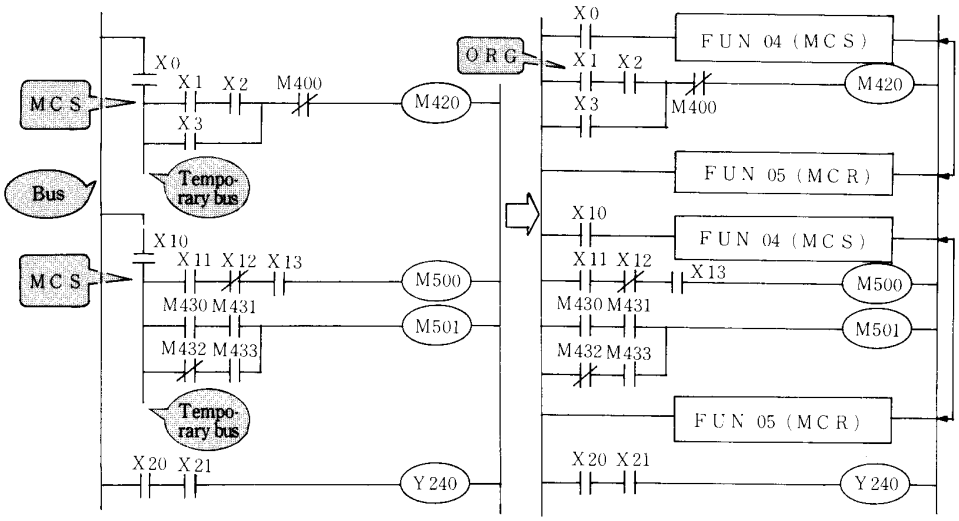
(3) When input X1 is turned ON, steps 402 and M404 turn ON simultaneously.Lines (a) and (b) are executed concurrently.

(4) In line (a), step M403 turns ON when input X2 is turned ON.

(5) In line (b), step M405 turns ON when input X3 is turned ON.

(6) When input X4 is turned ON with both M403 and M405 activated, the common step M406 turns on. However, the step M406 won't turn ON when M403 and M404 are activated in lines (a) and (b), respectively. Control of lines (a) and (b) returns to step M400 simultaneously when input X5 is set to ON.

Instruction	Symbol	Name	Function	Component	No. of words	Change in register AR	ER	C	Acc
FUN04	MCS	Master control	Sets common serial contacts.	None	1	·	·	·	·
FUN05	MCR		Resets common serial contacts.		1	·	·	·	·



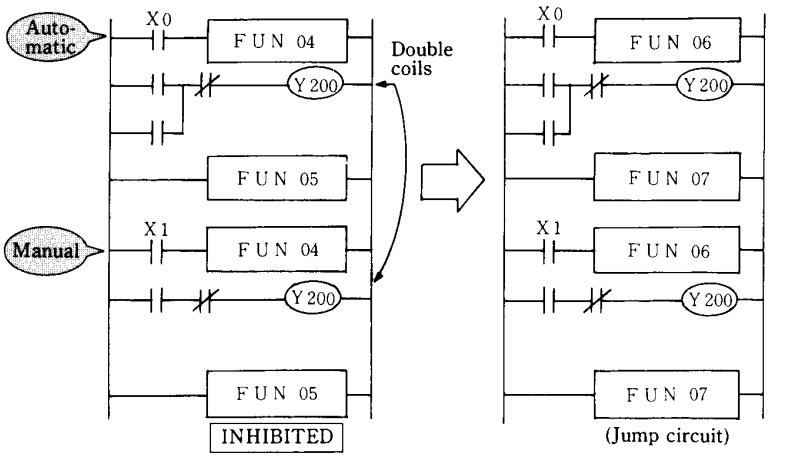
Code

```
ORG        X0
FUN04
ORG        X1
AND        X2
OR         X3
AND NOT    M400
OUT        M420
FUN05
ORG        X10
FUN04
ORG        X11
AND NOT    X12
AND        X13
OUT        M500
ORG        M430
AND        M431
STR NOT    M432
AND        M433
OR STR
OUT        M501
FUN05
ORG        X20
AND        X21
OUT        Y240
```

4.2

[Explanation]

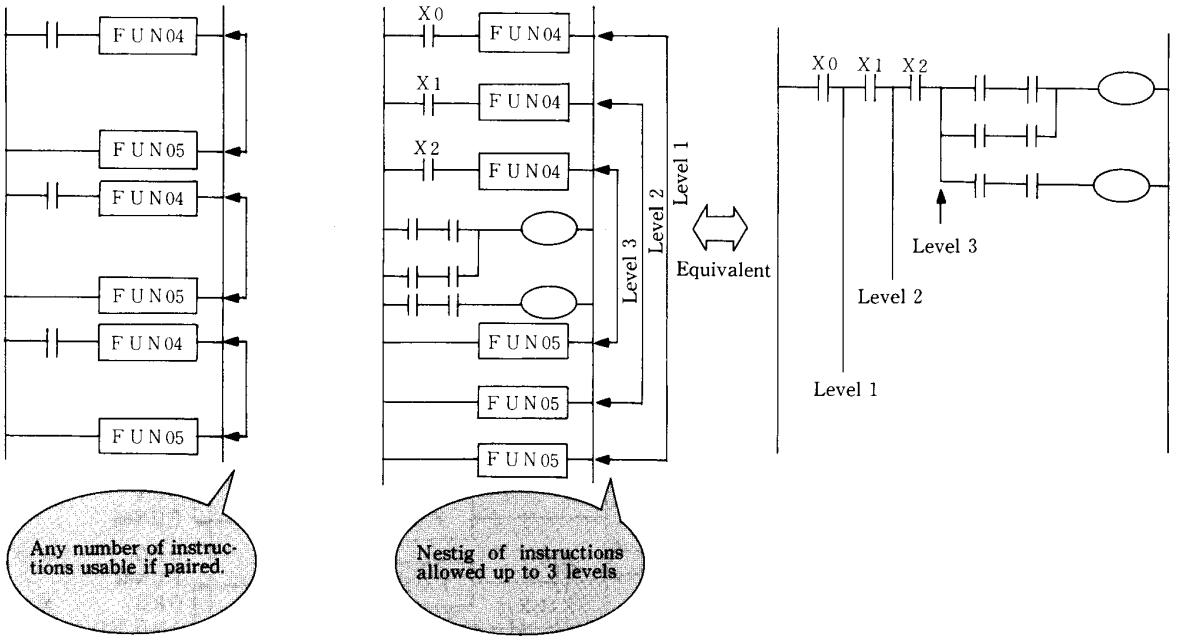
1. The FUN04 (MCS) and FUN05 (MCR) instructions are used for setting and resetting the common serial contacts, respectively. They must always be used as a pair. Otherwise, a syntax error occurs.
2. The FUN04 instructions must be followed by an ORG (or ORG NOT) instruction.
3. When the master control contacts are open, the subsequent output coil is set to OFF. In the example above, M420 is unconditionally OFF if input X0 is OFF.



NOTE

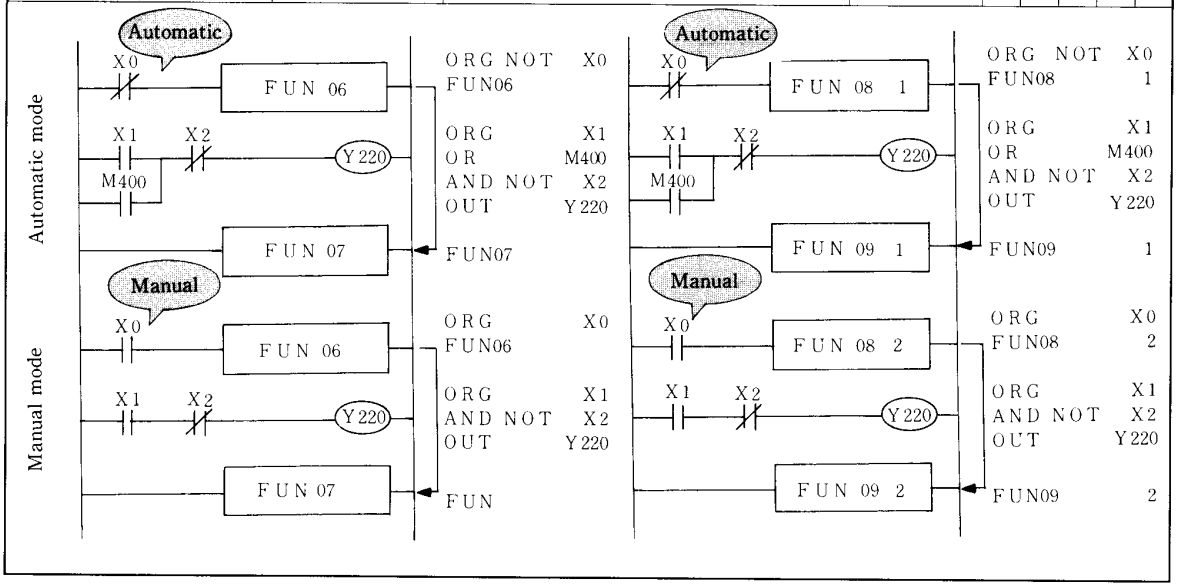
For switchover between manual and automatic controls according to a master control instruction, take care not to use double coils. If double coils must be used, specify the FUN06 or FUN07 instruction. Refer page 65 to use FUN06 and FUN07.

4. Any number of master control instructions can be used if they are paired unless nesting.



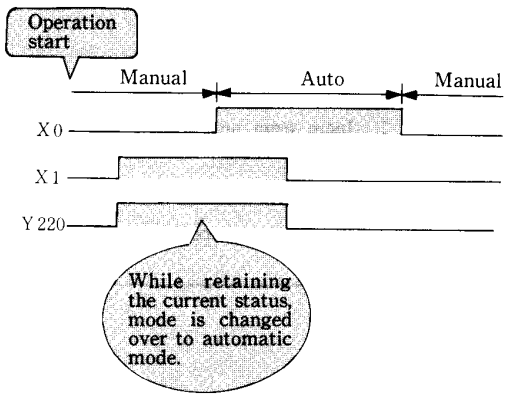
Instructions can be nested up to 3 levels. At four levels or more, syntax error will occur.

Instruction	Symbol	Name	Function	Component	No. of words	Change in register AR	ER	C	Acc
FUN06	JMP	Jump without addressing	Skips program till JEND.	None	1	·	·	·	·
FUN07	JEND				1	·	·	·	·
FUN08	AJMP	Jump with addressing	Jums to AJEND of corresponding address number.	Address No. (0 ~ 63)	2	·	·	·	·
FUN09	AJEND				2	·	·	·	·



[**Explanation**]

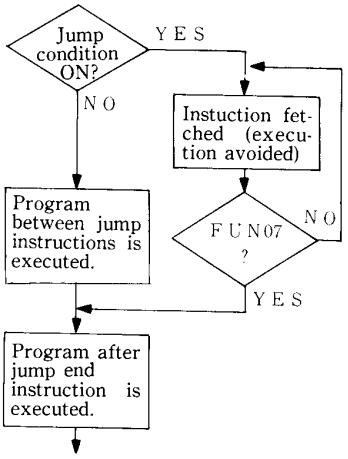
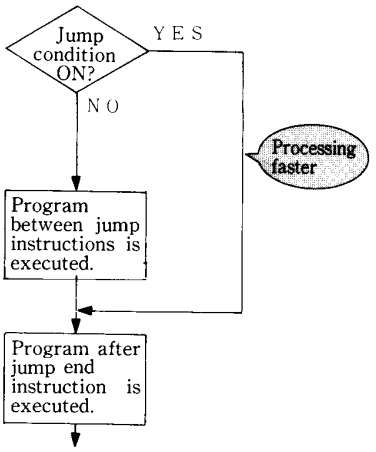
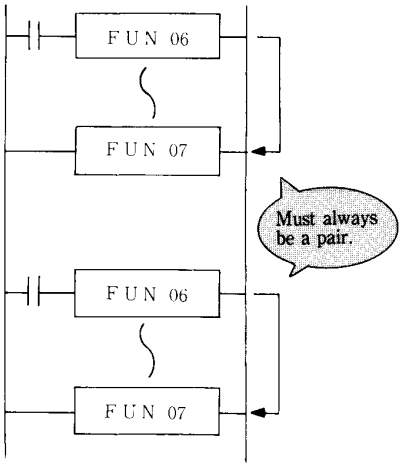
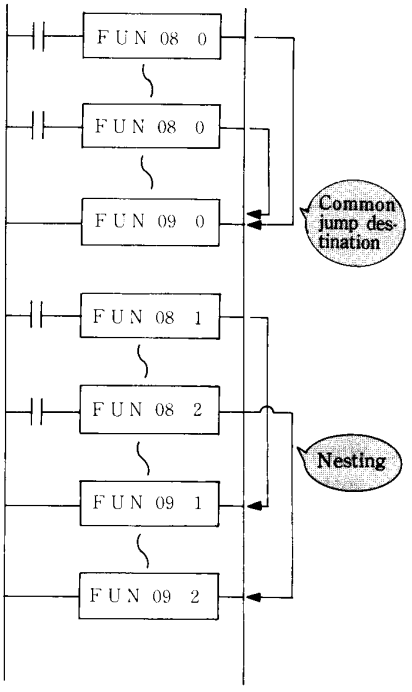
1. The FUN06 and FUN07 instructions specify jump without addressing, while the FUN08 and FUN 09 instructions specify jump with addressing. These instructions all cause cotrol to jump to JUMP END when the jump condition is set to ON.
2. When the jump conditions are satisfied, the program lines located between the current address and destination address are not executed. The output is held in the status before the jump. By using this function, a manual/auto switching circuit can be composed as illustrated above. If the same output coil is programmed between the jump circuits, a syntax error (double coil error E.) occurs, but operation can continue.



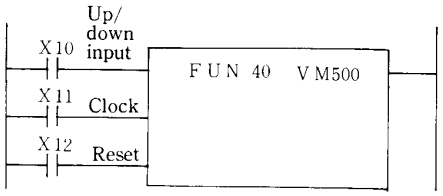
NOTE
If jump conditions are satisfied, the timer in the jump circuit stops operating. It restarts when the jump condetions are reset.

3. A jump instruction cannot be used between master control instructions.
4. The table below lists differences between the FUN06/07 instructions and FUN08/09 instructions. Scan time can be shortened by using the functional combination of FUN08/09.

Table: Differences between FUN06/07 and FUN08/09

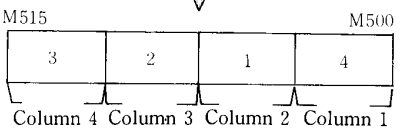
Item	FUN 06, FUN 07	FUN 08, FUN 09
Method of processing instruction		
Jump method	 1. These instructions must always be used as a pair. If not paired, a syntax error will occur. 2. Nesting is unallowable.	 1. Jump from multiple FUN08 insturcions to a single FUN09 instruction is allowed. 2. Nesting is allowed at different addresses. 3. Jump to a preceding step is also posible.

Instruction	Symbol	Name	Function	Component	No. of words	Change in register			
						AR	ER	C	Acc
FUN40	UDC	Up/down counter	Up/down counter	VM	1	·	·	·	·

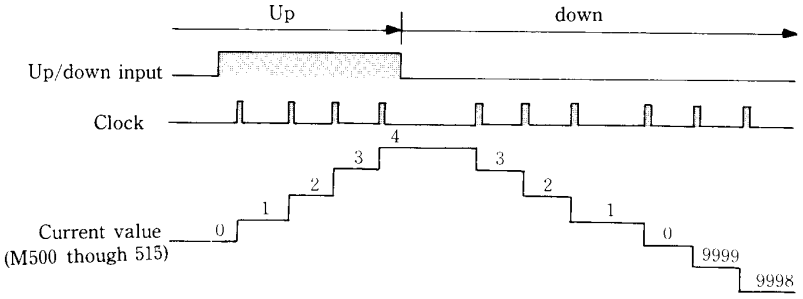


Code		Remarks
ORG	X10	
STR	X11	Use STR.
STR	X12	
FUN 40	VM500	UDC

16bits from M500 to M515are used for current value register.



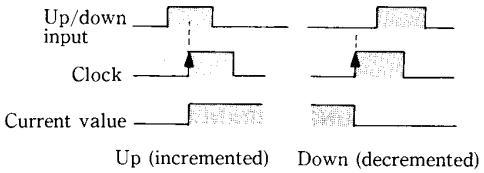
···· Indicates the current value of 3214 times (4-digit BCD value).



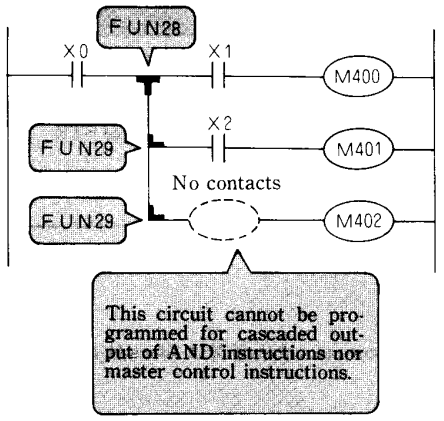
4.2

[Explanation]

1. FUN40 (UDC) is the up/down counter instruction. It is to be programmed in combination with an internal output (VM).
2. 16 bits staring from the coil number specified by that instruction (M500 through M515 in the example shown above) are used as the current value register of up/down counter. The current value is presented in BCD 4 digits.
3. The up/down input, clock input and reset input are programmed in that order.
   The current value changes at the rising edge of the clock (from OFF to ON). Either UP or DOWN condition is selected according to the ON or OFF status of up/down input as shown below.



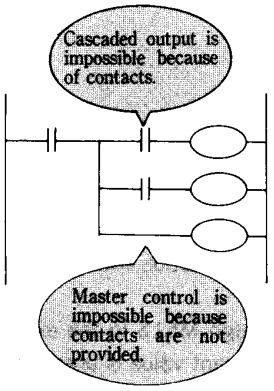
Instruction	Symbol	Name	Function	Component	No. of words	Change in register			
						AR	ER	C	Acc
FUN28	BRANCH	Branch	Stores Acc.	None	1	·	·	·	·
FUN29	RETURN	Return	Rturns stored Acc data .	None	1	·	·	·	↕



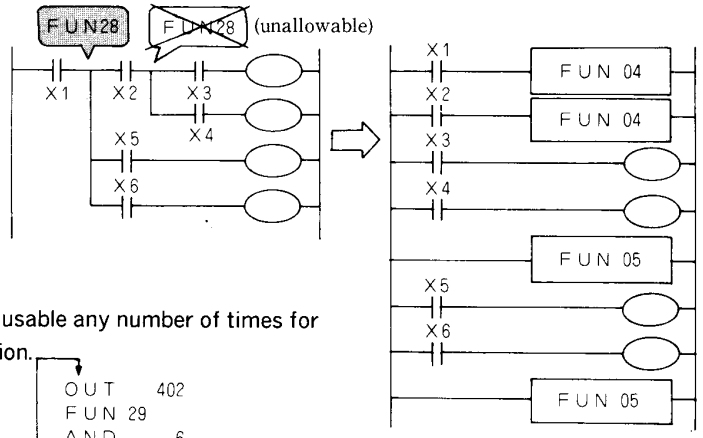
Code		Remarks
ORG	X0	
FUN 28		BRANCH
AND	X1	
OUT	M400	
FUN 29		RETURN
AND	X2	
OUT	M401	
FUN 29		RETURN
OUT	M402	

[**Explanation**]

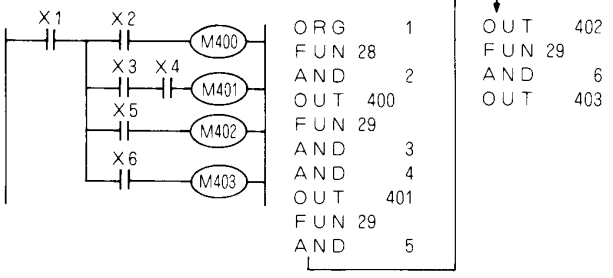
1. FUN28 (BRANCH) and FUN29 (RETURN) instructions allow programming of a circuit which is incompatible with the cascaded output of AND instruction and the master control instruction.



2. The FUN28 (BRANCH) instruction cannot be used more than once in the same circuit.
   For such a circuit below, master control instruction must be used.

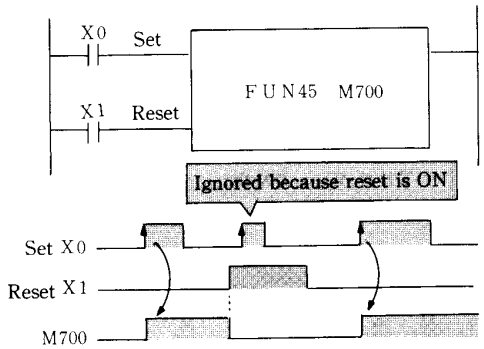


3. The FUN29 (RETURN) instruction is usable any number of times for a single FUN28 (BRANCH) instruction.



4. Master control instructions is unusable after FUN28 (BRANCH) instruction.

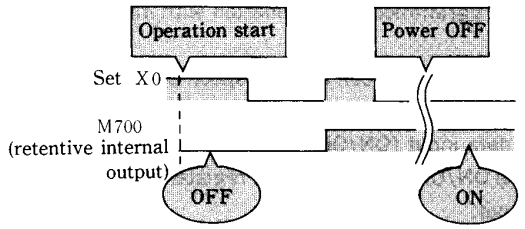
Instruction	Symbol	Name	Function	Component	No. of words	Change in register			
						AR	ER	C	Acc
FUN45	LATCH	Latch	Reset priority latch	M	1	·	·	·	·



Code		Remarks
ORG	X0	
STR	X1	Use STR.
FUN 45	M700	

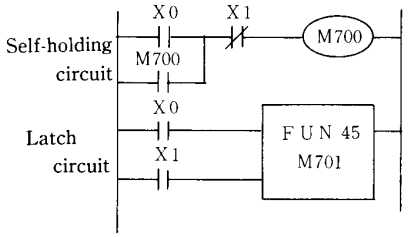
[**Explanation**]

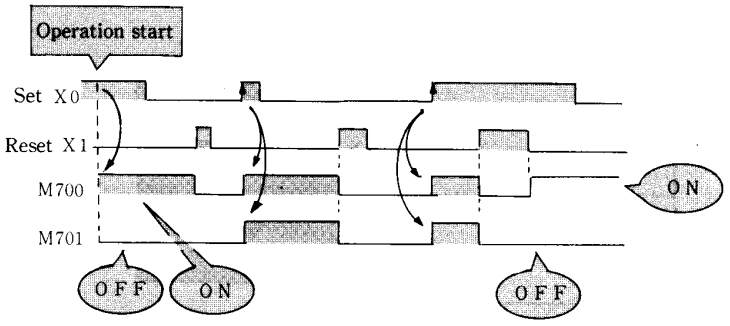
1. FUN45 (LATCH) is an edge triggered latch instruction with the reset priority signal. It must be programmed in combination with an internal output (M).
2. The ON status is set at the rising edge of the set input signal (from OFF to ON). The OFF status is set when the reset input goes ON. When the reset input is ON, the set input is rejected. If the set input and reset input go ON simultaneously, the reset input has a priority.
3. The FUN45 instruction can be combined with a retentive memory internal output (M) to produce the function of a keep relay.



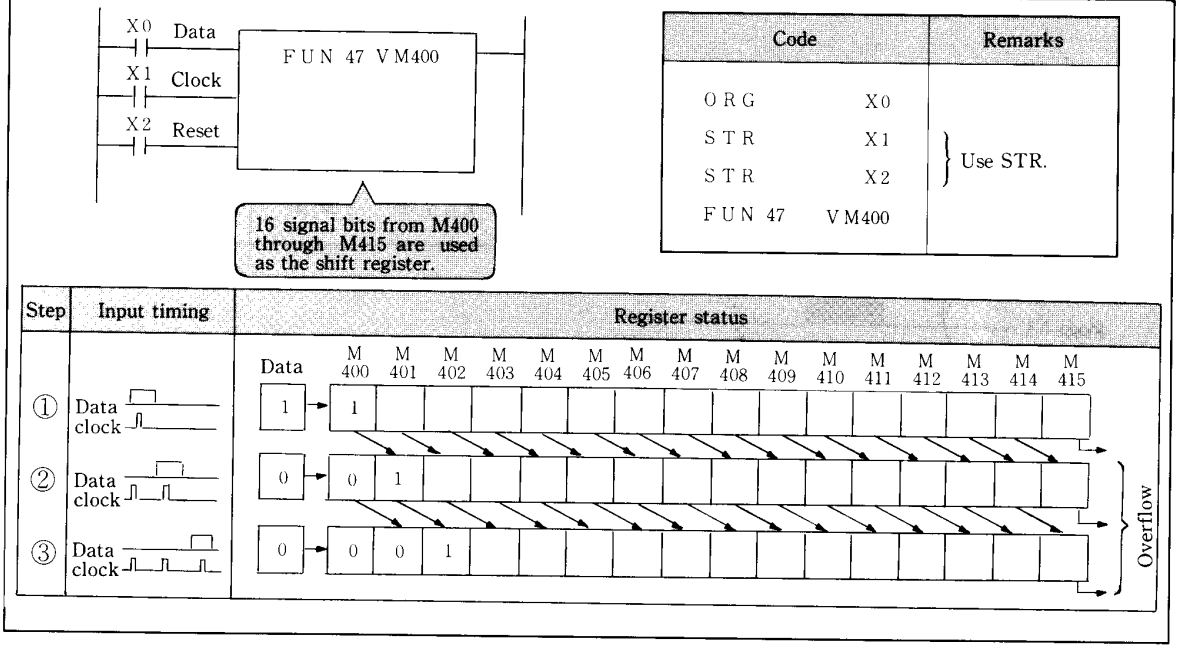
In the above sequence, the status of M700 at occurrence of power interruption is retained till its recovery because M700 is a retentive internal output.
Even when the set input X0 is turned ON at start of operation, edge will not be detected and M700 will not turn ON.

4. The self-holding circuit operates at a specific level (ON or OFF status), but the lath is operated at the signal edge. This causes the difference shown below.



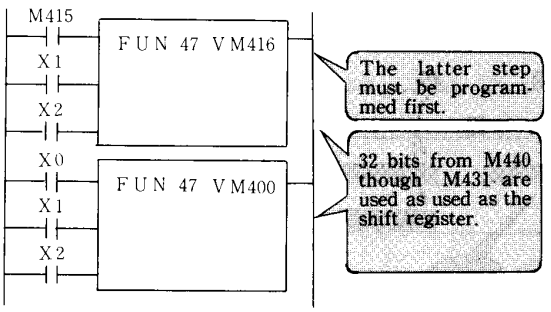


Instruction	Symbol	Name	Function	Component	No. of words	Change in register			
						AR	ER	C	Acc
FUN47	SFR	Shift register	16-bit shift register	VM	1	·	·	·	·

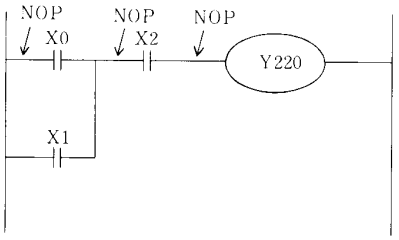


[Explanation]

1. FUN47 (SFR) is the shift register instruction. It is to be programmed in combination with internal output (VM).
2. 16 bits (M400 through M415 in the example shown above), from the coil number specified by the FUN47 instruction and higher, are used as the register.
3. When the clock input rises (from OFF to ON), concurrent data input (ON/OFF status) is written in the least significant bit of the register (M400 in this example). The ON/OFF status of each register is shifted to the next high-order bit synchronized with the rise of clock input.
4. Data of most significant bit (M415 in this example) may overflow as a result of shift operation. When connecting two (2) or more shift registers, the latter step (with a larger I/O number) must be programmed first in order to prevent data being lost due to overflow.



Instruction	Symbol	Name	Function	Component	No. of words	Change in register			
						AR	ER	C	Acc
FUN41	NOP	NOP	No operation	None	1	·	·	·	·



Code		Remarks
FUN41		NOP
ORG	X0	
OR	X1	
FUN41		NOP
AND	X2	
FUN41		NOP
OUT	Y220	

[Explanation]

1. FUN41 is NOP instruction. This instruction does not cause any execution in its step. It may be located anywhere in a program.

Arithmetic Instructions (1/2)

Classification	Instruction	Abbreviation	Name	Function	Component	No. of words	Change in register AR	ER	CR	Acc	Reference page
Load	FUN 0	LOADI	Load	Constant→AR	Constant (0000H～9999H)	2	↕	•	•	•	77
	FUN10	LOADW		I/O→AR	WX,WY,WM,T/C100～295	2	↕	•	•	•	77
	FUN20	LOADB		I/O→AR	VX,VY,VM,T/C0～95	2	↕	•	•	•	77
	FUN50	LBYTI		1 byte constant → AR_L (lower 8 bits)	Constant (00～FF)	2	↕	•	•	•	77
	FUN60	BLOAD		I/O→AR_L(lower 8 bits)	WX,WY,WM,T/C100～295	2	↕	•	•	•	77
Out	FUN21	OUTW	Out	AR→I/O	WY,WM,T/C100～295	2	•	•	•	•	79
	FUN22	OUTB		AR→I/O	VY,VM	2	•	•	•	•	79
	FUN71	BOUT		AR_L→I/O	WY,WM,T/C100～295	2	•	•	•	•	79
Add	FUN 1	ADDI	BCD add	AR B+constant→AR	Constant (0000H～9999H)	2	↕	•	↕	•	81
	FUN11	ADD		AR B+I/O→AR	WX,WY,WM,T/C100～295	2	↕	•	↕	•	81
	FUN51	ABYTI	BIN add	AR+constant→AR	Constant (0～FFFF)	2	↕	•	↕	•	81
	FUN61	ADBNR		AR+I/O→AR	WX,WY,WM,T/C100～295	2	↕	•	↕	•	81
Subtract	FUN 2	SUBI	BCD subtract	AR B−constant→AR	Constant (0000H～9999H)	2	↕	•	↕	•	82
	FUN12	SUB		AR B−I/O→AR	WX,WY,WM,T/C100～295	2	↕	•	↕	•	82
	FUN52	SBYTI	BIN subtract	AR−constant→AR	Constant (0～FFFF)	2	↕	•	↕	•	82
	FUN62	SUBNR		AR−I/O→AR	WX,WY,WM,T/C100～295	2	↕	•	↕	•	82
Multiply	FUN 3	MULI	BCD multiply	AR B*constant→AR	Constant (0000H～9999H)	2	↕	•	↕	•	83
	FUN13	MUL		AR B*I/O→AR	WX,WY,WM,T/C100～295	2	↕	•	↕	•	83
	FUN53	MBYTI	BIN multiply	AR*constant→AR	Constant (0～FFFF)	2	↕	↕	↕	•	83
	FUN63	MUBNR		AR*I/O→AR	WX,WY,WM,T/C100～295	2	↕	↕	↕	•	83
Divide	FUN 4	DIVI	BCD divide	AR B/constant→AR	Constant (0000H～9999H)	2	↕	•	↕	•	84
	FUN14	DIV		AR B/ I/O→AR	WX,WY,WM,T/C100～295	2	↕	•	↕	•	84
	FUN54	DBYTI	BIN divide	AR/constant→AR	Constant (0～FFFF)	2	↕	↕	↕	•	84
	FUN64	DIBNR		AR/I/O→AR	WX,WY,WM,T/C100～295	2	↕	↕	↕	•	84
Logic	FUN 5	ANDI	AND	AR AND constant→AR	Constant (0000H～9999H)	2	↕	•	•	•	85
	FUN15	AND		AR AND I/O→AR	WX,WY,WM,T/C100～295	2	↕	•	•	•	85
	FUN55	BANDI		AR_L AND constant→AR_L	Constant (00～FF)	2	↕	•	•	•	85
	FUN 6	ORI	OR	AR OR constant→AR	Constant (0000H～9999H)	2	↕	•	•	•	85
	FUN16	OR		AR EOR I/O→AR	WX,WY,WM,T/C100～295	2	↕	•	•	•	85
	FUN56	BORI		AR_L OR constant→AR_L	Constant (00～FF)	2	↕	•	•	•	85
	FUN66	EXOR	Exclusive-OR	AR EOR I/O→AR	WX,WY,WM,T/C1100～295	2	↕	•	•	•	85
	FUN85	WNOT	Logical not	$\overline{AR}$→AR	None	1	↕	•	•	•	85
Compare	FUN 7	CPEHI	Compare(≧)	AR ≧ constant……1→C	Constant (0000H～9999H)	2	•	•	↕	•	86
	FUN17	CPEH		AR≧I/O……1→C	WX,WY,WM,T/C100～295	2	•	•	↕	•	86
	FUN57	BCPHI		AR_L≧constant……1→C	Constant (00～FF)	2	•	•	↕	•	86
	FUN 8	CPEI	Compare(=)	AR=constant……1→C	Constant (0000H～9999H)	2	•	•	↕	•	86
	FUN18	CPE		AR=I/O……1→C	WX,WY,WM,T/C100～295	2	•	•	↕	•	86
	FUN58	BCPEI		AR_L=constant……1→C	Constant (00～FF)	2	•	•	↕	•	86
	FUN 9	CPLI	Compare(<)	AR<constant……1→C	Constant (0000H～9999H)	2	•	•	↕	•	86
	FUN19	CPL		AR<I/O……1→C	WX,WY,WM,T/C100～295	2	•	•	↕	•	86
	FUN59	BCPLI		AR_L<constant……1→C	Constant (00～FF)	2	•	•	↕	•	86
Carry	FUN23	OUC	Out carry	C→I/O	Y,M	1	•	•	•	•	87
	FUN83	CLC	clear carry	C←"0"	None	1	•	•	0	•	87
	FUN84	SEC	Set carry	C←"1"	None	1	•	•	1	•	87
Convert	FUN24	BCD	BCD convert	BCD convert	None	1	↕	•	↕	•	88
	FUN25	BNR	BIN convert	BIN convert	None	1	↕	•	↕	•	88
	FUN74	SEG	7-segment convert	Decodes AR_{LL} data into 7-segment display.	None	1	↕	•	•	•	89
	FUN75	ASC	ASCII convert	Converts AR_{LL} data into ASCII code.	None	1	↕	•	•	•	89
	FUN78	ENCOD	Encode	16 to 4	None	1	↕	•	↕	•	90
	FUN79	DECOD	Decode	4 to 16	None	1	↕	•	↕	•	90

Arithmetic Instructions (2/2)

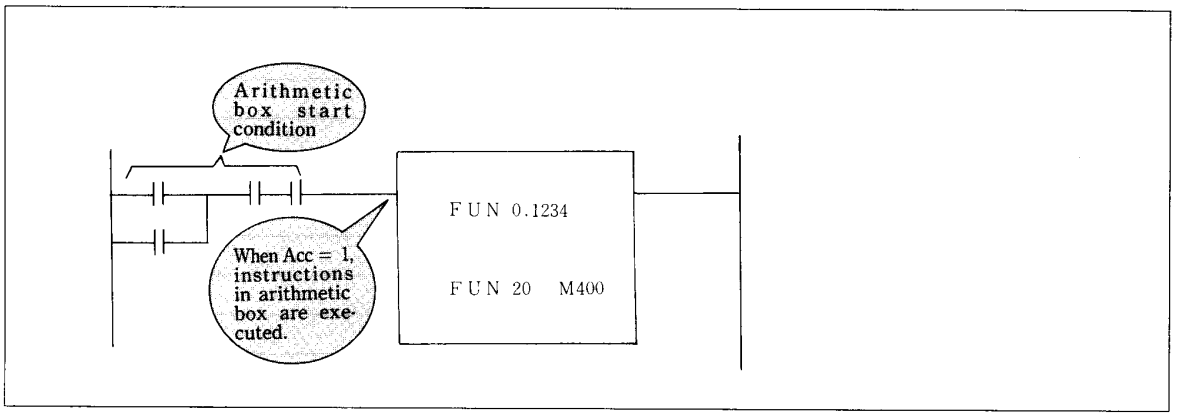
Classification	Instruction	Abbreviation	Name	Function	Component	No. of words	Change in register AR	ER	CR	Acc	Reference page
Shift	FUN26	LSFR	Shift	Left shift	None	1	↕	•	↕	•	91
	FUN27	RSFR		Right shift	None	1	↕	•	↕	•	91
	FUN76	ROL	Rotate	CW rotate	None	1	↕	•	↕	•	91
	FUN77	ROR		CCW rotation	None	1	↕	•	↕	•	91
Mask	FUN72	MASKL	Left mask	Masks AR by specified bits from left.	0~255	2	↕	•	•	•	92
	FUN73	MASKR	Right mask	Masks AR by specified bits from right.	0~255	2	↕	•	•	•	92
Exchange	FUN80	SWAP	AR_H/AR_L exchange	$AR_H \rightleftarrows AR_L$	None	1	↕	•	•	•	93
	FUN81	BSWAP	AR_{LH}/AR_{LL} exchange	$AR_{LH} \rightleftarrows AR_{LL}$	None	1	↕	•	•	•	93
	FUN82	XCG	AR/ER exchange	$AR \rightleftarrows ER$	None	1	↕	↕	•	•	93
Distribute /extract	FUN48	EX	Extract	Fetches data into AR from I/O address-specified by ER.	None	1	↕	•	↕	•	94
	FUN49	DB	Distribute	Outputs data from AR to I/O address-specified by ER.	None	1	•	•	↕	•	94

●: Register remains unchanged
↕: Register changed

Note: Although the table above contains entries of constants (00~FF) and (0~FFFF), the programmer is not provided with A to F keys which are indispensable for specification of hexadecimal constants.
Therefore, specification must be made in decimal constants.
Besides, each constant can be entered in up to 3 digits.

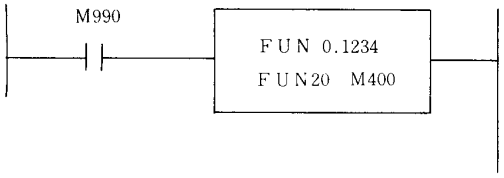
Example: FUN51 427 AR + 1ABH → AR
(Decimal 427 = hexadecimal 1ABH)

Concept of arithmetic instruction	Load	out	4-rule calculations				Logic	Compare	Carry	Convert	Shift	Mask	Exchange	Distribute/ extract
			Add	Subtract	Multiply	Divide								
76	77	79	81	82	83	84	85	86	87	88	91	92	93	94

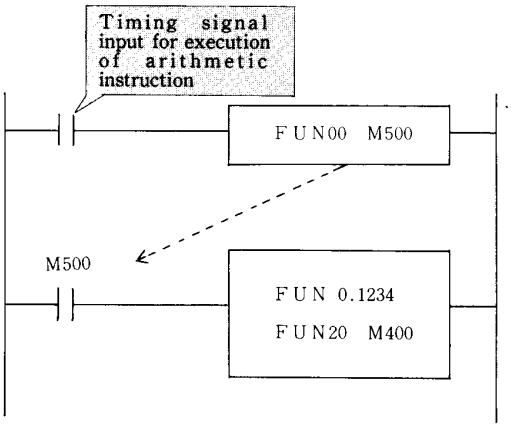


[Explanation]

1. Arithmetic instruction is assumed to be contained in the arithmetic box,and consecutive arithmetic instructions are put in the same arithmetic box. Before each arithmetic box, start condition is to be provided. When the start condition is satisfied (Acc=1), arithmetic instruction in the arithmetic box is excuted. This won't occur if the start condition is not satisfied (Acc=0), and the previous status is retained.
2. For an arithmetic instruction to be executed every scan, it is recommended to use the special internal output M990 for the start condition since ON status is always secured.

