CVM1-PRS21-EV1 Programming Console
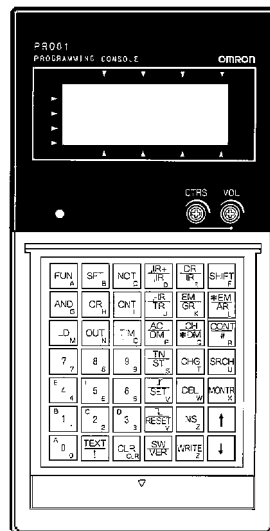
Operation Manual

Revised January 1999

Notice:

OMRON products are manufactured for use according to proper procedures by a qualified operator and only for the purposes described in this manual.

The following conventions are used to indicate and classify precautions in this manual. Always heed the information provided with them. Failure to heed precautions can result in injury to people or damage to property.

⚠ **DANGER**   Indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.

⚠ **WARNING**   Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.

⚠ **Caution**   Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury, or property damage.

OMRON Product References

All OMRON products are capitalized in this manual. The word "Unit" is also capitalized when it refers to an OMRON product, regardless of whether or not it appears in the proper name of the product.

The abbreviation "Ch," which appears in some displays and on some OMRON products, often means "word" and is abbreviated "Wd" in documentation in this sense.

The abbreviation "PC" means Programmable Controller and is not used as an abbreviation for anything else.

Visual Aids

The following headings appear in the left column of the manual to help you locate different types of information.

Note   Indicates information of particular interest for efficient and convenient operation of the product.

1, 2, 3...   1.   Indicates lists of one sort or another, such as procedures, checklists, etc.

TABLE OF CONTENTS

TABLE OF CONTENTS

About this Manual:

This manual describes the installation and operation of the CVM1-PRS21-EV1 Programming Console and includes the sections described below. This Programming Console is used for CVM1, CVM1D, and CV-series PCs and cannot be used for C-series PCs.

Please read this manual carefully and be sure you understand the information provided before attempting to install and operate the Programming Console.

Section 1 describes the applications environment required for the Programming Console and lists the operations it does and does not support. Details are provided in following sections.

Section 2 describes the functional parts of the Programming Console and connection procedures for installing the Programming Console.

Section 3 describes the basic functions of the keys on the keypad and the main displays that appear during operation.

Section 4 individually describes the operations that are possible from the Programming Console.

Section 5 describes errors that can occur during Programming Console operation, including both errors directly relation to Programming Console operation and PC errors.

Various appendices are also provided for easy reference. Refer to the table of contents for a list of these.

⚠ **WARNING** Failure to read and understand the information provided in this manual may result in personal injury or death, damage to the product, or product failure. Please read each section in its entirety and be sure you understand the information provided in the section and related sections before attempting any of the procedures or operations given.

PRECAUTIONS

This section provides general precautions for using the Programmable Controller (PC) and related devices.

The information contained in this section is important for the safe and reliable application of the Programmable Controller. You must read this section and understand the information contained before attempting to set up or operate a PC system.

1 Intended Audience

This manual is intended for the following personnel, who must also have knowledge of electrical systems (an electrical engineer or the equivalent).

• Personnel in charge of installing FA systems.

• Personnel in charge of designing FA systems.

• Personnel in charge of managing FA systems and facilities.

2 General Precautions

The user must operate the product according to the performance specifications described in the operation manuals.

Before using the product under conditions which are not described in the manual or applying the product to nuclear control systems, railroad systems, aviation systems, vehicles, combustion systems, medical equipment, amusement machines, safety equipment, and other systems, machines, and equipment that may have a serious influence on lives and property if used improperly, consult your OMRON representative.

Make sure that the ratings and performance characteristics of the product are sufficient for the systems, machines, and equipment, and be sure to provide the systems, machines, and equipment with double safety mechanisms.

This manual provides information for programming and operating the Unit. Be sure to read this manual before attempting to use the Unit and keep this manual close at hand for reference during operation.

⚠ **WARNING** It is extremely important that a PC and all PC Units be used for the specified purpose and under the specified conditions, especially in applications that can directly or indirectly affect human life. You must consult with your OMRON representative before applying a PC System to the above-mentioned applications.

3 Safety Precautions

⚠ **WARNING** Do not attempt disassembly a Unit while the power is being supplied. Doing so may result in electric shock.

⚠ **WARNING** Do not touch any of the terminals while the power is being supplied. Doing so may result in electric shock.

⚠ **WARNING** Do not attempt to disassemble, repair, or modify the Unit. Any attempt to do so may result in malfunction, fire, or electric shock.

⚠ **Caution** Execute online unit replacement (hot swapping) only after confirming that no adverse effects will be caused to the system. Otherwise, the system may malfunction.

4 Operating Environment Precautions

⚠ **Caution** Do not operate the control system in the following places:

- Locations subject to direct sunlight.
- Locations subject to temperatures outside the range 0 to 55°C.
- Locations subject to humidity outside the range 10 to 90%.
- Locations subject to condensation as the result of severe changes in temperature.
- Locations subject to corrosive or flammable gases.
- Locations subject to dust (especially iron dust) or salts.
- Locations subject to exposure to water, oil, or chemicals.
- Locations subject to shock or vibration.

⚠ **Caution** Take appropriate and sufficient countermeasures when installing systems in the following locations:

- Locations subject to static electricity or other forms of noise.
- Locations subject to strong electromagnetic fields.
- Locations subject to possible exposure to radioactivity.
- Locations close to power supplies.

⚠ **Caution** The operating environment of the PC System can have a large effect on the longevity and reliability of the system. Improper operating environments can lead to malfunction, failure, and other unforeseeable problems with the PC System. Be sure that the operating environment is within the specified conditions at installation and remains within the specified conditions during the life of the system.

5 Application Precautions

Observe the following precautions when using the PC System.

⚠ **WARNING** Always heed these precautions. Failure to abide by the following precautions could lead to serious or possibly fatal injury.

- Always connect to a ground of 100 Ω or less when installing the Units. Not connecting to a ground of 100 Ω or less may result in electric shock.
- Always turn OFF the power supply to the PC before attempting any of the following. Not turning OFF the power supply may result in malfunction or electric shock.
  - Mounting or dismounting Memory Cassettes.
  - Assembling the Units.
  - Setting switches.
  - Connecting or wiring the cables.

⚠ **Caution** Failure to abide by the following precautions could lead to faulty operation of the PC or the system, or could damage the PC or PC Units. Always heed these precautions.

- Fail-safe measures must be taken by the customer to ensure safety in the event of incorrect, missing, or abnormal signals caused by broken signal lines, momentary power interruptions, or other causes.

- Interlock circuits, limit circuits, and similar safety measures in external circuits (i.e., not in the Programmable Controller) must be provided by the customer.

- Always use the power supply voltage specified in the operation manuals. An incorrect voltage may result in malfunction or burning.

- In places where the power supply is unstable, take appropriate measures to ensure that the rated power supply is made available. An incorrect power supply may result in malfunction.

- Install external breakers and take other safety measures against short-circuiting in external wiring. Insufficient safety measures against short-circuiting may result in burning.

- Disconnect the functional ground terminal when performing withstand voltage tests. Not disconnecting the functional ground terminal may result in burning.

- Check the connectors before mounting. Not doing so may result in malfunction.

- Install the Unit properly as specified in the operation manual. Improper installation of the Unit may result in malfunction.

- Be sure that all the mounting screws, terminal screws, and cable connector screws are tightened to the torque specified in the relevant manuals. Incorrect tightening torque may result in malfunction.

- Use crimp terminals for wiring. Do not connect bare stranded wires directly to terminals. Connection of bare stranded wires may result in burning.

- Double-check all the wiring before turning ON the power supply. Incorrect wiring may result in burning.

- Be sure that the terminal blocks, Memory Units, expansion cables, and other items with locking devices are properly locked into place. Improper locking may result in malfunction.

- Check the user program for proper execution before actually running it on the Unit. Not checking the program may result in an unexpected operation.

- Confirm that no adverse effect will occur in the system before attempting any of the following. Not doing so may result in an unexpected operation.

    - Changing the operating mode of the PC.

    - Force-setting/force-resetting any bit in memory.

    - Changing any set value.

- Resume operation only after transferring to the new CPU Unit the contents of the DM and HR Areas required for resuming operation. Not doing so may result in an unexpected operation.

- Do not pull on the cables or bend the cables beyond their natural limit. Doing either of these may break the cables.

- Do not place objects on top of the cables. Doing so may break the cables.

- Do not hold the cables when transporting the Unit. Doing so may break the cables.

- When replacing parts, be sure to confirm that the rating of a new part is correct. Not doing so may result in malfunction or burning.

- Before touching the Unit, be sure to first touch a grounded metallic object in order to discharge any static built-up. Not doing so may result in malfunction or damage.

SECTION 1
Introduction

This section describes the applications environment required for the Programming Console and lists the operations it does and does not support. Details are provided in following sections.

1-1 Features

The CVM1-PRS21-EV1 Programming Console is an on-site Programming Device for SYSMAC CVM1, CVM1D, and CV-series PCs. It can be used together with other Programming Devices, such as the SSS (SYSMAC Support Software), to enable on-site maintenance of user programs and data.

1-2 Programming Console Capabilities

The procedure for programming and operating a PC is illustrated below. All of the operations in bold boxes can be performed from the Programming Console.

First, programs are written into the PC and debugged using another Programming Device. Then, the Programming Console can be used to alter or monitor the ladder program, read error data, or execute online Unit replacement (hot swapping).

```
┌──────────────────────────────────┐
│ Input ladder-diagram program.    │
└──────────────────────────────────┘
                │
┌──────────────────────────────────┐
│ Check program                    │
└──────────────────────────────────┘
                │
┌──────────────────────────────────┐
│ Create I/O table                 │
└──────────────────────────────────┘
                │
┌──────────────────────────────────┐
│ Transfer program to PC from      │
│ Programming Device               │
└──────────────────────────────────┘
                │
┌──────────────────────────────────┐      ┌──────────────────────────────────┐
│ Debug                            │      │ Correct program                  │
└──────────────────────────────────┘      └──────────────────────────────────┘
                │                                          ▲
          ┌──────────┐                                     │
          │ OK?      │─────────────────────────────────────┘
          └──────────┘
                │
┌──────────────────────────────────┐      ┌──────────────────────────────────┐
│ Run and monitor trial operation  │      │ Read error data and monitor      │
└──────────────────────────────────┘      └──────────────────────────────────┘
                │                                          ▲
          ┌──────────┐                                     │
          │ OK?      │─────────────────────────────────────┘
          └──────────┘
                │
┌──────────────────────────────────┐
│ Test and monitor full-system     │
│ operation and read error data    │
└──────────────────────────────────┘
```
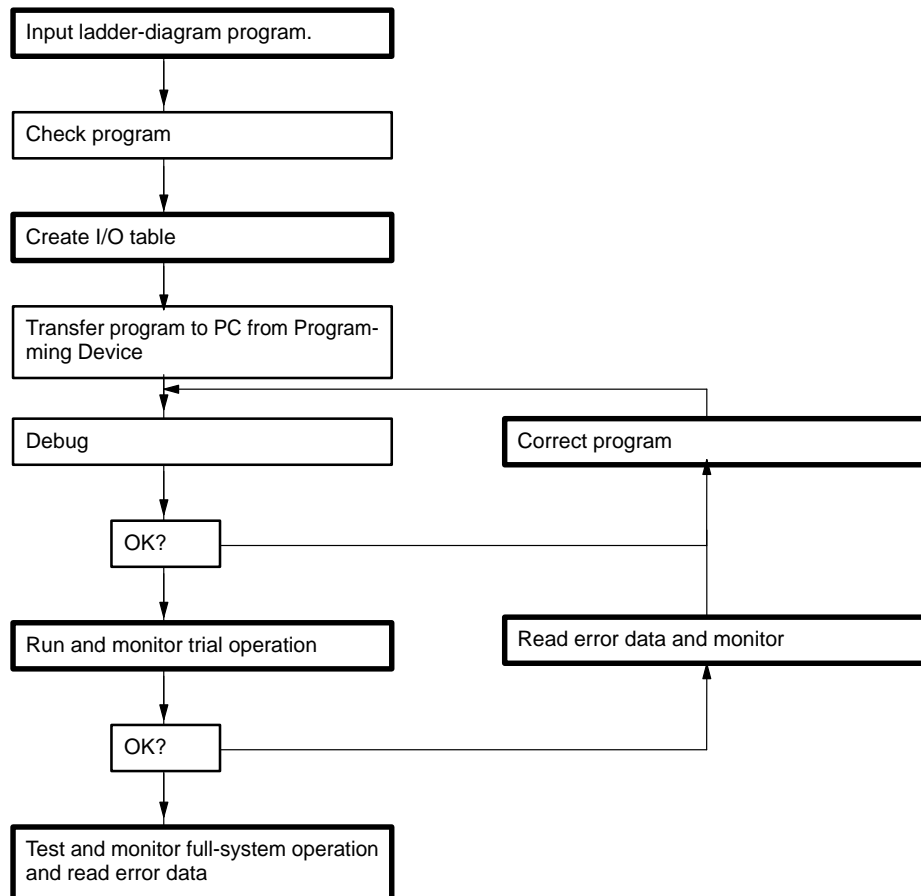
1-3   Connecting Programming Consoles

1-3-1   CVM1 and CV-series PCs

It is possible to connect the Programming Console to any of the following:

• CV500, CV1000, CV2000, or CVM1 CPU Units.

• CV500-II201 I/O Interface Units

• SYSMAC BUS/2 Remote I/O Slave Units.

Programming Devices cannot be connected to both the CPU Unit and an I/O Interface Unit at the same time on the same PC. If more than one is connected, only one, at most, will be operative. Refer to *1-3 Programming Devices Restrictions* for details.

It is possible to connect a maximum of four Programming Devices to the same PC, i.e., one to either the CPU Unit or I/O Interface Unit and up to three to SYSMAC BUS/2 Remote I/O Slave Units, or if one is not connected to the CPU Unit or I/O Interface Unit, up to four to SYSMAC BUS/2 Remote I/O Slave Units. A maximum of two Programming Consoles, however, can be connected to Slaves under the same Remote I/O Master Unit.

Note The DIP switch of the Unit to which the Programming Console is connected must be set to a baud rate of 50 kbps. The baud rate for the I/O Interface Unit is set on the DIP switch on the CPU Unit.
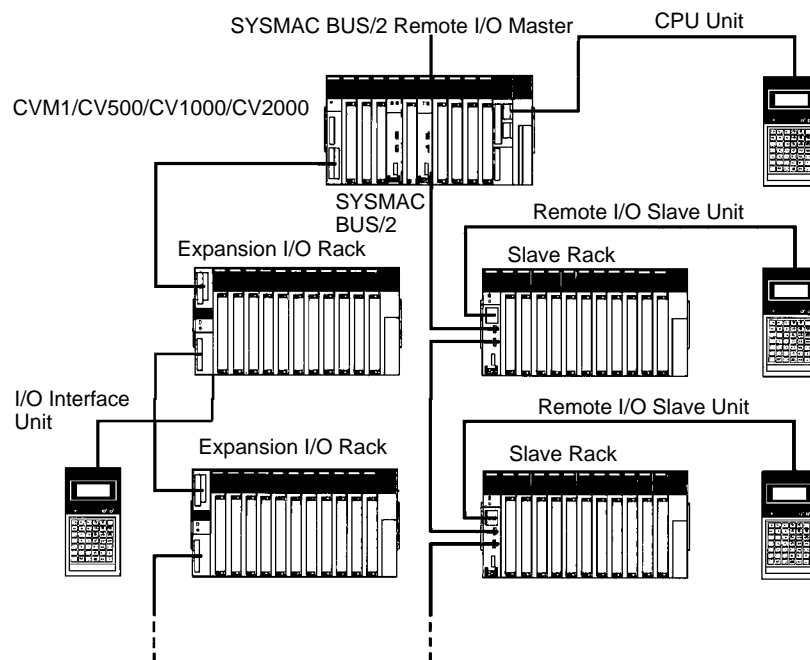
Programming Device Restrictions

The number of Programming Devices, including Programming Consoles and personal computers running CVSS (CV Support Software), SSS (SYSMAC Support Software), or SYSMAC-CPT (see note) must be within the following limitations for any one PC.

Connection point	Maximum number of Programming Devices		
CPU Unit	1	0	0
CV500-II201 I/O Interface Unit	0	1	
SYSMAC BUS/2 Remote I/O Slave Units	3	3	4
	(2 max. per Master)		
Maximum total	4		

Note The SYSMAC-CPT does not support the CVM1D Duplex System.

1-3-2 CVM1D PCs

Programming Consoles or other Programming Devices cannot be connected directly to the CPU Unit of a CVM1D Duplex System. Connect to the Duplex Unit instead of the CPU Unit.

The CVM1-PRS21-EV1 Programming Console can be used for the CVM1, CVM1D, and CV-series PC, but not for C-series PCs. The Programming Console can be connected to the following Units.

• CVM1D-DPL01 Duplex Unit

• CV500-II201 I/O Interface Unit (Expansion I/O Rack)

• CV500-RT211 Optical SYSMAC BUS/2 Remote I/O Slave Unit
  CV500-RT221 Wired SYSMAC BUS/2 Remote I/O Slave Unit

Programming Devices cannot be connected to both the Duplex Unit and an I/O Interface Unit at the same time on the same PC. Connect to only one of these at any one time.

It is possible to connect a maximum of four Programming Devices to the same PC, i.e., one to either the Duplex Unit or I/O Interface Unit and up to three to SYSMAC BUS/2 Remote I/O Slave Units. If one is not connected to the Duplex Unit or I/O Interface Unit, up to four can be connected to SYSMAC BUS/2 Remote I/O Slave Units. A maximum of two Programming Consoles, however, can be connected to Slaves under the same Remote I/O Master Unit.

Note A Unit to which a Programming Console is connected must be set to a baud rate of 50 kbps. The baud rate for the I/O Interface Unit is set on the DIP switch on the Duplex Unit.
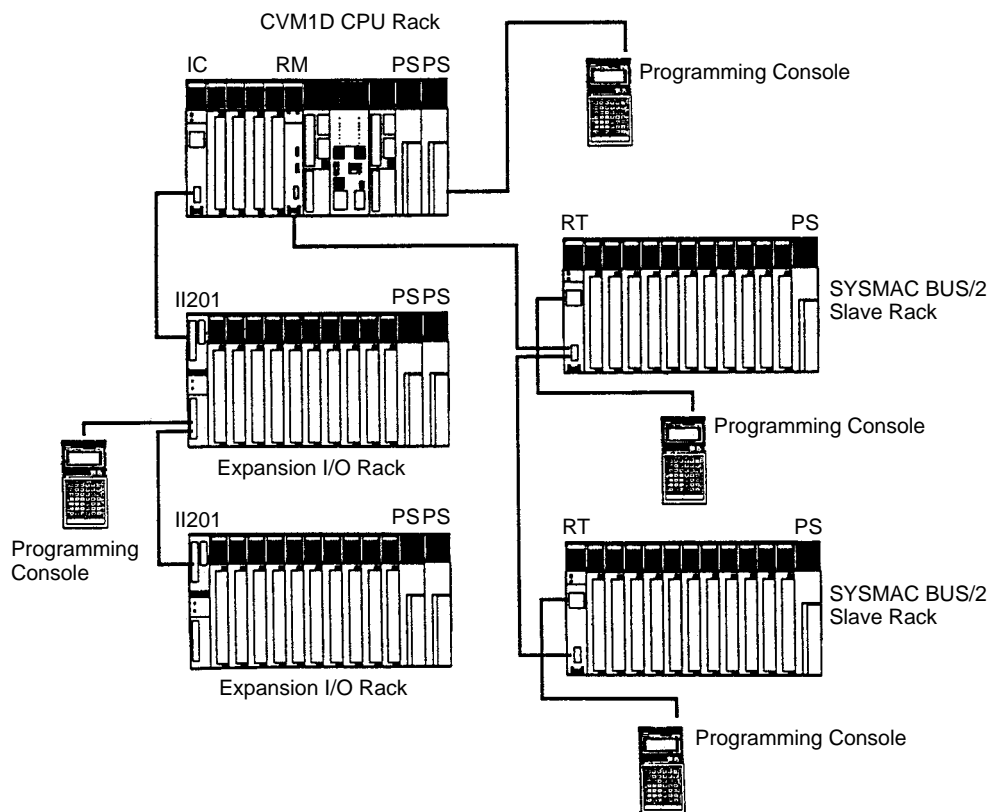
Note A CVM1D Backplane cannot be used for a SYSMAC BUS/2 Slave Rack.

1-3-3 Connecting Cables

Use one of the following Cables (sold separately) to connect the Programming Console to the PC.

Model	Cable length
CV500-CN224	2 m
CV500-CN424	4 m
CV500-CN624	6 m

1-3-4 Priority of Programming Devices

If Programming Devices are connected to both the CPU Unit and one or more I/O Interface Units within the same PC, only one, at most, will be operative. The following rules of priority apply.

1, 2, 3... 1. If a computer running the CVSS, SSS, or SYSMAC-CPT is connected to the peripheral device connector on the CPU Unit, it will take priority over a Programming Console connected anywhere.

2. If more than one computer running the CVSS, SSS, or SYSMAC-CPT is connected, all will be inoperative.

3. If more than one Programming Console is connected, the first one that is connected will take priority.

The following table is for easy reference when multiple Programming Devices are connected to the CPU Unit and I/O Interface Units.

Programming Devices connected later	Programming Devices connected first	
	CVSS, SSS, or SYSMAC-CPT	Programming Console
CVSS, SSS, or SYSMAC-CPT	All stop operating	CVSS, SSS, or SYSMAC-CPT
Programming Console	CVSS, SSS, or SYSMAC-CPT	Programming Console connected first

Programming Devices connected to Remote I/O Slave Units do not differ from one another in priority and will all operate as long as the maximum number of Programming Devices is not exceeded. Programming Devices connected to Remote I/O Slave Units and one Programming Device connected to the CPU Unit or an I/O Interface Unit will also have the same priority and operate simultaneously.

1-4 Lists of Operations

The Programming Console provides the following operations.

Classification	Operation	Sub-operations
Initial settings	System settings	Response monitor time Message number
	Program designation	---
	PC mode change	
Program editing	Address set	---
	Program read	
	Instruction search	
	Operand search	
	Instruction overwrite	
	Instruction insert	
	Instruction delete	
	Timer/Counter SV change	
Monitoring	Simple I/O monitor	Force reset/set PV change ASCII display Decimal display
	I/O multipoint monitor	
	4-word monitor	
	Word monitor	
	Differential monitor	
Special operations	Clock read/change	---
	UM protect	
	Active step monitor	
	SFC search	
	Action block change	
	Transition change	
Online Unit replacement	Online Unit replacement	
PC data read	Cycle time display	---
	Automatic error display	
	Error data read	Error display/clear Error history read Message read
PC setup	Startup mode	---
	Execution controls	Duplicate action error (CV Series only) Comparison error process
	Host link	---
	Scheduled interrupt (Except for CVM1D)	
	Momentary power interruption time (Except for CVM1D)	
	Cycle time	
	Cycle time monitoring time	
Memory card operations	Memory card format	---
	File write	
	File read	
	File compare	
	File delete	
I/O table operation	I/O table create	---
	I/O table compare	
Memory clear	Memory clear	---

1-5 Unsupported Operations

The CVM1, CVM1D, and CV-series Programming Consoles do not support the following operations. The CVSS, SSS, or SYSMAC-CPT must be used instead.

- Inputting or modifying SFC programs.
- Checking programs.
- The following I/O table operations:

    I/O table read
    I/O table change
    I/O table cancel

- Debugging operations
- The following PC Setup settings:

    Hold areas
    Startup hold
    Startup processing
    I/O refresh
    Execution controls 1
    Execution controls 2 (Execute process, I/O interrupts, Power OFF interrupt, Step timer, Startup trace , Indirect DM binary/BCD, JMP(004) 0000 operation)
    CPU bus link settings
    First words for local racks
    First words for group-1 and group-2 Slaves
    First words for I/O Terminals
    First words for group-3 Slaves
    Error history area
    Display modes at startup
    Scheduled interrupts (Not supported by CVM1D)
    Momentary power interruption time (Not supported by CVM1D)

- The following customization settings:

    Custom function codes
    Custom data area abbreviations (see note 1)
    Custom data areas (see note 2)

- Operations related to Link Units

Note Data area abbreviations changed for the CVSS, SSS, or SYSMAC-CPT cannot be displayed or used with a Programming Console. The normal abbreviations must be used instead. Data area abbreviations for custom data areas can be used along with the normal word/bit addresses.

SECTION 2
Component Parts and Installation

This section describes the functional parts of the Programming Console and connection procedures for installing the Programming Console. Details on basic key functions are provided in *Section 3 Key Functions and Displays*.

2-1   Nomenclature

Display

RUN indicator

Contrast adjustment

Volume adjustment

Keypad

Cable connectors

Panel mounting screw

System memory cassette
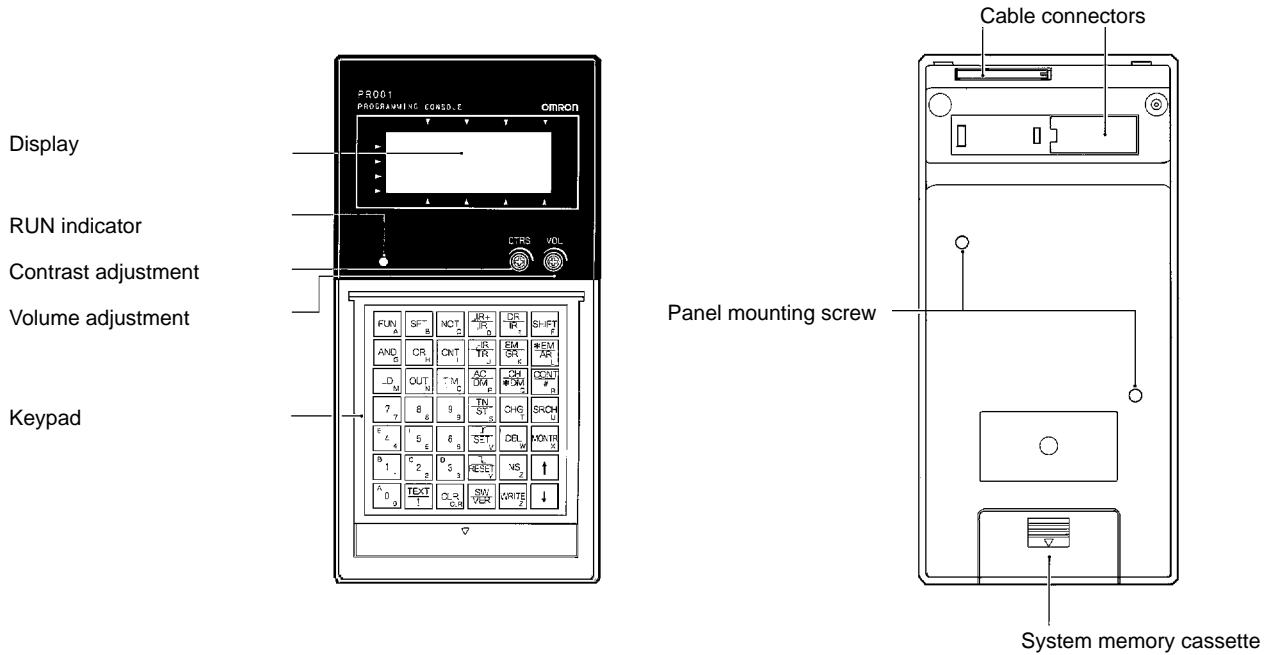
Display	Displays programs and other information.
RUN Indicator	Lights in synchronization with the RUN indicator on the CPU Unit.
Contrast Adjustment	Used to adjust the contrast of the display.
Volume Adjustment	Used to adjust the volume of the buzzer. The buzzer sounds when keys are pressed or when errors occur.
Keypad	Used to input programs and data.
System Memory Cassette	Includes system memory data used to operate the Programming Console. Do not remove the System Memory Cassette.
Cable Connectors	Used to connect the Programming Console to the PC. Use either one of the connectors, but not both at the same time.
Panel Mounting Screws	Used to mount the Programming Console to a panel using the C200H-ATT01 Mounting Bracket (sold separately).

2-2 Connections

Refer to *Section 1 Introduction* for details on connecting cables, connectable Units, and the maximum number of Programming Devices allowed per PC.

When connecting to the CPU Unit, connect the Programming Console to the peripheral device connector (PERIPHERAL). The Programming Console can be connected to the PC while the PC is turned on, and even while the PC is operating.

Connecting to the Programming Console

1, 2, 3...   1. Remove the cable connector cover.

The removed cable connector cover should be attached to the rear panel of the Programming Console in order not to lose it.

Cover

Open the cover with a small flat-blade screwdriver or fingernail.

Cover
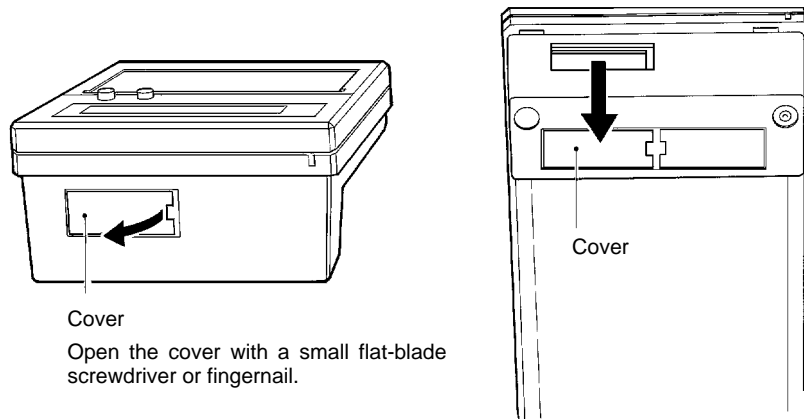
2. Insert the connector.

Be sure to check the direction of the connector and insert it correctly. Insert the connector until it locks into place.
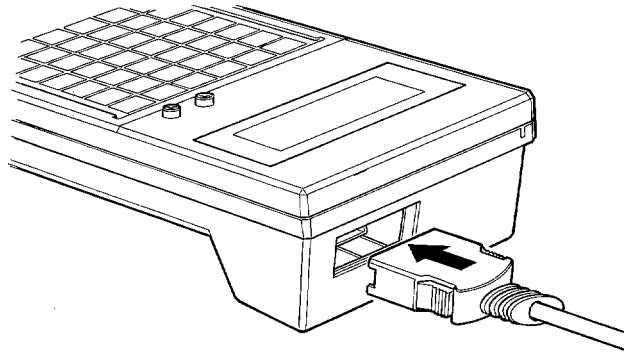
Connecting to the PC

Be sure to check the direction of the connector and insert it correctly. Insert the connector until it locks into place.

CPU Unit (CVM1and CV Series)
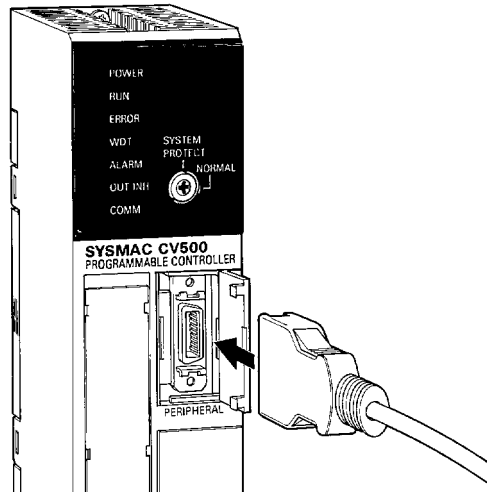
DPL Unit (CVM1D)
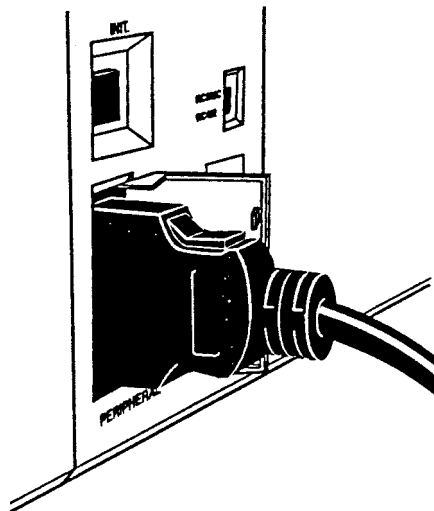
I/O Interface Units
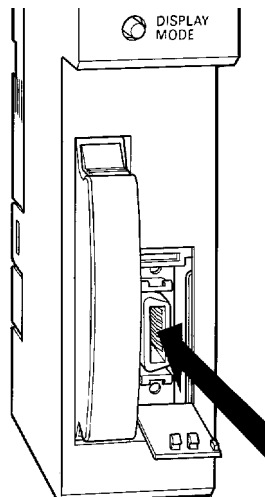
Remote I/O Slave Units
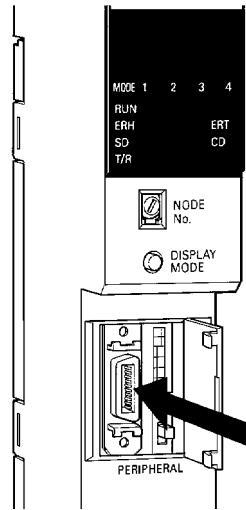
Disconnecting the Cable   Squeeze the lock release levers on the sides of the connector and pull the connector straight out to disconnect.

Lock release lever
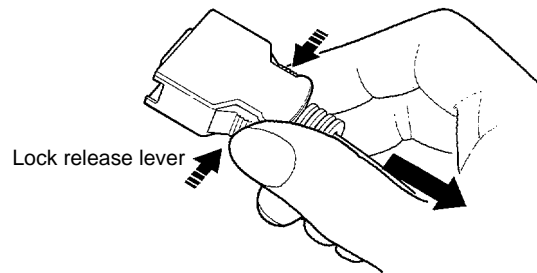
Switch Settings and Adjustments

PC Switch Settings   Set the DIP switch to a baud rate of 50 kbps. The DIP switch on the CPU Unit is effective for both the peripheral device connector and the connector on the I/O Interface Unit. The Remote I/O Slave Unit has its own DIP switch.

Contrast   Turn the contrast adjustment clockwise or counterclockwise to adjust the contrast of the display.

Buzzer   Turn the volume adjustment clockwise or counterclockwise to adjust the volume of the buzzer.

Replacing the Memory Cassette

Use the following procedure to replace the Memory Cassette if you are going to use the CVM1-MP201-V1 Memory Cassette.

1, 2, 3...   1. Remove the screw holding the Memory Cassette from the back of the Programming Console.

Memory cassette

13
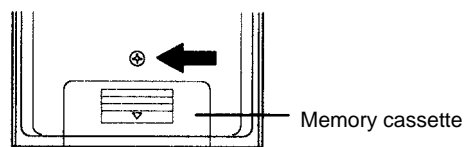
2. Remove the Memory Cassette and replace it with the new Memory Cassette.

DIP switch
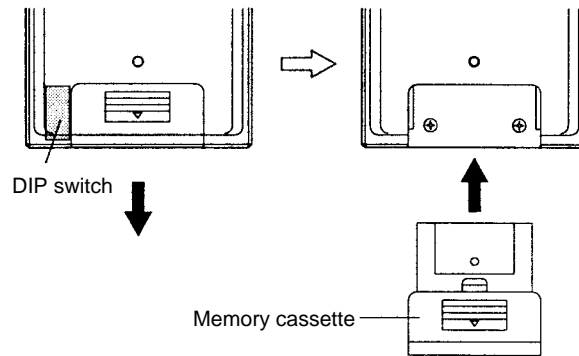
Memory cassette

3. Replace the screw removed in the first step.

Note There is a DIP switch on the left of the Memory Cassette that can be accessed when the Memory Cassette is removed. The language displayed on the Programming Console can be switched between Japanese and English by using the pin furthest to the back. Except for this feature, use with the factory settings unchanged. Set the pin towards the base for Japanese display and towards the case for English display.

Memory Cassette
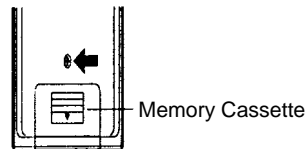
2-3   Panel Mounting

Use the C200H-ATT01 Mounting Bracket (sold separately) to mount the Programming Console to a panel. Cut an opening as shown in the following illustration. The panel cutout conforms to DIN43700.

186 +1.1/–0 mm

92 +.08/–0 mm
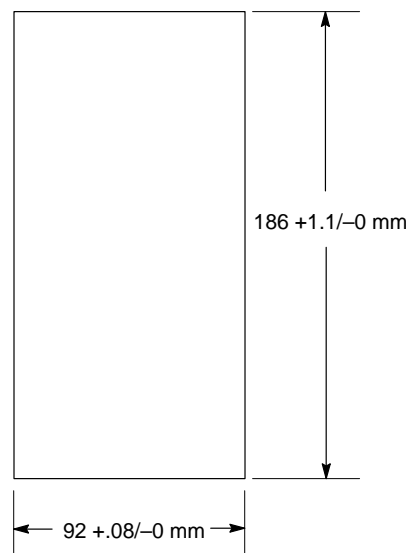
Leave enough space for the connectors as shown by the following illustration.

15.5 mm

80 mm min.

Use either one of the connectors.

70 mm min.
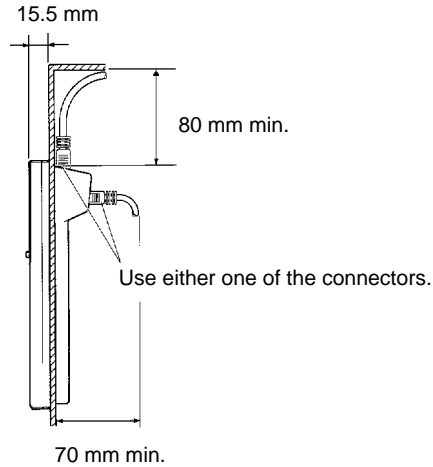
Insert the Programming Console into the cutout, attach the Mounting Bracket from the rear side of the panel, and secure the Mounting Bracket with screws.

Mounting Bracket

Two screws

Panel thickness: 1.0 to 3.2 mm
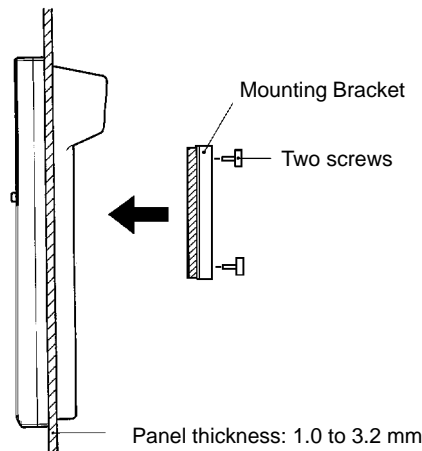
Mounting Bracket Dimensions

75.5 mm

116 mm

9.1 mm

14.1 mm
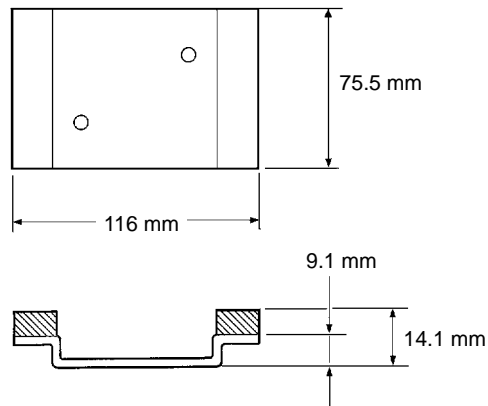
⚠ **Caution**   The ambient temperature must be 0 to 55 °C when the Programming Console is used inside a control panel. The Programming Console may not function properly and may be damaged at temperatures outside this range. Provide air conditioning, ventilation, or heating as required to maintain the temperature within this range.

15

2-4 Dimensions
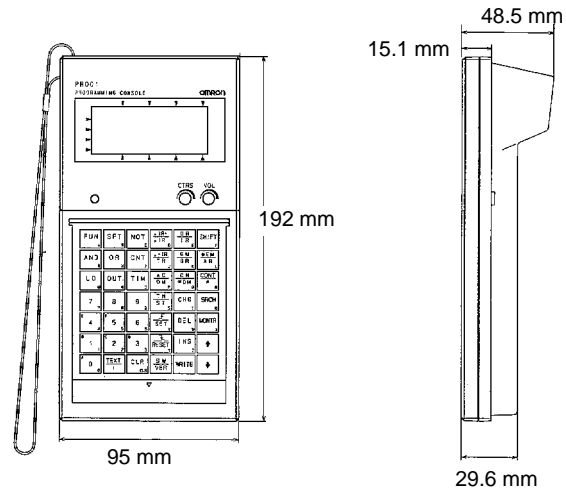
2-5 Precautions

Although the Programming Console does not require regular maintenance, observe the following precautions.

• Do not subject the Programming Console to excessive shock during transportation or operation. Handle the keypad and the display with care.

• Do not remove the System Memory Cassette.

• The Programming Console incorporates two connectors. Use only one of them at a time.

• Connect the cable to the Programming Console securely so that the lock levers of the Programming Console lock firmly into place.

• Do not pull or twist the cable with excessive force.

SECTION 3
Key Functions and Displays

This section describes the basic functions of the keys on the keypad and the main displays that appear during operation.

3-1 Keypad

The Programming Console's keypad is shown below. The use of individual keys is described in *Section 4 Operation* and in *Appendix D Inputs for Instructions and Operands.*

Note On the keys, "CONT" stands for "Bit" and "CH" stands for "Word." Also, the "SFT" Key is used to input the SHIFT instruction only. The "SHIFT" Key is used to toggle in and out of the shift input mode (see below).

FUN A	SFT B	NOT C	, IR+ , IR D	DR IR E	SHIFT F
AND G	OR H	CNT I	, −IR TR J	EM GR K	*EM AR L
LD M	OUT N	TIM O	AC DM P	CH *DM Q	CONT # R
7 7	8 8	9 9	TN ST S	CHG T	SRCH U
E 4 4	F 5 5	6 6	SET V	DEL W	MONTR X
B 1 1	C 2 2	D 3 3	RESET Y	INS Z	↑
A 0 0	TEXT !	CLR CLR	SW VER	WRITE	↓

Input Modes

There are three input modes used from the keypad: Normal input mode, Shift input mode, and Alphanumeric input mode.

Normal Input Mode
In normal input mode, the middle or the bottom function of each key is input when the key is pressed.

Shift Input Mode
When the Shift Key is pressed before another key, the function shown at the top or in the upper left-hand corner of the second key is input when the key is pressed.

Alphanumeric Input Mode
The alphanumeric input mode is a special mode used to input memory card file names or data area abbreviations for custom data areas. Press the Shift–Text Keys to enter Alphanumeric Input Mode.
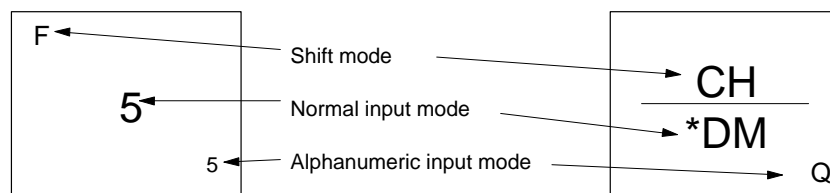
⚠ **Caution** The Programming Console cannot use custom abbreviations for data areas created on the CVSS or SSS except for abbreviations for custom data areas.

