

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

**Artisan Technology Group**
(217) 352-9330 | sales@artisantg.com | artisantg.com

**Full-service, independent repair center**
with experienced engineers and technicians on staff.

**We buy your excess, underutilized, and idle equipment**
along with credit for buybacks and trade-ins.

**Custom engineering**
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

**Expert team | Trust guarantee | 100% satisfaction**

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

**Visit our website - *Click HERE***

Cat. No. W341-E1-05

# SYSMAC CS/CJ Series

## CQM1H-PRO01-E/CQM1-PRO01-E/C200H-PRO27-E

# Programming Consoles

# OPERATION MANUAL

OMRON

# SYSMAC CS/CJ Series
# CQM1H-PRO01-E/CQM1-PRO01-E/
# C200H-PRO27-E
# Programming Consoles

## Operation Manual

*Revised October 2005*

# 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. Additionally, there may be severe property damage.

**⚠WARNING** Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury. Additionally, there may be severe property damage.

**⚠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 “PLC” means Programmable Controller. “PC” is used, however, in some Programming Device displays to mean Programmable Controller.

# 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.

**© OMRON, 1999**

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, mechanical, electronic, photocopying, recording, or otherwise, without the prior written permission of OMRON.

No patent liability is assumed with respect to the use of the information contained herein. Moreover, because OMRON is constantly striving to improve its high-quality products, the information contained in this manual is subject to change without notice. Every precaution has been taken in the preparation of this manual. Nevertheless, OMRON assumes no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained in this publication.

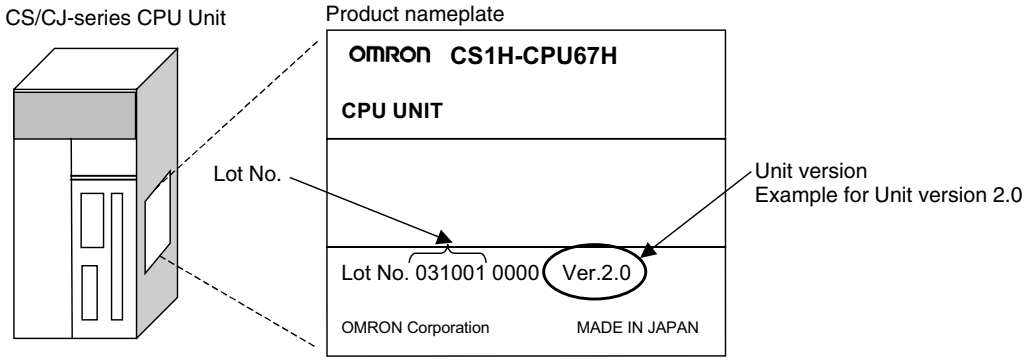
# *Unit Versions of CS/CJ-series CPU Units*

## Unit Versions

A “unit version” has been introduced to manage CPU Units in the CS/CJ Series according to differences in functionality accompanying Unit upgrades. This applies to the CS1-H, CJ1-H, CJ1M, and CS1D CPU Units.

### Notation of Unit Versions on Products

The unit version is given to the right of the lot number on the nameplate of the products for which unit versions are being managed, as shown below.




- CS1-H, CJ1-H, and CJ1M CPU Units (except for low-end models) manufactured on or before November 4, 2003 do not have a unit version given on the CPU Unit (i.e., the location for the unit version shown above is blank).
- The unit version of the CS1-H, CJ1-H, and CJ1M CPU Units, as well as the CS1D CPU Units for Single-CPU Systems, begins at version 2.0.
- The unit version of the CS1D CPU Units for Duplex-CPU Systems, begins at version 1.1.
- CPU Units for which a unit version is not given are called *Pre-Ver. □.□ CPU Units, such as Pre-Ver. 2.0 CPU Units* and *Pre-Ver. 1.1 CPU Units.*

### Confirming Unit Versions with Support Software

CX-Programmer version 4.0 can be used to confirm the unit version using one of the following two methods.

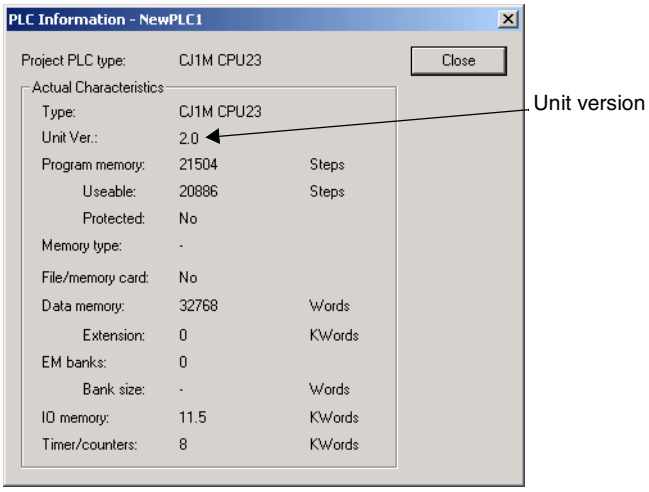
- Using the ***PLC Information***
- Using the ***Unit Manufacturing Information*** (This method can be used for Special I/O Units and CPU Bus Units as well.)

**Note** CX-Programmer version 3.3 or lower cannot be used to confirm unit versions.

**PLC Information**

- If you know the device type and CPU type, select them in the *Change PLC* Dialog Box, go online, and select ***PLC - Edit - Information*** from the menus.
- If you don’t know the device type and CPU type, but are connected directly to the CPU Unit on a serial line, select ***PLC - Auto Online*** to go online, and then select ***PLC - Edit - Information*** from the menus.

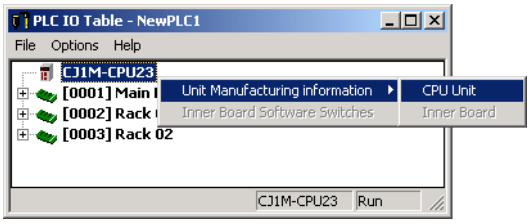
In either case, the following *PLC Information* Dialog Box will be displayed.



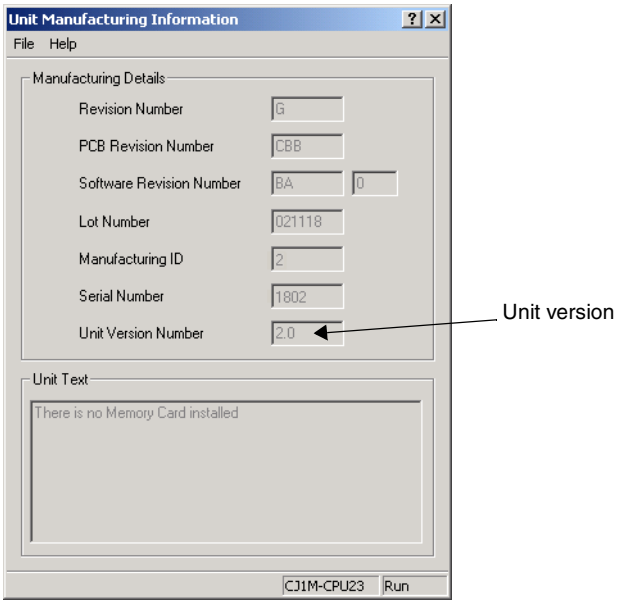

Use the above display to confirm the unit version of the CPU Unit.

## Unit Manufacturing Information

In the *IO Table* Window, right-click and select ***Unit Manufacturing information - CPU Unit.***



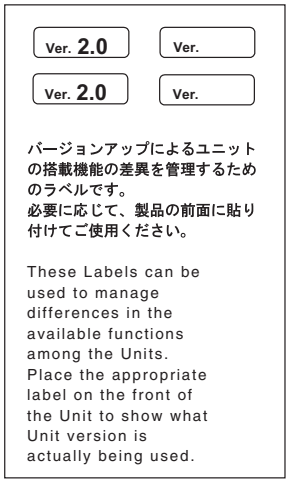

The following *Unit Manufacturing information* Dialog Box will be displayed




Use the above display to confirm the unit version of the CPU Unit connected online.

### Using the Unit Version Labels

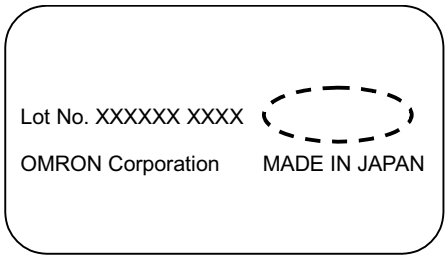
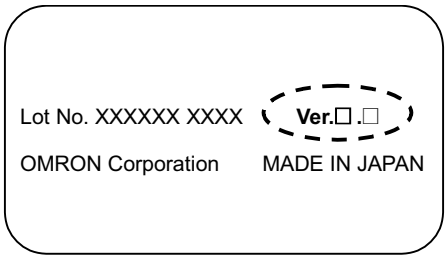
The following unit version labels are provided with the CPU Unit.




These labels can be attached to the front of previous CPU Units to differentiate between CPU Units of different unit versions.

## Unit Version Notation

In this manual, the unit version of a CPU Unit is given as shown in the following table.

| Product nameplate<br><br>Meaning | CPU Units on which no unit version is given<br> | Units on which a version is given (Ver. □.□)<br> |
|---|---|---|
| Designating individual CPU Units (e.g., the CS1H-CPU67H) | Pre-Ver. 2.0 CS1-H CPU Units | CS1H-CPU67H CPU Unit Ver. □.□ |
| Designating groups of CPU Units (e.g., the CS1-H CPU Units) | Pre-Ver. 2.0 CS1-H CPU Units | CS1-H CPU Units Ver. □.□ |
| Designating an entire series of CPU Units (e.g., the CS-series CPU Units) | Pre-Ver. 2.0 CS-series CPU Units | CS-series CPU Units Ver. □.□ |

## Unit Versions and Lot Numbers

| Series | Model | | | Data of manufacture | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Earlier | Sept. 2003 | Oct. 2003 | Nov. 2003 | Dec. 2003 | Later |
| CS Series | CS1 CPU Units | | CS1□-CPU□□ | No unit version | | | | | |
| | CS1-V1 CPU Units | | CS1□-CPU□□-V1 | No unit version | | | | | |
| | CS1-H CPU Units | | CS1□-CPU□□H | Pre-Ver. 2.0 CPU Units | | | CPU Units Ver. 2.0 (Lot No.: 031105 on) | | |
| | CS1D CPU Units | CPU Units for Duplex-CPU System | CS1D-CPU□□H | Pre-Ver. 1.1 CPU Units | | | | CPU Units Ver.1.1 (Lot No.: 031120 on) | |
| | | CPU Units for Single-CPU System | CS1D-CPU□□S | | | | | | CPU Units Ver. 2.0 (Lot No.: 031215 on) |
| CJ Series | CJ1 CPU Units | | CJ1G-CPU□□ | Pre-Ver. 2.0 CPU Units | | | | | |
| | CJ1-H CPU Units | | CJ1□-CPU□□H | Pre-Ver. 2.0 CPU Units | | | CPU Units Ver. 2.0 (Lot No.: 031105 on) | | |
| | CJ1M CPU Units except low-end models | | CJ1M-CPU□□ | Pre-Ver. 2.0 CPU Units | | | CPU Units Ver. 2.0 (Lot No.: 031105 on) | | |
| | CJ1M CPU Units, low-end models | | CJ1M-CPU11/21 | | | Unit Ver. 2.0 (Lot No.: 031002 on) | | | |
| Support Software | CX-Programmer | | WS02-CXPC1-EV□ | Ver.3.2 | | Ver.3.3 | | Ver.4.0 | |

## Function Support by Unit Version

### CS1-H CPU Units (CS1□-CPU□□H)

| Function | | Unit version | |
|---|---|---|---|
| | | Pre-Ver. 2.0 CPU Units | CPU Units Ver. 2.0 |
| Downloading and Uploading Individual Tasks | | --- | OK |
| Improved Read Protection Using Passwords | | --- | OK |
| Write Protection from FINS Commands Sent to CPU Units via Networks | | --- | OK |
| Online Network Connections without I/O Tables | | --- | OK |
| Communications through a Maximum of 8 Network Levels | | --- | OK |
| Connecting Online to PLCs via NS-series PTs | | OK from lot number 030201 | OK |
| Setting First Slot Words | | OK for up to 8 groups | OK for up to 64 groups |
| Automatic Transfers at Power ON without a Parameter File | | --- | OK |
| Automatic Detection of I/O Allocation Method for Automatic Transfer at Power ON | | --- | --- |
| Operation Start/End Times | | --- | OK |
| New Application Instructions | MILH, MILR, MILC | --- | OK |
| | =DT, <>DT, <DT, <=DT, >DT, >=DT | --- | OK |
| | BCMP2 | --- | OK |
| | GRY | OK from lot number 030201 | OK |
| | TPO | --- | OK |
| | DSW, TKY, HKY, MTR, 7SEG | --- | OK |
| | EXPLT, EGATR, ESATR, ECHRD, ECHWR | --- | OK |
| | Reading/Writing CPU Bus Units with IORD/IOWR | OK from lot number 030418 | OK |
| | PRV2 | --- | --- |

**CS1D CPU Units**

| Function | | CS1D CPU Units for Duplex-CPU Systems (CS1D-CPU□□H) | | CS1D CPU Units for Single-CPU Systems (CS1D-CPU□□S) |
|---|---|---|---|---|
| | | Pre-Ver. 1.1 CPU Units | CPU Unit Ver. 1.1 | CPU Unit Ver. 2.0 |
| Functions unique to CS1D CPU Units | Duplex CPU Units | OK | OK | --- |
| | Online Unit Replacement | OK | OK | OK |
| | Duplex Power Supply Units | OK | OK | OK |
| | Duplex Controller Link Units | OK | OK | OK |
| | Duplex Ethernet Units | --- | OK | OK |
| Downloading and Uploading Individual Tasks | | --- | --- | OK |
| Improved Read Protection Using Passwords | | --- | --- | OK |
| Write Protection from FINS Commands Sent to CPU Units via Networks | | --- | --- | OK |
| Online Network Connections without I/O Tables | | --- | --- | OK |
| Communications through a Maximum of 8 Network Levels | | --- | --- | OK |
| Connecting Online to PLCs via NS-series PTs | | --- | --- | OK |
| Setting First Slot Words | | --- | --- | OK for up to 64 groups |
| Automatic Transfers at Power ON without a Parameter File | | --- | --- | OK |
| Automatic Detection of I/O Allocation Method for Automatic Transfer at Power ON | | --- | --- | --- |
| Operation Start/End Times | | --- | OK | OK |
| New Application Instructions | MILH, MILR, MILC | --- | --- | OK |
| | =DT, <>DT, <DT, <=DT, >DT, >=DT | --- | --- | OK |
| | BCMP2 | --- | --- | OK |
| | GRY | --- | --- | OK |
| | TPO | --- | --- | OK |
| | DSW, TKY, HKY, MTR, 7SEG | --- | --- | OK |
| | EXPLT, EGATR, ESATR, ECHRD, ECHWR | --- | --- | OK |
| | Reading/Writing CPU Bus Units with IORD/IOWR | --- | --- | OK |
| | PRV2 | --- | --- | --- |

**CJ1-H/CJ1M CPU Units**

| Function | | CJ1-H CPU Units (CJ1□-CPU□□H) | | CJ1M CPU Units, except low-end models (CJ1M-CPU□□) | | CJ1M CPU Units, low-end models (CJ1M-CPU11/21) |
|---|---|---|---|---|---|---|
| | | Pre-Ver. 2.0 CPU Units | CPU Units Ver. 2.0 | Pre-Ver. 2.0 CPU Units | CPU Units Ver. 2.0 | CPU Units Ver. 2.0 |
| Downloading and Uploading Individual Tasks | | --- | OK | --- | OK | OK |
| Improved Read Protection Using Passwords | | --- | OK | --- | OK | OK |
| Write Protection from FINS Commands Sent to CPU Units via Networks | | --- | OK | --- | OK | OK |
| Online Network Connections without I/O Tables | | OK, but only if I/O table allocation at power ON is set | OK | OK, but only if I/O table allocation at power ON is set | OK | OK |
| Communications through a Maximum of 8 Network Levels | | OK for up to 8 groups | OK for up to 64 groups | OK for up to 8 groups | OK for up to 64 groups | OK for up to 64 groups |
| Connecting Online to PLCs via NS-series PTs | | OK from lot number 030201 | OK | OK from lot number 030201 | OK | OK |
| Setting First Slot Words | | --- | OK | --- | OK | OK |
| Automatic Transfers at Power ON without a Parameter File | | --- | OK | --- | OK | OK |
| Automatic Detection of I/O Allocation Method for Automatic Transfer at Power ON | | --- | OK | --- | OK | OK |
| Operation Start/End Times | | --- | OK | --- | OK | OK |
| New Application Instructions | MILH, MILR, MILC | --- | OK | --- | OK | OK |
| | =DT, <>DT, <DT, <=DT, >DT, >=DT | --- | OK | --- | OK | OK |
| | BCMP2 | --- | OK | OK | OK | OK |
| | GRY | OK from lot number 030201 | OK | OK from lot number 030201 | OK | OK |
| | TPO | --- | OK | --- | OK | OK |
| | DSW, TKY, HKY, MTR, 7SEG | --- | OK | --- | OK | OK |
| | EXPLT, EGATR, ESATR, ECHRD, ECHWR | --- | OK | --- | OK | OK |
| | Reading/Writing CPU Bus Units with IORD/IOWR | --- | OK | --- | OK | OK |
| | PRV2 | --- | --- | --- | OK, but only for models with built-in I/O | OK, but only for models with built-in I/O |

## Unit Versions and Programming Devices

CX-Programmer version 4.0 or higher must be used to enable using the functions added for CPU Unit Ver. 2.0. The following tables show the relationship between unit versions and CX-Programmer versions.