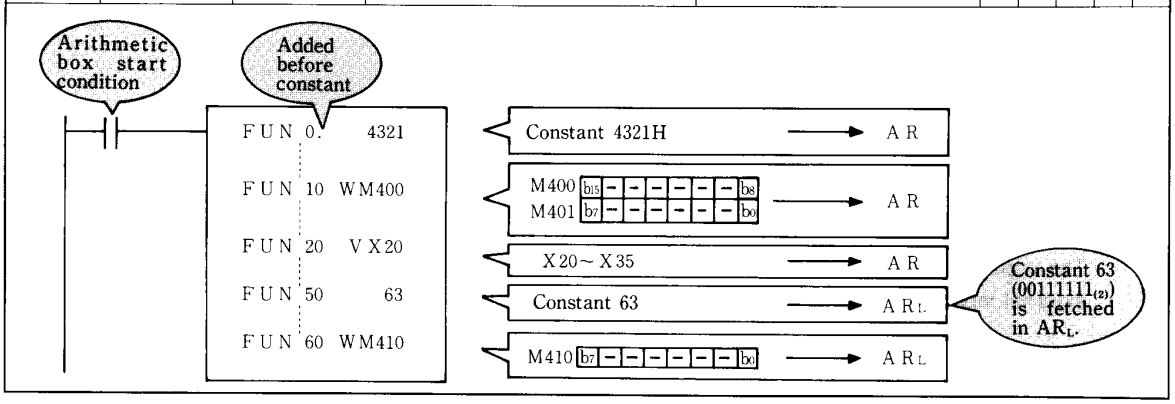


3. For an arithmetic instruction to be executed for only one scan at a certain timing, it is recommended to use the edge instruction as the start condition.



Concept of arithmetic instruction	Load	out	4-rule calculations				Logic	Compare	Carry	Convert	Shift	Mask	Exchange	Distribute/ extract
			Add	Subtract	Multiply	Divide								
76	77	79	81	82	83	84	85	86	87	88	91	92	93	94

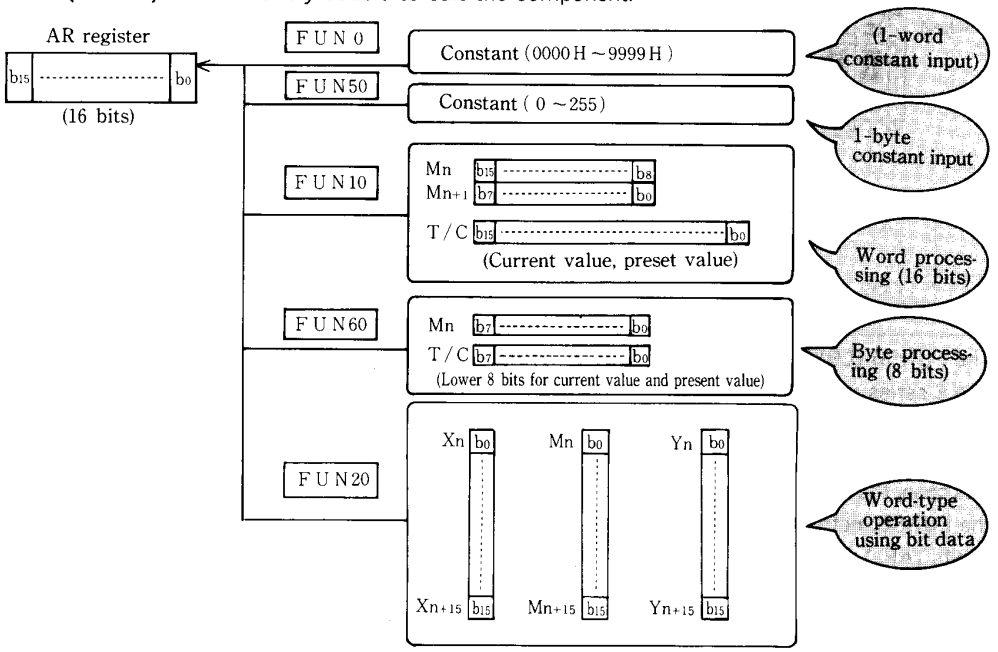
Instruction	Abbreviation	Name	Function	Component	No. of words	Change in register			
						AR	ER	CR	Acc
FUN 0	LOADI	Load	Constant → AR	Constant (0000H ~ 9999H)	2	↕	·	·	·
FUN10	LOADW		I/O → AR	WX, WY, WM, T/C100 ~ 295	2	↕	·	·	·
FUN20	LOADB		I/O → AR	VX, VY, VM, T/C0 ~ 95	2	↕	·	·	·
FUN50	LBYTI		1 byte constant → AR_L (lower 8 bits)	Constant (00 ~ FF)	2	↕	·	·	·
FUN60	BLOAD		I/O → AR_L (lower 8 bits)	WX, WY, WM, T/C100 ~ 295	2	↕	·	·	·



4.3

[Explanation]

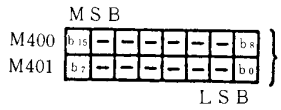
1. Load instruction loads the word data (16 bits) or byte data (8 bits) to be processed into the AR register. Five kinds of load instructions FUN0 (LOAD1), FUN10 (LOADW), FUN20 (LOADB), FUN50 (LBYT1) and FUN60 (BLOAD) are selectively usable to suit the component.



(1) The FUN0 (LOADI) instruction loads a one-word constant (0000H to 9999H) into the AR register. The constant must be preceded by a decimal point (.) when keying in.

(2) The FUN10 (LOADW) instruction loads one-word I/O data into the AR register.

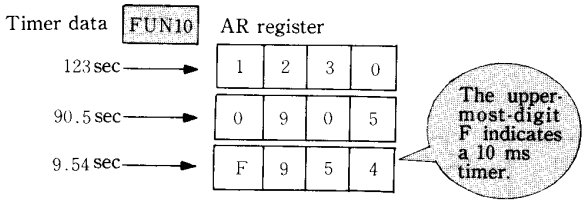
① Internal outputs are used for both bit and byte data (8-bit data for each number). 8 bit data of the specified internal output (Mn) and the next internal output (Mn+1), 16-bit data in total, are loaded into the AR register.



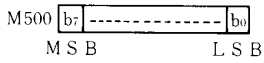
One 16-bit word consisting of data from M400 and M401 is used.

② The timer/counter current values (T/C100 to T/C195) and preset values (T/C200 to T/C295) are 4-digit BCD (16bits) data. The counter preset value and current value are loaded into the AR register without change. However, the timer value is processed as shown below before loaded into the AR register.

CAUTION

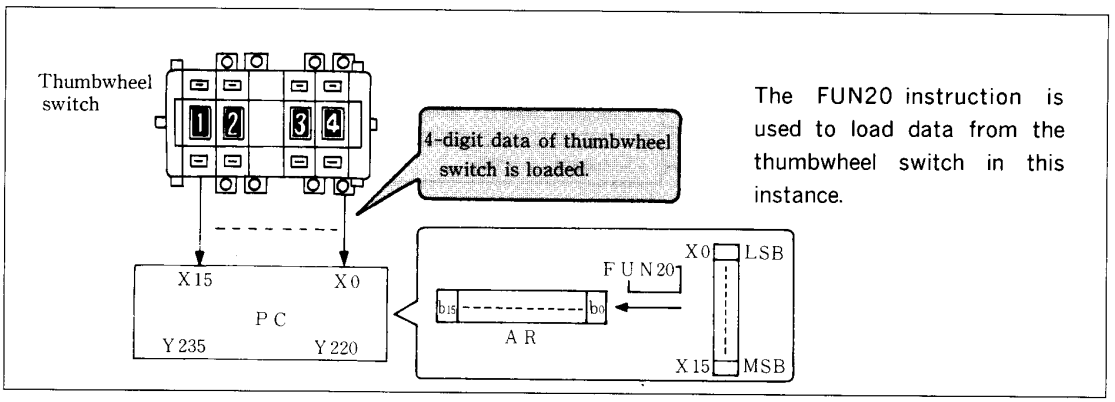


(3) The FUN60 (BLOAD) instruction loads 1-byte (8-bit) I/O data into the lower 8 bits (AR_L) of AR register. The upper 8 bits (AR_H) of AR register remain unchanged. The FUN60 instruction is used to load the external input of 8-bit analog module.



8-bit data of the specified number is processed.

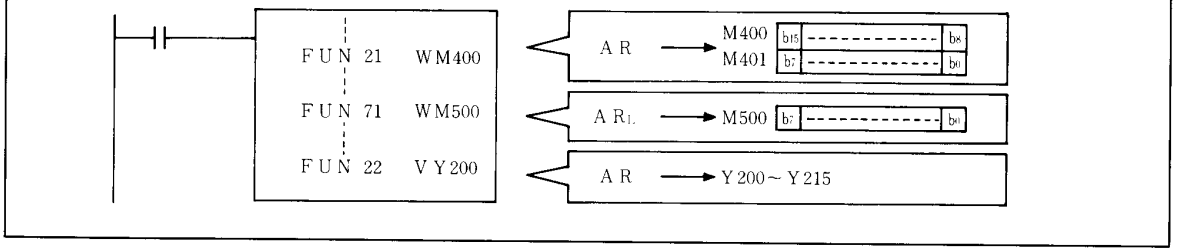
(4) The FUN20 (LOADB) instruction loads 16 I/O data simultaneously into the AR register. 16 data from the specified number and upward are loaded into the AR register.



The FUN20 instruction is used to load data from the thumbwheel switch in this instance.

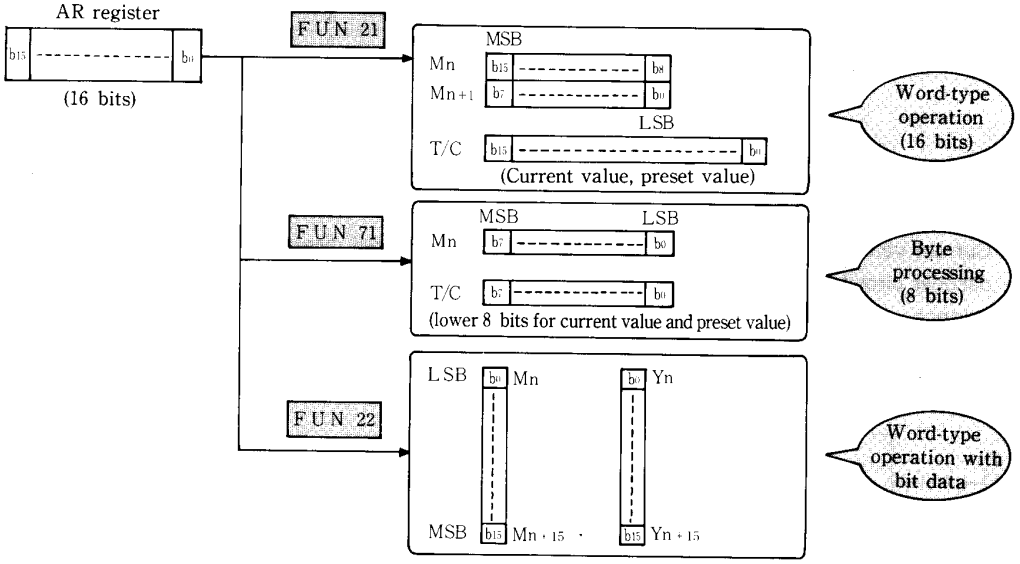
(5) The FUN50 (LBYT1) instruction loads a described bit pattern into the AR register.
The E-series programmers (PGMJ and PGM-R2) do not have keys A through F required for hexadecimal notation. However, when a decimal constant (0 to 255) is specified by the FUN50 instruction,it is handled as a one-byte data (00H to FFH) and loaded into the lower 8 bits (AR_L) of the AR register. In this case, the upper 8 bits (AR_H) of the same register remain unchanged. When used in combination with the FUN80 instruction, the FUN50 instruction is capable of loading a desired bit pattern into the upper 8 bits (AR_H).

Instruction	Abbreviation	Name	Function	Component	No. of words	Change in register			
						AR	ER	CR	Acc
FUN21	OUTW	Out	AR→I/O	WY, WM, T/C100~295	2	·	·	·	·
FUN71	BOUT		AR_L→I/O(8 bits)	WY, WM, T/C100~295	2	·	·	·	·
FUN22	OUTB		AR→I/O	VY, VM	2	·	·	·	·



[Explanation]

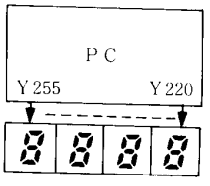
1. OUT instruction outputs data in the AR register to the destination component. Three kinds of OUT instructions below are selectively usable so as to meet the component.



(1) The FUN21 (OUTW) instruction outputs data in the AR register to the 16-bit area made up of the specified internal output (Mn) and the next internal output (Mn+1). This instruction is also used for outputting AR register data to current value (T/C100 through T/C195) or preset value (T/C200 to T/C295) of timer/counter.

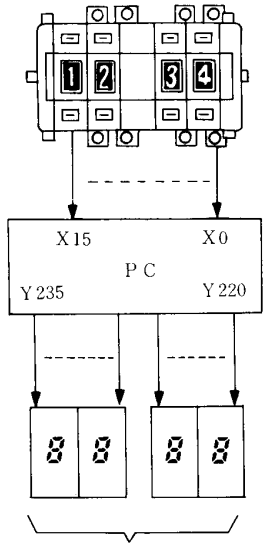
(2) The FUN71 (BOUT) instruction outputs the lower 8-bit data (AR_L) of AR register to the specified internal output (Mn). This instruction is used for analog output when the analog I/O module is mounted.

(3) The FUN22 (OUTB) instruction is used to output AR register data to the numerical display (7-segment LED).



[Application example of LOAD and OUT instructions]

1. Explanation of operations



Truth table of thumbwheel switch

Digit \ Switch terminal / PC terminal	8	4	2	1
	X 3	X 2	X 1	X 0
	X 7	X 6	X 5	X 4
	X11	X10	X 9	X 8
	X15	X14	X13	X12
0				
1				●
2			●	
3			●	●
4		●		
5		●		●
6		●	●	
7		●	●	●
8	●			
9	●			●

Thumbwheel switch dial

Indicates terminal wiring.

● ON

Truth table of display unit with decoder circuit

Digit \ Display terminal / PC terminal	D	C	B	A
	Y223	Y222	Y221	Y220
	Y227	Y226	Y225	Y224
	Y231	Y230	Y229	Y228
	Y235	Y234	Y233	Y232
0				
1				●
2			●	
3			●	●
4		●		
5		●		●
6		●	●	
7		●	●	●
8	●			
9	●			●

Numerical display unit

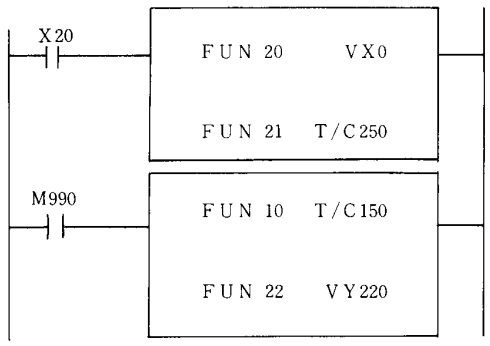
Indicates terminal wiring.

● ON

(1) Preset value of the counter in PC is set when X20 turns ON with a 4-digit BCD thumbwheel switch connected to the PC external input terminal.

(2) Current value of the counter in PC is output to the 7-segmnet display unit. This unit is provided with a decoder circuit.

2. Sequence



X0～X15→AR

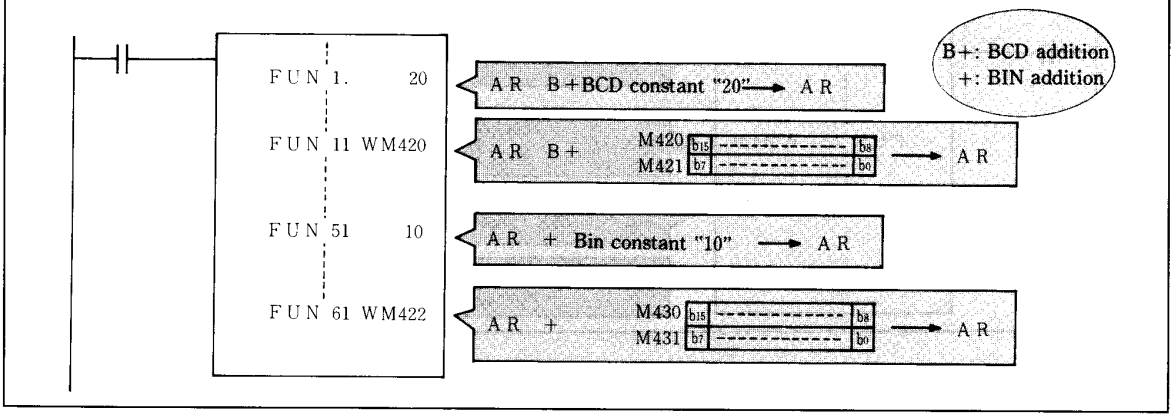
AR→preset value of T/C50 (T/C 250)

Current value of T/C50 (T/C150) →AR

AR→Y220～Y235

Concept of arithmetic instruction	Load	out	4-rule calculations				Logic	Compare	Carry	Convert	Shift	Mask	Exchange	Distribute/ extract
			Add	Subtract	Multiply	Divide								
76	77	79	81	82	83	84	85	86	87	88	91	92	93	94

Instruction	Abbreviation	Name	Function	Component	No. of words	Change in register			
						AR	ER	CR	Acc
FUN 1	ADDI	BCD add	AR B+ constant→AR	Constant (0000H~9999H)	2	↕	·	↕	·
FUN11	ADD	BCD add	AR B+ I/O→AR	WX, WY, WM, T/C100~295	2	↕	·	↕	·
FUN51	ABYTI	BIN add	AR + constant→AR	Constant (0~FFFF)	2	↕	·	↕	·
FUN61	ADBNR	BIN add	AR + I/O→AR	WX, WY, WM, T/C100~295	2	↕	·	↕	·



4.3

[Explanation]

1. ADD instructions add AR register data to component data and load the sum to the AR register. There are two kinds of ADD instructions; BCD and BIN ADD instructions, each of which consists of paired instructions for selective use depending on whether the compontent data is a constant or I/O.
2. When the sum is more than four (4) digits, the carry C turns OFF. In this case, instruction is handled as shown below.

Conditions	Instruction	AR	C	Remarks
Sum has exceeded 4 digits.	FUN 1	Remain unchanged	1	Carry C indicates occurrence of error.
	FUN 11		1	
	FUN 51	Sum of 4 digits or less loaded	1	Carry C indicates occurrence of a carry.
	FUN 61		1	

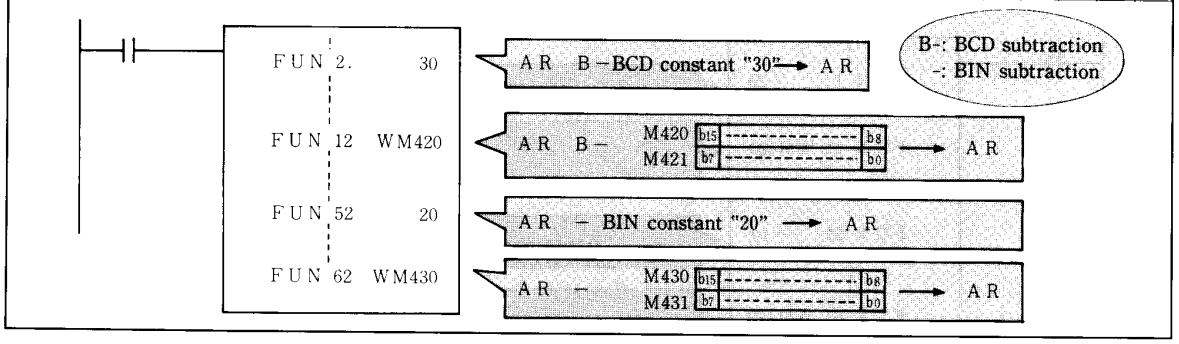
3. If a non-BCD data is handled by the FUN1 or FUN11 instruction, neither AR register nor carry C data is assured. The table below lists example programs for different components.

Classification	Component		Program	Explanation
BCD addition	Constant		FUN 1. 4321	AR B+ BCD constant 4321 →AR
	Internal output		FUN 11 WM500	AR B+ WM 500 →AR
	Timer counter	Current value	FUN 11 T/C150	AR B+ T/C50 current value →AR
		Preset value	FUN 11 T/C250	AR B+ T/C50 preset value →AR
Bin addition	Constant		FUN 51 735	AR + Bin constant 735 →AR
	Internal output		FUN 61 WM422	AR + WM 422 →AR

Note: In case of FUN51 (ABYTI), a decimal entry is automatically converted into a hexadecimal value before addition because of the restriction peculiar to the programmer. In addition, entry is possible only up to 3 digits. For instance, entry $(735)_{10}$ is converted into $(2DF)_{16}$.

Concept of arithmetic instruction	Load	out	4-rule calculations				Logic	Compare	Carry	Convert	Shift	Mask	Exchange	Distribute/ extract
			Add	Subtract	Multiply	Divide								
76	77	79	81	82	83	84	85	86	87	88	91	92	93	94

Instruction	Abbreviation	Name	Function	Component	No. of words	Change in register			
						AR	ER	CR	Acc
FUN 2	SUBI	BCD subtract	AR B− constant →AR	Constant (0000H~9999H)	2	↕	·	↕	·
FUN12	SUB		AR B− I/O→AR	WX, WY, WM, T/C100~295	2	↕	·	↕	·
FUN52	SBYTI	Bin subtract	AR − constant →AR	Constant (0~FFFF)	2	↕	·	↕	·
FUN62	SUBNR		AR − I/O→AR	WX, WY, WM, T/C100~295	2	↕	·	↕	·



[Explanation]

1. SUB instructions subtract component data from AR register data and load the difference to the the AR register.
   There are two kinds of SUB instructions; BCD and BIN SUB instructions, each of which consints of paired instructions for constant and I/O, respectively.
2. When subtraction results in 0 or a positive value, the carry C turns OFF. If the difference of subtraction is negative, each instruction is handled as listed below.

Condition	Instruction	AR	C	Remarks
Difference is negative.	FUN 2	Remain unchanged	1	Carry C indicates occurrence of error.
	FUN 12		1	
	FUN 52	Difference loaded (expressed in two's complement)	1	Carry C indicates decrement to next lower digit.
	FUN 62		1	

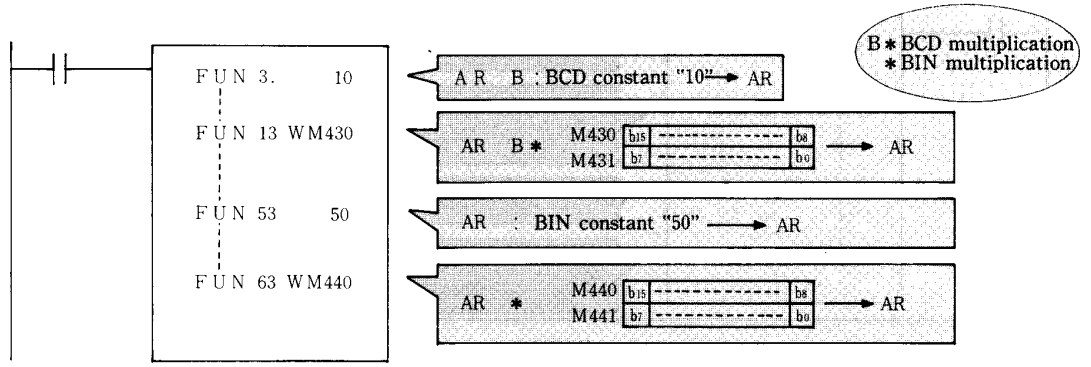
3. If a non-BCD data is handlded by the FUN2 or FUN12 instruction, neither AR register data nor carry C data is reliable. The table below lists example programs for different components.

Classification	Component		Program	Explanation
BCD subtraction	Constant		FUN 2. 4321	AR B− BCD constant 4321 →AR
	Internal output		FUN 12 WM500	AR B− WM 500 →AR
	Timer/ counter	Current value	FUN 12 T/C150	AR B− T/C 50 current value →AR
		Preset value	FUN 12 T/C250	AR B− T/C 50 preset value →AR
Bin subtraction	Constant		FUN 52 735	AR −BIN constant 735→AR
	Internal output		FUN 62 WM510	AR − WM510 →AR

Note: In case of FUN52 (SBYTI), a decimal entry is automatically converted into a hexadecimal value before subtraction because of the restriction peculiar to the programmer. In addition, entry is possible only up to 3 digits. For instance, entry $(735)_{10}$ is converted into $(2DF)_{16}$.

Concept of arithmetic instruction	Load	out	4-rule calculations				Logic	Compare	Carry	Convert	Shift	Mask	Exchange	Distribute/ extract
			Add	Subtract	Multiply	Divide								
76	77	79	81	82	83	84	85	86	87	88	91	92	93	94

Instruction	Abbreviation	Name	Function	Component	No. of words	Change in register			
						AR	ER	CR	Acc
FUN 3	MULI	BCD multiply	AR B* constant →AR	Constant (0000H~9999H)	2	↕	·	↕	·
FUN13	MUL		AR B* I/O→AR	WX, WY, WM, T/C100~295	2	↕	·	↕	·
FUN53	MBYTI	BIN multiply	AR * constant →AR	Constant (0~FFFF)	2	↕	↕	↕	·
FUN63	MUBNR		AR * I/O→AR	WX, WY, WM, T/C100~295	2	↕	↕	↕	·



4.3

[Explanation]

1. MUL instructions multiply AR register data with component data and load the product to the AR register. There are two kinds of MUL instructions; BCD and BIN MUL instructions, each of which consists of paired instructions for constant and I/O, respectively.
2. When multiplication results in 4 digits or less, the carry C turns OFF. If the product of multiplication exceeds 4 digits, each instruction is handled as listed below.

Condition	Instruction	AR	ER	C	Remarks
Product exceeds 4 digits.	FUN 3	Remain unchanged	Remain unchanged	1	Carry C indicates occurrence of error.
	FUN 13			1	
	FUN 53	4th digit and lower of product	5th digit and upper of product	1	Carry C indicates the product reaches 5 digits.
	FUN 63			1	

3. If a non-BCD data is handled by the FUN3 or FUN13 instruction, neither AR register data nor carry C data is reliable. The table below lists example programs for different components.

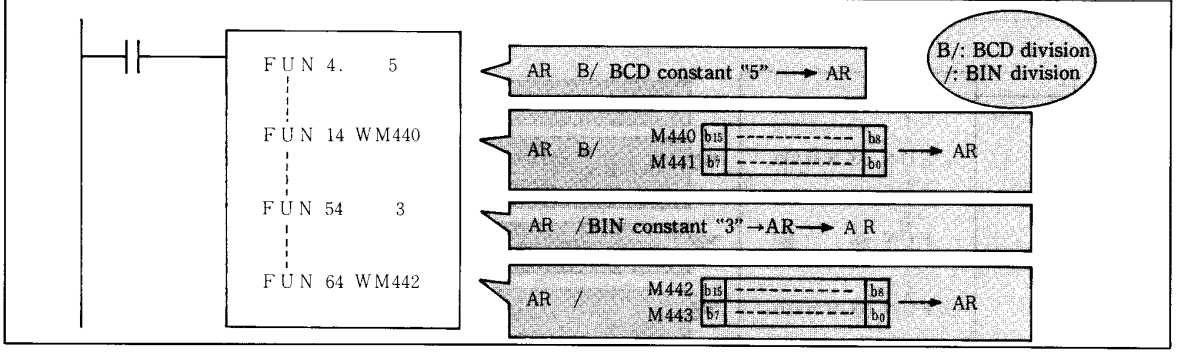
Classification	Component		Program	Explanation
Bin multiplication	Constant		FUN 3. 4321	AR B* BCD constant 4321 →AR
	Internal output		FUN 13 WM500	AR B* WM500 →AR
	Timer/ counter	Current value	FUN 13 T/C150	AR B* T/C50 current value →AR
		Preset value	FUN 13 T/C250	AR B* T/C50 preset value →AR
BCD multiplication	Constant		FUN 53 735	AR * BIN constant 735 →AR
	Internal output		FUN 63 WM510	AR * WM510 →AR

Note: In case of FUN52 (SBYTI), a decimal entry is automatically converted into a hexadecimal value before multiplication because of the restriction peculiar to the programmer. In addition, entry is possible only up to 3 digits. For instance, entry $(735)_{10}$ is converted into $(2DF)_{16}$.

Concept of arithmetic instruction	Load	out	4-rule calculations				Logic	Compare	Carry	Convert	Shift	Mask	Exchange	Distribute/ extract
			Add	Subtract	Multiply	Divide								

76	77	79	81	82	83	84	85	86	87	88	91	92	93	94

Instruction	Abbreviation	Name	Function	Component	No. of words	Change in register			
						AR	ER	CR	Acc
FUN4	DIVI	BCD divide	AR B/ constant →AR	Constant (0000H~9999H)	2	↕	·	↕	·
FUN14	DIV		AB B/ I/O→AR	WX, WY, WM, T/C100~295	2	↕	·	↕	·
FUN54	DBYTI	BIN divide	AR / constant →AR	constant (0~FFFF)	2	↕	↕	↕	·
FUN64	DIBNR		AR / I/O→AR	WX, WY, WM, T/C100~295	2	↕	↕	↕	·



[Explanation]

1. DIV instructions devide AR register with component data and load the quotient to the AR register. There are two kinds of DIV instructions; BCD and BIN divide instructions, each of which consists of paired instructions for constant and I/O, respectively.
2. Each DIV instruction is handled as listed below in cases of usual division and 0 division.

Condition	Instruction	AR	ER	C	Remarks
Usual division	FUN 4	Quotient	Remain unchanged.	0	Remainder is neglected.
	FUN 14			0	
	FUN 54	Quotient	Carry C indicates occurrence of error.	0	Remainder is loaded in ER.
	FUN 64			0	
÷0	FUN 4	Remain unchanged	Remain unchanged.	1	Carry C indicates occurrence of error.
	FUN 14			1	
	FUN 54			1	
	FUN 64			1	

3. If non-BCD constant is handled in in the FUN4 or FUN14 instruction, neither AR register data nor carry C data is reliable.

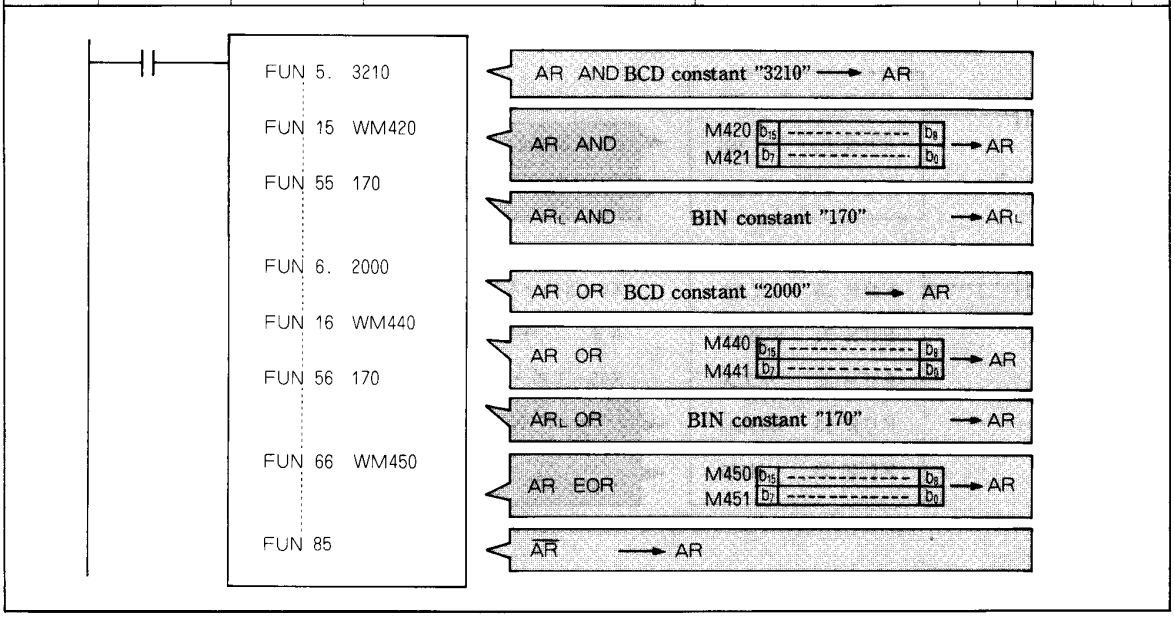
The table below lists example programs for different components.

Classification	Component		Program	Explanation
BCD division	Constant		FUN 4. 5	AR B/ BCD constant 5 →AR
	Internal output		FUN 14 WM500	AR B/ WM 500 →AR
	Timer/ counter	Current value	FUN 14 T/C150	AR B/ T/C50 current value →AR
		Preset value	FUN 14 T/C250	AR B/ T/C50 preset value →AR
BIN division	Constant		FUN 53 12	AR / BIN constant 12 →AR
	Internal output		FUN 64 WM510	AR / WM510 →AR

Note: In case of FUN54 (DBYT1), a decimal entry is automatically converted into a hexadecimal value before division because of the restriction peculiar to the programmer. Besides, entry is possible only up to 3 digits. For instance, entry $(12)_{10}$ is converted into $(C)_{16}$.

Concept of arithmetic instruction	Load	out	4-rule calculations				Logic	Compare	Carry	Convert	Shift	Mask	Exchange	Distribute/ extract
			Add	Subtract	Multiply	Divide								
76	77	79	81	82	83	84	85	86	87	88	91	92	93	94

Instruction	Abbreviation	Name	Function	Component	No. of words	Change in register			
						AR	ER	CR	Acc
FUN 5	ANDI	AND	AR AND constant → AR	constant (0000H ~ 9999H)	2	↕	•	•	•
FUN 15	AND		AR AND I/O → AR	WX, WY, WM, T/C 100 ~ 295	2	↕	•	•	•
FUN 55	BANDI		AR_L AND constant → AR_L	constant (00 ~ FF)	2	↕	•	•	•
FUN 6	ORI	OR	AR OR constant → AR	constant (0000H ~ 9999H)	2	↕	•	•	•
FUN 16	OR		AR OR I/O → AR	WX, WY, WM, T/C 100 ~ 295	2	↕	•	•	•
FUN 56	BORI		AR_L OR constant → AR_L	constant (00 ~ FF)	2	↕	•	•	•
FUN 66	EXOR	Exclusive-OR	AR EOR I/O → AR	WX, WY, WM, T/C 100 ~ 295	2	↕	•	•	•
FUN 85	WNOT	Logical NOT	$\overline{AR}$ → AR	None	1	↕	•	•	•



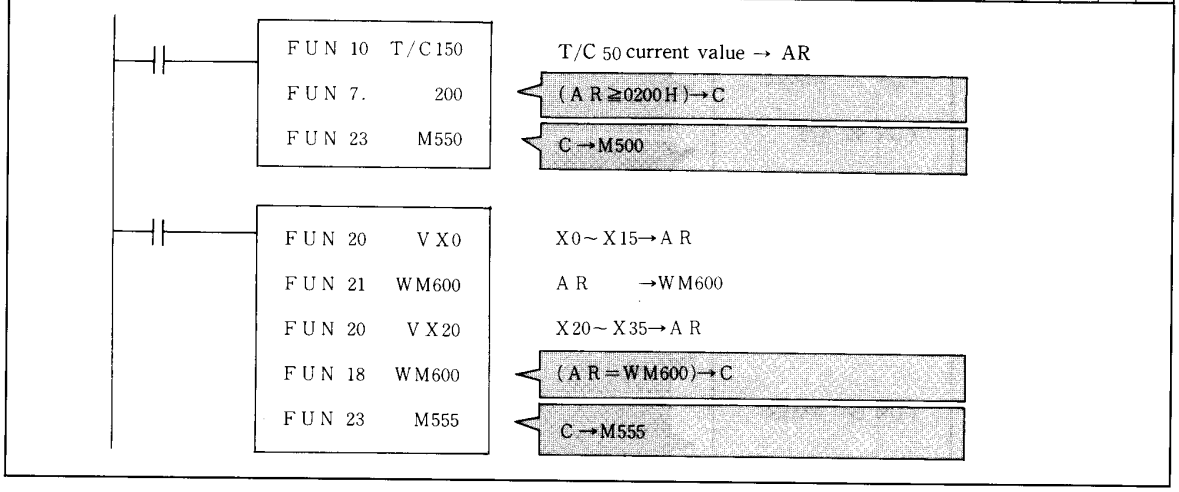
4.3

[**Explanation**]

1. FUN5 (ANDI) and FUN6 (ORI) instructions perform logical AND and OR operations between AR register data and constants 0000H to 9999H.
   FUN55 (BANDI) and FUN56 (BORI) instructions perform logical AND and OR operations between the lower 8 bits (AR_L) of AR regdster and constants 00 to FF. Because of the restriction peculiar to the progrmmer, a decimal entry is automatically converted into a hexadecimal value before logical AND/OR operations. For instance, entry $(170)_{10}$ is converted to $(AA)_{16}$.
2. FUN15 (AND) and FUN16 (OR) instructions perform logical AND/OR operations between AR register data and external input, external output, internal output or current value/preset value of timer/counter.
3. FUN66 (EXOR) instruction performs logical exclusive-OR operation between AR register data and external input, external output, internal output or current value/preset value of timer/counter.
4. FUN85 (WONT) instruction performs logical NOT operation of AR register data.

Concept of arithmetic instruction	Load	out	4-rule calculations				Logic	Compare	Carry	Convert	Shift	Mask	Exchange	Distribute/ extract
			Add	Subtract	Multiply	Divide								
76	77	79	81	82	83	84	85	86	87	88	91	92	93	94

Instruction	Abbreviation	Name	Function	Component	No. of words	Change in register			
						AR	ER	CR	Acc
FUN 7	CPEHI	Compare (≧)	AR≧constant……1→C AR<constant……0→C	constant (0000H～9999H)	2	•	•	↕	•
FUN 17	CPEH		AR≧I/O……1→C AR<I/O……0→C	WX, WY, WM. T/C 100～295	2	•	•	↕	•
FUN 57	BCPHI		AR_L≧constant……1→C AR_L<constant……0→C	constant (00～FF)	2	•	•	↕	•
FUN 8	CPEI	Compare (=)	AR=constant……1→C AR≠constant……0→C	constant (0000H～9999H)	2	•	•	↕	•
FUN 18	CPE		AR=I/O……1→C AR≠I/O……0→C	WX, WY, WM, T/C 100～295	2	•	•	↕	•
FUN 58	BCPEI		AR_L=constant……1→C AR_L≠constant……0→C	constant (00～FF)	2	•	•	↕	•
FUN 9	CPLI	Compare (<)	AR<constant……1→C AR≧constant……0→C	constant (0000H～9999H)	2	•	•	↕	•
FUN 19	CPL		AR<I/O……1→C AR≧I/O……0→C	WX, WY, WM, T/C 100～295	2	•	•	↕	•
FUN 59	BCPLI		AR_L<constant……1→C AR_L≧constant……0→C	constant (00～FF)	2	•	•	↕	•

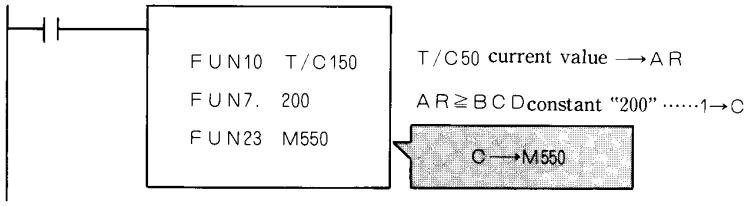


[Explanation]

1. Compare instructions are classified into 3 types: ≧, = and<. Each type consists of 3 kinds of instructions. So nine kinds of compare instructions in total are selectable to suit the component. AR register and component data are compared as binary numbers without sign. If the result of comparison is true, carry C is set to ON. If it is false, carry C is set to OFF.
2. FUN7 (CPEHI) and FUN9 (CPLI) are instructions to compare AR register data with constants 0000H to 9999H. FUN57 (BCPHI), FUN58 (BCPEI) and FUN59 (BCPLI) are instructions to compare the lower 8 bits (AR_L) of AR registem with constants 00to FF. Because of the restriction peculiar to the programmer, a decimal entry is automatically converted into a hexadecimal value before comparison. For instance, entry $(255)_{10}$ is converted into $(FF)_{16}$.
3. FUN17 (CPEH), FUN18(CPE) and FUN19(CPL) are instructions to compareAR register data with external input, external output, internal output, timer/counter current value and preset value. Component data need not be BCD data (0000H through 9999H).