Mode Displays

The current input mode is indicated in the upper right-hand corner of the display. If there is nothing displayed, the Programming Console is in normal input mode.

SHIFT → [A   ] ← Shift mode mark
Toggled when the Shift Key is pressed.

SHIFT | TEXT ! → [:   ] ← Alphanumeric input mode mark
Return to the normal input mode by pressing the Clear Key.
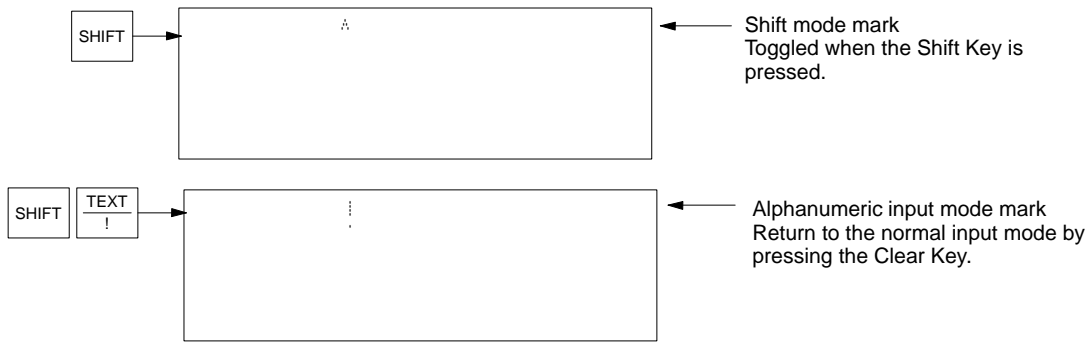
3-2 Initial Displays

The Programming Console is equipped with a series of initial displays that are used to set basic operating parameters and initiate operation. These displays and the key strokes to move between them are shown below. Operational details are provided in *Section 4 Operation*.

Initial Display: Settings

The following display will appear on the Programming Console after the Programming Console is turned on and connected to the PC. This display is used to set basic operating parameters for the Programming Console. Refer to *4-4-1 System Settings* for the setting methods and details on the response monitor time and message number.

```
CVM1 PRS21 V1 20
Mon time:10sec
Message #: 0
Read Setup Yes
```
→ Model/Version
→ Response monitor time
→ Message number
→ Read/don't read setting

CLR   SHIFT | CLR
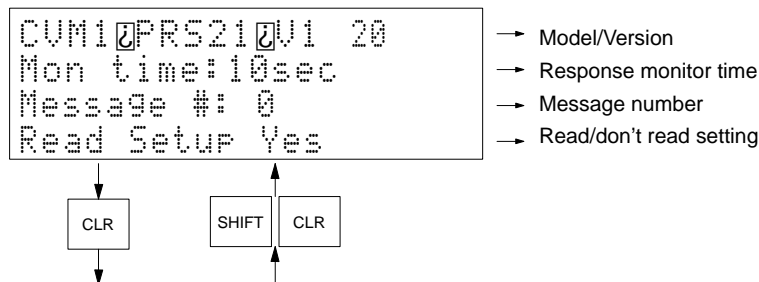
Program Designation Display The second display that appears shows the initial values of the program name to be read and the PC that the Programming Console is connected to. Refer to *4-4-2 Program Designation* to change the program to be read.

```
CVM1 CPU01 V2
Ladder
Main
```
→ Connecting PC
→ Program type
→ Program being read

CLR   SHIFT | CLR
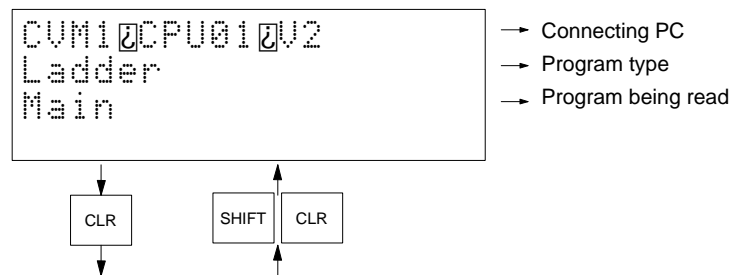
⚠ **Caution** Use version 20 or later of the Programming Console for the CVM1D. Earlier versions do not support online Unit replacement, and will not display error messages unique to the CVM1D.

Programming Display

The third display that appears shows the program address and the current PC operating mode.

Error Display:
"Cannot Connect"

The following display will appear and Programming Console operation will not be possible in the following cases.

• When the PC (including Expansion I/O Racks) is not turned ON.

• When the baud rate is not 50 kbps.

• When the number of connected Programming Devices exceeds the permissible number.

• When the Peripheral Service Disable Bit (A01704) is ON.

```
CVM1 PRS21 V1 20



Cannot connect
```
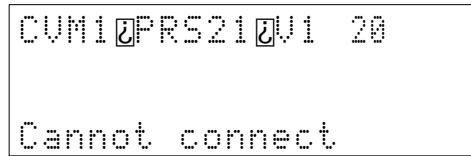
3-3   Sub-operations

Some Programming Console operations are executed directly from the main display for the programming operation and others are executed as sub-operations. The main display is used for main editing purposes (for example, programming). Sub-operations are used for PC mode selection, simple I/O monitor, etc. The overall structure of the initial displays, main display, and the sub-operations is shown below.
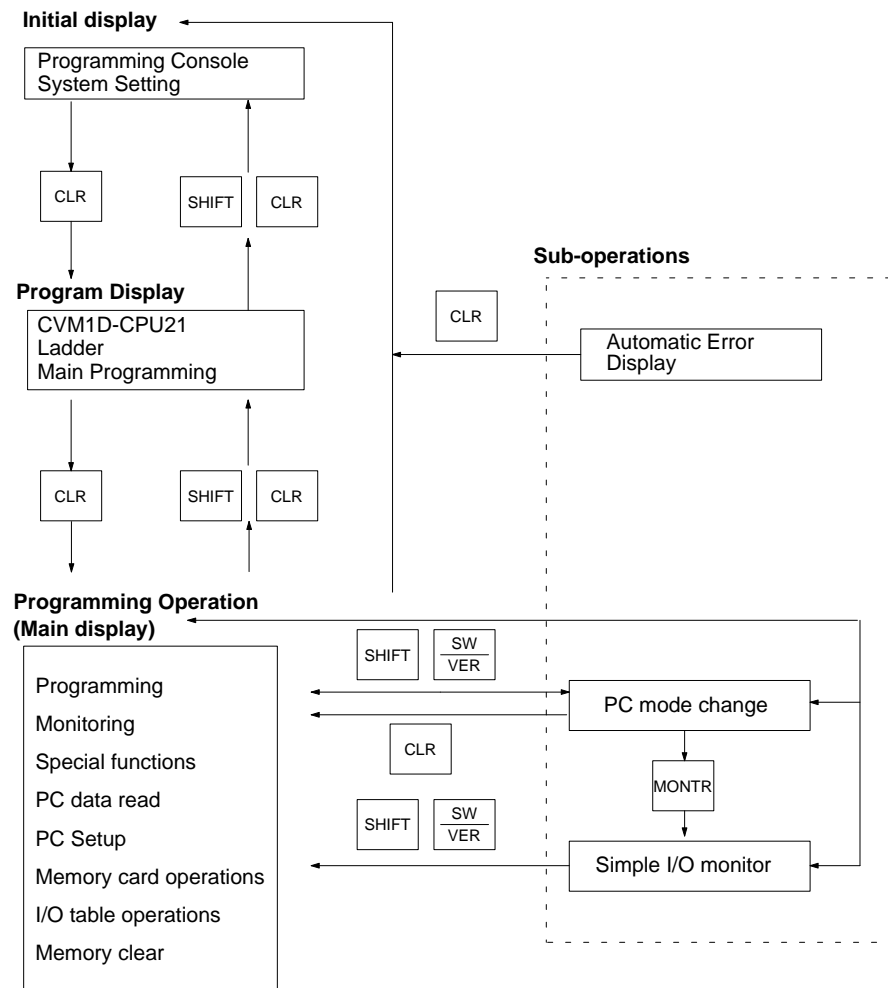
Processing on the main display will be interrupted if the display is changed to a sub-operation. If the main display is returned to from the sub-operation, the previous process will continue unless the mode has been changed using the PC mode change operation.

If an error occurs, an error message will be displayed automatically regardless of the current operation. When an error message is displayed, press the Clear Key to restore Programming Console operation. If the error is not fatal, the operation can be continued. Otherwise the initial display will appear.

3-4 Display Symbols and Abbreviations

The following symbols and abbreviations appear on the Programming Console.

PC Operating Modes
PRG: PROGRAM mode
DBG: DEBUG mode
MON: MONITOR mode
RUN: RUN mode

Instructions
!: Immediate refresh instruction (Do not use in CVM1D Duplex mode.)
=: Differentiate up instruction
O: Differentiate down instruction
N: NOT (The NOT(010) instruction is displayed as NOT.)

I/O Displays
I: Input bits
Q: Output bits
H: Hold bits

Data Areas
G: CPU Bus Link Area
A: Auxiliary Area
T: Timer Area
C: Counter Area
D: Data Memory (DM) Area
E: Expansion Data Memory (EM) Area
*D: Indirect DM address
*E: Indirect EM address
ST: Step
AC: Action
TN: Transition
TR: TR Area
IR: Index registers
DR: Data registers

Monitoring
ON: ON bit status
OFF: OFF bit status
O: ON flag status (TIM/CNT/steps)
Φ: OFF flag status (Steps)
S: Forced-set bit (ON)
R: Forced-reset bit (OFF)
▓: Temporary forced set or reset is being executed.

3-5 Display Examples

Program Read Displays

```
000000 PRG Read
!LD◊    I000000
```
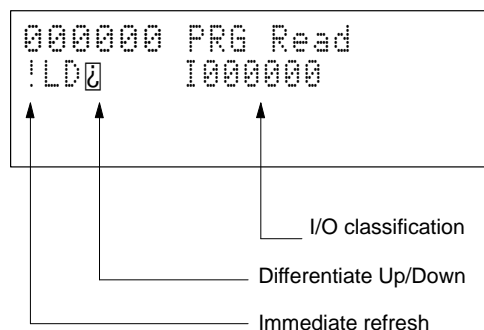
Note Differentiate Up/Down status is displayed after the instruction mnemonic for Ladder Diagram Instructions.

I/O classification – – – – (I : Input bit
                              Q : Output bit

Differentiate Up/Down – – – – (◊ : Differentiate up
                                ◊ : Differentiate down

Immediate refresh

```
000010 PRG Read
!↑MOV      1200H
(050)      D000000
```

⎯⎯⎯⎯⎯ Hold bit

⎯⎯⎯⎯⎯ Differentiate Up/Differentiate Down status

⎯⎯⎯⎯⎯ Immediate refresh

Note Differentiate Up/Down status is displayed before the instruction mnemonic for applications instructions.
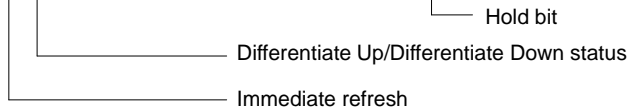
```
000002 PRG Read
AND  N  100000
```

⎯⎯⎯⎯⎯ NOT

Monitor Displays
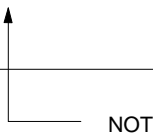
⎯⎯⎯⎯⎯ Completion Flag status

```
000030 OMON Read
TIM     0000
```

O : ON   (Displayed for the bit/flag on the top line.)
Φ : OFF

```
I000105    SON
```

⎯⎯⎯⎯⎯ Forced status   ----⎧ S : Force-set (ON)
                          ⎩ R : Force-reset (OFF)

```
T0015  ▓00000
```

⎯⎯⎯⎯⎯ Present value
⎯⎯⎯⎯⎯ Completion Flag status   ---- O : ON
⎯⎯⎯⎯⎯ Temporary forced status   Φ : OFF

```
ST0025    0E 0235
```
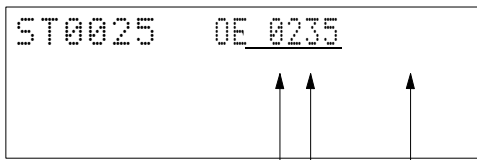
          └──── The present value of the step timer

        └──── Step status
        E: EXECUTE, P: PAUSE, H: HALT, I: INACTIVE
                      ◻: ON
     └──── Step Flag status   ◻: OFF

```
D01234    ABCD
```

                └──── Hexadecimal display

```
E01234    "JE"
```
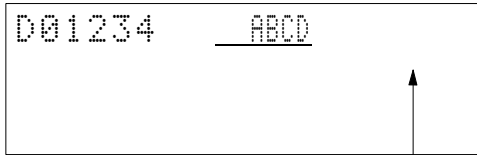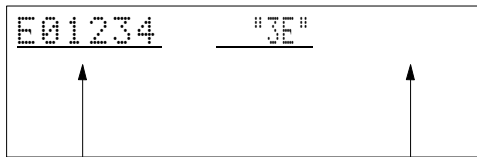
                └──── ASCII display
    └──── Address in current bank

```
E2 12345   +01234
```
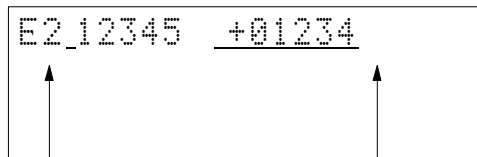
    └──── Bank number display   └──── Decimal display

```
IRI       8000
```
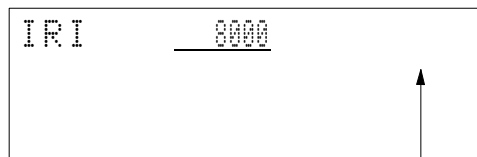
             └──── Present value

SECTION 4
Operations

This section individually describes the operations that are possible from the Programming Console. Key sequences for inputting ladder-diagram instructions and operands are provided in *Appendix D Inputs for Instructions and Operands*.

4-1 Overview

This section explains how to use each operation of the Programming Console and the things that must be taken into consideration when operating the Programming Console.

The following table lists all the operations that are explained in this section.

Classification	Operation	Page
Initial settings	System settings	27
	Program designation	29
	PC mode change	31
Program editing	Address set	32
	Program read	33
	Instruction search	35
	Operand search	36
	Instruction overwrite	38
	Instruction insert	39
	Instruction delete	41
	Timer/Counter SV change	42
Monitoring	Simple I/O monitor	44
	I/O multipoint monitor	48
	4-word monitor	51
	Word monitor	53
	Differential monitor	55
	Force set/reset	56
	PV change	59
	ASCII display	59
	Decimal display	62
Special operations	Clock read/change	63
	UM protect	65
	Active step monitor (See note 1)	67
	SFC search (See note 1)	68
	Action block change (See note 1)	71
	Transition change (See note 1)	74
PC data read	Cycle time display	75
	Automatic error display	76
	Error display/clear	79
	Error history read	80
	Message read	81
PC setup	Startup mode	82
	Execution controls: Comparison error process	84
	Host link	85
	Scheduled interrupt (See note 2)	87
	Momentary power interruption time (See note 2)	88
	Cycle time	89
	Cycle time monitoring time	90
Memory card operations	Memory card format	92
	File write	93
	File read	97
	File compare	99
	File delete	100

Classification	Operation	Page
I/O table operations	I/O table create	102
	I/O table compare	104
Memory clear	Memory clear	107
Online Unit replacement (See note 3)	Online Unit replacement	111

Note   1. Not possible with the CVM1 or CVM1D.

2. Setting not possible with the CVM1D.

3. Supported only by the CVM1D.

4-2   Layout of Descriptions

Operations are explained under the following headings.

Use
Explains the purpose of the operation.

Modes
Tells which PC operating modes the operation is supported in.

Key Sequence
Provides the basic key sequence for the operation.

Operation
Describes the operation through one or more examples.

Remarks
Provides additional information on operation.

Precautions
Lists points of importance in use of the operation.

4-3   Inputting Instructions and Operands

The key sequences used to specify instructions and operand data areas are provided in *Appendix D Inputs for Instructions and Operands*.

4-4   Operations

The key sequences and operation examples shown in this section start from either the initial Programming Console display or from a display resulting from a related operation. If the operation does not work as expected, make sure you are inputting from the proper starting point. Press the Clear Key to return to the initial display. You may have to press the Clear Key more than once.

4-4-1   System Settings

Use   Used to set the response monitor time and message number.

Modes

RUN	MONITOR	DEBUG	PROGRAM
OK	OK	OK	OK

Key Sequence
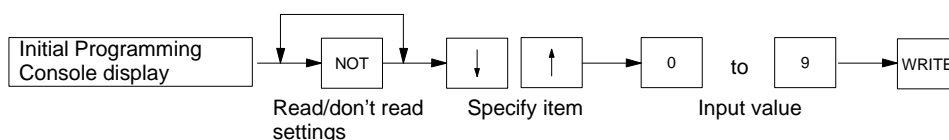
Operation   The input display for the response monitor time and message number is the initial display produced when the Programming Console is turned on. In this example, a response monitor time of 15 s and message number 1 are set.

```
CVM1 PRS21 V1 20
Mon time:  10sec
Message #: 0
Read Setup Yes
```
→ Model/Version
→ Response monitor time
→ Message number
→ Read/don't read PC Setup

| 1 | 5 |

Sets a response monitor time of 15 s.

| ↓ |

Moves the cursor to the message number.

| 1 |

Sets message number 1.

| WRITE |

Writes the values that have been set.

```
CVM1 PRS21 V1 20
Mon time:  15sec
Message #: 1
Read Setup Yes
```

Remarks   **Read/Don't Read Setup**

The NOT Key can be pressed to toggle between reading and not reading certain information from the PC. If this information is read when the Programming Console is connected to a Remote I/O Slave Unit, at least 30 s will be required to read the information before operation can be continued. If this information is not read, the program read operation will be entered immediately.

If Programming Console operation is begun without reading the above information from the PC, the following Programming Console functionality will be disabled.

• "I," "Q," and "H" will not be displayed to indicate I/O bits and words.

• "H" will not be displayed to indicate holding bits and words.

• All custom settings, including custom data areas and function codes, will be disabled and addresses must be input using their normal default values.

Response Monitor Time
• Setting range: 2 to 60 s.

• An error message will appear on the bottom of the display if there is no response from the PC within the response monitor time that has been set. The error message will disappear and the previous display will return the moment the Programming Console receives a response. Refer to *Section 5 Error Processing* for details on error messages.

Message Number
• Setting range: 0 to 7

• The contents of the programmed message will automatically appear when a MESSAGE instruction (MSG(195)) for the designated message number is executed in the PC program.

4-4-2 Program Designation

Use   Used to designate the program to be read by the Programming Console. The Programming Console can read the following programs.

Ladder-diagram Programming Only
Main program
Interrupt programs 00 to 35

SFC + Ladder-diagram Programming
Action programs
Transition programs

Modes

RUN	MONITOR	DEBUG	PROGRAM
See note.	OK	OK	OK

Key Sequence   The following key sequence is used when only ladder-diagram programming is used in the PC.
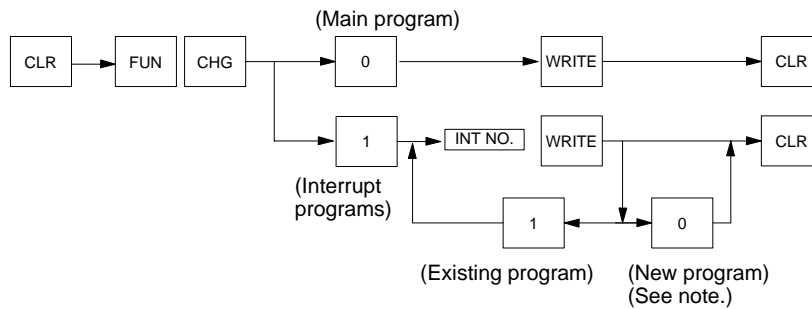
The following key sequence is used when SFC programming is being used.
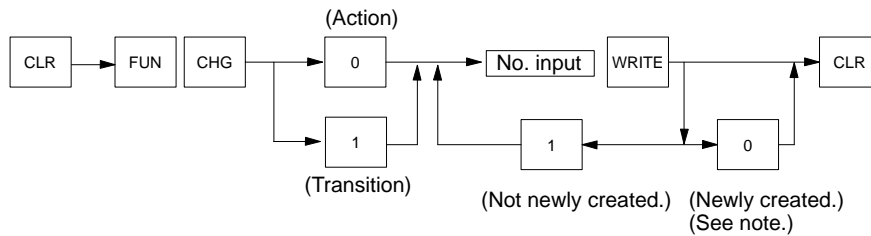
Note New creation not possible in RUN mode.

Remarks   The following displays are used to designate the program.

```
Ladder    Main
Seect program
0:Main program
1:Interrupts
```

```
SFC+LaddAC0000
Seect programs
0:Action
1:Transition
```

A display like the following one will appear if the program that has been designated does not exist.

```
INT12  missing
Create
0:Yes
1:No
```

Input 0 to create a new program. Input 1 to abort.

The END(001) instruction will be automatically created when a program is newly created. Use the instruction insert and instruction overwrite operations to complete inputting the program (refer to *4-4-9 Instruction Insert*).

Operation

The following example shows how to read interrupt program 01 when only ladder-diagram programming is being used.

| CLR | Call the programming operation. |

```
000000 PRG
```

| FUN | CHG | Call the program designation operation. |

```
Ladder    Main
Seect program
0:Main program
1:Interrupts
```

| 1 | Designate the interrupt program. |

```
00 31:IO   Main
32 33:Sce INT  _
34:Poer o
35:Poer on
```

| 1 | Set I/O interrupt number 01. |

```
00 31:IO Main
32 33:Sce INT01_
34:Poer o
35:Poer on
```

| WRITE | Execute by pressing the Write Key. |

30

```
Ladder      INT01
Seect program
0:Main program
1:Interrupts
```

[CLR]

The programming operation will be entered. If the specified program does not exist, a confirmation message will ask if you want to create the program. Refer to *Remarks* for details.

```
000000 PRG
```

4-4-3 PC Mode Change

Use This operation is not executed directly from the main display and instead executed as a sub-operation.

Used to control the PC's operating mode.

Note Make sure there will be no adverse effects on the control system before changing the PC's operating mode. Not doing so may result in injury.

Modes

RUN	MONITOR	DEBUG	PROGRAM
OK	OK	OK	OK

Key Sequence
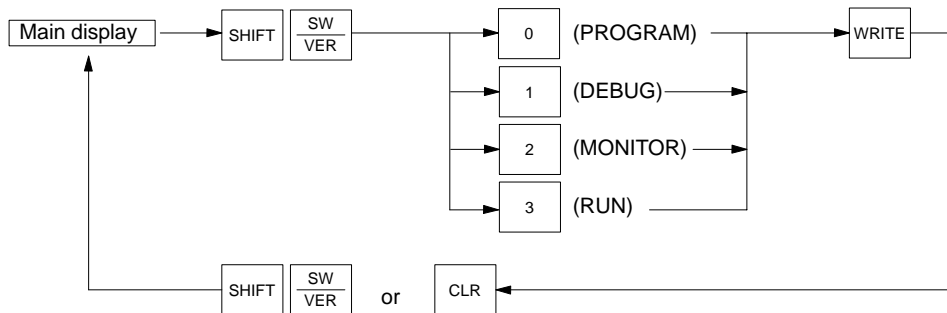
Operation In this example, the PC is switched to MONITOR.

Main display

[SW/VER] [WRITE] The mode menu will appear.

```
Mode     0:PRG
 PRG     1:DBG
         2:MON
         3:RUN
```

[2] Specify MONITOR mode.

31

```
Mode      0:PRG
 PRGMON  1:DBG
         2:MON
         3:RUN
```

↓

WRITE   Execute the mode change.

↓

```
Mode      0:PRG
 MON     1:DBG
         2:MON
         3:RUN
```

↓

CLR   The main display will appear when the Clear Key is pressed.

The Shift–SW Keys may also be used to return to the main display.

↓

Main display

PC Operating Modes

Operating mode	Function
PROGRAM (PRG)	Operation of the PC is stopped. Specify this mode when creating or altering the user's program or clearing memory.
DEBUG (DBG)	Specify this mode to confirm if the user's program and I/O data operate normally. The Programming Console, however, does not support DEBUG-mode operations.
MONITOR (MON)	The PC is in operation. I/O data is processed in this mode and in RUN mode. In MONITOR mode, it is also possible to monitor the PC operation, forcibly control data area contents (i.e., turn bits ON and OFF), change the set value or present value of timers and counters, and change the content of words. This mode is specified when adjusting the system in trial operation.
RUN (RUN)	RUN mode is used for normal PC operation. In this mode, the operation of the PC can be monitored, but it is impossible to forcibly turn bits ON and OFF or change present values or set values.

4-4-4 Address Set

Use   Used to set the program address used for reading, writing, inserting, deleting, and searching programs.

Modes

RUN	MONITOR	DEBUG	PROGRAM
OK	OK	OK	OK

Key Sequence

CLR → Input address

32
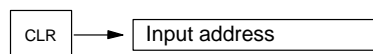
Operation   In this example program address 000123 is set.

CLR   Call the programming operation.

```
000000 PRG

```

1   2   3   Set the address.

```
000123 PRG

```

Remarks   It is not necessary to input leading zeros of the address, e.g., just input 1, 2, and 3 to set 000123.

4-4-5 Program Read

Use   Used to read the contents of a program.

Modes

RUN	MONITOR	DEBUG	PROGRAM
OK	OK	OK	OK

Key Sequence

Address set → ↓ / ↑
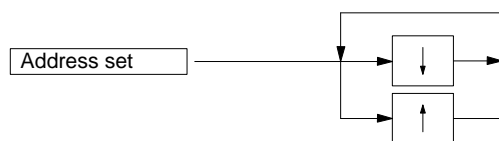
Operation   In this example, the contents of address 000123 is read.

CLR   1   2   3   Set address 000123.

```
000123 PRG

```

↓   Display the program for three lines beginning with address 000123.

```
000123 PRG  Read
LD      I000000
OR      120012H
AND N   Q001001
```
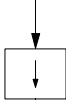
↓ Scroll downward one line with the Down Key. The address indicated in the display will change to the address of the instruction at the cursor.

```
000124 PRG  Read
OR      120012H
AND N   Q001001
OUT     Q000104
```
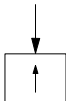
↑ Scroll upward one line with the Up Key.

```
000123 PRG  Read
LD      I000000
OR      120012H
AND N   Q001001
```

Remarks

Addresses will be scrolled downward or upward continuously when the Down Key or Up Key are pressed.

The message "Address too high" will appear when the Down Key is pressed after the last address has been reached.

```
00085  PRG  Read
OUT     0000001
END (001)
```

↓
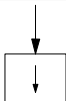
```
0008 PRG  Read
END(001)
```

↓

```
0008 PRG  Read


Address too 19
```

4-4-6 Instruction Search

Use Used to search for an instruction used in the program.

Modes

RUN	MONITOR	DEBUG	PROGRAM
OK	OK	OK	OK

Key Sequence

```
                              ┌──→ CLR ←──────┐
                              │               │
   Address set ──┐   ┌──→ Input instruction ──┼──→ SRCH ──┐
                 ├───┤                         │          │
   Program read ─┘   │                         ↑          │
                     └─────────────────────────┘          │
                              └───────────────────────────┘
```
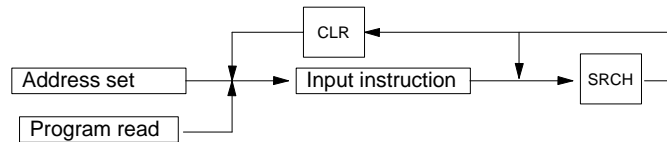
Operation In this example MOV(030) is searched starting with address 000030.

[3] [0] Set address 000030.

```
000030 PRG


```

[FUN] [0] [3] [0] Input the function code in full including leading zeros.

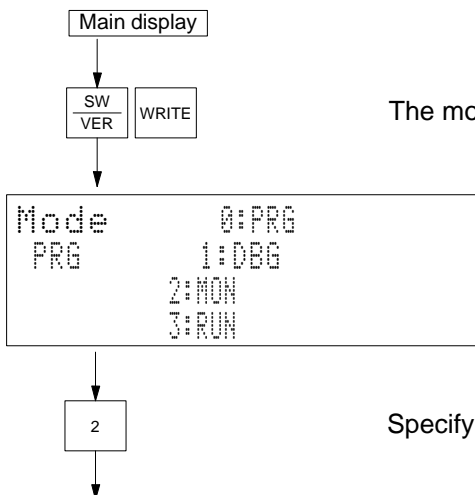[SRCH] The message "Busy" appears while the Programming Console is searching for the instruction.

```
000030 PRGSearc
MOV      0000
(030)    0000
Bus
```

The address, instruction, and operands for the first MOV(030) found beginning at address 000030 will be displayed.

```
000052 PRGSearc
MOV      0100
(030)    0200
Main
```

[SRCH] Press the Search Key to search for the next instruction.

```
          ┌──────┐
          │ SRCH │
          └──────┘
             │
             ▼
┌────────────────────────────────┐
│ 000030 PRGSearc                │
│ MOV      0000                   │
│ (030)    0000                   │
│ Instr not ound                 │
└────────────────────────────────┘
```

After all addresses have been searched, the display will return to the address where the search was started, and the message "Instr not found" will appear.

Remarks

All programs contained in the PC will be searched, e.g., it is impossible to search only the main program or only individual action programs.

When using only ladder programming, the following program names will appear:

Main program: Main

Interruption program: INT ** (interrupt number)

When using SFC programming, the following program names will appear:

Action programs: AC **** (action number)

Transition programs: TN **** (transition number)

Search Conditions

Searches are possible individually for instructions with and without NOT.

Searches are possible individually for different operands for instructions that have only one operand, for example, LD 000000 and LD N 000000 can be distinguished, as can LD 000001 and LD 000002. SBN 001 and SBN 002 can also be distinguished.

Individual searches can be made for instructions with and without immediate refresh specifications (!) and differential treatment (↓, ↑), for example, LD 000000 and ! LD 000000 are distinguished. LD 000000 and ↓LD 000000 are also distinguished.

Searches are possible individually for different operands for the following instructions: FAL(006), FALS(007), CNTR(012), TTIM(120), TCNT(123), TSR(124), TSW(125), and SA(210).

Individual searches are not possible for different operands for other instructions that have more than one operand, for example, MOV 0000 0001 and MOV 0002 0003 are not distinguished.

Refer to *Appendix D Inputs for Instructions and Operands* for the key sequences to specify instructions and operands.

Note The Up or Down Key can be pressed during the search operation to enter the program read operation.

Precautions

To cancel the present searching operation and to search for another instruction, press the Clear Key.

To search for any of the TIM, CNT, and NOT instructions while the program is being read, press the Clear Key first and then input the name of the instruction.

4-4-7 Operand Search

Use

Used to search for an operand in a program by designating a bit or word address.

Modes

RUN	MONITOR	DEBUG	PROGRAM
OK	OK	OK	OK

Key Sequence

```
                                                    ┌──── CLR ◄────┐
                                                    │              │
              ┌──────────────┐        │      ┌───────────────┐        │      ┌──────┐
              │ Address set  │────────┴─────►│ Input operand │────────┴─────►│ SRCH │──┐
              └──────────────┘               └───────────────┘               └──────┘  │
                                                                                       │
```
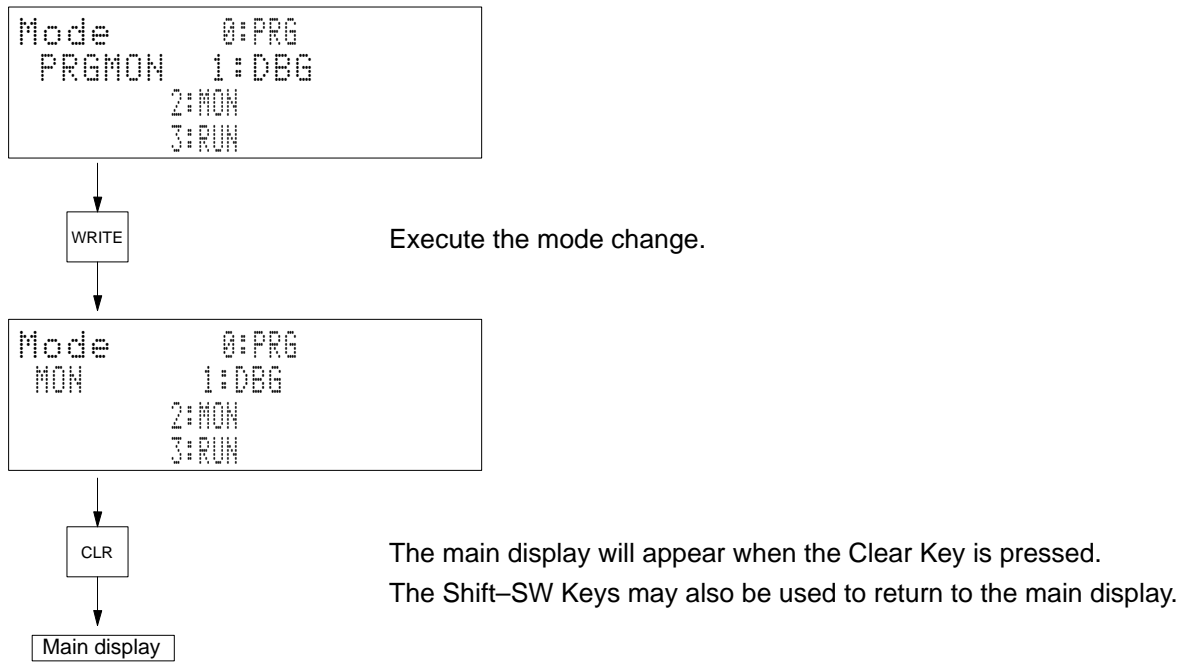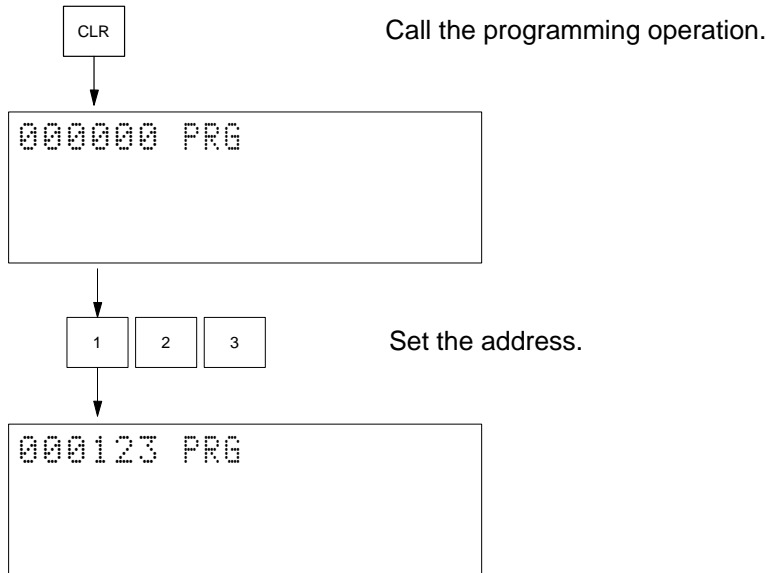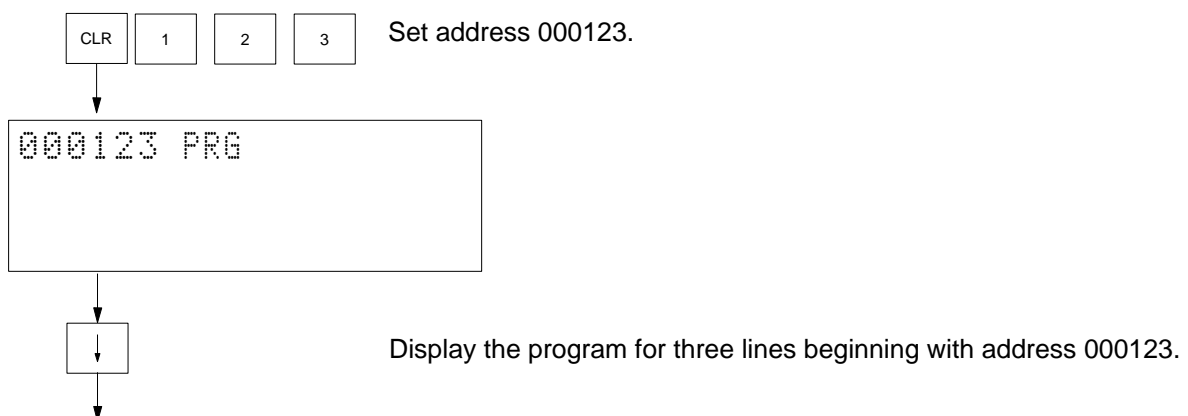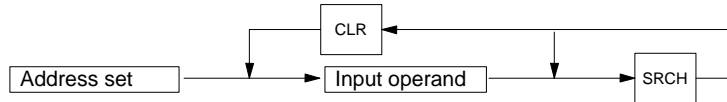
Operation   In this example CIO 001308 is searched for starting from program address 000012.

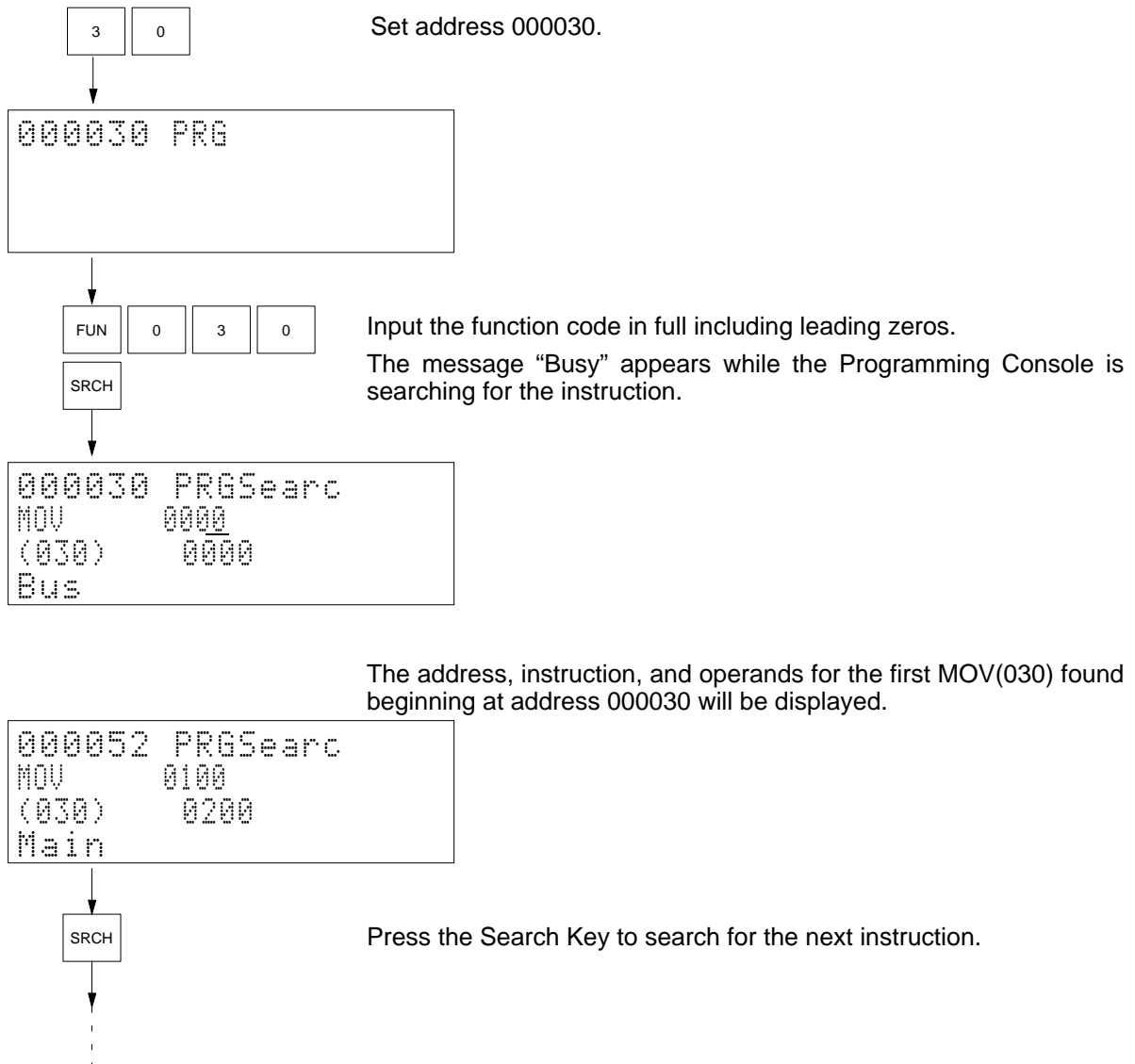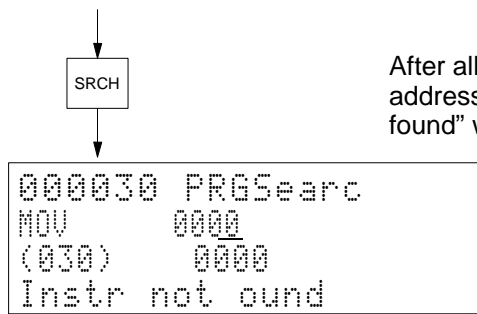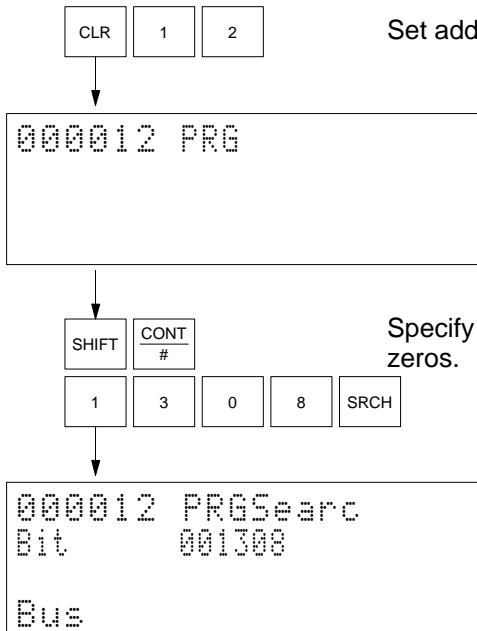┌─────┐ ┌───┐ ┌───┐   Set address 000012.
│ CLR │ │ 1 │ │ 2 │
└─────┘ └───┘ └───┘
   │
   ▼
```
┌─────────────────────────────┐
│ 000012 PRG                  │
│                             │
│                             │
│                             │
└─────────────────────────────┘
```
   │
   ▼
┌───────┐ ┌──────┐   Specify the bit to be searched for. It is not necessary to input leading
│ SHIFT │ │ CONT │   zeros.
└───────┘ │ #   │
          └──────┘
┌───┐ ┌───┐ ┌───┐ ┌───┐ ┌──────┐
│ 1 │ │ 3 │ │ 0 │ │ 8 │ │ SRCH │
└───┘ └───┘ └───┘ └───┘ └──────┘
   │
   ▼
```
┌─────────────────────────────┐
│ 000012 PRGSearc             │
│ Bit     001308              │
│                             │
│ Bus                         │
└─────────────────────────────┘
```

                                   The message "Busy" appears while the Programming Console is searching.
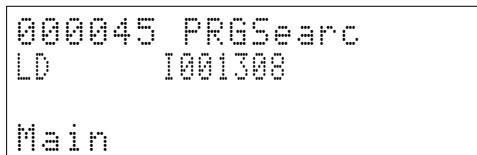
```
┌─────────────────────────────┐
│ 000045 PRGSearc             │
│ LD      I001308             │
│                             │
│ Main                        │
└─────────────────────────────┘
```
   │
   ▼
┌──────┐   The address, instruction, and program that includes the first use of CIO
│ SRCH │   001308 as an operand will be displayed.
└──────┘
   │   The Programming Console will search for the next occurrence of CIO
   ┊   001308 when the Search Key is pressed again.
   ┊
   ▼
┌──────┐   After all programs have been searched, the display will return to the
│ SRCH │   address where the search started, and the message "Oper not found"
└──────┘   will appear.
   │
   ▼
```
┌─────────────────────────────┐
│ 000012 PRGSearc             │
│ Bit     001308              │
│ Oper not ound               │
│                             │
└─────────────────────────────┘
```
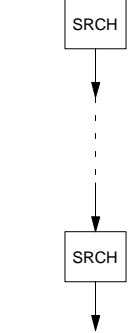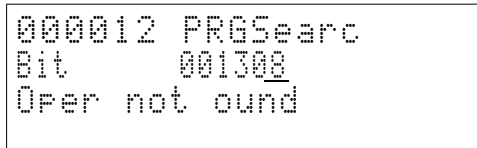
Remarks
All programs contained in the PC will be searched, e.g., it is impossible to search only the main program or only individual action programs.