**Unit Versions and Programming Devices**

| CPU Unit | Functions | | CX-Programmer | | | Programming Console |
|---|---|---|---|---|---|---|
| | | | Ver. 3.2 or lower | Ver. 3.3 | Ver. 4.0 or higher | |
| CJ1M CPU Units, low-end models, Unit Ver. 2.0 | Functions added for unit version 2.0 | Using new functions | --- | --- | OK | No restrictions |
| | | Not using new functions | --- | OK | OK | |
| CS1-H, CJ1-H, and CJ1M CPU Units except low-end models, Unit Ver. 2.0 | Functions added for unit version 2.0 | Using new functions | --- | --- | OK | |
| | | Not using new functions | OK | OK | OK | |
| CS1D CPU Units for Single-CPU Systems, Unit Ver. 2.0 | Functions added for unit version 2.0 | Using new functions | --- | --- | OK | |
| | | Not using new functions | | | | |
| CS1D CPU Units for Duplex-CPU Systems, Unit Ver.1. | Functions added for unit version 1.1 | Using new functions | --- | --- | OK | |
| | | Not using new functions | OK | OK | OK | |

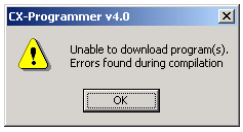
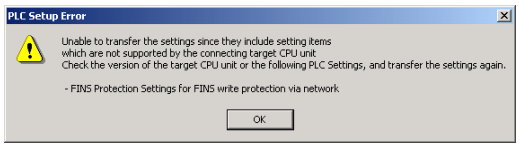
**Note** As shown above, there is no need to upgrade to CX-Programmer version 4.0 as long as the functions added for unit version 2.0 or unit version 1.1 are not used.

**Device Type Setting**

The unit version does not affect the setting made for the device type on the CX-Programmer. Select the device type as shown in the following table regardless of the unit version of the CPU Unit.

| Series | CPU Unit group | CPU Unit model | Device type setting on CX-Programmer Ver. 4.0 or higher |
|---|---|---|---|
| CS Series | CS1-H CPU Units | CS1G-CPU□□H | CS1G-H |
| | | CS1H-CPU□□H | CS1H-H |
| | CS1D CPU Units for Duplex-CPU Systems | CS1D-CPU□□H | CS1D-H (or CS1H-H) |
| | CS1D CPU Units for Single-CPU Systems | CS1D-CPU□□S | CS1D-S |
| CJ Series | CJ1-H CPU Units | CJ1G-CPU□□H | CJ1G-H |
| | | CJ1H-CPU□□H | CJ1H-H |
| | CJ1M CPU Units | CJ1M-CPU□□ | CJ1M |

## Troubleshooting Problems with Unit Versions on the CX-Programmer

| Problem | Cause | Solution |
|---|---|---|
| <br>After the above message is displayed, a compiling error will be displayed on the *Compile* Tab Page in the Output Window. | An attempt was made using CX-Programmer version 4.0 or higher to download a program containing instructions supported only by CPU Units Ver. 2.0 or later to a Pre-Ver. 2.0 CPU Units. | Check the program or change the CPU Unit being downloaded to a CPU Unit Ver. 2.0 or later. |
|  | An attempt was made using CX-Programmer version 4.0 or higher to download a PLC Setup containing settings supported only by CPU Units Ver. 2.0 or later (i.e., not set to their default values) to a Pre-Ver. 2.0 CPU Units. | Check the settings in the PLC Setup or change the CPU Unit being downloaded to a CPU Unit Ver. 2.0 or later. |
| "????" is displayed in a program transferred from the PLC to the CX-Programmer. | CX-Programmer version 3.3 or lower was used to upload a program containing instructions supported only by CPU Units Ver. 2.0 or later from a CPU Unit Ver. 2.0 or later. | The new instructions cannot be uploaded using CX-Programmer version 3.3 or lower. Use CX-Programmer version 4.0 or higher. |

# TABLE OF CONTENTS


**PRECAUTIONS . . . . . xxiii**

1 Intended Audience . . . . . xxiv
2 General Precautions . . . . . xxiv
3 Safety Precautions . . . . . xxiv
4 Operating Environment Precautions . . . . . xxv
5 Application Precautions . . . . . xxvi

**SECTION 1**
**Installation . . . . . 1**

1-1 Applicable Models . . . . . 2
1-2 Using the Programming Console . . . . . 3
1-3 Programming Console Functions . . . . . 5
1-4 Unsupported Functions . . . . . 7
1-5 Functions and Nomenclature . . . . . 7
1-6 Programming Console Connection and Installation . . . . . 14
1-7 Comparison with Previous Models . . . . . 16

**SECTION 2**
**Using the Programming Console . . . . . 17**

2-1 Programming . . . . . 18
2-2 Connecting the Programming Console . . . . . 19
2-3 Using the Programming Console for the First Time . . . . . 19
2-4 Inputting Programs . . . . . 22
2-5 Checking Program Operation . . . . . 26

**SECTION 3**
**Operation . . . . . 31**

3-1 Starting Operation . . . . . 32
3-2 Changing Operating Modes . . . . . 35
3-3 Key Functions . . . . . 36
3-4 Clearing Memory Areas . . . . . 37
3-5 Buzzer Operation . . . . . 42
3-6 Selecting Tasks . . . . . 43
3-7 I/O Table Operations . . . . . 46

**SECTION 4**
**Writing and Editing Programs . . . . . 53**

4-1 Inputting Instructions . . . . . 54
4-2 Reading and Searching Programs . . . . . 67
4-3 Editing Programs . . . . . 71

# TABLE OF CONTENTS


## SECTION 5
## Checking and Adjusting Programs . . . . . 79

5-1 Reading Program Execution Status . . . . . 80
5-2 Simple I/O Monitor . . . . . 81
5-3 I/O Multipoint Monitor . . . . . 87
5-4 Monitor Display Format . . . . . 90
5-5 Changing Word (16-bit) Present Values . . . . . 95
5-6 Forcing Bits ON/OFF . . . . . 101
5-7 Differential Monitor. . . . . 103
5-8 Online Editing . . . . . 104

## SECTION 6
## Maintenance Operations . . . . . 115

6-1 Clock Read/Change . . . . . 116
6-2 Cycle Time Read . . . . . 117
6-3 Reading/Clearing Error Messages . . . . . 118

## SECTION 7
## Memory Card Operations . . . . . 123

7-1 File Memory Operations . . . . . 124
7-2 Memory Card Format . . . . . 125
7-3 File Write . . . . . 126
7-4 File Read . . . . . 130
7-5 File Verify . . . . . 134
7-6 File Delete . . . . . 137

## SECTION 8
## PLC Setup Procedure . . . . . 141

8-1 PLC Setup Procedure. . . . . 142

## Appendices

A ASCII Coding Sheet . . . . . 145
B Error Messages . . . . . 147
C PLC Setup Coding Sheets for Programming Console . . . . . 149

## Index . . . . . 161

## Revision History . . . . . 165

# About this Manual:

This manual describes the installation and operation of the CQM1H-PRO01-E, CQM1-PRO01-E and C200H-PRO27-E Programming Consoles for use with CS-series and CJ-series Programmable Controllers and includes the sections described below. The CS Series and CJ Series are subdivided as shown in the following table.

| Unit | CS Series | CJ Series |
|---|---|---|
| CPU Units | CS1-H CPU Units: CS1H-CPU□□H<br>CS1G-CPU□□H | CJ1-H CPU Units: CJ1H-CPU□□H<br>CJ1G-CPU□□H |
| | CS1 CPU Units: CS1H-CPU□□-EV1<br>CS1G-CPU□□-EV1 | CJ1 CPU Units: CJ1G-CPU□□-EV1<br>CJ1M CPU Units: CJ1M-CPU□□ |
| | CS1D CPU Units:<br>CS1D CPU Units for Duplex-CPU Systems:<br>CS1D-CPU□□H<br>CS1D CPU Units for Single-CPU Systems:<br>CS1D-CPU□□S<br>CS1D Process CPU Units:<br>CS1D-CPU□□P | |
| Basic I/O Units | CS-series Basic I/O Units | CJ-series Basic I/O Units |
| Special I/O Units | CS-series Special I/O Units | CJ-series Special I/O Units |
| CPU Bus Units | CS-series CPU Bus Units | CJ-series CPU Bus Units |
| Power Supply Units | CS-series Power Supply Units<br>Note: Use the special CS1D Power Supply Units for CS1D PLCs. | CJ-series Power Supply Units |

Please read this manual and all related manuals listed in the table on the next page carefully and be sure you understand the information provided before attempting to use the CQM1H-PRO01-E, CQM1-PRO01-E, or C200H-PRO27-E Programming Console to program, set up, or operate a PLC System.

***Section 1*** gives a brief overview of the functions performed by the Programming Console, as well as Programming Console specifications and connection procedures. It also lists the differences between the operation of the Programming Consoles for CS/CJ-series PLCs and previous Programming Consoles.

***Section 2*** provides information on how to set up the Programming Console for operation. It also describes how to write a simple program from the Programming Console.

***Section 3*** describes the basic operations of the Programming Console, including initializing memory and creating I/O tables. The three input modes available on the Programming Console are also described.

***Section 4*** describes how to create and edit programs from the Programming Console.

***Section 5*** describes how to monitor programs in different display modes to check and modify the user programs. Change values from various displays is also described.

***Section 6*** includes information on reading and setting the clock, reading the cycle time, and reading/clearing error messages.

***Section 7*** provides information on how to format Memory Cards before use, and procedures for transferring data between Memory Cards and the CPU Unit.

***Section 8*** lists the settings in the PLC Setup.

***Appendix A*** provides a ASCII coding list, ***Appendix B*** provides a list of operating error messages, and ***Appendix C*** provides a PLC Setup Setting Sheets.

# About this Manual, Continued

| Name | Cat. No. | Contents |
|---|---|---|
| SYSMAC CS/CJ Series<br>Programming Consoles Operation Manual<br>CQM1H-PRO01-E, CQM1-PRO01-E,<br>C200H-PRO27-E | W341 | Provides information on how to program and operate CS/CJ-series PLCs using a Programming Console. (This manual) |
| SYSMAC CS Series<br>CS1G/H-CPU□□-EV1, CS1G/H-CPU□□H<br>Programmable Controllers Operation Manual | W339 | Describes the installation and operation of the CS-series PLCs. |
| SYSMAC CJ Series<br>CJ1G-CPU□□, CJ1G/H-CPU□□H<br>Programmable Controllers Operation Manual | W393 | Describes the installation and operation of the CJ-series PLCs. |
| SYSMAC CS Series<br>CS1D-CPU□□H CPU Units<br>CS1D-CPU□□S CPU Units<br>CS1D-DPL01 Duplex Unit<br>CS1D-PA/PD□□□ Power Supply Unit<br>Duplex System Operation Manual | W405 | Describes the installation and operation of the Duplex System based on CS1D CPU Units. |
| SYSMAC CS/CJ Series<br>CS1G/H-CPU□□-EV1, CS1G/H-CPU□□H,<br>CS1D-CPU□□H/S, CJ1G-CPU□□,<br>CJ1G/H-CPU□□H<br>Programmable Controllers Programming Manual | W394 | Describes the ladder diagram programming functions and other functions supported by CS-series and CJ-series PLCs. |
| SYSMAC CS/CJ Series<br>CS1G/H-CPU□□-EV1, CS1G/H-CPU□□H,<br>CS1D-CPU□□H/S, CJ1G-CPU□□,<br>CJ1G/H-CPU□□H<br>Programmable Controllers<br>Instructions Reference Manual | W340 | Describes the ladder diagram programming instructions supported by CS-series and CJ-series PLCs. |
| SYSMAC<br>CX-Programmer Operation Manual<br>WS02-CXP□□-E | W414 | Provides information on how to use the CX-Programmer, a programming device that supports the CS/CJ-series PLCs, and the CX-Net contained within CX-Programmer. |
| SYSMAC<br>CX-Programmer Operation Manual<br>WS02-CXP□□-EV4 | W425 | |

⚠ **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.

# Read and Understand this Manual

Please read and understand this manual before using the product. Please consult your OMRON representative if you have any questions or comments.

# Warranty and Limitations of Liability

## WARRANTY

OMRON's exclusive warranty is that the products are free from defects in materials and workmanship for a period of one year (or other period if specified) from date of sale by OMRON.

OMRON MAKES NO WARRANTY OR REPRESENTATION, EXPRESS OR IMPLIED, REGARDING NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR PARTICULAR PURPOSE OF THE PRODUCTS. ANY BUYER OR USER ACKNOWLEDGES THAT THE BUYER OR USER ALONE HAS DETERMINED THAT THE PRODUCTS WILL SUITABLY MEET THE REQUIREMENTS OF THEIR INTENDED USE. OMRON DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED.

## LIMITATIONS OF LIABILITY

OMRON SHALL NOT BE RESPONSIBLE FOR SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES, LOSS OF PROFITS OR COMMERCIAL LOSS IN ANY WAY CONNECTED WITH THE PRODUCTS, WHETHER SUCH CLAIM IS BASED ON CONTRACT, WARRANTY, NEGLIGENCE, OR STRICT LIABILITY.

In no event shall the responsibility of OMRON for any act exceed the individual price of the product on which liability is asserted.

IN NO EVENT SHALL OMRON BE RESPONSIBLE FOR WARRANTY, REPAIR, OR OTHER CLAIMS REGARDING THE PRODUCTS UNLESS OMRON'S ANALYSIS CONFIRMS THAT THE PRODUCTS WERE PROPERLY HANDLED, STORED, INSTALLED, AND MAINTAINED AND NOT SUBJECT TO CONTAMINATION, ABUSE, MISUSE, OR INAPPROPRIATE MODIFICATION OR REPAIR.

# Application Considerations

## SUITABILITY FOR USE

OMRON shall not be responsible for conformity with any standards, codes, or regulations that apply to the combination of products in the customer's application or use of the products.

At the customer's request, OMRON will provide applicable third party certification documents identifying ratings and limitations of use that apply to the products. This information by itself is not sufficient for a complete determination of the suitability of the products in combination with the end product, machine, system, or other application or use.

The following are some examples of applications for which particular attention must be given. This is not intended to be an exhaustive list of all possible uses of the products, nor is it intended to imply that the uses listed may be suitable for the products:

- Outdoor use, uses involving potential chemical contamination or electrical interference, or conditions or uses not described in this manual.
- Nuclear energy control systems, combustion systems, railroad systems, aviation systems, medical equipment, amusement machines, vehicles, safety equipment, and installations subject to separate industry or government regulations.
- Systems, machines, and equipment that could present a risk to life or property.

Please know and observe all prohibitions of use applicable to the products.

NEVER USE THE PRODUCTS FOR AN APPLICATION INVOLVING SERIOUS RISK TO LIFE OR PROPERTY WITHOUT ENSURING THAT THE SYSTEM AS A WHOLE HAS BEEN DESIGNED TO ADDRESS THE RISKS, AND THAT THE OMRON PRODUCTS ARE PROPERLY RATED AND INSTALLED FOR THE INTENDED USE WITHIN THE OVERALL EQUIPMENT OR SYSTEM.

## PROGRAMMABLE PRODUCTS

OMRON shall not be responsible for the user's programming of a programmable product, or any consequence thereof.

# Disclaimers

## CHANGE IN SPECIFICATIONS

Product specifications and accessories may be changed at any time based on improvements and other reasons.

It is our practice to change model numbers when published ratings or features are changed, or when significant construction changes are made. However, some specifications of the products may be changed without any notice. When in doubt, special model numbers may be assigned to fix or establish key specifications for your application on your request. Please consult with your OMRON representative at any time to confirm actual specifications of purchased products.

## DIMENSIONS AND WEIGHTS

Dimensions and weights are nominal and are not to be used for manufacturing purposes, even when tolerances are shown.

## PERFORMANCE DATA

Performance data given in this manual is provided as a guide for the user in determining suitability and does not constitute a warranty. It may represent the result of OMRON's test conditions, and the users must correlate it to actual application requirements. Actual performance is subject to the OMRON Warranty and Limitations of Liability.