Concept of arithmetic instruction	Load	out	4-rule calculations				Logic	Compare	Carry	Convert	Shift	Mask	Exchange	Distribute/ extract
			Add	Subtract	Multiply	Divide								
76	77	79	81	82	83	84	85	86	87	88	91	92	93	94

Instruction	Abbreviation	Name	Function	Component	No. of words	Change in register			
						AR	ER	CR	Acc
FUN23	OUC	Out carry	C→I/O	Y, M	1	·	·	·	·
FUN83	CLC	Clear carry	C←0	None	1	·	·	0	·
FUN84	SEC	Set carry	C←1	None	1	·	·	1	·



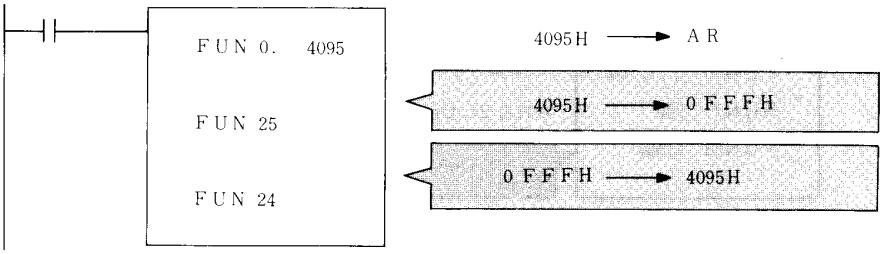
[Explanation]

1. The FUN23 (OCU) instruction outputs carry C data to internal output (M) or external output (Y).
2. The FUN84 (SEC) instruction sets "1" to carry C. The FUN83 (CLC) resets carry C to "0"

Concept of arithmetic instruction	Load	out	4-rule calculations				Logic	Compare	Carry	Convert	Shift	Mask	Exchange	Distribute/ extract
			Add	Subtract	Multiply	Divide								

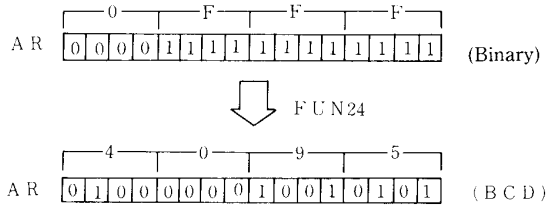
76	77	79	81	82	83	84	85	86	87	88	91	92	93	94

Instruction	Abbreviation	Name	Function	Component	No. of words	Change in register			
						AR	ER	CR	Acc
FUN24	BCD	BCD convert	AR $\xrightarrow{\text{BCD convert}}$ AR	None	1	↕	·	↕	·
FUN25	BNR	BIN convert	AR $\xrightarrow{\text{BIN convert}}$ AR	None	1	↕	·	↕	·



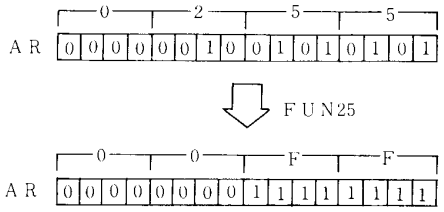
[Explanation]

1. The FUN24 (BCD) instruction converts the binary data in the AR register into BCD data. If the result of conversion is 4 digits or less, carry C turns OFF.



If the result of conversion overflows 4 digits, the AR register data is not converted (the contents of register remain unchanged) and carry C turns ON.

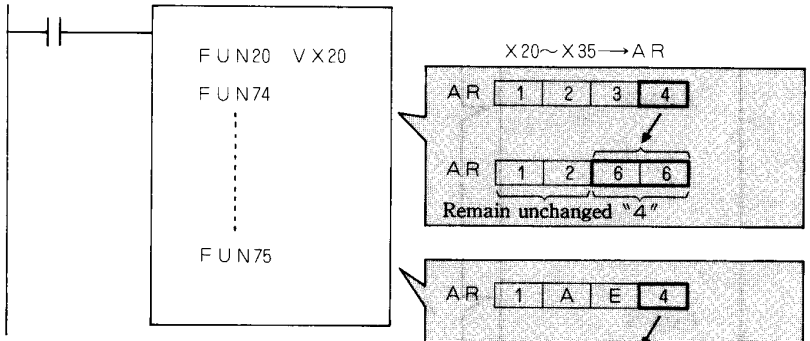
2. The FUN25 (NR) instruction converts the BCD data in the AR register into binary data. When the AR register contains BCD data before conversion, carry C turns OFF.



Before conversion, each digit of the AR register must be value in the range of 0 to 9. If the AR register data is within A to F, it will not be converted (the contents of register remain unchanged)and the value of carry C will become unreliable.

Concept of arithmetic instruction	Load	out	4-rule calculations				Logic	Compare	Carry	Convert	Shift	Mask	Exchange	Distribute/ extract
			Add	Subtract	Multiply	Divide								
76	77	79	81	82	83	84	85	86	87	88	91	92	93	94

Instruction	Abbreviation	Name	Function	Component	No. of words	Change in register			
						AR	ER	CR	ACC
FUN74	SEG	7-segment decode	AR_{LL} 7-segment convert AR	None	1	↕	·	·	·
FUN75	ASC	ASCII convert	AR_{LL} ASCII convert AR	None	1	↕	·	·	·

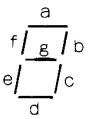


4.3

[**Explanation**]

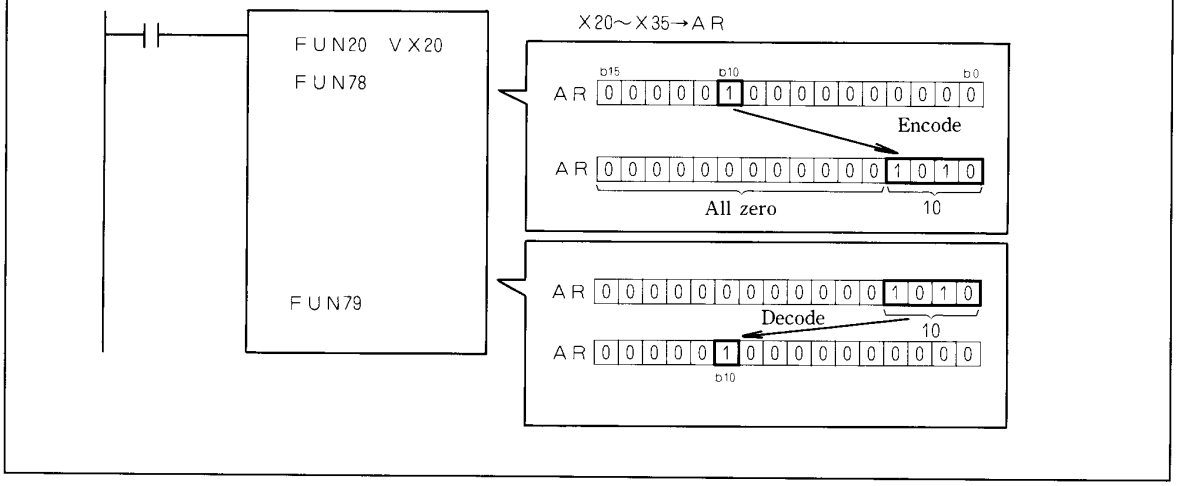
1. The FUN74 (SEG) instruction covers the lower 4 bit (AR_{LL}) data of AR register into a 7-segment display code and stores in the lower 8 bits (AR_L) of that register.
2. The FUN75 (ASC) instruction converts the lower 4-bit (AR_{LL}) data of AR register into an ASCII code and stores it in the lower 8 bits (AR_L) of that register.
3. The upper 8 bits (AR_H) of AR register remain unchanged before and after excution of the FUN74 (SEG) or FUN75 (ASC) instruction.
4. Shown below is a FUN74 (SEG) and FUN75 (ASC) conversion table.

Input data 4 bits	FUN74(SEG) Output data								Display	FUN75(ASC) Output data
	g	f	e	d	c	b	a			
0	0	0	1	1	1	1	1	1	0	30
1	0	0	0	0	1	1	0		1	31
2	0	1	0	1	1	0	1	1	2	32
3	0	1	0	0	1	1	1	1	3	33
4	0	1	1	0	0	1	1	0	4	34
5	0	1	1	0	1	1	0	1	5	35
6	0	1	1	1	1	1	0	1	6	36
7	0	0	1	0	0	1	1	1	7	37
8	0	1	1	1	1	1	1	1	8	38
9	0	1	1	0	1	1	1	1	9	39
A	0	1	1	1	0	1	1	1	A	41
B	0	1	1	1	1	1	0	0	b	42
C	0	0	1	1	1	0	0	1	C	43
D	0	1	0	1	1	1	1	0	d	44
E	0	1	1	1	1	0	0	1	E	45
F	0	1	1	1	0	0	0	1	F	46



Display segments

Instruction	Abbreviation	Name	Function	Component	No. of words	Change in register			
						AR	ER	CR	Acc
FUN78	ENCOD	Encode	16→4 encode	None	1	↕	·	↕	·
FUN79	DECOD	Decode	4→16 decode	None	1	↕	·	↕	·

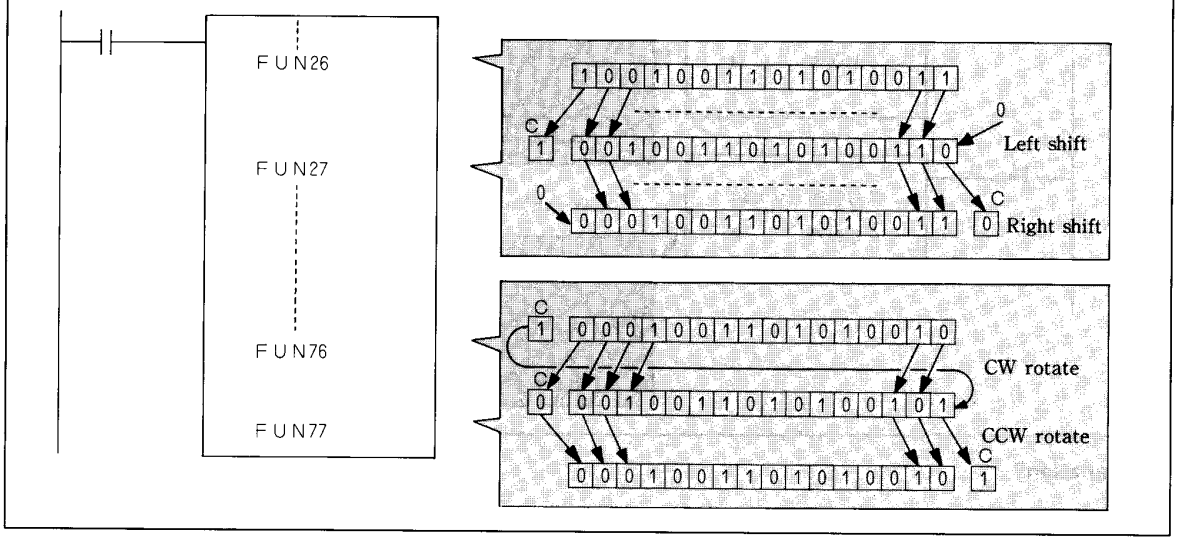


[**Explanation**]

1. The FUN78 (ENCODE) instruction sets into the AR the uppermost bit position (1 to 15), where "1" is set, among the bits of the register. In case all bits are 0, AR and C become 0 and 1, respectivenly. When two or more bits are 1, the uppermost bit is selected.
2. The FUN79 (DECODE) instruction sets 1 at the bit position corresponding to the value of AR register (0to 15) and clears all other bits to 0. In case the AR register value is 16 or more, AR and C become 0 and 1, respectively.

Concept of arithmetic instruction	Load	out	4-rule calculations				Logic	Compare	Carry	Convert	Shift	Mask	Exchange	Distribute/ extract
			Add	Subtract	Multiply	Divide								
76	77	79	81	82	83	84	85	86	87	88	91	92	93	94

Instruction	Abbreviation	Name	Function	Component	No. of words	Change in register			
						AR	ER	CR	Acc
FUN26	LSFR	Left shift	C← AR ←0	None	1	↕	·	↕	·
FUN27	RSFR	Right shift	0→ AR →C	None	1	↕	·	↕	·
FUN76	ROL	CW rotate	C← AR ←	None	1	↕	·	↕	·
FUN77	ROR	CCW rotate	→ AR →C	None	1	↕	·	↕	·

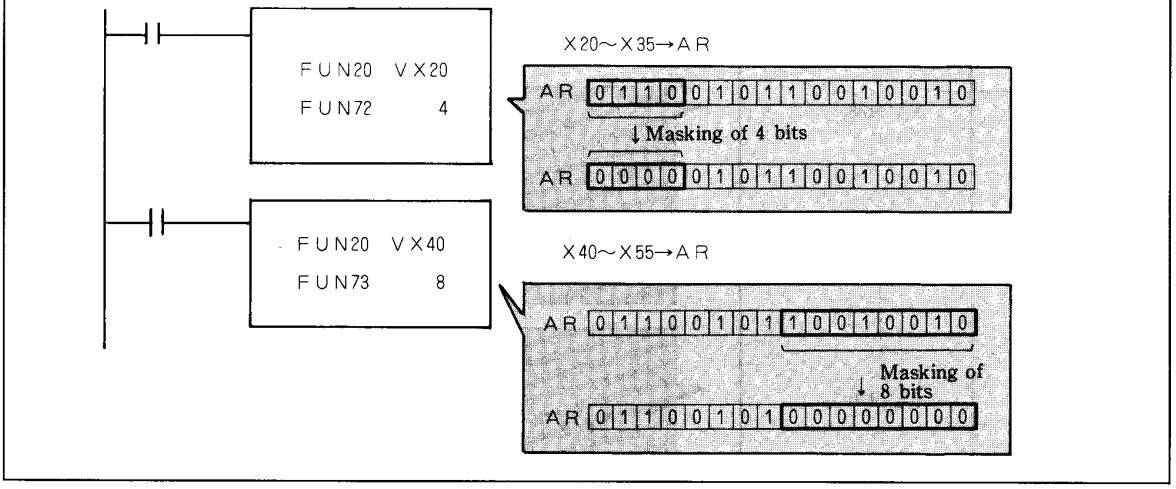


[**Explanation**]

1. The Fun26 (LSFR) instruction shifts AR register data 1 bit to the left. Upon shift, the least significant bit is padded with zero and the overflow bit is set to carry C.
2. The FUN27 (RSFR) instruction shifts AR register data 1 bit to the right. Upon shift, the most significant bit is paddedwith zero and the overflow bit is set to carry C.
3. The FUN76 (ROL) instruction shifts AR register data 1 bit to the left. Upon shift, the overflow bit is set to carry C and the least significant bit is padded with the previous data in the carry C.
4. TheFUN 77 (ROR) instruction shifts AR register data 1 bit to the right. Upon shift, the overflow bit is set to carry C and the most significant bit is padded with the previous data in the carry C.

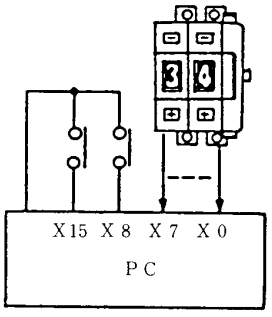
Concept of arithmetic instruction	Load	out	4-rule calculations				Logic	Compare	Carry	Convert	Shift	Mask	Exchange	Distribute/ extract
			Add	Subtract	Multiply	Divide								
76	77	79	81	82	83	84	85	86	87	88	91	92	93	94

Instruction	Abbreviation	Name	Function	Component	No. of words	Change in register			
						AR	ER	CR	Acc
FUN72	MASKL	Mask	Masks by specified bits from uppermost bit.	0~255	2	↕	·	·	·
FUN73	MASKR		Masks by specified bits from lowermost bit.	0~255	2	↕	·	·	·

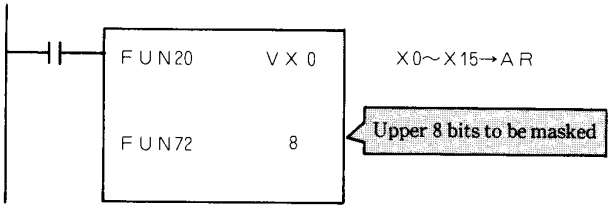


[**Explanation**]

1. FUN72(MASKL) and FUN73 (MASKR) instructions mask the AR register data by the specified number of bits.
   The FUN72 (MASKL) instruction masks the data by the specified number of bits starting from the most significant bit (b_{15}).
   The FUN73 (MASKR) instruction masks the data by thespecified number of bits starting from the least significant bit (b_0).
   Even when 17 or more bits are specified, only 16 bits are validated.
2. Application example of mask instruction

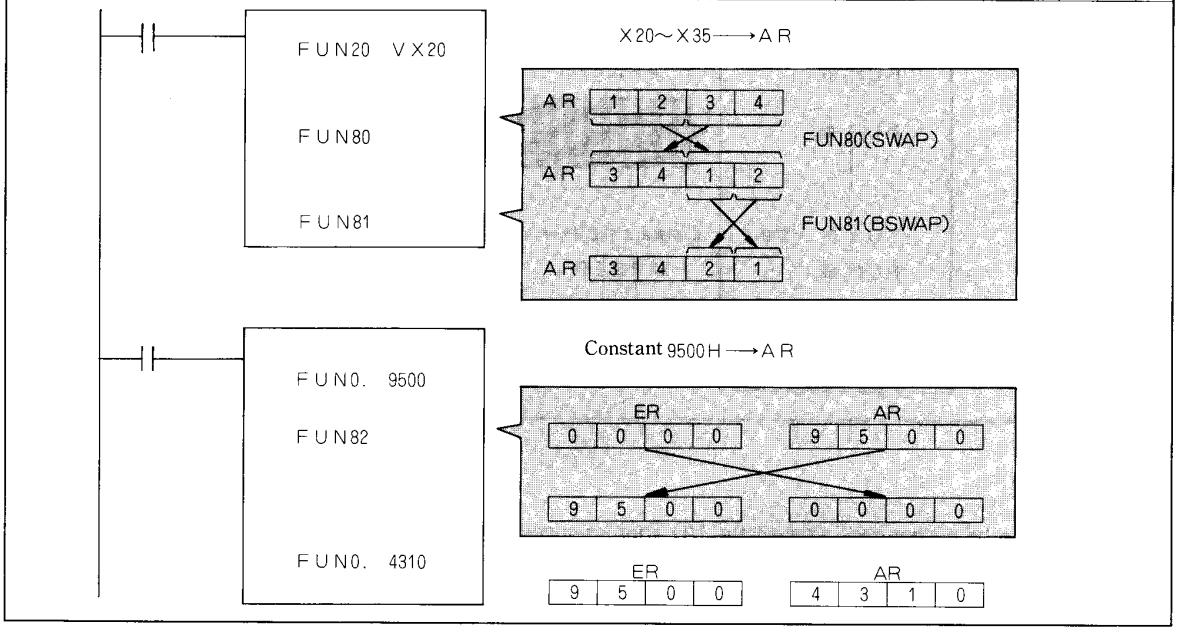


When loading the 2-digit thumbwheel switch data into X0 through X7 to be followed by loading of ordinary input signals in X8 through X15, the switch data is also loaded into X8 through X15 automatically. This is because the FUN20 (LOADB) instruction operates on a data word of 16 bits long. To mask X8 through X15, use the FUN72 (MASKL) instruction.



Concept of arithmetic instruction	Load	out	4-rule calculations				Logic	Compare	Carry	Convert	Shift	Mask	Exchange	Distribute/ extract
			Add	Subtract	Multiply	Divide								
76	77	79	81	82	83	84	85	86	87	88	91	92	93	94

Instruction	Abbreviation	Name	Function	Component	No. of words	Change in register			
						AR	ER	CR	Acc
FUN80	SWAP	Exchange	$AR_H \rightleftarrows AR_L$	None	1	↕	·	·	·
FUN81	BSWAP		$AR_{LH} \rightleftarrows AR_{LL}$	None	1	↕	·	·	·
FUN82	XCG		AR ⇄ ER	None	1	↕	↕	·	·



4.3

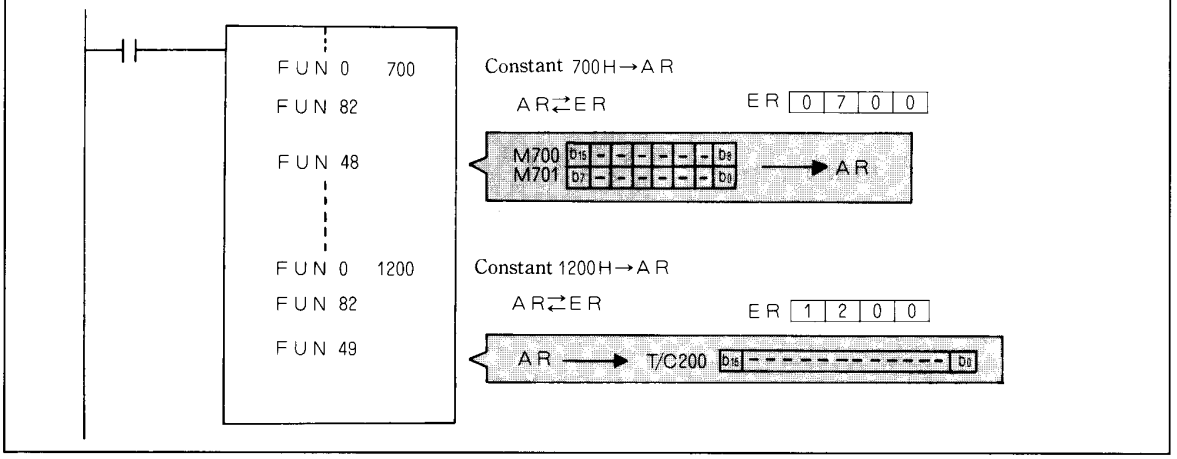
[**Explanation**]

1. The FUN80 (SWAP) instruction exchanges the upper byte (b_8 through b_{15}) and the lower byte (b_0 through b_7) of the AR register.
2. The FUN81 (BSWAP) instruction exchanges the upper nibble (b_4 through b_7) and lower nibble (b_0 through b_3) of lower byte in the AR register.
3. The FUN82 (XCG) instruction exchanges the AR register and ER register. TheFUN82 instruction is used for setting data in the ER register.

Concept of arithmetic instruction	Load	out	4-rule calculations				Logic	Compare	Carry	Convert	Shift	Mask	Exchange	Distribute/ extract
			Add	Subtract	Multiply	Divide								

76	77	79	81	82	83	84	85	86	87	88	91	92	93	94

Instruction	Abbreviation	Name	Function	Component	No. of words	Change in register			
						AR	ER	CR	Acc
FUN48	EX	Extract	Fetches data into AR from I/0 address-specified by ER.	None	1	↕	·	↕	·
FUN49	DB	Distribute	Outputs data from AR to I/0 address-spesifid by ER.	None	1	·	·	↕	·



[**Explanation**]

1. Data is to be exchanged between the I/O address-specified by the ER regitser and the arithmetic register AR. The ER register contains BCD data. The most significant digit 0 andl stand for usual I/O and timer/ counter, respectively. FUN48 (EX) fetches data into AR, and FUN49 (DB) outputs data to I/O.
2. CR will become 1 if either instruction is executed with an undefined I/O specified by theER register (only when Acc = 1).

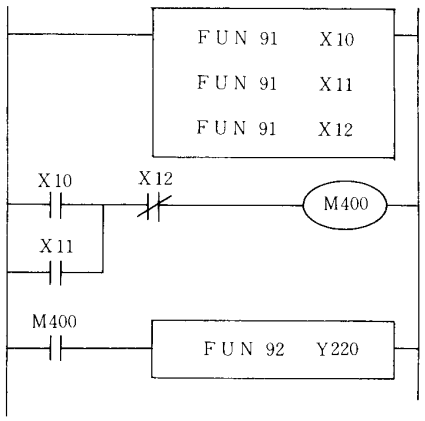
Application Instructions (II)

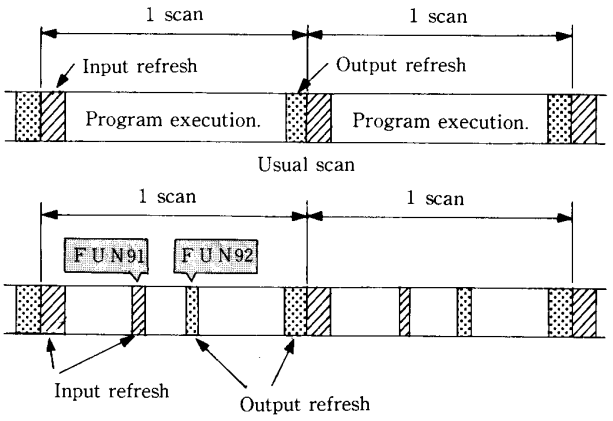
Classification	Instruction	Symbol	Name	Function	Component	No. of words	Change in register				Reference page
							AR	ER	CR	Acc	
Refresh	FUN91	REFX	I/0 refresh	Refreshes specified input.	X	1	●	●	●	●	97
	FUN92	REFY		Refreshes specified output.	Y	1	●	●	●	●	97
Interrupt	FUN93	INT	Declares interrupt.	Argument 2 = 0: Declares interrupt at fixed intervals of 10 ms.	Argument 2	2	–	–	–	–	98
	FUN94	RTI	Recovery from interrupt	Recovery from interrupt	None	1	Value before interrupt				98
Subroutine	FUN42	CALL	Subroutine	Calls subroutine.	Arguments 0 to 63	2	●	●	●	●	99
	FUN43	SB		Defines subroutine.	Arguments 0 to 63	2	–	–	–	–	99
	FUN44	RTS		Recovery from subroutine.	None	1	Value before subroutine call				99

●: Register remains unchanged
↕: Register changed
–: Register cleared

I/O refresh	Interrupt	Subroutine
97	98	99

Instruction	Symbol	Name	Function	Component	No. of words	Change in register			
						AR	ER	CR	Acc
FUN91	REFX	I/0 refresh	Inputs specified I/0.	X	1	·	·	·	·
FUN92	REFY		Outputs specified I/0.	Y	1	·	·	·	·





I/0 refresh according to FUN91/FUN92 instructions

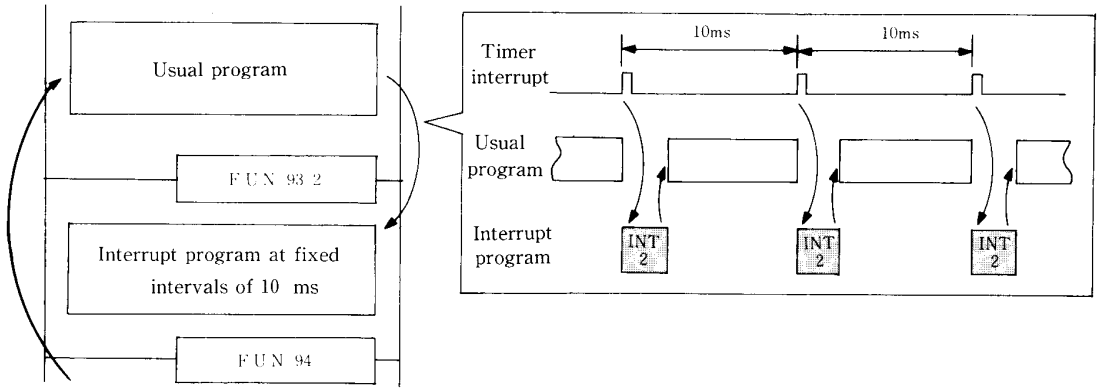
(Specified input and output can be refreshed in the middle of scan.)

4.4

[Explanation]

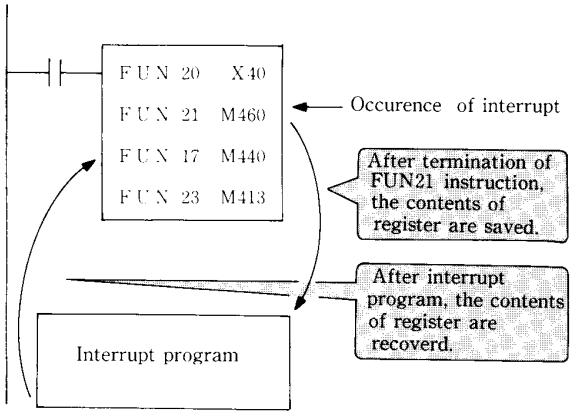
1. FUN91 (REFX) is imputrefresh instruction. It rewrites data memory of the specified input number in the course of scan (upon its execution). This instruction does not have a start condition.
2. FUN92 (REFY) is output refresh instruction. It rewrites the specified output number and its data memory the same as in the current Acc register during scan (upon its execution).
3. Input signals shorter than scan time can be acquired by uniform allocation of the refresh instruction at several locations in the entire program.

Instruction	Symbol	Name	Function	Component	No. of words	Change in register			
						AR	ER	CR	Acc
FUN93	INT	Declares interrupt.	Argument 2 | Interrupt at fixed intervals of 10 ms	Argument 2	2	–	–	–	–
FUN94	RTI	Recovery from interrupt	Recovery from interrupt	None	1	Value before interrupt			



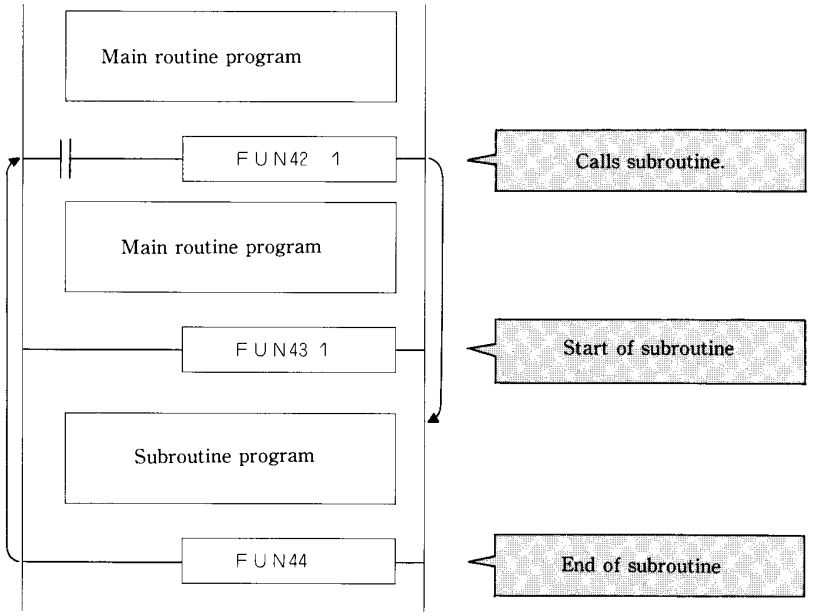
[**Explanation**]

1. An interrupt program is to be located next to a usual program. These programs are to be separated by the FUN93 (INT) instruction. FUN99 (END) is not used. The end of interrupt program must always be the FUN94 (RTI) instruction. Neither FUN93 nor FUN94 requires start condition.
2. Interrupt program is excuted every 10 ms when it is written between the FUN93 2 (INT2) and FUN94 (RTI) instructions after a usual program.
3. When applying an interrupt, the instruction under execution is terminated and the relevant interrupt program is excuted once. On this occasion, the contents of register are automatically saved.
   After termination of the interrupt program, the usual program before interrupt program returns and the contents of register are recovered.



4. Interrupt instruction and jump instruction without adressing cannot coexist.

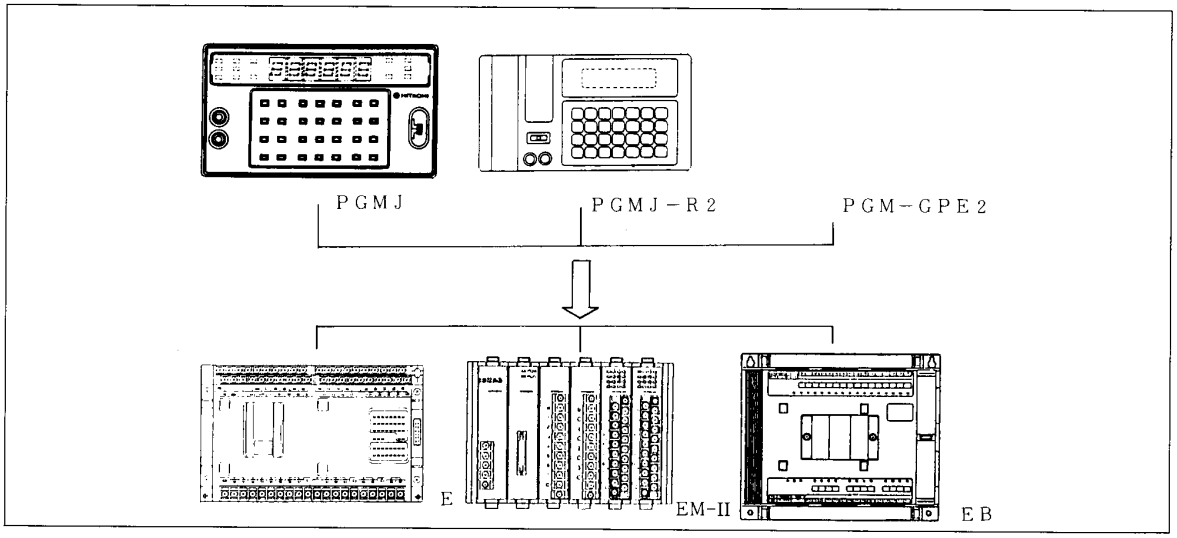
Instruction	Symbol	Name	Function	Component	No. of words	Change in register AR	ER	CR	ACC
FUN42	CALL	Subroutine	Calls subroutine.	Arguments O to 63	2	·	·	·	·
FUN43	SB		Defines subroutine	Arguments O to 63	2	–	–	–	–
FUN44	RTS		Recovery from subroutine	None	1	Value before subroutine call			



4.4

[Explanation]

1. Subroutine program is to be located next to the main routine program. At the head of subroutine program, the FUN43 (SB) instruction is required to be set. Each subroutine program must be terminated by the FUN44 (RTS) instruction. Subroutine can be called by the FUN42 (CALL) instruction.
2. The FUN99 (END) instruction is unnecessary between the main routine program and subroutine program. Neither FUN43 nor FUN44 requires the start condition.
3. In a subroutine, jump and master control instructions are unusable.



[**Explanation**]

1. Kinds of peripheral equipment

Peripheral equipment, or programmer is selectable among three kinds: standard programmer PGMJ, universal programmer PGMJ-R2 and portable graphic programmer PGM-GPE2. Each programmer can be used with E, EM, EM-II and EB series.

Besides, IBM*1 PC XT*2 personal comupter is usable for programming by running the personal computer programming software E-LADDER. The functions of each peripheral equipment are listed in the table below.

Table 5-1 Function of Peripheral Equipment

Model	Programming				CMT	ROM	RS-232C		Parallel
	Online	Offline	Instruction set	Ladder	I／F	writer	Printer	Personal computer	printer
PGMJ	○	–	○	–	○	–	–	–	–
PGMJ－R2	○	–	○	–	○	○	○	○	–
PGM－GPE2	○	○	○	○	○	○	○	○	–
Personal computer software 〈E-LADDER〉	○	○	○	○	–	–	–	–	○

o: Possible
–: Function unavailable

*1. IBM is a trademark of International Business Machines Corporation.

*2. PC XT is a product of International Business Machines Corporation.

2. Compatibilty

The module PGMJ, PGMJ-R and PGM-GPE in your possession are also usable for CPM-E2 or CPM-E3. However, each programming has restrictions as listed in the table below.

Table 5-2 Compatibility of Peripheral Equipment

Item	PGMJ	PGMJ-R		PGMJ -R2	PGM -GPE	PGM -GPE2	E-LADDER	
		Up to V : 4	V : 5				V:4	V:5
Programming in up to 2K words	○	○	○	○	○	○	○	○
Programming in up to 4K words	○	○	○	○	×	○	×	○
Programming by instructions compatible with EM	○	○	○	○	○	○	○	○
Printout according to instructions compatible with EM	–	○	○	○	○	○	○	○
Programming according to new instructions for EM	○	×	○	○	×	○	×	○
Printout accoring to new instructions for EM	–	×	○	○	×	○	×	○
Decimal and hexadecimal monitoring *1	○	×	○	○	×	○	×	○
CMT function in up to 2K words	○	○	○	○	○	○	–	–
CMT function in up to 4K words	○	×	×	○	×	○	–	–
ROM writer function in up to 2K words	–	○	○	○	○	○	–	–
ROM writer function in up to 4K words	–	×	×	○	×	○	–	–
Time point of enhancement	–	–	Jun, 1989	May, 1990	–	May, 1990	–	Near future

*1. Error code in syntax check cannot be observed unless decimal monitoring is possible.

5

Table 5-3 Specifications of Peripheral Equipment

Item \ Model			PGMJ	PGMJ-R2	PGM-GPE2
Programming function	Display unit		Digital display (LED)	Liquid crystal	Liquid crystal
	Input system		Instruction set		Instructions, ladder deagram
	Editing function		Write, read, change, insert, delete, search		
	Monitoring function		One-point monitoring		Multi-point monitoring
	Test function		Forced output, forced setting/resetting		
CMT interface function			Audio cassette tape recording, reproduction and verification		
ROM writer function			—	Memory pack copying, reproduction and verification	
RS-232C function	Synchronization		—	Asynchronous	
	Baud rate		300, 600, 1,200, 2,400, 4,800, 9,600, 19,200, 38,400, bps (Selectable by DIP switch. Rate set to 4,800 bps before shipment)		
	Word length		Start bit: 1 bit Data bit: 8 bits } Set before shipment Stop bit: 1 bit (Other 6 kinds selectable by DIP switch)		
	Printer function		Code list, ladder diagram and cross reference printed out		
	Personal computer function		Data exchange with personal computer		
	Connectable peripheral equipment	Printer	Printer: EPSON SP-80T (old models RP-80 and EP-80 also connectable) Interface circuit board: No. 8148 (old model No. 8145 also connectable)		
		Personal computer	IBM PC XT		
General specifications	Operating temperature		0~55℃	5~40℃	0~40℃
	Storage temperature		−10~65℃	−10~60℃	−10~50℃
	Operating humidity		30 to 90% RH (non-condensing)		
	Power supply		Supplied from basic unit		Supplied from basic unit or via AC adapter

Key part names and external dimensions of each programmer are shown below.