When using only ladder programming, the following program names will appear:

Main program:	Main
Interruption program:	INT ** (interrupt number)

When using SFC programming, the following program names will appear:

Action programs:	AC **** (action number)
Transition programs:	TN **** (transition number)

Refer to *Appendix D Inputs for Instructions and Operands* for the key sequences to specify instructions or operands.

Note The Up or Down Key can be press during the search operation to enter the program read operation.

4-4-8 Instruction Overwrite

Use
Used to overwrite the instruction at a designated address in a program. If the designated address contains END(001), the END(001) will not be overwritten, but the new instruction will be inserted and END(001) will be moved to the next address. This operation cannot be used to add instructions where they did not exist before, e.g., after END(01).

Modes

RUN	MONITOR	DEBUG	PROGRAM
No	OK	OK	OK

Key Sequence

When editing on-line (in MONITOR mode or DEBUG mode)
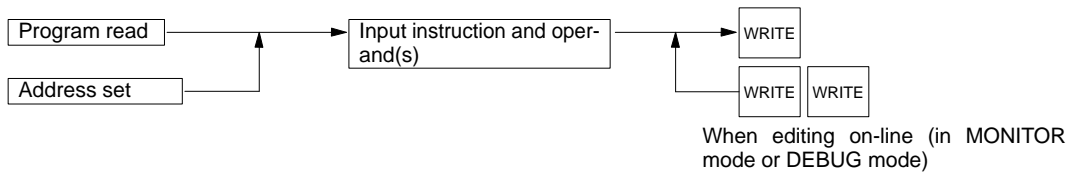
Operation
In this example, the instruction at address 000003 is replaced by MOV(030) D00010 0002.

Read the instruction currently at the address.

```
000003 PRG Read
CMP     D00100
(020)     0020
AND     150012H
```

Input MOV(030).

```
000003 PRG
MOV      0000
(030)     0000
```

Input the first operand. Press the Down Key to move the cursor to the second operand.
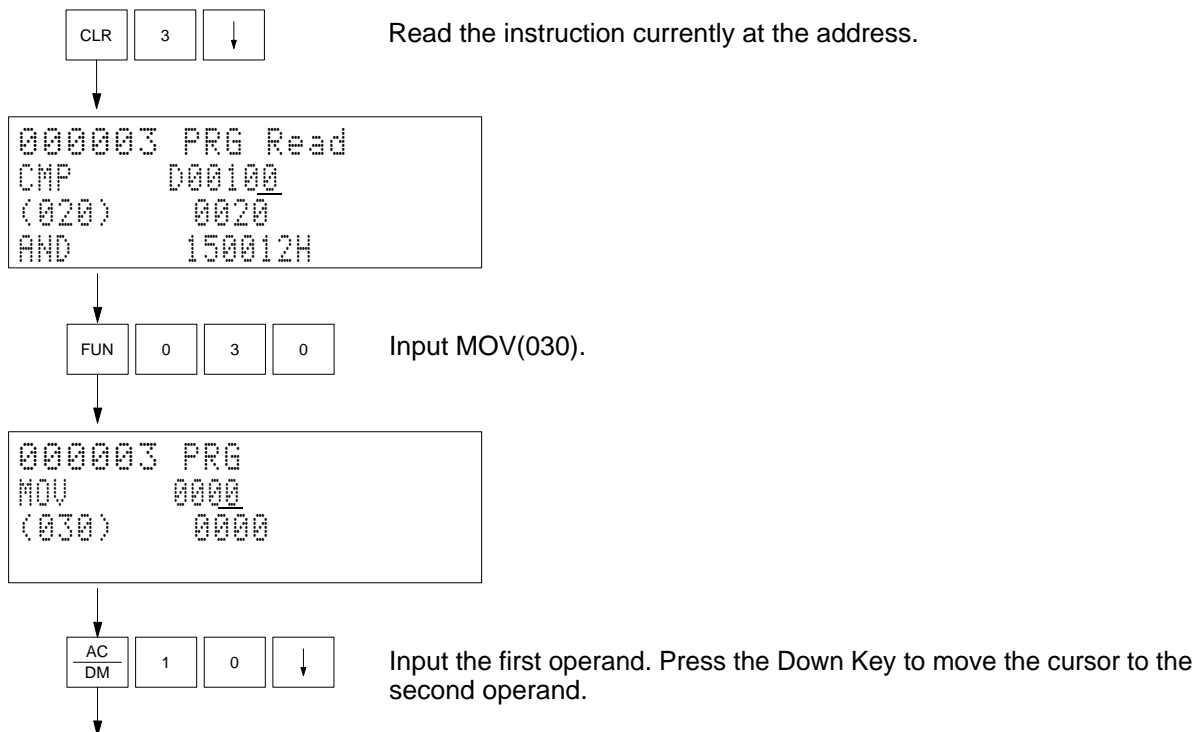
```
000003 PRG
MOV     D00010
(030)    0000
```

↓

```
 2
```

Input the second operand.

↓

```
000003 PRG
MOV     D00010
(030)    0002
```

↓

```
WRITE
```

Press the Write Key to write the instruction.

The next address will appear and program reading can be continued.

↓

```
000004 PRG Read
AND     150012H
OUT     0001001
LD N    100515
```

Remarks
The following display will appear if the Write Key is pressed in MONITOR or DEBUG mode.

```
000003 MON
MOV     D00010
(030)    0002
Onine edit
```

Press the Write Key to overwrite the current instruction.

or Press the Clear Key to abort.

The next address will appear after overwriting the instruction with the Write Key. Use the Up Key to return to the previous address to check instructions that have been rewritten.

Precautions
This operation overwrites the previous instruction (except for END(001), see *Use* for details). Use the operation described in the next section to insert instructions.

To write the TIM or CNT instruction when reading the program and the first operand for the current instruction is not a bit, press the Clear Key first and then input the instruction.

The message "Address too high" will appear if the user attempts to write an instruction at an address that currently has no instruction and the write will not be allowed.

Programs cannot be checked from the Programming Console. Use the GPC, CVSS/SSS, or SYSMAC-CPT.

4-4-9 Instruction Insert

Use
Used to insert an instruction at a designated address in a program. Instructions cannot be inserted after the last instruction.

Modes

RUN	MONITOR	DEBUG	PROGRAM
No	OK	OK	OK

Key Sequence

```
┌──────────────┐
│ Program read │──────────┐
└──────────────┘          │         ┌──────────────────────┐      ┌─────┐      ┌───┐
                          ├────────►│ Input instruction and│─────►│ INS │─────►│ ↓ │
┌──────────────┐          │         │ operand(s).          │      └─────┘      └───┘
│ Address set  │──────────┘         └──────────────────────┘
└──────────────┘
```
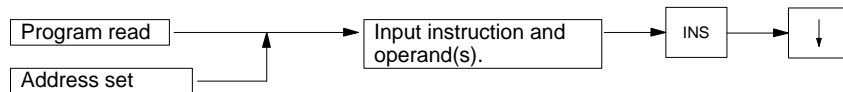
Operation   In this example, AND 000109 is inserted at program address 000021.

```
┌─────┐  ┌───┐  ┌───┐  ┌───┐
│ CLR │  │ 2 │  │ 1 │  │ ↓ │
└─────┘  └───┘  └───┘  └───┘
```
Read the address where the instruction is to be inserted.

```
000021 PRG Read
OR      150012H
OUT     0001001
LD N    100515
```

```
┌─────┐  ┌───┐  ┌───┐  ┌───┐
│ AND │  │ 1 │  │ 0 │  │ 9 │
└─────┘  └───┘  └───┘  └───┘
```
Input the instruction and operand.

```
000021 PRG
AND     00010_
```

```
┌─────┐
│ INS │
└─────┘
```
A confirmation display will appear when the Insert Key is pressed.

```
000021 PRG Ins
AND     00010
```

```
┌───┐
│ ↓ │
└───┘
```
Press the Down Key to insert the instruction at the current address and move the instruction currently at the address and all following instructions down one address.

```
000022 PRG Read
OR      150012H
OUT     0001001
LD N    100515
```

40

Remarks

The following display will appear if the Insert Key is pressed in MONITOR or DEBUG mode. Press the Down Key to insert the instruction at the current address.

```
000023 MOV Ins
MOV     A012
(030)   D01000
Onine edit
```

The next address will appear after inserting an instruction. Use the Up Key to return to the address to check the instruction that has been inserted.

Precautions

To insert the TIM or CNT instruction while reading the program and the first operand for the current instruction is not a bit, press the Clear Key first and then input the name of the instruction.

Programs cannot be checked from the Programming Console. Use the GPC, CVSS/SSS, or SYSMAC-CPT.

The message "Address too high" will appear if the user attempts to insert an instruction at an address where no instruction currently exists and the insertion will not be allowed.

4-4-10 Instruction Delete

Use

Used to delete the instruction at a designated address in a program.

Modes

RUN	MONITOR	DEBUG	PROGRAM
No	OK	OK	OK

Key Sequence

```
Program read  ────────┐
                      ├──→  DEL  →  ↑
Address set  ─────────┘
```
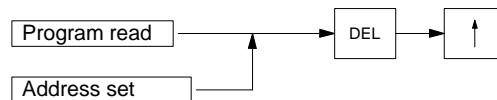
Operation

In this example, XCHG(034) at address 000301 is deleted.

| CLR | 3 | 0 | 1 |
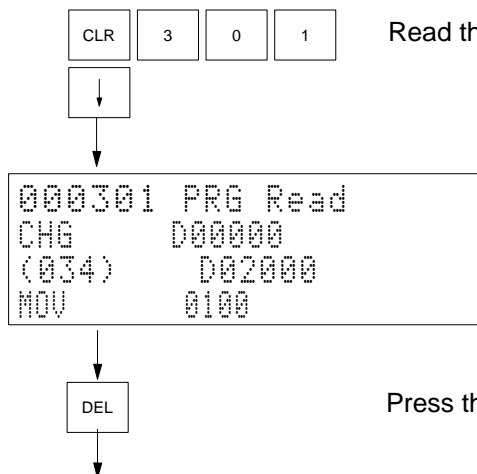
Read the instruction to be deleted.

↓

```
000301 PRG Read
CHG     D00000
(034)   D02000
MOV     0100
```

↓

DEL

Press the Delete Key. A confirmation display will appear.

↓

41

```
000301 PRG De
CHG     D00000
(034)   D02000
MOV     0100
```


Press the Up Key to delete the instruction. The succeeding instructions will be displayed.

```
000301 PRG Read
MOV     0100
(030)   D12345
AND     000000
```

Remarks The following display will appear if the Delete Key is pressed in MONITOR or DEBUG mode. Press the Up Key to delete the instruction.

```
000301 PRG De
CHG     D00000
(034)   D02000
Onine edit
```
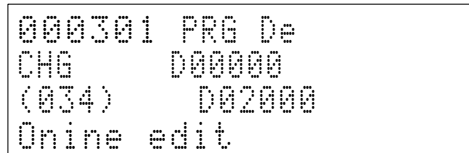
The next address will appear after deleting the instruction with the Up Key. Press the Up Key again to return to the previous address to make sure that the instruction was deleted.

4-4-11 Timer/Counter SV Change

Use Used to change the set value of a timer or counter.

⚠ **Caution** Make sure that there will be no adverse effects on the control system before changing a SV. Not doing so may result in injury.

Modes

RUN	MONITOR	DEBUG	PROGRAM
No	OK	OK	OK

Key Sequence
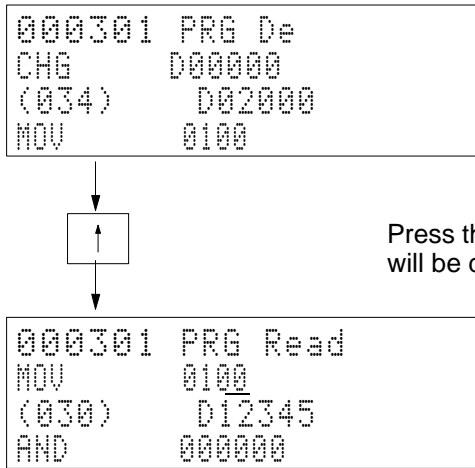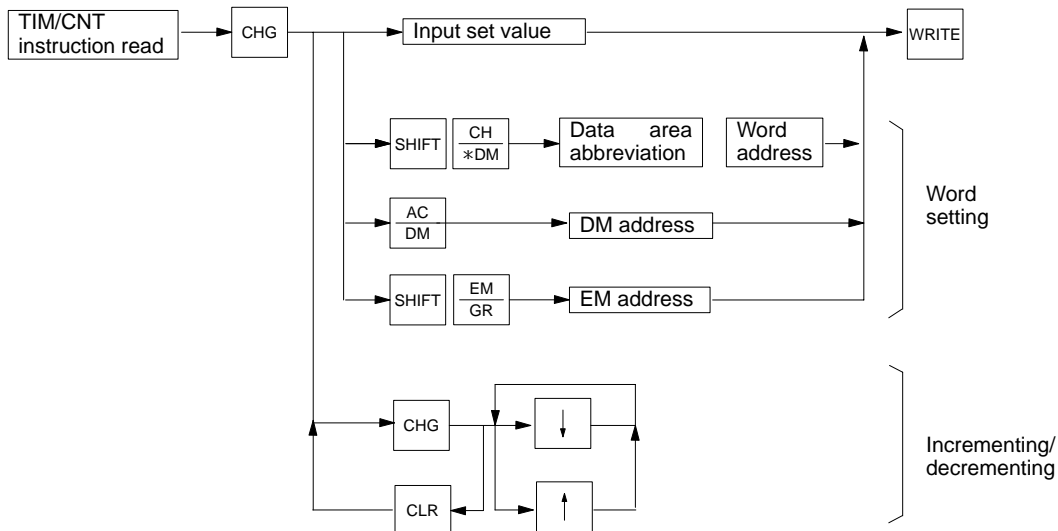
Operation

Two examples of operation are provided. The first example shows how to input a new SV. The second example shows how to increment/decrement an existing SV.

Example 1

In this example, TIM 0001's SV is changed from #0100 to #0085. First, read the TIM 0001 instruction to produce the following display.

```
00028  PRG Read
TIM      0001
     #0100
OUT     0001001
```
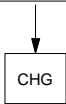
CHG | Call the SV change operation.

```
00028  Cg SV
TIM      0001
     #0100
     #
```
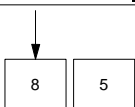
8 | 5 | Input the new SV.
To specify a word address, input the desired address.

```
00028  Cg SV
TIM      0001
     #0100
     #0085
```

WRITE | Press the Write Key to write the data and return to the read operation.

```
00028  PRG Read
TIM      0001
     #0085
OUT     0001001
```

Example 2

In this example, TIM 0001's SV is incremented from #0100 to #0106.

```
00028  PRG Read
TIM      0001
     #0100
OUT     0001001
```
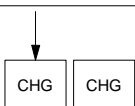
CHG | CHG | Specify incrementing/decrementing.

```
00028   Ad SV
TIM      0001
        #0100
(⬚⬚)   #0
```
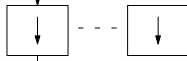
The value will increase or decrease by one when the Up or Down Key is pressed once. Press the Down Key six times to get #0106.

```
00028   Ad SV
TIM      0001
        #0100
(⬚⬚)   #010
```

[CLR] [CLR] Press the Clear Key twice to return to the read operation.

```
00028   PRG Read
TIM      0001
        #010
OUT      001001
```

Remarks It is possible to change the SV for the TIM, TIMH(015), TTIM(120), TIML(121), CNT, CNTR(012), and TCNT(123) instructions with this operation.

Incrementing and decrementing can be performed only for constants.

4-4-12 Simple I/O Monitor

Use Used to monitor memory status for bits, words, timers (PV), counters (PV), etc. Also used to change word contents or force-set or force-reset bits.

This operation is executed as an interrupt operation (sub-operation) for operations performed from the main display. The process that was interrupted can be continued after returning to the main display.

Modes

RUN	MONITOR	DEBUG	PROGRAM
See note.	OK	OK	OK

Note Although monitoring is possible, it is not possible to change contents or force set or force reset status in the RUN mode.

Key Sequence
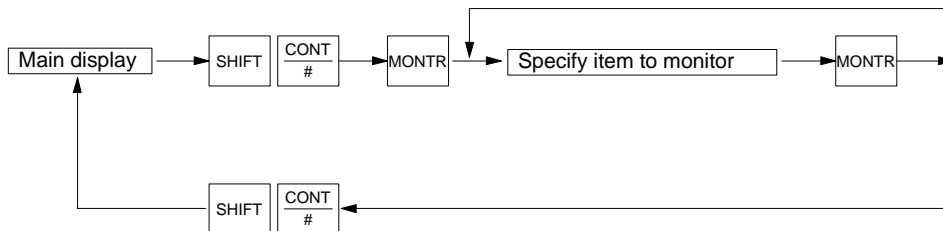
Main display → SHIFT → CONT # → MONTR → Specify item to monitor → MONTR →

SHIFT → CONT # ←

Key Functions
Keys are used as described below.
Monitor Key: Used for one-line scroll.

Up and Down Keys: Used to increment or decrement the word and bit address on the first line.

Clear Key: Used to delete the first line.

Change Key, (Data), Write Key: Used to change the PV on the first line.

Set/Reset Keys: Used to temporary force-set or force-reset the first line (while the keys are pressed).

Shift–Set/Shift–Reset Keys: Used to force set or force reset the first line.

NOT Key: Used to release the force-set or force-reset status on the first line.

Monitor Displays

The following can appear on monitor displays to indicate status.

ON: ON condition of a bit.

OFF: OFF condition of a bit.

O: ON condition of a flag (TIM/CNT/step).

Φ: OFF condition of a flag (step).

S: Force-set bit.

R: Force-reset bit.

▓:   Temporary force-set or force-reset
(appears while Set or Reset Key is pressed).

Remarks   Up to 8 items can be input for monitoring, out of which, 4 will be displayed at any one time.

An arrow will be displayed in front of status or word contents to indicate you are in the simple I/O monitor operation and not in the I/O multipoint monitor operation.

Refer to *Appendix D Inputs for Instructions and Operands* for the key sequences to specify instructions and operands.

Operation   Three examples of monitoring are given below. The first one shows how to monitor multiple items; the second one shows how to change the value of a word in memory, and the third one shows how to control bit status while monitoring.

Example 1

In this example the following five points are monitored.

     TIM 0200

     CIO 0030

     DM 05000

     CIO 000101

     CNT 0011
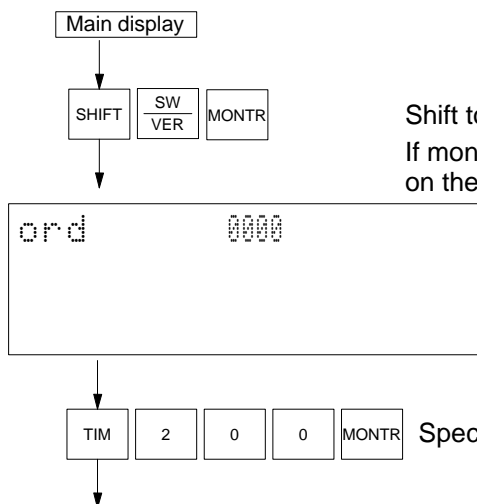
Shift to the simple I/O monitor operation.
If monitor items have been previously designated, they will appear on the monitor display.

Specify TIM 0200 for monitoring.

```
T0200      0250
```

Inputs for remaining
four items.

Specify the remaining 4 items for monitoring. Up to 8 items can be specified.

```
C0011      0500
I000101     OFF
D05000     2A5F
 0030      12
```

↓   ↓   ↓

To increment or decrement the word and bit address on the first line, press the Up or Down Key. To specify a different address in the same area for the item on the first line, you only need to input the address and not the area abbreviation.

```
C0014      0125
I000101     OFF
D5000      2A5F
 0030      12
```

MONTR

Press the Monitor Key to scroll one line. If more than four items are being monitored, the bottom line will move to a memory buffer and the next one held in memory will be moved onto the display.

```
T0200      0030
C0014      0125
I000101     OFF
D05000     2A5F
```

CLR

Press the Clear Key to delete the item on the first line.

```
C0014      0125
I000101     OFF
D05000     2A5F
 0030      12
```

Example 2

In this example the PV of CNT 0014 is changed to #0020. Only the item on the first line can be changed.

```
C0014      0125
I000101     OFF
D05000     2A5F
 0030      12
```

Call the change operation.

```
C0014      0125
Cange          ‾
I000101     OFF
D05000     2A5F
```

Input the new PV and press the Write Key to execute the change.

```
C0014      0020
I000101     OFF
D05000     2A5E
 0030      12
```

Example 3

In this example, CIO 000101 is temporary force-set, force-reset, and then the forced status is cleared

```
I000101     OFF
D05000     2A5F
 0030      12
C0014      0020
```

Press the Set Key to temporarily force-set the bit. ▓ will be displayed while the Set Key (or Reset Key) is pressed and the bit is force-set. When the key is released, the force-set status is canceled and the natural ON/OFF status of the bit is displayed.

```
I000101    ▓ON
D05000     2A5F
 0030      12
C0014      0020
```

Press the Shift–Set Keys to force-set the bit. S will be displayed and the bit will turn ON and remain ON until forced status is cleared by pressing the NOT Key or by globally clearing all forced status. The S will change to R if the Shift–Reset Key is pressed to force-reset the bit.

```
I000101    SON
D05000     2A5F
  0030     12
C0014      0020
```

↓
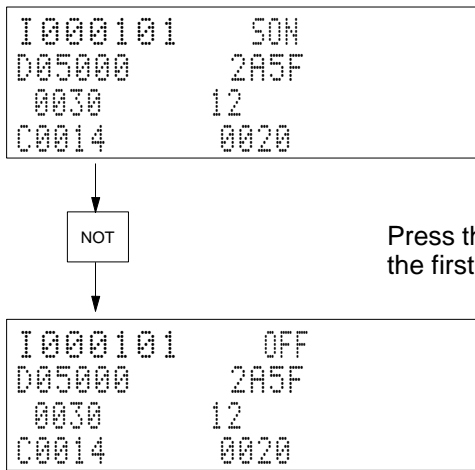
NOT   Press the NOT Key to release the forced status of the bit monitored on
                       the first line.

↓

```
I000101    OFF
D05000     2A5F
  0030     12
C0014      0020
```

4-4-13 I/O Multipoint Monitor

Use   Used to monitor memory status for bits, words, timers, counters, etc.

                       This operation is executed from the programming operation.

Modes

RUN	MONITOR	DEBUG	PROGRAM
OK	OK	OK	OK
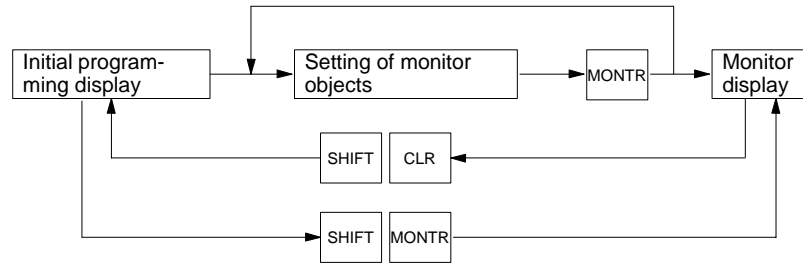
Key Sequence

Key Functions

Keys are used as described below.

> **Monitor Key:** Used for one-line scroll.

> **Up and Down Keys:** Used to increment or decrement the word and bit address on the first line.

> **Clear Key:** Used to delete the first line.

Monitor Displays

The following can appear on monitor displays to indicate status.

> ON: ON condition of a bit.

> OFF: OFF condition of a bit.

> �‚:   ON condition of a flag (TIM/CNT/step).

> ◊:   OFF condition of a flag (step).

Remarks   Up to 8 items can be input for monitoring, out of which, 4 will be displayed at any
                       one time.

                       An arrow will **not** be displayed in front of status or word contents to indicate you
                       are in the I/O multipoint monitor operation and not in the simple I/O monitor
                       operation.

                       Refer to *Appendix D Inputs for Instructions and Operands* for the key sequences
                       to specify instructions and operands.

The Shift–Clear Keys can be pressed to return to the program edit operation. The items being monitored will be recorded. Once this has been done, the Shift–Monitor Keys can be pressed to resume monitoring the items that were recorded in memory.

The following operations can be executed from with the I/O multipoint monitor operation:

4-word monitor, word monitor, force-set/reset, PV change, ASCII display, decimal display, and differential monitor

Operation   In this example the following five points are monitored.

TIM 0200

CIO 0030

DM 05000

CIO 000101

CNT 0011

```
000000 MON

```


Specify TIM 0200 for monitoring.

```
T0200       0250

```

Inputs for remaining four items

Specify the remaining 4 items for monitoring. Up to 8 items can be specified.

```
C0011       0500
I000101      OFF
D05000      2A5F
 0030        12
```


To increment or decrement the word or bit address on the first line, press the Up or Down Key. To specify a different address in the same area for the item on the first line, you only need to input the address and not the area abbreviation.
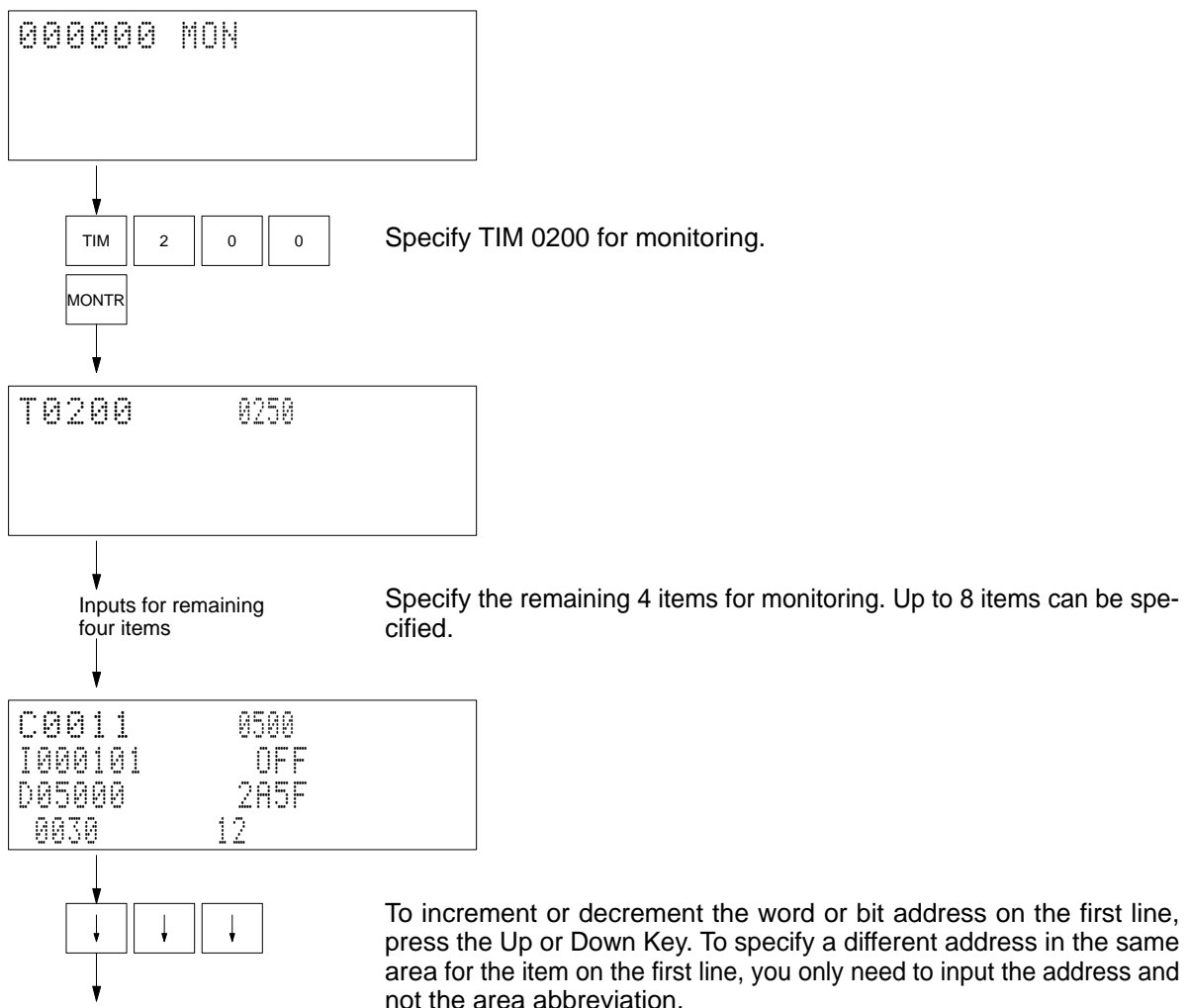
```
C0014      0125
I000101     OFF
D05000     2A5F
 0030      12
```

↓

[MONTR]

Press the Monitor Key to scroll by one line.

↓

```
T0200      0032
C0014      0125
I000101     OFF
D05000     2A5F
```

↓

[CLR]

Press the Clear Key to delete the first line.

↓

```
C0014      0125
I000101     OFF
D05000     2A5F
 0030      12
```

↓

[SHIFT] [CLR]

Press the Shift–Clear Keys to return to the programming operation, at which time, all items being monitored will be held in memory.

↓

```
000000 MON
```
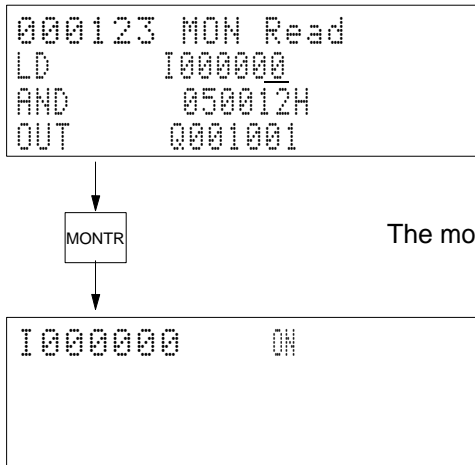
↓

[SHIFT] [MONTR]

Press the Shift–Monitor Keys again to resume the monitoring operation with the items held in memory.

↓

```
C0014      0125
I000101     OFF
D05000     2A5F
 0030      12
```

Remarks   The monitor operation can be entered by pressing the Monitor Key from the pro-
gram read operation.

```
000123 MON Read
LD      I000000
AND     050012H
OUT     Q001001
```

┌─────────┐
│ MONTR │   The monitor display will appear showing the word or bit on the first line.
└─────────┘

```
I000000     ON
```

Precautions   It is possible to return to the monitor operation from the initial programming dis-
play by pressing the Monitor Key; however, all the items memorized will be lost.
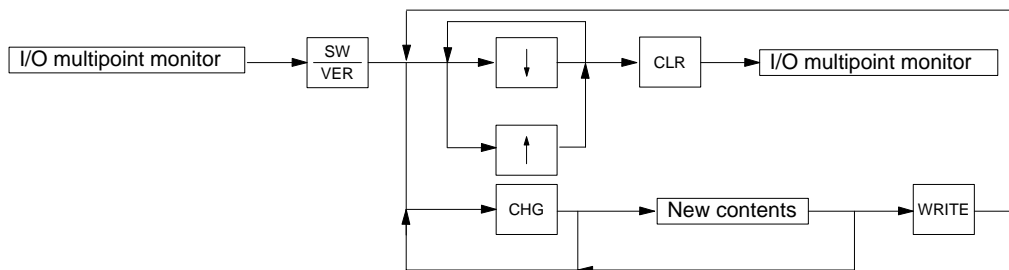
4-4-14 Four-word Monitor

Use   Used to monitor four consecutive words of data. It is possible change the con-
tents of addresses being monitored.

Modes

RUN	MONITOR	DEBUG	PROGRAM
See note.	OK	OK	OK

Note Although monitoring is possible, it is not possible to change memory contents in
the RUN mode.

Key Sequence

Operation   In this example, counters from CNT 0014 to CNT 0017 are monitored. This
operation is entered from the I/O multipoint monitor operation, so first CNT 0014
must be monitored on the first line.

```
C0014      0125
I000101     OFF
D05000     2A5F
 0030       12
```
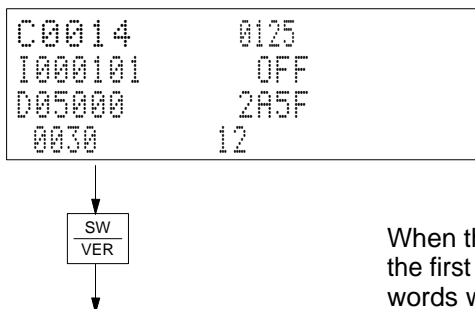
┌─────────┐
│   SW   │
│   VER   │   When the four-word monitor operation is entered, the word that was on
└─────────┘   the first line of the I/O multipoint monitor display and the following three
                  words will be displayed (here, CNT 0014 to CNT 0017).

51

```
C001        0100
C001        0000
C0015       0052
C0014       0125
```

↑   ↑   ↑

It is possible to increment or decrement the words being monitored by pressing the Up or Down Key.

```
C1014       0125
C0013       0040
C0012       0000
C0011       0084
```

CHG

The PV change operation can be entered directly from the four-word monitor operation. Here, the value of CNT 0013 is changed to 0060.

```
C0014      _0125
C0013       0040
C0012       0000
C0011       0084
```

CHG

The item that can be changed will move down one line each time the Change Key is pressed.

```
C0014       0125
C0013      _0040
C0012       0000
C0011       0084
```

6   0   WRITE
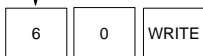
Input the new value.

```
C0014       0125
C0013       000
C0012       0000
C0011       0084
```

CLR

Press the Clear Key to return to the I/O multipoint monitor operation, or press the Shift–Clear Keys to return to the main display.

```
C0011       0004
I000101      OFF
D05000      2A5F
 0030        12
```

Remarks   The items being monitored are recorded in memory when the Shift–Clear Keys are pressed. After recording monitor items in memory, the Shift–Monitor Keys can be pressed to recall the same items to the monitor display. If the Monitor Key is pressed after recalling monitor items stored in memory, the initial monitor display will be returned to and the items stored in memory will be displayed.

When monitoring in ASCII, the display will change to hexadecimal the first time the Change Key is pressed.

When monitoring in decimal, contents cannot be changed.
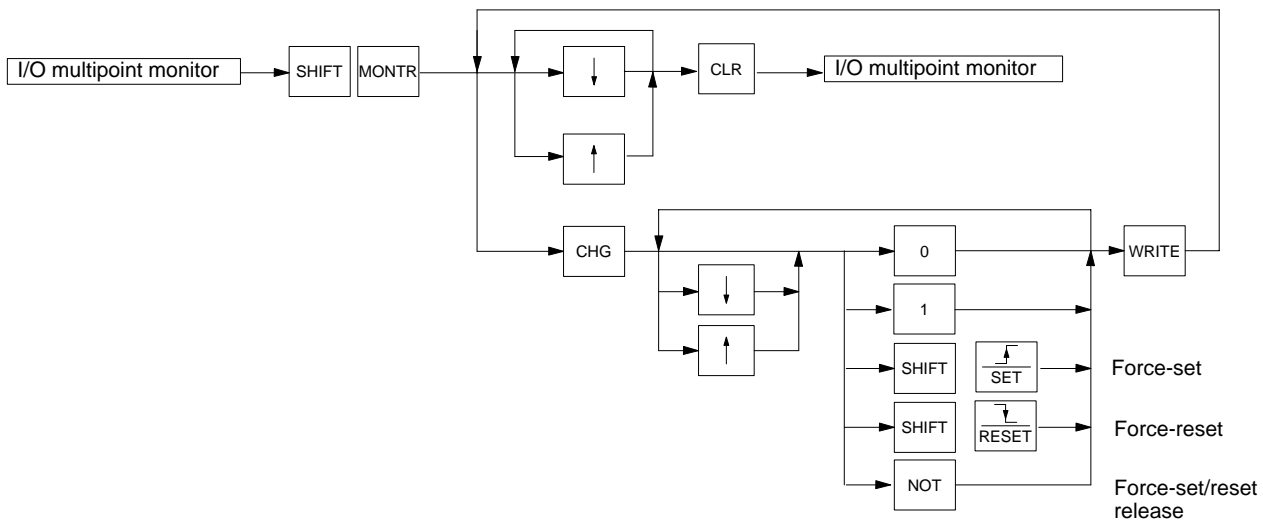
4-4-15 Word Monitor

Use   Used to monitor word contents in 16 digits of binary. Also used to set, reset, force-set, and force-reset individual bit status.

Modes

RUN	MONITOR	DEBUG	Monitoring
See note.	OK	OK	OK

Note   Although monitoring is possible, it is not possible to change memory contents or force-set/force-reset bits in the RUN mode.

Remarks

Monitor Displays
The following can appear on monitor displays to indicate status.

1:      ON condition of a bit.

0:      OFF condition of a bit.

S: Force-set of a bit or flag.

R: Force-reset of a bit or flag.

Precautions   This operation cannot be use to monitor timers and counters.

Bits in the following areas can be force-set or force-reset.

CIO Area (Core I/O Area):   0000 to 2555

CPU Bus Link Area: G008 to G255

Operation   In this example, output word CIO 0000 is monitored and then the monitor is
                                       change to CIO 0001 and bit status changed.

```
Q0000      8B3F
A25501     OFF
0000000    ON
I050012H   OFF
```

 ↓

[SHIFT] [MONTR]   Enter the word monitor operation from the I/O multipoint monitor
                                       operation with CIO 0000 on the first line.

 ↓

```
Q0000      Read
1000101100111111
A25501     OFF
0000000    ON
```
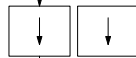
 ↓

[↓]   It is possible to increment or decrement the word address by pressing
                                       the Up or Down Key.

 ↓

```
Q0001      Read
0001000000S110R1
A25501     OFF
0000000    ON
```

 ↓

[CHG]   The change operation can be entered directly from the word monitor
                                       operation. Here, bit 13 of CIO 0001 is force-set.

 ↓

```
Q0001      Cg
0001000000S110R1
A25501     OFF
0000000    ON
```

 ↓

[↓] [↓]   Press the Up or Down Key to move the cursor.

 ↓

```
Q0001      Cg
0001000000S110R1
A25501     OFF
0000000    ON
```

 ↓
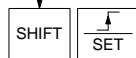
[SHIFT] [SET]   Press the Shift–Set Keys to force-set the bit where the cursor is
                                       located.
                                       The 0, 1, Shift–Reset Keys can also be input to force-set/reset bits.

 ↓

```
Q0001      C9
00S1000000S110R1
A25501        OFF
0000000       ON
```
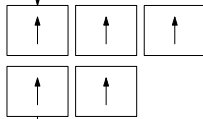
↑ ↑ ↑   Move the cursor to bit 01.

↑ ↑

```
Q0001      C9
00S1000000S110R1
A25501        OFF
0000000       ON
```

NOT   Press the NOT Key to clear the force-set or force-reset status of the bit where the cursor is located.

```
Q0001      C9
0S10000000S11001
A25501        OFF
0000000       ON
```

WRITE   Press the Write Key to write the changed data and return to the word monitor operation.

```
Q0001      Read
00S1000000S11001
A25501        OFF
0000000       ON
```

CLR   Press the Clear Key to return to the I/O multipoint monitor operation, or press the Shift–Clear Keys to return to the main display.

```
Q0001      303
A25501        OFF
0000000       ON
I050012H      OFF
```

Remarks   The items being monitored are recorded in memory when the Shift–Clear Keys are pressed. After recording monitor items in memory, the Shift–Monitor Keys can be pressed to recall the same items to the monitor display. If the Monitor Key is pressed after recalling monitor items stored in memory, the initial monitor display will be returned to and the items stored in memory will be displayed.
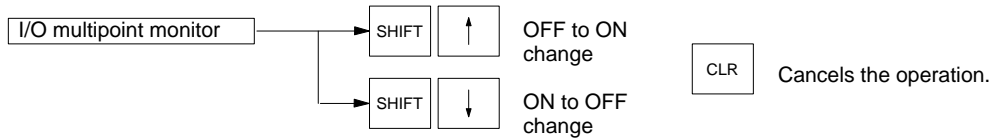
4-4-16 Differential Monitor

Use   Used to detect the OFF-to-ON or ON-to-OFF change in a designated bit. The Programming Console will beep when a change in the status of the designated bit is detected.

Modes

RUN	MONITOR	DEBUG	PROGRAM
OK	OK	OK	OK

Key Sequence

```
┌─────────────────────────┐         ┌───────┐ ┌───┐
│ I/O multipoint monitor   │────┬──▶│ SHIFT │ │ ↑ │   OFF to ON
└─────────────────────────┘    │   └───────┘ └───┘   change
                               │                              ┌─────┐
                               │                              │ CLR │   Cancels the operation.
                               │   ┌───────┐ ┌───┐            └─────┘
                               └──▶│ SHIFT │ │ ↓ │   ON to OFF
                                   └───────┘ └───┘   change
```

Operation In this example, input bit CIO 000002 is monitored to detect when it turns ON.

```
I000002     OFF
D05000      8B3F
D00000      8080
A25501      OFF
```

```
┌───────┐ ┌───┐
│ SHIFT │ │ ↑ │
└───────┘ └───┘
```

Enter the differential monitor operation. An arrow will appear on the display to indicate that the Programming Console is waiting for a change in bit status. The operation can be canceled by pressing the Clear Key.

```
I000002   █  OFF
D05000      8B3F
D00000      8080
A25501      OFF
```

The Programming Console will beep when CIO 000002 turns ON.

```
I000002      ON
D05000      8B3F
D00000      8080
A25501      OFF
```

Remarks It is possible to monitor the following areas with this operation:
CIO Area (CIO 000000 to CIO 255515)
CPU Bus Link Area (G)
Auxiliary Area (A)
Timer Area (T)
Counter Area(C)
Action Flag Area (AC)
Transition Flag Area (TN).