## ERRORS AND OMISSIONS

The information in this manual has been carefully checked and is believed to be accurate; however, no responsibility is assumed for clerical, typographical, or proofreading errors, or omissions.

# PRECAUTIONS

This section provides general precautions for using the Programmable Controller (PLC) 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 PLC system.**


1 Intended Audience . . . . . xxiv
2 General Precautions . . . . . xxiv
3 Safety Precautions . . . . . xxiv
4 Operating Environment Precautions . . . . . xxv
5 Application Precautions . . . . . xxvi

# 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 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 PLC and all PLC 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 PLC System to the above-mentioned applications.

# 3 Safety Precautions

**⚠ WARNING** Do not attempt to take any Unit apart while the power is being supplied. Doing so may result in electric shock.

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

**⚠ Caution** The CPU Unit refreshes I/O even when operation has been stopped in PROGRAM mode. Always confirm safety before changing data in the output area allocated to the Output Units or changing data in any memory area allocated to Special I/O Units or CPU Bus Units using any of the following operations. The loads connected to the Output Units, Special I/O Units, or CPU Bus Units may operate unexpectedly.

- Transferring I/O memory to the CPU Unit using a peripheral device (personal computer software).
- Changing the present value using a peripheral device.
- Force-setting/resetting using a peripheral device.
- Transferring I/O memory files to the CPU Unit from the Memory Card or EM File Memory.

- Transferring I/O memory data from other personal computers or host computers on the network.

⚠ **Caution** Confirm that the equipment is operating safely before starting actual operation.

⚠ **Caution** Execute online edit only after confirming that no adverse effects will be caused by extending the cycle time. Otherwise, the input signals may not be readable.

⚠ **Caution** Confirm that no adverse effect will occur in the system before executing online edit.

# 4 Operating Environment Precautions

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

- Locations subject to direct sunlight.
- Locations subject to temperatures or humidity outside the range specified in the specifications.
- 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 PLC 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 PLC 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 PLC 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 class-3 ground (to 100 Ω or less) when installing the Units. Not connecting to a class-3 ground may result in electric shock.
- Always turn OFF the power supply to the PLC before attempting any of the following. Not turning OFF the power supply may result in malfunction or electric shock.
  - Mounting or dismounting I/O Units, CPU Unit, Power Supply Units, Inner Boards, or any other Units.
  - Assembling the Units.
  - Setting DIP switches or rotary switches.
  - Connecting or wiring the cables.
  - Connecting or disconnecting the connectors.

**⚠ Caution** Failure to abide by the following precautions could lead to faulty operation of the PLC or the system, or could damage the PLC or PLC 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.
- Install external breakers and take other safety measures against short-circuiting in external wiring. Insufficient safety measures against short-circuiting may result in burning.
- 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.
- Always use the power supply voltage specified in the operation manuals. An incorrect voltage may result in malfunction or burning.
- Take appropriate measures to ensure that the specified power with the rated voltage and frequency is supplied. Be particularly careful in places where the power supply is unstable. An incorrect power supply may result in malfunction.
- Do not apply voltages to the Input Units in excess of the rated input voltage. Excess voltages may result in burning.
- Do not apply voltages or connect loads to the Output Units in excess of the maximum switching capacity. Excess voltage or loads may result in burning.
- Wire the Unit correctly.
- Mount the Unit only after checking the terminal block completely.

- Use crimp terminals for wiring. Do not connect bare stranded wires directly to terminals. Connection of bare stranded wires may result in burning.
- Leave the label attached to the Unit when wiring. Removing the label may result in malfunction.
- Remove the label after the completion of wiring to ensure proper heat dissipation. Leaving the label attached may result in malfunction.
- Disconnect the functional ground terminal when performing withstand voltage tests. Not disconnecting the functional ground terminal may result in burning.
- Check the orientation and polarity of terminal blocks and connectors before connecting them.
- Be sure that the terminal blocks, expansion cables, and other items with locking devices are properly locked into place. Improper locking may result in malfunction.
- Double-check all the wiring before turning ON the power supply. Incorrect wiring may result in burning.
- 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 PLC.
  - Force-setting/force-resetting any bit in memory.
  - Changing the present value of any word or any set value in memory.
- Transfer any essential data for restarting the PLC, such as data memory and hold bits to the CPU Unit before restarting the PLC.
- 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.
- 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
# Installation

This section describes the Programming Console used with CS/CJ-series PLCs. It includes a brief overview of the functions performed by the Programming Console, as well as Programming Console installation procedures.

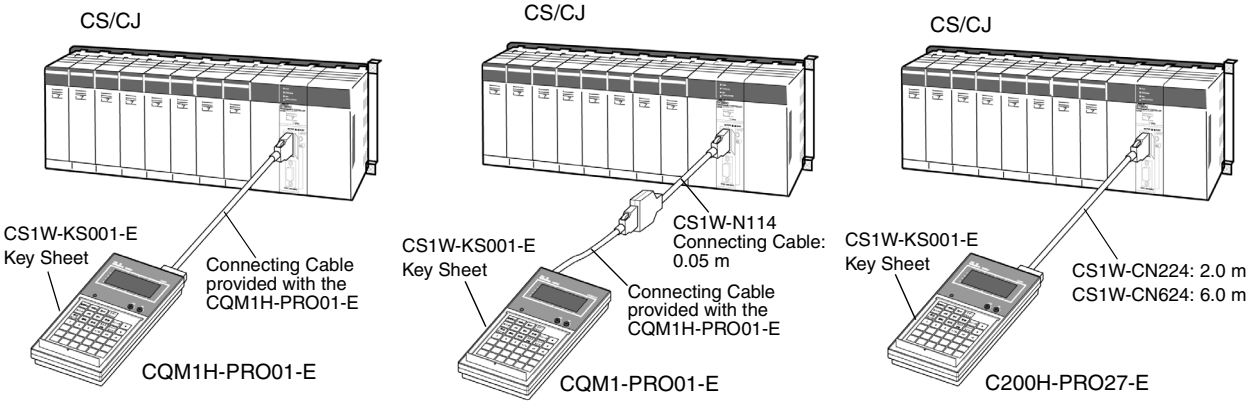

1-1 Applicable Models . . . . . 2
1-2 Using the Programming Console . . . . . 3
1-3 Programming Console Functions . . . . . 5
1-4 Unsupported Functions . . . . . 7
1-5 Functions and Nomenclature . . . . . 7
1-5-1 Nomenclature . . . . . 8
1-5-2 The Mode Switch and Operating Modes . . . . . 10
1-5-3 Key Functions . . . . . 11
1-6 Programming Console Connection and Installation . . . . . 14
1-7 Comparison with Previous Models . . . . . 16

# 1-1 Applicable Models

**Applicable Programming Consoles**

Any of the following Programming Consoles can be used with CS/CJ-series PLCs: CQM1H-PRO01-E, CQM1-PRO01-E and the C200H-PRO27-E. The Key Sheet and Connecting Cables listed below are required.

| Programming Console | Key Sheet | Connecting Cables |
|---|---|---|
| CQM1H-PRO01-E | CS1W-KS001-E | 2-m cable included |
| CQM1-PRO01-E | | CS1W-CN114 (0.05 m) |
| C200H-PRO27-E | | CS1W-CN224 (2.0 m) or CS1W-CN624 (6.0 m) |




The Programming Console is connected to the peripheral port on the CPU Unit. It cannot be connected to the RS-232C port.

In a CS1D Duplex-CPU System, the Programming Console is connected to the active CPU Unit.

**Applicable CPU Units**

| Series | CPU Units | Abbreviation in this manual |
|---|---|---|
| CS Series | CS1H-CPU6□-V1<br>CS1G-CPU4□-V1 | CS1 CPU Unit |
| | CS1H-CPU6□H<br>CS1G-CPU4□H | CS1-H CPU Unit |
| | CS1D-CPU6□H<br>CS1D-CPU□□S | CS1D CPU Unit |
| CJ Series | CJ1G-CPU4□ | CJ1 CPU Unit |
| | CJ1H-CPU6□H<br>CJ1G-CPU4□H | CJ1-H CPU Unit |
| | CJ1M-CPU□□ | CJ1M CPU Unit |

**Operational Differences for CPU Units**

The operation of the Programming Console will vary with the CPU Unit that is connected as shown in the following table. These are the only differences in Programming Console operation that vary with the CPU Unit.

| Operation | CS Series | | CJ Series | |
|---|---|---|---|---|
| | CS1 CPU Units | CS1-H CPU Units | CJ1 CPU Units | CJ1-H CPU Units |
| Operating mode when at startup (when PLC Setup is set to the default setting and the Programming Console is not connected) | PROGRAM | RUN | | |
| Selecting the display language | Pin 3 on DIP switch on front panel of CPU Unit | Programming Console key switch | | |

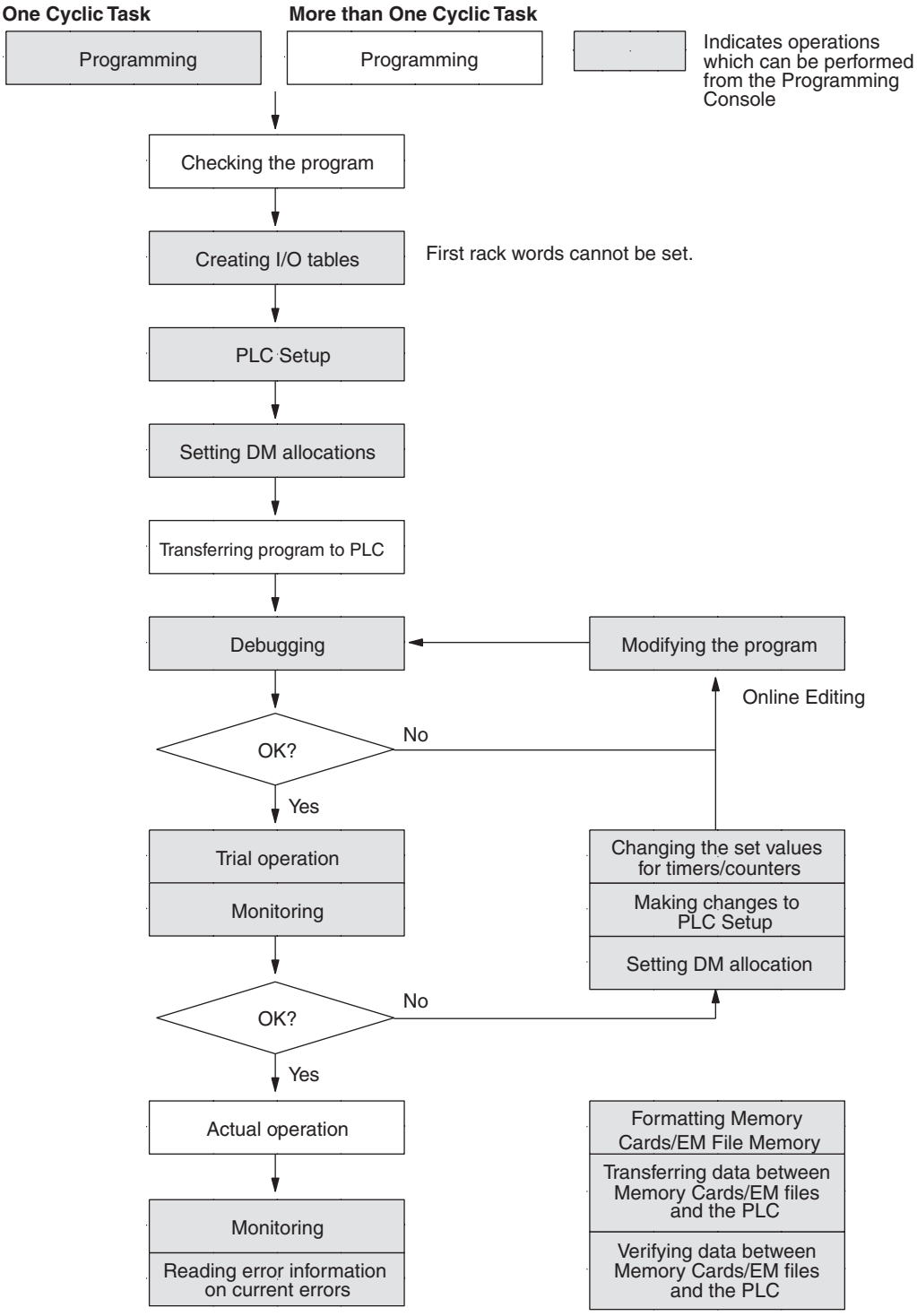
# 1-2 Using the Programming Console

**Programming Console**

The Programming Console for CS/CJ-series PLCs is used to write, to make on-site adjustments to, and to protect user programs. To create and edit relatively large user programs, the CX-Programmer (running on a Windows computer) should be used.

**Programming Console Features**

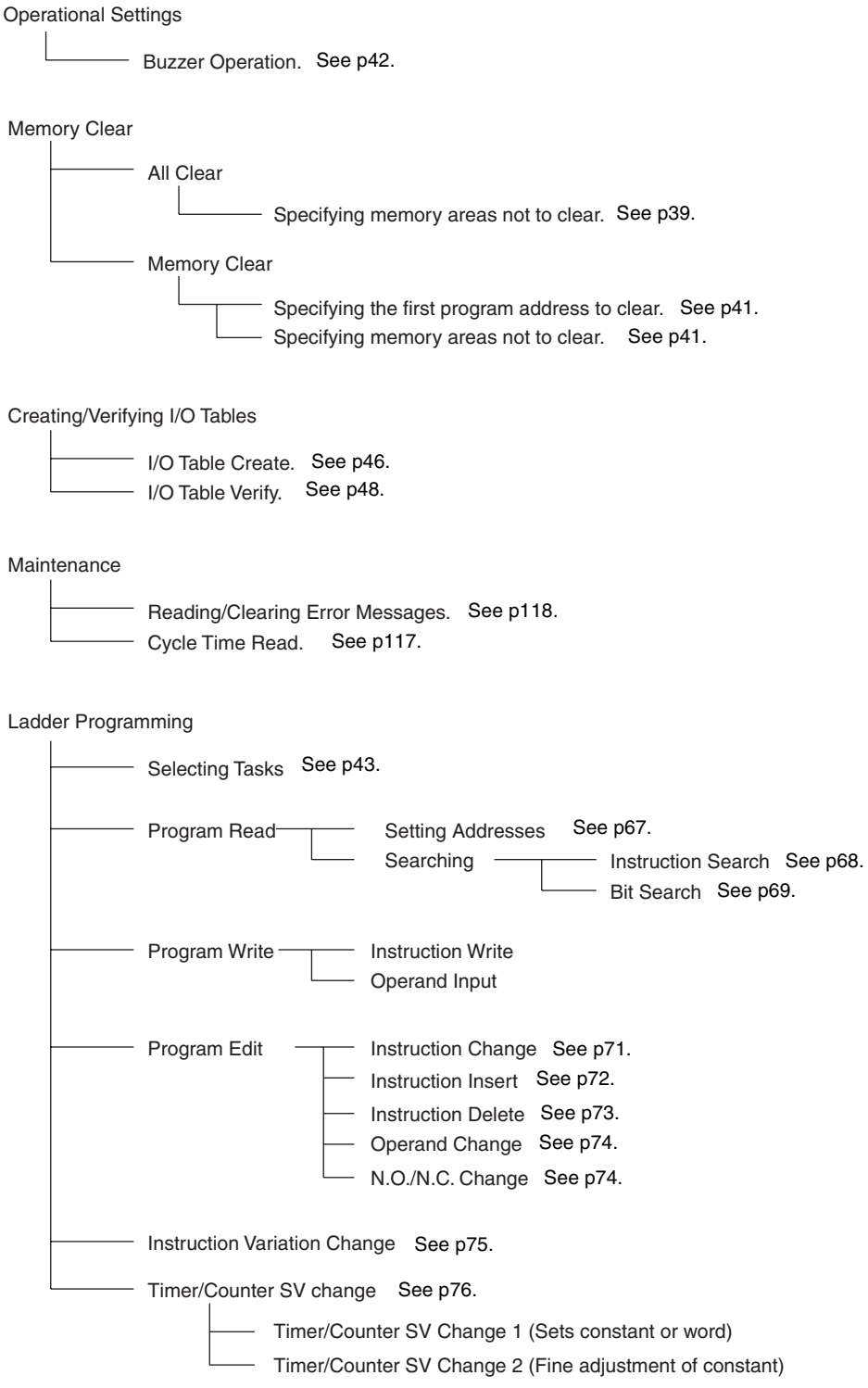
Programming is started by using the CX-Programmer, a programming and monitoring software package that runs on a Windows computer, to create the program. Programming is completed by debugging the program on the PLC. The Programming Console is used after programming has been completed to change the operating mode, change sections of the ladder program, monitor operation, change present values in I/O memory, change the PLC Setup, and read error information. The Programming Console can also be used to transfer and verify data between EM File Memory and the PLC.