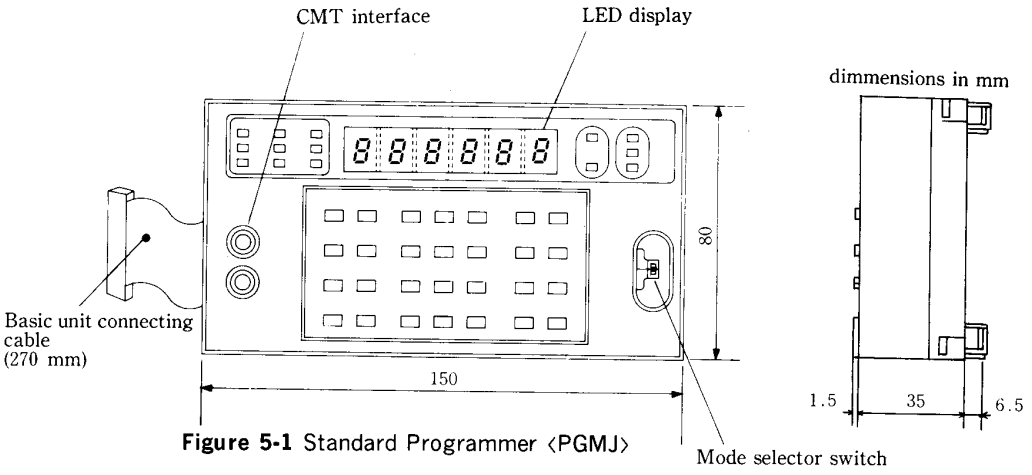


Figure 5-1 Standard Programmer ⟨PGMJ⟩

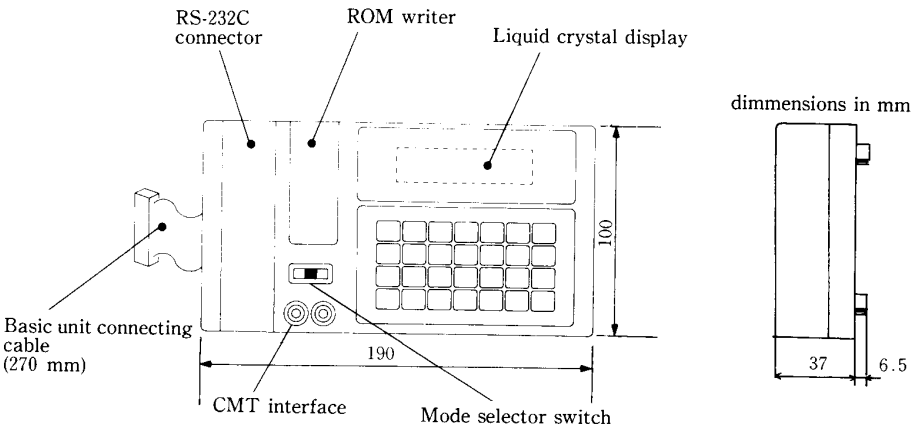


Figure 5-2 Universal Programmer ⟨PGMJ-R2⟩

Note: The power switch functions only in offline mode.

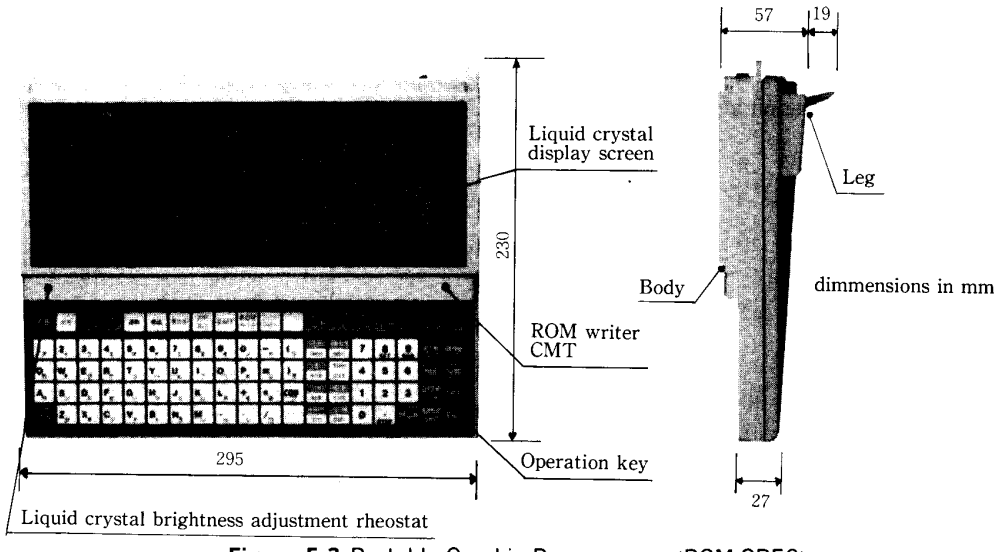


Figure 5-3 Portable Graphic Programmer ⟨PGM-GPE2⟩

(1) All clear
Clear the memory before writing an entirely new program.

(2) Write-in
Write a program for each step.

(3) Read-out
Read out the program for each step to check if there is any programming error.

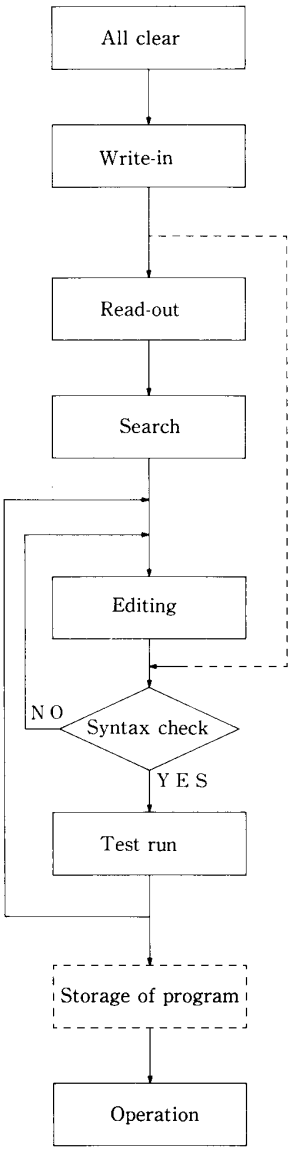
(4) Search
Search for desired input/output number, step number or instruction word.

(5) Editing
Change, insert or delete the program.

(6) Syntax Check
Check if there is any syntax error in the program written in. If there is any, correct the program.

(7) Test Run
Perform test run after making sure that wiring has been made properly through forced output.

(8) Operation
Proceed to operation after completing test run.



Procedures before operation

The table below shows a list of programmer key-in procedures.

a The contents of display correspond to the standard programmer.

b Key-in procedures are the same between the standard and universal programmers.

c For operation of the portable graphic programmer PGM-GPE2, refer to its manual.

Table 5-4 Programmer Key-in Procedures

No	Function			Key-in procedure	Contents of display: Step No.	Data	Preset value	Current value	Continuity	DATA ☼	STEP ☼	Mode: PROG	TEST	RUN	Operation	Stop
1	Program all clear			CLR ENT DEL			○			○		○	×	×	×	○
2	Write-in		Write-in of new program	CLR ENT DEL → Generation of program for each step → ENT; Continuous write-in		○		○		○		○	×	×	×	○
			Write-in of additional program	CLR STEP ↓ → Generation of program for each step → ENT; Continuous write-in		○		○		○		○	×	×	×	○
3	Read-out		Starting from step 000	CLR STEP +		○		○	○	○		○	○	○	○	○
			Starting from specified step	CLR Step No. STEP ↓		○		○	○	○		○	○	○	○	○
			From searched I/O or instruction	CLR I/O No. or instruction SRC		○		○	○	○		○	○	○	○	○
			First step of unprogrammed area	CLR STEP ↓		○				○		○	○	○	○	○
				STEP + or STEP − : 1 step forward or backward												
	Switchover between data display and step display			Read-out STEP ↓ (Data or step is selected by this STEP ↓ key.)	○	○		○		○	○	○	○	○	○	○
4	Search	I/O number	External input/output and internal output Timer and counter	CLR I/O No. SRC / Timer/counter No. SRC (1 step forward or backward)		○		○	○	○		○	○	○	○	○
		Coil	External output and internal output Timer and counter	CLR OUT I/O No. SRC / Timer/counter No. SRC → STEP + or STEP − → SRC		○		○	○	○		○	○	○	○	○
		Instruction	Basic instruction Application instruction	CLR Instruction word SRC (Search for step in which searched data is written next)		○		○	○	○		○	○	○	○	○
5	Editing		Insertion	Read-out of step to be inserted (INCL+) Generation of program to be inserted INS		○		○		○		○	×	×	×	○
			Deletion	Read-out of step to be deleted DEL		○		○		○		○	×	×	×	○
			Change	Read-out of step to be changed (INCL+) Generation of new program ENT		○		○		○		○	×	×	×	○
6	Monitor	Contacts	External input/output and internal output Timer and counter	CLR I/O No. MON / CLR Timer/counter No. MON / Read-out of program NON → STEP + or STEP −		○			○	○		○	○	○	○	○
		Coil	External output and internal output Timer and counter	CLR OUT I/O No. MON / CLR OUT Timer/counter No. MON / Read-out of program NON → STEP + or STEP − → MON		○			○	○		○	○	○	○	○
7	Check		Syntax check	CLR SRC (SRC) Continuation of syntax check (possible only for double coil error)	○						○	○	○	○	○	○
8	Maintenance function	Forced output	External output	CLR SET SET ENT FUN 3 OUT External output No. SET or RES		○			○	○		×	○	×	×	○
		Forced setting/resetting Simulation input	External input/output Internal output Timer/counter	CLR OUT Internal output No. MON SET or RES		○			○	○		×	○	○	○	×
				CLR OUT Timer/counter No. MON SET or RES		○			○	○		×	○	○	○	×

5

Function	Programmer mode			Operational status	
	PROG	TEST	RUN	Operation	Stop
All Clear	○	×	×	×	○

· Key-in procedure and display

Key-in procedure	Display			Remarks
	Instruction	Numerical display	Mode display	
CLR			· PROG · DATA	
ENT		E		
DEL		-		All Clear complete

[Explanation]

1. Be sure to perform "All Clear" before writing new programs. ("All Clear" operation has been performed before shipment from the factory.)
2. "All Clear" clears all the programs written in. In addition, timer/counter data is cleared, and the internal output protected from power failure and the shift register are reset.

[Display switchover between data and step]

1. In usual operation, step is not displayed and data alone is displayed. Press the STEP key for step No. display. When pressing this key under step No. display, data display returns.

Key-in procedure	Display			Remarks
	Instruction	Numerical display	Mode display	
CLR ENT DEL		-	· PROG · DATA	Data display
STEP		0	· PROG · STEP	Step display
STEP		-	· PROG · DATA	Data display

[Explanation]

1. If "All Clear" is keyed in with 925-step program written in, a maximum of 5 sec is required before completion of this operation (during this time period, programmer display remains off). "All Clear" operation is completed when "_" (underline) appears on the display. It will take 19 seconds to clear a 3997-step program.
2. The contents of display shown in the above table correspond to the standard programmer PGMJ. **Hereafter, this applies to all displays.**

Function	Programmer mode			Operational status	
	PRŌG	TEST	RUN	Operation	Stop
Second half initialization of memory pack MPM-2E	○	×	×	×	○

Key-in procedure	Display			Remarks
	Instruction	Numerical display	Mode display	
CLR			· PROG · DATA	
ENT		E		
INS	· ORG	000		Step 0
ENT	· AND	000		Sum value normalized

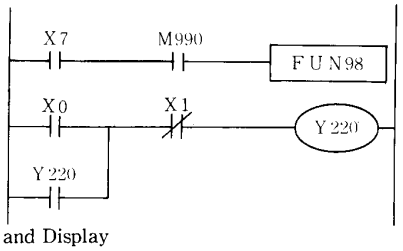
[**Explanation**]

1. This operation is required when reconnecting the memory pack MPM-2E (1950 words) used for the CPU module CPM-E to the EM-II. Contents over 1951 words are initialized.
2. Error may occur if the memory pack is used neglecting the above step.
3. However, the contents up to 1950 words remain unchanged.

5

Function	Programmer mode			Operational status	
Write-in of program	PROG	TEST	RUN	Operation	Stop
	○	×	×	×	○

· Sequence

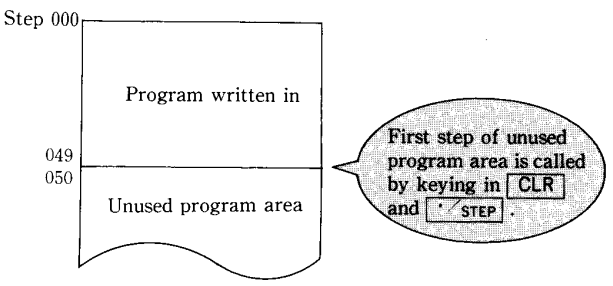


· Key-in procedure and Display

Key-in procedure	Display			Remarks
	Instruction	Numerical display	Mode display	
ORG 7 ENT	· ORG	7		X7 written in
AND 9 9 0 ENT	· AND	990		M990 〃
FUN 9 8 ENT	· FUN	98	· DATA	FUN98 〃
ORG 0 ENT	· ORG	0	· PROG	X0 written in
OR 2 2 0 ENT	· OR	220		Y220 〃
AND NOT 1 ENT	· AND · NOT	1		X1 〃
OUT 2 2 0 ENT	· OUT	220		Y220 〃

[Explanation]

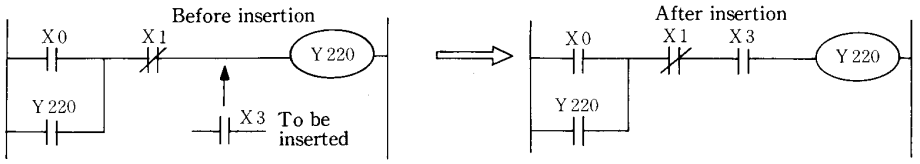
1. When pressing the [ENT] key, the contents shown on the display unit are written in the memory and program moves on the next step.
2. The contents of display exemplified above are those before pressing the [ENT] key.
3. Write-in of additional program
   When keying in [CLR] and [·/STEP] the first step of unused program area (step 50 in the example at right) is called. Program can be written in this area.



4. Program write-in from first step
   Step 0 is called by keying in [CLR] and [STEP +]. So program can be written from the first step.

Function	Programmer mode			Operational status	
	PROG	TEST	RUN	Operation	Stop
Insertion of program	○	×	×	×	○

· Sequence



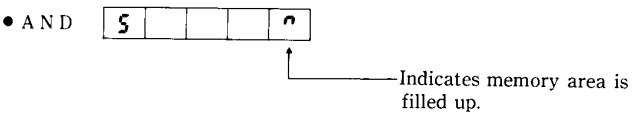
· Key-in procedure and display

Key-in procedure	Display			Remarks
	Instruction	Numerical display	Mode display	
CLR OUT 2 2 0 SRC	· OUT	220	· DATA · PROG	Searches for coil Y200.
DCLR				Clears display.
AND 3	· AND	3		Inserts contacts X3.
INS				

[Exlpanation]

1. Read out the step following the one into which a program is to be inserted. In above example, output coil —(Y220)— is searched since the contact set —| |— X3 is to be inserted before the coil. Press the [DCLR] key to erase instruction and data display, and key in the program to be inserted, then press the [INS] key. This completes insertion of one step. Upon pressing the [INS] key, the next step is displayed.
   Note that the step numbers of the programs aftem the one inserted will be automatically incremented by one.
2. After completion of inserting the new program, be sure to perform syntax check (by keying in [CLR] [SRC]) to ascertain that there is no programming error.
3. An error will occure if you attempt to insert a program when the memory area is fully loaded, because program can no longer be inserted.

   [AND] [5] [INS]

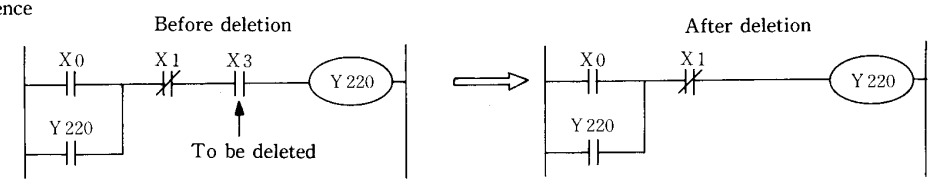
   

4. If a program insertion is made to the first step a program consisting of 900 steps, it will take about 5 sec for its completion. (Before completion, program display is turned off.)
5. Confirmation is required before pressing the [INS] key, because displayed programs are inserted sequentially whenever pressing the key.

5

Function	Programmer mode			Operational status	
Deletion of program	PROG	TEST	RUN	Operation	Stop
	○	×	×	×	○

· Sequence



· Key-in procedure and display

Key-in procedure	Display			Remarks
	Instruction	Numerical display	Mode display	
CLR 3 SRC	· AND	3	· DATA	Searches for contacts X3.
DEL	· OUT	220	· PROG	Deletes contacts X3.

[Explanation]

1. Read out the step to be deleted. When pressing the DEL key, the programs under the deleted one will be automatically decremented by one.
2. After deleting the program, be sure to perform syntax check (by keying in CLR SRC) to make sure that there is no programming error.
3. Confirmation is required before pressing the DEL key, because displayed programs are deleted sequentially whenever pressing the key.
4. If a program deletion is made from the first step of a progrmam consisting of 900 steps, it will take about 5 sec for its completion. (Before completion, program display is turned off.)
5. After insertion or delecion, the step numbers of the relevent program and thereafter will be automatically incremented.

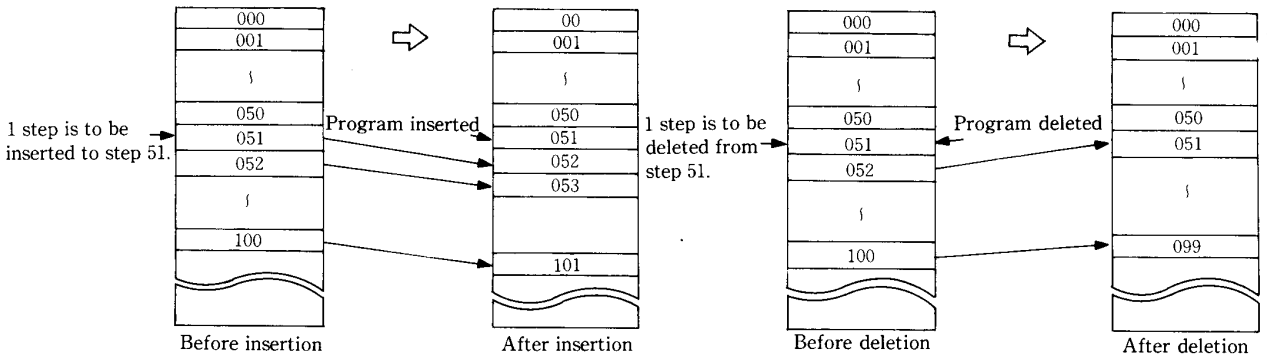
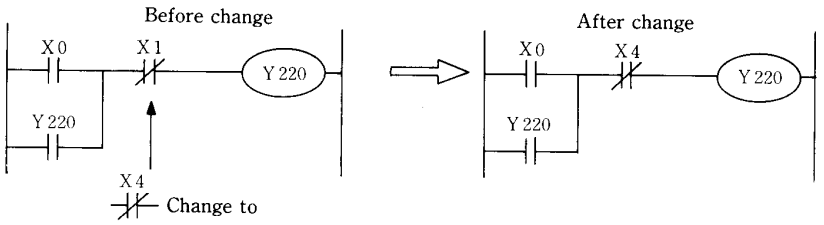


Figure 5-4 Insertion and Deletion of program

Function	Programmer mode			Operational status	
Change of program	PROG	TEST	RUN	Operation	Stop
	○	×	×	×	○

· Sequenhe



· Key-in procedure and display

Key-in procedure	Display			Remarks
	Instruction	Numerical display	Mode display	
CLR 1 SRC	· AND · NOT	001	· DATA · PROG	Searches for contacts X1
DCLR				Clears display.
AND NOT 4	· AND · NOT	4		Writes in contacts X4.
ENT	· OUT	220		

[Explanation]

1. Read out the step to be changed. Press the DCLR key to clear the instruction and data under display. Write a program beginning with an instruction. Pressing the ENT key completes the progrmam change for one step. Upon pressing this key, the next step is displayed. In case the number of words is different before and after change, the step numbers of the programs after the change one will be automatically incremented or decremented. The previous program remains unless the ENT key is pressed after program change.
2. Before change, the DCLR key must be pressed as a rule. However, a new program can be written even when instruction and data are displayed.
3. The preset value of timer/counter can be changed not only by the method above, but also by directly entering a new value as exemplified below after searching for the coil.

CLR OUT T_C 0 SRC ···Searches for timer T00 coil.

1 1 • 4 ENT ···Writes in new preset value.

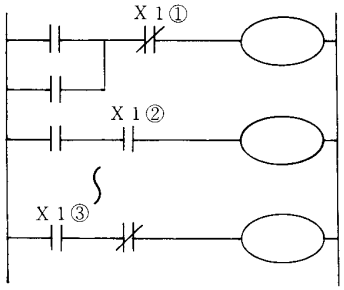
5

Function	Programmer mode			Operational status	
	PROG	TEST	RUN	Operation	Stop
Read out and search of program	○	○	○	○	○

Classification			Key-in procedure
Read-out	From start step		CLR → STEP(+) → STEP(+)
	From specified step		CLR → Step No. → STEP(/) → STEP(+) or STEP(−)
	From final step		CLR → STEP(/) → STEP(−)
Search	X, Y, M	I/O No.	CLR → Input/output No. → SRC → STEP(+) or STEP(−) or SRC
		Output No.	CLR → OUT → Output No. → SRC → STEP(+) or STEP(−) or SRC
	T/C	I/O No.	CLR → T/C → Input/output No. → SRC → STEP(+) or STEP(−) or SRC
		Output No.	CLR → OUT → T/C → Output No. → SRC → STEP(+) or STEP(−) or SRC
	Instruction word		CLR → Instruction word → SRC → STEP(+) or STEP(−) or SRC

[Explanation]

1. When pressing the STEP(/) key after specifying a step number, data written in the specified step is displayed. Then the programs before and after this step can be read out by using the keys STEP(+) and STEP(−).
2. When pressing the SRC key after specifying X, Y, M, T/C number or instruction word, data in the step where the specified number or instruction word is written is displayed.
3. Continuous search for the same number is made by the following precedure.



Uey-in procedure	Display		Remarks
	Instruction	Numerical display	
CLR → 1 → SRC	· AND · NOT	1	Searches for contact (1).
SRC	· AND	1	Searches for contact (2).
SRC	· ORG	1	Searches for contact (3).

When pressing the SRC key again after completion of one search, another step written in the same number is searched for and displayed.

4. In case the specifyed number cannot be found in the program as a result of search operation, the first step of unused program area is displayed ("_" (underline) appears).
5. Programs before and after the search one can be read out by using the keys STEP(+) and STEP(−).

Function	Programmer mode			Operational status	
	PROG	TEST	RUN	Operation	Stop
Syntax check	○	○	○	○	○

Key-in procedure	Judgement	Display			Remarks
		Instruction	Numerical display	Mode display	
CLR SRC	No error		300	・STEP (・PROG ・TEST ・RUN)	Displays first step of unprogrammed area.
	Error detected		115 E		Indicates error is found in step 115.

[Explanatien]

1. Syntax check is required after writing a program. So far as no error is found in the program, the first step number in the unprogrammed area is displayed.
2. The table below lists the error display which is presented when program contains an error, together with its factor. Each error factor can be judged by decimal monitoring of the special internal output WM980. Whenever performing syntax check, the result of the previous syntax check is cleared and the new rusult is displayed.
3. Only in case a double coil error, syntax check is performed contimuously from the first step by pressing the SRC key. Note, however, that no error will occur even if dual coil is specified for the output coil following FUN02 and FUN03.

Table 5-5 User Program Syntax Error Code List

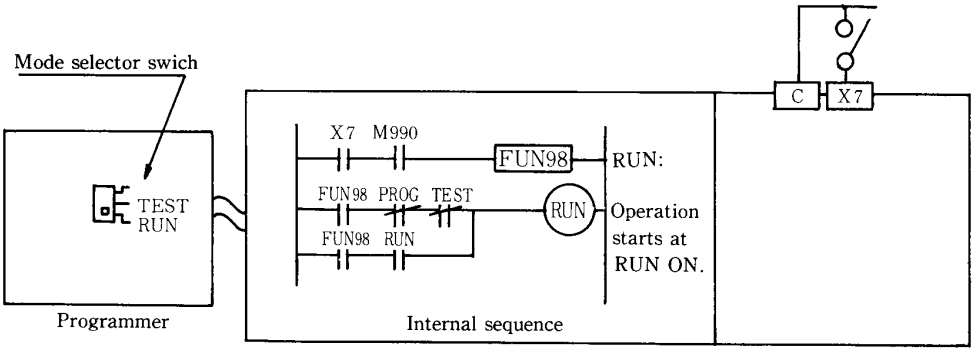
Syntax error code (decimal)	Error display on PGMJ	Error display on PGMJ-R2	Error content
0	Blank	Blank	No error
1	E	E	Combination of instruction words does not meet syntax rule.
2	E	E	The structure of main routine or interrupt processing routine is abnormal.
3	E	E	The argument of INT instruction having the relevant number is not defined.
4	E	E	The structure of FUN06 and FUN07 is abnormal.
5	E	E	The structure of FUN08 and FUN09 is abnormal.
6	⌊	uE	STR level is under the one specified for instruction word.
7	⌈	oE	STR level is over the one specified for instruction word.
8	⌊	oE	Master control level is under the one specified for instruction word.
9	⌈	oE	Master control level is over the one specified for instruction word.
10	E	E	IF or IFR is duplicated. Prohibited instruction (OUT T/C) is written after IF or IFR.
11	E	E	The I/O number, constant or the like of instruction word is not within the specified range.
12	E	E	Prohibited dual coil is specified.
13	E	dE	Dual coil is specified though operation is bone (alarm).
14	E	E	There are multiple SB instructions. CALL does not correspond to SB.
15	E	E	JMP and INT instructions are used in the same step.
20	F	fE	Undefined operation coed or operand is used. So program cannot be interpreted. Or the user memory area is not formatted normally.
30	E	E	Error is detected in check sum of user program.

・At occurrence of an error, its code can be observed through decimal monitoring of WM980. CLR 9 8 0 MON MON

・These error codes are not cleared by turning on/off power supply (they are retained in memory).

5

Function	Programmer mode		
	PROG	TEST	RUN
Operation and stop	×	○	○



[Explanation]

1. Operation and stop are controlled according to the input codition of FUN98 (STA: Start). The start signal is processed in the basic unit along the above sequence as follows.
   (1) Operation with programmer
   - ○ Operation starts when the start signal turns ON (external input X7 turns ON in the above examlple) with the mode selector switch of programmer set at RUN or TEST. Operation will not start in the PROG mode.
   - ○ Mode cannot be changed over by manipulating the mode selector during operation. Therefore, operation will continue even if the mode selector switch is turned to PROG once operation has started.

   (2) Operation without programmer
   Operation starts when the start signal turns ON.
   (3) Operation starts when turning on power supply with the start signal turned ON.
2. As soon as operation starts, the RUN lamp of CPU module comes on.
3. Operation stops when the start signal turns OFF.
4. The programmer is mountable and dismountable while the basic unit is energized. This brings about a mode change in the basic unit.
   (1) When dismounting the progrmammer, the basic unit is set in the same status as when turning the mode selector switch of programmer to RUN.
   (2) When mounting the programmer during operation, operation continues indifferently to programmer mode. For matching the mode of basic unit with the mode selector switch setting of programmer, stop operation or turn off and then on the power supply to the basic unit.
5. For programming of start, refer to "Start and end" in "4.2Application Instruction (I)."
6. In case operation and stop are programmed by using a personnal compoter (runnning E-LADDER), the contacts like X7 in the above example must be kept open.

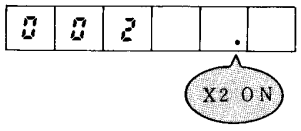
Function	Component	Programmer mode			Operational status	
Monitor	X, Y, M, T/C	PROG	TEST	RUN	Operation	Stop
	WX, WY, WM	○	○	○	○	○

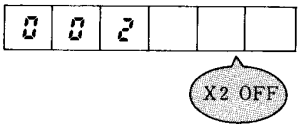
Classification			Key-in procedure	Display: Instruction	Display: Numerical display	Display: Mode display	Remarks
Bit	X, Y, M		CLR 2 MON		002 .	・DATA (・PROG ・TEST ・RUN)	X2 ON
	T/C	I/O	CLR T/C 1 0 MON	・T/C	010 .		T/C10 ON
		Coil	CLR OUT T/C 1 1 MON	・OUT ・T/C	11.065		T/C11 current value
Word	WX, WY		CLR 4 0 0 MON		400		M400 (bit)
	WM		MON		00255		WM400 (decimal)
	T/C (current value, preset value)		MON		00FFH		WM400 (hexadecimal)
			MON		400		M400 (bit)

[Explanation]

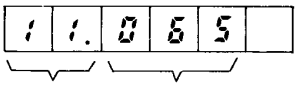
1. Monitoring of bit data (X, X, M, T/C) can be done as shown below.

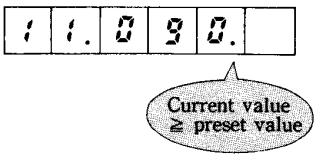
(1) The ON/OFF status of bit data is indicated by means of a decimal point at the second lowest digit of the numerical display.





(2) The contacts of timer and counter are monitored in the same way as above. When the coil of timer/counter is monitored, its current value is displayed simultaneously. This value is displayed simultaneously. This value is incremented. When the current value becomes equal to or larger than the preset value, a decimal point appears at the second lowest digit of the numerical display.





(3) The number of bit data is incremented or decremented by 1 whenever pressing the STEP + of STEP − key in succession to MON. Therefore, ON/OFF status can be checked for successive numbers in both directions.

2. Monitoring of word data (WX, WY, WM, T/C ‹current value, preset value›) can be done as shown below. After monitoring of bit data, word data can be monitored by pressing the [MON] key.
   (1) When pressing the [MON] key, word data is displayed in decimal notation.
   (2) When pressing the [MON] key again, word data is displayed in hexadecimal notation.
   (3) When pressing the [MON] key again, display returns to monitoring of bit data.
   (Example) The method of monitoring WM400 and WM402 word data is shown below.

M400	0	0	0	0	0	0	0	0
M401	1	1	1	1	1	1	1	1
M402	0	0	0	0	0	0	0	0
M403	0	0	0	0	1	0	1	0

WM400……"255" in decimal number and "FF" in hexadecimal number

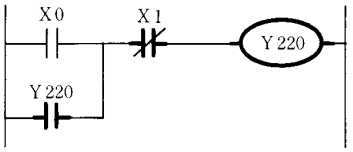
WM402……"10" in decimal number and "A" in hexadecimal number

Key-in procedure	Display	Description
[CLR] [4] [0] [0] [MON]	400	Bit monitorring of M400
[MON]	00255	Decimal monitoring of WM400
[MON]	00FFH	Hexadecimal monitoring of WM400 "H" indicates hexadecimal notation.
[MON]	400	Bit monitoring of M400
[STEP +]	401	Bit monitoring of M401
[STEP +]	402	Bit monitoring of M402
[MON]	00010	Decimal monitoring of WM402
[MON]	000AH	Hexadecimal monitoring of WM402
[MON]	402	Bit monitoring of M402

3. Monitoring is possible even in the stop status. However, the contents of external input to be monitored during stop correspond to the ON/OFF status just before stop.

Function	Programmer mode			Operational status	
Conduction check	PROG	TEST	RUN	Operation	Stop
	×	○	○	○	×

· Sequence



Contacts X1 Y220 — Conductive

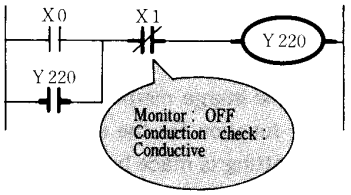
Contacts X0 — Non-conductive

· Key-in procedure and display

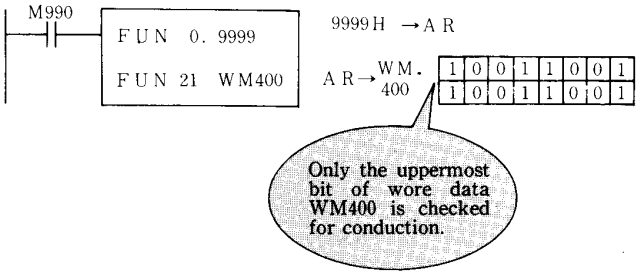
Key-in procedure	Display			Remarks
	Instruction	Numerical display	Mode display	
CLR OUT 2 2 0 SRC	· OUT	220 .	· DATA (· TEST · RUN)	Y220 ON
STEP	· AND · NOT	001 .		X1 conductive
STEP	· OR	220 .		Y220 conductive
STEP	· ORG	000		X0 non-conductive

[Explanation]

1. This functios enables you to check the contacts contained in the circuit sequentially for conduction. That is, when a contact set is conductive, a decimal point (.) will appear at the second digit counting from the lowest one of the numerical display.
2. Key-in procedure for conduction check is the same as for search and read-out.
3. Difference between conduction check and monitor
   (1) The monitor is a function for displaying the ON/OFF status of coil irrespective of sequence.
   (2) Conduction check is a function for displaying the conductive or non-conductive status of contacts while following the sequence.



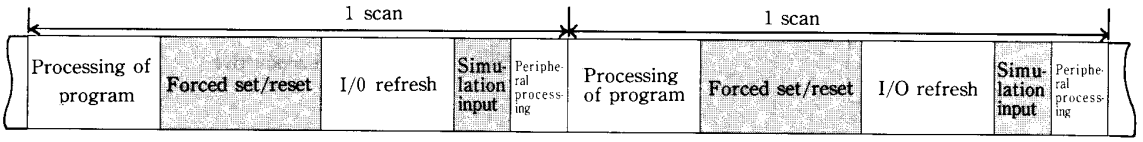
4. Conduction check of word data covers only the uppermost bit (b15) of that data, which does not have any significance. So word data must be checked by using the monitor function.



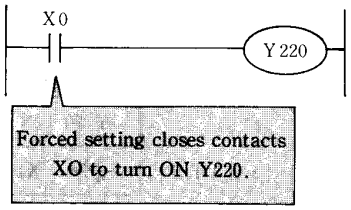
5

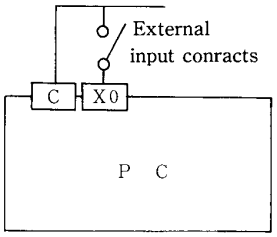
Function	Component	Programmer mode			Operational status	
Forced set/reset (bit)	X, Y, M	PROG	TEST	RUN	Operation	Stop
	T/C	×	○	○	○	×

· Processing timing of forced set/reset within one scan



· Sequence





· Operation

X, Y, M or T/C on image memory is set/reset indifferently to the ON/OFF status of external input conracts.

· Key-in procedure and display

Key-in procedure	Display			Remarks
	Instruction	Numerical display	Mode display	
CLR OUT 0 MON		000	· DATA (· RUN · TEST)	Monitors X0.
SET		000 .		Forcibly sets X0 and turns on Y220 simultaneously.
RES		000		Forcibly resets X0 and turns off Y220.
CLR				Releases forced set/reset mode. Operation follows the ON/OFF status of external input contacts.

[Explanation]

1. When forced set/reset function is activated, X, Y, M or T/C on the image memory is set or reset.
2. Simulation input is enabled by utilizing the forced set and reset function for the external input (X). However, the input indicator lamp does not turn on when forced setting is made because the lamp responds to the physical conditions of the contacts. So judge input by activating the monitor function of programmer.
3. The forced set/reset mode is released by pressing the keys for interrupting the monitoring in the relevant I/O number such as CLR, STEP + and STEP −.

Function	Component	Programmer mode			Operational status	
		PROG	TEST	RUN	Operation	Stop
Forced setting of decimal/hexadecimal numbers (word)	WY, WM T/C100~295	○	○	○	○	○

Furced setting of desimal number

· Key-in procedure

Key-in procedure	Display			Remarks
	Instruction	Numerical display	Mode display	
CLR OUT 7 0 0 MON	· OUT	700	· DATA	Bit monitoring of M700
MON	· OUT	00000	(· PROG · TEST · RUN)	Decimal monitoring of WM700
1 2 3 4 5	· FUN · OUT	12345		Decimal number in 5 digits
SET	· OUT	12345		Forced setting of desimal number to WM700

Forced setting of hexadecimal number

· Key-in procedure

Key-in procedure	Display			Remarks
	Instruction	Numerical display	Mode display	
CLR OUT 8 0 0 MON	· OUT	800	· DATA	Bit monitoring of M800
MON	· OUT	00000	(· PROG · TEST · RUN)	Decimal monitoring of WM800
MON	· OUT	0000H		Hexadecimal monitorring of WM800
0 1 2 3	· FUN · OUT	0123		Hexadecimal number in 4 digits
SET	· OUT	0123H		Forced setting of hexadecimal number in WM800

5

[Explanation]

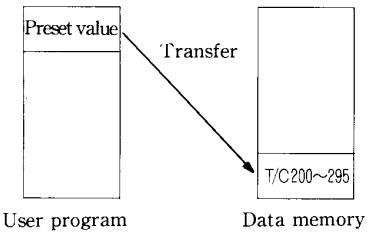
1. When entering a value and pressing the SET key while the monitor function is activated, the value is set as word data. Before pressing the SET key, be sure to enter decimal number in 5 digits, and a hexadecimal number in 4 digits.
2. Clock time can be set by forced setting of a hexadecimal number.
3. Listed below are applicable range of monitor and forced set/rset functions.

Table 5-6 Application Range of Monitor and Forced Set/Reset Functions

Classification	Monitor			Forced set/reset					
				Under stop			Under oparation		
	Bit	Decimal number	Hexadecimal number	Bit	Decimal number	Hexadecimal number	Bit	Decimal number	Hexadecimal number
X 0~195	○	○	○	×	×	×	◎	×	×
Y 200~395, M400~655	○	○	○	×	×	×	◎	○	○
M700~ 955	○	○	○	○	○	○	◎	○	○
M 960~991	○	○	○	×	×	×	×	×	×
T/C 0~95	○	×	×	×	×	×	◎	×	×
T/C 100~195, T/C 200~295	○	○	○	(Note) ○	(Note) ○	(Note) ○	×	○	○

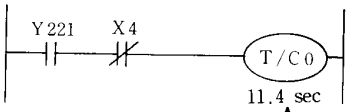
Symbol
- ○ : Possible
- × : Impossible
- ◎ : Possible (write-in every scan)

(Note) Just before operation, the preset values of timers/counters are transferred to T/C200 to 295. Therefore, rewriting the contents of T/C200 to 295 during stop, though possible, is meaningless because the contents are all replaced with the preset values before operation.



Function	Component	Programmer mode			Operational status	
		PROG	TEST	RUN	Operation	Stop
Change of timer/counter preset value during operation	T/C	×	○	×	○	×

· Sequence



· Key-in procedure and display

Key-in procedure	Display			Remarks
	Instruction	Numerical display	Mode display	
CLR OUT T/C 0 SRC	· OUT · T/C	0.011.4	· DATA	Search for T/C0 coil
1 3 • 5 ENT	· OUT · T/C	0.13.5	· TEST	Write-in of new preset value

[Explanation]

1. Each preset value of timers and counters is changeable by turning the programmer mode to TEST during operation. The value is unchangeable in the RUN mode.
2. Search for the coil of timr or counter whose preset value is to be changed. Then key in a new preset value and press the ENT key. The new value will be written in EEPROM of the basic unit and T/C200 through T/C295, and the current value will be reset to 0 sec or 0 time.

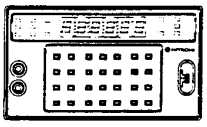
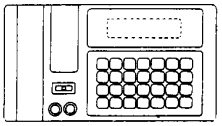
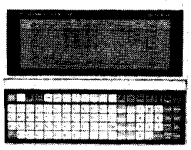
NOTE

If preset value is changed during operation, both timer and counter operate according to the new value immediately after change.

5

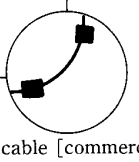
Function	Programmer mode			Operational status	
	PROG	TEST	RUN	Operation	Stop
CMT function	○	×	×	×	○

· Configuration

PGMJ　　PGMJ－R2　　PGM－GPE2

Audio cassette tape recorder [commercially available one]



Connecting cable [commercially available one] (without resistor)

· Key-in procedure and display (when using PGMJ)

	Function	Key-in procedure: Tape recorder	Key-in procedure: Programmer	Display: Instruction	Display: Numerical display		Display: Mode display	Remarks
1	CMT function setting		CLR SET SET ENT FUN 1		R - - -		· PROG · DATA	
					C - - -	CMT function		
2	Recording (DUMP)	Recording MIC ◎-□-□-◎ Microphone (Programmer) (Tape recorder)	OUT ENT	· OUT	C - - P	Recording		Basic unit (EEPROM) ↓ Cassette tape
					C - - -	End		
	Playback (LOAD)	Playback EAR ◎-□-□-◎ Earphone (Programmer) (Tape recorder)	STR ENT	· STR	C - - R	Waiting for start bit (30 sec)		Basic unit (EEPROM) ↑ Cassette tape
					C - - P	Playing back		
					C - - -	End		
	Verification (VERIFY)	Playback EAR ◎-□-□-◎ Earphone (Programmer)(Tape recorder)	AND ENT	· AND	C - - R	Waiting for start bit (30 sec)		Basic unit (EEPROM) ↕ Cassette tape
					C - - P	Verifying		
					C - - -	End		
	Error display			· Instruction	C - - E	Operation error		Press CLR key to clear error and retry.
					C 6 2 E	Playback error		
					C 7 - E	Verification error		
					C 8 - E	Format error		
3	CMT function clear		CLR RES RES ENT					

※ Be sure to verify data after every recording or playback.