4-4-17 Force-set/Reset

Use Used to control the ON/OFF status of bits/flags in memory. Also used to release forced status for individual bits or all bits at once.
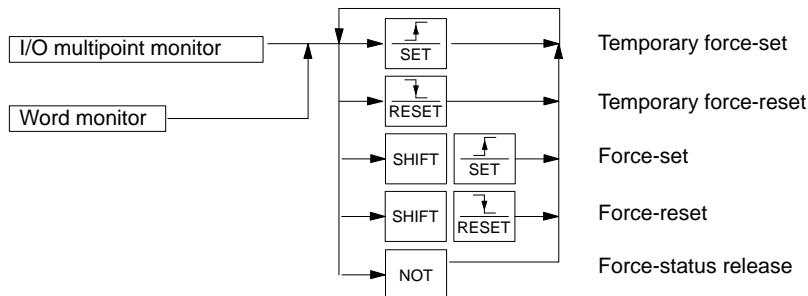
⚠ **Caution** Make sure that there will be no adverse effects on the control system before force-setting or force-resetting bits. Not doing so may result in injury.

Modes

RUN	MONITOR	DEBUG	PROGRAM
No	OK	OK	OK

Key Sequence

Global Forced Status Release

Note a) Press the Shift–Clear Keys to return to the programming operation.

b) The buzzer will beep when the Set–Reset–NOT Keys are pressed to indicate that forced status has been cleared. It does not indicate an error in operation.

Monitor Displays
The following can appear on monitor displays to indicate status.

ON: ON condition of a bit.

OFF: OFF condition of a bit.

O: ON condition of a flag (TIM/CNT/step).

Φ: OFF condition of a flag (step).

S: Force-set bit.

R: Force-reset bit.

▓:   Temporary force-set or force-reset
        (appears while Set or Reset Key is pressed).

Operation

It is possible to force-set or force-reset bits in the following data areas with this operation.

CIO Area (Core I/O Area): 000000 to 255515

CPU Bus Link Area: G00800 to G25515

Timer (T) and Counter (C) Areas

Example 1
In this example, CIO 000101 is temporary force-set, force-set, and then the force-set status is released. Start by monitoring CIO 000101 on the top line of the display.

```
I000101     OFF
D05000      2A5F
 0030       12
D0014       0020
```
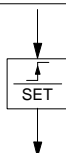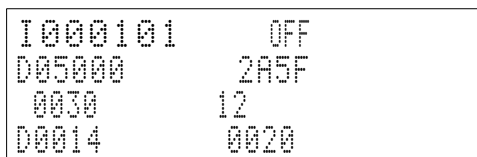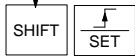

Press the Set Key to temporary force-set the bit. ▓ will be displayed. Forced status will remain in affect only as long as the key is held down. When the key is released, the natural status of the bit will return.

```
I000101   ▓ON
D05000      2A5F
 0030      12
C0014      0020
```

SHIFT SET

Press the Shift–Set Keys to force-set the bit. S will be displayed and the bit will remain ON until forced status is released by pressing the NOT Key or by globally releasing forced status. The S will change to R if the Shift–Reset Key is pressed to force-reset the bit.

```
I000101    SON
D05000      2A5F
 0030      12
C0014      0020
```

NOT

Press the NOT Key to release the force-set or force-reset status of the bit monitored on the first line.

```
I000101     OFF
D05000      2A5F
 0030      12
C0014      0020
```

Example 2

In this example, forced status is globally released. Start by pressing the Shift–Clear Keys to return to the programming operation.

```
000000 MON


```

SET RESET

Press the Set–Reset Keys. A confirmation display will appear and the buzzer will beep to prompt for a response. This is not an indication of an error in operation.

```
000000 MON

Force reease
```

NOT

Press the NOT Key to execute. All forced status will be released.

```
000000 MON

Reease End
```

4-4-18 PV Change

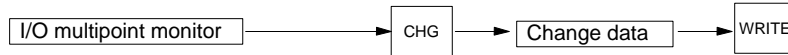**Use**   Used to change the content of a word that is being monitored.

⚠ **Caution**   Make sure that there will be no adverse effects on the control system before changing a PV. Not doing so may result in injury.

Modes

RUN	MONITOR	DEBUG	PROGRAM
No	OK	OK	OK

Key Sequence

I/O multipoint monitor ⟶ CHG ⟶ Change data ⟶ WRITE

Operation   In this example, PV C0014 is changed to #0020.

```
C0014      0125
I000101     OFF
D05000     2A5F
 0030      12
```
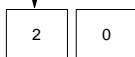
⬇

CHG   Call the PV change operation.

⬇

```
C0014      0125
Cange
I000101     OFF
D05000     2A5F
```

⬇

2   0   Input a new value.

⬇

```
C0014      0125
Cange     0020
I000101     OFF
D05000     2A5F
```

⬇

WRITE   Press the Write Key to execute.

⬇

```
C0014      0020
I000101     OFF
D05000     2A5F
 0030      12
```
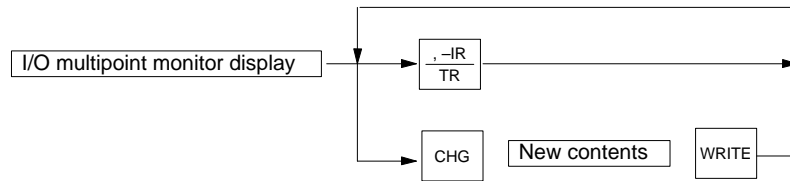
4-4-19 ASCII Display

Use   Used to display data in ASCII. It is possible to change data while displaying in ASCII.
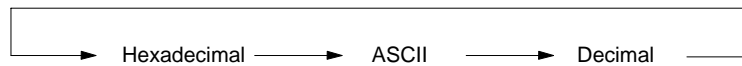
59

Modes

RUN	MONITOR	DEBUG	PROGRAM
See note.	OK	OK	OK

Note Although monitoring is possible, it is not possible to change values in the RUN mode.

Key Sequence

```
┌──────────────────────────────┐    ┌─────────┐
│ I/O multipoint monitor display│───▶│  , –IR  │──────────────────────────▶
└──────────────────────────────┘ │  │   TR    │
                                  │  └─────────┘
                                  │
                                  │  ┌─────┐  ┌──────────────┐  ┌───────┐
                                  └─▶│ CHG │  │ New contents │  │ WRITE │
                                     └─────┘  └──────────────┘  └───────┘
```

The form of data display will change in sequence each time the TR Key is pressed.

```
┌──────────────────────────────────────────────────────────┐
│    ──▶  Hexadecimal  ──▶  ASCII  ──▶  Decimal  ───────────┘
```

Operation

In this example, the contents of D01001 is display in ASCII and then changed to 3142.

```
D01001      325A
D05000      28
A25501      OFF
0000000     OFF
```

↓

┌─────────┐
│ , –IR │
│   TR   │
└─────────┘

Press the TR Key to display the data on the first line in ASCII.

↓

```
D01001      "2"
D05000      28
A25501      OFF
0000000     OFF
```

↓

┌─────┐
│ CHG │
└─────┘

Input the new value in hexadecimal.

┌───┐ ┌───┐ ┌───┐ ┌───┐
│ 3 │ │ 1 │ │ 4 │ │ 2 │
└───┘ └───┘ └───┘ └───┘

↓

```
D01001  "2"325A
Cange "1B"3142 _
D05000      28
A25501      OFF
```

↓

┌───────┐
│ WRITE │
└───────┘

Press the Write Key to execute. The contents will be changed and displayed in ASCII.

↓

60

```
D01001      "1B"
D05000      28
A25501      OFF
Q0000000     OFF
```

Remarks   It is possible to display data from the following areas in ASCII.

CIO Area (CIO 0000 to CIO 2555)

CPU Bus Link Area (G)

Auxiliary Area (A), DM Area

EM Area

Step timers

Index registers (IR)
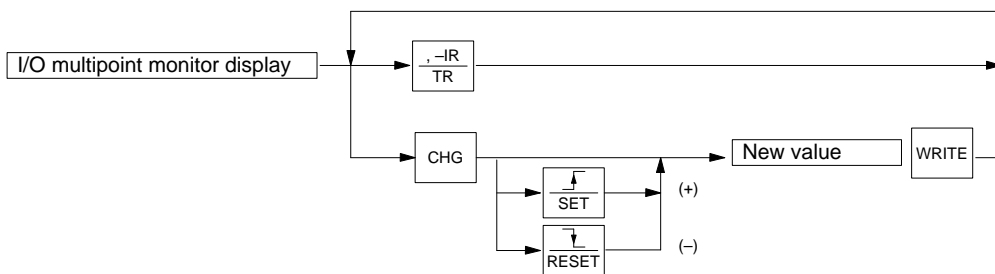
Data registers (DR)

4-4-20 Decimal Display

Use

Used to display the contents of a displayed word in signed decimal. It is possible to change the contents in decimal.
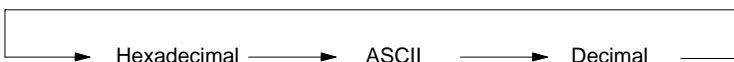
Modes

RUN	MONITOR	DEBUG	PROGRAM
See note.	OK	OK	OK

Note Although monitoring is possible, it is not possible to change word content in RUN mode.

Key Sequence

The form of data display will change in sequence each time the TR Key is pressed.


Remarks

The signed decimal value is determined by using the bit 15 as the sign bit (0 for + and 1 for –). For example, if 003A is the hexadecimal value, the signed-decimal display will be +00058, and if FFFF is the value, it will be –00001.

It is possible to display data from the following areas in ASCII.

CIO Area (CIO 0000 to CIO 2555)

CPU Bus Link Area (G)

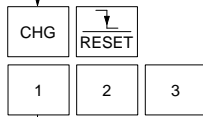Auxiliary Area (A), DM Area

EM Area

Operation

In this example, the contents of E00123 is displayed in signed decimal and then changed to –00123.
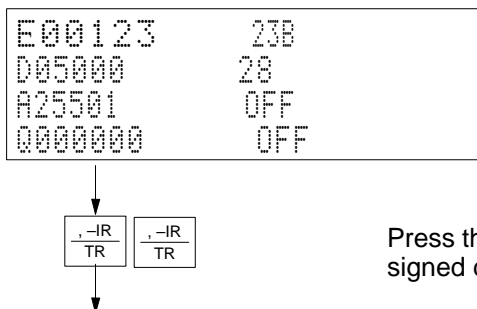
Press the TR Key to display the contents of the word on the first line in signed decimal.

```
E00123      +00
D05000       28
A25501      OFF
0000000     OFF
```

↓

| CHG | ⊥ / RESET |

| 1 | 2 | 3 |

Input the new value.

↓

```
E00123      +00
CanGe      ₫00123
D05000       28
A25501      OFF
```

↓

| WRITE |

Press the Write Key to execute.

↓

```
E00123      ₫00123
D05000       28
A25501      OFF
0000000     OFF
```

↓

| , −IR / TR |

Return to hexadecimal display by pressing the TR Key.

↓

```
E00123      FF85
D05000       28
A25501      OFF
0000000     OFF
```
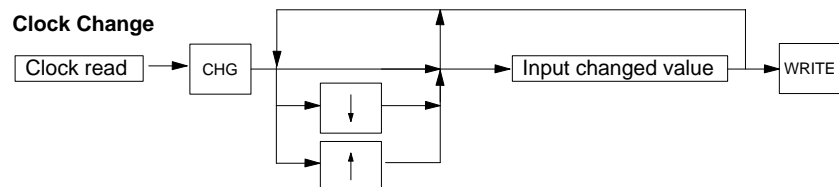
4-4-21 Clock Read/Change

Use   Used to display and change the time, day, and date of the PC's clock.

Modes

RUN	MONITOR	DEBUG	PROGRAM
OK	OK	OK	OK

Key Sequence

Clock Read | CLR | → | FUN | SHIFT | MONTR | → | 0 |

Clock Change

| Clock read | → | CHG | → | Input changed value | → | WRITE |

with ↓ and ↑ branch keys.

Remarks   The following keys are used to change the day.

                                     0: Sunday

                                     1: Monday

                                     2: Tuesday

                                     3: Wednesday

                                     4: Thursday

                                     5: Friday

                                     6: Saturday

Operation   In this example, the clock is read and the day is changed to Monday.

```
000000 MON
```

| FUN | SHIFT | MONTR |

Access the Misc. menu.

```
Misc       12
0:PC Coc
1:Protect UM
2:Step Monitor
```

| 0 |

Specify "PC Clock."
The clock will be displayed.

```
Coc       Read
2011 1
14:15:40
SUN   0
```

| CHG |

Press the Change Key to initiate the change.

```
Coc       C9
2011 1
14:15:40
Sun   0
```

| ↑ |

Move the cursor to the number for the day.

```
Coc          Cg
_2011 1
14:15:40
Sun     0
```

Press 1 for Monday.

```
Coc          Cg
_2011 1
14:40:40
Mon     1
```

Execute with the Write Key.
Press the Clear Key to return to the menu.

```
Coc          Read
2011 1
14:40:40
Mon     1
```
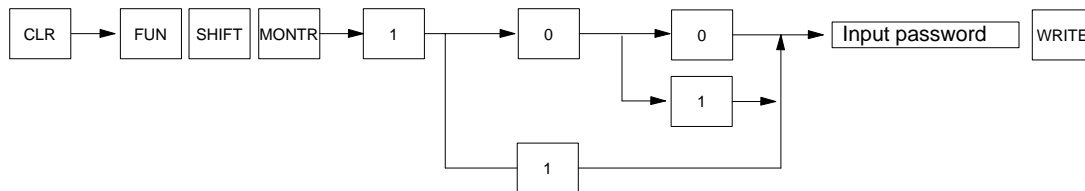
4-4-22 UM Protect

Use

Used to set or clear the read/write protection for the user's program memory (UM). Also used to clear part-protect if it has been set by using the CVSS/SSS.

Modes

RUN	MONITOR	DEBUG	PROGRAM
OK	OK	OK	OK

Key Sequence

CLR → FUN → SHIFT MONTR → 1 → 0 → 0 → Input password → WRITE
                                    ↘ 1 ↗
                          ↘ 1 ↗

Operation

In this example, read/write protection is set.

```
00000 PRG
```

FUN | SHIFT | MONTR

Access the Misc. menu.

```
Misc        12
0:PC Coc
1:Protect UM
2:Step Monitor
```

↓

[1]   Specify "Protect UM."

↓

```
Protect UM
  0:Tota Protect
  1:Part Protect
```

↓

[0]   Specify "Total protect."

↓

```
Tota   Protect
  0:Set
  1:Reease
```

↓

[0]   Specify "Set."

↓

```
Tota   Protect
Set
Passd            _
```

↓

[SHIFT][0]   Set a password.
The password can be 8 digits in hexadecimal.
The same password will be needed to clear protection later.

↓

```
Tota   Protect
Set
Passd 0000000A_
```

↓

[WRITE]   Press the Write Key to execute.
Press the Clear Key to return to the menu.

↓

```
Tota   Protect


Protection set
```

Remarks

The part-protection clear operation clears part-protection set using the CVSS, SSS, or SYSMAC-CPT. The same password that was used for the part-protect operation must be used.

If there is an error, an error message will be displayed when the Write Key is pressed. The following errors may occur.

Protect error: A protection error occurred when the user attempted to protect user memory that has been already protected.

A protect error also occurs when the user has attempted to clear the protection of the user's memory but actually it was not protected.

Password error: A password error occurs when a wrong password was input to clear the protection of the user's memory that has been protected.

If one of the above errors occurs, press the Clear Key to return to the previous display.

Precautions

If the user's memory is protected, programs cannot be read, written, or monitored.

If the user should forget the password, the protection of the user's memory cannot be cleared. The only solution in this case is to clear the memory, which will clear protection, and transfer a backup of the program to the Programming Console from a memory card or other source.
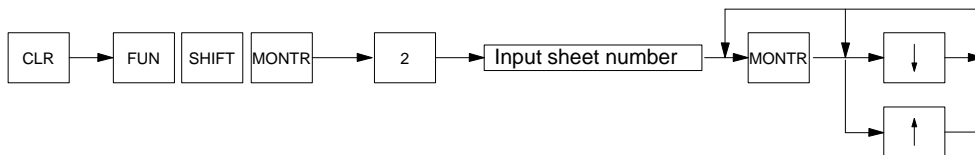
4-4-23 Active Step Monitor (CV-series PCs Only)

Use

Used to search for the active steps in the SFC program and display the step numbers for a specified sheet.
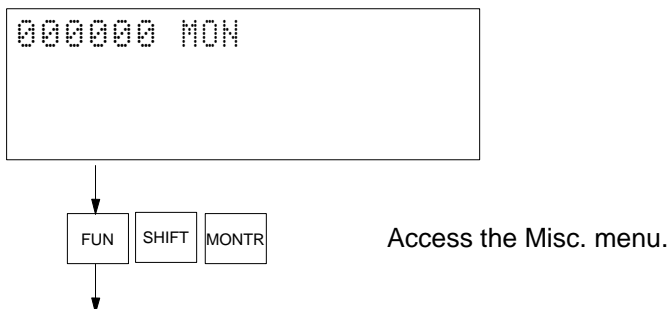
Modes

RUN	MONITOR	DEBUG	PROGRAM
OK	OK	OK	OK

Key Sequence

Operation

In this example, the active steps in sheet number 011 are displayed.

Press the Clear Key to access the initial display.

Access the Misc. menu.

67

```
Misc        12
0:PC Coc
1:Protect UM
2:Step Monitor
```

```
  2
```
Specify "Step monitor."

```
Actie  Step Moni
Seet #000_
```

```
  1      1
```
Input the sheet number.

```
Actie  Step Moni
Seet #011_
```

```
MONTR
```
Press the Monitor Key to search and display active step numbers.

A set of nine active step numbers will be displayed at the same time for 0.7 s then each succeeding sets will be displayed if there are more than nine active step numbers.

Press the Up Key to freeze the present monitor display.

Press the Down Key to read and freeze the next display.

Press the Monitor Key to allow the display to change every 0.7 s.

```
Seet  #011
0000  0010  0150
0200  0020  0103
1050  0003  05
```

Remarks

If there is an error, one of the following messages will be displayed.

Sheet not found: Occurs when the user has attempted to designate a sheet that does not exist.

No active steps: Occurs when no step is active. When a step becomes active, it will be displayed and the message removed.

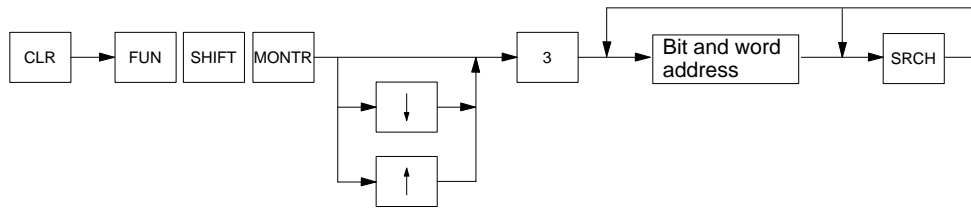## 4-4-24 SFC Search (CV-series PCs Only)

Use

Used to search for bit and word addresses in the SFC program area and display the sheet number and step/transition number where it exists.

Note This operation does not search ladder diagrams used within the SFC program, only the SFC portions of the program.
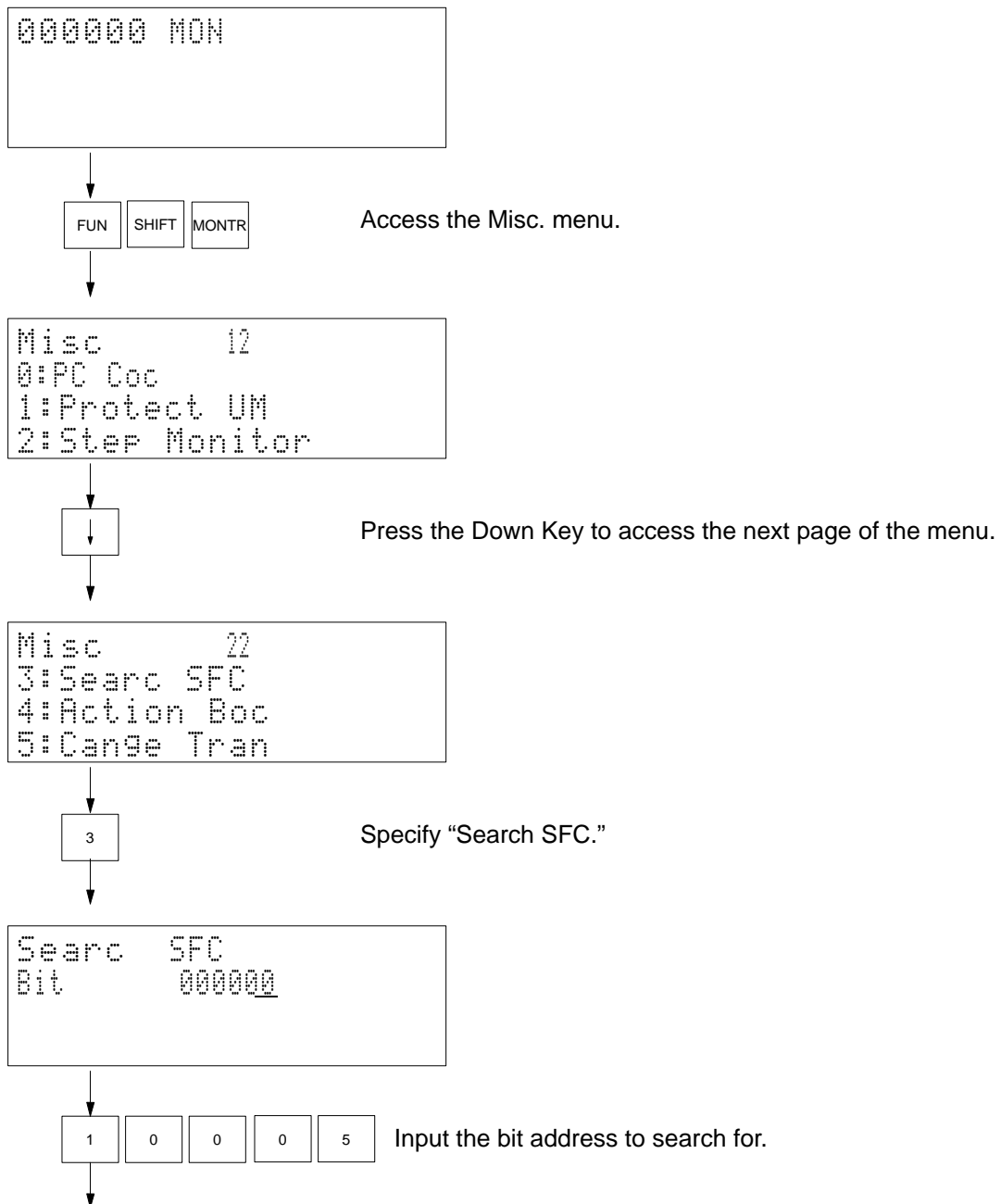
Modes

RUN	MONITOR	DEBUG	PROGRAM
OK	OK	OK	OK

Key Sequence

```
CLR ──► FUN  SHIFT  MONTR ──────────────────► 3 ──► Bit and word ──► SRCH
                              │      ▲              address
                              ▼      │
                            [ ↓ ]────┤
                              │      ▲
                              ▼      │
                            [ ↑ ]────┘
```

Operation In this example, a search is made for address 010005.

```
000000 MON
```

FUN SHIFT MONTR   Access the Misc. menu.

```
Misc       1⁄2
0:PC Coc
1:Protect UM
2:Step Monitor
```

[↓]   Press the Down Key to access the next page of the menu.

```
Misc       2⁄2
3:Searc SFC
4:Action Boc
5:Cange Tran
```

[3]   Specify "Search SFC."

```
Searc  SFC
Bit       000000
```

[1] [0] [0] [0] [5]   Input the bit address to search for.

69

```
Searc   SFC
Bit      010005
```

SRCH   Press the Search Key to start the search operation.

```
Searc   SFC
Bit      010005

Bus
```

When the first occurrence is found, the sheet number, step or transition number, and action or FV where it was found will be displayed.

```
Searc   SFC
Bit      I010005
Tran:      TN0002
St 001
```

SRCH   Press the Search Key to search for the next occurrence.

```
Searc   SFC
Bit      I010005
Tran:      TN0002
St 001
```

SRCH   The message "No such data" will be displayed when no more occurrences are found.

Press the Clear Key to return to the menu.

```
Searc   SFC
Bit      010005

No suc data
```

Remarks

Refer to *Appendix D Inputs for Instructions and Operands* for the key sequences to specify instructions and operands.

Bits and words can be specified for the following parts of an SFC program.

Bits: Actions, transitions, FV (feedback variable)

Words: AQ set value

4-4-25 Action Block Change (CV-series PCs Only)

Modes
Used to read the contents of the designated step number's action block and change AQs, set values, actions, and FVs within the action block.

Modes

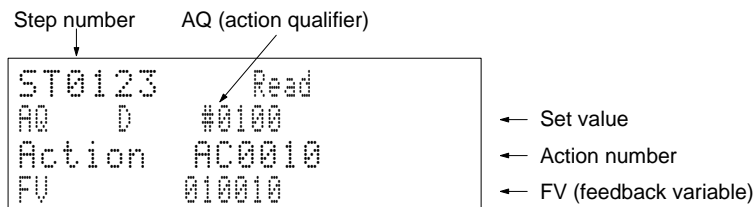| RUN	MONITOR	DEBUG	PROGRAM
See note.	OK	OK	OK

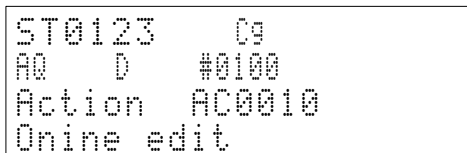**Note** Data can only be read in the RUN mode.

Key Sequence
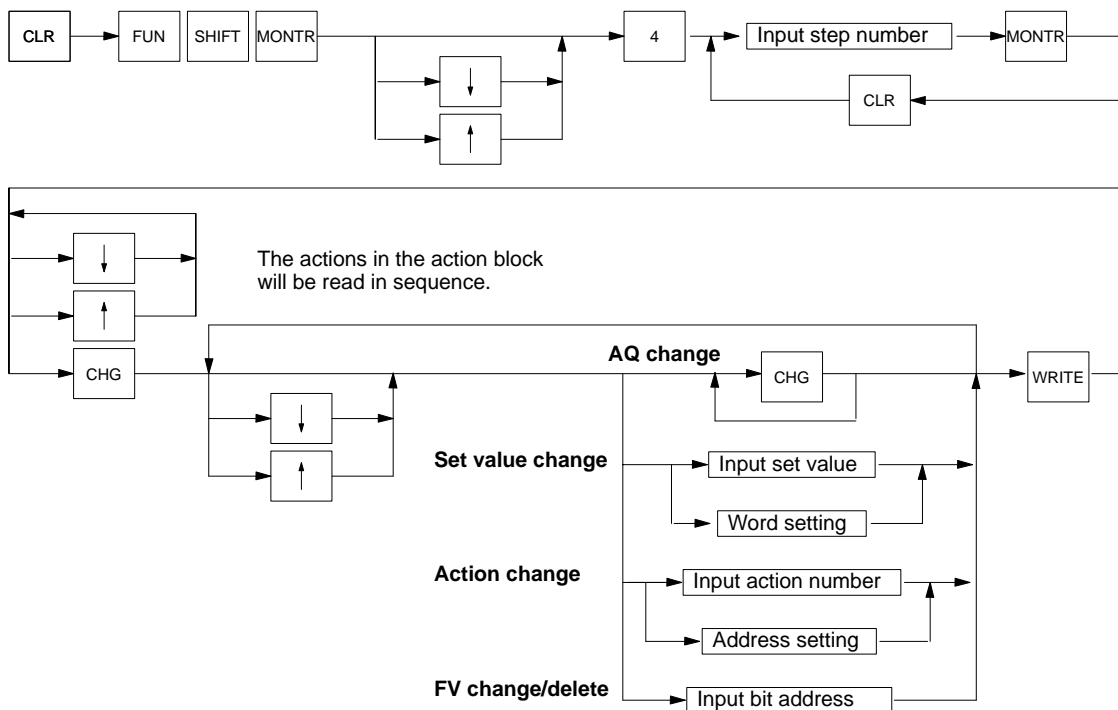
Remarks
The following is the configuration of the action read display.

The following display will appear when the Write Key is pressed in the MONITOR or DEBUG mode.

Press the Write Key to change the data.

or Press the Clear Key to abort.

71

If there is an error, one of the following messages will be displayed.

No such step: Occurs when the user has attempted to designate a step that does not exist in the program.

No act block: Occurs when no action block exists for the designated step number.

Operation

In this example, the contents of ST0123's action block is read and AC 0010 is changed as follows:

AQ: From N to D

Set value: To #0100

Action: From AC 0010 to CIO 001002

```
000000 MON



```

↓

| FUN | SHIFT | MONTR |

Access the Misc. menu.

↓

```
Misc        12
0:PC Coc
1:Protect UM
2:Step Monitor
```

↓

| ↓ |

Press the Down Key to access the next menu.

↓

```
Misc        22
3:Searc SFC
4:Action Boc
5:Change Tran
```

↓

| 4 |

Specify "Action block."

↓

```
Action Boc
Step #:  ST0000
```

↓

| 1 | 2 | 3 | MONTR |

Display the first action in ST0123's action block.
Other actions can be displayed in sequence by pressing the UP or Down Key.
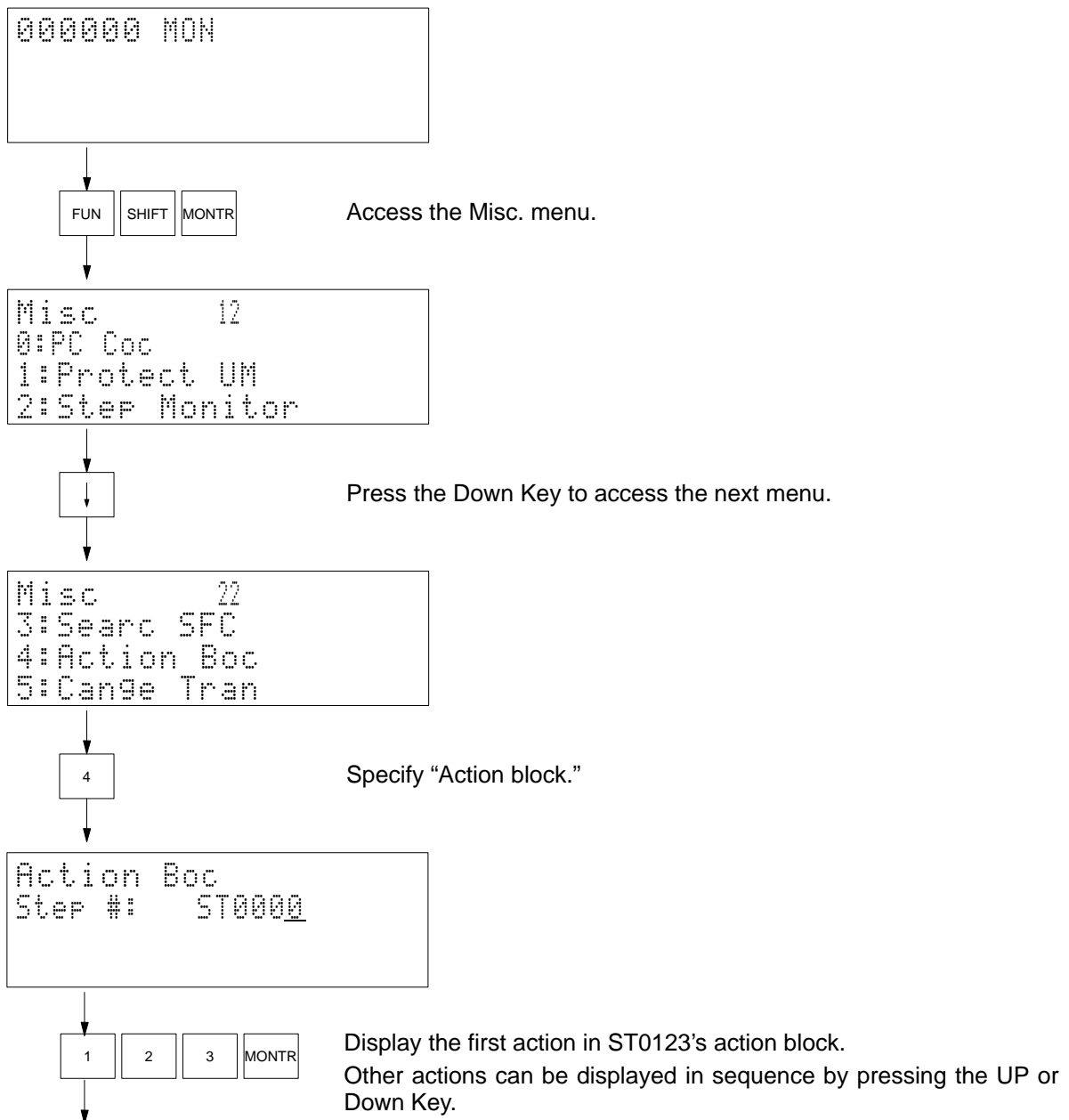
```
ST0123    Read
AQ   N   ▨▨▨▨▨▨
Action  AC0010
FV      010010
```

CHG   Press the Change Key to enable changing the action.

```
ST0123    Cg
AQ    N  ▨▨▨▨▨▨
Action  AC0010
FV      010010
```

CHG   CHG   CHG   CHG   The AQ will change each time the Change Key pressed. Press the
↓   Change Key until the required AQ appears.

Press the Down Key to go to the next part of the action.

```
ST0123    Cg
AQ    D  #0000
Action  AC0010
FV      010010
```

1   0   0   ↓   Input the set value and press the Down Key to go to the next part of the
                  action.

```
ST0123    Cg
AQ    D  #0100
Action  AC0010
FV      010010
```

SHIFT   CONT #   Change the action to CIO 001002.

1   0   0   2

```
ST0123    Cg
AQ    D  #0100
Action  001002
FV      010010
```

WRITE   Execute the change by pressing the Write Key.

```
ST0123    Read
AO   D  #0100
Action  001002
FV     010010
```

4-4-26 Transition Change (CV-series PCs Only)

Use   Used to add a bit address to a transition, change the number of a transition, or delete the bit address for a transition.

Modes

RUN	MONITOR	DEBUG	PROGRAM
See note.	OK	OK	OK

Note Data can only be read in the RUN mode.

Key Sequence
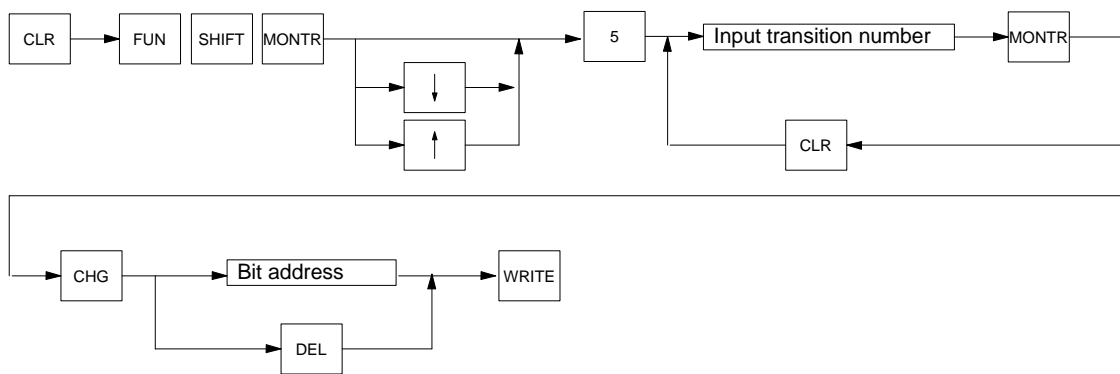
Remarks   The following display will appear when the Write Key is pressed in the MONITOR or DEBUG mode.

```
Cange  Tran
Transition #0123

Onine edit
```

Press the Write Key to change the data.

or Press the Clear Key to abort.

If the user has attempted to designate a transition number that does not exist in the program, an error message will be displayed.

Precautions   If a bit address is added to a transition, the bit address will define the transition even if another transition program has been set.

Operation   In this example, CIO 000105 is added to TN 0123.

```
000000 MON
```


Access the Misc. menu.

```
Misc        12
0:PC Coc
1:Protect UM
2:Step Monitor
```

↓ Press the Down Key to access the next menu.

```
Misc        22
3:Searc SFC
4:Action Boc
5:Cange Tran
```

5 Specify "Change trans."

```
Cange   Tran
Transition #0000
```

1 2 3 MONTR Input the transition number.

```
Cange  Tr  Read
Transition #0123

      ??????
```

CHG SHIFT CONT #   Add the bit. The Programming Console will return to the read opera-
1 0 5 WRITE   tion.

```
Cange   Tr   Read
Transition #0123

      000105
```
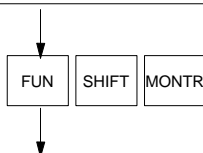
4-4-27 Cycle Time Display

Use Used to display the maximum, minimum, and average value of the cycle time of the program that is being executed.

Modes

RUN	MONITOR	DEBUG	PROGRAM
OK	OK	No	No

Key Sequence
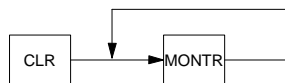
Operation   In this example, the cycle time is read.

```
000000 MON
```
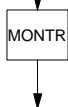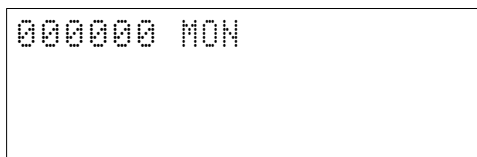

The present values for the cycle time will appear when the Monitor Key is pressed.

The cycle time display will be refreshed each time the Monitor Key is pressed.

```
Cce   time
Ag:     50ms
Ma:    000ms
Min:    00ms
```
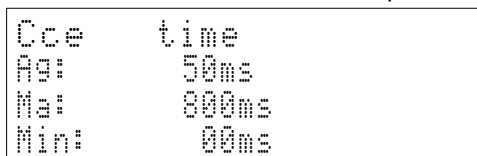
4-4-28 Automatic Error Display

Use   This operation is not executed directly from the main display and instead executed as a sub-operation.

The Programming Console will interrupt all processing operation and automatically display an error message if an error occurs.

If the user program executes MSG(195) for the message number that has been set for the Programming Console, the message generated by the instruction will automatically be displayed.

Modes

RUN	MONITOR	DEBUG	PROGRAM
OK	OK	OK	OK

Note It is possible to clear a fatal error in PROGRAM mode only.

Key Sequence

(Switches between the FAL/FALS number and message)
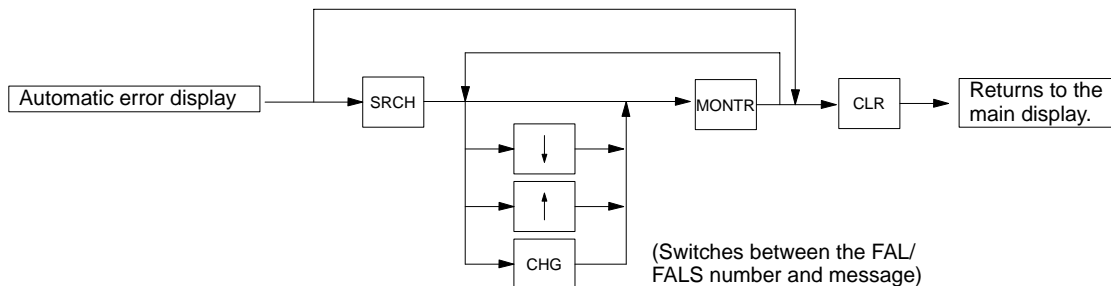
Operation   In this example, an error is cleared to return to the main display.

```
      ERROR
IO eri error

```

│
▼

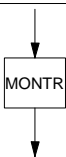```
SRCH
```
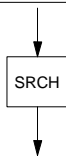
Errors are read when the Search Key is pressed.

Up to three errors are displayed on one display. The succeeding errors, if any, will be displayed on the succeeding displays. Press the Up or Down Key to scroll.

```
Disp error (Su)
Memor error
Sstem FAL   001
IO eri error
```
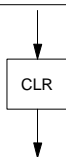
│
▼

```
MONTR
```

Press the Monitor Key successively to clear the errors in sequence. To clear a specific error, move the cursor to the error before pressing the Monitor Key.

The following display will appear when all errors have been cleared.

│
┆
▼

```
MONTR
```

│
▼

```
Disp error (Su)


No errors
```

│
▼

```
CLR
```

Press the Clear Key to return to the main display.
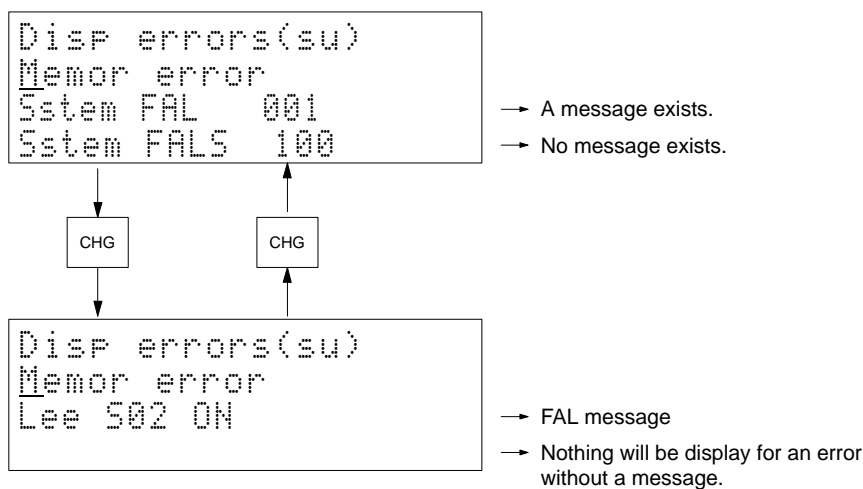
│
▼

```
Main display.
```

Remarks
The following table lists the display priority of error messages. ??? indicates where the FAL number would appear in the actual display.

Priority	Type	Message
High priority	Fatal errors	Memory error
		I/O bus error
		Dup I/O error
		CPU bus error
		Too many I/O
		I/O set error
		Program error
		Cycle time over
		SFC stop error
		System FALS ???
	Non-fatal errors	System FAL ???
		PS error*
		JMP error
		DM address error
		SFC cont error
		I/O verify error
		CPU Bus Unit err
		SYSMAC BUS/2 err
		SYSMAC BUS err
		Battery error
		CPU SIO set err (CPU Bus setting error)
		Power off
		DPL bus error*
Low priority		DPL verify error*

Note Error messages indicated with asterisks appear only for the CVM1D.

Press the Change Key to change the display between the FAL/FALS number and the message (if a message exists).