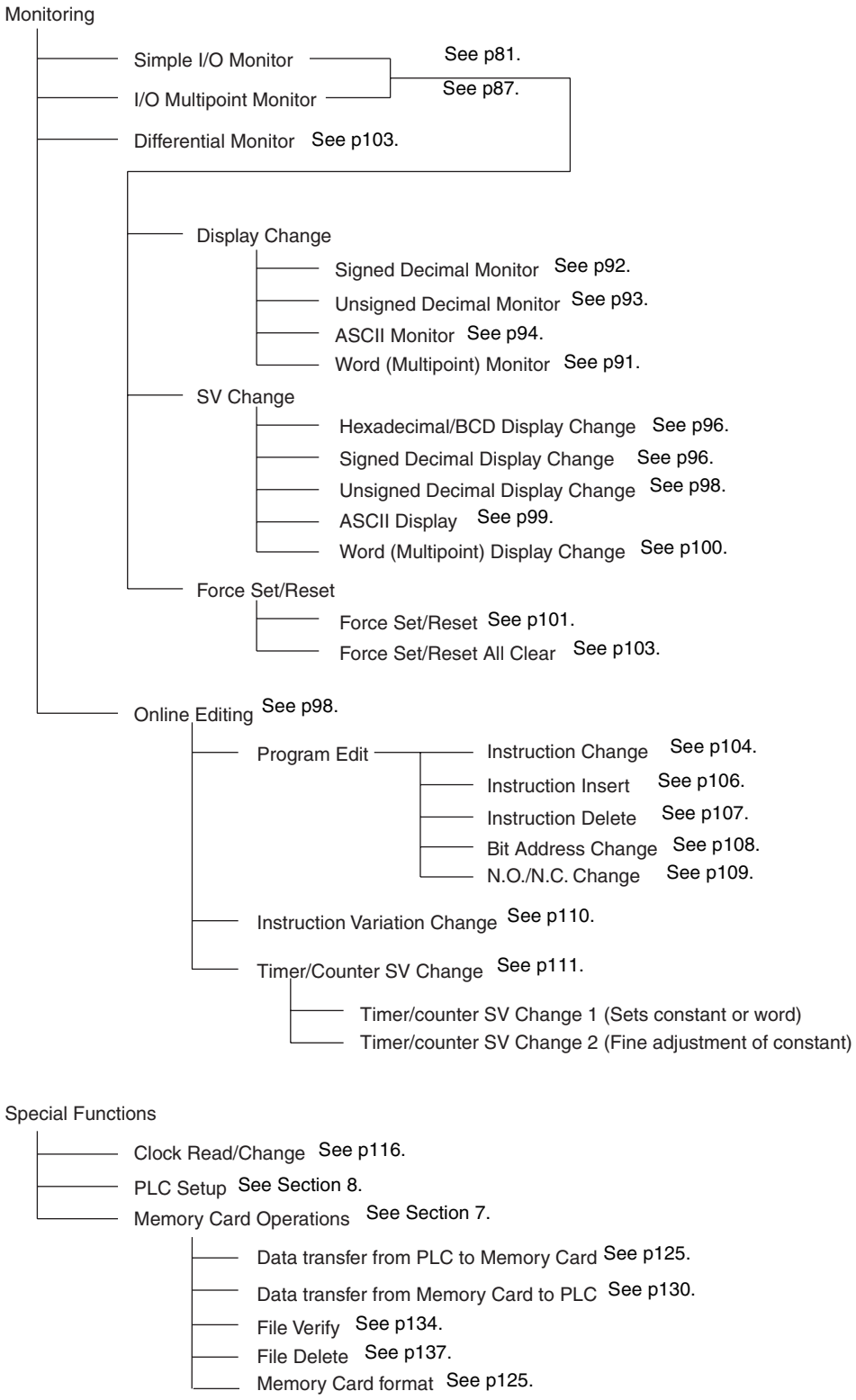



**Note** The Programming Console cannot be used to create programs with more than one cyclic task. Use the CX-Programmer if more than one cyclic task is required.

# 1-3 Programming Console Functions

The Programming Console performs the following functions.




Monitoring
Simple I/O Monitor
See p81.
I/O Multipoint Monitor
See p87.
Differential Monitor
See p103.
Display Change
Signed Decimal Monitor
See p92.
Unsigned Decimal Monitor
See p93.
ASCII Monitor
See p94.
Word (Multipoint) Monitor
See p91.
SV Change
Hexadecimal/BCD Display Change
See p96.
Signed Decimal Display Change
See p96.
Unsigned Decimal Display Change
See p98.
ASCII Display
See p99.
Word (Multipoint) Display Change
See p100.
Force Set/Reset
Force Set/Reset
See p101.
Force Set/Reset All Clear
See p103.
Online Editing
See p98.
Program Edit
Instruction Change
See p104.
Instruction Insert
See p106.
Instruction Delete
See p107.
Bit Address Change
See p108.
N.O./N.C. Change
See p109.
Instruction Variation Change
See p110.
Timer/Counter SV Change
See p111.
Timer/counter SV Change 1 (Sets constant or word)
Timer/counter SV Change 2 (Fine adjustment of constant)


Special Functions
Clock Read/Change
See p116.
PLC Setup
See Section 8.
Memory Card Operations
See Section 7.
Data transfer from PLC to Memory Card
See p125.
Data transfer from Memory Card to PLC
See p130.
File Verify
See p134.
File Delete
See p137.
Memory Card format
See p125.

# 1-4 Unsupported Functions

The following operations cannot be performed on the Programming Console. Use the CX-Programmer to perform these operations.

- Creating several cyclic tasks.
- Checking programs.
- Operations related to Communications Units.
- Displaying error history information.
- Setting the first rack words.
- Data tracing.
- Setting and clearing program read protection.

# 1-5 Functions and Nomenclature

**Handling 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 LCD display with care.
- The C200H-PRO27-E Programming Console has two connectors. Use only one of them at a time.
- Connect the cable to the Programming Console firmly until you hear it click into place, indicating that the cable is locked firmly in place.
- When removing the cable, hold the levers on both sides of the cable, release the lock, and pull the cable out.
- Do not pull or twist the cable with excessive force.
- The ambient operating temperature is 0 to 55°C. Be careful that this temperature is not exceeded when the Programming Console is used mounted to a panel.

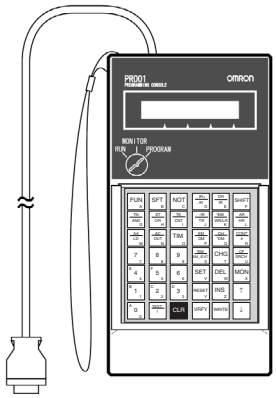
**Operating Environment**

Do not install or operate the Programming Console in any of the following locations.

- Locations subject to temperatures or humidities outside the ranges specified in the specifications.
- Locations directly subject to excessive shock.
- Locations subject to strong magnetic fields or electromagnetic waves.
- Locations subject to direct sunlight.

## 1-5-1 Nomenclature

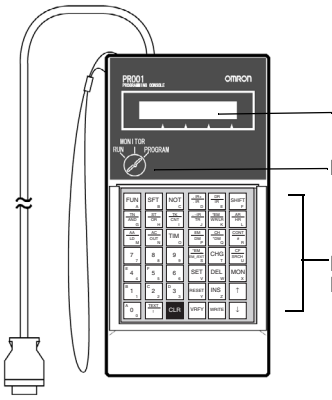
**CQM1H-PRO01-E**

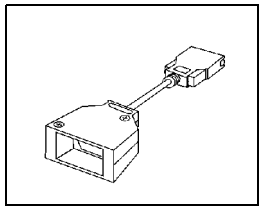


Cable length: 2 m

(The CQM1H-PRO01-E can be connected directly to the PLC.)

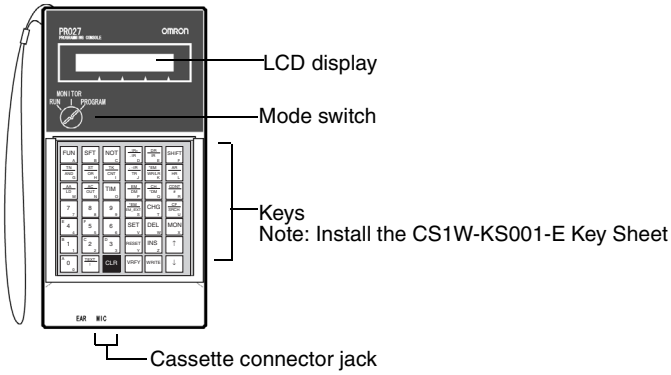
**CQM1H-PRO01-E**

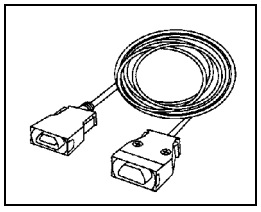


Connecting Cable
CS1W-CN114: 0.05 m

**C200H-PRO27-E**

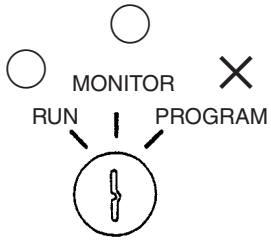


Connecting Cables
CS1W-CN224: 2.0 m
CS1W-CN624: 6.0 m

**Mode Switch**

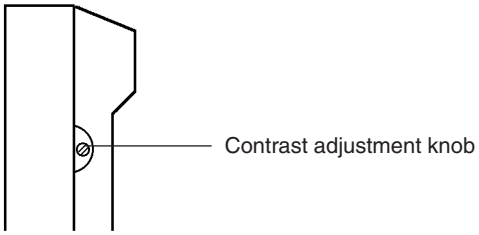
The mode switch key can be removed from the switch when the switch is in RUN or MONITOR modes. It cannot be removed when in PROGRAM mode.




○: Key can be removed
×: Key cannot be removed

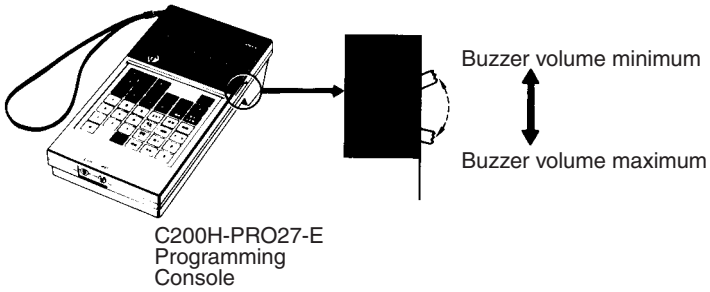
**Display Contrast**

The contrast of the liquid crystal display can be adjusted using the knob on the right of the display.




**Buzzer Volume**

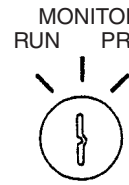
With the C200H-PRO27-E Programming Console, the buzzer volume can be adjusted using the lever on the right-hand side of the Console. The CQM1H-PRO01-E and CQM1-PRO01-E does not have buzzer volume control.




Refer to *3-5 Buzzer Operation* for further information on adjusting the buzzer volume.

## 1-5-2 The Mode Switch and Operating Modes

The relation between the operating mode of the CPU Unit and the mode switch is as follows:

| Key operation | Operating mode | Function |
|---|---|---|
|  | PROGRAM mode | The CPU Unit is stopped. Programming operations, such as writing or changing programs, clearing memory, and checking the program, can be performed. |
|  | MONITOR mode | The CPU Unit is operating and I/O processing is being performed. In this mode, CPU Unit operation can be monitored and functions such as forcing bits ON/OFF, changing timer/counter SV/PC, changing word data PVs, and online editing can be used. This mode is often used for making program adjustments and for trial system operations. |
|  | RUN mode | Used for normal operation of the CPU Unit. The operating status of the CPU Unit can be monitored in this mode, but functions such as forcing bits ON/OFF and changing PVs and SVs cannot be performed. |

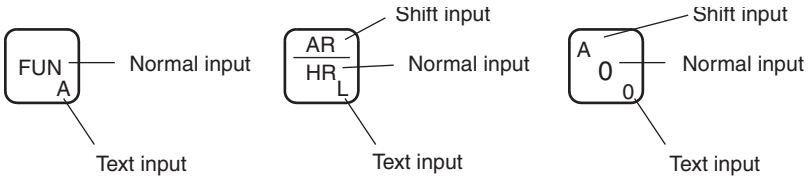
**Startup Operating Mode**

The operating mode of the CPU Unit when the power is turned ON depends on the status of address 81 in the PLC Setup (Startup Mode) and the connection status of peripheral devices.

| Startup Mode setting in PLC Setup (address 81) | Peripheral device | Startup operating mode |
|---|---|---|
| PRCN: Mode set on Programming Console's mode switch | Nothing connected | CS1 CPU Unit:<br>PROGRAM mode<br>CJ1/CJ1-H/CS1-H CPU Unit:<br>RUN mode |
| | Programming Console connected | The mode set on the mode switch on the Programming Console |
| | Peripheral device other than Programming Console connected | CS1 CPU Unit:<br>PROGRAM mode<br>CJ1/CJ1-H/CS1-H CPU Unit:<br>RUN mode |
| PRG: PROGRAM mode | Not relevant | PROGRAM mode |
| MON: MONITOR mode | Not relevant | MONITOR mode |
| RUN: RUN mode | Not relevant | RUN mode |

## 1-5-3 Key Functions

Each key has three possible inputs: The normal input, a shift input, and a text input. Refer to page 37.




- Press the **SHIFT** Key first to use the shift input.
- Press the **SHIFT** and then **TEXT** Keys first to use the text input.

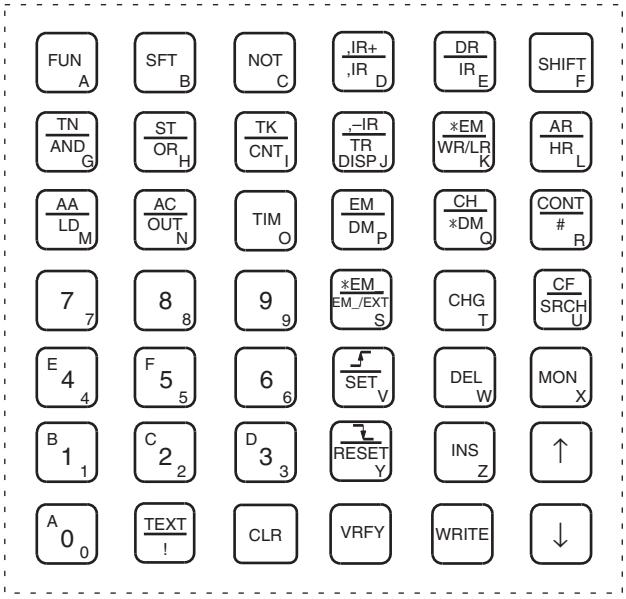
**Note** Keystrokes for procedures are illustrated using graphics of the buttons on the keypad for each step. The following list provides key names used in the text of this manual.

| Key | Name | Normal input | Shift input | Text input |
|---|---|---|---|---|
| FUN A | FUN Key | Function code<br>Select/change task<br>PLC Setup<br>(shift, monitor etc.) | --- | A |
| SFT B | SFT Key | SFT | --- | B |
| NOT C | NOT Key | NOT | --- | C |
| ,IR+ / ,IR D | ,IR+/,IR Key | ,IR<br>(Indirect index register) | ,IR+ | D |
| DR / IR E | DR/IR Key | IR<br>(Direct index register) | DR<br>(Data register) | E |
| SHIFT F | SHIFT Key | Alters the function of other keys. | --- | F |
| TN / AND G | TN/AND Key | AND | --- | G |
| ST / OR H | ST/OR Key | OR | --- | H |
| TK / CNT I | TK/CNT Key | CNT (Counter) | TK (Task flag) | I |
| ,–IR / TR DISP J | ,–IR/TR Key | Display changes<br>TR (Temporary bit) | ,–IR<br>(Index register auto-decrement) | J |
| ∗EM / WR/LR K | ∗EM/WR.LR Key | WR (Work bits)<br>LR (Link bits) | ∗EM<br>Indirect EM<br>(Extended data memory) | K |
| AR / HR L | AR/HR Key | HR (Holding bits) | AR<br>(Auxiliary bits) | L |

| Key | Name | Normal input | Shift input | Text input |
|---|---|---|---|---|
|  | AA/LD Key | LD | --- | M |
|  | AC/OUT Key | OUT | --- | N |
|  | TIM Key | TIM (Timer) | --- | O |
|  | EM/DM Key | DM (Data memory) | EM<br>(Data memory of current bank) | P |
|  | CH/*DM Key | *DM<br>(Indirect data memory) | CIO word | Q |
|  | CONT/# Key | #Constant<br>(Binary or BCD) | Operand<br>(Bit address) | R |
|  | *EM_/EM_.EXT Key | EM_<br>(Expansion Data Memory including bank number)<br>EXT (memory all clear) | *EM (indirect address) | S |
|  | CHG Key | CHG<br>(Changes to SVs, timer/counter etc.) | --- | T |
|  | CF/SRCH Key | SRCH | CF<br>(Condition Flag) | U |
|  | SET Key | SET | OFF-ON differentiation<br>Force Reset | V |
|  | DEL Key | DEL (Delete) | --- | W |
|  | MON Key | MON<br>(Simple I/O Monitor, I/O Multipoint Monitor, Memory area) | --- | X |
|  | RESET Key | RESET | ON-OFF differentiation<br>Force Reset | Y |
|  | INS Key | INS (Insert) | --- | Z |
|  | TEXT/! Key | !<br>(Immediate refresh) | Alphanumeric input mode | --- |
|  | CLR Key | Clear values<br>Return to previous value | --- | Returns to the normal input mode. |
|  | VRFY Key | VRFY (Verify) | --- | --- |
|  | WRITE Key | WRITE | --- | --- |

| Key | Name | Normal input | Shift input | Text input |
|---|---|---|---|---|
| ↑ | Up Key | Return to the previous program address<br>Delete/write | Differential Monitor | --- |
| ↓ | Down Key | Proceed to the next program address<br>Move to the next setting | Differential Monitor | --- |
| A 0 0<br>⋮<br>F 5 5<br>⋮<br>9 9 | Numeric Keys | 0<br>•<br>•<br>•<br>5<br>•<br>•<br>9 | Hexadecimal<br>A<br>•<br>•<br>•<br>F | 0<br>•<br>•<br>•<br>5<br>•<br>•<br>9 |

**Key Layout**



# 1-6 Programming Console Connection and Installation

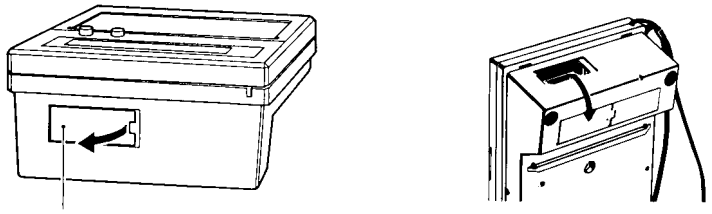
**Connection**

The Programming Console can be connected even when power is being supplied to the PLC and regardless of whether the CPU Unit is in RUN, MONITOR, or PROGRAM mode.