● Setting of cassette tape recorder

Item	Description
Type of cassette tape recorder	Use a monaural cassette tape recorder.
Tone quality	Set the tone adjusting knob to maximum.
Tone volume	Set the tone volume knob to maximum.
Tape	Select a tape not scratched nor wrinkled.

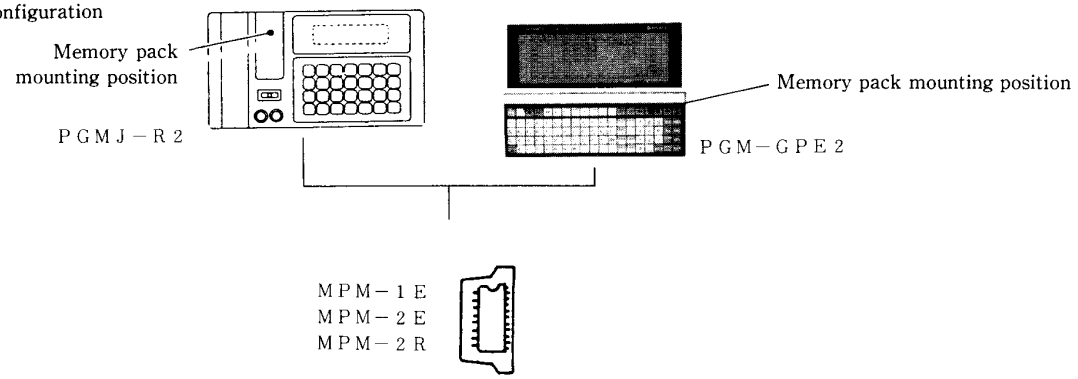
[**Explanation**]

1. Programs are storable on a cassete tape by using a commercially available tape recorder.
   Key-in procedure and display are exemplified above when the above-mentioned PGMJ is used. Key-in precedure remains the same when using the PGMJ-R2 instead of the PGMJ.
2. Be sure to rewind the tape to the beginning before recording, playing back or verifying a program.
3. If power is turned off, tape is taken out, or the CLR key is pressed during a process, then key-in precedure must be restated from the beginning.

4. For data palyback or verification, symbol H is presented on the LED for about 30 sec until the tape is positioned at the start bit. If the symbol does not disappear even after 30 sec, it can be judged that nothing has been recorded on the tape. In this case, record data again or change the tape to a proper one.
   Recording becomes impossible when using a tape recorder cord with a resistor. So be sure to use a cord without a resistor.
5. When a stereo cassette tape recorder is to be used, set the tape monaurally. In addition, turn the tone volume and balance knobs on the connection terminal side to the maximum position.
6. Execution time will be increased according to the number of program steps.
   Execution time $\fallingdotseq$ 40sec + number of steps x 0.22 sec
7. A tape recorded with the PGMJ can be played back with the PGMJ-R2 or PGM-PGM-GPE2. However, a tape recorded with the PGMJ-R2 or PGM-GPE2 cannot be reproduced with the PGMJ.
8. For key-in prodcedure with PGM-GPE2, refer its instruction manual.

Function	Programmer mode			Operational status	
	PROG	TEST	RUN	Operation	Stop
ROM writer function	○	×	×	×	○

· Configuration



Memory pack for EM series

· Key-in procedure and display (when using PGMJ-R2)

No.	Function	Key-in procedure	Display	Remarks
1	Entrance into ROM	CLR SET SET ENT FUN 2	PROG P－ 2 ROM MODE	
2	Recording (copying)	OUT 0 0 ENT	PROG R－P OUT00 2 ROM MODE	Basic unit →Memory EEPROM pack
	Reproduction (load)	STR 0 0 ENT	PROG R－P STR00 2 ROM MODE	Baisc unit ← Memory pack
	Verification (verify)	AND 0 0 ENT	PROG R－P AND00 2 ROM MODE	Basic unit ↔ Memory pack
	Blank check	NOT 0 6 ENT	PROG R－P NOT 2 ROM MODE	EPROM erasure check
	Error display	Key-in error	R－E	
		Copying error	R62E OUT	Exchange memory pack.
		Verification error	R7－E AND	
		Blank chenk error	R61E NOT	EPROM not yet erased
3	Releasing of ROM function mode	CLR RES RES ENT		

Note:"P"display disappears when procedure is completed.

[Explanation]

1. Programs can be stored in the memory pack by using the PGMJ-R2.
2. For recording (copying), the kind of memory must be keyed in.

[0] [0] ···925 Word EEPROM (MPM－1E)
[0] [1] ··1949 Word EEPROM (MPM－2E)
[0] [2] ··1949 Word EPROM (MPM－2R)
[0] [4] ··3997 Word EEPROM (MPM－2E)
[0] [6] ··3997 Word EPROM (MPM－2R)

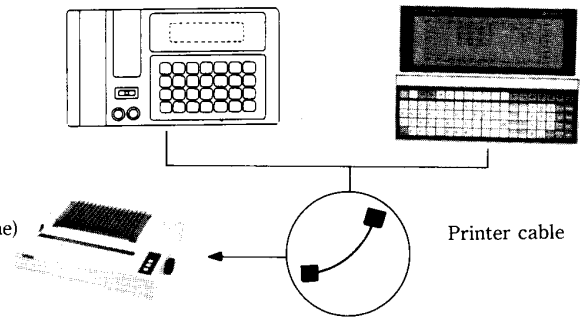
CAUTION

If EPROM is specified even though the EEPROM memory pack is used, the memory pack might be destroyed. To prevent this, the kind of memory must be confirmed.

3. If attempting to reproduce data with no memery pack mounted, undefined data is written in the basic unit.
4. For key-in procedure with PGM-GPE2, refer to its instruction manual.

Function	Programmer mode			Operational status	
Print-out	PROG	TEST	RUN	Operation	Stop
	○	×	×	×	○

· Configuration



· Key-in procedure and display (when using PGM-R2)

No.	Function	Key-is procedure	Display	Remarks
1	Change to printer function mode	CLR SET SET ENT FUN 6	PROG P － PRINT － OUT	FUN6 must be specified when the basic unit belongs to EM-II.
2	Specification of print-out	OUT 0 2 ENT ↑ Ladder diagram	PROG P － P－OUT02 PRINT－OUT	Basic unit → Printer EEPROM
3	Releasing of printer function mode (Note)	CLR RES RES ENT		

Note: Procedure is completed when "P"display disappears.

Specification of print-out format

Format specification (key operation)	Title	Code list	Ladder diagram	Cross reference
0 0	○	○	○	○
0 1	－	○	－	－
0 2	－	－	○	－
0 3	－	－	－	○

○ : Print-out

[Explanation]

1. Programs can be printed out onto a printer connected to the PGMJ-R2 or PGM-GPE2. Code list, ladder diagram and/or cross reference is selecteable for print-out. In case of PGMJ-R2, the printout format is to be specified in any case of codes 00 through 03 as listed above.
2. FUN6 must be specified for changeover to the printer function mode when the EM-II series (CPM-E2 or CM-E3) is used.

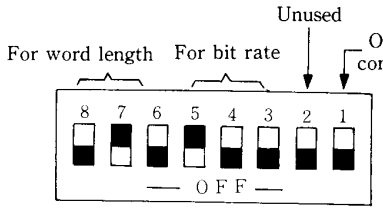
Although print-out is possible with the preceding PGMJ-R, it is restricted as listed below.

Table 5-7 Restrictions on Print-out with PGMJ-R

Item	Before June, 1989 (up to 9EXX)	From June, 1989 (from 9FXX on)
Change to printer function mode	FUN5 (specifies EM)	FUN6 (specifies EM-II)
Instructions commonly used between EM and EM-II	Can be printed	Can be printed
Instructions exclusive for EM-II	Cannot be printed correctly	Can be printed
Print-out capacity	2K words	2K words

3. DIP switch setting of PGMJ-R2 and PGM-GPE2

Bit rate and word length are changeable by the internal DIP switch. Settable bit rates and word lengths are listed in Tables 5-8 and 5-9, respectively.



Use a small blade-edge screwdriver for changing the settings of DIP switch.

■ indicates the current switch position.

Table 5-8 Bit Rate Setting

Switch No.			Bit rate (kbps)	Remarks
5	4	3		
ON	ON	ON	38.4	
ON	ON	OFF	19.2	
ON	OFF	ON	9.6	
ON	OFF	OFF	4.8	Setting made upon shipment
OFF	ON	ON	2.4	
OFF	ON	OFF	1.2	
OFF	OFF	ON	0.6	
OFF	OFF	OFF	0.3	

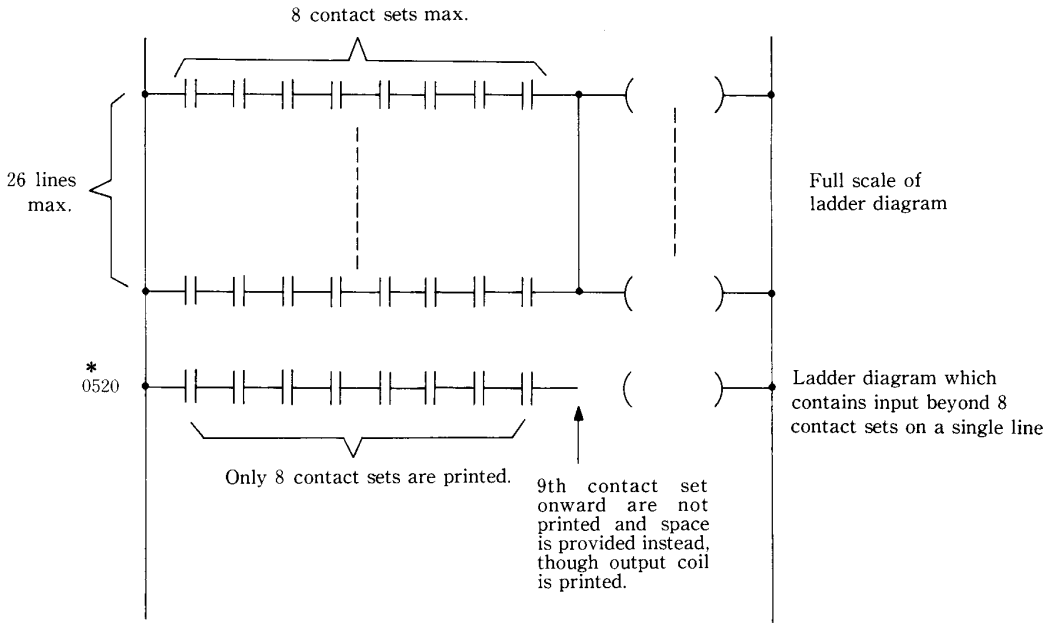
Table 5-9 Setting of Word Length

Switch No.			Word length				Remarks
8	7	6	Start bit	Data bit	Parity bit	Stop bit	
ON	ON	ON	1	7	1 (even number)	2	
ON	ON	OFF	1	7	1 (odd number)	2	
ON	OFF	ON	1	7	1 (even number)	1	
ON	OFF	OFF	1	7	1 (odd number)	1	
OFF	ON	ON	1	8	–	2	
OFF	ON	OFF	1	8	–	1	Setting made upon shipment
OFF	OFF	ON	1	8	1 (even number)	1	
OFF	OFF	OFF	1	8	1 (odd number)	1	

4. Print-out specifications of printer are explained below.

(1) Ladder diagram can be printed out normally when it contains 8 contact sets max. on each of up to 26 lines and when the number of concurrent blocks is within 8.

(2) If a ladder diagram exceeds the above limits, it is printed out only within the limits.
For example, when 10 contact sets are written on a single line, only 8 contact sets are printed excluding the ninth and tenth contact sets. In case the horizontal limit is exceeded, the asterisk "*" is printed at first step of the relevant circuit.



(3) Up to 9 contact sets can be printed on each line when using the PGM-GPE2.

5. Printer specifications

Connectable prints are limited to the ones made by ESPON*1. The table below lists the combinations of connectable printers and serial interface boards.

Table 5-10 Connectable Printers and Interface Boards

Printer model / Interface board	RP-80*2 (old version)	RP-8011*2 (old version)	FP-80*2 (old version)	SP-80T*2
No. 8143*2 (old version)	○	○	○	○
No. 8145*2 (old version)	○	○	○	○
No. 8148*2	○	○	○	○

Setting of the DIP switch in the printer have not been changed from those made upon shipment from the factory. Setting of the DIP switch on the interface board are listed on the next page.

*1. EPSON is a trademark of SEIKO EPSON corporation.

*2. RP-80, RP-8011, FP-80, SP-80T and Interface boards No. 8143, No. 8145, No. 8148 are products of SEIKO EPSON corporation.

Table 5-11 Settings of DIP Switch on Interface Board

Switch No. \ Interface board		No.8143	No.8145	(Note) No.8148
SW 1	1	ON	OFF	OFF
	2	OFF	ON	OFF
	3	ON	OFF	OFF
	4	OFF	OFF	OFF
	5	OFF	OFF	OFF
	6	OFF	OFF	ON
	7	OFF	ON	OFF
	8	ON	OFF	ON
SW 2	1		OFF	ON
	2		ON	OFF
	3		OFF	ON
	4		ON	ON
	5			OFF
	6			OFF

Note: The interface board No. 8148 requires shorting of the jumper wire J6.

6. Cable specifications

The cables for connecting the PGMJ-R2/PGM-GPE2 and serial printer are not included in the standard equipment. They must be prepared separately. If utilizing cables in your possesion, confirm beforehand that connection meets the figure below.

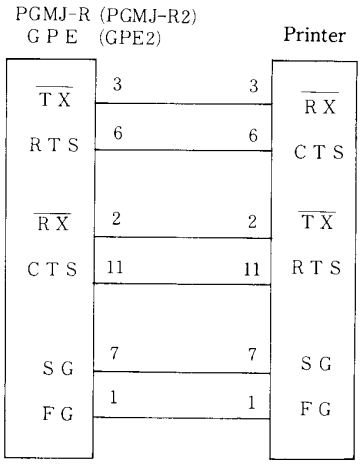
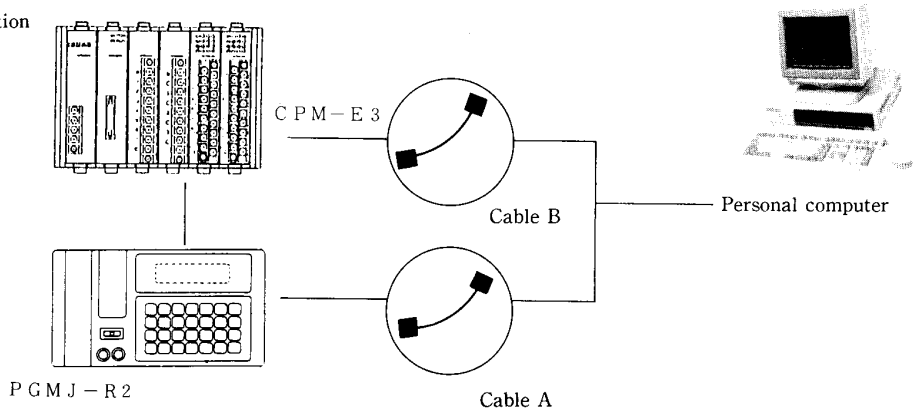


Figure 5-5 Printer Cable Connection Diagram

5

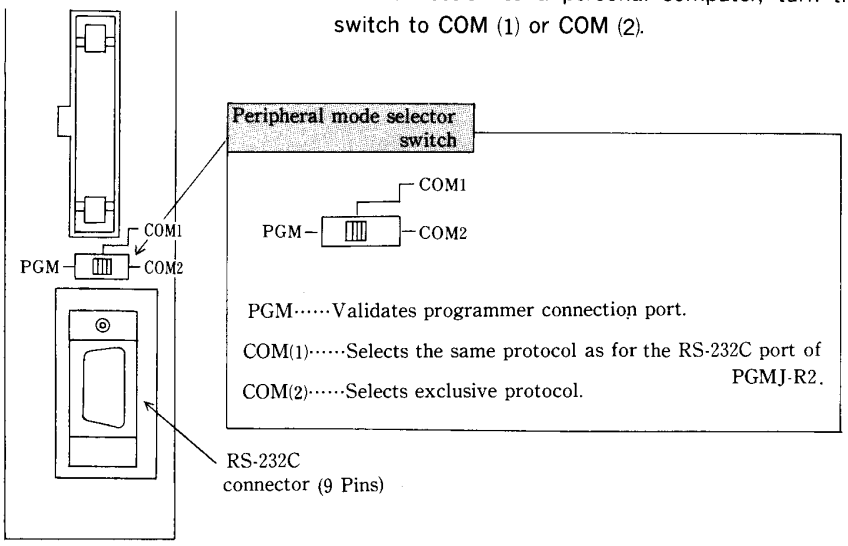
Personal computer interface function

· Configuration



[Explanation]

1. Since the CPM-E3 comprises an RS-232C interface, it is directly connectable to a personal computer. The CPM-E2 can be hooked up to a personal computer via the PGMJ-R2.
2. **Connection via PGMJ-R2**
   A personal computer is enable to program and monitor sequence by running the personal computer programming software E-LADDER. This software package has been prepared for IBM PC XT* series. For details, refer to the personal computer programming manual 〈E-LADDER〉 (NJI 022 (X) -1).
3. **Direct connection to personal computer [CPM-E3]**
   For connection to a personal computer, turn the peripheral mode selector switch to COM (1) or COM (2).



At the PGM position, the programmer connection port is validated. At COM (1) or COM (2) position, the RS-232C port is validated, namely the programmer connection port is unusable.

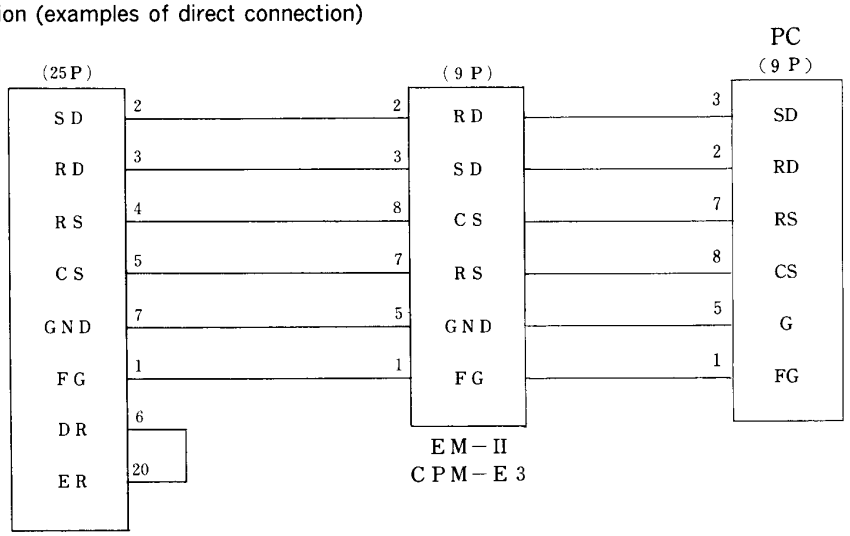
NOTE

The status of the peripheral mede selector switch is determined just when turning on power supply. Alteration of switch setting after energization is ineffective. For mode change, power supply must be turned off.

(1) At the COM (1) position, programming and monitoring are possible by use of the personal computer programming software E-LADDER.
In this mode, protocol is the same as in connection to a personal computer via the PGMJ-R2.

(2) At the COM (2) position, the exclusive protocol is selected.
Refer to the EB/EM-II protocol manual separately issued.

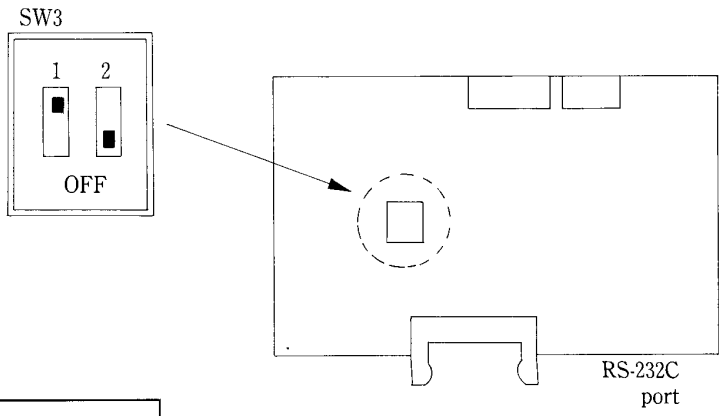
4. Cable connection (examples of direct connection)



* IBM PC XT is a product of International Business Machine corporation.

By the DIP switch (SW3) Position on PC board at power supply energized, the mode of CPM-E3 is decided.
After this time, CPU mode can not be changed by changing switch position.
The switch postion is shown as follows.

1 : Transmission speed control switch
2 : Operation control command selector



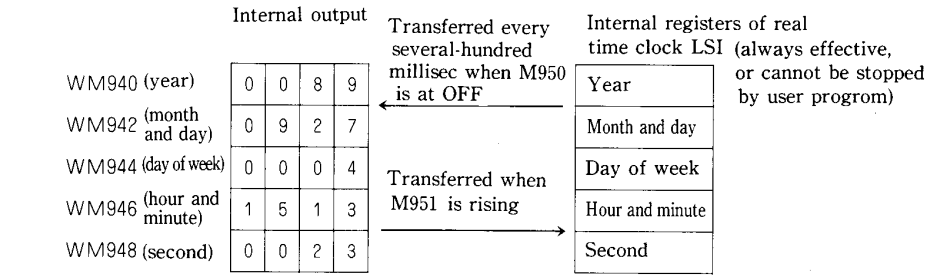
transmission speed (bps)	transmission speed switch Position
4800	OFF
9600	ON (preset state at shipment)

mode	operation control command	peripheral node selector	operation control selector
COMMAND MODE	uneffective	COM2 side	ON
	effecttive	COM2 side	OFF*

*Preset state at shipment

Clock function

Configuration of calendar clock (15:13:23, Wenesday, September 27, 1989)



0/1 --- M950 (selects display or editing mode)

0/1 --- M953 (turns on at battery error)

⊥ --- M951 (sets time point)

⊥ --- M952 (adjusts in ±30 sec)

- For year, the lower 2 digits of Christian year are represented by those of WM940, whose upper 2 digits are fixed at 00H.
- Month and day represented by the upper and lower 2 digits of WM942, respectively.
- Day of week is represented by the lower 2 digits of WM944 in the following way. The upper 2 digits are fixed at 00H.

Day of week	Sun.	Mon.	Tues.	Wed.	Thur.	Fri.	Sat.
Lower 2 digits of WM944	01H	02H	03H	04H	05H	06H	07H

- Hour and minute are represented by the upper and lower 2 digits fo WM946, respectively.
- Second is represented by the lower 2 digits of WM948, whose upper 2 digits are fixed at 00H.

[Explanation]

1. Registers (M940 to M955) for calendar clock

Only the CPM-E3 uses 16 internal outputs M940 through M955 as registers for calendar clock. (The contents of these registers cannot be cleared by CLR, ENT and DEL operations.)

In case of the CPM-E2, the internal outputs M940 through M955 serve as usual memory-retentive internal outputs.

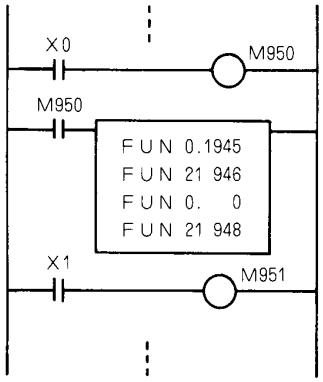
The real time clock LSI has registers for representing year, month,day of week, hour, minute and second. These registers are always operative indifferently to operation/stop and power on/off status.

Example of time setting with programmer (setting of 19:45:36, Saturday, July 28, 1990)

Key-in procedure	Display	Description
CLR OUT 9 5 0 MON	950	Bit monitoring of M950
SET	950.	Forced setting of M950 (editing mode)
CLR OUT 9 4 0 MON MON MON	0089H	Hexadecimal monitoring of WM940
0 0 9 0 SET	0090H	Forced setting of hexadecimal number (year) in WM940
STEP+ STEP+	0927H	Hexadecimal monitoring of WM942
0 7 2 8 SET	0728H	Forced setting of hexadecimal numbers (month and day) in WM942
STEP+ STEP+	0004H	Hexadecimal monitoring of WM944
0 0 0 7 SET	0007H	Forced setting of Hexadecimal number (day of week) in WM944
STEP+ STEP+	1513H	Hexadecimal monitoring of WM946
1 9 4 5 SET	1945H	Forced setting of hexadecimal numbers (hour and minute) in WM946
STEP+ STEP+	0023H	Hexadecimal monitoring of WM948
0 0 3 6 SET	0036H	Forced setting of hexadecimal number (second) in WM948
CLR OUT 9 5 1 MON	951 (Note)	Bit monitoring of M951
SET	951.	Forced setting of M951 (5 time setting)

Note: Make sure that M951 is at OFF. If at ON, press the RES key and then SET key. After completion of setting, press the RES key to forcibly reset M951.

Example of time setting with programmer (setting of 19:45:00)



X0: Turns on to write hour, minute and second values in WM946 and WM948.

X1: Sets time when status changes from OFF to ON.

5

(1) When M950 is OFF, the contents of registers in the real time clock LSI are transferred to the internal outputs M940 through M949 every several-hundred ms. So the user can know the current time point with the internal outputs.
When M950 is at ON, the contents of the same register are not transferred to the internal outputs M940 through M949. Therefore, time does not advance in the internal outputs M940 through M949. However, the registers in the real time clock LSI are functioning. So internal outputs M940 through M949 resume operation at the exact time point when turning off M950 again.

(2) M951 is used for rewriting the registers in the real time clock LSI. When M951 changes from OFF to ON (at the rising edge), data in the internal outputs M940 through M949 are transferred to the registers in the real time clock LSI and, at the same time, M950 is turned off.

(3) M952 is used for ±30sec adjustment. Time point is adjusted to 0 sec when current value is within 0 to 29 sec, and to 59 sec when within 30 to 59 sec.
At OFF to ON change of M952 (at the rising edge), only the registers in the real time clock LSI are subjected to ±30 sec adjustment. Within 1 sec, the adjusted contents of the registers are transferred to the internal outputs M940 through M949 when M950 is at OFF.

(4) M950, M951 and M952 are automatically turned off upon energization.

(5) M954 and M955 are for functional expansion and unused (undefined) at present.

2. Setting of time point

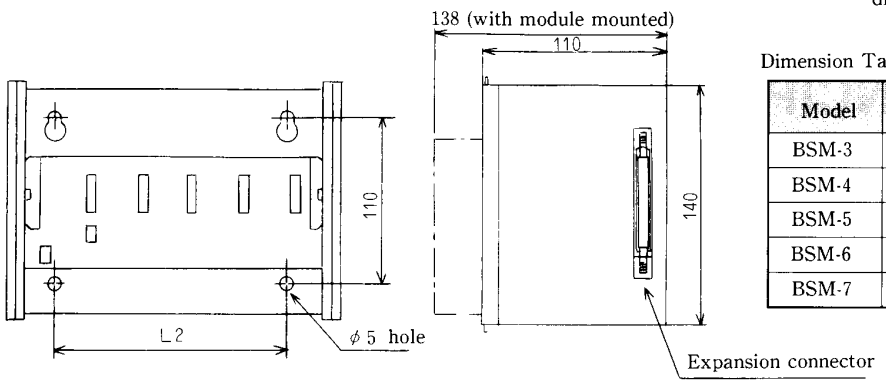
A desired time point is settable by turning M951 from OFF to ON after rewriting the contents of registers with M950 turned on to set the editing mode.
Time point is settable by utilizing the forced set/reset function of the programmer. It can also be set during operation as programmed.

3. Accuracy

Calendar clock has an accuracy of +30 sec and −3 min per month (at 0 to 45°C).
This accuracy may not be retained if ambient temperature rises beyond 45°C.

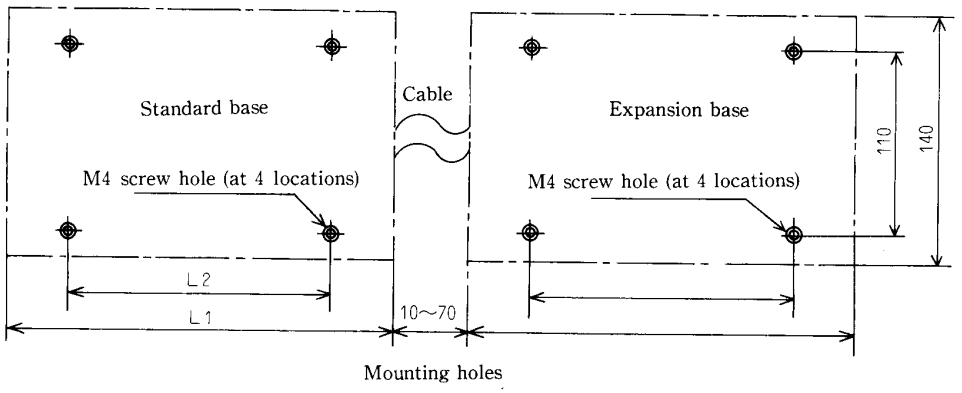
[External dimensions and mounting dimensions]

dimmensions in mm



Dimension Table

Model	L1 (external dimension)	L2 (mounting dimension)
BSM-3	160	80
BSM-4	195	120
BSM-5	230	160
BSM-6	265	200
BSM-7	300	240

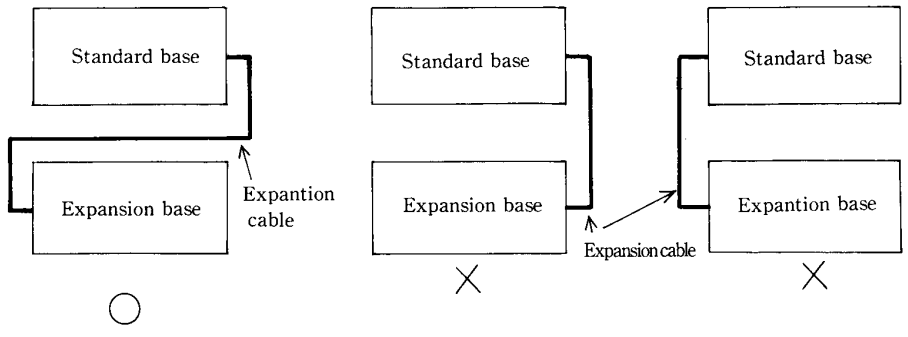


[Explanation]

1. The base alone with no module is to be mounted to control panel.
2. The expansion unit is to be installed at the right of the basic unit as a rule. Installation gap must be 10 to 70mm when using the 10 cm-long expansion cable ‹CNM-01›

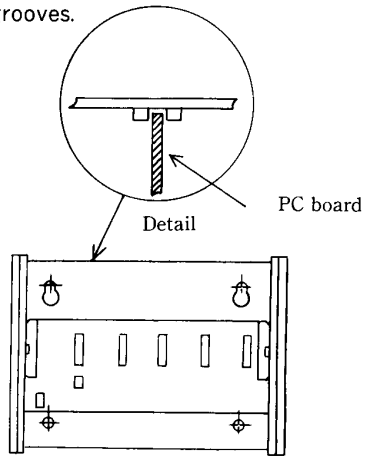
CAUTION

For vertical instllation with the 60 cm-long expansion cable ‹CNM-06› , attention must be paid to its connecting direction. If the cable is not connected correctly, not only will operation be impossible but the module might be broken.



3. Mounting of module

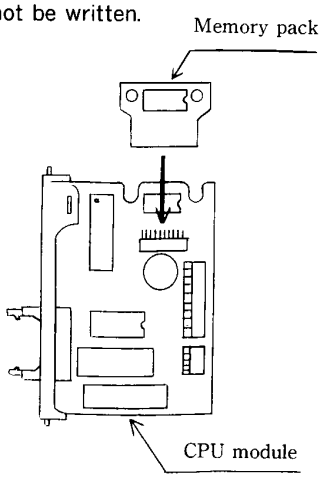
The base has grooves shown below at the top and bottom. Push in the PC board of each module while matching it with the upper and lower grooves.



Base sturcture

4. Mounting the memory pack

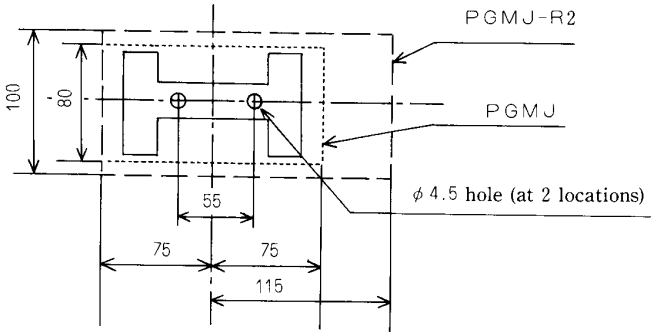
The CPU module necessitates a memory pack. Plug the memory pack into the 30P connector. Unless the memory pack is mounted, program cannot be written.



Mounting of Memory Pack

5. How to Mount Programmer

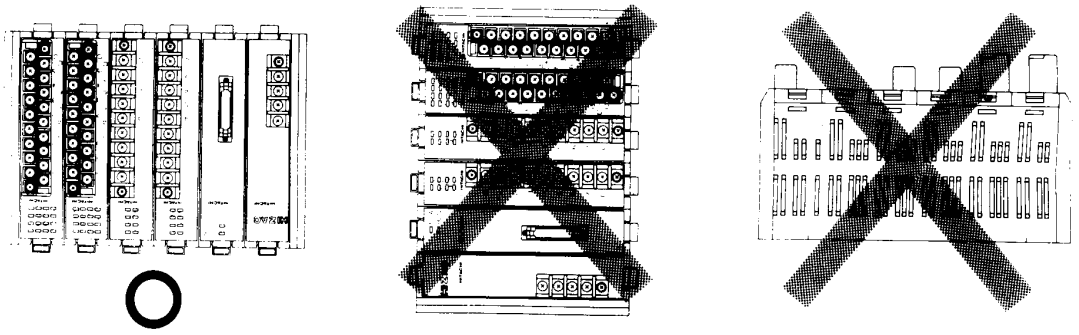
The programmer (model PGMJ) and universal programmer (model PGMJ-R2) must be mounted in the dimentions shown below when using the programmer mounting seat (model PAM-E).



Program Mounting Dimensions

6. Mounting direction

The programmable controller is mountable upside down, but neither vertical nor reverse installation is allowable.



—— NOTICE ——

1. Installation Clearance

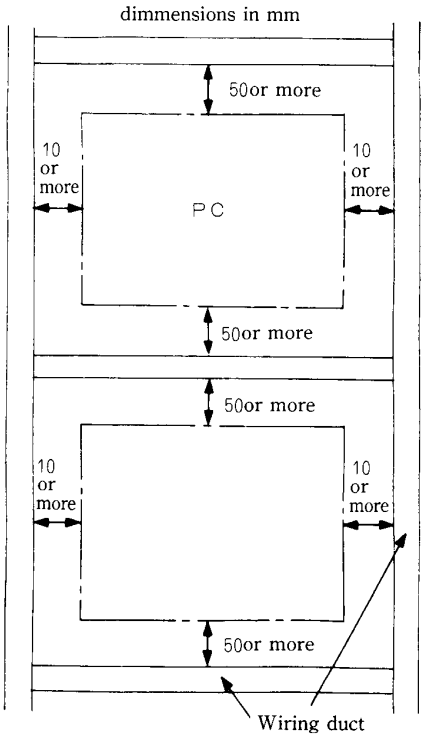
1. Provide a space of 50 mm or more at the top and bottom of each unit for facilitating ventilation and maintenance. Also secure a free space of 10 mm min. at the left and right for ventilation.
2. During installation, pay strict attention not to let fragments due to drilling or wiring fall into the programmable controller.
3. Avoid installation right above 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.

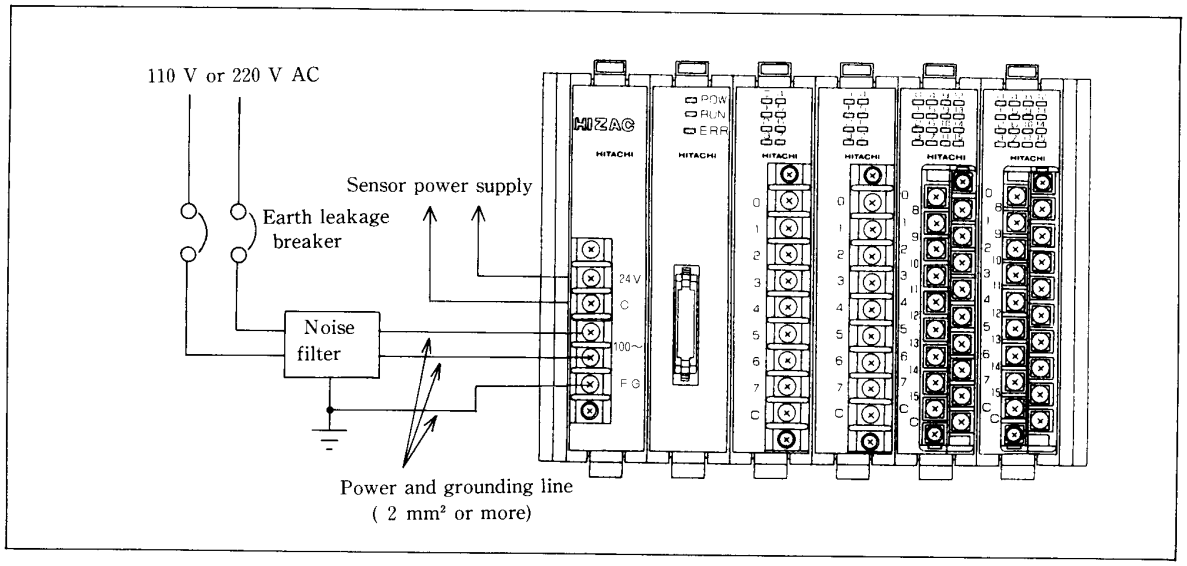
2. Installation environment

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

(2) Installation is unallowable at locations where the atomospheric air contains dust, oil vapor, smoke, conductive dust or corrosive gas in a significant amount.

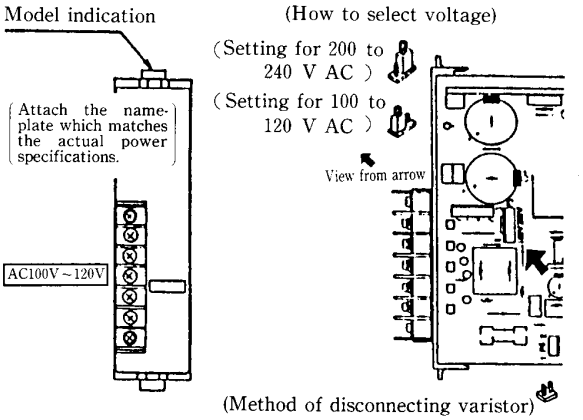
(3) Do not install the programmable controller at locations at which vibration or shock will be directly applied.