```
Disp errors(su)
Memor error
Sstem FAL   001      → A message exists.
Sstem FALS  100      → No message exists.
```

```
Disp errors(su)
Memor error
Lee S02 ON           → FAL message
                     → Nothing will be display for an error
                       without a message.
```

The title of messages displayed as a result of executing MSG(195) will be "MSG exec" and will appear along with an error message when the Search Key is pressed. Press the Clear Key to clear the title.

Precautions
Clear errors after eliminating the cause. See the next operation for details.
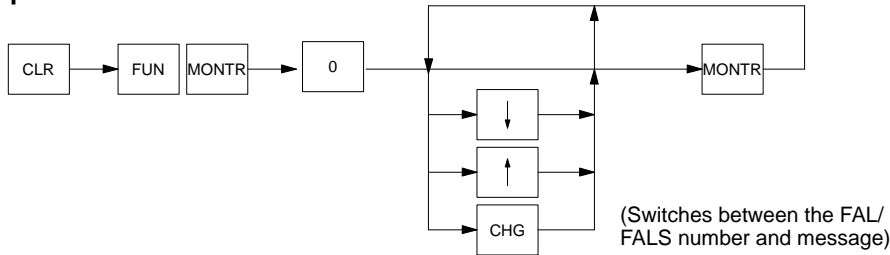
4-4-29 Error Display/Clear

Use Used to read and clear errors.

Modes

RUN	MONITOR	DEBUG	PROGRAM
OK	OK	OK	OK

Note It is possible to clear a fatal error in PROGRAM mode only.

Key Sequence

```
CLR → FUN MONTR → 0 →┬─────────────────────→ MONTR
                     │   ┌───┐
                     ├──→│ ↓ │──→
                     │   └───┘
                     │   ┌───┐
                     ├──→│ ↑ │──→
                     │   └───┘
                     │   ┌─────┐   (Switches between the FAL/
                     └──→│ CHG │    FALS number and message)
                         └─────┘
```

Operation In this example, an error is read and cleared.

```
000000 MON


```

```
┌─────┬───────┐
│ FUN │ MONTR │    Access the Read Errors menu.
└─────┴───────┘
```

```
Read Errors
0:Error displa
1:Histor
2:Messages
```

```
┌───┐
│ 0 │    Specify "Error display."
└───┘
```

Three errors will be displayed on one display. Further errors, if any, will be displayed on following display(s). Press the Up or Down Key to scroll the display.

```
Disp error Read
Memor error
SFC cont error
IO eri error
```

```
┌───────┐
│ MONTR │    Press the Monitor Key successively to clear the errors in sequence.
└───────┘    Move the cursor to an error to clear a specific error.
```

The following display will appear when all errors have been cleared.

Press the Clear Key to return to the main display.

```
┌───────┐
│ MONTR │
└───────┘
```

```
Disp error Read



No errors
```

Remarks   Error messages will be displayed in the order of priority order shown in the table on page 78.

Press the Change Key to change the display between the FAL/FALS number and the message (if a message exists).

```
Disp error Read
Memor error
Sstem FAL    001        → A message exists.
Sstem FALS   100        → No message exists.
```



```
Disp error Read
Memor error
Lee S02 ON              → FAL message
                        → Nothing will be displayed for an error
                           without a message.
```

Precautions   Clear errors after removing the cause.

4-4-30 Error History Read

Use   Used to read the contents of the data stored in the error history area.

Modes

| RUN	MONITOR	DEBUG	PROGRAM
OK	OK	OK	OK

Key Sequence
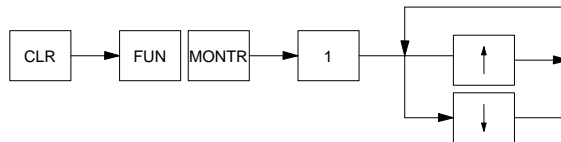
Operation   In this example, the data stored in the error history area is read.

```
000000 MON


```


Access the Read Errors menu.

```
Read Errors
0:Error dispa
1:Histor
2:Messages
```

```
1
```

Specify "History."

The latest error record will be displayed.

Press the Down Key to read the errors in chronological order from the latest error. Press the Up Key to scroll up.

Press the Clear Key to return to the menu.

```
Histor     Read
Cce time oer
2█12█0
21:21:41
```

Remarks The contents of the display are as follows:

```
Histor      Read
PS error
2█12█0
18:05:20
```

←── The contents of the error
←── Date (year - month - day)
←── Time (hr:min:s)

The default values of the error history area are as follows:

Storage area: Words A100 to A199

Maximum number of records: 20

The error history area will be cleared when A00014 (Error Log Reset Bit) is turned ON.

4-4-31 Message Read

Use Used to read and/or clear messages generated by MSG(195).

Modes

RUN	MONITOR	DEBUG	PROGRAM
OK	OK	OK	OK

Key Sequence

(Message cleared)
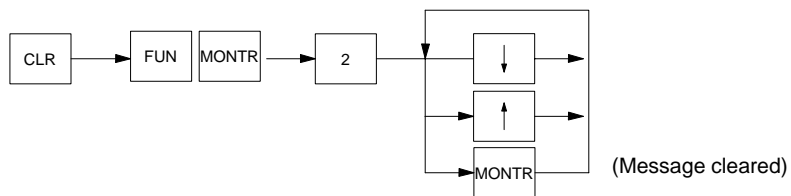
Operation In this example, messages generated by MSG(195) are read.

```
000000 MON
```

```
FUN   MONTR
```
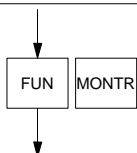
Access the Read Errors menu.

```
Read Errors
0:Error dispa
1:Histor
2:Messages
```

```
  2
```

Specify "Messages."

The message number that will appear first will be the one that was set in the Programming Console system settings.

The following display will appear if MSG(195) has not been executed.

```
Message #1


NO Message
```

```
  ↓
```

Press the Down or Up Key to change the message number.

```
  ↓
```

```
Message #5

Parts inised
No assemin9
```

```
MONTR
```

The message will disappear when the Monitor Key is pressed.

```
Message #5


NO Message
```

4-4-32 Startup Mode

Use   Used to set the PC mode that will be used when the power is turned on.

Modes

RUN	MONITOR	DEBUG	PROGRAM
No	OK	OK	OK

Key Sequence

CLR → FUN → SW/VER → 0 → 0 to 3 → WRITE

82
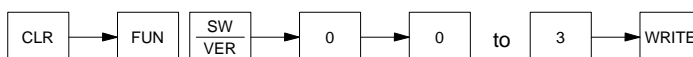
Operation   In this example, the startup mode is changed from MONITOR to RUN mode so that the PC will begin operation immediately the next time it is turned on.

```
000000 MON
```
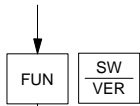
[FUN] [SW / VER]   Access the PC Setup menu.

```
PC Ss Setup 13
0:Startup mode
1:Eec contros
2:Host in
```
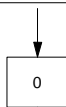
[0]   Specify "Startup mode."
The present mode will be displayed on the second line.

```
Startup mode
        MON
0:PRG 1:DBG
2:MON 3:RUN
```
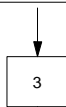
[3]   Specify RUN.

```
Startup mode
        RUN
0:PRG 1:DBG
2:MON 3:RUN
```

[WRITE]   Press the Write Key to execute the change.
Press the Clear Key to return to the menu.

```
Startup mode
        RUN
0:PRG 1:DBG
2:MON 3:RUN
```

Remarks   The new mode will be effective when the power is turned on again.

4-4-33 Execution Controls

Use

Used to determine whether or not to regard multiple, simultaneous execution of the same action from two steps as an non-fatal SFC error, and whether or not to continue PC operation when an I/O comparison error occurs. This operation is not supported by the CVM1D.
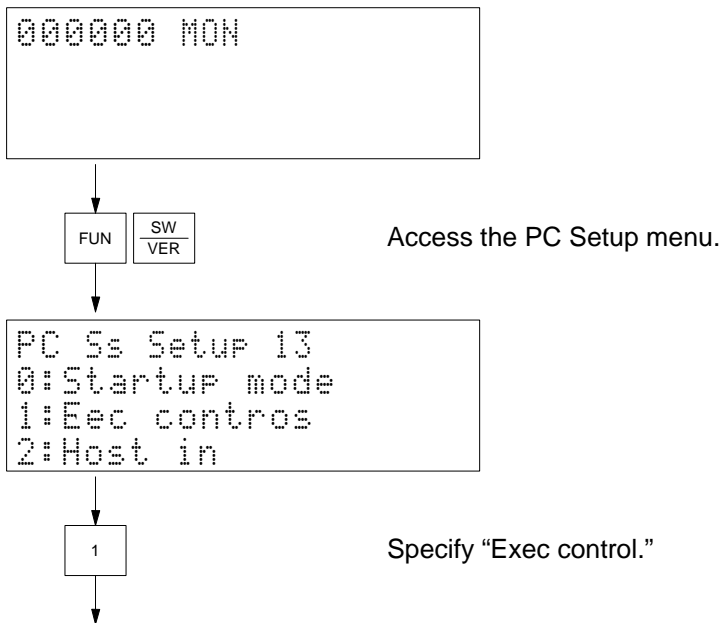
Modes

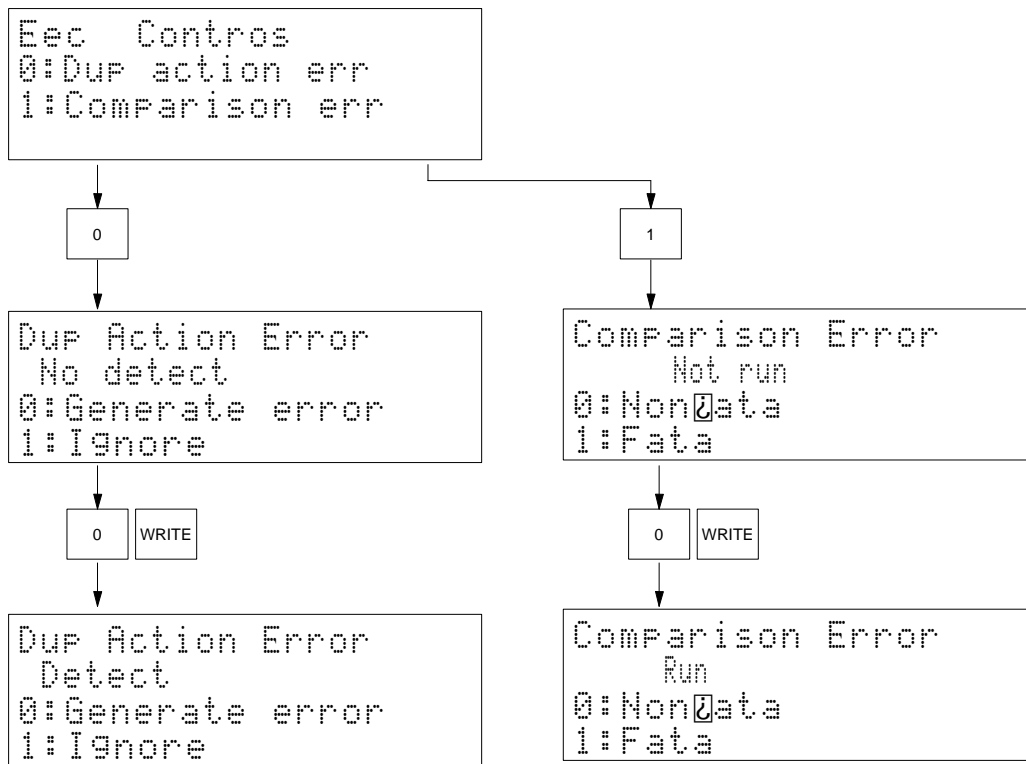| RUN	MONITOR	DEBUG	PROGRAM
No	OK	OK	OK

Key Sequence

```
CLR → FUN → SW/VER → 1 → 0 → 0 → WRITE
                          1     1
```
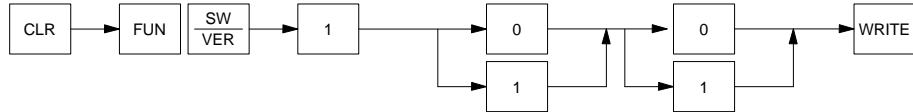
Operation

In this example, multiple action execution is set as a non-fatal SFC error and the PC is set to continue operation even if an I/O comparison error occurs.

```
000000 MON
```

```
FUN   SW
      VER
```
Access the PC Setup menu.

```
PC Ss Setup 13
0:Startup mode
1:Eec contros
2:Host in
```

```
1
```
Specify "Exec control."

```
Eec  Contros
0:Dup action err
1:Comparison err
```

```
                  0
```

```
                  1
```

```
Dup Action Error
 No detect
0:Generate error
1:Ignore
```

```
Comparison Error
     Not run
0:NonData
1:Fata
```

```
         0      WRITE
```

```
         0      WRITE
```

```
Dup Action Error
 Detect
0:Generate error
1:Ignore
```

```
Comparison Error
      Run
0:NonData
1:Fata
```

Remarks The data changed will be effective immediately.

4-4-34 Host Link

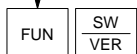**Use** Used to set the communication parameters for the host interface.

Modes

RUN	MONITOR	DEBUG	PROGRAM
No	OK	OK	OK

Key Sequence

CLR → FUN → SW/VER → 2 → 0 to 2 → Menu selection → WRITE
↓ → 3
4 → Input unit number
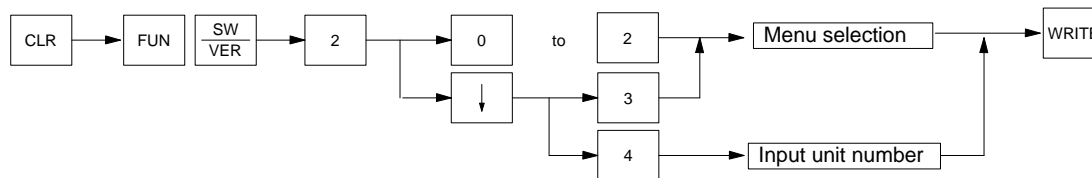
Operation In this example, the host link settings are read.

```
000000 MON
```
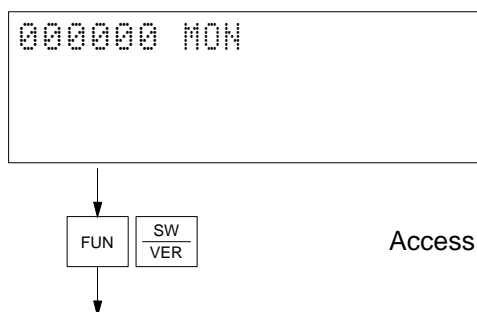
FUN SW/VER   Access the PC Sys Setup menu.

```
PC Ss Setup 13
0:Startup mode
1:Eec contros
2:Host in
```

⬇

```
2
```

Specify "Host link."

⬇

```
Host in   12
0:Baud rate
1:Stop its
2:Parit
```

⬇

```
↓
```

Press the Down or Up Key to change the display and specify the desired item.

⬇

```
Host in   22
3:Data engt
4:Unit numer
```

Settings

```
Baud rate
     1200 _ps
0:12 1: 2:48
3:24  4:12(100)
```

```
Stop Bits
      ⌐its
1:1 it
2:2 its
```

Specify the desired setting and press the Write Key.

```
Parit
      ⌐en
0:None 1:Odd
2:Een
```

```
Data engt
    _its
: its
8:8 its
```

Input the unit number and press the Write Key. Then press the Clear Key to return to the menu.

```
Unit Numer
      00
00 31
```
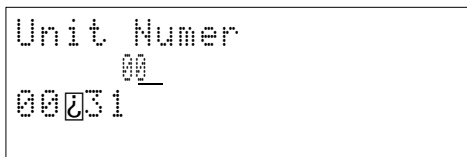
Any parameters that are changed will be effective immediately.

4-4-35 Scheduled Interrupt

Use

Used to set the time unit for scheduled interrupts. This operation is not supported by the CVM1D.

Modes

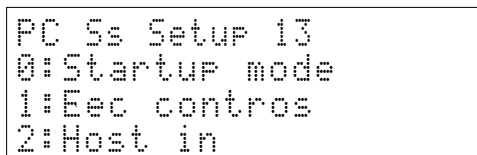| RUN	MONITOR	DEBUG	PROGRAM
No	OK	OK	OK

Key Sequence

CLR → FUN → SW/VER → ↓ → 3 → 0 → to → 2 → WRITE
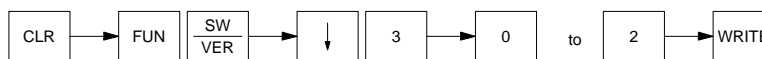
Operation

In this example, a scheduled interrupt time unit of 1.0 ms is set.
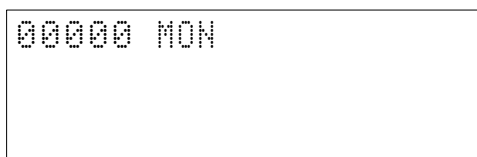
```
00000 MON
```

FUN SW/VER

Access the PC Sys Setup menu.

```
PC Ss Setup 13
0:Startup mode
1:Eec contros
2:Host in
```

↓

Press the Down or Up Key to change the menu.

```
PC Ss Setup 23
3:Scedued
4:Momen pr int
5:Cce time
```

3

Specify "Scheduled."

87

```
Scedued    unit
       100_ms
0:100 1:10
2: 05
```

```
[ 1 ]  [ WRITE ]
```
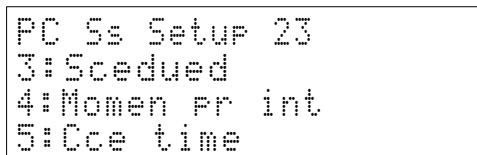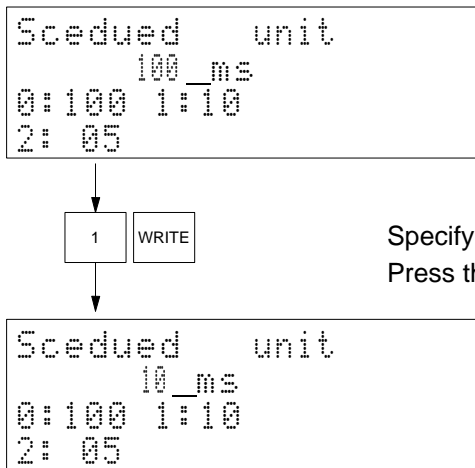
Specify 1.0 and press the Write Key.
Press the Clear Key to return to the menu.

```
Scedued    unit
       10_ms
0:100 1:10
2: 05
```

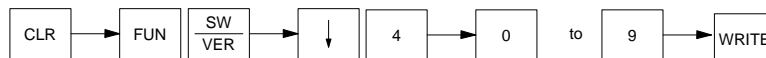**Remarks**   The data changed will be effective from the next time operation is started.

4-4-36 Momentary Power Interruption Time

Use   Used to set a momentary power interruption time, which is used to determine
                                   when a momentary power interruption has occurred. This operation is not sup-
                                   ported by the CVM1D.

Modes

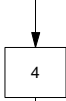| RUN	MONITOR	DEBUG	PROGRAM
No	OK	OK	OK

Key Sequence

```
[ CLR ] → [ FUN ] → [ SW/VER ] → [ ↓ ] → [ 4 ] → [ 0 ] to [ 9 ] → [ WRITE ]
```

Operation   In this example, a momentary power interruption time of 5 ms is set.
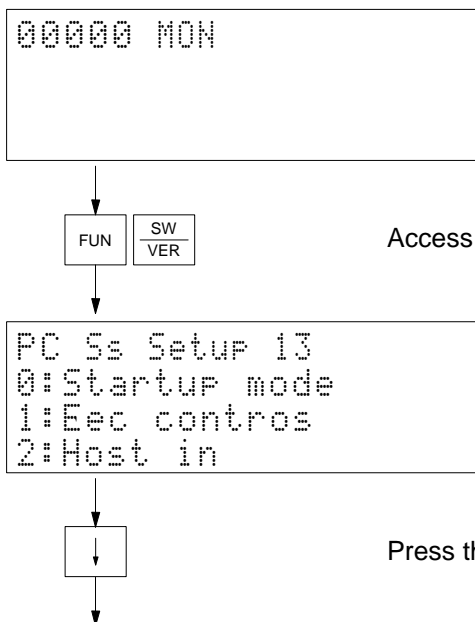
```
00000 MON


```

```
[ FUN ] [ SW/VER ]
```

Access the PC Sys Setup menu.

```
PC Ss Setup 13
0:Startup mode
1:Eec contros
2:Host in
```

```
[ ↓ ]
```
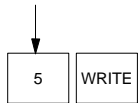
Press the Down or Up Key to change the menu.

88

```
PC Ss Setup 23
3:Scedued
4:Momen pr int
5:Cce time
```

[4]   Specify "Momen pwr int."

```
Mo Pr Int Time
      00_ms

00ↇ0
```

[5] [WRITE]   Specify the desired value and press the Write Key.
                             Press the Clear Key to return to the first menu.

```
Mo Pr Int Time
      05_ms

00ↇ0
```

Remarks   The data changed will be effective immediately.

                             Set the momentary power interruption time within a range of 0 to 9 ms.

                             If the supply voltage returns to normal within the momentary interruption defini-
                             tion time that has been set, the PC will continue operating.

4-4-37 Cycle Time

Use   Used to set a minimum or variable value for the cycle time.

Modes

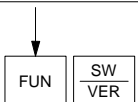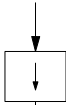| RUN	MONITOR	DEBUG	PROGRAM
No	OK	OK	OK

Key Sequence

[CLR] → [FUN] → [SW/VER] → [↓] → [5] → [0] → [WRITE]
                                          → [1] [WRITE] → [Input cycle time] → [WRITE]
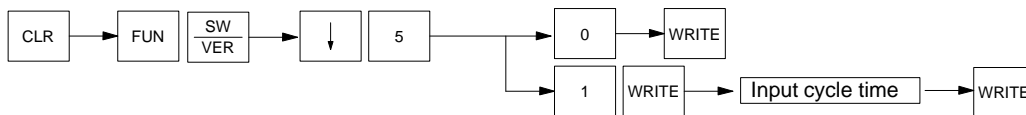
Operation   In this example, a minimum cycle time of 120 ms is input.
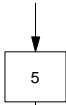
```
00000 MON


```

[FUN] [SW/VER]   Access the PC Sys Setup menu.

```
PC Ss Setup 13
0:Startup mode
1:Eec contros
2:Host in
```
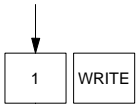
│
↓

Press the Down or Up Key to change the menu.

```
PC Ss Setup 23
3:Scedued
4:Momen pr int
5:Cce time
```

[5]   Specify "Cycle time."

```
Set cce time
 V_ar
0:Cce ariae
1:Cce constant
```

[1] [WRITE]   Specify "Cycle constant."
The present set value will be displayed on the second line.

```
Set cce time
      00100_ms
1¿32000
```

[1] [2] [0] [WRITE]   Input the desired minimum cycle time, press the Write Key, and check the results.
Press the Clear Key to return to the menu.

```
Set cce time
      00120_ms
1¿32000
```

Remarks   Set the cycle time within a range between 1 and 32,000 ms.

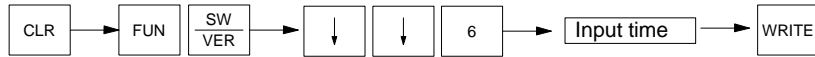The data changed will be effective immediately.

4-4-38 Cycle Time Monitoring Time

Use   Used to set a cycle time monitoring time.

Modes

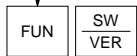| RUN	MONITOR	DEBUG	PROGRAM
No	OK	OK	OK

Key Sequence

```
┌─────┐   ┌─────┐ ┌─────┐   ┌───┐ ┌───┐ ┌───┐   ┌──────────┐   ┌───────┐
│ CLR │─▶│ FUN │ │ SW  │─▶│ ↓ │ │ ↓ │ │ 6 │─▶│Input time│─▶│ WRITE │
└─────┘   └─────┘ │ VER │   └───┘ └───┘ └───┘   └──────────┘   └───────┘
                  └─────┘
```

Operation   In this example, a cycle time monitoring time of 1,200 ms is set.

```
┌──────────────────────────────┐
│ 000000 MON                   │
│                              │
│                              │
│                              │
└──────────────────────────────┘
```
          │
          ▼
┌─────┐ ┌─────┐
│ FUN │ │ SW │   Access the PC Sys Setup menu.
└─────┘ │ VER │
        └─────┘
          │
          ▼
```
┌──────────────────────────────┐
│ PC Ss Setup 13               │
│ 0:Startup mode               │
│ 1:Eec contros                │
│ 2:Host in                    │
└──────────────────────────────┘
```
          │
          ▼
┌───┐ ┌───┐
│ ↓ │ │ ↓ │   Move to the third menu.
└───┘ └───┘
          │
          ▼
```
┌──────────────────────────────┐
│ PC Ss Setup 33               │
│ :Monitor time                │
│                              │
│                              │
└──────────────────────────────┘
```
          │
          ▼
┌───┐
│ 6 │   Specify "Monitor time."
└───┘
          │
          ▼
```
┌──────────────────────────────┐
│ atc    cce time              │
│      0100_0ms                │
│ 1ᴇ4000 (10)                  │
│                              │
└──────────────────────────────┘
```
          │
          ▼
┌───┐ ┌───┐ ┌───┐ ┌───────┐   Input the time and press the Write Key to execute the change.
│ 1 │ │ 2 │ │ 0 │ │ WRITE │   Press the Clear Key to return to the first menu.
└───┘ └───┘ └───┘ └───────┘
          │
          ▼
```
┌──────────────────────────────┐
│ atc    cce time              │
│      0120_0ms                │
│ 1ᴇ4000 (10)                  │
│                              │
└──────────────────────────────┘
```
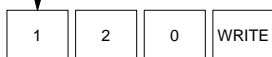
Remarks   Set the cycle time within a range of 10 to 40,000 ms.

                                  The data changed will be effective immediately.
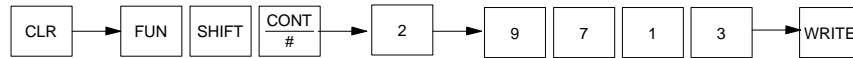
4-4-39 Memory Card Format

Use   Used to format the memory card mounted in the CPU Unit.

Modes

RUN	MONITOR	DEBUG	PROGRAM
OK	OK	OK	OK

Key Sequence

CLR → FUN | SHIFT | CONT # → 2 → 9 | 7 | 1 | 3 → WRITE

Operation   In this example, the RAM memory card mounted in the CPU Unit is formatted.

```
000000 PRG


```

FUN | SHIFT | CONT #   Access the Memory Card Oper menu.

```
Memor  Card Oper
0:Tran   1:Comp
2:Format 3:Erase
```

2   Specify "Format."

```
Format
Passd      _
Drie :0
```

9 | 7 | 1 | 3   The password is 9713.

```
Format
Passd  13 _
Drie :0
```

WRITE   Press the Write Key to start formatting.
The following display will appear when the memory card has been for-matted.

```
Finised
```

Remarks
Only RAM memory cards can be formatted. Write-enable the card before mounting it. The drive name of the memory card is 0.

If there is an error, one of the following error messages will appear.

No file device: No memory card has been mounted in the CPU Unit.

Read only: The memory card is write-protected.

4-4-40 File Write

Use
Used to write the contents of the PC's memory to the memory card mounted in the CPU Unit. The data that can be written includes the user's program, I/O memory, and extended PC setup.

Modes

RUN	MONITOR	DEBUG	PROGRAM
OK	OK	OK	OK

Key Sequence
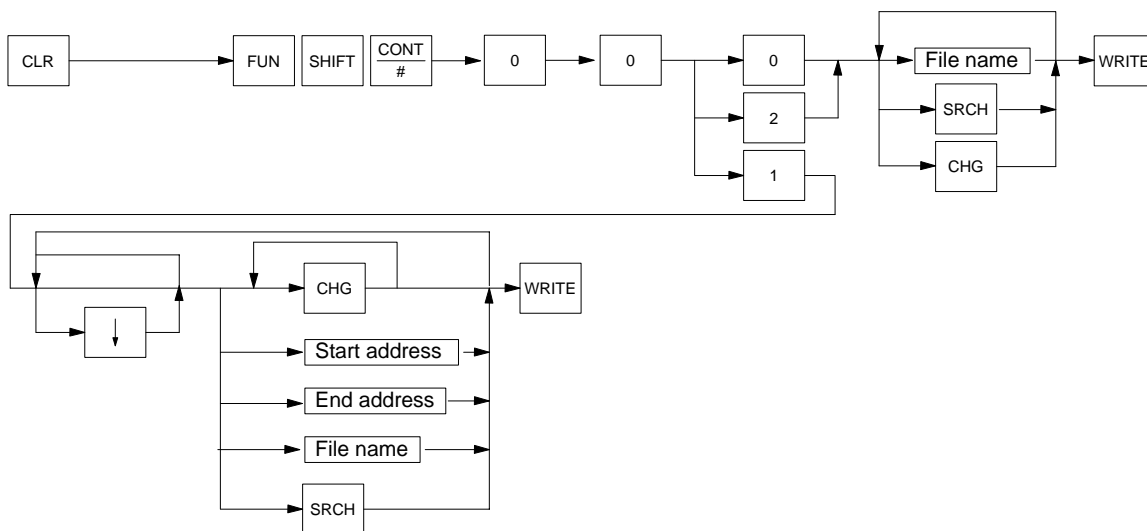
Remarks
Only RAM memory cards can be written. Write enable the card before mounting it. The drive name for the memory card is 0.

```
PC to Card(OB)


Fie"0:  _     "
```
Drive name
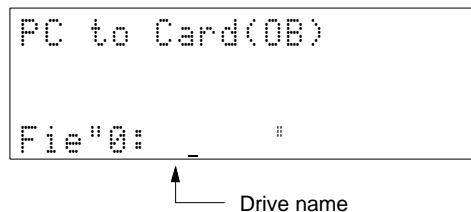
File Names
It is possible to use the following keys to input a file name. A file name can consist of 8 characters maximum. Input is possible in one of two modes: normal (numbers only) or alphanumeric. Press the Shift–Text Keys to enter alphanumeric input mode and ! will be displayed on the upper right-hand corner of the screen. Press the Clear Key to return to the normal mode.

Normal Input Mode
0 to 9

Up and Down Keys: Used to move the cursor to the right or left.

DEL: Used to delete the character where the cursor is located.

CLR: Used to delete all the characters of a file name.

SEARCH: Used to read a file name in the memory card. It is possible to read a single file name each time the Search Key is pressed.

WRITE: Used to input a file name and start writing the file.

Alphanumeric Input Mode
0 to 9 and A to Z (the lower right-hand character on each key) are effective.

Automatic File Names
The following type of file name can be automatically input by pressing the Change Key in the normal input mode.

File type (see below) + PC's clock data (Date, month, and year)

P: User's program

I: I/O memory

S: Extended PC Setup

File Extensions
When a file name is written, one of the following extensions will be added to it.

.OBJ: User's program

.IOM: I/O Memory

.STD: Extended PC Setup

Data Area Designations
The data area designation will be changed and displayed in the following order each time the Change Key is pressed: CIO, G, A, DM, EM0 to EM7.

Existing File Names
If the designated file name already exists, the following message will appear.

```
PC to Card (IOM)
Area:    DM
Fie  0:P203250
Oerrite
```

Press the Write Key to overwrite.

 or Press the Clear Key to abort.

Errors
If there is an error, one of the following error messages will appear according to the error.

No file device: No memory card has been mounted in the CPU Unit.

Read only: The memory card is write-protected or an EPROM or EEPROM memory card has been mounted.

Operation

In this example, the file name SAMPLE is written containing the data from between D00000 and D08191. The initial display will appear when the Clear Key is pressed.

```
000000 PRG

```

↓

| FUN | SHIFT | CONT # |

Access the Memory Card Oper menu.

↓

```
Memor  Card Oper
0:Tran   1:Comp
2:Format 3:Erase

```

↓

| 0 |

Specify "Trans."

↓

```
Data Transer
0:PC to Card
1:Card to PC

```

↓

| 0 |

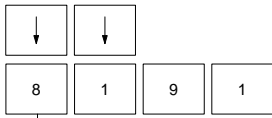Specify "PC to Card."

↓

```
PC to Card
0:Programs (OB)
1:IOM     (IOM)
2:Settings (STD)
```

↓

| 1 |

Specify "IOM."

↓

```
PC to Card (IOM)
Area:    CIO
St 00000End00000
Fie"0:      "
```

↓

| CHG | CHG | CHG |

Change the area.

↓

```
PC to Card (IOM)
Area:      DM
St 00000End00000
Fie"0:       "
```

↓ ↓

8 1 9 1

Move the cursor to "End" and input the end address.

```
PC to Card (IOM)
Area:      DM
St 00000End0011_
Fie"0:       "
```

↓ ↓
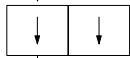
Move the cursor to the file name input area.

```
PC to Card (IOM)
Area:      DM
St 00000End0011
Fie"0:  _    "
```

SHIFT | TEXT / !

S A M P L E

Input the file name in the alphanumeric input mode.

```
PC to Card(IOM)!
Area:      DM
St 00000End0011
Fie"0:SAMPLE  _"
```

CLR WRITE

Press the Clear Key to leave the alphanumeric input mode and press the Write Key.

The following display will appear when the PC has finished writing the file.

```
PC to Card (IOM)
Area:      DM
Fie"0:SAMPLE  "
Finised
```

4-4-41 File Read

Use Used so that the contents of the memory card mounted in the CPU Unit can be read in the PC's memory. The data that can be written includes the user's program, I/O memory, and extended PC setup.

Modes

RUN	MONITOR	DEBUG	PROGRAM
No	See note.	See note.	OK

Note Only IOM files can be read in the MONITOR or DEBUG mode.

Key Sequence
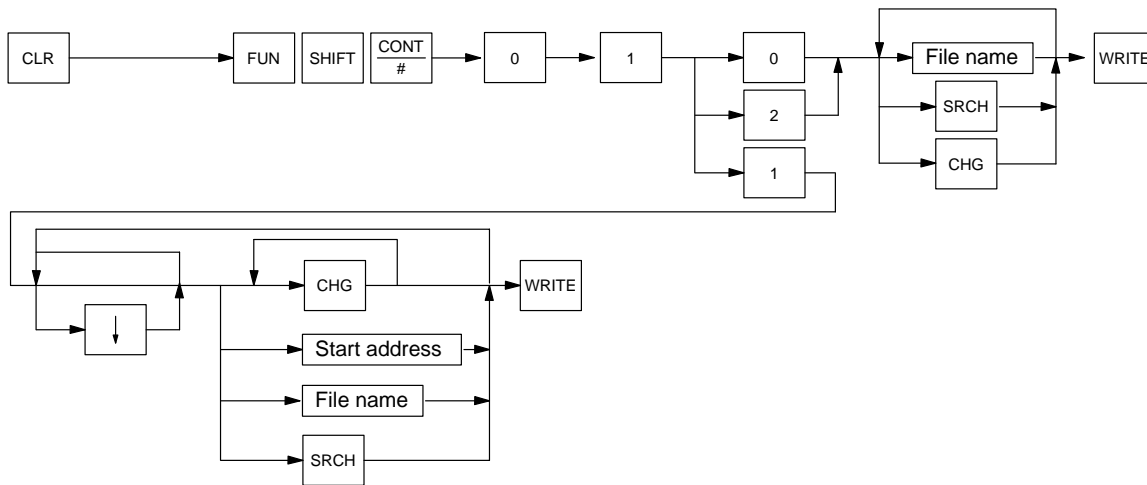
Remarks It is possible to read any type of memory card. The drive name for the memory card is 0.

File Names
Press the Search Key to read the file name from the memory card.

Data Area Designations
The data area designation will be changed and displayed in the following order each time the Change Key is pressed: CIO, G, A, DM, EM0 to EM7.

Error Messages
If there is an error, an error message will be displayed. The following errors may occur.

No file device: No memory card has been mounted in the CPU Unit.

No such file: The specified file does not exist.

Operation In this example, the user's program file SAMPLE in the memory card is read into the PC.
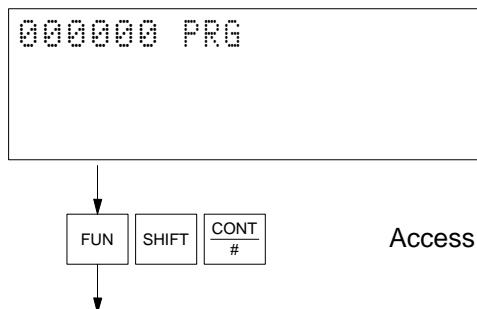
```
000000 PRG
```


Access the Memory Card Oper menu.

97

```
Memor  Card Oper
0:Tran   1:Comp
2:Format 3:Erase
```

↓

| 0 |

Specify "Trans."

↓

```
Data Transer
0:PC to Card
1:Card to PC
```

↓

| 1 |

Specify "PC to Card."

↓

```
Card to PC
0:Programs (OB)
1:IOM     (IOM)
2:Settings (STD)
```

↓

| 0 |

Specify "Programs."

↓

```
Card to PC(OB)


Fie"0:  _    "
```

↓

| SRCH | - - - - | SRCH |

Press the Search Key to find the file named SAMPLE.

↓

```
Card to PC(OB)


Fie"0:SAMPLE  _ "
```

↓

| WRITE |

Press the Write Key to start reading the file.

The following display will appear when the Programming Console has finished reading the file into the PC.

↓

```
Card to PC(OB)

Fie"0:SAMPLE   "
Finised
```

4-4-42 File Compare

Use Used to compare the contents of files in the memory card mounted in the CPU Unit and the PC's memory contents (user's program, I/O memory, or extended PC Setup).

Modes

RUN	MONITOR	DEBUG	PROGRAM
OK	OK	OK	OK

Key Sequence
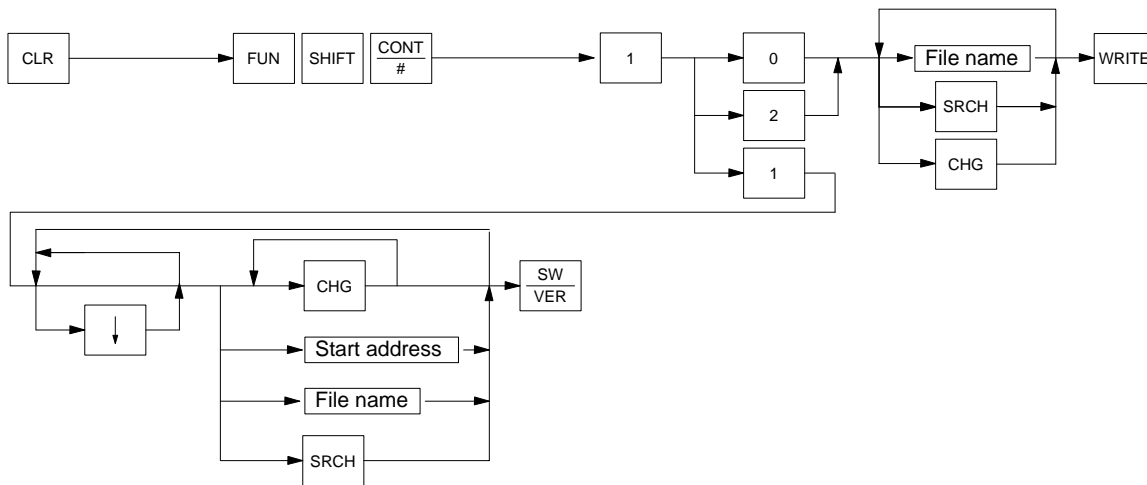
Remarks It is possible to compare the file contents of any type of memory card with the PC's memory contents.

The drive name for the memory card is 0.

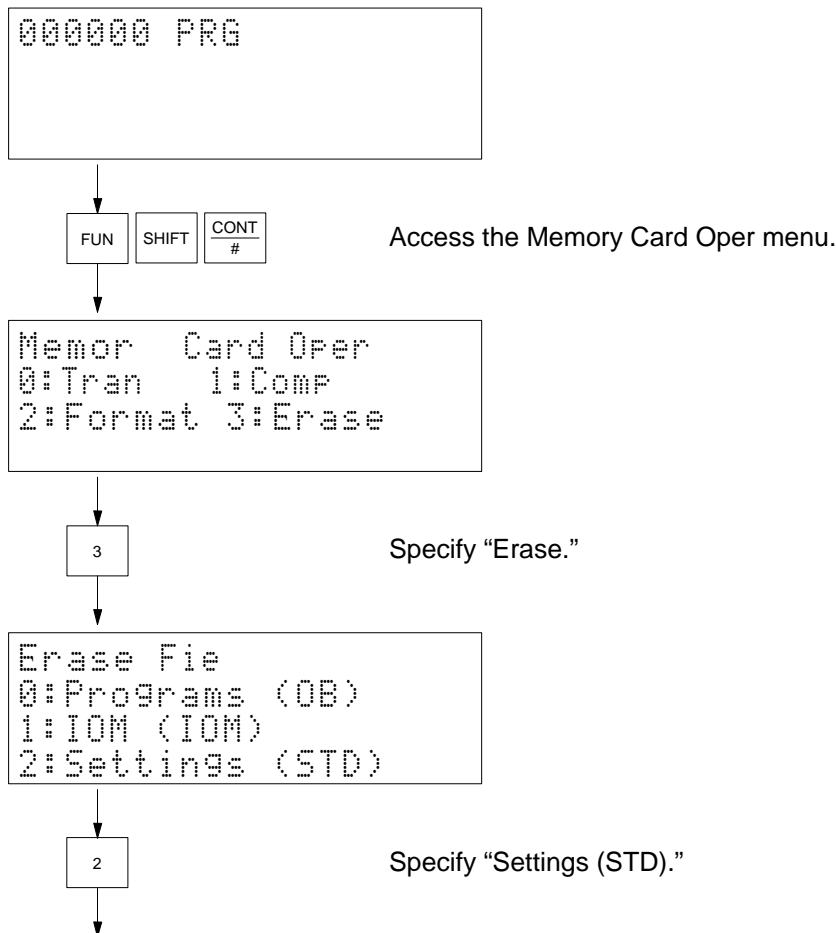Press the Search Key to read the file name from the memory card.

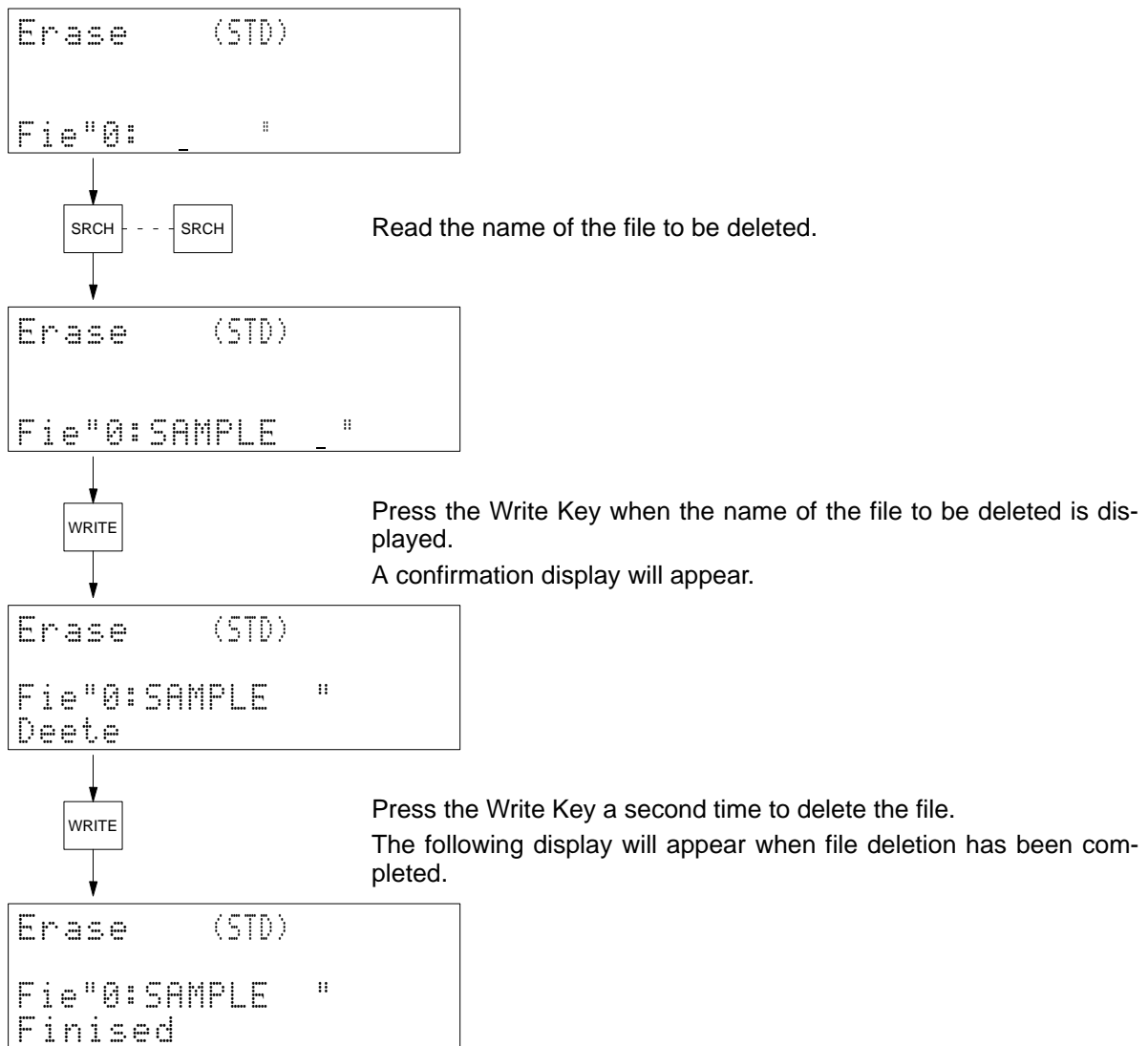If there is an error, one of the following error messages will appear.

No file device: No memory card has been mounted in the CPU Unit.

No such file: The specified file does not exist.

Data differs: The data did not coincide with each other.

Operation In this example, the user's program file SAMPLE in the memory card is compared with the contents of the user program area in the PC.

The initial display will appear when the Clear Key is pressed.