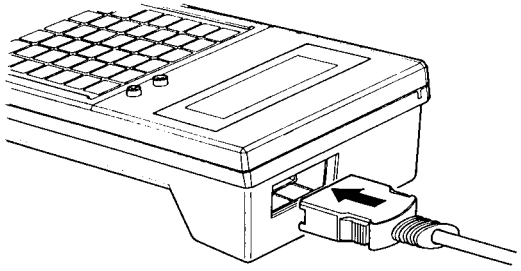
***1,2,3...***

1. For the C200H-PRO27-E Programming Console, connect the Connecting Cable's (CS1W-CN224 or CS1W-CN624) connector to the Programming Console.

   a) The C200H-PRO27-E has two cable connectors. Use one of these connectors and retain the detached cover as shown in the diagram below.

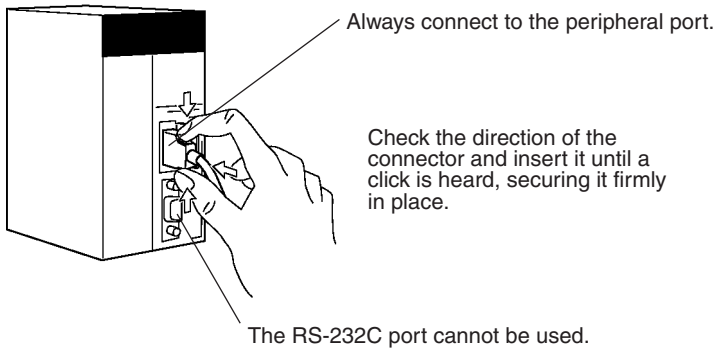



   b) Check the direction of the connector and insert it until you hear a click to secure it firmly in place.

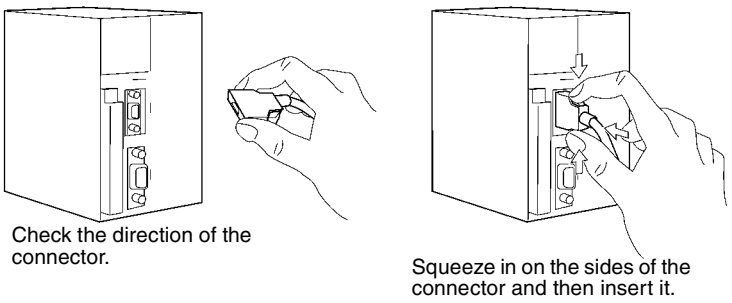


2. Connect the Programming Console Cable to the peripheral port. Do not connect it to the RS-232C port.

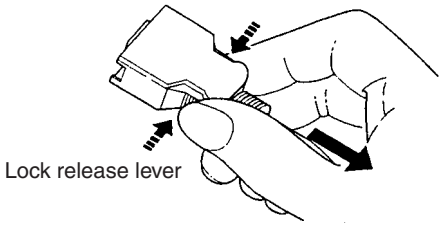
   **CS-series PLCs**



**CJ-series PLCs**

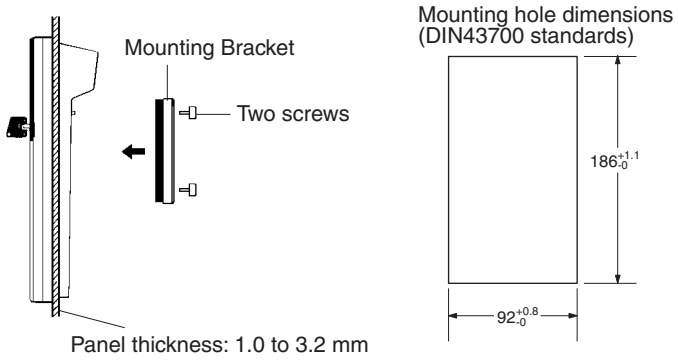



3. To disconnect the connector, squeeze the lock release levers on the sides and pull the connector straight out.

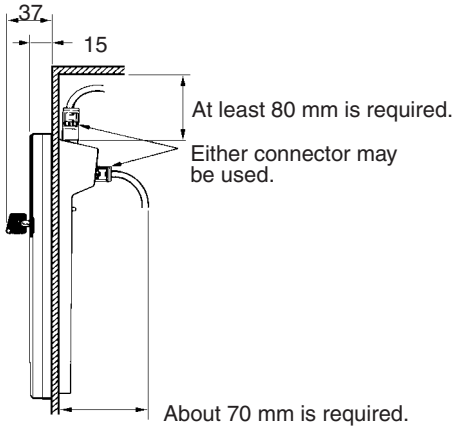



**Panel Mounting**

Use the C200H-PRO27-E Programming Console when the Programming Console needs to be mounted to a panel. Follow the instructions below for mounting the Console to a panel. (The C200H-ATT01 Mounting Bracket is sold separately.)

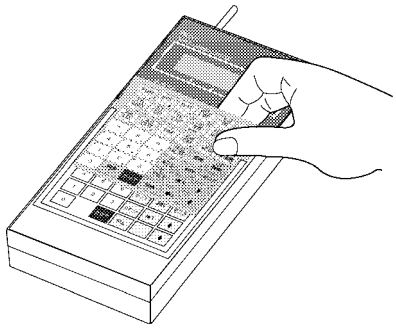



When mounting the Console to a panel, allow enough space for the cables.



**Attaching the Key Sheet**

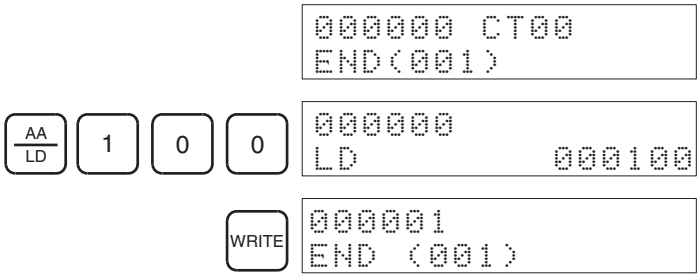
To attach the Key Sheet to the Programming Console, slide it under the grooves on the front of the Unit to insert. (The CS1W-KS001-E Key Sheet is sold separately.)



# 1-7 Comparison with Previous Models

## Inputting Programs with END(001) Instruction Displayed

When the END(001) instruction is displayed, programs can be input in the insert mode. In the following example, LD 000100 is input.




- When memory is cleared using All Clear, the END instruction will be written to the first address 000000. The NOP instruction is not written to all program addresses as in previous models.
- If the END instruction is deleted, ***ADR OVER*** will be displayed and a buzzer will sound.
- When programs are input, it is performed as an overwrite operation, as with previous models. Therefore to insert instructions into a program, this operation must be performed as an insert operation.

## I/O Multipoint Monitor Display

The I/O Multipoint Monitor display is 2 rows. In previous models it was a 3-column display.

## Start Mode

If default startup mode (i.e.,for the PLC to start up in the mode set on the Programming Console) is set in the PLC Setup and a Programming Console is not connected, a CS1 CPU Unit will start in PROGRAM mode, but a CJ1/CJ1-H/CS1-H CPU Unit PLC will start in RUN mode.

# SECTION 2
# Using the Programming Console

This section provides information on how to setup the Programming Console for operation. It also describes how to write a simple program from the Programming Console.


2-1 Programming . . . . . 18
2-2 Connecting the Programming Console . . . . . 19
2-3 Using the Programming Console for the First Time . . . . . 19
2-4 Inputting Programs . . . . . 22
2-5 Checking Program Operation . . . . . 26

# 2-1 Programming

When using Programming Consoles for a CS/CJ-series PLC for the first time, use the following procedure. Details are described in the sections *2-2 Connecting the Programming Console* through *2-3 Using the Programming Console for the First Time*.

***1,2,3...***

1. Turn OFF the power supply to the Power Supply Unit.
2. Install the RAM backup battery into the CPU Unit.
3. Set the DIP switch on the CPU Unit as follows:

   Pin 1: OFF (Data can be written to the UM area.)
   Pin 3: ON (English messages) (CS1 CPU Units only)
   Pin 4: OFF (Peripheral port automatic recognition.)

   Note Be sure to turn OFF pin 1 and pin 4 to enable writing the program via the Programming Console.
4. Connect the Programming Console to the CPU Unit.
5. Set the mode switch on the Programming Console to PROGRAM.
6. Be sure that the Programming Console's operating mode switch is set to PROGRAM, and then turn ON the power supply to the CPU Unit.

   Note Any program in the CPU Unit will be executed if the operating mode is not set to PROGRAM.
7. Select the display language (CJ1/CJ1-H/CS1-H CPU Units only).
8. Enter the password. (Press the **CLR** and **MON** Keys.)
9. Clear all memory.
10. Create I/O tables.

    Note a) For CS-series PLCs, I/O tables must be created.

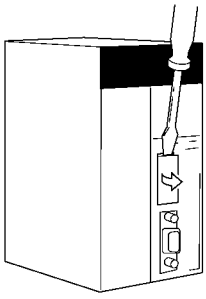
    b) For CJ-series PLCs, I/O tables must be created to detect incorrect Units or to register unused words. Otherwise, they are not required.
11. Read and clear error messages.
12. Start programming.

## 2-2 Connecting the Programming Console

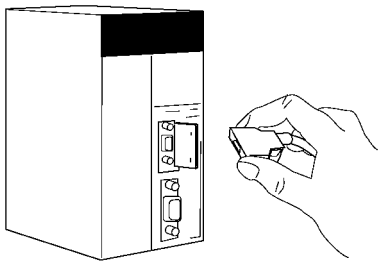
The Programming Console can be connected even when power is being supplied to the PLC and regardless of whether the CPU Unit is in RUN, MONITOR, or PROGRAM mode.

**Note** Always connect the Programming Console Cable into the peripheral port. Do not connect it to the RS-232C port.

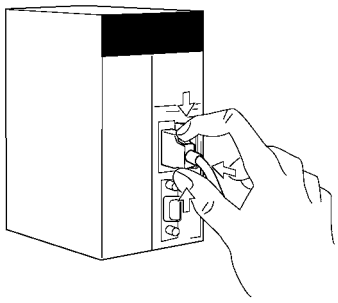
**CS-series**



Insert a small flat-blade screwdriver into the opening at the top of the Console and pull forwards.

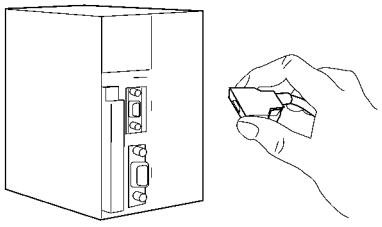


Check the direction of the connector.

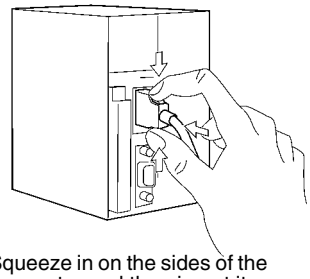


Connect by pressing the connector tab in place.

**CJ-series**



Check the direction of the connector.



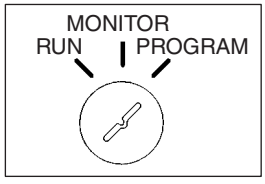
Squeeze in on the sides of the connector and then insert it.

## 2-3 Using the Programming Console for the First Time

When using the Programming Console for the first time, perform the following procedure.

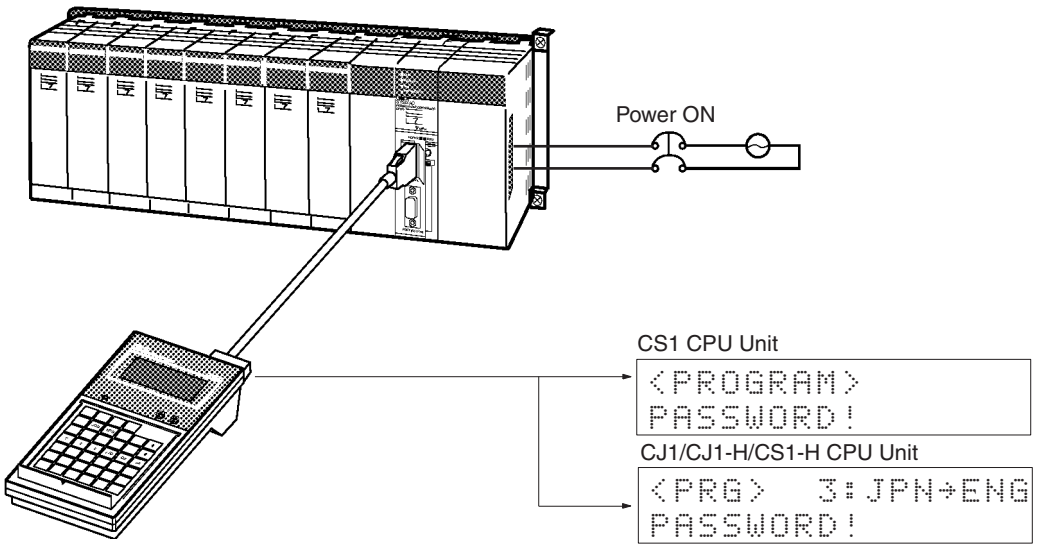
**Note** Keystrokes for procedures are illustrated using graphics of the keypad buttons for each step. A list of key names used in the text is provided in *Section 1 Installation* on page 11.

***1,2,3...*** 1. Make sure that the mode switch is set to PROGRAM, and then turn ON the power supply to the Power Supply Unit.

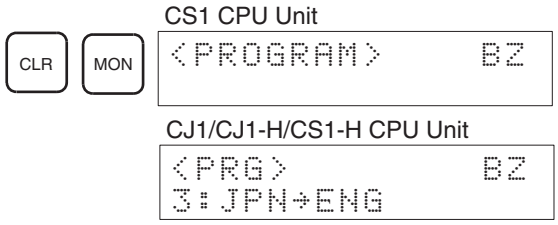


The Programming Console can be connected even when power is being supplied to the PLC and regardless of whether the CPU Unit is in RUN, MONITOR, or PROGRAM mode.

2. When the power is turned ON, the POWER indicator on the Power Supply Unit will light and the LCD display on the Programming Console will display the following.

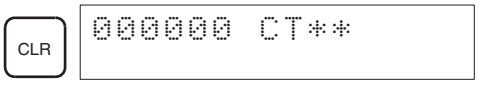



- Enter the password.




- The Programming Console is equipped with a password so that the PLC is operated only by experienced staff to ensure safe operation and prevent accidents. It will be necessary to enter the password by pressing the **CLR** Key and then the **MON** Key after the power is turned ON or after the Programming Console is connected. Refer to *3-1 Starting Operation* for details.
- The key buzzer can be turned OFF by pressing the **SHIFT** Key and then the **1** Key after entering the password.