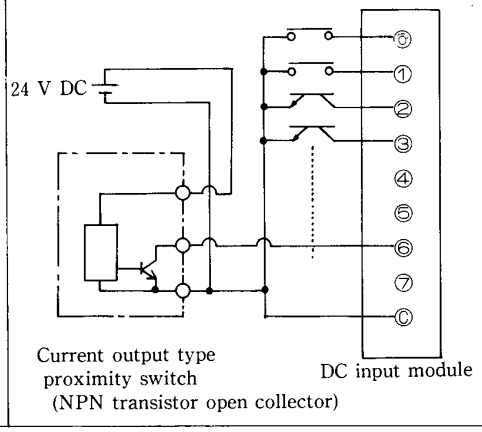
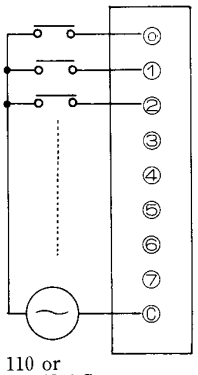


[**Explanation**]

1. **Line voltage**
   This instrument operates on either 110V or 220 V AC system. However, standard setting is 220 V AC (factory setting on shipment). For receiving 110 V AC, setting must be changed as shown at right.
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 100Ω or less using a cable of 2 mm² or more.
   Restrict the length of grounding cable within 20m.
   (1) Grounding can be shared with an instrument panel or relay panel.
   (2) Common grounding must be avoided with equipment which may generate high-level noise such as a high-frequency furnace, large-scale power panel (beyond a few kW), thyristor converter and electric welding machine.
   (3) In case line voltage fluctuates excessively, use of a noise filter is recommended.



4. **Insulation resistance and dielectric strength tests**
   A varistor (470 V class) is incorporated for suppressing a lightning surge. Before insulation resistance or dielectric strength test, be sure to disconnect the connector.
   If either test is conducted without disconnecting the connector, the power module might be damaged.

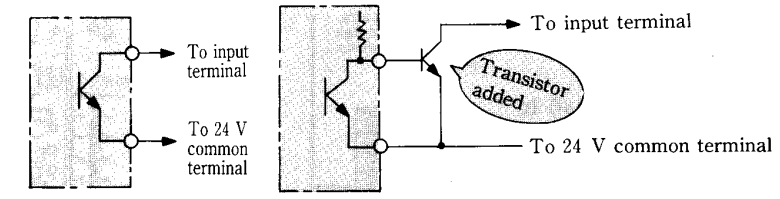
Item	DC input (PIM-D/DH, etc.)	AC input (PIM-A/AH, etc.)
External wiring		

[Explanation]

1. Wiring of DC input module (Example of negative logic input)

(1) The EM-II series 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 rule, a current of about 10 mA flows from the PC to the external input contacts.

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



○ (directly connectable) × (requiring addition of transistor)

(3) Although the instrument is sensitive to an input current within 4 to 6 mA, 7 mA or more for reliable ON operation and 1 mA or less for reliable OFF operation.
Note: For connecting a 2-wire type proximity switch, LED display-equipped limit switch or the like, confirm its input impedance and single out a sensor within the above current specifications.

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

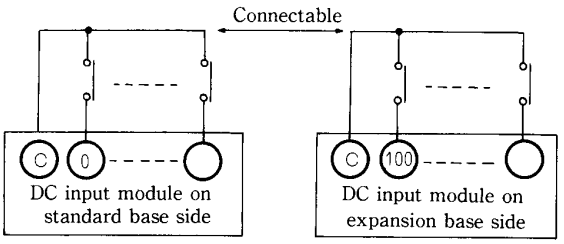
[I = CH3 capacity of PSM-A (450 mA) − CH3 current consumed by I/O module]

(5) For installing a switching regulator of 24 V DC for supplying power to the sensor, connect the negative pole of power supply with the common terminal of DC input module. (Refer to the above example of DC input wiring.)

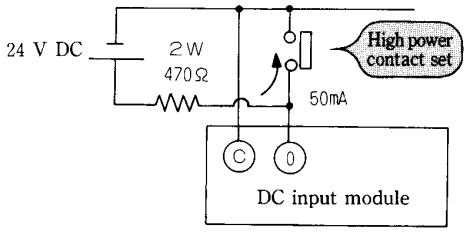
(6) Connection of common terminal of DC input module

○ The common terminals of DC input module need not be connected within the same base. (Because they are connected via the mother board in the base.)

○ It is recommended to separate the common terminals of DC input between the standard base and expansion base. However, if this is impossible, connection of the common terminals between these bases does not pose any problem.



(7) Prevention of poor contact of high power contacts



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

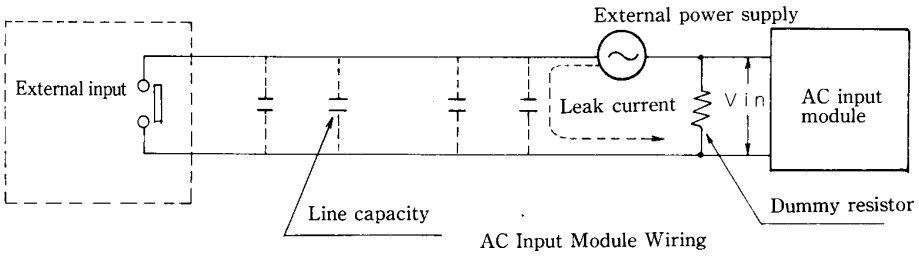
(8) Length of input wiring

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.

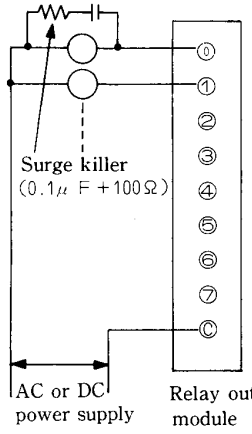
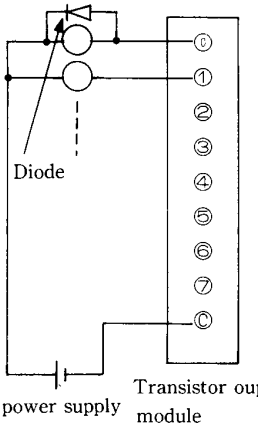
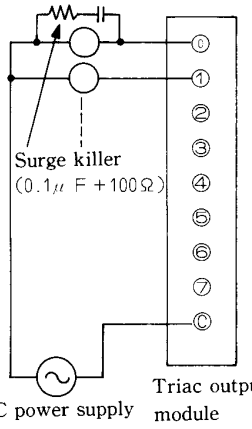
2. Wiring of AC input module

(1) With the AC input module, a voltage appears at the input terminal when wiring distance becomes long, though there is no signal actually.



AC Input Module Wiring

Even with the external input contacts open, if voltage applied to the input terminal because of leak current through line capacity exceeds the maximum OFF voltage of the input module, the module is under the same condition as when input signal is applied to it. Therefore, the module may operate. To prevent this, connect a dummy resistor in parallel of input module and thereby curb the terminal voltage due to electrostatic capacity to one half or less of the maximum OFF voltage of this module.

Item	Relay output (POM-R/RH, etc.)	Transistor output (POM-T/TH, etc.)	Triac output (POM-S/SH, etc.)
External wiring			

[**Explanation**]

1. Wiring of relay output module

(1) Service life of relay

○ Relay contacts are operable 200,000 times or more under a resistive load of 120 V AC, 2 A and 1,000, 000 times or more under a load of electromagnetic contactor (Hitachi H10 with an inductive load of 170 VA upon energization and 6 VA after energization).

○ Relay life is in inverse proportions to the square of current (life quadrupled by reducing current to one half). So the life will be significantly shortened when breaking rush current or directly driving a capacitor load. For opening/closing the contacts at a high frequency, use of the transistor module or triac output module is recommended.

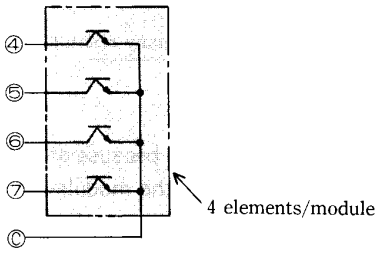
(2) Surge killer

In case of an inductive load whose coil capacity exceeds 10 VA, a surge killer (such as a 0.1 μF capacitor + 100Ω resistor combination) must be connected in parallel with the load. For DC load, connect a flywheel diode.

2. Wiring of transistor output module

(1) This module is used to control DC load. Although a protective circuit is incorporated against a surge which bauses inductive load, it is recommended to suppress the occurrence of surge by connecting a flywheel diode (current capacity 1 A and inverse dielectric strength 250 V as a standard) in parallel with inductive load.

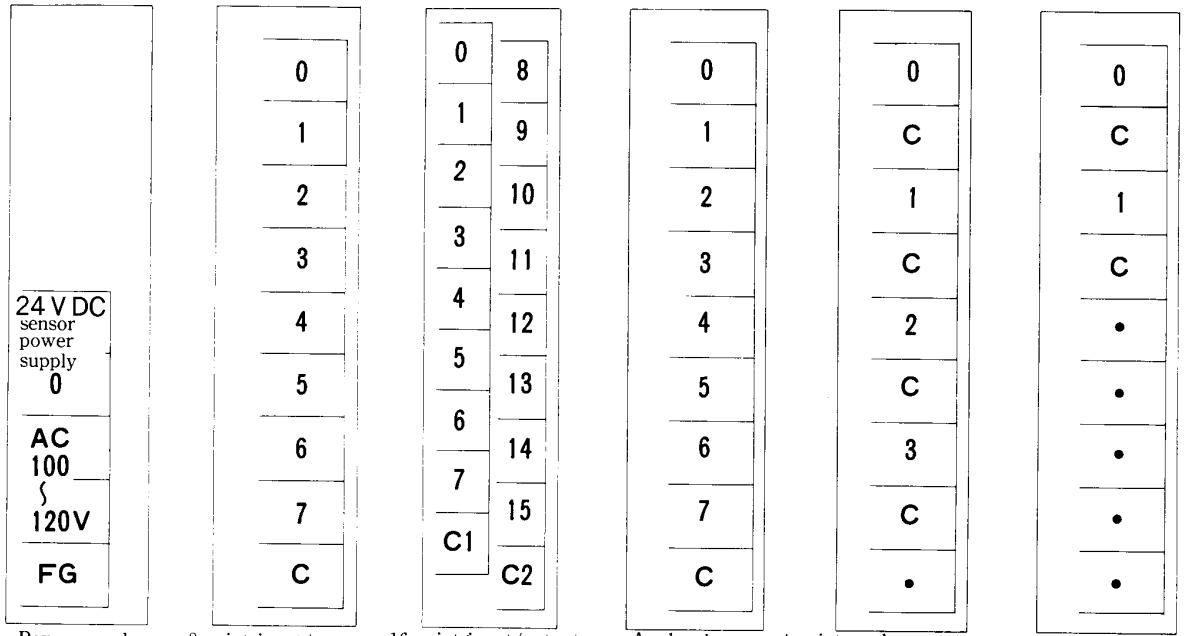
(2) The transistor is a composite part made up of 4 elements. Maximum current is restricted to 1.25 A for a total of 4 circuits which correspond to terminals 0 to 3, 4 to 7, 8 to 11 and 12 to 15. Allocate load so that maximum load current will not be exceeded.



3. Wiring of triac output module

(1) This module is used to control AC load.

(2) Leak current flows (3 mA at 220 V AC and 1.5 mA at 110 V AC) because a snubber circuit is comprised in the module for protecting the triac.
If a slight current load or lamp load is connected, the triac may be turned on in error or unable to be turned off. In such case, connect a dummy load (aforementioned surge killer of 0.1 μF + 100Ω) in parallel with the above load to prevent influence by leak current.



Power supply
PSM-A

8-point input/output
PIM-A,D
POM-R,S,T

16-point input/output
PIM-AH,DH
POM-RH,SH,TH

Analog input
AGM-I

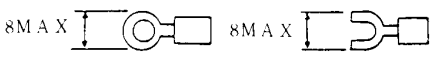
4-point analog output
AGM-O

2-point analog output
AGM-OD

*The DC power supply for sensor is shared by external input (9 mA/point). Its maximum current is rated at 450 mA.

[**Explanation**]

1. The above figure shows the terminal layout of EM-II series.
2. Each terminal is threaded in M3.5. When a solderless terminal is used, its outside diameter must be 8 mm max. Each terminal is allowed to hold a maximum of 2 solderless terminal tongues. Do not fasten 3 or more tongues at a time.



Function	Programmer mode			Operational status	
Forced output	PROG	TEST	RUN	Operation	Stop
	×	○	×	×	○

· Key-in procedure and display

Key-in procedure	Display			Remarks
	Instruction	Numerical display	Mode display	
CLR SET SET ENT		A - - -	• DATA • TEST	Specification of forced output mode RUN contacts ON
FUN 3		o - - - -		External output 200 ON
CLR OUT 2 0 0 SET		o - 200 .		External output 201 ON
CLR OUT 2 0 1 SET		o - 201 .		External output 201 OFF
RES		o - 201		External output 200 OFF
CLR OUT 2 0 0 RES		o - 200		Release of forced output mode
CLR RES RES ENT				

[Explanation]

1. After wiring, external output (Y) can be turned on/off according to forced output unrelated to program. So output wiring can be checked easily.
2. The forced output function can be activated when the programmer is set in the test mode with the basic unit in the stop state.

Item	Key-in procedure	Operation
Forced output ON	CLR OUT Output No. SET	External output is turned on and remains in this status.
Forced output OFF	CLR OUT Output No. RES	Activated external output is turned off.

CAUTION

Operation must be carried out in adequate consideration of safety.

3. An error will occur if the forced output is activated for the external input number.

6

[Caution on Mounting]

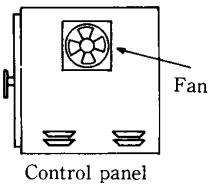
(1) Installation of EM-II series, avoid locations listed in Table 6-1.

Table 6-1 Installation-Prohibited Environment

No.	Environment
1	Location exposed to direct sunlight.
2	Location where ambient temperature exceeds a range of 0 to 55°C (Note 1).
3	Location where relative humidity exceeds a range of 30 to 90%. Location where temperature changes suddenly or condensation occurs (Note 2).
4	Location where atmospheric air contains much corrosive gas or inflammable gas.
5	Location where atmospheric air is laden with excessive dust, salinity or iron powder.
6	Location directly subjected to strong vibration or shock.

Notes: 1. If ambient temperature rises beyond 55 °C, it must be reduced to within 55 °C by using a cooling device such as a fan. If ambient temperature falls below 0 °C, avoid cutting off power supply or else provide a heater or the like in order to keep the temperature above 0 °C.

2. If there is a possibility of condensation, preventive measure such as provision of a heater is required.

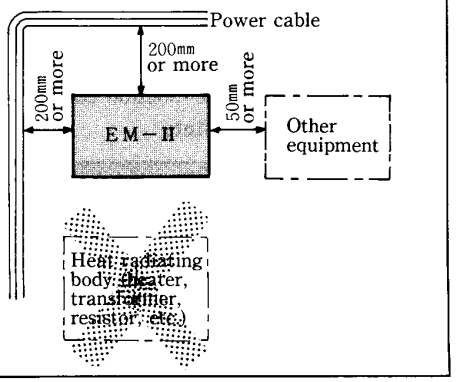


(2) Mounting

The EM-II series must be mounted in the cotrol panel while observing the precautions listed in Table 6-2.

Table 6-2 Precautions on Installation in Panel

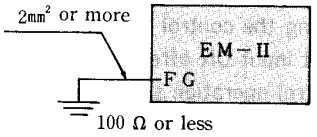
No.	Precautions
1	During installation, pay stric attention not to let fragments due to drilling or wiring fall into the programmable controller. The EM-11 series is provided with a dust-preventive sheet for protection against falling wire fragments. Do not remove the sheet before completion of installation and wiring.
2	Provide a space of 50 mm or more between the EM-II and other equipment or structure for facilitating ventilation.
3	Avoid installation right above equipment which radiates much heat (such as 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.



(3) Wiring

The EM-II must be wired while observing the precautions listed in Table 6-3.

Table 6-3 Precautions on Wiring

No.	Precautions on wiring
1	Connect the grounding terminal (FG terminal) to a cable having a ground resistance of 100 Ω or less which is not used for high power grounding. Restrict the length of grounding cable within 20 m. 
2	Avoid passing the I/O cables through a duct which houses other power cable and bundling these cables together. Do not pass the expansion cable through a duct used for the I/O cables and bundle these cables together.
3	Restrict the length of I/O cabling within 30 m. If cabling byeond 30 m is unavoidable, separation of I/O cables or like measure is required. (Cabling beyond 100 m is unallowable in any case.)

(4) Emergency stop circuit

The EM-II series incorporates an adequate noise suppressing measure so that it withstands a noise level of 1,500 Vp-p or more (when measured by Hitachi method). If larger noise than above enters, misoperation might occur.

The following check functions are prepared for detecting anomaly upon misoperation.

1) Watch dog timer check
2) Undefined instruction check

On detection of anomaly:

1) All outputs turn off.
2) RUN lamp and RUN contacts turn off (a single output point programmed as RUN contacts). However, avoid complete dependence on these anomaly detection functions. **A safety ensuring cirucit such as for emergecy stop must be configured by utilizing an external relay or the like as shown in Fig. 6-4.**

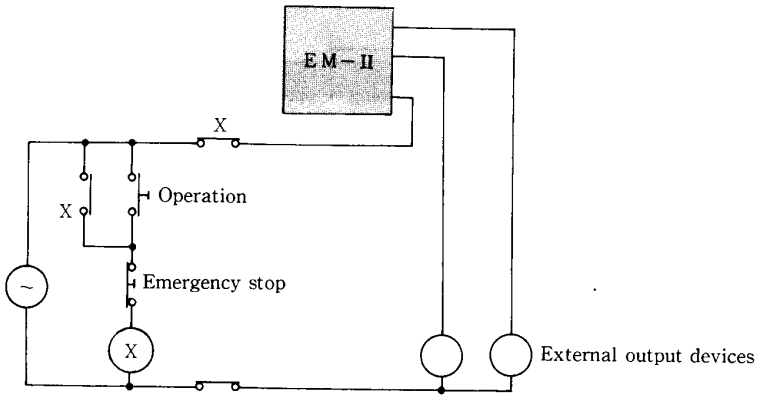


Fig. 6-4 Emergency Stop Circuit

[**Internal Sequence upon Energization**]

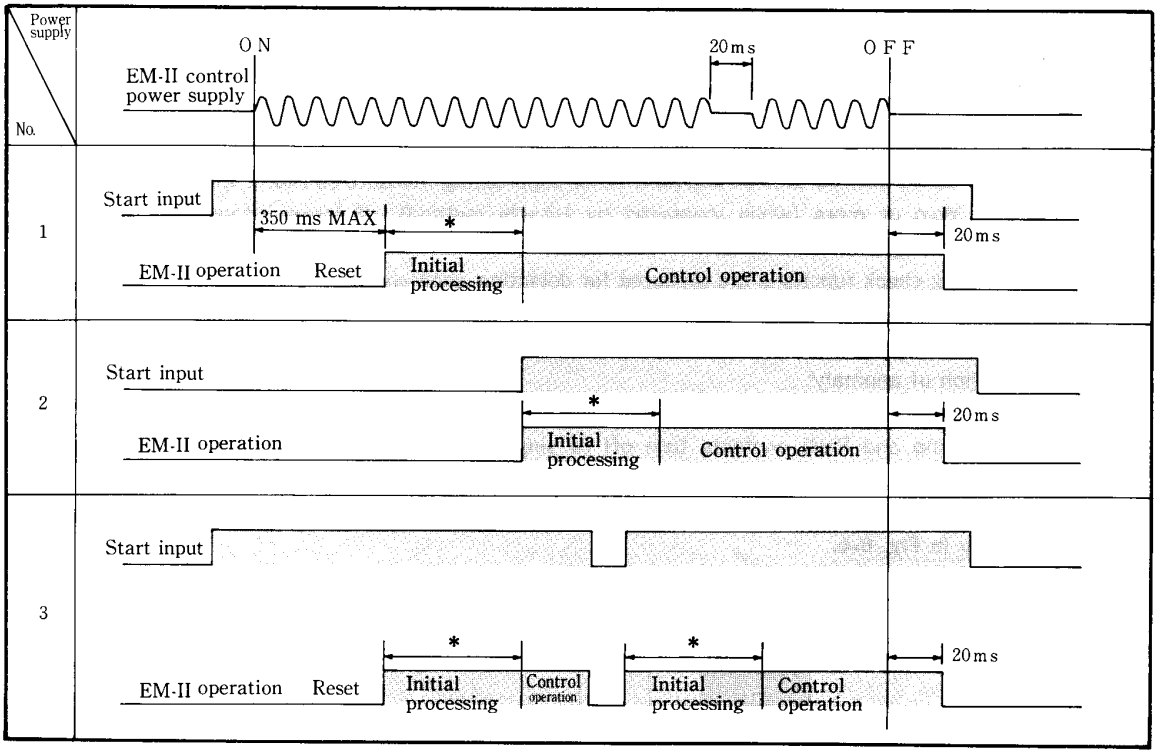
Table 6-5 shows the waveform of power supply to EM-II and its operational status in poperational status in response to start input.

(1) Energization with EM-II start input at ON (C and start input shorted)
Control operation starts as shown in No. 1.
During the control operation, external input is not fetched as a signal even when it turns on.

(2) Start input ON after EM-II energization
Control operation starts as shown in No. 2.

(3) Start input OFF during operation
When start input turns off, operation stops (EM-II is reset) as shown in No. 3.
Upon release of start input OFF, the EM-II performs control operation after initial processing.

Table 6-5 Internal Sequence upon Energization



(4) Operation at momentary power interruption
Control operation continues despite a momentary power interruption if shorter than 20 ms.
The EM-II detects power interruption through a voltage drop in the 5 V DC power supply. Therefore, **operation may continue for 100 ms or more despite power interruption if the 5 V DC charge is retained for a longer time because of a lighter load in a system which consists of the basic unit alone (does not have a programmer).**

(Note*) This time period varies with the length of program. As a standard:
Approx. 4 sec with 925 steps
Approx. 8 sec with 1950 steps

Display in normal status

Status / Display			Lamp indication		
			POW	RUN	ERR
Normal status	Operation	Standard base	☼	☼	●
		Expansion base	☼	—	—
	Stop	Standard base	☼	●	●
		Expansion base	☼	—	—

☼: Lit

● : Extinguished

Other Check Items

(1) Abnormal temperature rise due to heat source or direct sunlight

(2) Entrance of dust, chips or wiring scraps into panel

(3) Loosening of wire and terminal connections

[Explanation]

1. The EM-II series incorporates neither a battery nor a consumable whose life reaches its end in a short period of time. However, attention must be paid to the service life of the output relay in case it is activated frequently.
2. The aluminum electrolytic capacitor used in the power supply unit also has a limited lifetime.
   In this capacitor, a chemical reaction is taking place. And its lifetime changes widely at diffent ambient temperatures. Electrolytic capacitor is generally subordinate to the "Arrhenius's equation (double effect rule with change of 10°C)." This signifies that its lifetime is reduced by half with a temperature rise of 10°C and lengthened to a twofold value with a temperature fall of 10°C.
   For a longer service life, an adequate ventilation and appropriate ambient temperture should be ensured at installation.
3. Never use lacquer thinner or the like for cleaning because such a substance may cause the cover surface to be dissolved or discolored.

Table 7-1 Troubleshooting (1/2)

No.	Phenomenon	Check item	Check result	Remedy
1	POW lamp does not light when turning on power supply.	Check line voltage.	Abnormal	Correct to normal line voltage.
			Normal	Exchange the product.
2	Operation does not start though operation control input turns on.	Check programmer switch.	Set at PROG.	Set to TEST or RUN.
		Conduct syntax check by keying in (CLR SRC)	Error detected	Correct program.
			Error not detected	Exchange the product.
3	During operation, RUN lamp went off and operation stopped. (Or RUN lamp went off shortly after start of operation.)	Check if ERR lamp is lit.	Lit	Eliminate noise source and recheck program. Then restart operation. (If error recurs even after eliminating noise source, the product must be exchanged with a new one.)
		Check if shorter program can be run.	Can be run.	Shorten scan time somehow because it is longer than 100 ms.
			Cannot be run.	Exchange the product.
4	Input lamp stays OFF.	Connect the relevant input terminal and 24 V terminal to check if the lamp lights up.	Lights up.	Correct external wiring or exchange external input device.
			Does not light up.	Utilize unassigned input terminal or exchange the product.
5	Input lamp won't go off.	Input the relevant input terminal and check if the lamp goes off.	Goes off.	Correct external wiring or exchange external input device.
			Does not go off.	Utilize unassigned input terminal or exchange the product.

7

Table 7-1 Troubleshooting (2/2)

No.	Phenomenon	Check item	Check result	Remedy
6	Output lamp will not come on or go off.	Monitor the relevant output with programmer and confirm that the lamp status matches the monitored contents.	Matches	Correct program.
			Does not match.	Utilize unassigned output terminal or exchange the product.
7	Output lamp does not meet load ON/OFF status.	Check for conductivity across relevant output terminal and C terminal (with the aid of tester).	Output lamp matches conductive status.	Correct external wiring or exchange external output device.
			Output lamp does not match conductive status.	Utilize unassigned output terminal or exchange the product. (If the contacts of internal relay are fused because of excessively large load current, an intermediate relay is required.)

[Explanation]

1. If a trouble occurs on the system under normal operation, we must judge first as to whether the trouble is attributable to the EM-II series or other section.
   Check and take a measure as per the table above.

Table 7-2 Syntax Error Codes

Syntax error code (decimal)	Error display on PGMJ	Error display on PGMJ-R2	Description
0	Blank	Blank	No error
1	E	E	Combination of instruction words does not comply with the syntax rule.
2	E	E	The structure of main routine or interrupt processing routine is abnormal.
3	E	E	The argument of INT instruction having the relevant number is undefined.
4	E	E	The FUN06-FUN07 structure is abnormal.
5	E	E	The FUN08-FUN09 structure is abnormal.
6	⌊	u E	STR level is under that specified for instruction word.
7	⌐	o E	STR level is over that specified for instruction word.
8	⌊	u E	The level of master control is under that specified for instruction word.
9	⌐	o E	The level of master control is over that specified for instruction word.
10	E	E	IF or IFR is duplicated. An impermissible instruction (OUT T/C) is written after IF or IFR.
11	E	E	The I/0 number, constant or other element of instruction word is not within the specified range.
12	E	E	This double coil is impermissible.
13	E	d E	Occurrence of double coil though operation is allowed. (Alarm is issued.)
14	E	E	There are multiple SB instructions. CALL and SB do not correspond to each other.
15	E	E	Both JMP and INT instructions are used in the same step.
20	F	f E	Program cannot be interpreted because an undefined operation code or operand is used. Or the user memory area is not formatted correctly.
30	E	E	User program is judged to be abnormal according to the result of sum check.

[Explanation]

1. Syntax check of program is carried out just before operation or by keying in [CLR] [SRC]
2. If error is detected in syntax check, syntax error code can be checked by keying in [CLR] [9] [8] [0] [MON] [MON] since the contents of error are coded in the special internal output WM980.

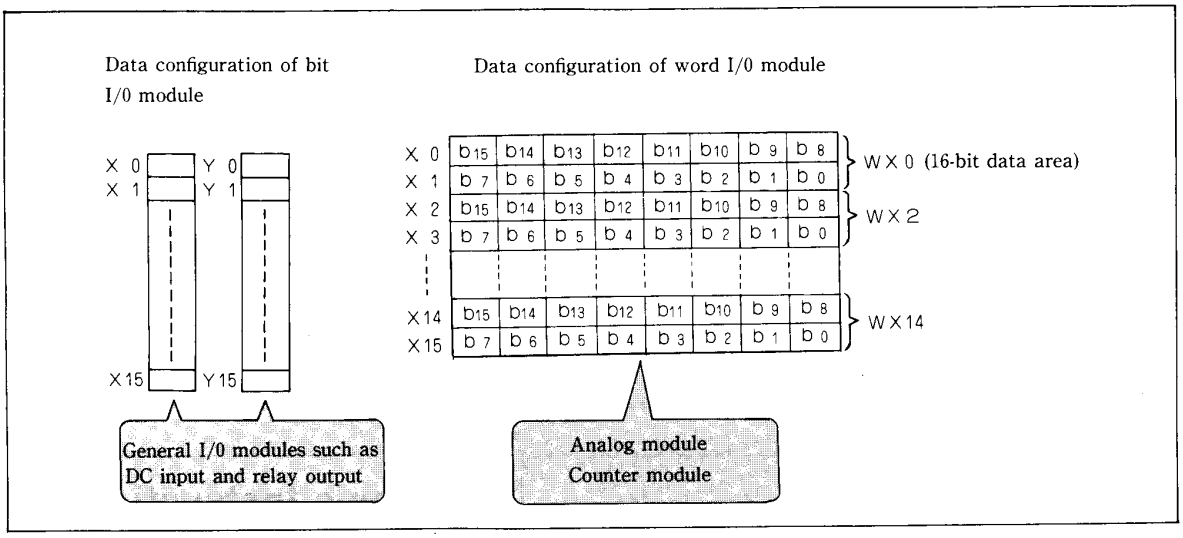
7

Table 7-3 System ERROR Codes

Syntax error code (decimal)	Description
10	Trap interruption has occurred.
11	Stack pointer abnorality is detected.
12	Contradiction to logic is detected.
13	Improbable interruption has occurred.
14	NMI interruption has occurred.
20	Data has not been written successfully in the user program memory.
21	Sum-check error is detected in system ROM.
30	Undefined PCS instruction word is fetched.
31	PCS stack pointer abnormality is detected.
32	Sum-check error has occurred in user program during operation.
40	Received signal has overflowed the buffer.

[Explanation]

1. If the ERR lamp comes on, system error code can be checked by keying in [CLR] [9] [7] [0] [MON] [MON] after turning on power supply again since the contents of system error are coded in the special internal output WM970.



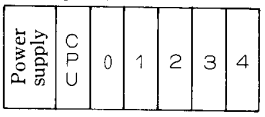
[Explanation]

1. General I/O modules such as DC input module and relay output module have only 1-bit data in each I/O number.
   Therefore, these modules are called "bit I/O modules."
   The analog module and counter module have 8-bit data area in each I/O number. Therefore, these modules are called "word I/O module."
2. Word input and output are represented by WX and WX.
   WX and WX handle two numbers (X0 and X1 in the above example) as one word (WX0).
3. For WX/WY calculation, the aforementioned arithmetic instructions are used. These instructions are usable when WM is specified for component.

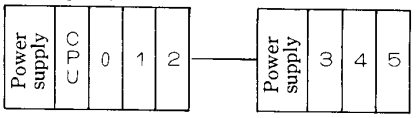
Arithmetic Instructions Applicable to WX and WY			
FUN10(LOADW),	FUN60(BLOAD),	FUN21(OUTW),	FUN71(BOUT),
FUN11(ADD),	FUN61(ADBNR),	FUN12(SUB),	FUN62(SUBNR),
FUN13(MUL),	FUN63(MUBNR),	FUN14(DIV),	FUN64(DIBNR),
FUN15(AND),	FUN16(OR),	FUN66(EXOR),	FUN17(CPEH),
FUN18(CPE),	FUN19(CPL),		

[Mounting of analog module] ···Mountable in a desired slot other than for power supply and CPU.

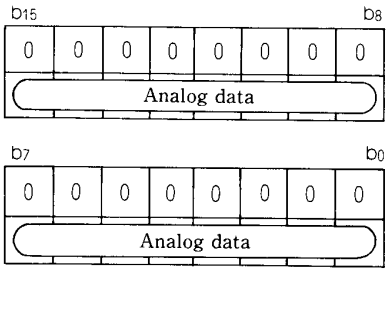
(Example 1)



(Example 2)



[Data configuration] ···Assigned to the lower 8 bits (b_0 to b_7) of word data with an 8-bit resolution.



Bits b_8 through b_{15} are all "0".

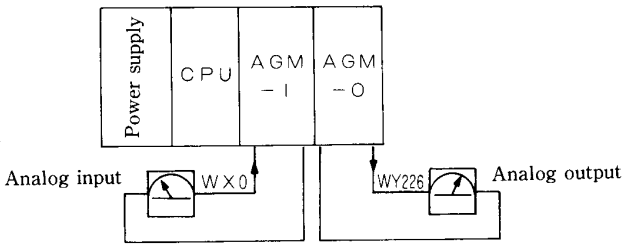
[Assignment table of analog inputs/outputs] ···All inputs and outputs of possible channels have an even number.

Slot	Channel	Analog input	Analog output	
		AGM-I, IV	AGM-O, OV	AGM-OD, ODV
0	0	WX 0	WY 200	WY 200
	1	WX 2	WY 202	WY 202
	2	WX 4	WY 204	—
	3	WX 6	WY 206	—
	4	WX 8	—	—
	5	WX 10	—	—
	6	WX 12	—	—
	7	WX 14	—	—
1		WX 20, WX 22, WX 24, WX 26, WX 28, WX 30, WX 32, WX 34	WY220, WY222, WY224, WY226	WY220, WY222
2		WX 40, WX 42, WX 44, WX 46, WX 48, WX 50, WX 52, WX 54	WY240, WY242, WY244, WY246	WY240, WY242
3		WX 60, WX 62, WX 64, WX 66, WX 68, WX 70, WX 72, WX 74	WY260, WY262, WY264, WY266	WY260, WY262
4		WX 80, WX 82, WX 84, WX 86, WX 88, WX 90, WX 92, WX 94	WY280, WY282, WY284, WY286	WY280, WY282
5		WX100, WX102, WX104, WX106, WX108, WX110, WX112, WX114	WY300, WY302, WY304, WY306	WY300, WY302
6		WX120, WX122, WX124, WX126, WX128, WX130, WX132, WX134	WY320, WY322, WY324, WY326	WY320, WY322
7		WX140, WX142, WX144, WX146, WX148, WX150, WX152, WX154	WY340, WY342, WY344, WY346	WY340, WY342
8		WX160, WX162, WX164, WX166, WX168, WX170, WX172, WX174	WY360, WY362, WY364, WY366	WY360, WY362
9		WX180, WX182, WX184, WX186, WX188, WX190, WX192, WX194	WY380, WY382, WY384, WY386	WY380, WY382

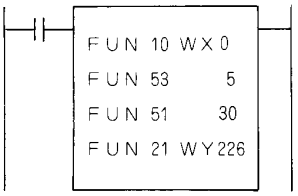
[Arithmetic expression]

$Y = X * A + B$

- X: Analog input (assigned to WXO)
- Y: Analog output (assigned to WY226)
- A: 5 (decimal constant)
- B: 30 (decimal constant)



[Programming]



Analog input WX0 (BIN) →AR(BIN)
AR * 5H (hexadecimal constant) →AR(BIN)
AR + 1EH (hexadecimal constant) →AR(BIN)
AR(BIN) →Analog output WY226 (output to 4th channel)

[Explanation]

1. The analog module has a resolution of 8 bits. It comes in two types: current and voltage types. The relationship between analog data and digital data is as shown below. This module handles only BIN data.

Type	A→D conversion (input)	D→A conversion (output)
Current type	4 to 20 mA → 00 to FF	00 to FF→ 4 to 20 mA
Voltage type	0 to 10 V → 00 to FF	00 to FF →0 to 10 V

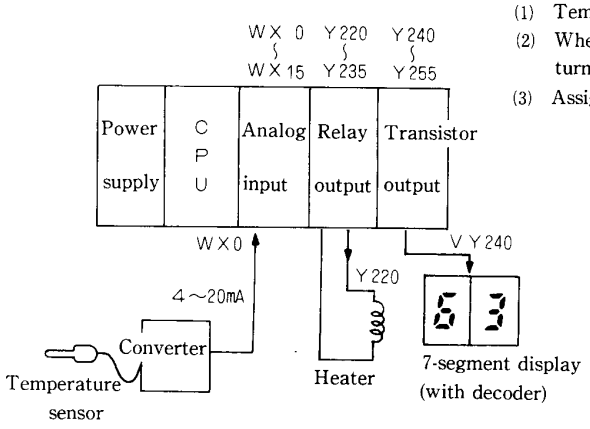
   - ○ Convert into BCD data by the FUN24 (BCD) instruction when necessary.
   - ○ For data output to the analog output module after BCD operatio, data must be converted into BIN format by the FUN25 (BNR) instruction.

2. The specifications of the analog modules are listed in the table below.

Item \ Model	Current type			Voltage type		
	AGM-I	AGM-O	AGM-OD	AGM-IV	AGM-OV	AGM-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 method	Photocoupler			Photocoupler		
Inolation between channels	Not provided			Not provided		

〈Application example of analog module〉

[Explanation of operation]



(1) Temperature is measured and indicated on the 7-segment display.
(2) When the measured temperature is lower than 65℃, the heater turns on, and it turns off when equal to or higher than 65℃.
(3) Assignment

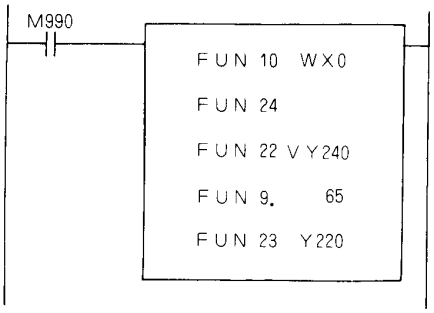
Temperature data……WX0

Heater output……Y220

Display……Y240—Y247

(2 digits)

[Programming]



Analog input WX 0(BIN)→AR(BIN)

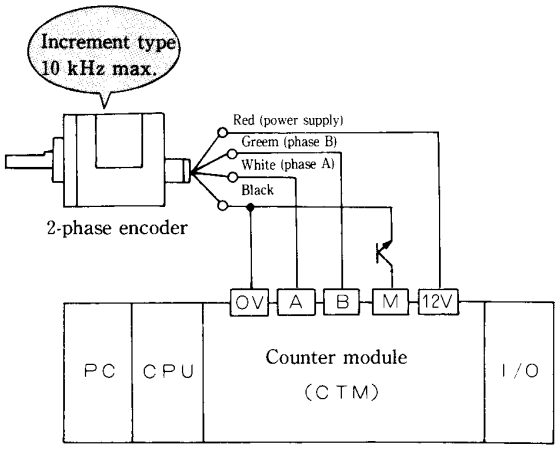
BIN⇨BCD conversion →AR(BCD)

AR→Y240—Y247·· Temperature indication on display

AR<65···1→C
C→Y220
} Heater turns on when temperature is lower than 65℃.

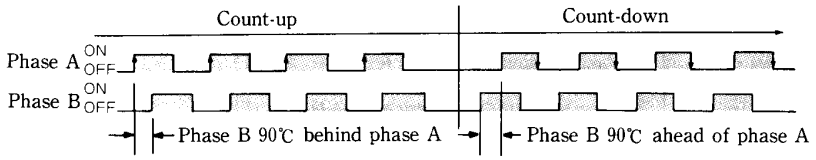
In this example 0°c at 4mA and 255°c at 20mA.