```
000000 PRG
```


Access the Memory Card Oper menu.

```
Memor  Card Oper
0:Tran   1:Comp
2:Format 3:Erase
```

| 1 |

Specify "Comp."

```
PC Card compar
0:Programs (OB)
1:IOM      (IOM)
2:Settings (STD)
```

| 0 |

Specify "Programs."

```
Compare   (OB)


Fie"0:  _    "
```

| SRCH | - - - - | SRCH |

Specify the file name.

```
Compare    (OB)


Fie"0:SAMPLE  _ "
```

| SW / VER |

The comparison will start when the Verify Key is pressed.

The following display will appear if the comparison finishes without finding any differences in the data being compared.

```
Compare   (OB)

Fie"0:SAMPLE   "
Data identica
```
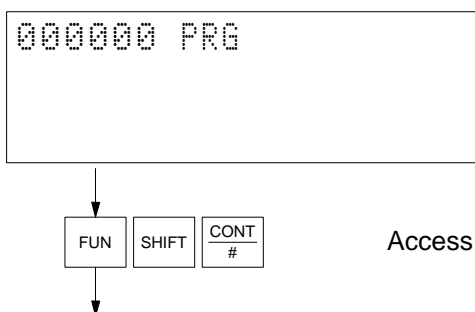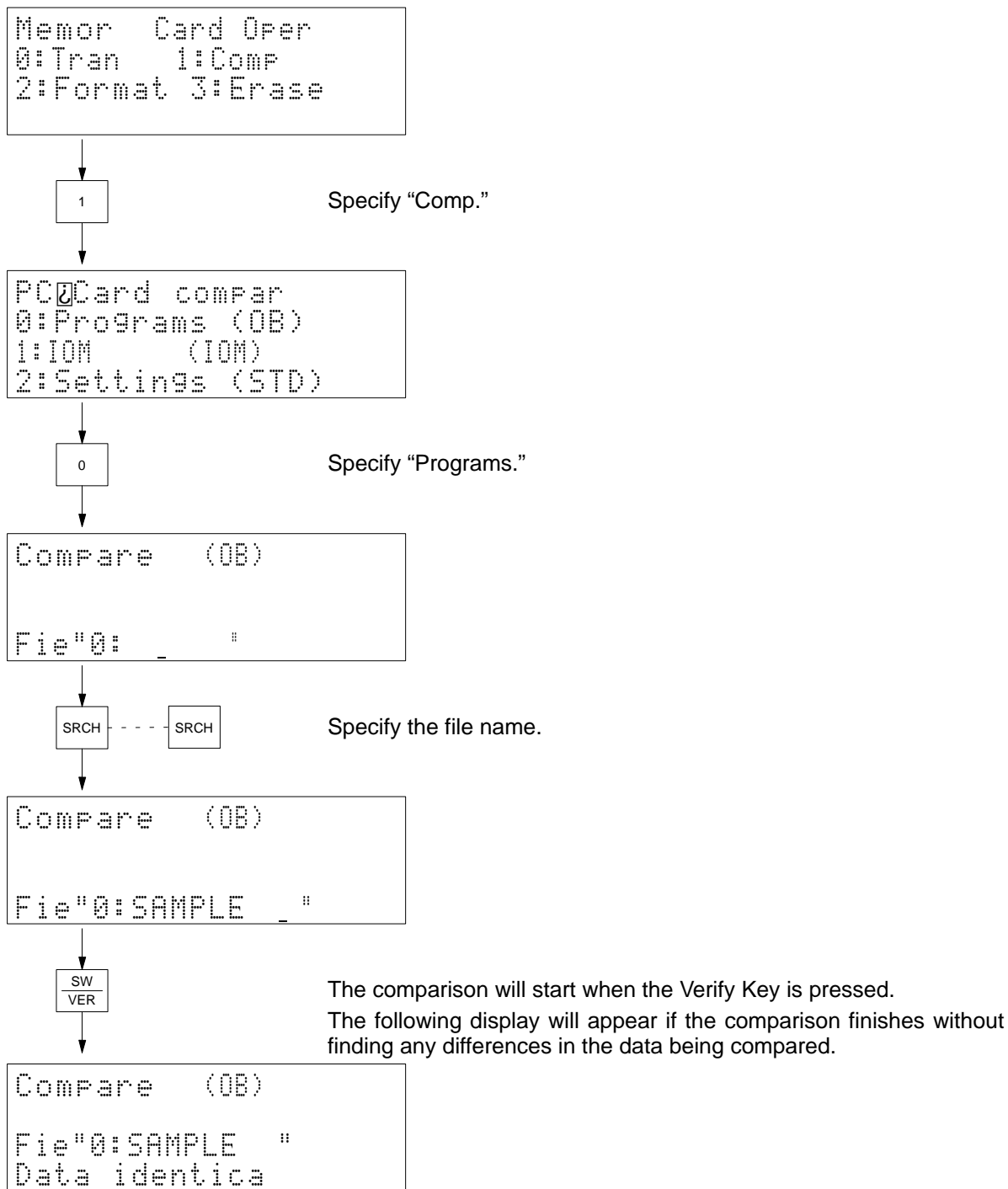
4-4-43 File Delete

Use

Used to delete a designated file (user program, I/O memory, or extended PC Setup) of the memory card mounted in the CPU Unit.

Modes

RUN	MONITOR	DEBUG	PROGRAM
OK	OK	OK	OK

100

Key Sequence
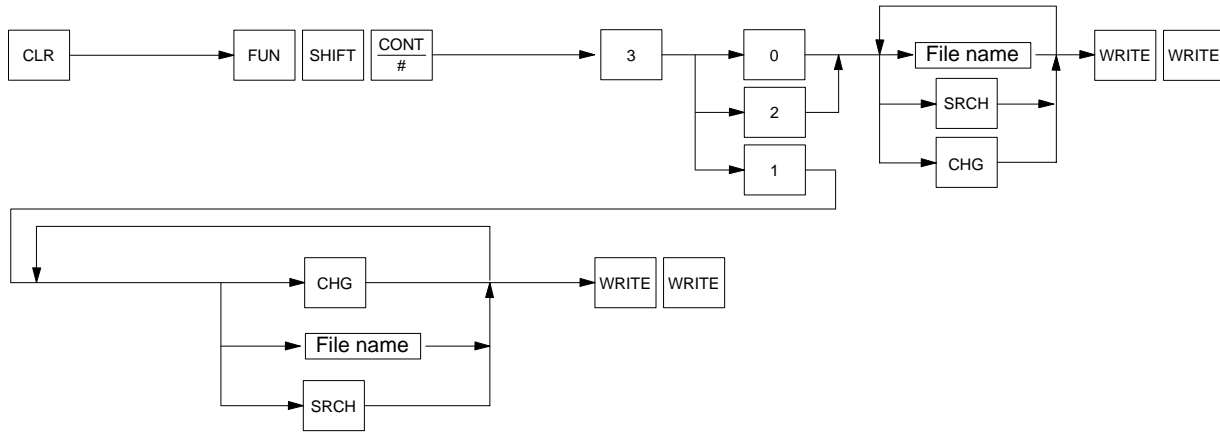
Remarks

Only RAM memory cards can be deleted from using this operation. Be sure to write-enable the card before mounting it.

The drive name for the memory card is 0.

Press the Search Key to read the file name from the memory card.

If there is an error, one of the following error messages will appear.

No file device: No memory card has been mounted in the CPU Unit.

No such file: The specified file does not exist.

Operation

In this example, the extended PC Setup file SAMPLE in the memory card is deleted.


```
000000 PRG
```

```
FUN   SHIFT   CONT
              #
```
Access the Memory Card Oper menu.

```
Memor  Card Oper
0:Tran   1:Comp
2:Format 3:Erase
```

```
3
```
Specify "Erase."

```
Erase Fie
0:Programs (OB)
1:IOM (IOM)
2:Settings (STD)
```

```
2
```
Specify "Settings (STD)."

```
Erase    (STD)


Fie"0:  _      "
```

↓

SRCH - - - SRCH   Read the name of the file to be deleted.

↓

```
Erase    (STD)


Fie"0:SAMPLE  _"
```

↓

WRITE   Press the Write Key when the name of the file to be deleted is displayed.

A confirmation display will appear.

↓

```
Erase    (STD)

Fie"0:SAMPLE   "
Deete
```

↓

WRITE   Press the Write Key a second time to delete the file.

The following display will appear when file deletion has been completed.

↓

```
Erase    (STD)

Fie"0:SAMPLE   "
Finised
```

4-4-44 I/O Table Create

Use   Used to register the I/O table in the PC according to the Units currently mounted or connected to the PC.

Modes

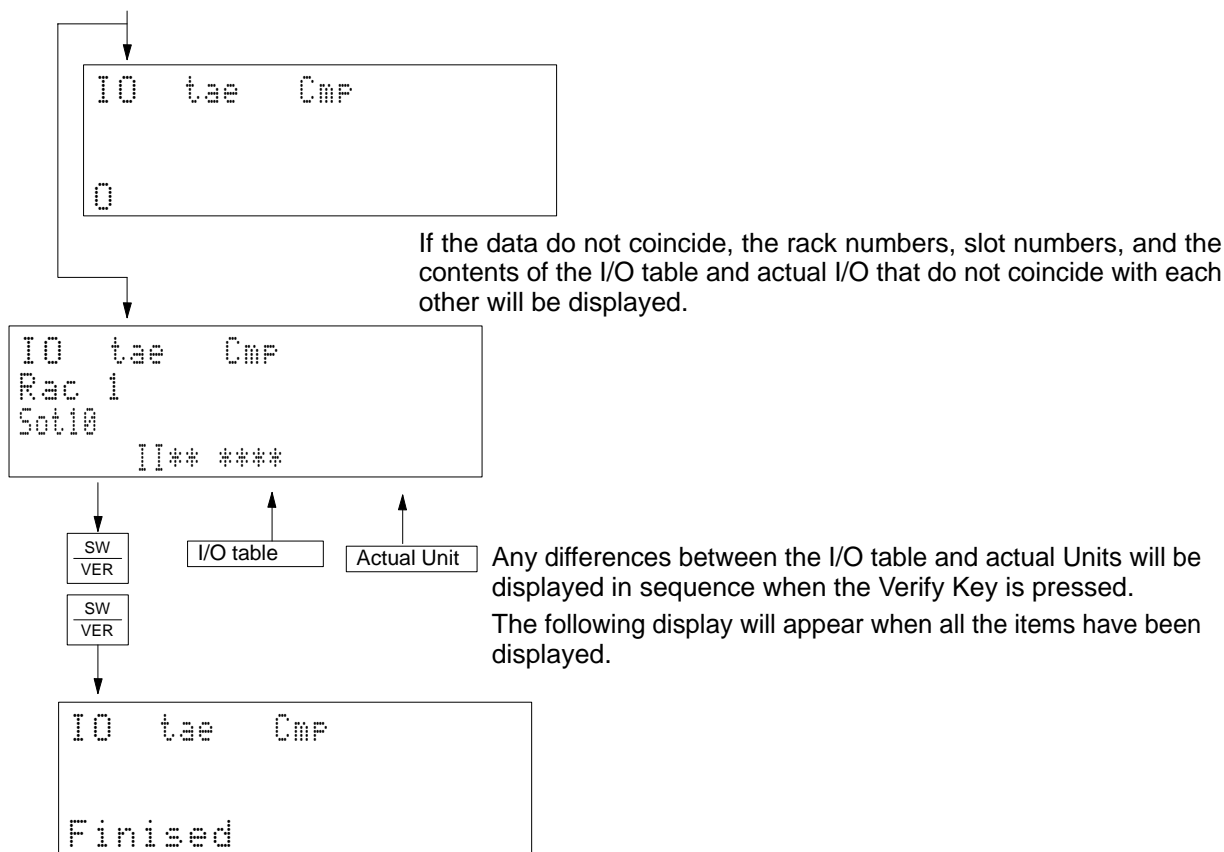| RUN	MONITOR	DEBUG	PROGRAM
No	No	No	OK

Key Sequence

CLR → FUN SHIFT CH/∗DM → 0 → 9 7 1 3 → WRITE → 0

CHG → 1
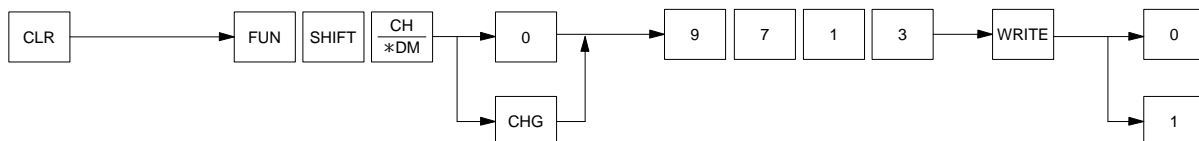
Remarks   All I/O and works bits are cleared when the I/O table is generated.

There is no need to create the PC's I/O table each time the PC is turned on unless the user changes the configuration of the Units mounted to the PC.

If a Unit is added to the PC or an existing Unit replaced with a different type of Unit after the I/O table is generated, an I/O setting error or I/O verification error will occur. Check the Units on the PC before creating the I/O table.

102

"Clear CPU BUnit?" means to clear the contents of software settings for CPU Bus Units or data link tables set with CVSS/SSS online operations. If this data is held, all data will be held regardless of where or not the Units the settings are for actual exist. If the data is cleared, only data for Units not currently mounted in the system will be cleared.

Precautions This operation cannot be used from a Programming Console connected to a SYSMAC BUS/2 Remote I/O Slave Unit.

Operation In this example, an I/O table is created after clearing the CPU Bus Unit settings and data link tables.

```
000000 PRG
```

| FUN | SHIFT | CH
∗DM |

Access the I/O table menu.

```
IO  tae
0:Create
1:Compare
2:Repace IO
```

| 0 |

Specify "Create."

```
IO  tae Create
```

| 9 | 7 | 1 | 3 |

Input the password 9713.

```
IO  tae Create
   13
```

| WRITE |

Press the Write Key to access the following display to specify processing for CPU Bus Unit settings.

103

```
IO  tae Create
Cear CPU BUnit
0:Cear
1:Hod
```

```
   0
```
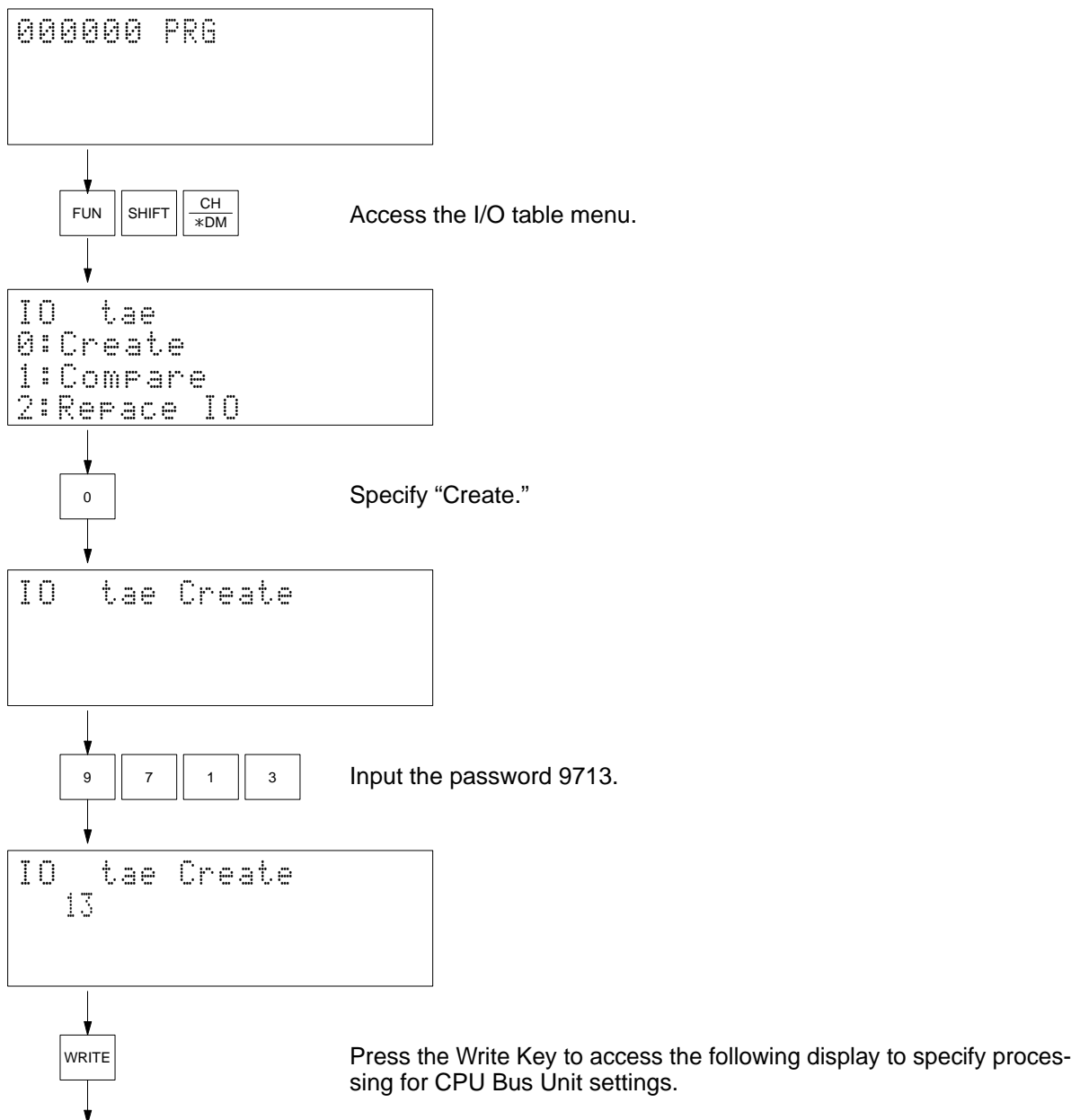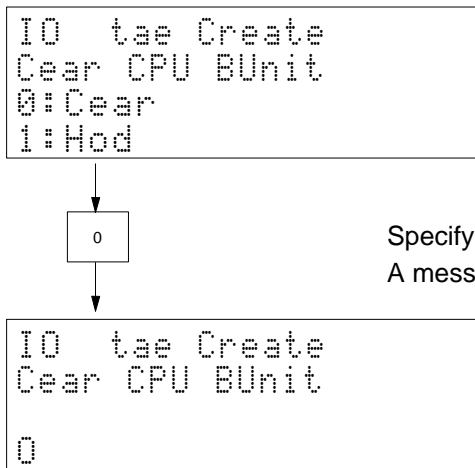
Specify "Clear."
A message will appear when the operation has been completed.

```
IO  tae Create
Cear CPU BUnit

0
```

4-4-45 I/O Table Compare

Used to compare the I/O table registered in the PC to the Units actually mounted or connected to it.

Modes

RUN	MONITOR	DEBUG	PROGRAM
OK	OK	OK	OK

Key Sequence

```
CLR ──────────▶ FUN  SHIFT  CH/∗DM ──▶ 1 ──▶ SW/VER
                                   SW/VER
```
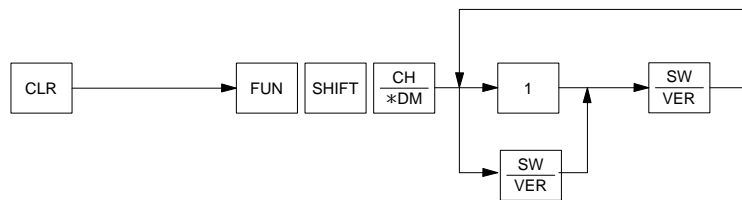
Operation

In this example, the I/O table is compared to actual Units to confirm that the table is correct.

```
000000 PRG

```

```
FUN   SHIFT   CH/∗DM
```

Access the I/O table menu.

```
IO  tae
0:Create
1:Compare
2:Repace IO
```

```
   1
```
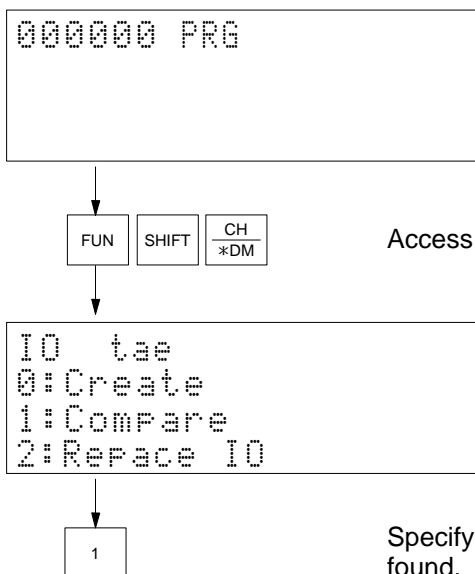
Specify "Comp." The following display will appear if no differences are found.

```
IO  tae  Cmp



O
```

If the data do not coincide, the rack numbers, slot numbers, and the contents of the I/O table and actual I/O that do not coincide with each other will be displayed.

```
IO  tae  Cmp
Rac 1
Sot10
       II** ****
```

| SW / VER | | I/O table | | Actual Unit |

| SW / VER |

Any differences between the I/O table and actual Units will be displayed in sequence when the Verify Key is pressed.

The following display will appear when all the items have been displayed.

```
IO  tae  Cmp



Finised
```

Remarks If there is any differences between the I/O table and actual Units, check the Units. If any differences exist and all Units are mounted or connected according to system specifications, correct the I/O table or create a new I/O table.

Displays for differences will be as follows:

Rack number →
```
IO  tae  Cmp
Rac 1
Sot10
       II** ****
```
Slot number

I/O table entry   Actual I/O Unit
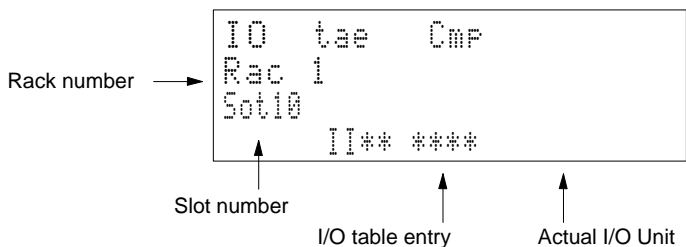
⚠ **Caution** Do not change the I/O table without confirming that the program is written correctly for the allocations that will be produced. Refer to the *CV-series PC Operation Manual: Ladder Diagrams* for details.

I/O Units

Display	Unit	Display	Unit
****	No Unit	0000	64 output points
o***	8 output points	IIII	64 input points
i***	8 input points	00II	64 I/O points
O***	16 output points	NNNN	64 dummy I/O points
I***	16 input points	GG**	128 output points
N***	16 dummy I/O points	LL**	128 input points
00**	32 output points	GL**	128 I/O points
II**	32 input points	HH**	128 dummy I/O points
OI**	32 I/O points	GGGG	256 output points
NN**	32 dummy I/O points	LLLL	256 input points
		GGLL	256 I/O points
		HHHH	256 dummy I/O points

Other Units

RM *(Master #)*: Remote I/O Master Unit
(Master number: SYSMAC BUS: 0 to 7, SYSMAC BUS/2: 0 to 3)

INT: Interruption Unit

SL: SYSMAC LINK Unit

SN: SYSMAC NET Link Unit

BA: BASIC Unit

VP: Personal Computer Unit

CPU Bus Units

CPU Bus Unit number

Unit in I/O table   Actual Unit

SYSMAC BUS Units

Remote I/O Master number (0 to 7)

Remote I/O Slave number (0 to 7)

Slot number

Actual I/O Unit

Unit in I/O table

Allocated word

Remote I/O Master address (0 to 7)

H/L word discrimination

Unit in I/O table   Actual I/O Unit
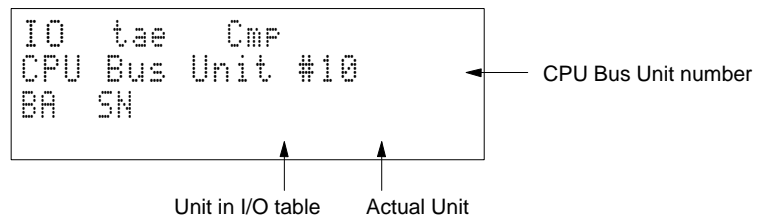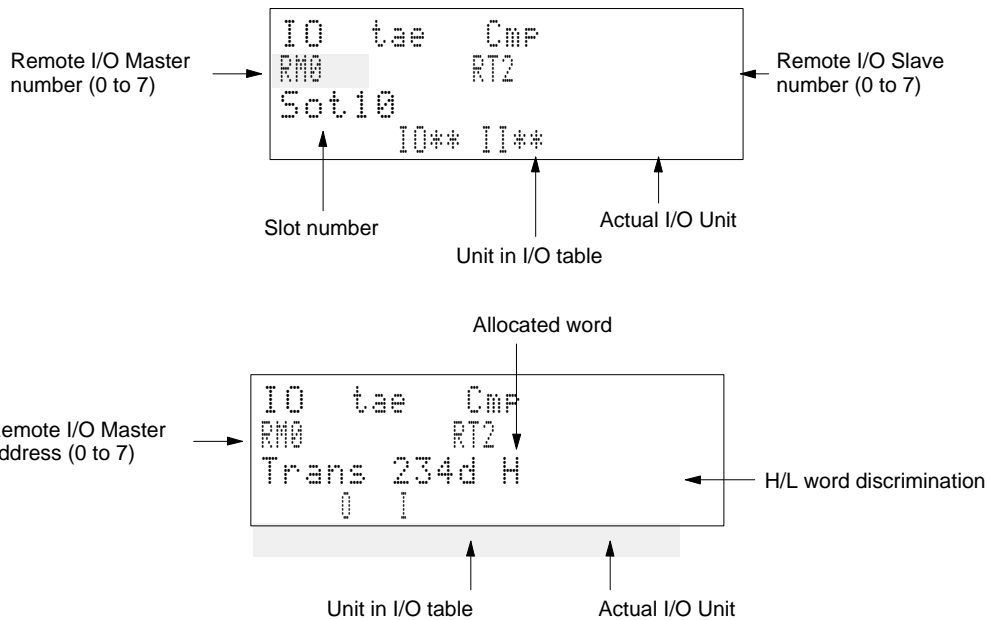
Optical I/O Units

Display	Unit
O	8 output points OUT
I	8 input points
	Duplication allocation
☀	No Unit

SYSMAC BUS/2 Units

CPU Bus Unit number · Slave group (3)

Remote I/O Master address (0 to 3)

```
IO   tae  Cmp
RM2⎕04 G3    RT5
Sot 10
      IO☀☀ II☀☀
```

Remote I/O Slave number (0 to 7)

Remote I/O Slave slot number · Unit in I/O table · Actual I/O Unit


```
IO   tae  Cmp
RM2⎕04 G3    RT5

RT status err
```

The type of Slave (58M/112M/54MH) is different.

Slave group number (0 to 1) · CPU Bus Unit number

Remote I/O Master Unit (0 to 3)

```
IO   tae  Cmp
RM1⎕03 G2
Unit 15
      0☀☀☀ 1☀☀☀
```

Slave Unit number · Unit in I/O table · Actual I/O Unit

4-4-46 Memory Clear

Use Used to clear all or part of the user program memory area. Always clear memory before inputting a new program when using ladder-only programming.

Modes

RUN	MONITOR	DEBUG	PROGRAM
No	See note.	See note.	OK

Note It is possible to clear the only designated data areas in the MONITOR or DEBUG mode.

Key Sequence
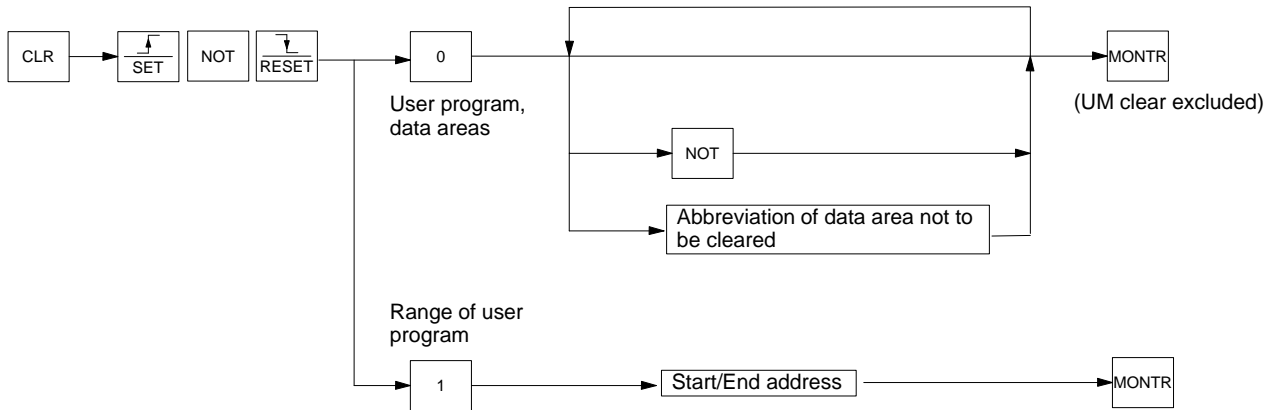
Operation

Either the entire program area and/or designated data areas can be deleted, or a specified range of the user program area can be deleted. When 0 is specified, either all or none of the program area must be cleared; a specified range cannot be cleared. When the entire program area is cleared, the program type will set to "ladder only" and will need to be changed to use SFC programming. Even though the entire program is cleared, an END(001) instruction is automatically generated.

Press the NOT Key to clear only data areas without clearing the program area.

The data areas that are displayed will be cleared. If there are any data areas that should not be cleared, input the data area abbreviation to delete the display for that area. Specify the EM bank by using the 0 to 7 Keys.

To clear a specified range, specify "1" and the start and end addresses. Do not set an address larger than the address that contains the END(001) instruction or an error will result and "Address too high" will be displayed. If the END(001) instruction is cleared, a program error will result if the user continues operation without adding an END(001) instruction.

The user's program will be cleared even if the UM is protected via Programming Device settings, but will not be cleared if it is protected with the key switch on the CPU Unit and "Read only" will be displayed.

Refer to *Appendix D Inputs for Instructions and Operands* for the key sequences to specify instructions and operands.

Note The buzzer will beep when the Set–Not–Reset Keys are pressed to warn that memory will be cleared if you continue with the operation. It is not an indication of an error in operation.

Example 1
In this example, all areas except DM are cleared. The program is not cleared.
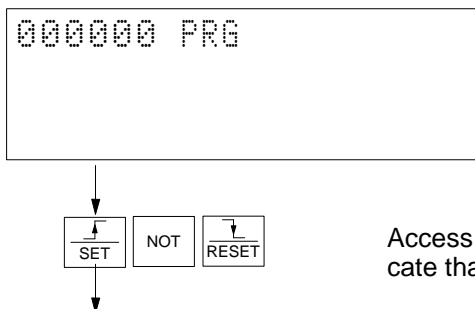
Access the menu of Clear Memory menu. The buzzer will beep to indicate that you are entering the memory clear operation.

```
Cear  memor
0:A UMIOM
1:UM Part
```

↓

```
0
```

Specify "All UM/IOM."

The displayed items will be cleared. EM Area banks will not be displayed for PCs that do not have the EM Area.

↓

```
Cear   memor
A UM
D A  G  TIM CNT
DM EM:012345
```

↓

```
NOT
```

Press the NOT Key to eliminate the program area (UM) from the deletion process.

↓

```
Cear   memor

D A  G  TIM CNT DM
EM:012345
```

↓

```
AC
DM
```

Press the DM Key to eliminate the DM area from the deletion process.

↓

```
Cear   memor

D A  G  TIM CNT
   EM:012345
```

↓

```
MONTR
```

Press the Monitor Key to start execution.

"End" will be displayed when the operation has been completed.

↓

```
Cear   End

D A  G  TIM CNT
   EM:012345
```

Example 2
In this example, action program AC 0001's addresses 000050 to 000120 are cleared. Use the program designation operation to designate program AC 0001 in advance. The initial display will appear when the Clear Key is pressed.

```
000000 PRG
```

↓

[SET] [NOT] [RESET]

Access the menu of Clear Memory menu. The buzzer will beep to indicate that you are entering the memory clear operation.

```
Cear  memor
0:A UMIOM
1:UM part
```

↓

[1]

Specify "UM part."

```
Cear   memor
AC000
Start :000000
End   :000000
```

↓

[5] [0] [↓]
[1] [2] [0]

Input the start address and end address.
Press the UP or Down Keys to move the cursor up or down.

```
Cear   memor
AC000
Start :000050
End   :000120
```

↓

[MONTR]

Press the Monitor Key to start execution.
"End" will be displayed when the operation has been completed.

```
Cear   End
AC000
Start :000050
End   :000120
```

4-4-47 Online Unit Replacement

Use   Used to replace an Input or Output Unit without stopping PC operation. Online Unit replacement can be performed for an Input Unit or Output Unit mounted to a CVM1D CPU Rack, an Expansion CPU Rack built on a CVM1D Backplane, or an Expansion I/O Rack. Online Unit replacement cannot be executed for Special I/O Units or CPU Bus Units.

Modes

RUN	MONITOR	DEBUG	PROGRAM
OK	OK	OK	OK

Key Sequence

CLR → FUN → SHIFT → CH/∗DM → 2   Select 2: Swap Unit from I/O Table Operations.

→ Rack No. → Slot No. → SRCH   Specify the Rack and slot number for which online Unit replacement is to be executed.

→ DEL → 0 (or 1) → ↑ → Replace the Unit.   Either clear or hold the output data from the Rack and replace the Units.

→ INS → ↓   End the replacement operation.
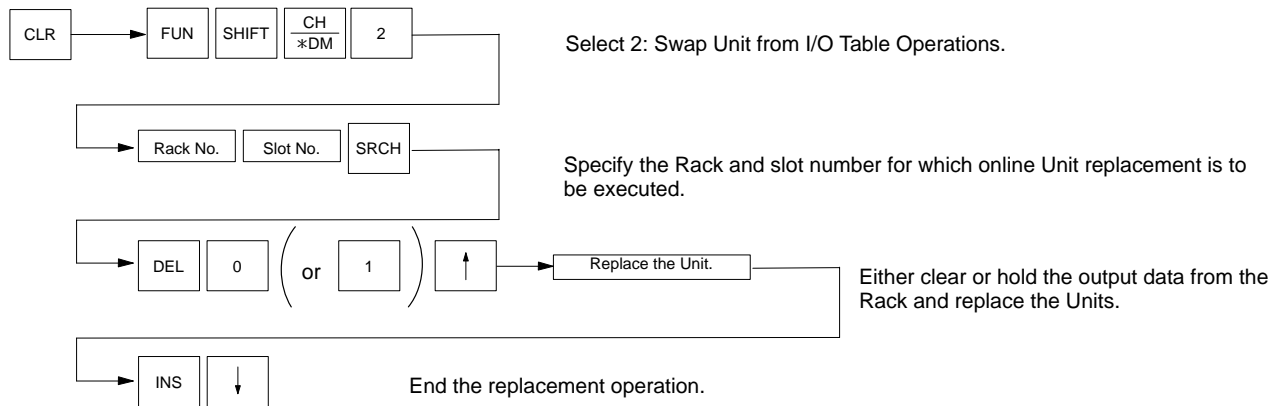
Note When executing online Unit replacement, either cut off the loads or hold the current status for the entire Rack. Online Unit replacement cannot be executed while cutting off the loads or holding status for individual Units.

During online Unit replacement, the inputs and outputs for the Units will be cut off. The ON/OFF status of the Rack will be maintained.

Only one Unit can be changed each time the operation is performed.

Online Unit replacement is possible in either the Duplex mode or the Simplex mode of the CVM1D Duplex System.

The status of online Unit replacement can be read from the display on the I/O Control Unit or the I/O Interface Unit.

I/O Control Unit
(CV500-IC101/201/301)

I/O Interface Unit
(CV500-II101/201)

Display

Display

Rack Mode 1   Slot number of Unit
                   being replaced.
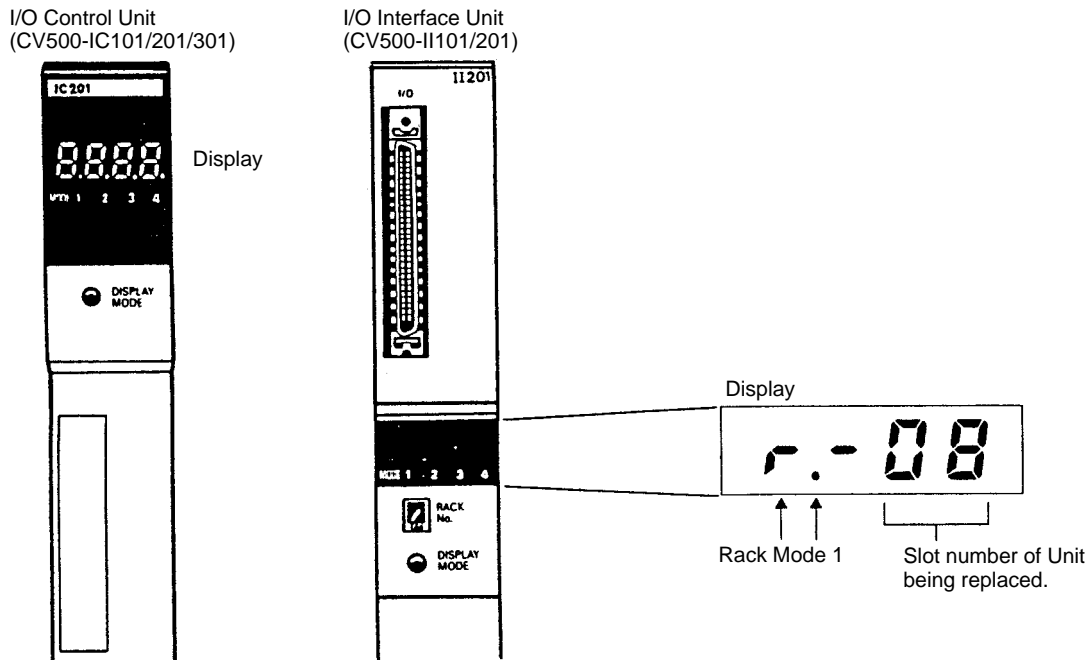
During online replacement the following information is output to the auxiliary area. Refer to the next table for the actual contents of A399.

Word address	Contents
A399	Online replacement status

The status of online Unit replacement will be output to A399 during online Unit replacement, as shown in the following table.

Address	Bit(s)	Contents	
A399	00 to 07	Slot number of Unit being replaced	00 to 09 (BCD, 2 digits)
	08 to 11	Rack number of Unit being replaced	0 to 9 (BCD, 1 digit)
	12 to 13	Not used.	
	14	Treatment of data from Input and Output Units other than the Unit being replaced.	0: Clear 1: Save
	15	Online Unit replacement status	0 Online Unit replacement not under execution 1: Online Unit replacement under execution

Operation   In this example, a 32-point Output Unit in slot 8 (counted from the left) of an
                                Expansion I/O Rack (Rack 5) is replaced. Use a Unit with the same model num-
                                ber and specifications as the Unit being replaced. The initial display will appear
                                when the Clear Key is pressed.

```
000000 PRG
```

[FUN] [SHIFT] [CH / *DM]   Access the I/O Table Menu.

```
IO  Tae
0:Create
1:Compare
2:Repace IO
```

[2]   Select 2: Swap Unit.

```
Repace  IO
Rac         _
Sot
```

[5]   Set the rack number.

                                The rack number for the CPU Rack is 0.
                                The rack numbers for Expansion CPU Racks and Expansion I/O
                                Racks are set with the I/O Interface Unit and are from 1 to 7.