3. Access the initial display.

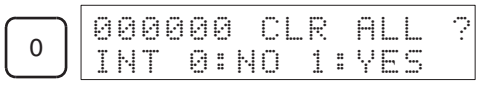



- If a memory error is displayed, press the **CLR** Key several times to return to the initial display.
- If an I/O table verification error occurs, ***I/O VRFY ERR*** will be displayed. Either connect the correct Unit, or press the **CLR** Key to ignore the error.

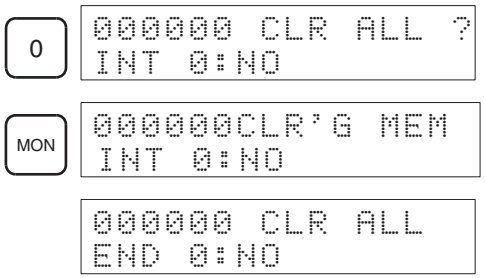
4. Clear all memory.
   - The memory clear operation initializes (formats) the memory area.




- To create one cyclic task only, select ***0: ALL***.




- To create one cyclic task and one or more interrupt tasks, select ***INT 1: YES***. To create one cyclic task only, select ***INT 0: NO***.

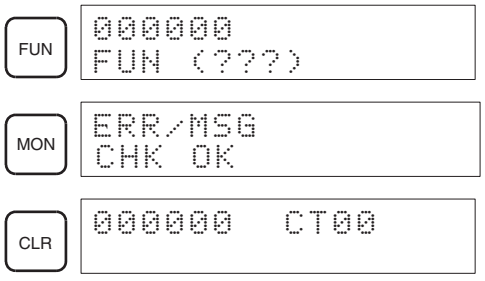



- When creating an interrupt task always select ***INT 1: YES*** when clearing memory. If ***INT 0: NO*** is selected, you will not be able to create interrupt tasks and it will be necessary to clear memory again using ***INT 1: YES***.

Note The Programming Console cannot be used to create programs with more than one cyclic task. Use the CX-Programmer if more than one cyclic task is required.

5. Create an I/O table using the procedure given on page 46. With CS1-series PLCs, an I/O table must be created.
6. Read/clear error messages.

   If an error message is displayed, pressing the **MON** Key after removing the cause of the error will clear the error message display. If there are several error messages, repeat this operation several times.

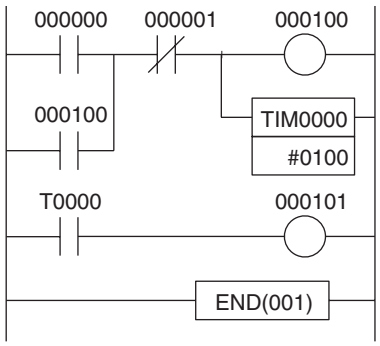


# 2-4 Inputting Programs

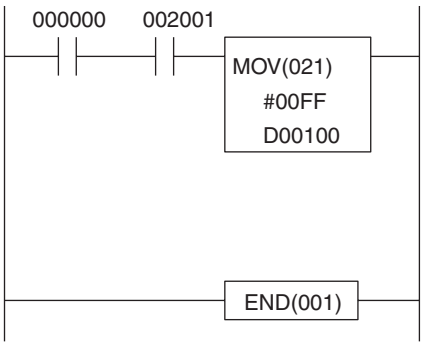
When preparations for operation have been completed, input the following simple program to get accustomed to using the Programming Console.

The following programs will be used in this example.

Cyclic task No. 0 (CT00)




Interrupt task No. 2 (IT002)




Mnemonic

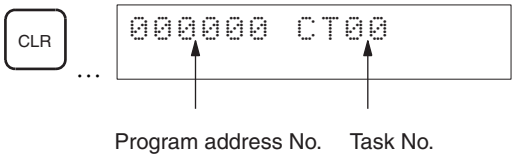
| Program address | Instruction | Operand |
|---|---|---|
| 000000 | LD | 000000 |
| 000001 | OR | 000100 |
| 000002 | AND NOT | 000001 |
| 000003 | OUT | 000100 |
| | TIM | 0000 |
| 000004 | | #0100 |
| 000005 | LD | T0000 |
| 000006 | OUT | 000101 |
| 000007 | END(001) | --- |

Mnemonic

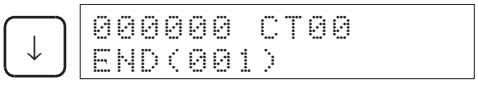
| Program address | Instruction | Operand |
|---|---|---|
| 000000 | LD | 000000 |
| 000001 | AND | 002001 |
| 000002 | MOV(021) | |
| 000003 | --- | #00FF |
| 000004 | --- | D00100 |
| 000005 | END(001) | --- |

- When inputting the above program, always select ***1: YES*** when clearing memory to enable inputting the interrupt task. Refer to page 20.
- Refer to the *CS1 Operation Manual* for further details on programming.

***1,2,3...*** 1. Access the initial display.




2. Press the **Down** Key.

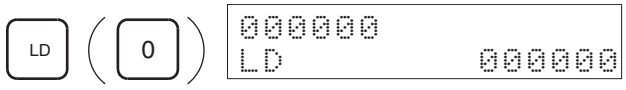



When the **Down** Key is pressed, the END instruction will be generated. When all memory has been cleared, the END instruction will be written to program address 0 of all tasks.

New programs will be written from the address where the END instruction is displayed.

3. Input the cyclic task program using the following steps.

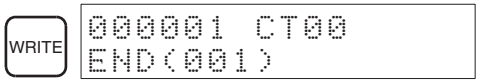
a) Input LD 000000.




Leading zeros do not need to be entered.

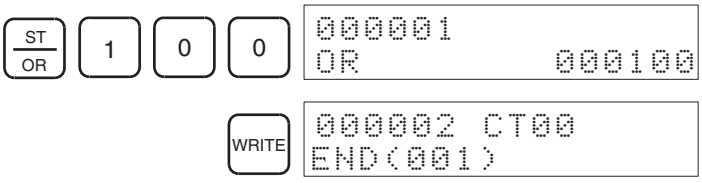
b) Press the **WRITE** Key.

***Write*** will be displayed temporarily. When inputting is complete, the address will move forward by one and the END instructions will automatically be sent to the next address.

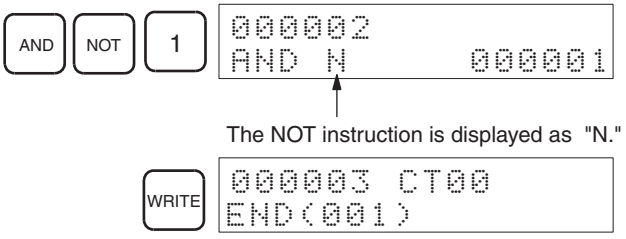



Note Inputs are generally written into memory by overwriting the current memory contents. Inputs are inserted only when the END(001) instruction is displayed on the Programming Console.

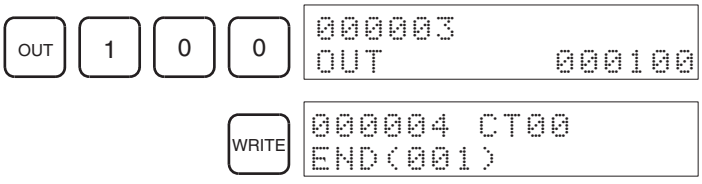
c) Input OR 000100



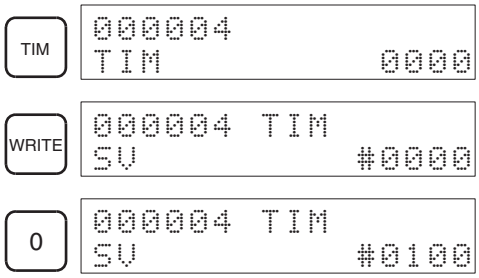

d) Input AND NOT 000001

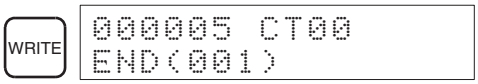



e) Input OUT 000100.

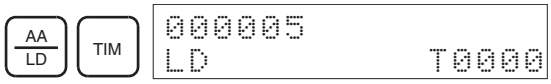


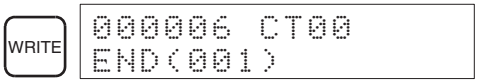

f) Input the TIM (timer) instruction.



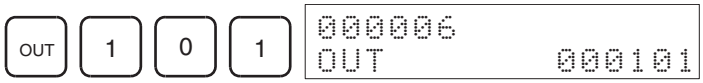


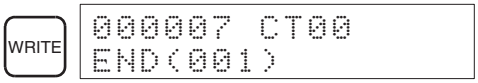

g) Input LD TIM 00000.



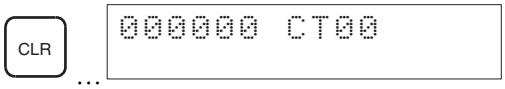

h) Input OUT 000101



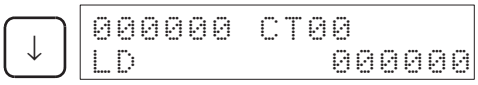



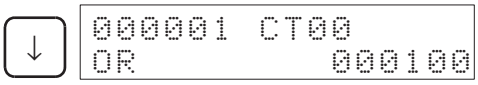

4. Read and check the cyclic task program using the following steps.

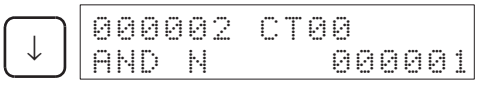
a) Access the initial display.

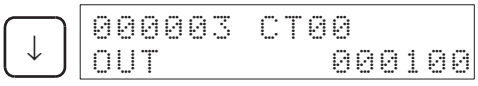


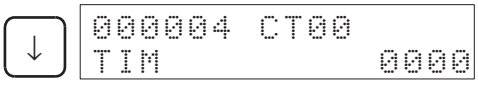

b) Press the **Down** Key, read the program in order, and confirm that the mnemonics are correct. Correct any errors in the program.

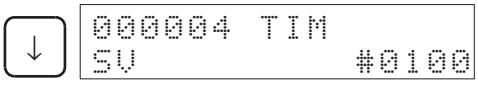


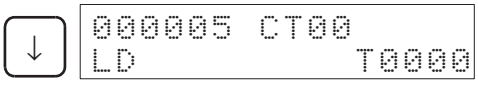



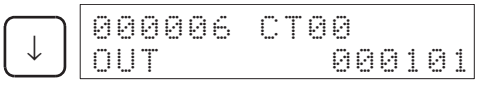



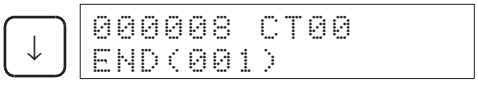



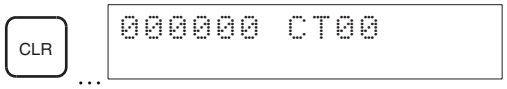



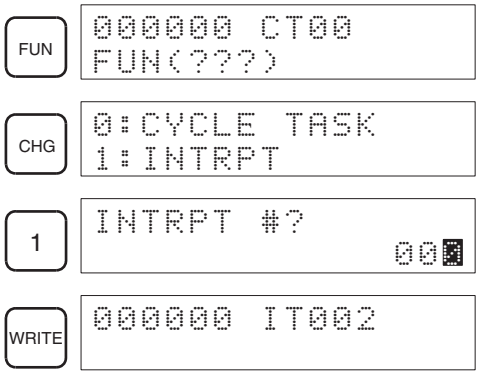



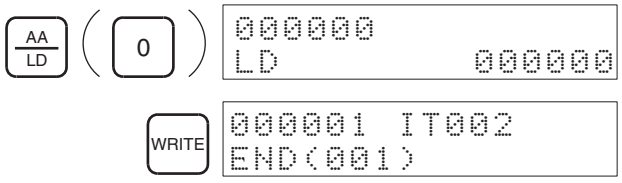
5. Switch to interrupt task using the following steps.
   a) Access the initial display.



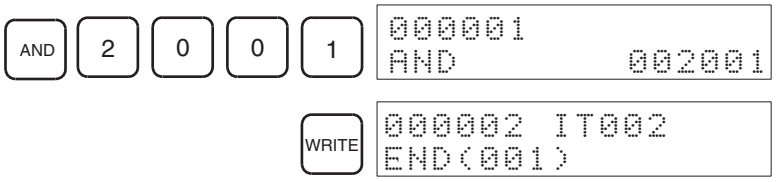

   b) Switch tasks.



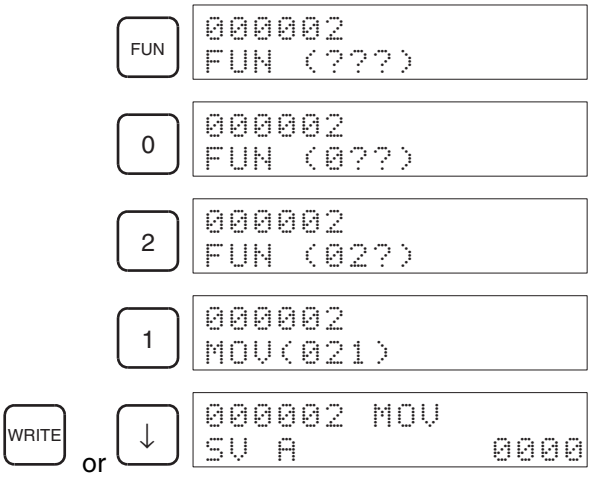

6. Input the interrupt task program using the following steps.
   a) Input LD 000000




   b) Input AND 002001.



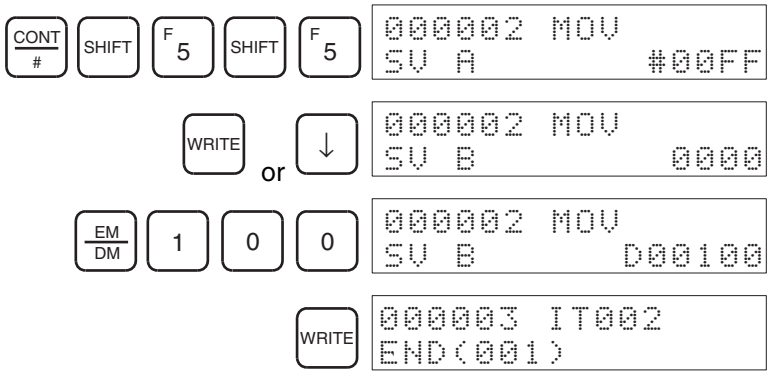

   c) Input MOV(021).




Note When writing special instructions be sure to input the function number correctly as a 3-digit number. If the number is abbreviated to “0,”

it cannot be written. Press the **FUN** Key once more and input correctly.

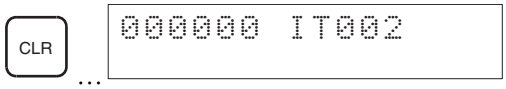
d) Input the data to be sent.




Note Always press the **WRITE** Key after inputting the last set value. Write cannot be performed using the **Down** Key.

7. Read and check the interrupt task program using the following steps.

a) Bring up the initial display.




b) Press the **Down** Key, read the program in order, and confirm that the mnemonics are correct. Correct any errors in the program.

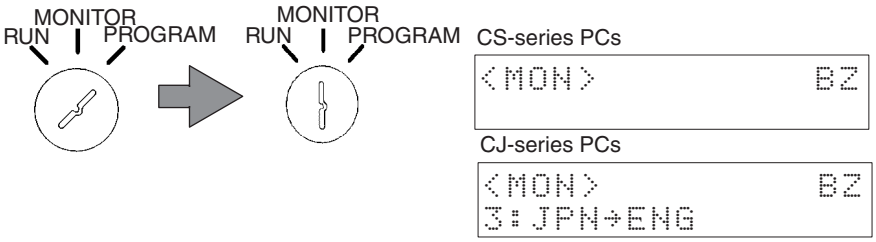



## 2-5 Checking Program Operation

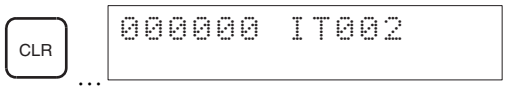
After checking that the program has been written correctly, perform a trial execution of the program. It is not necessary to have I/O Units installed, but when the PLC is switched to MONITOR mode, operation will start, so if I/O Units are connected, either remove the output wires or set the Output OFF Bit to ON. Refer to page 35.

***1,2,3...*** 1. Switch to MONITOR mode by setting the mode switch on the Programming Console to MONITOR mode.

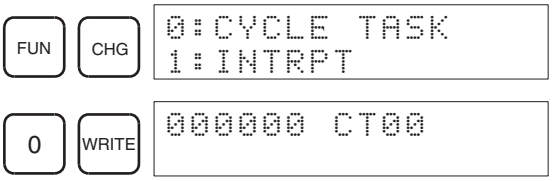
The PLC will start executing the program in MONITOR mode.