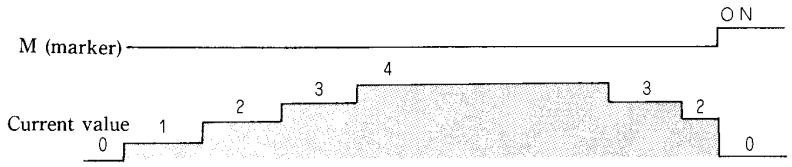
8



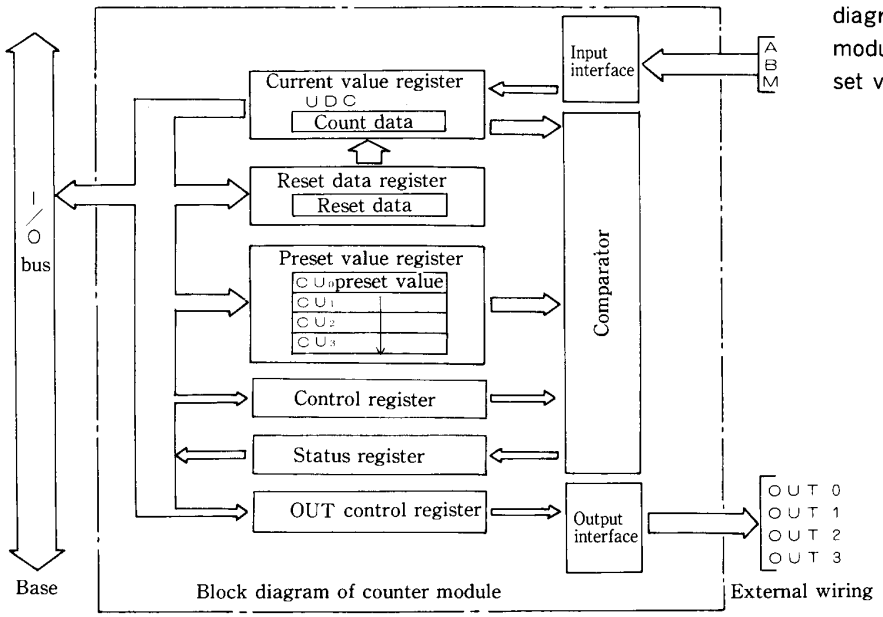
[Explanation of operation]

1. 2-phase pulse signal 10kHz max. from the encoder is fetched for comparison of the preset value and current value.
2. Count is incremented or decremented depending on the phase difference between phase A and B pulses.
3. On changeover from count-up to count-down and vice versa, current value is controlled so that it does not change for one pulse.
4. When M (marker) turns on, current value is reset to "0000."

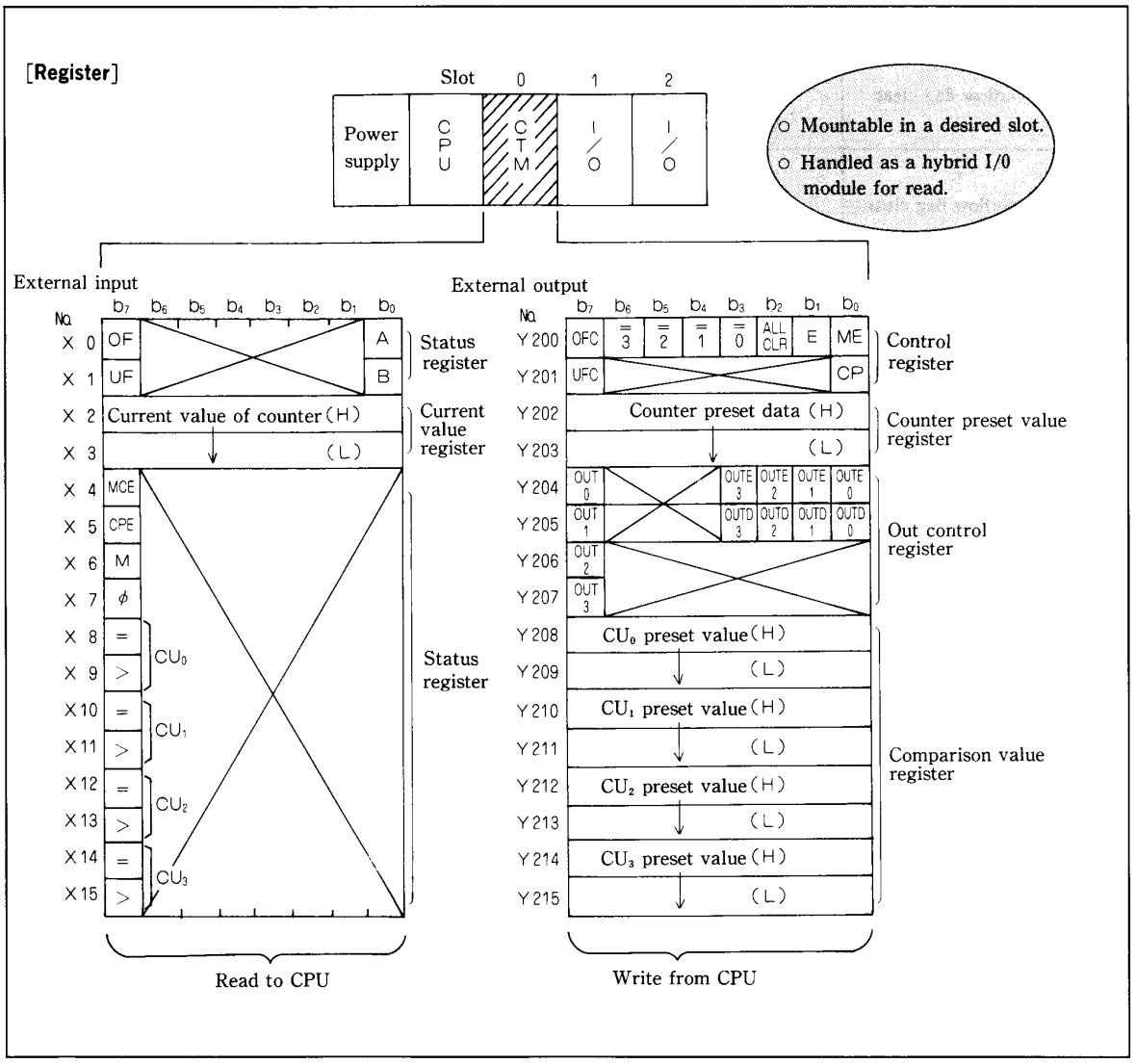




5. Shown at left is a block diagram of the counter module. Up to 4 preset values are settable.



〈Register function of counter module〉

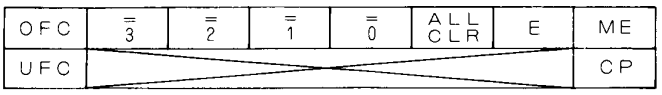


[Explanation]

The function of each register is explained below.

1. Control register

This register specifies operation of the counter module. It is set from the CPU.



Control register

8

Control register Functions of Control Register

No.	Flag name	Symbol		Function
1	Overflow flag clear	OFC	0	Does not clear the overflow flag (OF).
			1	Clears overflow flag (OF). (While at "1", 1 → OF will not occur despite overflow.)
2	Underflow flag clear	UFC	0	Does not clear underflow flag (UF).
			1	Clears underflow flag (UF). (While at "1", 1→ UF will not occur despite underflow.)
3	Marker enable	ME	0	Invalidates marker terminal input.
			1	Validates marker terminal input. Marker terminal ON: Resets current value to 0.
4	Counter preset	CP	1	Preset data → current value Note 2 (While at "1", counting stops.)
5	= flag clear	= = = = 0 — 3	0	Does not release retention of = flag.
			1	Releases retention of = flag.
6	All clear	ALL CLR	1	Clearrs current value of counter and releases retention of all flags. Note 1
7	Counter enable	E	1	Enables counter read and write from CPU. Note 1

Notes: 1. On initialization, be sure to make the following setting.

"1" (continuous) → E

"1" (pulse) → ALL CLR

2. "1" of counter reset CP requires 1 scan pulse.

2. Counter preset value register

This register preset a current value to a predetermined value. When the counter preset flag [CP] of the control register is set to "1," the preset value is fetched as a current value.

3. Comparison value register

Four values CU_0 to CU_3 can be written. Current value and comparison value are compared and the result is output to the status register and output terminals (OUT0 to OUT3).

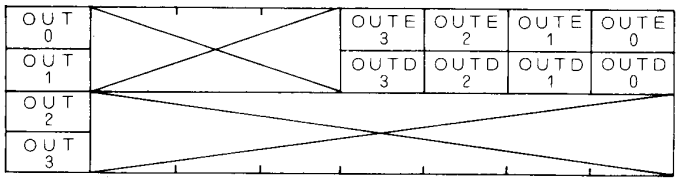
Comparison value register

CU_0	5 (4th digit)	4 (3rd digit)	(H)
	3 (2nd digit)	2 (1st digit)	(L)
CU_3	7 (4th digit)	8 (3rd digit)	(H)
	9 (2nd digit)	0 (1st digit)	(L)

In this example, the CU_0 and CU_3 comparison values are 5432 and 7890, respectively.

4. OUT control register

Used to select external output when the result of comparison is "=" or ">." Also available is a function for forcibly turning on the output terminals OUT_0 through OUT_3 regardless of the result of comparison between the current value and comparison value.



OUT control register

Functions of OUT Control Register

No.	Flag name	Symbol	Function
1	OUT control flag	OUT_0 | OUT_3	0 The contents of ">" flag are transmitted to output terminals OUT_0 through OUT_3. 1 The contents of "=" flag are transmitted to output terminals OUT_0 through OUT_3. Each of OUT_0 through OUT_3 is independently controllable.
2	Forced output	$OUTE_0$ | $OUTE_3$	When setting "1" to $OUTD_0$ through $OUTD_3$ with the output enable flags ($OUTE_0$ through $OUTE_3$) at "1", the output terminals OUT_0 through OUT_3 are forcibly turned on. (This occurs indifferently to the result of comparison between current value and comparison value.) Each of OUT_0 through OUT_3 is independently controllable.
3	Forced output flag	$OUTD_0$ | $OUTD_3$	

5. Status register

Indicates the operational status of counter module: the result of comparison between comparison value and current value, and overflow/underflow of current value.

Functions of Status Register

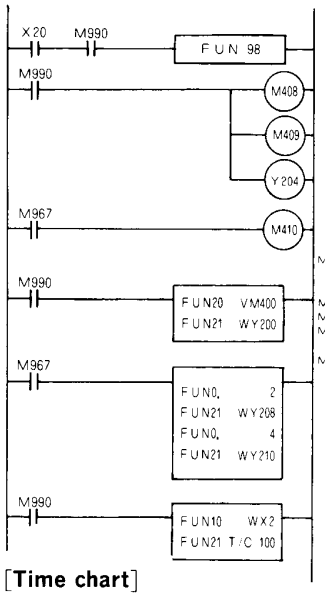
No.	Flag name	Symbol	Function
1	Overflow flag	OF	Turns to "1" when current value returns from 9999 to 0. This flag is not cleared until overflow flag clear (OFC) turns to "1".
2	Underflow flag	UF	Turns to "1" when current value returns from 0 to 9999. This flag is not cleared until underflow flag clear (UFC) turns to "1".
3	Marker enable set end flag	MCE	Ends the "1" status of marker enable (ME) of control register.
4	Reset end flag	CPE	Indicates completion of substituting preset data to current value of counter. Ends the "1" status of CP.
5	Input signal flag	A, B, M, O	Indicates ON/YFF status of input signals A, B, M and O from each terminal.
6	= flag	= (CU_0—CU_3)	Set or turned to "1" when current value equals comparison value. CU_0 to CU_3 = flags are set in response to comparison values CU_0 through CU_3. These flags are not cleared (held) until the = flag clear of control register turns to "1".
7	> flag	> (CU_0—CU_3)	Set or turned to "1" when current value exceeds comparison value. CU_0 to CU_3 > flags are set in response to comparison values CU_0 through CU_3.

6. Current value register

The current value of counter is output in 4 digits.

‹Setting of registers›

[Programming]



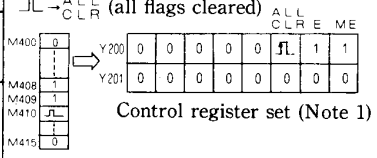
Start of operation

"1"→ME (marker enable)

"1"→E (counter enable)

"1"→OUT 0 (OUT 0 issued when current value equals comparison value)

→ALL CLR (all flags cleared)



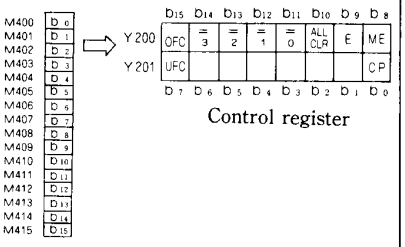
Control register set (Note 1)

Constant "2" → Sets comparison value CU_0.

Constant "4" → Sets comparison value CU_1.

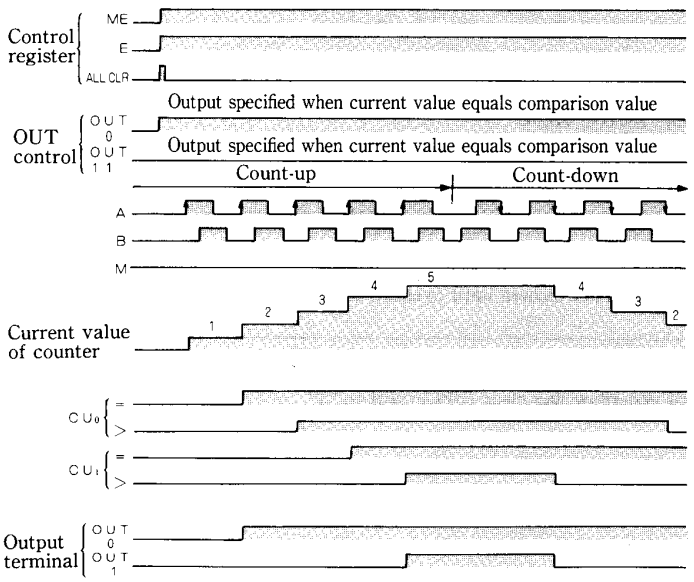
Current value of counter → current value of T/C 0 (Note 2)

Note 1: Control register is set by setting its desired status in the bit table of internal outputs M400 through M415 and outputting the bit table to the control register.



Control register

[Time chart]



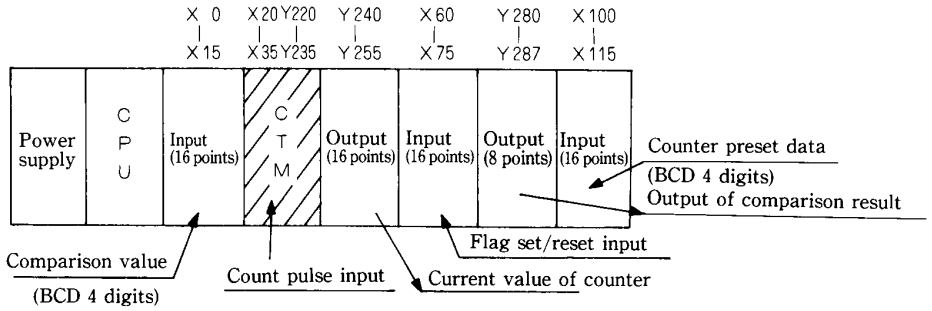
Note 2: When the current value of counter is output to that of soft counter (T/C100), it can be monitored by monitoring T/C0 with programmer.

[Explanation of operation]

1. Suppose that comparison values of counter are 2 (CU_0) and 4 (CU_1). When current value reaches count 2, the output terminal OUT0 turns on, and OUT1 turns on when current value exceeds count 4.
2. The "=" flag turns to "1" when current value equals the comparison value. This status is held until flag clear is specified.

‹Programming of counter module›

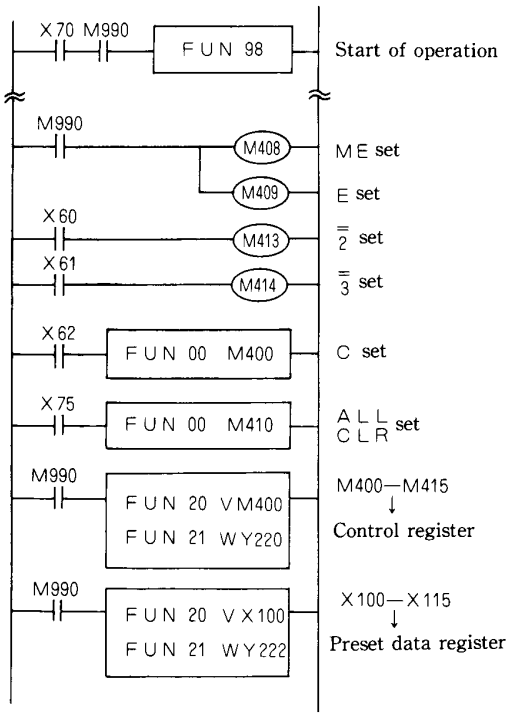
[Configuration]

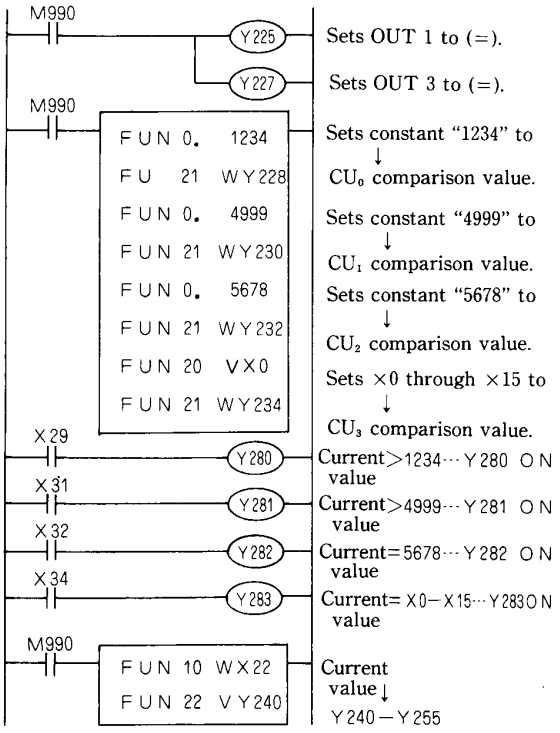


[Operation]

1. Operation starts when ×70 turns on.
2. When current value is larger than 1234, Y280 and OUT0 in CTM turn on.
3. When current value is larger than 4999, Y281 turns on. When current value equals 4999, OUT1 in CTM turns on.
4. When current value equals 5678, Y282 turns on (it is reset when ×60 turns on). When current value is larger than 5678, OUT2 in CTM turns on.
5. When current value equals ×0 through ×15 (BCD 4 digits), Y283 and OUT3 in CTM turn on (it is reset when ×61 turns on).
6. Current value is sent to outputs Y240 through Y255.
7. When ×62 turns from OFF to ON, preset data (×100 through ×115) is substituted for current value.
8. When ×75 turns on, Y282/Y283 and OUT1/OUT3 in CTM are turned off (= flag reset).

[Programming]

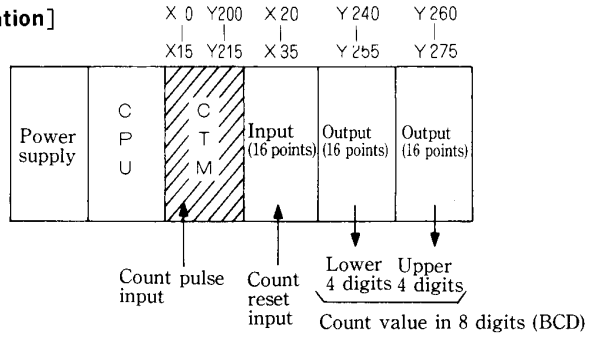




8

〈Application example of counter module〉

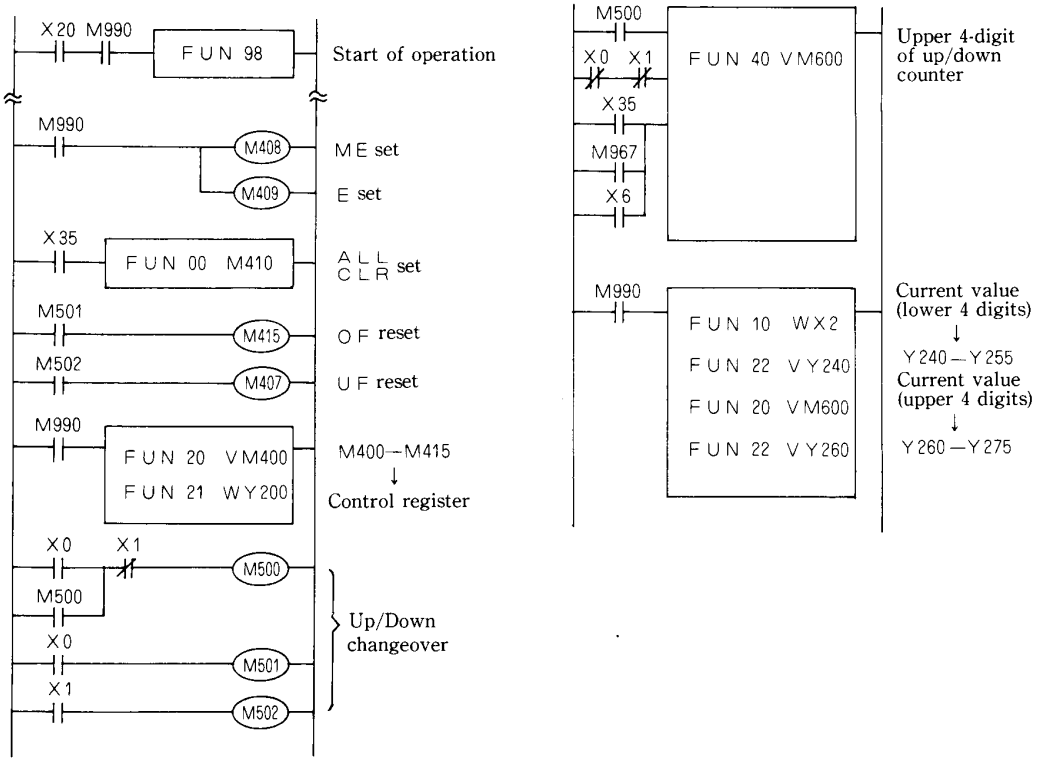
[Configuration]



[Operation] ···8-digit up/down counter

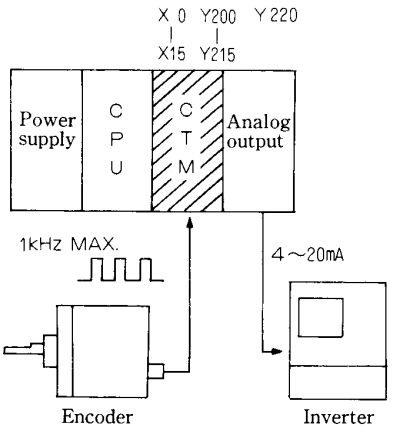
1. When ×20 turns on, operation starts.
2. When ×35 turns on, current value is reset.
3. 8-digit value is output to Y240 through Y255 (lower 4 digits) and to Y260 through Y275 (upper 4 digits).

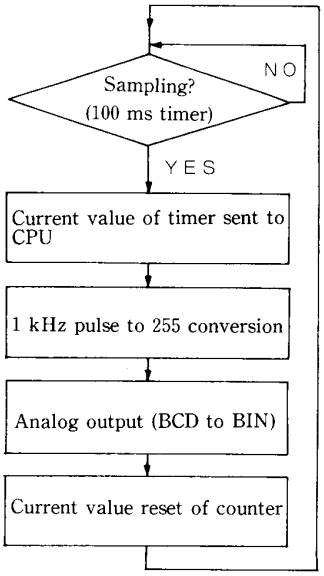
[Programming]



〈Application example of counter module〉

[Explanation of operation]





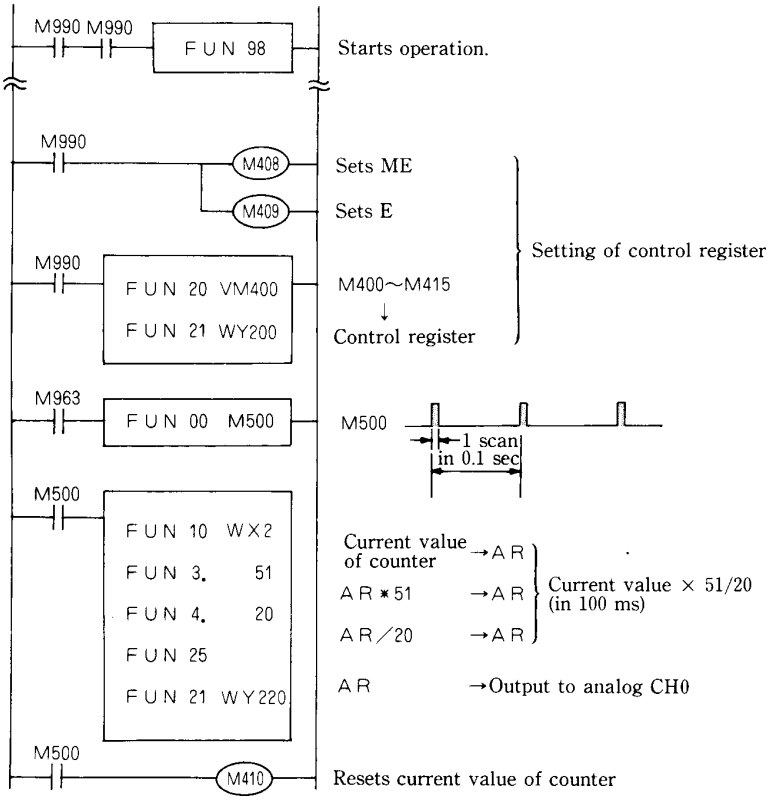
(1) The counter module counts pulse (1 kHz max.) from the encoder for 100 ms.

(2) Conversion is made so that 1 kHz pulse (100 pulses/100 ms) is converted to 255.
$100 \times 51/20 = 255$
↑ Conversion factor

(3) The analog module specifies speed to the inverter.

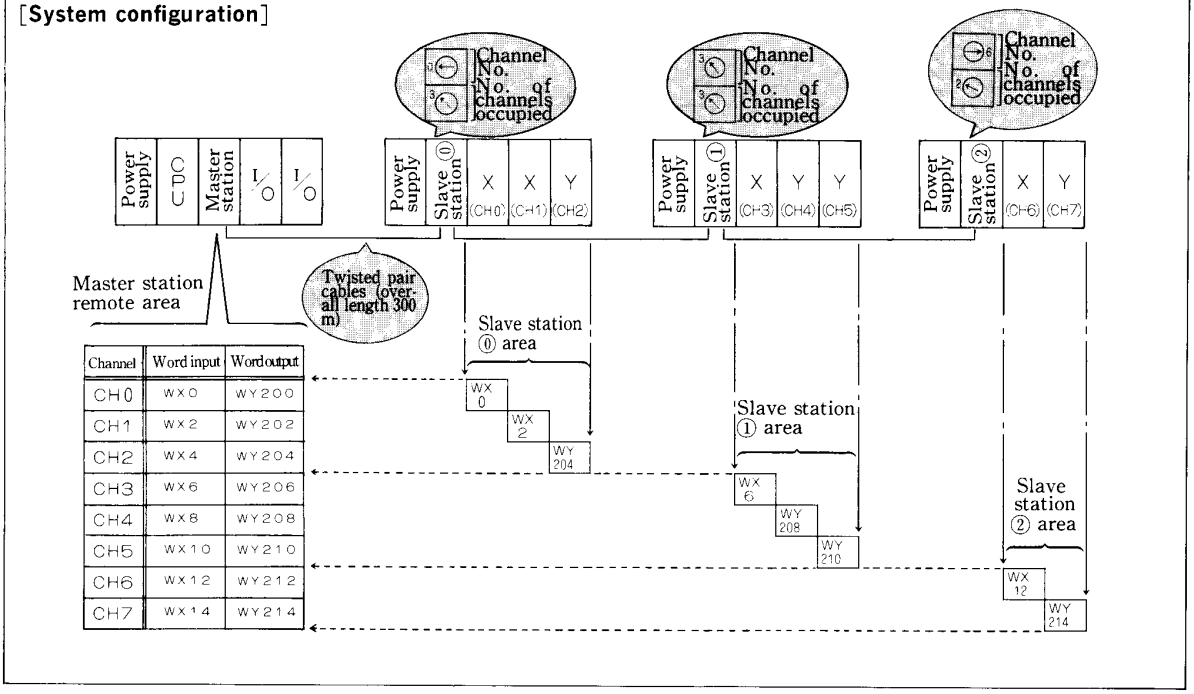
Inverter speed is now controllable in response to the rotational speed of encoder.

[Programming]



8

[System configuration]

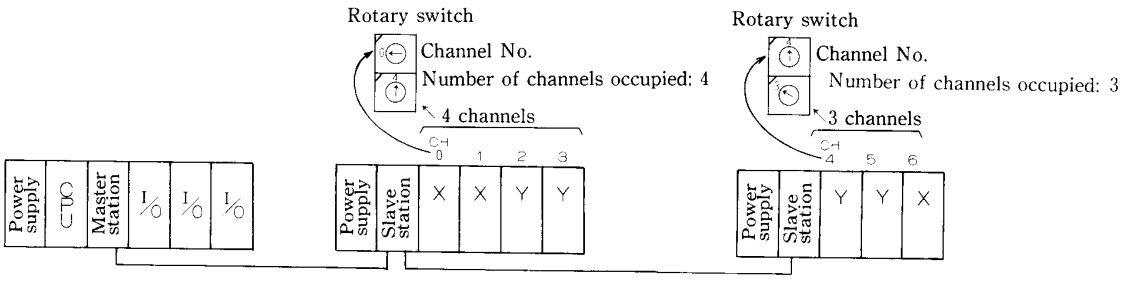


Channel	Word input	Word output
CH0	WX0	WY200
CH1	WX2	WY202
CH2	WX4	WY204
CH3	WX6	WY206
CH4	WX8	WY208
CH5	WX10	WY210
CH6	WX12	WY212
CH7	WX14	WY214

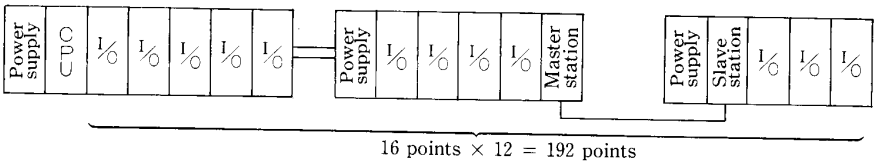
[Explanation]

1. A remote I/O system can be composed by connecting the remote master station [RITOM-TM] and slave station [RIOM-TL] with twisted pair cables. In case operation switches and indicator lamps are installed separately from the main control panel, I/O information is transmissible via the twisted pair cables. This is helpful for reducing the wiring work.
2. The master station has a remote I/O area of 8 channels (CH0 through CH7). The I/O modules after the heading slave station correspond to the channel numbers 0 to 7 of master station area in this order.
3. Each channel of master station area consists of a 16-bit word input (WX) and a 16-bit word output (WY). When the I/O module at the remote end is in the input status, the word input (WX) area is occupied, and the word output (WY) area is occupied when the I/O module is in the output status.
   Therefore, a single master station can transmit a total of 8 I/O words (128 points with 16 I/O modules) to the remote end.
4. Irrespective of whether any I/O module of slave station is of 8 points or 16 points, it occupies a single channel of master station area.
5. Up to 8 slave stations are connectable to each master station.

6. The slave station is provided with rotary switches, by which channel numbers and the number of channels to be occupied by that station are set. Slave stations are numbered 0, 1, 2····· starting from the station nearest the master station.



7. There are two kinds of transmission cables, which allow overall extensions of 300 m and 150 m, respectively.
8. The EM-II series has a nominal I/O capacity of 160 points. However, number of I/O points can be increased by using the remote I/O module.
Shown below is an example of 192 I/O points.



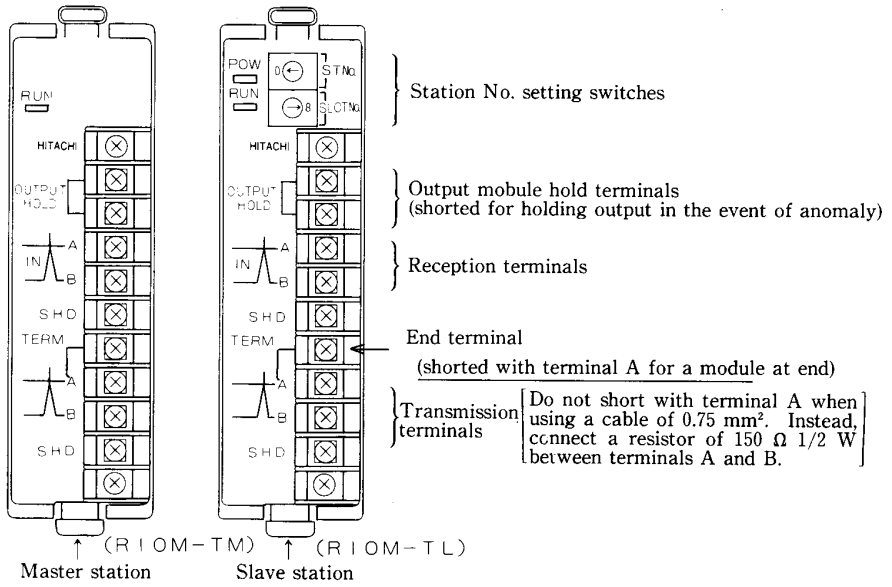
9. Only bit I/O modules are mountable on the slave station side. Word I/O module (analog or counter module) cannot be mounted.
10. The table below lists the specifications of remote master and slave stations.

	Item		Master station [RIOM-TM]	Slave station [RIOM-TL]
General specifications	Operating temperature		0~55°C	
	Storage temperature		−20~70°C	
	Operating humidity		30~90% RH (non-condensing)	
	Current consumption	CH1 (5V)	130 mA	150mA
		CH2 (24V)	20mA	20mA
		CH3 (24V)	5mA	5mA
	Dimensions (mm)		35W × 150H × 117D	
	Weight (kg)		0.2	
Functional specifications	No. of connectable units		Up to 8 slave stations/master station	
	No. of remote terminals		128 points in total of inputs and outputs (8-word)	
	Transmission speed		768kbps	
	Refresh time		Approx. 5 ms	
	Error check		Reverse double transmission	

Transmission line	Recommended cable			Cable length		Terminating resistor
	Type	Maker	Outside diameter	Between stations	Overall extension	
	CO-SPEV-SB-1P −0.3mm²	Hitachi Cable	About ϕ5.5	150mMAX	150mMAX	Incorporated in module (100 Ω)
	CO-EV-SX-1P −0.75mm²		About ϕ16	300mMAX	300mMAX	Requires external connection (150 Ω)

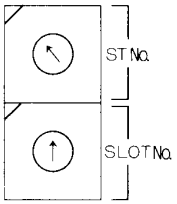
8

[Name of each portion]



[Explanation]

1. Method of setting slave station number (slave station)



ST No·····Sets the number of the remote area channel of master station to which the heading I/O module of the relevant slave station is assigned.

SLOT NO. ·····Sets the number of I/O modules used in the relevant slave station.

2. OUT. HOLD terminals

(1) Master station (RIOM -TM)

These terminals are used to determine the data to be transmitted from the remote-module master station to the CPU module if the remote control system becomes abnormal.

○OUT HOLD terminals shorted·····Holds remote data before occurrence of anomaly.

○OUT HOLD terminals open·····Clears all remote data (reset all to "0").

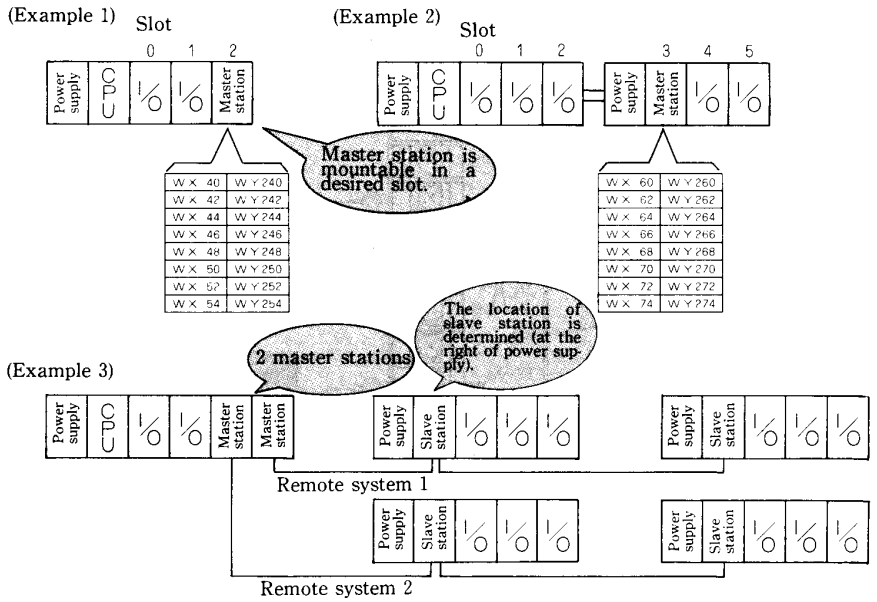
(2) Slave station (RIOM-TL)

These terminals are used to determine the data to be sent from the output module on the slave station side if the remote control system becomes anomalous.

○OUT HOLD terminals shorted·····Holds output data before occurrence of anomaly.

○OUT HOLD terminals open·····Clears all output data to "0."

[Mounting location]



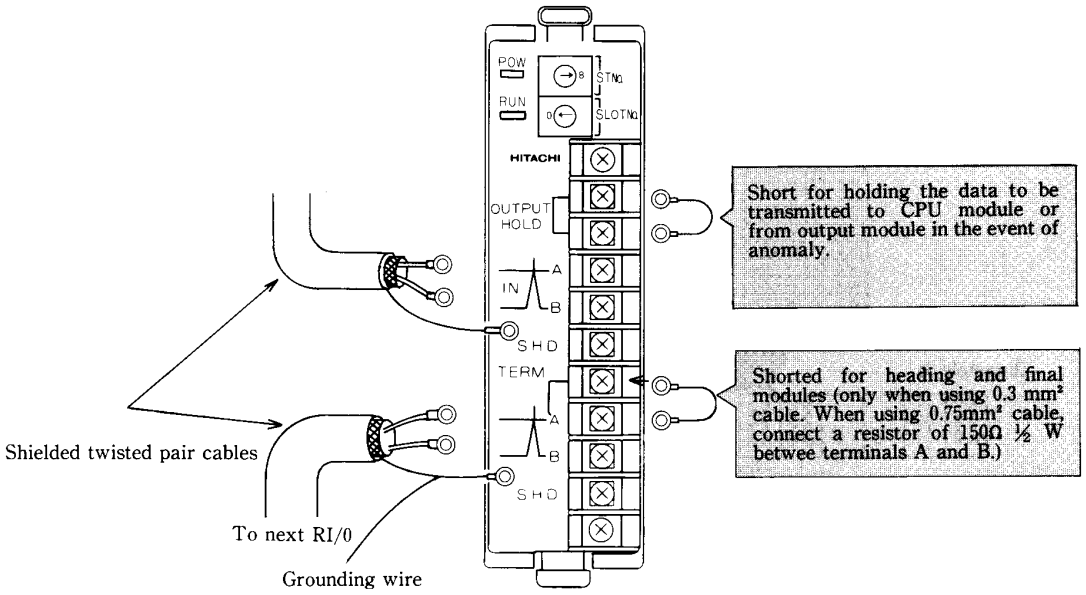
[Explanation]

1. Master station is mountable in any slot other than for CPU and power supply. The remote area address of master station is determined according to the mounting slot position as in other I/O modules. Master station has a 1-slot width.
2. Two or more master stations can be mounted to compose multiple remote systems as shown in the example 3 above. However, remote system must be composed so as to be within the current capacity of power supply module.
3. Mounting location of slave station is fixed. Be sure to install a slave station at the right of a power supply module.
   Slave station also has a 1-slot width.