```
Repace  IO
Rac       5
Sot         _
```

[8]   Set the slot number.

113

```
Repace  IO
Rac    5
Sot    8
```

(Slot number starts at 0 from left edge)

II201   Slot 8

Expansion I/O Rack

Rack number set to 5.
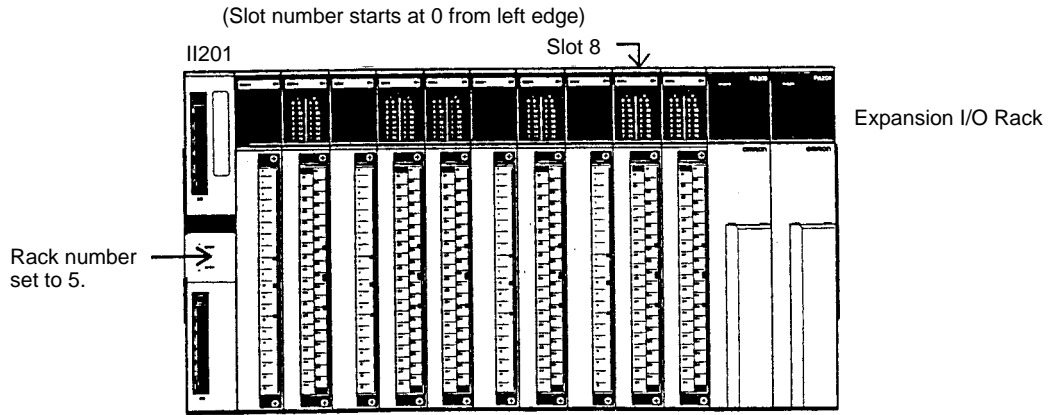
```
SRCH
```
Press the Search Key. The type of Unit connected will be displayed.

```
Repace  IO
Rac    5
Sot    8
     00**
```
Unit type

```
DEL
```
Press the Delete Key. The screen for selecting whether or not to execute the replacement will be displayed.

```
Repace  IO St
Rac OUT
0:Cear
1:Hod   00**
```

```
0
```
Press 0 to clear output data for the entire Rack. A confirmation screen for clearing output data from the Rack will be displayed.

```
Repace  IO St
Rac OUT
```

```
↑
```
If the Delete Key is pressed, the screen for selecting whether or not to execute replacement will return. If the Down Key is pressed, the output data will be cleared.

Press the Up Key to continue online Unit replacement.

```
Repace   IO
Rac     5
Sot     8
■       00**
```

The dark box at the start of the 4th line indicates that the Unit can be replaced.

Replace the Unit.

Slot 8

`r.-08`
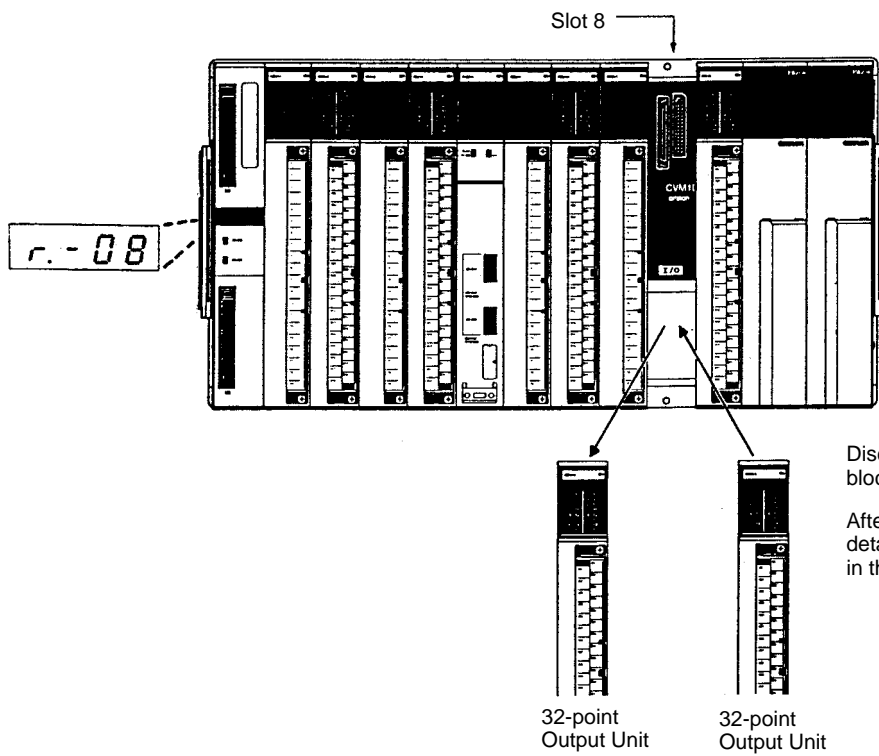
Disconnect the wiring from the detachable terminal block before replacing the Unit.

After replacement, be sure to connect the wiring for the detachable terminal blocks in the correct positions and in the correct directions.

32-point Output Unit   32-point Output Unit

INS

After the Unit has been replaced, end the online Unit replacement operation.

```
Repace   IO End
Rac     5
Sot     8
■       00**
```

↓

Press the Down Key. The screen for executing online Unit replacement will reappear.

```
Repace   IO
Rac
Sot
        00**
```

```
┌─────┐
│ CLR │
└─────┘
```

The initial display will appear when the online Unit replacement operation has been ended.

```
00000
```

SECTION 5
Error Processing

This section describes errors that can occur during Programming Console operation, including both errors directly relation to Programming Console operation and PC errors. Refer to the *CV-series PC Operation Manual: Ladder Diagrams* for further details on PC operation and errors.

5-1 Reading Errors

It is possible to read the following three kinds of errors.

- Communications errors
- Operating errors
- PC errors

Communications Errors

Messages for errors occurring in communications between the Programming Console and PC will be displayed automatically. It will be impossible to operate the Programming Console if there is a communications error. Continue operation after first eliminating the cause of the error.

```
000123 PRG
AND      I000012
OUT      Q001001
Comm err aitg
```
← Appears on the fourth line.

Operating Errors

An error message will appear if an operating error occurs during program editing or monitoring.

```
005123 MON
END(001)

Address too ig
```
← Appears on the fourth line.

PC Errors

PC errors are displayed in one of three ways: automatically, via the error read operation, and from the error history. Refer to sections starting on page 75 for details.

Error Automatic Read
When an error occurs, operation will be interrupted and an error message will appear automatically.

```
        ERROR
IO eri error


```

Read Errors/Clear
Error messages will appear when the Clear Key, Function–Monitor Keys, and 0 Key are pressed in sequence. Clear error messages after removing the causes of the errors.

```
Disp error Read
Memor error
PS error
IO eri error
```

Move the cursor to an error and press the Monitor Key to clear the error.

Error History Read
It is possible to read the data recorded in the error history area by pressing the Clear Key, Function–Monitor Keys, and 1 Key.