2. Access the initial display.



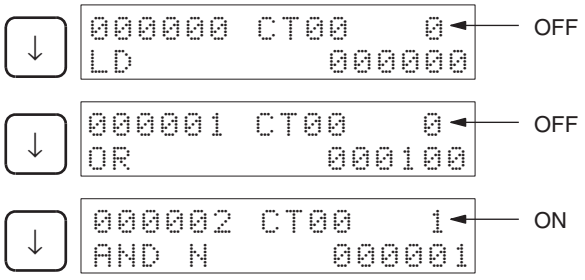

3. Switch to the cyclic task.



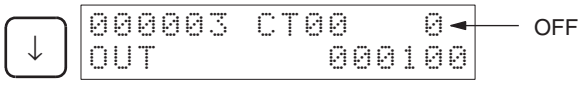

4. Monitor the I/O bit status step by step during program execution.

   Press the **Up** and **Down** Keys to read the program and monitor I/O bit status as shown below. This operation is called monitoring I/O bit status.

   The status resulting from one complete cycle of execution is shown on the display when monitoring I/O bit status, not the status during program execution.



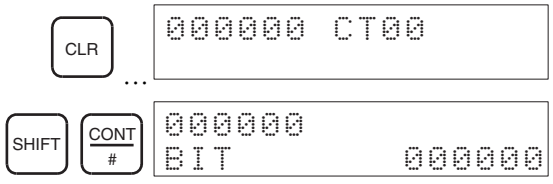

   When a normally closed bit is OFF, the I/O bit status will be displayed as "1."

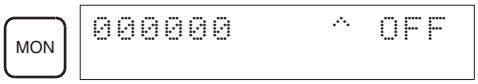



5. Monitor bit status as described next.

   Check to see whether or not CIO 000100 turns ON and holds itself ON when CIO 000000 is turned ON.

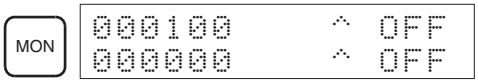
   a) Monitor CIO 000000. (Simple I/O Monitor)






b) Monitor CIO 000100 at the same time. (I/O Multipoint Monitor)

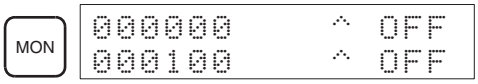



- The most recently monitored bits will be displayed first.
- I/O Multipoint Monitor can be used to monitor not only bits, but also words (16-bit data), data memory, and timer/counter present values. Up to 4 items can be specified.

6. Force ON CIO 000000 as described next.

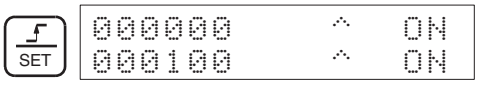
a) Press the **MON** Key until CIO 000000 is displayed at the top.




For I/O Multipoint Monitor, the uppermost bit is force set/reset.

b) Force ON CIO 000000.

When the **SET** Key is pressed:



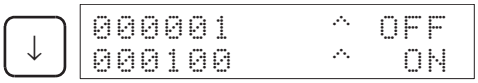

When the **SET** Key is released:




- When there is no I/O Unit at word 0, the status is held ON. When the **RESET** Key is pressed, the status is set to OFF.
- The uppermost bit CIO 000000 is only set to ON, while the **SET** Key is being pressed.
- The program is being executed so when CIO 000000 is ON CIO 000100 is also ON. (Even when CIO 000000 is OFF, CIO 000100 will hold ON.)

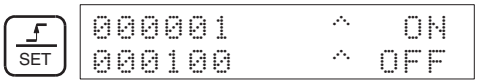
7. Force OFF CIO 000001 as described next.

a) Monitor CIO 000001. To monitor CIO 000001, press the **Down** Key at the monitor display for CIO 000000.




b) Use Force Set to set normally closed bit 000001 to OFF.

When the **SET** key is pressed:



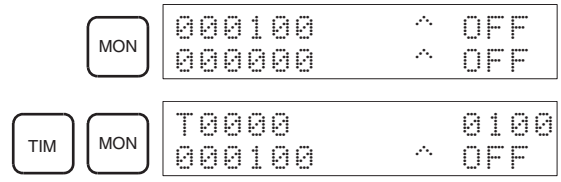

When the **SET** Key is released:



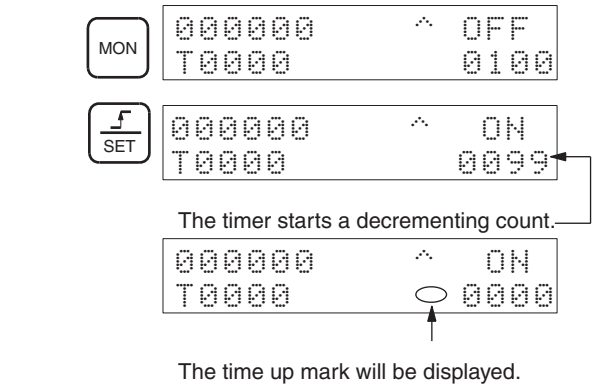

- When there is no I/O Unit at word 0, the status is held ON. When the **RESET** Key is pressed, the status is set to OFF.
- When the normally closed CIO 000001 is set to OFF, the self-holding status is cleared, and CIO 000100 is set to OFF also.

8. Check timer operation using the following steps.

a) Monitor TIM 0000.

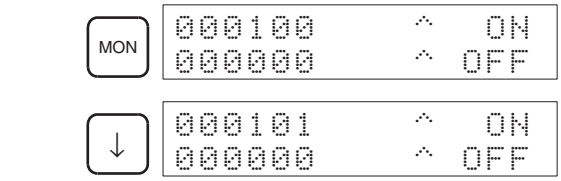



b) Move bit CIO 000000 to the top.




The time up mark will be displayed.

c) Check bit CIO 000101.



# SECTION 3
# Operation

This section describes the basic operations of the Programming Console, including initializing memory and creating I/O tables. It also covers the 3 input modes available on the Programming Console.


3-1 Starting Operation . . . . . 32
3-1-1 CS1 CPU Units . . . . . 32
3-1-2 CJ1/CJ1-H/CS1-H CPU Units . . . . . 33
3-2 Changing Operating Modes . . . . . 35
3-3 Key Functions . . . . . 36
3-4 Clearing Memory Areas . . . . . 37
3-4-1 All Clear . . . . . 39
3-4-2 Memory Clear . . . . . 41
3-5 Buzzer Operation . . . . . 42
3-6 Selecting Tasks . . . . . 43
3-7 I/O Table Operations . . . . . 46
3-7-1 I/O Table Create . . . . . 46
3-7-2 I/O Table Verify . . . . . 48

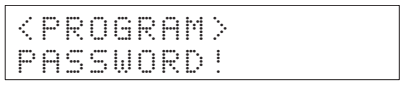
# 3-1 Starting Operation

## 3-1-1 CS1 CPU Units

A password must be entered before any key operations can be performed. This is necessary whenever the power to the Power Supply Unit is turned ON when the Programming Console is connected to the CPU Unit, or when the Programming Console is connected while the power is ON.

***1,2,3...***

1. When the power to the PLC is turned ON, the POWER indicator on the CPU Unit will light and the LCD display on the Programming Console will display the following.



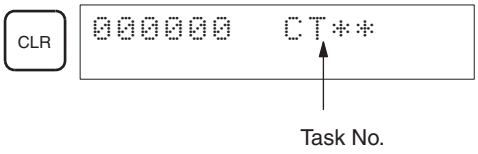

2. Enter the password (i.e., the **CLR** and **MON** Keys). The mode display will appear.




Note The Programming Console is equipped with a password so that the PLC is operated only by qualified staff to ensure safe operation and prevent accidents.

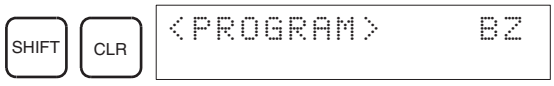
- After entering the password, Programming Console key operations can be performed and the mode can be changed.
- The key buzzer can be turned OFF by pressing the **SHIFT** Key and then the **1** Key after entering the password. Refer to *3-5 Buzzer Operation*.

3. Press the **CLR** Key to move from the mode display to the initial display.




Note When the power is turned ON for the first time (before All Clear has been performed), "CT∗∗" will be displayed as the task number.

- When a memory error is displayed, press the **CLR** Key several times to go to the initial display.
- To return to the mode display from the initial display, press the **SHIFT** then **CLR** Keys.




Note If the Programming Console display is in Japanese for a CS1 CPU Unit, turn the PLC OFF once and change the CPU Unit's DIP switch setting. Refer to page 18.

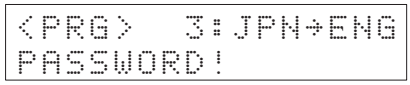
## 3-1-2 CJ1/CJ1-H/CS1-H CPU Units

A password must be entered before any key operations can be performed. This is necessary whenever the power to the Power Supply Unit is turned ON when the Programming Console is connected to the CPU Unit, or when the Programming Console is connected while the power is ON.

The language that Programming Console messages will be displayed can be selected from the password or mode display. (There is no DIP switch setting for the language, as there is for the CS1 CPU Units.)

***1,2,3...***

1. When the power to the PLC is turned ON, the POWER indicator on the CPU Unit will light and the LCD display on the Programming Console will display the following.



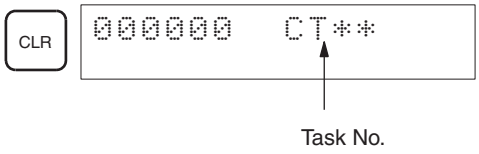

2. Press the **3** Key to switch between Japanese (JPN) and English (ENG) language displays or messages will be displayed in Japanese.
3. Enter the password (i.e., the **CLR** and **MON** Keys). The mode display will appear




Note The Programming Console is equipped with a password so that the PLC is operated only by qualified staff to ensure safe operation and prevent accidents.

- After entering the password, Programming Console key operations can be performed, the mode can be changed, and the display language can be selected.
- The key buzzer can be turned OFF by pressing the **SHIFT** Key and then the **1** Key after entering the password. Refer to *3-5 Buzzer Operation*.

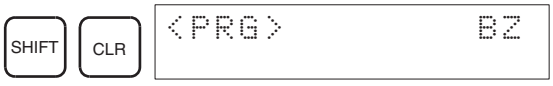
4. Press the **CLR** Key to move from the mode display to the initial display.

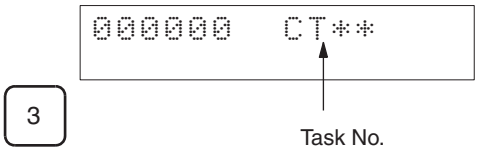



Note a) When the power is turned ON for the first time (before All Clear has been performed), "CT**" will be displayed as the task number.

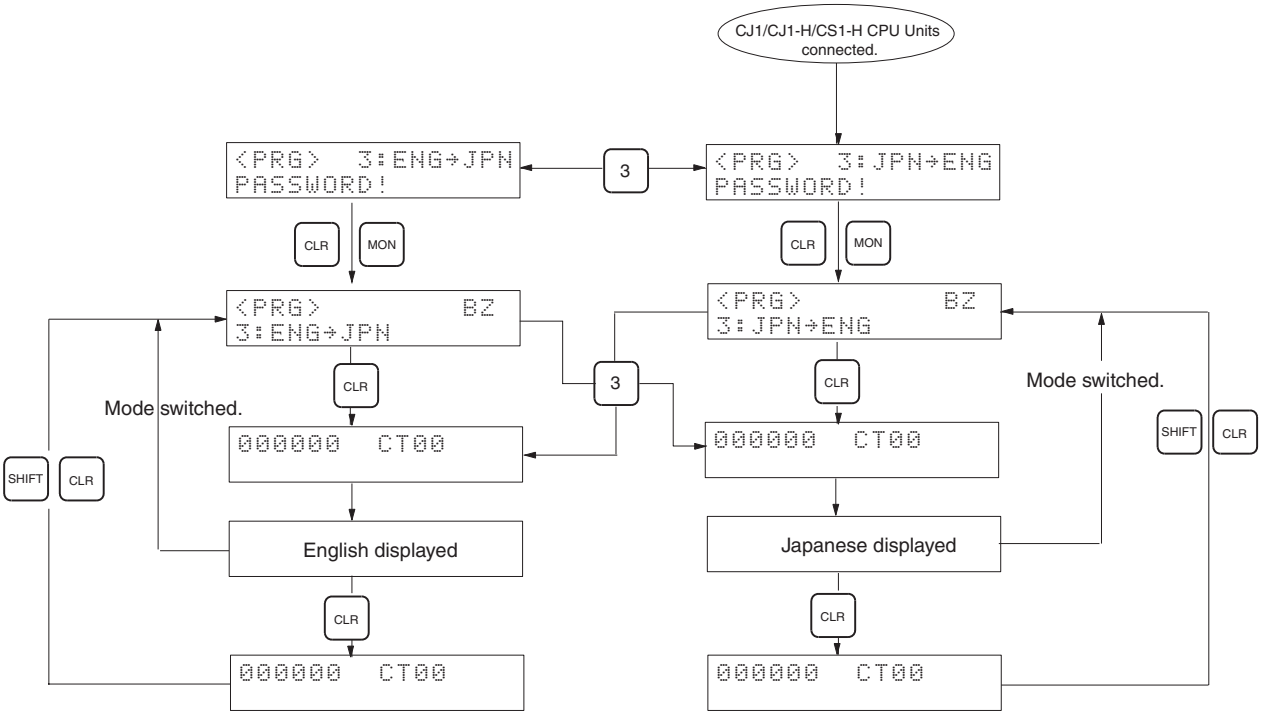
b) The **3** Key can be pressed instead of the **CLR** Key to access the initial display in the other language. Refer to the diagram on the next page.

- When a memory error is displayed, press the **CLR** Key several times to go to the initial display.
- To return to the mode display from the initial display, press the **SHIFT** then **CLR** Keys.






**Note** The password input, language selection, and initial displays can be switched for CJ1/CJ1-H/CS1-H CPU Units as shown in the following diagram.

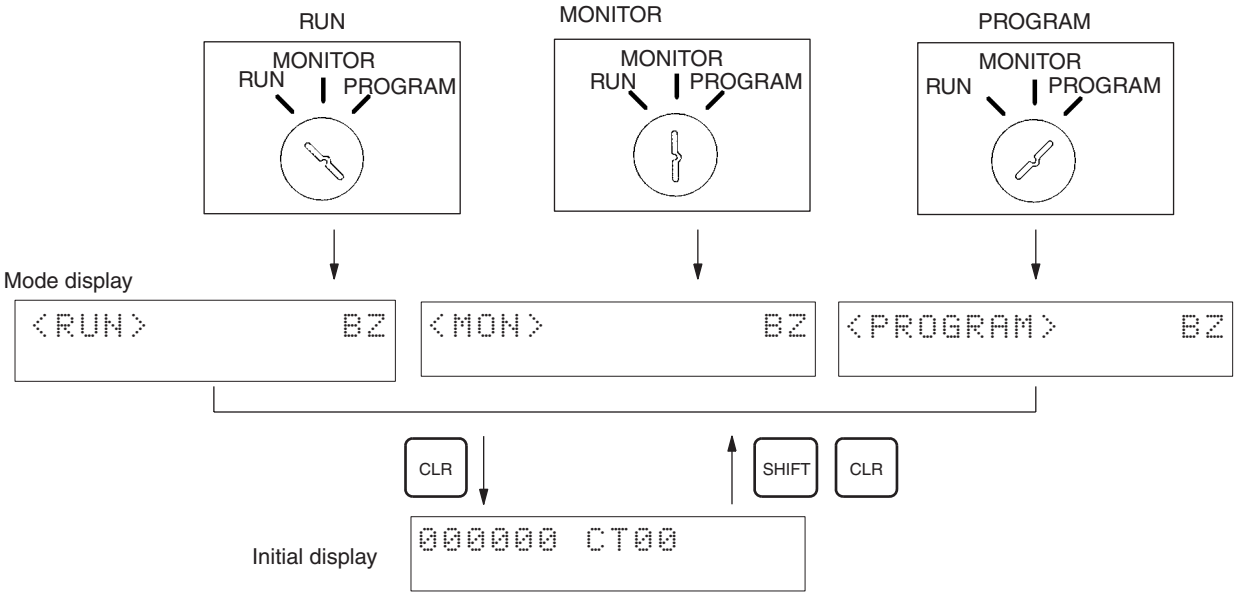


# 3-2 Changing Operating Modes

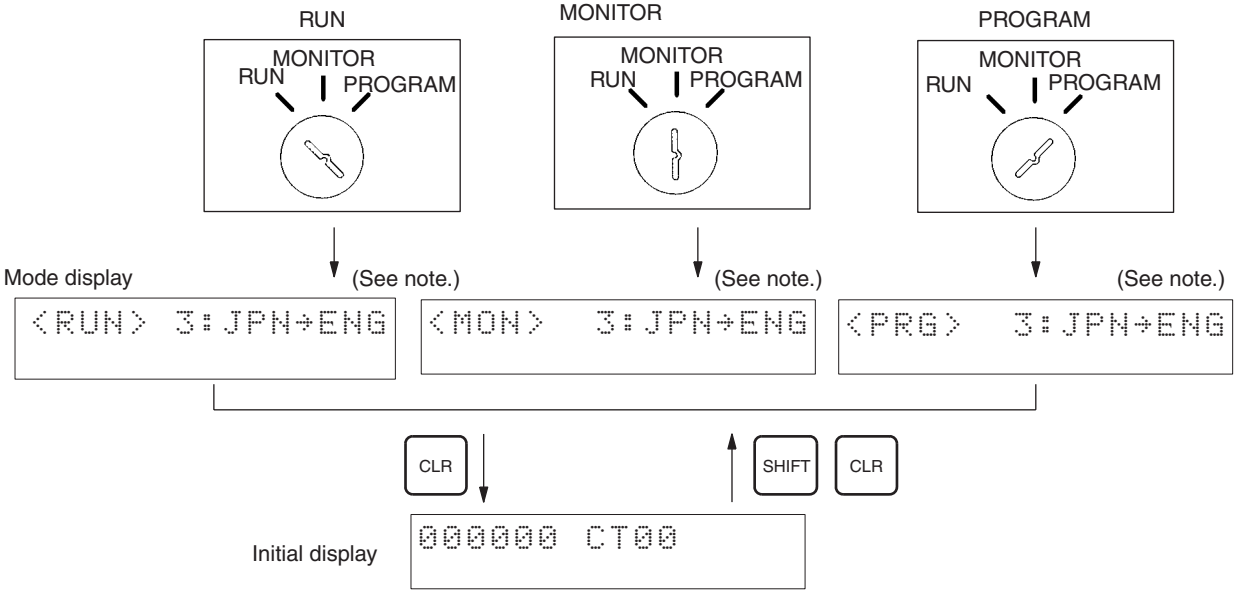
The operating mode of the CPU Unit can be changed using the mode switch on the Programming Console. The mode switch will be enabled after the password has been accepted.