[Caution]

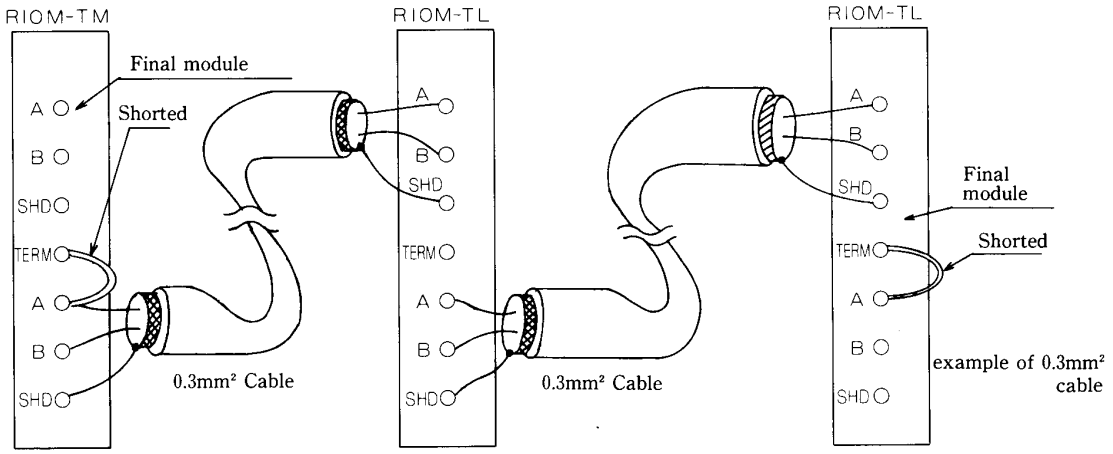
Only bit I/O modules are mountable on the slave station side. Word I/O modules (analog and counter modules) cannot be mounted.

[Wiring method]



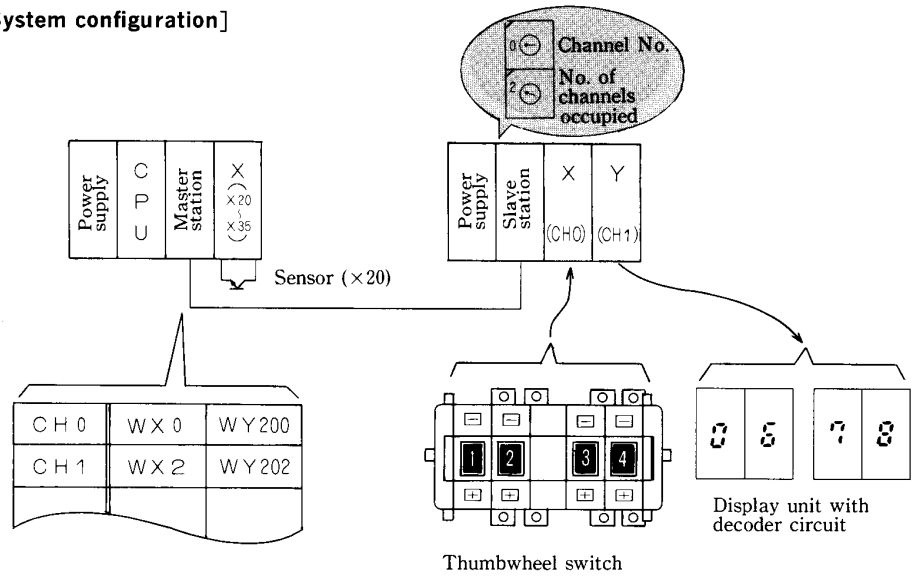
[Explanation]

1. Wiring is exemplified below.



〈Application example (1) of remote I/O〉

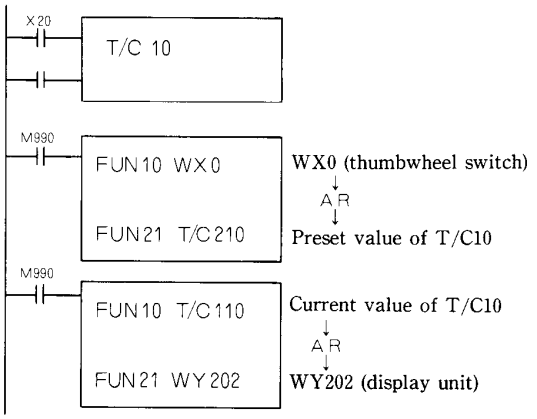
[System configuration]



[Explanation of operation]

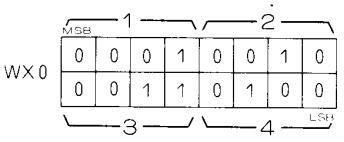
1. The ON/OFF operations of sensor ×20 are counted by the counter T/C10.
2. Preset value of the counter T/C10 is determined by the thumbwheel switch on the slave station side.
3. The current value of counter T/C10 is indicated on the display unit on the slave station side.

[Programming]



[Explanation]

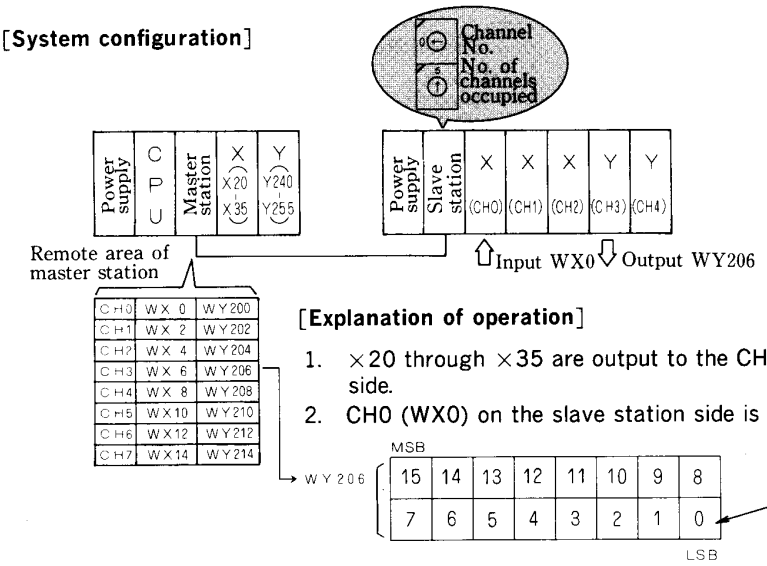
1. When the thumbwheel switch is set to "1234," WX0 of the master station becomes as follows.



8

〈**Application example (2) of remote I/O**〉

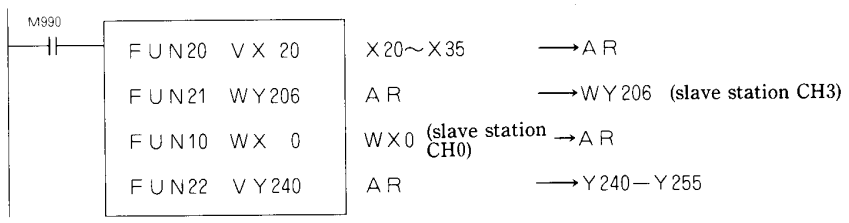
[System configuration]



[Explanation of operation]

1. ×20 through ×35 are output to the CH3 (WY206) on the slave station side.
2. CH0 (WX0) on the slave station side is output to Y240 through Y255.

[Programming]

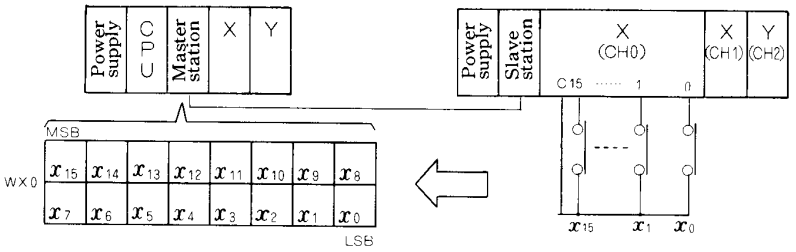


[Explanation]

1. Remote master station is programmed as a hybrid word I/O module (input 8 words, output 8 words). When loading (by FUN 10) the word input (WX) of master station, the ON/OFF status of external input on the slave station side is fetched into AR in steps of word (16 bits).
   When outputting (by FUN21) to the word output (WY) of master station, output is available from the external output of the slave station side in steps of word (16 bits).
2. In the above sequence, the processing below are made.
   (1) External inputs X20 through X35 $\xrightarrow{\text{FUN20}}$ AR $\xrightarrow{\text{FUN21}}$ Output to slave station CH3 (WY206)
   (2) Slave station CH0 (WX0) $\xrightarrow{\text{FUN10}}$ AR $\xrightarrow{\text{FUN22}}$ Output to external output Y240 through Y255

〈Application example (3) of remote I/O〉

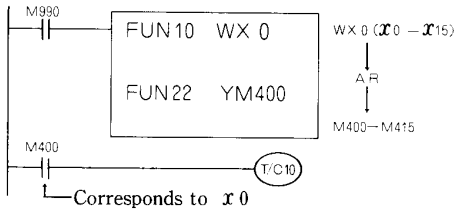
[System configuration]



[Explanation of operation]

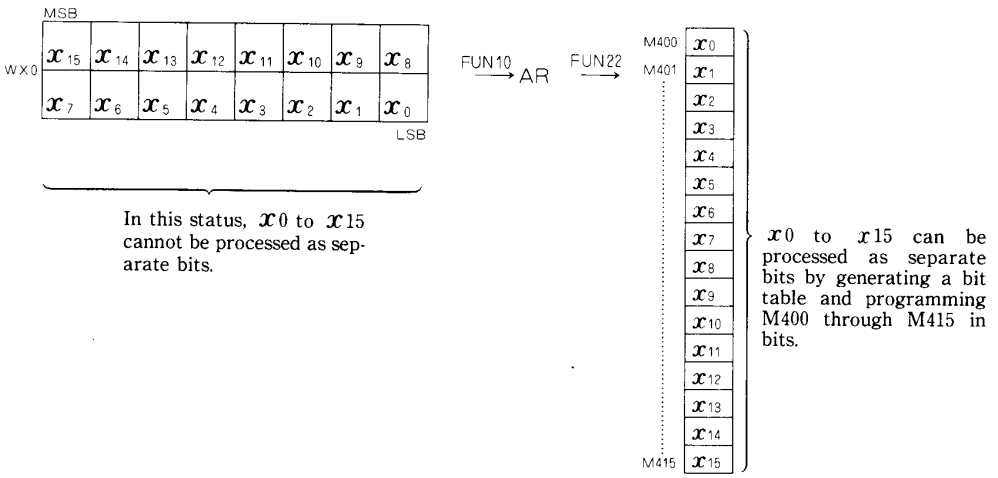
1. External inputs (x 0 to x 15) of slave station CH0 are fetched in words and output to internal outputs M400 through M415.
2. Timer T/C10 is started by ×0 of slave station.

[Programming]



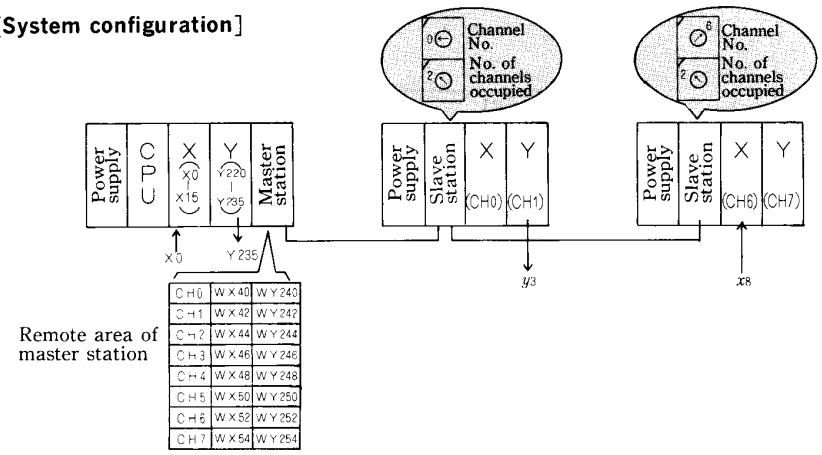
[Explanation]

1. For programming data on slave station side as bit data, a bit table must be generated with internal outputs and the relevant internal output is subjected to bit processing. A bit table must also be generated for output.



〈Application example (4) of remote I/O〉

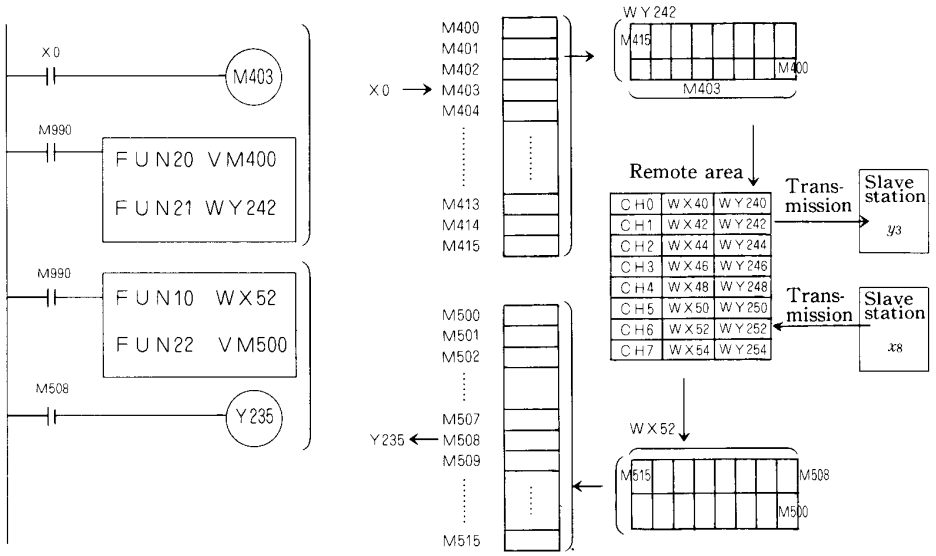
[System configuration]



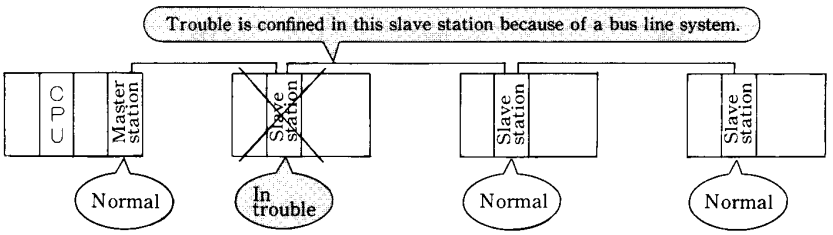
[Explanation of operation]

1. Input ×8 on the slave station side is sent to output Y235 on the master station side.
2. Input ×0 on the master station side is sent to output Y3 on the slave station side.

[Programming]



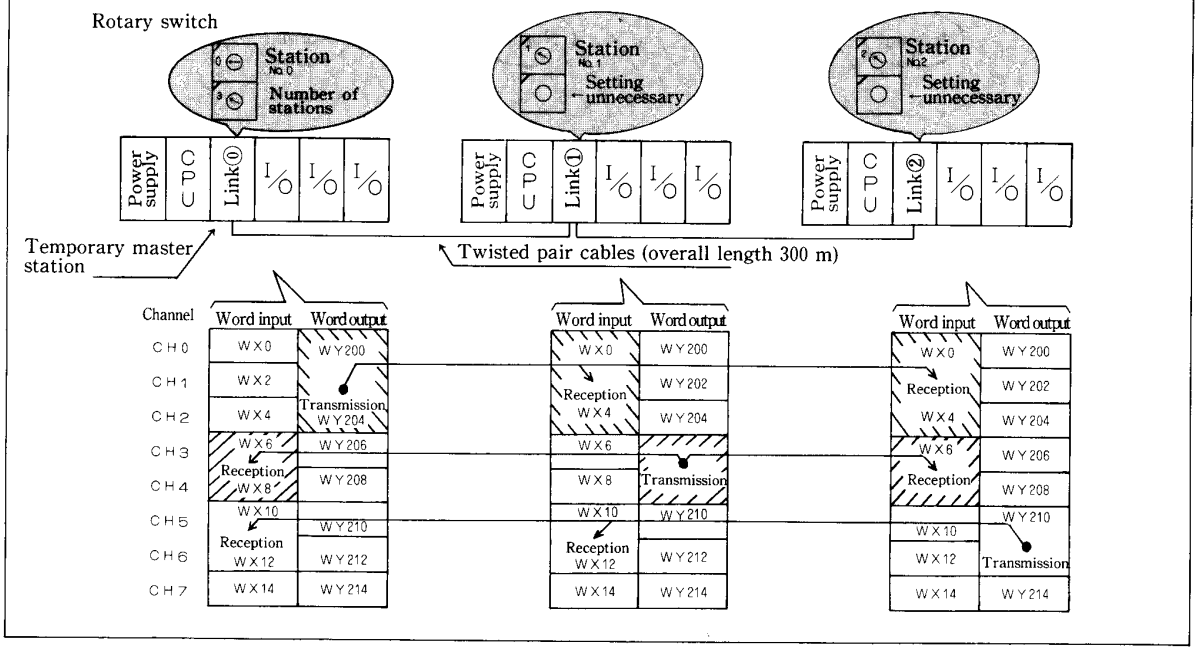
[Troubleshooting]



[Explanation]

1. Even if trouble occurs on a remote slave station, it seldom results in overall system down.
   In the above example, the slave station modules other than the one in trouble operate normally.
   If power supply to one of the slave station modules is cut off, other modules operate normally.
2. In case master station cannot receive data:
   When the master station does not receive data from a slave station for about 500ms, the slave station is judged to be inoperative and the data in that slave station's channel to be output to the CPU module is handled as follows.
   (1) The value before occurrence of trouble is held when the OUT HOLD terminals are shorted.
   (2) All data are cleared to "0" when the OUT HOLD terminals are open.
   If all slave station modules are inoperative, the RUN lamp of master goes off.
3. In case slave station cannot receive data:
   When a slave station does not receive data in about 500 ms, the RUN lamp goes off and output data of the output module on the slave station side is handled as follows.
   (1) The value before occurence of trouble is held when the OUT HOLD terminals are shorted.
   (2) All data are cleared to "0" when the OUT HOLD terminals are open.
4. Wrong setting of rotary switch (slave station)
   The conditions below are checked upon energization. If wrong setting is detected, the RUN lamp flickers at 0.1 sec intervals. So make correct setting and turn on power supply again.
   (1) Station No. (channel No.) within 0 to 7
   (2) Number of channels occupied within 1 to 8
   (3) Station No. (channel No.) + number of channels occupied 1 to 8
   Each module reads the setting of each rotary switch only on energization.

[System configuration]



[Explanation]

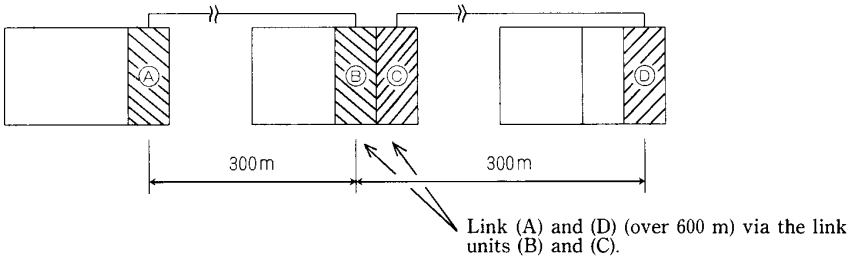
1. I/O link system can be composed by interconnecting the I/O link modules [IOLM-T] with twisted pair cables.
   Up to 8 I/O link modules are connectable.
2. The EM-II series does not have a dedicated link area. Therefore, I/O area is used for linkage. Link area consists of 8 words and it is freely assigned in blocks of word (16 points) in each module.
   An area assigned to transmission in each module corresponds to the reception area in other modules.
   Therefore, the data output in words to the link area of one link module can be received in the word input area in the same channel of all other link modules.
3. The link module has rotary swithes, by which the following items need be set.
   (1) Station No.
   Set sequentially starting from No. 0. Station No. 0 cannot be omitted because it represents a temporary master station.
   (2) Only the module of station No. 0 is required to set the number of IOLM units used in the same link system (3 units in the above example). This setting is unnecessary in modules other than station No. 0.

[**Specifications of I/O link module**]

	Item		Specifications				
General specifications	Operating temperature		0~55°C				
	Storage temperature		−20~70°C				
	Operating humidity		30~90% RH (non-condensing)				
	Power consumption	CH1 (5V)	150mA				
		CH2 (24V)	20mA				
		CH3 (24V)	5mA				
	Dimensions (mm)		35W × 150H × 117D				
	Weight (kg)		0.2				
Functional specifications	No. of connectable units		8 units/temporary master station (including temporary master station)				
	No. of link points		8 words (128 points)				
	Transmission speed		768Kbps				
	Refresh time		10ms × no. of stations				
	Error check		Reverse double transmission				
Transmission line	Recommended cable				Cable length		Terminating resistor
	Type		Maker	Outside diameter	Between stations	Overall extension	
	CO-SPEV-SB-1P	−03mm²	Hitachi Cable	about ϕ5.5	150mMAX	150MAX	Incorporated in module (100 Ω)
	CO-EV-SX-1P	−0.75mm²		about ϕ16	300mMAX	300mMAX	External connection required (150 Ω)

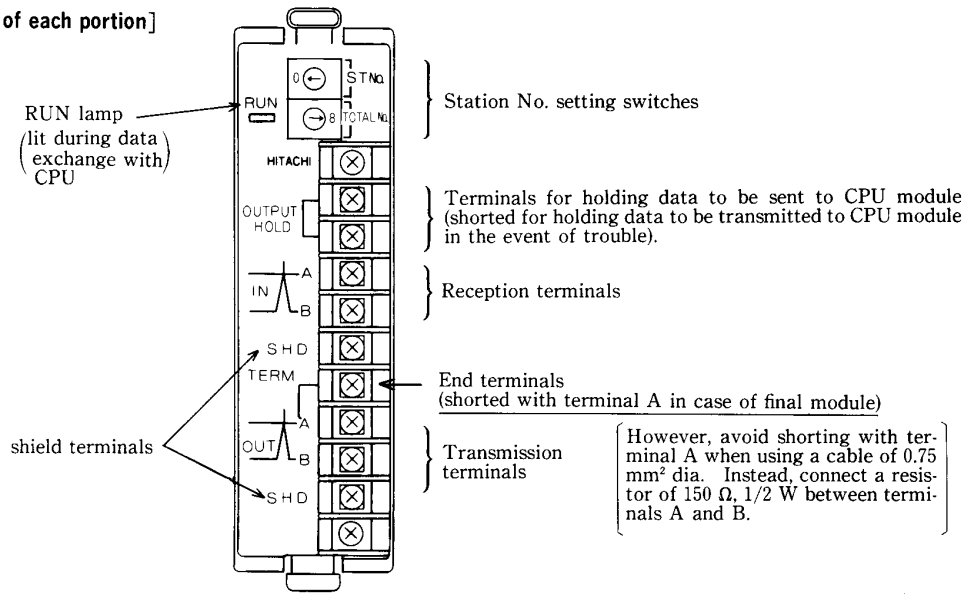
[**Explanation**]

1. Each station takes 10 ms for refreshing. Necessary time is extended in proportion to the number of stations.
2. When the overall distance of transmission line exceeds 300 m, the measure shown below must be taken.



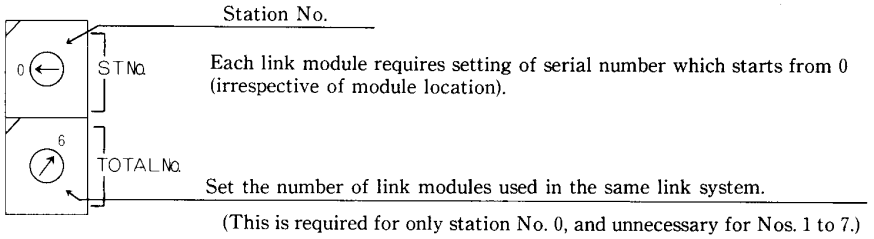
8

[Name of each portion]



[Explanation]

1. Method of setting station No.

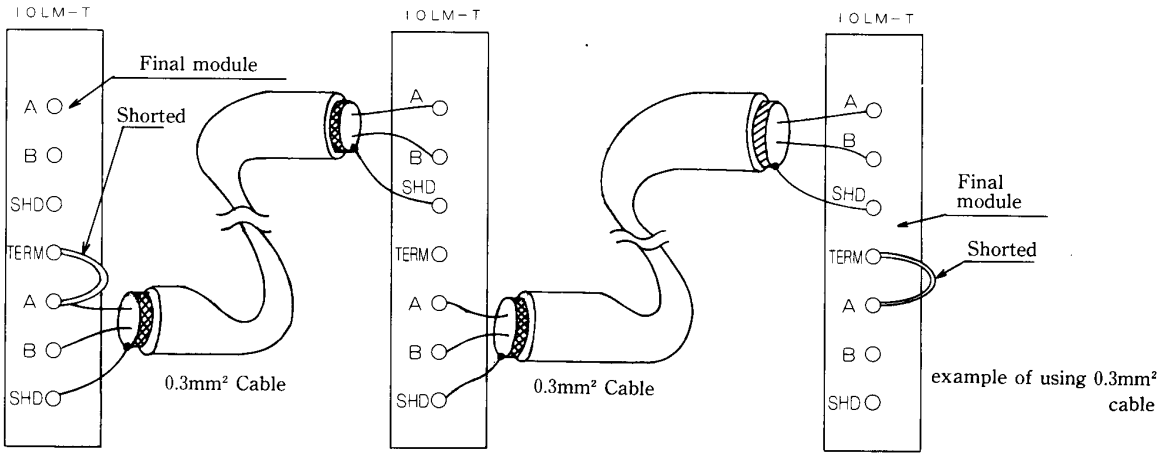


2. OUT HOLD terminals

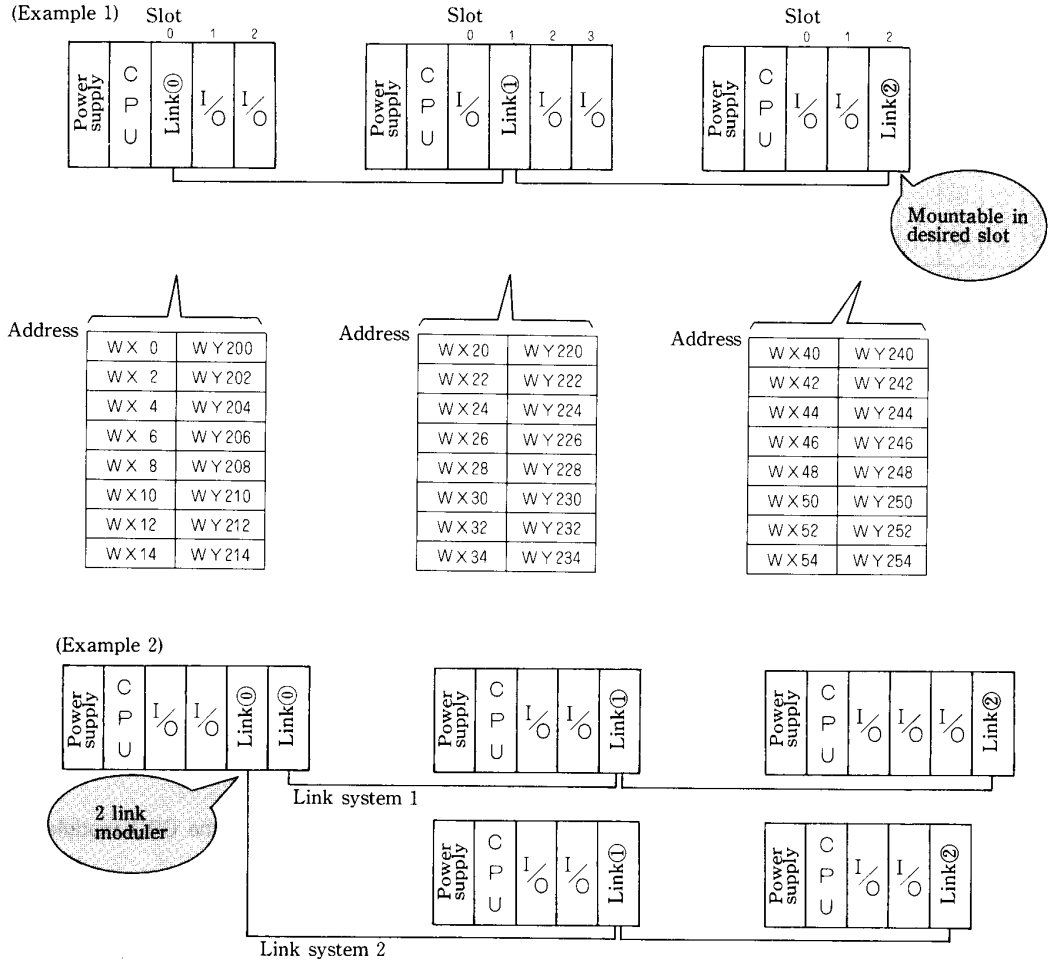
These terminals determine the data to be transmitted from a link module to the CPU module if the link system is abnormal.

- ○ OUT HOLD terminals shorted····· Link data is held at the value before occurrence of trouble.
- ○ OUT HOLD terminals open····· All link data are cleared to "0."

3. Wiring is exemplified below.



[Mounting location]



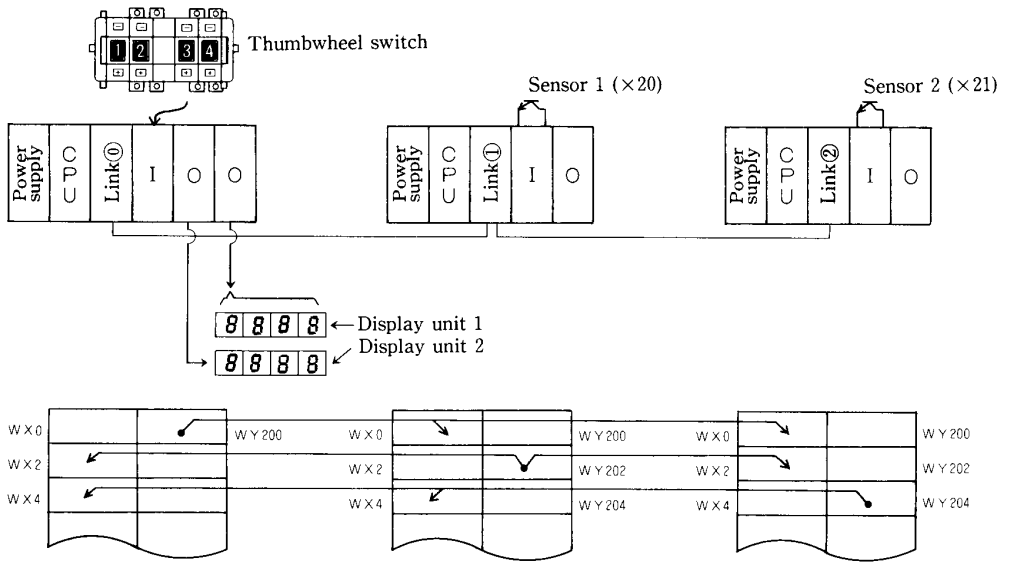
[Explanation]

1. Link module is mountable in a desired slot other than for CPU and power supply modules. The address of link module is determined according to the mounting slot position as in other I/O modules. Link module has a 1-slot width.
2. Multiple link systems can be composed by mounting 2 or more temporary master stations (station No. 0) as shown in the example 2 above. However, such a system must be composed within the current capacity of power supply module.

8

〈Application example (1) of I/O link〉

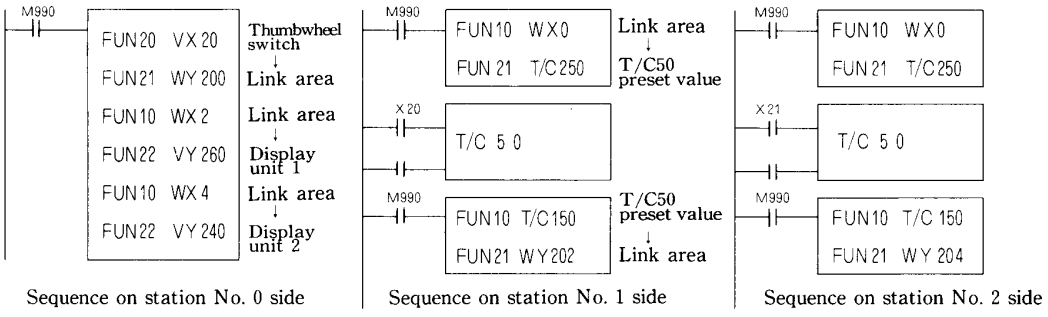
[System configuration]



[Explanation of operation]

1. The thumbwheel switch on the station No. 0 side is used to set the preset value of counter T/C50 in the CPU module on the station No. 1/2 side.
2. ON/OFF operations of sensors 1 and 2 are counted by the counters T/C50 in the CPU modules on the station No. 1 and 2 sides, respectively.
3. The current value of counter on the station No. 1 side is indicated on the display unit 1 on the station No. 0 side, while the current value of counter on the station No. 2 side is indicated on the display unit 2 on the station No. 0 side.

[Programming]



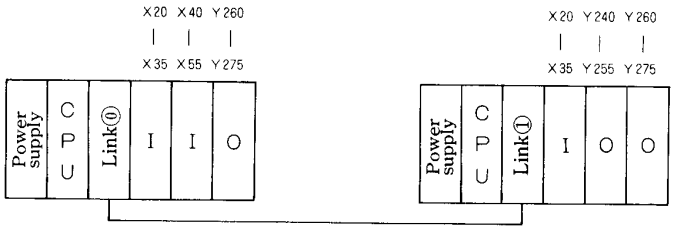
Sequence on station No. 0 side

Sequence on station No. 1 side

Sequence on station No. 2 side

〈Application example (2) of I/O link〉

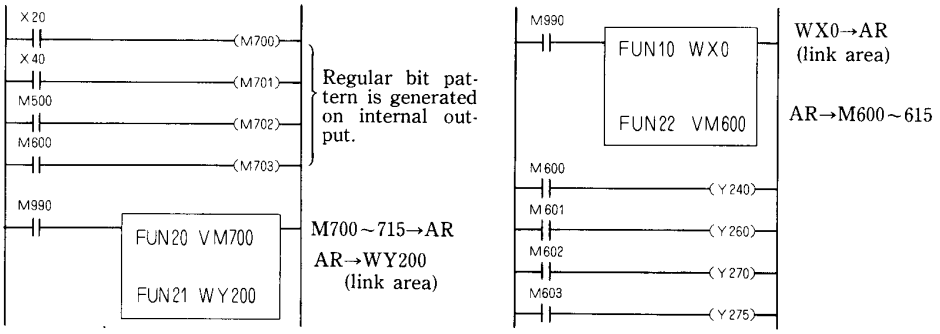
[System configuration]



[Explanation of operation]

1. Bit data on station No. 0 side is sent to the outputs on the station No. 1 side as follows.
   X20 → Y240, X40 →Y260, M500 →Y270, M600→Y275

[Programming]

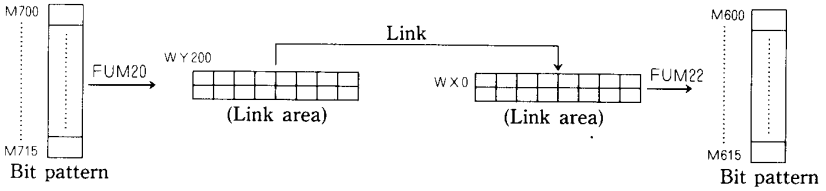


Sequence of station No. 0

Sequence of station No. 1

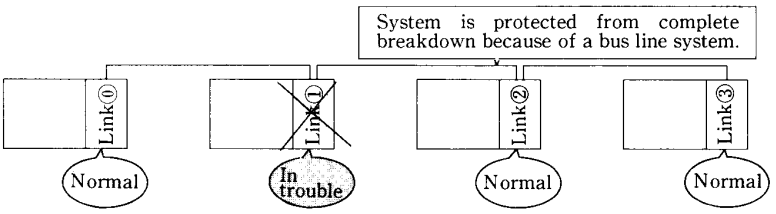
[Explanation]

1. For linking bit data, generate a bit pattern on internal output as shown above (M700 through M715, M600 through M615 in the above example) and output it to link area.



8

[Troubleshooting]



[Explanation]

1. Even if trouble occurs on a link module, it seldom results in overall system down .
   In the above example, the stations other than the link module of station No. 1 is trouble operate normally. If power supply to the station No. 1 is cut off, other modules operate normally.
2. In case master station cannot receive data:
   When the master station does not receive data for a time period 6 times as long as the refresh time (480 ms max.), system is judged to have gone down. And the RUN lamp master station goes off and transmission is interrupted. Thereafter, send request is made to the slave station every 9 ms.
   When the slave station recovers from trouble, the master station is automatically reset to compose the original link system again.
3. In case slave station cannot receive data:
   When slave station does not receive data in about 500 ms, system is judged to have gone down. And the RUN lamp of that slave station goes off and it stands by for reception.
4. **Data transmission between CPU module and link module of EM-II series in the event of system down**
   (1) OUT HOLD terminals shorted……Holds link data before occurence of trouble.
   (2) OUT HOLD terminals open……Clears all link data to "0."
5. **Wrong setting of rotary switch**
   The conditions below are checked upon energinzation. If wrong setting is detected, the RUN lamp of link module flickers at 0.1 sec intervals. So correct setting and then turn on power supply again.
   (1) Station No. within 0 to 7
   (2) No. of stations within 2 to 8 (only on master station)
6. **Double assignment of station No.**
   If a station number is used doubly, transmission does not occur and the RUN lamp of link module flickers at 0.5 sec intervals. So make correct setting and turn on power supply again.

Appendix : Excution time of EM-II

	Instructions		Time
Basic instructions	ORG		1.5μs
	ORG NOT		1.5
	STR		1.5
	STR NOT		1.5
	AND		1.5
	AND NOT		1.5
	OR		1.5
	OR NOT		1.5
	AND STR		1.5
	OR STR		1.5
	OUT		1.5
	OUT NOT		1.5
	OUT(T, 0.1s)		85.5
	OUT(T, 10ms)		94.5
	OUT(C)		86
Application Instructions	FUN 00	DIF	13
	01	DFN	13
	02	IF	1.5
	03	IFR	1.5
	04	MCS	15
	05	MCR	13.5
	06	JMP	1.5
	07	JEND	1.5
	08	AJMP	13
	09	AJEND	3
	28	BRANCH	1.5
	29	RETURN	1.5
	40	UDC	427.5
	41	NOP	1.5
	45	LACH	23
	47	SFR	162.5
	88	SET	1.5
	89	RES	1.5
	98	STA	1.5
	99	END	*1)
	42	CALL	51
	43	SB	—
	44	RTS	41.5
	91	REFX	21.5
	92	REFY	15
	93	INT	—
	94	RTI	68

*1) END processing
CPM-E2···2,200μs
CPM-E3···2,300μs

	Instructions		Time
Arithmetic Instructions	FUN 0	LOADI	8.5μs
	1	ANDI	19.5
	2	SUBI	457
	3	MULI	523.5
	4	DIVI	1445
	5	ANDI	11.5
	6	ORI	11.5
	7	CPEHI	13
	8	CPEI	13
	9	CPLI	13
	10	LOADW	14.5
	11	ADD	26
	12	SUB	463
	13	MUL	529.5
	14	DIV	1451
	15	AND	18
	16	OR	18
	17	CPEH	19
	18	CPE	19
	19	CPL	19
	20	LOADB	187
	21	OUTW	14.5
	22	OUTB	247.5
	23	OUTC	8.5
	24	BCD	279
	25	BNR	97.5
	26	LSFR	18.5
	27	RSFR	18.5
	48	EX	1153.5
	49	DB	1150
	50	LDBYTI	7.5
	51	ABYTI	15
	52	SBYTI	15
	53	MBYTI	106.5
	54	DBYTI	1028
	55	BANDI	9
	56	BORI	9
	57	BCPEHI	12
	58	BCPEDI	12
	59	BCPLI	12
	60	BLOAD	13.5
	61	ADDBNR	19.5
	62	SUBBNR	19.5
	63	MULBNR	112.5
	64	DIVBNR	1034
	66	EXOR	18
	71	BOUT	13.5
	72	MASKL	27
	73	MASKR	27
	74	SEG	11
	75	ASC	11
	76	ROL	17
	77	ROR	17
	78	ENCOD	107.5
	79	DECOD	19
	80	SWAP	8
	81	BSWAP	9
	82	XCG	11
	83	CLC	6
	84	SEC	6
	85	WNOT	9

8