```
Histor    Read
Cce time oer
2 1 2 0
20:21:41
```

5-2 Error Messages

Error message	Probable cause	Possible correction
Cannot connect	Another Programming Device was already connected or a Programming Device with a higher priority was connected.	Do not connect more than one Programming Device to the CPU Unit or I/O Interface Unit.
Comm error	There is a communications error between the PC and Programming Console.	Check the destination Unit. Check the cable connections.
Comm err wait'g	Time ran out before a response has returned for a command.	
Response overflo	A communications error has occurred.	Check the communications path between the Programming Console and PC.

5-3 Operating Errors

Error message	Probable cause	Possible correction
# of data error	The source node did not have the designated number of data when the data was transferred from the memory card to PC or from the PC to memory card.	Check the user program. Check the number of send words. Check the file designation.
Add range exceed	The memory card does not have enough space.	Use another memory card. Delete unnecessary files.
Address too high	The user attempted to read in an area exceeding the highest address of the user program.	Check the input address and input the address correctly.
	The designated word and data area exceeds the possible range.	
Can't chg; active	No action can be changed because the designated step is active.	Check the operating condition of the PC.
File not found	The designated file does not exist in the memory card.	Check the file name.
Instr not found	The designated instruction does not exist in the user memory.	Check the user program.
Memory error	A parity or check sum error occurred in PC memory.	Correct the data in memory.
	The user program is not correct.	Check and correct the program with the CVSS/SSS or GPC and transfer the program to the PC again.
	An error occurred in the memory card.	Mount the memory card correctly.
	An access error occurred while the user was accessing the memory card.	Check and correct the data in the memory card.
	A checksum error occurred for the data in the memory card.	
Mode error	An operation was attempted in the wrong PC mode (RUN, MONITOR, DEBUG, PROGRAM).	Set the PC to the correct mode.
No access right	No data can be changed because data is being written by another Programming Device.	Write the data after the other Programming Device finishes writing the data.

Error message	Probable cause	Possible correction
No action block	No actions exist in the designated step.	Input the correct step number. Check the program.
No active steps	There are no active steps in the designated sheet.	Check the user program.
No areas	The data area designation is wrong.	Designate correctly.
	The designated program does not exist.	Check the program
No file device	No memory card is mounted in the CPU Unit.	Mount the memory card in the CPU Unit.
No program	The data in the user memory is corrupted or no program has been input. No main program exists or no program exists for AC0000.	Transfer the program to the PC again. Designate the program.
No space	The memory card does not have enough space.	Use another memory card. Delete unnecessary files.
No such data	The designated data does not exist.	Input the correct data.
No such step	The designated step does not exist in the user memory.	Check the user program.
Ope not found	The designated operand does not exist in the user memory.	Check the user program.
Parameter error	There is a communications error between the Programming Console and PC.	Check the communications path.
Protected	The user memory has been protected.	Release UM protect.
Protection error	The user has attempted to protect user memory that has been already protected. The user has attempted to clear the protection of the user memory that has not been protected.	Check the settings for UM protect.
Read-only	The memory card has been protected or is EEPROM or PROM.	Write-enable the memory card. Use a RAM memory card.
	The system has been protected with the protect keyswitch on the CPU Unit.	Release the protection of the system with the system protect keyswitch.
Sheet not found	The designated sheet does not exist in the program.	Input the correct sheet number.
Too many files	The number of files exceeds the permissible number of files that can be registered in the memory card.	Delete unnecessary files.
Trans not found	The designated transition does not exist in the user memory.	Check the user program.
Wrong password	A wrong password was input.	Input the correct password.

5-4 PC Errors

5-4-1 Initialization Errors

The following errors occur before program execution has been started. The POWER indicator will be lit and the RUN indicator will not be lit for any of these. The RUN output will be OFF for each of these errors. The alarm indicator (ALARM) will be ON for the I/O table verification error.

Programming Console error message	Probable cause	Flag(s)	Error code (A400)	Possible remedy
CPU wait	Start input on CPU Rack's Power Supply Unit is OFF.	A30600	None	Short start input terminals on CPU Unit's Power Supply Unit.
CPU wait	Power to Slave is off or terminator is missing.	A30602	None	Check power supply to Slaves and terminator setting.
CPU wait	Initializing CPU Bus Unit or SYSMAC BUS/2 terminator missing.	A30603	None	Wait for CPU Bus Unit Initialization to be completed. Check power supply to Slaves and terminator setting.
CPU wait	A Unit has been changed making the I/O table incorrect. (PC Setup)	A30601 A40209	00E7	Check the I/O table from the CVSS/SSS and either connect Dummy I/O Units or correct the I/O table.

Note The I/O table verification error can be set as an initialization error in the PC Setup.

5-4-2 Non-fatal Operating Errors

The following errors occur after program execution has been started. PC operation and program execution will continue after one or more of these errors have occurred. For each of these errors, the POWER, RUN, and ALARM indicators will be lit and the ERROR indicator will not be lit. The RUN output will be ON.

Programming Console error message	Probable cause	Flag(s)	Error code (A400)	Possible remedy
System FAL	FAL(006) has been executed in program. Check the FAL number to determine conditions that would cause execution (programmed by user).	A40215	4101 to 42FF correspond to FAL numbers 001 to 511.	Correct according to cause indicated by FAL number (set by user).
JMP error	No JMP(004) in the program.	A40213	00F9	Check and correct the program.
DM address error	Indirectly addressed DM address data is not BCD.	A40212	00F8	
SFC cont error	An error has occurred during SFC execution.	A40211	00F4	Check and correct the program. (Refer to the next table for information on non-fatal SFC errors.)
I/O verify error	Unit has been removed making I/O table incorrect.	A40209	00E7	Check the I/O table from the CVSS/SSS and either connect Dummy I/O Units or correct the I/O table.
CPU Bus Unit error	A parity error has occurred in the transfer of data between the CPU Unit and a CPU Bus Unit or in the CPU Bus Link.	A40207	0200 to 0215 or 0231[1]	Check the errant Unit.
SYSMAC BUS/2 error	An error has occurred between a Master and Slave Rack.	A40206	00B0 to 00B3 (Masters #0 to #3.)[2]	Verify that the Slave Rack is operating properly, and check the transmission line.
SYSMAC BUS error	An error has occurred between a Master and Slave Rack.	A40205	00A0 to 00A7 (Masters #0 to #7.)[3]	Verify that the Slave Rack is operating properly, and check the transmission line.

Programming Console error message	Probable cause	Flag(s)	Error code (A400)	Possible remedy
Battery error	CPU Unit or Memory Card backup battery is missing or its voltage has dropped.	A40204	00F7	Check battery and replace if necessary.[4]
CPU SIO set error	The registered CPU Bus Unit number doesn't agree with the registered unit number.	A40203	0400 to 0415 or 0431[5]	Check the errant Unit.
PWR off	A momentary power interruption can be set as an non-fatal error in the PC Setup.	A40202	0002	Check the power supply voltage and lines.

Note
1. Error codes 0200 to 0215 indicate CPU Bus Units #00 to #15, respectively, while 0231 indicates a CPU bus link error. Also, A422 contains the errant Unit's number, and A42315 is turned ON to indicate a CPU bus link error.

2. A424 contains the unit number of the Master involved, and A480 to A499 contain the unit number of the Slave involved.

3. A425 contains the unit number of the Master involved, and A470 to A477 contain the unit number of the Slave involved.

4. A42615 is turned ON to indicate a battery error, and A42614 is turned ON to indicate a Memory Card battery error.

5. Error codes 0400 to 0415 indicate CPU Bus Units #00 to #15, respectively, while 0431 indicates a CPU Bus Link error. Also, A427 contains the errant Unit's number.

5-4-3 Fatal Operating Errors

The following errors occur after program execution has been started. PC operation and program execution will stop and all outputs from the PC will be turned OFF when any of the following errors occur.

None of the CPU Unit indicators will be lit for the power interruption error, and only the POWER indicator will be lit for the Expansion Rack power interruption error. The POWER and WDT indicators will be lit for the CPU Unit error. For all other fatal operating errors, the POWER and ERROR indicators will be lit. The RUN output will be OFF.

Programming Console error message	Probable Cause	Flag(s)	Error code (A400)	Possible remedy
(No message is displayed)	A power interruption longer than the momentary power interruption time has occurred.	None	None	Check power supply voltage and power lines. Try to power-up again.
Expans Rack OFF	A power interruption to an Expansion Rack has occurred.	None	None	Turn on the power supply to the Expansion CPU and Expansion I/O Racks.
CPU error	Watchdog timer has exceeded maximum setting.	None	80FF	Turn the power OFF and restart.
Memory error	An error has occurred during check of the PC or EM Unit memory, or program transfer could not be completed at start-up.	A40115	80F1	Check Memory Card and EM Unit to make sure they are mounted and backed up properly. Perform a Program Check Operation to locate cause of error. If error not correctable, try inputting program again.

Programming Console error message	Probable Cause	Flag(s)	Error code (A400)	Possible remedy
I/O bus error	Error has occurred in the bus line between the CPU Unit and I/O Units.	A40114	80C0 to 80C7 or 80CE or 80CF[1]	Check cable connections between the I/O Units and Racks. Check I/O Units for improper mounting or damage. Verify that terminators are connected, and then clear the error. A404 contains the errant rack/slot number.
Dup I/O error	The same number has been allocated to more than one Expansion Rack, more than one CPU Bus Unit, or one I/O word has been allocated to more than one I/O Unit.	A40113	80E9	After checking the rack numbers, unit numbers, or word allocation, turn the relevant power supply ON, OFF, and ON again, and then clear the error. A409 contains the duplicate rack number, and A410 contains the duplicate unit number.
CPU bus error	Error has occurred in the bus line between the CPU Unit and CPU Bus Units, or watchdog timer error has occurred with a CPU Bus Unit.	A40112	8100 to 8115 (indicate Units 0 to 15)	Check cable connections between the CPU Unit and CPU Bus Units, and then clear the error. A405 contains the errant unit number.
Too many I/O	Maximum number of I/O points or I/O Units has been exceeded in the I/O Table.	A40111	80E1	Check the number of points with I/O Table Read. If necessary, reduce number of Units in the system to keep within maximum number of I/O points and register the I/O table again.[2]
I/O set error	Input and output word designations registered in I/O table do no agree with input/output words required by Units actually mounted.	A40110	80E0	Check the I/O table with I/O Table Verification operation and check all Units to see that they are in correct configuration. When the system has been confirmed, register the I/O table again.
Program error	END(001) is not written anywhere in program or the program exceeds memory capacity.	A40109	80F0	Correct the program and then clear the error.
Cycle time over	The cycle time has exceeded the maximum cycle time set in the PC Setup.	A40108	809F	Change the program or the maximum cycle time.[3]
SFC stop error	The program contains an error in SFC syntax.	A40107	80F3	Correct the program and then clear the error. (Refer to the next table for information on SFC fatal errors.)
System FALS	FALS has been executed by the program. Check the FAL number to determine conditions that would cause execution.	A40106	C101 to C2FF correspond to FAL numbers 001 to 511.	Correct according to cause indicated by FAL number.

Note 1. Error codes 80C0 to 80C7 indicate Rack numbers 0 to 7, respectively. Error codes 80CE and 80CF indicate that the terminator is missing in operating system 0 and 1, respectively.

2. The total number of I/O words allocated to the CPU, CPU Expansion, and Expansion I/O Racks is contained in A407. A408 contains the total number of I/O words allocated to the SYSMAC BUS/2 System, and A478 contains the total number of I/O words allocated to the SYSMAC BUS System.

3. The maximum cycle time since start-up is contained in A462 and 463, and the present cycle time is contained in A464 and 465.

Appendix A
Standard Models

Name	Specifications	Model number
CV-series Programming Console	Connectable to CV-series CPU Unit, CVM1 CPU Unit, CVM1D Duplex Unit, I/O Interface Unit, or SYSMAC BUS/2 Remote I/O Slave Unit. Connecting Cable sold separately (see below).	CVM1-PRS21-EV1
Memory Cassette	Installed in the CVM1-PRS21-EV1 Programming Console.	CVM1-MP201-V1
Connecting Cable	2 m	CV500-CN224
	4 m	CV500-CN424
	6 m	CV500-CN624

Appendix B
Specifications

Item	Specification
Power supply	300 mA max. at 5 VDC +5%/–10% (supplied from the connecting Unit)
Ambient operating temperature	0 to 55°C
Ambient operating humidity	10% to 90% with no condensation
Ambient operating environment	Must be free of corrosive gas.
Ambient storage temperature	−20 to 75°C
Display	Semitransparent LCD with LED illumination 16 characters x 4 lines, 5 x 7 dots/character
Weight	370 g
Dimensions	96 x 192 x 48.5 mm (W x H x D)
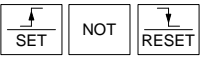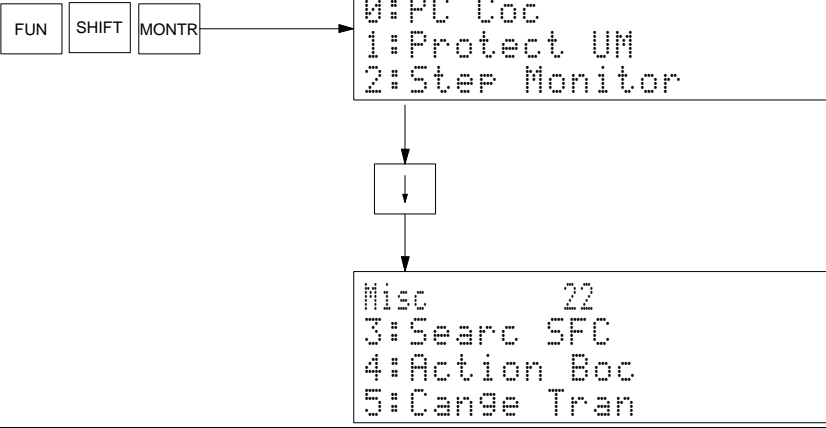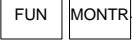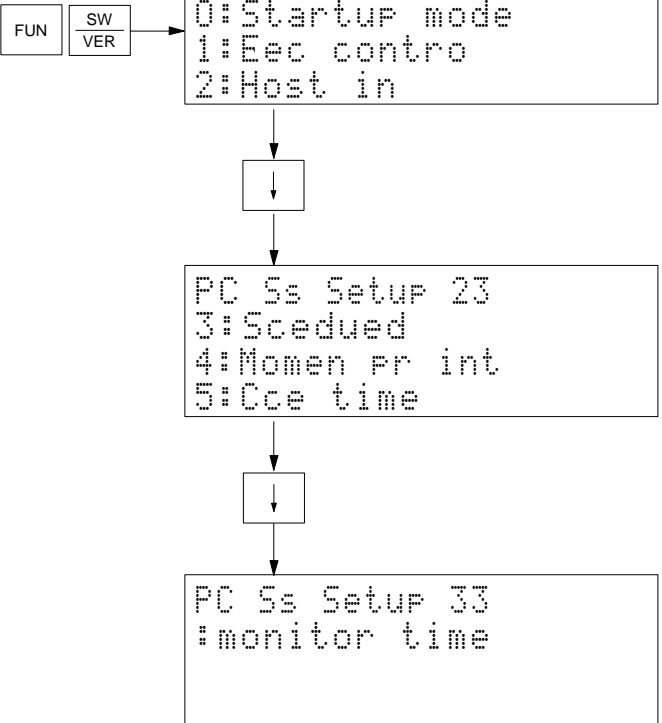
Appendix C
Programming Console Operations

Press the CLEAR Key and return to the initial menu before continue operating.

Main Key Sequence

Operation	Key sequences and displays	Page

System settings	`CVM1⌷PRS21⌷V1  20`
	`Mon time:02sec`
	`Message #: 0`
	`Read setup Yes`
	
	[CLR] ↓ [SHIFT] [CLR] ↑
		27

Program name read	`CV500⌷CPU01⌷V2`
	`Ladder`
	`Main`
	
	[CLR] ↓ [SHIFT] [CLR] ↑
		19

Program editing	`000000 PRG`	31

PC mode change	[SHIFT] [SW/VER] ↓↑ [CLR] ↑
	
	`Mode:     0:PRG`
	` PRG      1:DBG`
	`          2:MON`
	`          3:RUN`
		31

Simple I/O monitor	[MONTR] ↓ [SHIFT] [SW/VER] ↑ `ord    0000_`	44

Operation	Key sequences and displays	Page
Changing to programming operation	Other operations → SHIFT CLR → Programming operation / CLR ... CLR	---

Program designation	FUN CHG →
	
	`Ladder    Main`
	`Seect  program`
	`0:Main program`
	`1:Interrupts`
	
	`SFC+LaddAC0012`
	`Seect  programs`
	`0:Action`
	`1:Tran`
		29

Monitoring	Programming operation `000000 MON`	48

I/O multipoint monitor	Set items to monitor. → MONTR → SHIFT CLR SHIFT MONTR
	
	`00014     0125`
	`I000101    0FF`
	`D05000    2A5F`
	` 0030     12`
		48

4-word monitor	SW / VER
	
	CLR
	
	`Q001     0108`
	`I001     0000`
	`00015    0052`
	`00014    0125`
		51

Word monitor	SHIFT MONTR CLR	53
Differential monitor	`00001    Read` `0001000000S110R1`	55
	SHIFT ↑ → `I000101   ¿ 0FF` SHIFT ↓ →	
Force-set/ reset	SET RESET NOT	56

Operation	Key sequences and displays	Page
Memory clear	SET NOT RESET → `Cear memor` `o:A UMIOM` `1:UM Part`	107
Special operations	FUN SHIFT MONTR → `Misc     12` `0:PC Coc` `1:Protect UM` `2:Step Monitor` ↓ `Misc     22` `3:Searc SFC` `4:Action Boc` `5:Can9e Tran`	63
Cycle time display	MONTR → `Cce    time` `A9:     50ms` `Ma:     000ms` `Min:     00ms`	75
Error data read	FUN MONTR → `Read Errors` `0:Error dispa` `1:Histor` `2:Messa9es`	76
PC setup	FUN SW/VER → `PC Ss Setup 13` `0:Startup mode` `1:Eec contro` `2:Host in` ↓ `PC Ss Setup 23` `3:Scedued` `4:Momen pr int` `5:Cce time` ↓ `PC Ss Setup 33` `:monitor time`	82

Operation	Key sequences and displays	Page
Memory card operations	[FUN] [SHIFT] [CONT #] → ```Memor Card Oper 0:Tran 1:Comp 2:Format 3:Erase```	92
I/O table operations	[FUN] [SHIFT] [CH *DM] → ```IO tae 0:Create 1:Compare 2:Repace IO```	102

Key Sequences

All basic key sequences here beginning with the initial menu omits the CLEAR Key input.

Initial Settings

Operation	Mode	Key sequence	Page
System settings	RUN, MON, DEB, PRO	Initial menu of the Programming Console → NOT → [↓] [↑] (Select item) → [0] to [9] (Input value) → WRITE	27
Program designation	RUN, MON, DEB, PRO	**Ladder Only** (Main): FUN CHG → [0] → WRITE → CLR. (Interrupt): FUN CHG → [1] → Interrupt No. → WRITE → CLR; [1] (Not newly created.) ← [0] (Newly created.). **SFC and Ladder** (Action): FUN CHG → [0] → Program No. → WRITE → CLR; (Transition): [1]; [1] (Not newly created.) ← [0] (Newly created.)	29
PC mode change	RUN, MON, DEB, PRO	Main menu → SHIFT [SW/VER] (Move to sub-menu) → [0] (PROGRAM) → WRITE; [1] (DEBUG); [2] (MONITOR); [3] (RUN). SHIFT [SW/VER] or CLR → Main menu	31
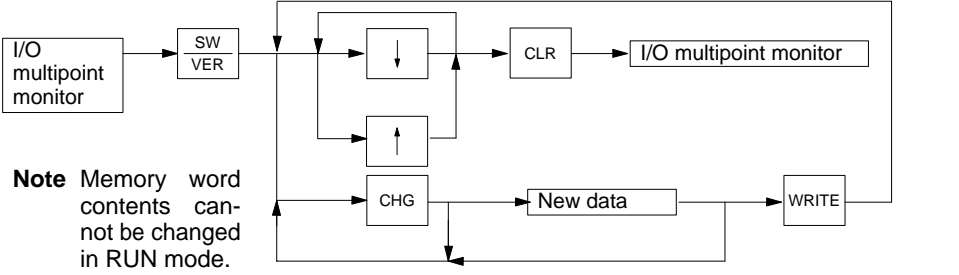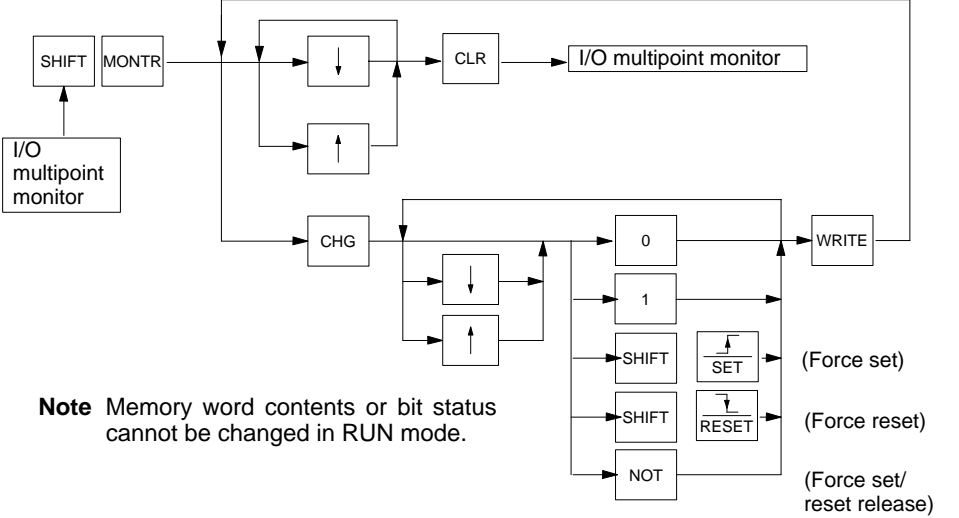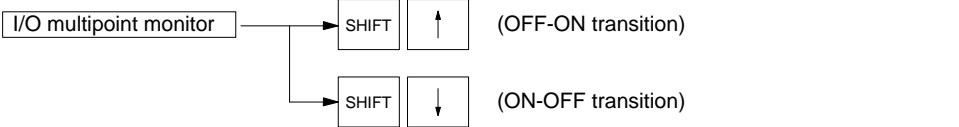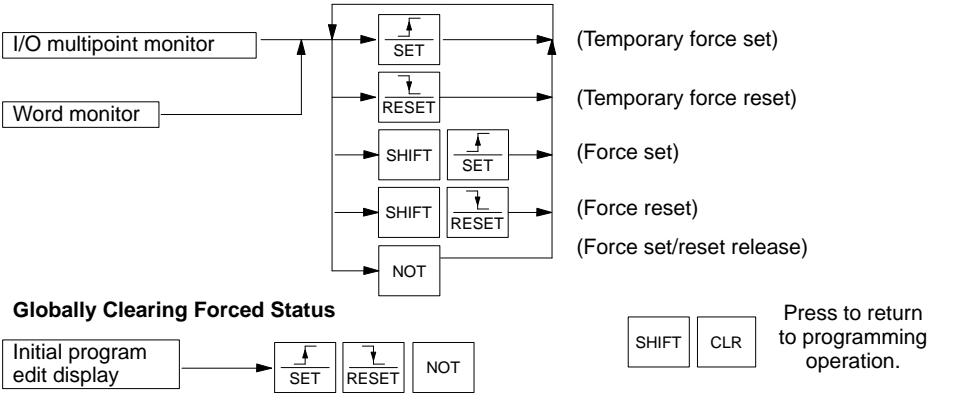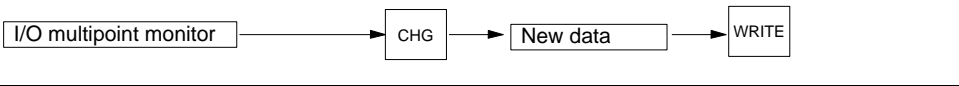
Program Editing

Operation	Mode	Key sequence	Page
Address set	RUN, MON, DEB, PRO	Input address	32
Program read	RUN, MON, DEB, PRO	Address set → [↓] / [↑]	33
Instruction search	RUN, MON, DEB, PRO	Address set → Instruction → SRCH → CLR; Program read	35

Operation	Mode	Key sequence	Page
Operand search	RUN, MON, DEB, PRO	Address set → Operand → SRCH → CLR (loop)	36
Instruction overwrite	MON, DEB, PRO	Program read / Address set → Instruction/operand → WRITE (In DEBUG or MONITOR mode: WRITE WRITE)	38
Instruction insert	MON, DEB, PRO	Program read / Address set → Instruction/operand → INS → ↓	39
Instruction delete	MON, DEB, PRO	Program read / Address set → DEL → ↑	41
Timer/ Counter SV change	MON, DEB, PRO	TIM/CNT instruction read → CHG → Set value → WRITE; SHIFT [CH/∗DM] → Data area abbreviation → Word address; [AC/DM] → DM address; SHIFT [EM/GR] → EM address (Set via memory contents); CHG → ↓ / CLR → ↑ (Increment/decrement)	42

Monitoring

Operation	Mode	Key sequence	Page
Simple I/O monitor	RUN (See note.), MON, DEB, PRO	Main menu → SHIFT [SW/VER] → MONTR (Transition to submenu) → Specify items to monitor → MONTR; SHIFT [SW/VER] (return to Main menu). **Note** Memory word contents or bit status cannot be changed in RUN mode.	44
I/O multipoint monitor	RUN, MON, DEB, PRO	Initial program edit display → Specify items to monitor → MONTR → Monitor menu; SHIFT CLR; SHIFT MONTR. **Note** Memory word contents or bit status cannot be changed in RUN mode.	48

Operation	Mode	Key sequence	Page
4-word monitor	RUN (See note.), MON, DEB, PRO	I/O multipoint monitor → SW VER → [↓] → CLR → I/O multipoint monitor; [↑]; CHG → New data → WRITE. **Note** Memory word contents cannot be changed in RUN mode.	51
Word monitor	RUN (See note.), MON, DEB, PRO	SHIFT MONTR ← I/O multipoint monitor; → [↓] → CLR → I/O multipoint monitor; [↑]; CHG → [↓] [↑] → 0 → WRITE; 1; SHIFT SET (Force set); SHIFT RESET (Force reset); NOT (Force set/reset release). **Note** Memory word contents or bit status cannot be changed in RUN mode.	53
Differential monitor	RUN, MON, DEB, PRO	I/O multipoint monitor → SHIFT [↑] (OFF-ON transition); SHIFT [↓] (ON-OFF transition)	55
Force set/reset	MON, DEB, PRO	I/O multipoint monitor; Word monitor → SET (Temporary force set); RESET (Temporary force reset); SHIFT SET (Force set); SHIFT RESET (Force reset); NOT (Force set/reset release). **Globally Clearing Forced Status** Initial program edit display → SET RESET NOT. SHIFT CLR Press to return to programming operation.	55
PV change	MON, DEB, PRO	I/O multipoint monitor → CHG → New data → WRITE	59

Operation	Mode	Key sequence	Page
ASCII display	RUN (See note.), MON, DEB, PRO	I/O multipoint monitor menu → [, –IR / TR] ; or → [CHG] → [New data] → [WRITE]. **Note** Memory word contents cannot be changed in RUN mode.	59
Decimal display	RUN (See note.), MON, DEB, PRO	I/O multipoint monitor menu → [, –IR / TR] ; or → [CHG] → [↓ SET] (+) / [↑ RESET] (–) → [New data] → [WRITE]. **Note** Memory word contents cannot be changed in RUN mode.	62

Special Operations

Operation	Mode	Key sequence	Page
Clock read/change	RUN, MON, DEB, PRO	**Clock Read** [FUN] [SHIFT] [MONTR] → [0] **Clock Change** [Clock read] → [CHG] → [↓] / [↑] → [New value] → [WRITE]	63
UM protect	RUN, MON, DEB, PRO	[FUN] [SHIFT] [MONTR] → [1] → [0] → [0] / [1] → [Password] → [WRITE] ; [1]	65
Active step monitor	RUN, MON, DEB, PRO	[FUN] [SHIFT] [MONTR] → [2] → [Sheet number] → [MONTR] → [↓] / [↑]	67
SFC search	RUN, MON, DEB, PRO	[FUN] [SHIFT] [MONTR] → [3] → [↓] / [↑] → [Bit and word address] → [SRCH]	68

Operation	Mode	Key sequence	Page
Action block change	RUN (See note.), MON, DEB, PRO	FUN → SHIFT → MONTR → ↓ ↑ → 4 → Step number → MONTR → CLR. The actions in the action block will be read in sequence. ↓ ↑ → CHG → ↓ ↑ → CHG → WRITE (AQ change); Set value / Word setting (Set value change); Action number / Address set (Action change); Bit address (FV change/deletion). **Note** Data can only be monitored in RUN mode; it cannot be changed.	71
Transition change	RUN (See note.), MON, DEB, PRO	FUN → SHIFT → MONTR → ↓ ↑ → 5 → Transition number → MONTR → CLR. CHG → Bit address → WRITE / DEL. **Note** Data can only be monitored in RUN mode; it cannot be changed.	74

PC Data Read
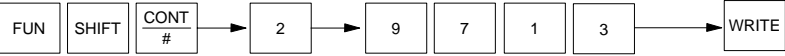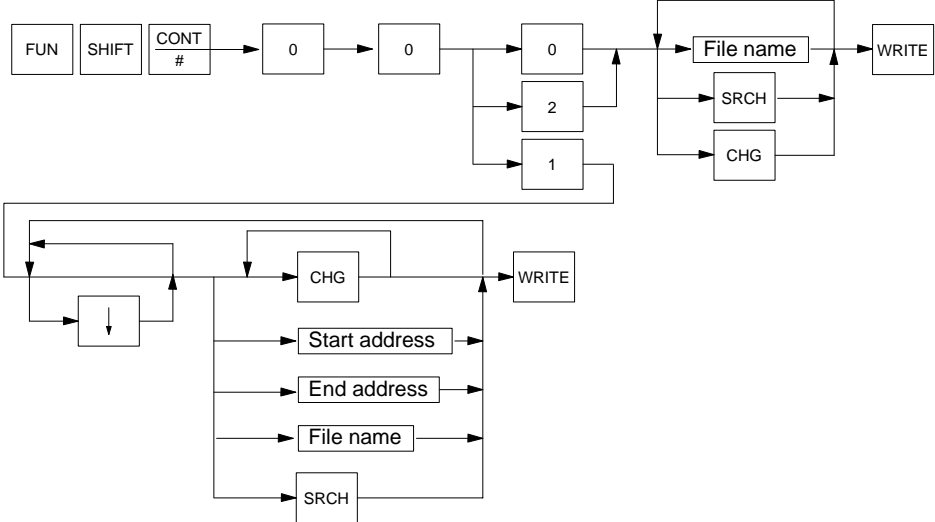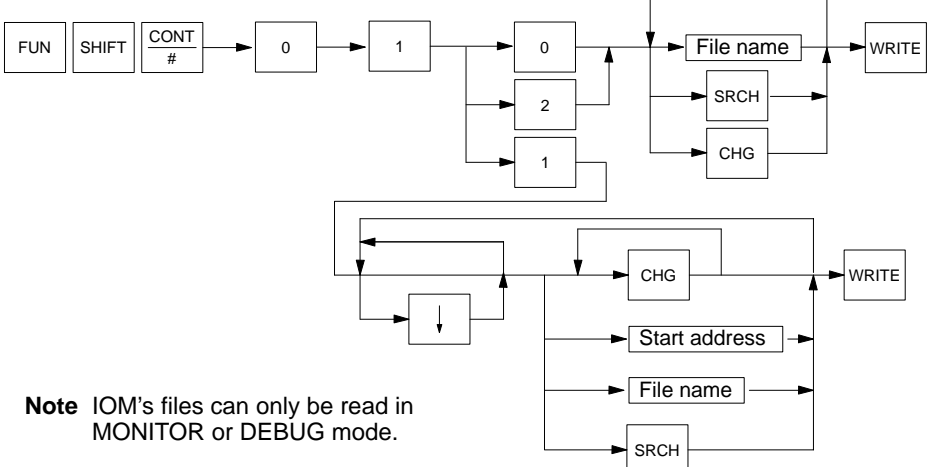
Operation	Mode	Key sequence	Page
Cycle time display	RUN, MON	MONTR	75
Error automatic display	RUN, MON, DEB, PRO	Error automatic display → SRCH → ↓ ↑ CHG → MONTR → CLR → Returns to the initial menu. (Selection of the FAL/FALS number and message) **Note** Errors can be cleared in PROGRAM mode only.	76

Operation	Mode	Key sequence	Page
Error display/ clear	RUN, MON, DEB, PRO	FUN → MONTR → 0 → (↓ / ↑ / CHG) → MONTR (loop back). (Selection of the FAL/FALS number and message). **Note** Errors can be cleared in PROGRAM mode only.	79
Error history read	RUN, MON, DEB, PRO	FUN → MONTR → 1 → (↑ / ↓)	80
Message read	RUN, MON, DEB, PRO	FUN → MONTR → 2 → (↓ / ↑ / MONTR → Clear message)	81

PC Setup

Operation	Mode	Key sequence	Page
Startup mode	MON, DEB, PRO	FUN → SW/VER → 0 → 0 to 3 → WRITE	82
Execution controls (Duplicate action error, Comparison error process)	MON, DEB, PRO	FUN → SW/VER → 1 → (0 / 1) → (0 / 1) → WRITE	84
Host link	MON, DEB, PRO	FUN → SW/VER → 2 → 0 to 2 / ↓ → (3 → Menu selection → WRITE) / (4 → Unit number input)	85
Scheduled interrupt setting	MON, DEB, PRO	FUN → SW/VER → ↓ → 3 → 0 to 2 → WRITE	87
Momentary power interruption time	MON, DEB, PRO	FUN → SW/VER → ↓ → 4 → 0 to 9 → WRITE	88
Cycle time	MON, DEB, PRO	FUN → SW/VER → ↓ → 5 → (0 → WRITE) / (1 → WRITE → Cycle time input → WRITE → WRITE)	89
Cycle time monitoring time	MON, DEB, PRO	FUN → SW/VER → ↓ → ↓ → 6 → time input → WRITE	90

Memory Card Operations

Operation	Mode	Key sequence	Page
Memory card format	RUN, MON, DEB, PRO	FUN → SHIFT → CONT/# → 2 → 9 → 7 → 1 → 3 → WRITE	92
File write	RUN, MON, DEB, PRO	FUN → SHIFT → CONT/# → 0 → 0 → [0 / 2 / 1] → [File name / SRCH / CHG] → WRITE; [↓] → CHG → WRITE; Start address; End address; File name; SRCH	93
File read	MON (See note.), DEB (See note.), PRO	FUN → SHIFT → CONT/# → 0 → 1 → [0 / 2 / 1] → [File name / SRCH / CHG] → WRITE; [↓] → CHG → WRITE; Start address; File name; SRCH	97

Note IOM's files can only be read in MONITOR or DEBUG mode.

Operation	Mode	Key sequence	Page
File compare	RUN, MON, DEB, PRO	FUN → SHIFT → CONT # → 1 → 0/2/1 → File name → SW/VER; SRCH; CHG. CHG → SW/VER; Start address; File name; SRCH (with ↓ loop)	99
File delete	RUN, MON, DEB, PRO	FUN → SHIFT → CONT # → 3 → 0/2/1 → File name → WRITE → WRITE; SRCH; CHG. CHG → WRITE → WRITE; File name; SRCH	100

I/O Table Operations

Operation	Mode	Key sequence	Page
I/O table create	PRO	FUN → SHIFT → CH/*DM → 0 → 9 → 7 → 1 → 3 → WRITE → 0/1; CHG	102
I/O table compare	RUN, MON, DEB, PRO	FUN → SHIFT → CH/*DM → 1 → CLR; SW/VER	104

Memory Clear

Operation	Mode	Key sequence	Page
Clear memory	MON (See note.), DEB (See note.), PRO	**Note** The program cannot be cleared in MONITOR or DEBUG mode.	107

Online Unit Replacement

Operation	Mode	Key sequence	Page
Online Unit Replace-ment	RUN, MON, DEB, PRO		111

Appendix D
Inputs for Instructions and Operands

The following tables show the key strokes required to input instructions and operands.

Instructions

Instruction	Keys	Remarks
LOAD	LD	---
AND	AND	---
OR	OR	---
LOAD NOT	LD NOT	Displayed as LD N.
AND NOT	AND NOT	Displayed as AND N.
OR NOT	OR NOT	Displayed as OR N.
AND LOAD	AND LD	---
OR LOAD	OR LD	---
NOT	FUN 0 1 0	Cannot be input using the NOT Key.
Differentiate up instructions	SHIFT SET	The same keys will toggle the specification.
Differentiate down instructions	SHIFT RESET	
Immediate refresh instructions	TEXT !	
OUTPUT	OUT	---
OUTPUT NOT	OUT NOT	Displayed as OUT N.
TIMER	TIM	---
COUNTER	CNT	---
SHIFT REGISTER (SFT(050))	SFT	Can also be input via function code 050.
Special instructions with function codes (see next line for block programming instructions)	FUN 0 to 9 Three digits of function code	All leading zeros in function codes must be input.
Block programming instructions	SHIFT FUN 0 to 9 Three digits of function code	All leading zeros in function codes must be input.

Operands

Data area abbreviations set in custom operations from the CVSS/SSS or other Programming Devices are not displayed and cannot be used from the Programming Console. Custom data areas and abbreviations for them, however, can be used (as can the default addresses). Press the Shift+Text Keys before inputting custom data area abbreviations and then press the Clear Key before inputting an address.

Item	Bit/Flag addresses	Word addresses	Display	Remarks
Specifying bit/word input	[SHIFT] [CONT/#]	[SHIFT] [CH/∗DM]	---	---
Custom data area address	[SHIFT] [TEXT/!] *Abbreviation* [CLR] *Bit address*	[SHIFT] [TEXT/!] *Abbreviation* [CLR] *Word address*		
CIO Area	*Bit address*	*Word address*	I (inputs) Q (outputs)	I/O classification is displayed after I/O table is created.
CPU Bus Link Area	[EM/GR] *Bit address*	[EM/GR] *Word address*	G	---
Auxiliary Area	[∗EM/AR] *Bit address*	[∗EM/AR] *Word address*	A	---
TR Area	[, −IR/TR] *Bit address*		---	---
Timer Area	[TIM] *Timer number*	[TIM] *Timer number*	T	Indicate the Completion Flag and the PV.
Counter Area	[CNT] *Counter number*	[CNT] *Counter number*	C	Indicate the Completion Flag and the PV.
Constant	---	[CONT/#] *Value*	#	Press the Number (#) Key to switch from word input to constant input. A to F can be input by pressing the SHIFT together with the 0 to 5 Keys.
DM Area, direct address	---	[AC/DM] *DM address*	D	---
DM Area, indirect address	---	[CH/∗DM] *DM address*	∗D	∗ is not displayed when monitoring.
EM Area, direct address	---	[SHIFT] [EM/GR] *EM address*	E	---
EM Area, indirect address	---	[SHIFT] [∗EM/AR] *EM address*	∗E	∗ is not displayed when monitoring.
Data registers	---	[SHIFT] [DR/IR] *DR address*	DR	---
Index registers (IR), direct address	---	[DR/IR] *IR address*	IR	---
IR, indirect address	---	[, IR+/, IR] *IR address*	,IR	---
IR, constant offset	---	[, IR+/, IR] *IR address* [RESET] or [SET] *Offset* (− offset) (+ offset)	+,IR −,IR	---
IR, data register offset	---	[, IR+/, IR] *IR address* [SHIFT] [DR/IR] *DR address*	DR ,IR	---

Item	Bit/Flag addresses	Word addresses	Display	Remarks
IR, auto-increment	---	[SHIFT] [, IR+ / , IR] *IR address*	,IR+	---
		[SHIFT] [, IR+ / , IR] [SHIFT] [, IR+ / , IR] *IR address*	,IR++	
IR, auto-decrement	---	[SHIFT] [, −IR / TR] *IR address*	,−IR	---
		[SHIFT] [, −IR / TR] [SHIFT] [, −IR / TR] *IR address*	,−−IR	
Step	[TN / ST] *Step number*	---	ST	---
Transition	[SHIFT] [TN / ST] *Transition number*	---	TN	---
Action	[SHIFT] [AC / DM] *Action number*	---	AC	---

Appendix E
Alphabetic List of Instructions by Function Code

Sequence Control, Error Handling, and Step Control Instructions

Code	Mnemonic	Name
000	NOP	NO OPERATION
001	END	END
002	IL	INTERLOCK
003	ILC	INTERLOCK CLEAR
004	JMP	JUMP
005	JME	JUMP END
006	FAL(↑)	FAILURE ALARM
007	FALS	FAILURE ALARM
008	STEP	STEP DEFINE
009	SNXT	STEP START

Sequence I/O Instructions

Code	Mnemonic	Name
010	NOT	NOT
011	KEEP(!)	KEEP
012	CNTR	REVERSIBLE COUNTER
013	DIFU(!)	DIFFERENTIATE UP
014	DIFD(!)	DIFFERENTIATE DOWN
015	TIMH	HIGH-SPEED TIMER
016	SET(!↑↓)	SET
017	RSET(!↑↓)	RSET
018	UP*	CONDITION ON
019	DOWN*	CONDITION OFF

Data Compare Instructions

Code	Mnemonic	Name
020	CMP(!↑)	COMPARE
021	CMPL	DOUBLE COMPARE
022	BCMP(↑)	BLOCK COMPARE
023	TCMP(↑)	TABLE COMPARE
024	MCMP(↑)	MULTIPLE COMPARE
025	EQU(↑)	EQUAL
026	CPS(!)*	SIGNED BINARY COMPARE
027	CPSL*	DOUBLE SIGNED BINARY COMPARE
028	CMP(!)*	UNSIGNED COMPARE
029	CMPL*	DOUBLE UNSIGNED COMPARE

Data Move and Sequence Output Instructions

Code	Mnemonic	Name
030	MOV(!↑)	MOVE
031	MVN(↑)	MOVE NOT
032	MOVL(↑)	DOUBLE MOVE
033	MVNL(↑)	DOUBLE MOVE NOT
034	XCHG(↑)	DATA EXCHANGE
035	XCGL(↑)	DOUBLE DATA EXCHANGE
036	MOVR(↑)	MOVE TO REGISTER
037	MOVQ	MOVE QUICK
038	XFRB(↑)*	MULTIPLE BIT TRANSFER
040	XFER(↑)	BLOCK TRANSFER
041	BSET(↑)	BLOCK SET
042	MOVB(↑)	MOVE BIT
043	MOVD(↑)	MOVE DIGIT
044	DIST(↑)	SINGLE WORD DISTRIBUTE
045	COLL(↑)	DATA COLLECT
046	BXFR(↑)*	INTERBANK BLOCK TRANSFER
047	SETA(↑)*	MULTIPLE BIT SET
048	RSTA(↑)*	MULTIPLE BIT RESET

Note Instructions with one asterisk (*) are supported by the CVM1D and version-2 CVM1 CPU Units only.
Instructions with two asterisks (**) are supported by the CV500, CV1000, and CV2000 only.
Instructions with three asterisks (***) are not supported by the CVM1D.

Data Shift Instructions

Code	Mnemonic	Name
050	SFT	SHIFT REGISTER
051	SFTR(↑)	REVERSIBLE SHIFT REGISTER
052	ASFT(↑)	ASYNCHRONOUS SHIFT REGISTER
053	WSFT(↑)	WORD SHIFT
054	NSFL*	SHIFT N-BIT DATA LEFT
055	NSFR*	SHIFT N-BIT DATA RIGHT
056	NASL*	SHIFT N-BITS LEFT
057	NASR*	SHIFT N-BITS RIGHT
058	NSLL*	DOUBLE SHIFT N-BIT LEFT
059	NSRL*	DOUBLE SHIFT N-BIT RIGHT
060	ASL(↑)	ARITHMETIC SHIFT LEFT
061	ASR(↑)	ARITHMETIC SHIFT RIGHT
062	ROL(↑)	ROTATE LEFT
063	ROR(↑)	ROTATE RIGHT
064	ASLL(↑)	DOUBLE SHIFT LEFT
065	ASRL(↑)	DOUBLE SHIFT RIGHT
066	ROLL(↑)	DOUBLE ROTATE LEFT
067	RORL(↑)	DOUBLE ROTATE RIGHT
068	SLD(↑)	SHIFT DIGIT LEFT
069	SRD(↑)	SHIFT DIGIT RIGHT

BCD Calculation and Carry Instructions

Code	Mnemonic	Name
070	ADD(↑)	BCD ADD
071	SUB(↑)	BCD SUBTRACT
072	MUL(↑)	BCD MULTIPLY
073	DIV(↑)	BCD DIVIDE
074	ADDL(↑)	DOUBLE BCD ADD
075	SUBL(↑)	DOUBLE BCD SUBTRACT
076	MULL(↑)	DOUBLE BCD MULTIPLY
077	DIVL(↑)	DOUBLE BCD DIVIDE
078	STC(↑)	SET CARRY
079	CLC(↑)	CLEAR CARRY

Binary Calculation Instructions

Code	Mnemonic	Name
080	ADB(↑)	BINARY ADD
081	SBB(↑)	BINARY SUBTRACT
082	MLB(↑)	BINARY MULTIPLY
083	DVB(↑)	BINARY DIVIDE
084	ADBL(↑)	DOUBLE BINARY ADD
085	SBBL(↑)	DOUBLE BINARY SUBTRACT
086	MLBL(↑)	DOUBLE BINARY MULTIPLY
087	DVBL(↑)	DOUBLE BINARY DIVIDE

Increment/Decrement Instructions

Code	Mnemonic	Name
090	INC(↑)	INCREMENT BCD
091	DEC(↑)	DECREMENT BCD
092	INCB(↑)	INCREMENT BINARY
093	DECB(↑)	DECREMENT BINARY
094	INCL(↑)	DOUBLE INCREMENT BCD
095	DECL(↑)	DOUBLE DECREMENT BCD
096	INBL(↑)	DOUBLE INCREMENT BINARY
097	DCBL(↑)	DOUBLE DECREMENT BINARY

Data Format Conversion and Special Calculation Instructions

Code	Mnemonic	Name
100	BIN(↑)	BCD-TO-BINARY
101	BCD(↑)	BINARY-TO-BCD
102	BINL(↑)	DOUBLE BCD-TO-DOUBLE BINARY
103	BCDL(↑)	DOUBLE BINARY-TO-DOUBLE BCD
104	NEG(↑)	2'S COMPLEMENT
105	NEGL(↑)	DOUBLE 2'S COMPLEMENT
106	SIGN(↑)	SIGN

Basic I/O Unit Instructions

Code	Mnemonic	Name
110	MLPX(↑)	4-TO-16 DECODER
111	DMPX(↑)	16-TO-4 ENCODER
112	SDEC(↑)	7-SEGMENT DECODER
113	ASC(↑)	ASCII CONVERT
114	BCNT(↑)	BIT COUNTER
115	LINE(↑)	COLUMN-TO-LINE
116	COLM(↑)	LINE-TO-COLUMN
117	HEX(↑)*	ASCII-TO-HEX

Note Instructions with one asterisk (*) are supported by the CVM1D and version-2 CVM1 CPU Units only.

Instructions with two asterisks (**) are supported by the CV500, CV1000, and CV2000 only.
Instructions with three asterisks (***) are not supported by the CVM1D.

Special Timer and SFC Control Instructions

Code	Mnemonic	Name
120	TTIM	ACCUMULATIVE TIMER
121	TIML	DOUBLE TIMER
122	MTIM	MULTI-OUTPUT TIMER
123	TCNT**	TRANSITION COUNTER
124	TSR(↑)**	READ STEP TIMER
125	TSW(↑)**	WRITE STEP TIMER

Logical Instructions

Code	Mnemonic	Name
130	ANDW(↑)	LOGICAL AND
131	ORW(↑)	LOGICAL OR
132	XORW(↑)	EXCLUSIVE OR
133	XNRW(↑)	EXCLUSIVE NOR
134	ANDL(↑)	DOUBLE LOGICAL AND
135	ORWL(↑)	DOUBLE LOGICAL OR
136	XORL(↑)	DOUBLE EXCLUSIVE OR
137	XNRL(↑)	DOUBLE EXCLUSIVE NOR
138	COM(↑)	COMPLEMENT
139	COML(↑)	DOUBLE COMPLEMENT

Special and Time-related Instructions

Code	Mnemonic	Name
140	ROOT(↑)	BCD SQUARE ROOT
141	FDIV(↑)	FLOATING POINT DIVIDE(BCD)
142	APR(↑)	ARITHMETIC PROCESS
143	SEC(↑)	HOURS-TO-SECONDS
144	HMS(↑)	SECONDS-TO-HOURS
145	CADD(↑)	CALENDAR ADD
146	CSUB(↑)	CALENDAR SUBTRACT

Subroutine and Interrupt Instructions

Code	Mnemonic	Name
150	SBN	SUBROUTINE ENTER
151	SBS(↑)	SUBROUTINE CALL
152	RET	SUBROUTINE RETURN
153	MSKS(↑)** *	INTERRUPT MASK
154	CLI(↑)***	CLEAR INTERRUPT
155	MSKR(↑)** *	READ MASK
156	MCRO(↑)*	MACRO

Table Data Processing Instructions

Code	Mnemonic	Name
160	SSET(↑)	SET STACK
161	PUSH(↑)	PUSH ONTO STACK
162	LIFO(↑)	LAST IN FIRST OUT
163	FIFO(↑)	FIRST IN FIRST OUT
164	SRCH(↑)	DATA SEARCH
165	MAX(↑)	FIND MAXIMUM
166	MIN(↑)	FIND MINIMUM
167	SUM(↑)	SUM

Debugging, Special, Error Processing and Time-related Instructions

Code	Mnemonic	Name
170	TRSM	TRACE MEMORY
171	EMBC(↑)	SELECT EM BANK
172	CCL(↑)	LOAD FLAGS
173	CCS(↑)	SAVE FLAGS
174	MARK	MARK TRACE
175	REGL(↑)	LOAD REGISTER
176	REGS(↑)	SAVE REGISTER
177	FPD*	FAILURE POINT DETECTION
178	WDT(↑)*	MAXIMUM CYCLE TIME EXTEND
179	DATE(↑)*	CLOCK COMPENSATION

File Memory, Basic I/O, and Special I/O Instructions

Code	Mnemonic	Name
180	FILR(↑)***	READ DATA FILE
181	FILW(↑)***	WRITE DATA FILE
182	FILP(↑)***	READ PROGRAM FILE
183	FLSP(↑)***	CHANGE STEP PROGRAM
184	IORF(↑)	I/O REFRESH
187	IOSP(↑)	DISABLE ACCESS
188	IORS	ENABLE ACCESS
189	IODP(↑)	I/O DISPLAY
190	READ	I/O READ
191	WRIT	I/O WRITE
192	SEND(↑)	NETWORK SEND
193	RECV(↑)	NETWORK RECEIVE
194	CMND(↑)	DELIVER COMMAND
195	MSG(↑)	MESSAGE

Note Instructions with one asterisk (*) are supported by the CVM1D and version-2 CVM1 CPU Units only.
Instructions with two asterisks (**) are supported by the CV500, CV1000, and CV2000 only.
Instructions with three asterisks (***) are not supported by the CVM1D.

149

SFC Control Instructions

Code	Mnemonic	Name
202	TOUT**	TRANSITION OUTPUT
210	SA(↑)**	ACTIVATE STEP
211	SP(↑)**	PAUSE STEP
212	SR(↑)**	RESTART STEP
213	SF(↑)**	END STEP
214	SE(↑)**	DEACTIVATE STEP
215	SOFF(↑)**	RESET STEP

Sequence Control and Timer/Counter Reset Instructions

Code	Mnemonic	Name
221	CJP*	CONDITIONAL JUMP
222	CJPN*	CONDITIONAL JUMP
236	CNR(↑)	RESET TIMER/COUNTER

Block Program Instruction( )

Code	Mnemonic	Name
250	BPRG*	BLOCK PROGRAM

Block Program Instructions < >

Code	Mnemonic	Name
<001>	BEND*	BLOCK PROGRAM END
<002>	IF(NOT)*	CONDITIONAL BRANCH
<003>	ELSE*	NO CONDITIONAL BRANCH
<004>	IEND*	END OF BRANCH
<005>	WAIT(NOT)*	1-SCAN WAIT
<006>	EXIT(NOT)*	CONDITIONAL END
<009>	LOOP*	REPEAT BLOCK
<010>	LEND(NOT)*	REPEAT BLOCK END
<011>	BPPS*	BLOCK PROGRAM PAUSE
<012>	BPRS*	BLOCK PROGRAM RESTART
<013>	TIMW*	TIMER WAIT
<014>	CNTW*	COUNTER WAIT
<015>	TMHW*	HIGH-SPEED TIMER WAIT

Data Shift Instructions

Code	Mnemonic	Name
260	RLNC(↑)*	ROTATE LEFT WITHOUT CARRY
261	RRNC(↑)*	ROTATE RIGHT WITHOUT CARRY
262	RLNL(↑)*	DOUBLE ROTATE LEFT WITHOUT CARRY
263	RRNL(↑)*	ROTATE LEFT WITHOUT CARRY

Data Control, Special Calculation, and Data Conversion Instructions

Code	Mnemonic	Name
270	PID*	PID CONTROL
271	LMT(↑)*	LIMIT CONTROL
272	BAND(↑)*	DEAD BAND CONTROL
273	ZONE(↑)*	DEAD-ZONE CONTROL
274	ROTB(↑)*	BINARY ROOT
275	BINS(↑)*	SIGNED BCD-TO-BINARY
276	BCDS(↑)*	SIGNED BINARY-TO-BCD
277	BISL(↑)*	DOUBLE SIGNED BCD-TO-BINARY
278	BDSL(↑)*	DOUBLE SIGNED BINARY-TO-BCD

Special I/O Instructions

Code	Mnemonic	Name
280	RD2*	I/O READ 2
281	WR2*	I/O UNIT WRITE 2

Data Comparison Instructions

Code	Mnemonic	Name
300	=*	EQUAL
301	=L*	DOUBLE EQUAL
302	=S*	SIGNED EQUAL
303	=SL*	DOUBLE SIGNED EQUAL
305	<>*	NOT EQUAL
306	<>L*	DOUBLE NOT EQUAL
307	<>S*	SIGNED NOT EQUAL
308	<>SL*	DOUBLE SIGNED NOT EQUAL
310	<*	LESS THAN
311	<L*	DOUBLE LESS THAN
312	<S*	SIGNED LESS THAN
313	<SL*	DOUBLE SIGNED LESS THAN
315	<=*	LESS THAN OR EQUAL
316	<=L*	DOUBLE LESS THAN OR EQUAL
317	<=S*	SIGNED LESS THAN OR EQUAL
318	<=SL*	DOUBLE SIGNED LESS THAN OR EQUAL
320	>*	GREATER THAN
321	>L*	DOUBLE GREATER THAN
322	>S*	SIGNED GREATER THAN
323	>SL*	DOUBLE SIGNED GREATER THAN
325	>=*	GREATER THAN OR EQUAL
326	>=L*	DOUBLE GREATER THAN OR EQUAL
327	>=S*	SIGNED GREATER THAN OR EQUAL

Note Instructions with one asterisk (*) are supported by the CVM1D and version-2 CVM1 CPU Units only.

150 Instructions with two asterisks (**) are supported by the CV500, CV1000, and CV2000 only.
Instructions with three asterisks (***) are not supported by the CVM1D.

328	>=SL*	DOUBLE SIGNED GREATER THAN OR EQUAL

Bit Tests

Code	Mnemonic	Name
350	TST*	BIT TEST
351	TSTN*	BIT TEST

Symbol Math Instructions

Code	Mnemonic	Name
400	+(↑)*	SIGNED BINARY ADD WITHOUT CARRY
401	+L(↑)*	DOUBLE SIGNED BINARY ADD WITHOUT CARRY
402	+C(↑)*	SIGNED BINARY ADD WITH CARRY
403	+CL(↑)*	DOUBLE SIGNED BINARY ADD WITH CARRY
404	+B(↑)*	BCD ADD WITHOUT CARRY
405	+BL(↑)*	DOUBLE BCD ADD WITHOUT CARRY
406	+BC(↑)*	BCD ADD WITH CARRY
407	+BCL(↑)*	DOUBLE BCD ADD WITH CARRY
410	–(↑)*	SIGNED BINARY SUBTRACT WITHOUT CARRY
411	-L(↑)*	DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY
412	–C(↑)*	SIGNED BINARY SUBTRACT WITH CARRY
413	–CL(↑)*	DOUBLE SIGNED BINARY SUBTRACT WITH CARRY
414	–B(↑)*	BCD SUBTRACT WITHOUT CARRY
415	–BL(↑)*	DOUBLE BCD SUBTRACT WITHOUT CARRY
416	–BC(↑)*	BCD SUBTRACT WITH CARRY
417	–BCL(↑)*	DOUBLE BCD SUBTRACT WITH CARRY
420	∗(↑)*	SIGNED BINARY MULTIPLY
421	∗L(↑)*	DOUBLE SIGNED BINARY MULTIPLY
422	∗U(↑)*	UNSIGNED BINARY MULTIPLY
423	∗UL(↑)*	DOUBLE UNSIGNED BINARY MULTIPLY
424	∗B(↑)*	BCD MULTIPLY
425	∗BL(↑)*	DOUBLE BCD MULTIPLY
430	/(↑)*	SIGNED BINARY DIVIDE
431	/L(↑)*	DOUBLE SIGNED BINARY DIVIDE
432	/U(↑)*	UNSIGNED BINARY DIVIDE
433	/UL(↑)*	DOUBLE UNSIGNED BINARY DIVIDE
434	/B(↑)*	BCD DIVIDE

Note Instructions with one asterisk (*) are supported by the CVM1D and version-2 CVM1 CPU Units only.
Instructions with two asterisks (**) are supported by the CV500, CV1000, and CV2000 only.
Instructions with three asterisks (***) are not supported by the CVM1D.

151

435	/BL(↑)*	DOUBLE BCD DIVIDE

Floating-point Math Instructions

Code	Mnemonic	Name
450	FIX(↑)*	FLOATING-TO-16-BIT
451	FIXL(↑)*	FLOATING-TO-32-BIT
452	FLT(↑)*	16-BIT-TO-FLOATING
453	FLTL(↑)*	32-BIT-TO-FLOATING
454	+F(↑)*	FLOATING-POINT ADD
455	−F(↑)*	FLOATING-POINT SUBTRACT
456	∗F(↑)*	FLOATING-POINT MULTIPLY
457	/F(↑)*	FLOATING-POINT DIVIDE
458	RAD(↑)*	DEGREES-TO-RADIANS
459	DEG(↑)*	RADIANS-TO-DEGREES
460	SIN(↑)*	SINE
461	COS(↑)*	COSINE
462	TAN(↑)*	TANGENT
463	ASIN(↑)*	SINE-TO-ANGLE
464	ACOS(↑)*	COSINE-TO-ANGLE
465	ATAN(↑)*	TANGENT-TO-ANGLE
466	SQRT(↑)*	SQUARE ROOT
467	EXP(↑)*	EXPONENT
468	LOG(↑)*	LOGARITHM

Note Instructions with one asterisk (*) are supported by the CVM1D and version-2 CVM1 CPU Units only.

152 Instructions with two asterisks (**) are supported by the CV500, CV1000, and CV2000 only.
Instructions with three asterisks (***) are not supported by the CVM1D.

Appendix F
Alphabetic List of Instructions by Mnemonics

Mnemonic	Code	Name
ACOS(↑)*	464	COSINE-TO-ANGLE
ADB(↑)	080	BINARY ADD
ADBL(↑)	084	DOUBLE BINARY ADD
ADD(↑)	070	BCD ADD
ADDL(↑)	074	DOUBLE BCD ADD
ANDL(↑)	134	DOUBLE LOGICAL AND
ANDW(↑)	130	LOGICAL AND
APR(↑)	142	ARITHMETIC PROCESS
ASC(↑)	113	ASCII CONVERT
ASFT(↑)	052	ASYNCHRONOUS SHIFT REGISTER
ASIN(↑)*	463	SINE-TO-ANGLE
ASL(↑)	060	ARITHMETIC SHIFT LEFT
ASLL(↑)	064	DOUBLE SHIFT LEFT
ASR(↑)	061	ARITHMETIC SHIFT RIGHT
ASRL(↑)	065	DOUBLE SHIFT RIGHT
ATAN(↑)*	465	TANGENT-TO-ANGLE
BAND(↑)*	272	DEAD BAND CONTROL
BCD(↑)	101	BINARY-TO-BCD
BCDL(↑)	103	DOUBLE BINARY-TO-DOUBLE BCD
BCDS(↑)*	276	SIGNED BINARY-TO-BCD
BCMP(↑)	022	BLOCK COMPARE
BCNT(↑)	114	BIT COUNTER
BDSL(↑)*	278	DOUBLE SIGNED BINARY-TO-BCD
BEND*	<001>	BLOCK PROGRAM END
BIN(↑)	100	BCD-TO-BINARY
BINL(↑)	102	DOUBLE BCD-TO-DOUBLE BINARY
BINS(↑)*	275	SIGNED BCD-TO-BINARY
BISL(↑)*	277	DOUBLE SIGNED BCD-TO-BINARY
BPPS*	<011>	BLOCK PROGRAM PAUSE
BPRG*	250	BLOCK PROGRAM
BPRS*	<012>	BLOCK PROGRAM RESTART
BSET(↑)	041	BLOCK SET

Mnemonic	Code	Name
BXFR(↑)*	046	INTERBANK BLOCK TRANSFER
CADD(↑)	145	CALENDAR ADD
CCL(↑)	172	LOAD FLAGS
CCS(↑)	173	SAVE FLAGS
CJP*	221	CONDITIONAL JUMP
CJPN*	222	CONDITIONAL JUMP
CLC(↑)	079	CLEAR CARRY
CLI(↑)***	154	CLEAR INTERRUPT
CMND(↑)	194	DELIVER COMMAND
CMP(!)	020	COMPARE
CMP(!)*	028	UNSIGNED COMPARE
CMPL	021	DOUBLE COMPARE
CMPL*	029	DOUBLE UNSIGNED COMPARE
CNR(↑)	236	RESET TIMER/COUNTER
CNTR	012	REVERSIBLE COUNTER
CNTW*	<014>	COUNTER WAIT
COLL(↑)	045	DATA COLLECT
COLM(↑)	116	LINE-TO-COLUMN
COM(↑)	138	COMPLEMENT
COML(↑)	139	DOUBLE COMPLEMENT
COS*	461	COSINE
CPS(!)*	026	SIGNED BINARY COMPARE
CPSL*	027	DOUBLE SIGNED BINARY COMPARE
CSUB(↑)	146	CALENDAR SUBTRACT
DATE(↑)*	179	CLOCK COMPENSATION
DCBL(↑)	097	DOUBLE DECREMENT BINARY
DEC(↑)	091	DECREMENT BCD
DECB(↑)	093	DECREMENT BINARY
DECL(↑)	095	DOUBLE DECREMENT BCD
DEG(↑)*	459	RADIANS-TO-DEGREES
DIFD(!)	014	DIFFERENTIATE DOWN
DIFU(!)	013	DIFFERENTIATE UP
DIST(↑)	044	SINGLE WORD DISTRIBUTE
DIV(↑)	073	BCD DIVIDE
DIVL(↑)	077	DOUBLE BCD DIVIDE

Note Instructions with one asterisk (*) are supported by the CVM1D and version-2 CVM1 CPU Units only.
Instructions with two asterisks (**) are supported by the CV500, CV1000, and CV2000 only. **153**
Instructions with three asterisks (***) are not supported by the CVM1D.

Mnemonic	Code	Name
DMPX(↑)	111	16-TO-4/256-8 ENCODER
DOWN*	019	CONDITION OFF
DVB(↑)	083	BINARY DIVIDE
DVBL(↑)	087	DOUBLE BINARY DIVIDE
ELSE*	<003>	NO CONDITIONAL BRANCH
EMBC(↑)	171	SELECT EM BANK
END	001	END
EQU(↑)	025	EQUAL
EXIT(NOT)*	<006>	CONDITIONAL END
EXP(↑)*	467	EXPONENT
FAL(↑)	006	FAILURE ALARM
FALS(↑)	007	FAILURE ALARM
FDIV(↑)	141	FLOATING POINT DIVIDE(BCD)
FIFO(↑)	163	FIRST IN FIRST OUT
FILP(↑)***	182	READ PROGRAM FILE
FILR(↑)***	180	READ DATA FILE
FILW(↑)***	181	WRITE DATA FILE
FIX(↑)*	450	FLOATING-TO-16-BIT
FIXL(↑)*	451	FLOATING-TO-32-BIT
FLSP(↑)***	183	CHANGE STEP PROGRAM
FLT(↑)*	452	16-BIT-TO-FLOATING
FLTL(↑)*	453	32-BIT-TO-FLOATING
FPD*	177	FAILURE POINT DETECTION
HEX(↑)*	117	ASCII-TO-HEX
HMS(↑)	144	SECONDS-TO-HOURS
IEND*	<004>	END OF BRANCH
IF(NOT)*	<002>	CONDITIONAL BRANCH
IL	002	INTERLOCK
ILC	003	INTERLOCK CLEAR
INBL(↑)	096	DOUBLE INCREMENT BINARY
INC(↑)	090	INCREMENT BCD
INCB(↑)	092	INCREMENT BINARY
INCL(↑)	094	DOUBLE INCREMENT BCD
IODP(↑)	189	I/O DISPLAY
IORF(↑)	184	I/O REFRESH
IORS	188	ENABLE ACCESS
IOSP(↑)	187	DISABLE ACCESS
JME	005	JUMP END
JMP	004	JUMP
KEEP(!)	011	KEEP

Mnemonic	Code	Name
LEND(NOT)*	<010>	REPEAT BLOCK END
LIFO(↑)	162	LAST IN FIRST OUT
LINE(↑)	115	COLUMN-TO-LINE
LMT(↑)*	271	LIMIT CONTROL
LOG(↑)*	468	LOGARITHM
LOOP*	<009>	REPEAT BLOCK
MARK	174	MARK TRACE
MAX(↑)	165	FIND MAXIMUM
MCMP(↑)	024	MULTIPLE COMPARE
MCRO(↑)*	156	MACRO
MIN(↑)	166	FIND MINIMUM
MLB(↑)	082	BINARY MULTIPLY
MLBL(↑)	086	DOUBLE BINARY MULTIPLY
MLPX(↑)	110	4-TO-16/8-TO-256 DECODER
MOV(!↑)	030	MOVE
MOVB(↑)	042	MOVE BIT
MOVD(↑)	043	MOVE DIGIT
MOVL(↑)	032	DOUBLE MOVE
MOVQ	037	MOVE QUICK
MOVR(↑)	036	MOVE TO REGISTER
MSG(↑)	195	MESSAGE
MSKR(↑)***	155	READ MASK
MSKS(↑)***	153	INTERRUPT MASK
MTIM	122	MULTI-OUTPUT TIMER
MUL(↑)	072	BCD MULTIPLY
MULL(↑)	076	DOUBLE BCD MULTIPLY
MVN(↑)	031	MOVE NOT
MVNL(↑)	033	DOUBLE MOVE NOT
NASL(↑)*	056	SHIFT N-BITS LEFT
NASR(↑)*	057	SHIFT N-BITS RIGHT
NEG(↑)	104	2'S COMPLEMENT
NEGL(↑)	105	DOUBLE 2'S COMPLEMENT
NOP	000	NO OPERATION
NOT	010	NOT
NSFL(↑)*	054	SHIFT N-BIT DATA LEFT
NSFR(↑)*	055	SHIFT N-BIT DATA RIGHT
NSLL(↑)*	058	DOUBLE SHIFT N-BIT LEFT
NSRL(↑)*	059	DOUBLE SHIFT N-BIT RIGHT
ORW(↑)	131	LOGICAL OR
ORWL(↑)	135	DOUBLE LOGICAL OR
PID*	270	PID CONTROL
PUSH(↑)	161	PUSH ONTO STACK

Note Instructions with one asterisk (*) are supported by the CVM1D and version-2 CVM1 CPU Units only.

154 Instructions with two asterisks (**) are supported by the CV500, CV1000, and CV2000 only.
Instructions with three asterisks (***) are not supported by the CVM1D.

Mnemonic	Code	Name
RAD(↑)*	458	DEGREES-TO-RADIANS
RD2*	280	I/O READ 2
READ	190	I/O READ
RECV(↑)	193	NETWORK RECEIVE
REGL(↑)	175	LOAD REGISTER
REGS(↑)	176	SAVE REGISTER
RET	152	SUBROUTINE RETURN
RLNC(↑)*	260	ROTATE LEFT WITHOUT CARRY
RLNL(↑)*	262	DOUBLE ROTATE LEFT WITHOUT CARRY
ROL(↑)	062	ROTATE LEFT WITH CARRY
ROLL(↑)	066	DOUBLE ROTATE LEFT WITH CARRY
ROOT(↑)	140	BCD SQUARE ROOT
ROR(↑)	063	ROTATE RIGHT WITH CARRY
RORL(↑)	067	DOUBLE ROTATE RIGHT WITH CARRY
ROTB(↑)*	274	BINARY ROOT
RRNC(↑)*	261	ROTATE RIGHT WITHOUT CARRY
RRNL(↑)*	263	ROTATE LEFT WITHOUT CARRY
RSET(!↑↓)	017	RSET
RSTA(↑)*	048	MULTIPLE BIT RESET
SA(↑)**	210	ACTIVATE STEP
SBB(↑)	081	BINARY SUBTRACT
SBBL(↑)	085	DOUBLE BINARY SUBTRACT
SBN	150	SUBROUTINE ENTER
SBS(↑)	151	SUBROUTINE CALL
SDEC(↑)	112	7-SEGMENT DECODER
SE(↑)**	214	DEACTIVATE STEP
SEC(↑)	143	HOURS-TO-SECONDS
SEND(↑)	192	NETWORK SEND
SET(!↑↓)	016	SET
SETA(↑)*	047	MULTIPLE BIT SET
SF(↑)**	213	END STEP
SFT	050	SHIFT REGISTER
SFTR(↑)	051	REVERSIBLE SHIFT REGISTER
SIGN(↑)	106	SIGN
SIN(↑)*	460	SINE
SLD(↑)	068	SHIFT DIGIT LEFT
SNXT	009	STEP START
SOFF(↑)**	215	RESET STEP
SP(↑)**	211	PAUSE STEP

Mnemonic	Code	Name
SQRT(↑)*	466	SQUARE ROOT
SR(↑)**	212	RESTART STEP
SRCH(↑)	164	DATA SEARCH
SRD(↑)	069	SHIFT DIGIT RIGHT
SSET(↑)	160	SET STACK
STC(↑)	078	SET CARRY
STEP	008	STEP DEFINE
SUB(↑)	071	BCD SUBTRACT
SUBL(↑)	075	DOUBLE BCD SUBTRACT
SUM(↑)	167	SUM
TAN(↑)*	462	TANGENT
TCMP(↑)	023	TABLE COMPARE
TCNT**	123	TRANSITION COUNTER
TIMH	015	HIGH-SPEED TIMER
TIML	121	DOUBLE TIMER
TIMW*	<013>	TIMER WAIT
TMHW*	<015>	HIGH-SPEED TIMER WAIT
TOUT**	202	TRANSITION OUTPUT
TRSM	170	TRACE MEMORY
TSR(↑)**	124	READ STEP TIMER
TST*	350	BIT TEST
TSTN*	351	BIT TEST
TSW(↑)**	125	WRITE STEP TIMER
TTIM	120	ACCUMULATIVE TIMER
UP*	018	CONDITION ON
WAIT(NOT)*	<005>	1-SCAN WAIT
WDT*	178	MAXIMUM CYCLE TIME EXTEND
WR2*	281	I/O UNIT WRITE 2
WRIT	191	I/O WRITE
WSFT(↑)	053	WORD SHIFT
XCGL(↑)	035	DOUBLE DATA EXCHANGE
XCHG(↑)	034	DATA EXCHANGE
XFER(↑)	040	BLOCK TRANSFER
XFRB(↑)*	038	MULTIPLE BIT TRANSFER
XNRL(↑)	137	DOUBLE EXCLUSIVE NOR
XNRW(↑)	133	EXCLUSIVE NOR
XORL(↑)	136	DOUBLE EXCLUSIVE OR
XORW(↑)	132	EXCLUSIVE OR
ZONE(↑)*	273	DEAD-ZONE CONTROL
/(↑)*	430	SIGNED BINARY DIVIDE
/B(↑)*	434	BCD DIVIDE
/BL(↑)*	435	DOUBLE BCD DIVIDE

Note Instructions with one asterisk (*) are supported by the CVM1D and version-2 CVM1 CPU Units only.
Instructions with two asterisks (**) are supported by the CV500, CV1000, and CV2000 only.
Instructions with three asterisks (***) are not supported by the CVM1D.

155

Mnemonic	Code	Name
/F(↑)*	457	FLOATING-POINT DIVIDE
/L(↑)*	431	DOUBLE SIGNED BINARY DIVIDE
/U(↑)*	432	UNSIGNED BINARY DIVIDE
/UL(↑)*	433	DOUBLE UNSIGNED BINARY DIVIDE
+(↑)*	400	SIGNED BINARY ADD WITHOUT CARRY
+B(↑)*	404	BCD ADD WITHOUT CARRY
+BC(↑)*	406	BCD ADD WITH CARRY
+BCL(↑)*	407	DOUBLE BCD ADD WITH CARRY
+BL(↑)*	405	DOUBLE BCD ADD WITHOUT CARRY
+C(↑)*	402	SIGNED BINARY ADD WITH CARRY
+CL(↑)*	403	DOUBLE SIGNED BINARY ADD WITH CARRY
+F(↑)*	454	FLOATING-POINT ADD
+L(↑)*	401	DOUBLE SIGNED BINARY ADD WITHOUT CARRY
−(↑)*	410	SIGNED BINARY SUBTRACT WITHOUT CARRY
−B(↑)*	414	BCD SUBTRACT WITHOUT CARRY
−BC(↑)*	416	BCD SUBTRACT WITH CARRY
−BCL(↑)*	417	DOUBLE BCD SUBTRACT WITH CARRY
−BL(↑)*	415	DOUBLE BCD SUBTRACT WITHOUT CARRY
−C(↑)*	412	SIGNED BINARY SUBTRACT WITH CARRY
−CL(↑)*	413	DOUBLE SIGNED BINARY SUBTRACT WITH CARRY
−F(↑)*	455	FLOATING-POINT SUBTRACT
-L(↑)*	411	DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY
=*	300	EQUAL
=L*	301	DOUBLE EQUAL

Mnemonic	Code	Name
=S*	302	SIGNED EQUAL
=SL*	303	DOUBLE SIGNED EQUAL
<*	310	LESS THAN
<=*	315	LESS THAN OR EQUAL
<=L*	316	DOUBLE LESS THAN OR EQUAL
<=S*	317	SIGNED LESS THAN OR EQUAL
<=SL*	318	DOUBLE SIGNED LESS THAN OR EQUAL
<>*	305	NOT EQUAL
<>L*	306	DOUBLE NOT EQUAL
<>S*	307	SIGNED NOT EQUAL
<>SL*	308	DOUBLE SIGNED NOT EQUAL
<L*	311	DOUBLE LESS THAN
<S*	312	SIGNED LESS THAN
<SL*	313	DOUBLE SIGNED LESS THAN
>*	320	GREATER THAN
>=*	325	GREATER THAN OR EQUAL
>=L*	326	DOUBLE GREATER THAN OR EQUAL
>=S*	327	SIGNED GREATER THAN OR EQUAL
>=SL*	328	DOUBLE SIGNED GREATER THAN OR EQUAL
>L*	321	DOUBLE GREATER THAN
>S*	322	SIGNED GREATER THAN
>SL*	323	DOUBLE SIGNED GREATER THAN
∗(↑)*	420	SIGNED BINARY MULTIPLY
∗B(↑)*	424	BCD MULTIPLY
∗BL(↑)*	425	DOUBLE BCD MULTIPLY
∗F(↑)*	456	FLOATING-POINT MULTIPLY
∗L(↑)*	421	DOUBLE SIGNED BINARY MULTIPLY
∗U(↑)*	422	UNSIGNED BINARY MULTIPLY
∗UL(↑)*	423	DOUBLE UNSIGNED BINARY MULTIPLY

Note Instructions with one asterisk (*) are supported by the CVM1D and version-2 CVM1 CPU Units only.

156 Instructions with two asterisks (**) are supported by the CV500, CV1000, and CV2000 only.
Instructions with three asterisks (***) are not supported by the CVM1D.

Index

A–B

action blocks, changing,

alphanumeric input mode,

ASCII display, display example,

baud rate, ,

buzzer, volume adjustment, ,

C

cable connector, cover,

clock, reading/changing,

communications
    baud rate setting, ,
    errors,
    host link parameters,

comparing
    I/O table and Units,
    memory cards and PC memory,

Connecting Cables,
    connectors,
    disconnecting,
    models,

connections,
    cable connectors,
    cables,
    connecting to PC,
    connecting to Programming Console,
    priority,
    restrictions in number,

contrast, adjustment, ,

counters. *See* timers/counters

customizing, unsupported operations,

cycle time
    displaying,
    minimum cycle time setting,
    monitoring time,

D

data areas, display symbols and abbreviations,

debugging,

decimal display, display example,

differentiate up/down instructions, display example,

dimensions
    Mounting Bracket,
    Programming Console,

DIP switch
    baud rate setting,

PC baud rate setting, ,

displays,
    ASCII,
    automatic error displays,
    contrast adjustment, ,
    cycle time,
    decimal,
    examples,
    FAL, ,
    FALS, ,
    hexadecimal,
    I/O tables,
    initial display,
    MSG(195),
    program designation display,
    programming display,
    symbols and abbreviations,

Duplex Unit,
    connection,

E

error history, reading,

errors
    automatic error displays,
    automatic reading,
    Cannot connect,
    clearing, ,
    communications,
    displaying and clearing,
    error display priority,
    error history,
    error messages,
        operating errors,
    FAL displays, ,
    FALS displays, ,
    operating errors,
    PC, ,
        fatal errors,
        initialization errors,
        non-fatal errors,
    reading,
    reading error history,
    reading messages,

execution controls,

F

FAL
    error displays, ,
    errors,

FALS
    error displays, ,
    errors,

features,

files, deleting from memory cards,

Revision History

A manual revision code appears as a suffix to the catalog number on the front cover of the manual.

Cat. No. W222-E1-3

Revision code

The following table outlines the changes made to the manual during each revision. Page numbers refer to the previous version.

Revision code	Date	Revised content
1	February 1993	Original production
2	February 1995	Minor changes to add CV2000, CVM1 (V2), and SSS.
		Page 11: Memory Cassette added.
		Page 113: "Jump error" description changed.
		Page 117: Memory Casette added.
		Page 133: Added line to end of table.
		Page 137 on: Added new table of instructions including V2 CVM1 PCs.
3	January 1999	Minor changes to add CVM1D and I/O Unit online replacement operation (hot swapping) for CVM1D. A precautions section was also added to the beginning of the manual. Precautions were also added at various points in the manual and corrections were made to update information on Programming Devices.