The mode switch can be changed from any screen.

## CS1 CPU Units




## CJ1/CJ1-H/CS1-H CPU Units




**Note** "3: JPN→ENG" indicates that Japanese language messages will be displayed. "3: ENG→JPN" indicates that English language messages will be displayed. Press the 3 Key to change the display language.

⚠ **Caution** Before changing the operating mode of the CPU Unit, make sure that doing so will not affect other equipment.

**Note**

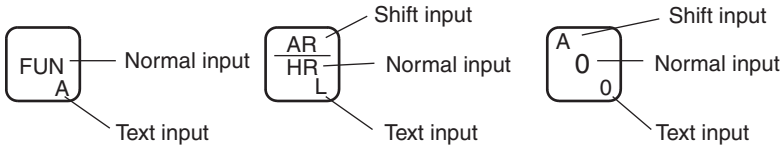
1. The CPU Unit is set by default so that the operating mode on the mode switch of the connected Programming Console is used when power is turned ON. The Programming Console mode will be used as long as the default is set for the “Startup Mode” in the PLC Setup (address +81 set to PRCN).
2. If a Programming Console is not connected when power is turned ON and the Startup Mode in the PLC Setup is set to PRCN, a CS1 CPU Unit will automatically start in PROGRAM mode and a CJ1/CJ1-H/CS1-H CPU Unit will automatically start in RUN mode.
3. The mode switch can be used to change the operating mode from any display without affecting the display. This is particularly useful during monitoring, enabling the user to display monitoring details while changing the operating mode. Hold down the **SHIFT** Key when changing the mode switch.

# 3-3 Key Functions

The Programming Console has three key input modes. They are as follows:

- Normal input mode: Pressing the key alone
- Shift input mode: Pressing the **SHIFT** Key and then the key
- Text input mode: Pressing the **TEXT** Key and then the key

As indicated in the following diagram, keys are marked with input characters for each of the 3 modes: Normal, shift, and text input.



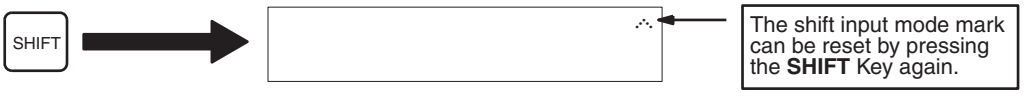

**Normal Input Mode**

The shift input mode is used to enter the text in the center or lower-center of the key.

**Shift Input Mode**

The shift input mode is used to enter the text on the top of the key or on the upper-left corner of the key. Shift mode is also used to input hexadecimal A to F.

The key sequence to switch to shift input mode, and the shift input mode display are as follows:




- If a key is pressed when the shift input mode mark is being displayed, the text at either the top or the upper-left corner of the key will be entered.
- Press the **SHIFT** Key again to return to the normal key input mode.

**Text Input Mode**

The text input mode is used to enter the text (text or number) on the lower-right corner of the key. The text input mode is used to enter file names when Memory Cards are used.

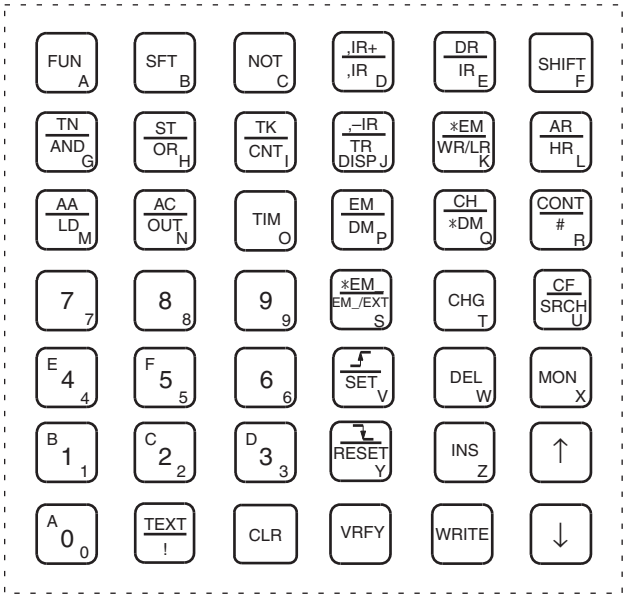
The key sequence to switch to text input mode, and the text input mode display are as follows:



- When the text input mode mark is displayed and a key is pressed, the text or numeral on the lower-right corner of the key can be entered.
- Pressing the **CLR** Key to return to the normal key input mode.

**Keypad**

The key layout of the CS1W-KS001-E Key Sheet is illustrated below.




# 3-4 Clearing Memory Areas

This operation is used to clear program memory and I/O memory and initialize the PLC Setup. The following two methods for clearing memory are possible. The memory areas cleared in each method are as follows:

| Memory clear method | Memory area | | |
|---|---|---|---|
| | Program | I/O memory | PLC Setup |
| All clear | Creates a task and writes the END instruction. (See note 2.) | Clears designated memory areas. (See note 3.) | All PLC Setup settings |
| Memory clear | Clears from the designated program address of the task currently being edited (part of the program.) (See note 1.) | Clears designated memory areas. (See note 3.) | All PLC Setup settings. |

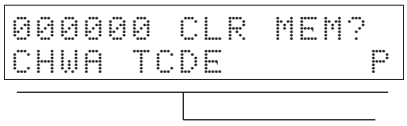
**Note**

1. If program address 000000 is designated, the program of the task currently being edited will all be cleared.
2. The entire program will be cleared. After clearing the programs in all tasks, either interrupt tasks and one cyclic task or just one cyclic task can be created.
3. An address range within specific memory areas cannot be specified. All data within the specified memory areas will be cleared.

When the All Clear or Memory Clear operation is executed, an END(001) instruction will be written to the first address of the user ladder program. NOP(000) instructions will not be written.

### Specifying Memory Areas Not to be Cleared

Letters will be displayed at the bottom of the display for the memory areas that will be cleared. If an area is specified to not be cleared, the corresponding letter will disappear.




Letters at the bottom of the display indicate the memory areas that will be cleared.

| Letter | Areas to be cleared | Key sequence |
|---|---|---|
| C | CIO (See note) | SHIFT, CH/*DM |
| H | HR (Holding bits) | AR/HR |
| W | WR (Work bits) | *EM/WR/LR |
| A | AR (Auxiliary bits) | SHIFT, AR/HR |
| T | T (Completion Flags/PVs) | TIM |
| C | C (Completion Flags/PVs) | TK/CNT |
| D | DM (Data memory) | EM/DM |
| E | EM (Extended data memory) | SHIFT, EM/DM, Bank number |
| P | PLC Setup | SHIFT, CF/SRCH |

**Note** The CIO Area is used for I/O word data: I/O bits, Data Link bits, CS/CJ CPU Bus Unit bits, Inner Board bits, SYSMAC BUS bits, I/O Terminal bits, Special I/O Unit bits, CompoBus/D (DeviceNet) bits, and work bits. (Inner Board bits, SYSMAC BUS bits, and I/O Terminal bits are supported by CS-series PLCs only.)

To clear the memory areas that are specified not to be cleared, repeat the same key sequence. The appropriate letter will be displayed at the bottom of the display.

The EM Area cannot be cleared if it has been converted to file memory. Refer to *Section 7 Memory Card Operations* for information on formatting file memory and restoring the normal EM Area.

## 3-4-1 All Clear

| RUN | MONITOR | PROGRAM |
|---|---|---|
| No | No | OK |

This operation is used to clear (format) memory areas. When “All Clear” is performed from the Programming Console, all programs in all tasks will be cleared. One cyclic task is created, and the user can chose whether or not to also create interrupt tasks.

**Note** The Programming Console cannot be used to create programs with more than one cyclic task. Use the CX-Programmer if more than one cyclic task is required.

### Creating One Cyclic Task without Interrupt Tasks

Use the following procedure to clear memory and create one cyclic task (CT00) and no interrupt tasks.

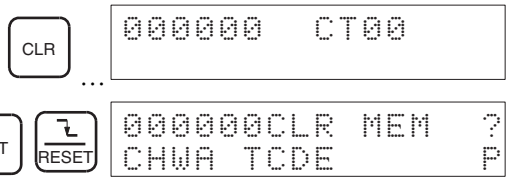
**Note** To create interrupt tasks, always select ***INT 1: YES*** when clearing memory. If ***INT 0: NO*** is selected, you will not be able to create interrupt tasks unless you clear memory again using ***INT 1: YES***.

**Key Sequence**



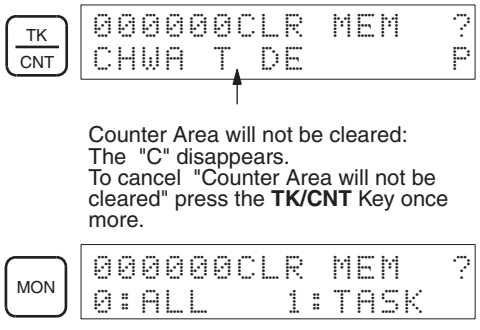

**Operation Example**

***1,2,3...*** 1. Start the All Clear operation from the initial display.

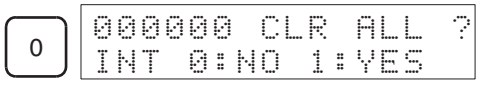



2. Specify the areas that are not to be cleared. Refer to page 38 for further details on specifying the areas.

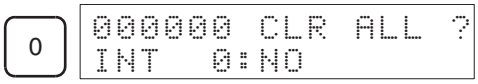
   **Example:** Specifying the Counter Area to no be cleared.



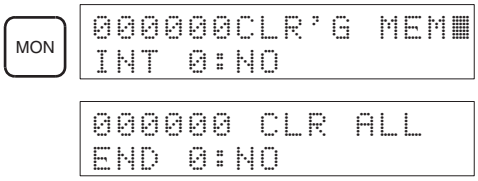

3. Select ***0: ALL*** to clear all memory.



4. Select ***INT 0: NO*** to create one cyclic task only (and no interrupts).




5. Start the All Clear operation.




- Press the **CLR** Key to return to the original display.

## Creating One Cyclic Task and Interrupt Tasks

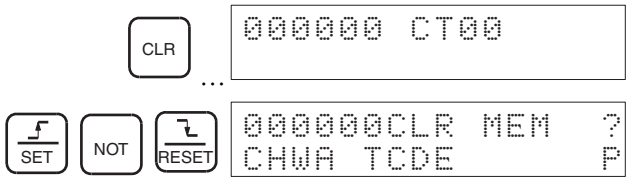
Use the following procedure to clear memory and create one cyclic task (CT00) and interrupt tasks (IT001 to 003 and IT100 to 131).

**Key Sequence**



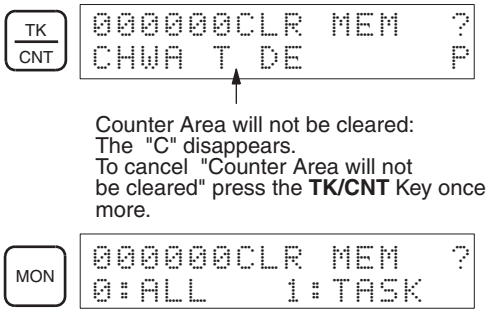

**Operation Example**

***1,2,3...*** 1. Start the All Clear operation from the initial display.

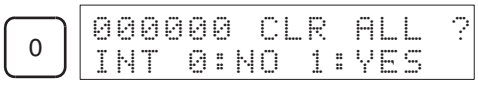



2. Specify the areas that are not to be cleared. Refer to page 38 for further details on specifying areas.

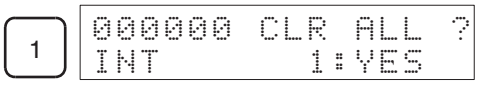
   **Example:** Specifying the Counter Area not to be cleared.



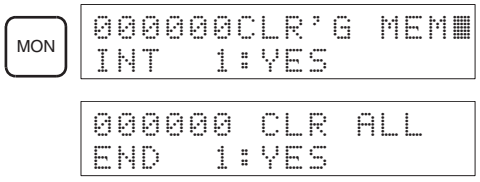

3. Select ***0: ALL*** to clear all memory.




4. Select ***INT 1: YES*** to create interrupt tasks.



5. Start the All Clear operation.




- Press the **CLR** Key to return to the original display.

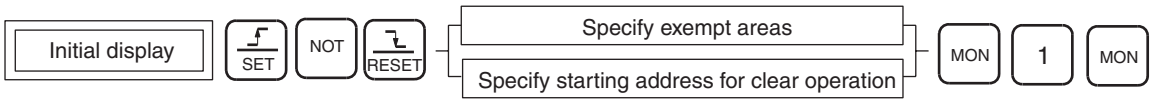
Note When the All Clear operation is performed from the Programming Console, cyclic task No. 0 will be automatically activated at PLC startup.

## 3-4-2 Memory Clear

| RUN | MONITOR | PROGRAM |
|---|---|---|
| No | No | OK |

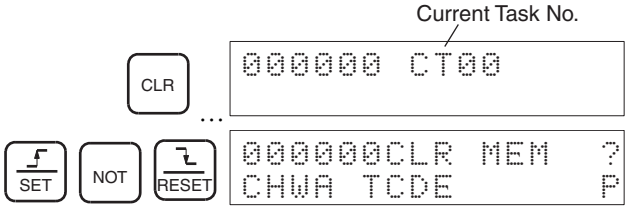
This operation is used to clear the program in the current task starting with a specified program address within the current task. It is also used to clear I/O memory and the PLC Setup.

**Key Sequence**



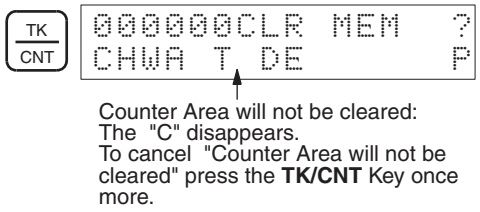

**Operation Example**

***1,2,3...*** 1. Start the Memory Clear operation.



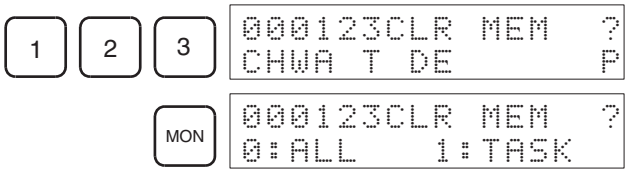

2. Specify the areas that are not to be cleared. Refer to page 38 for further details on specifying areas.

**Example:** Specifying the Counter Area to not be cleared.

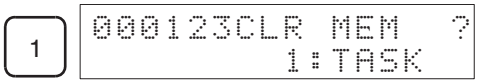



3. Specify the first address to be cleared in the program.

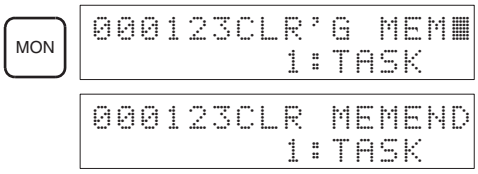
**Example:** Clear the program starting from program address 000123 (including 000123).



4. Select ***1: TASK*** to clear the contents of the task.




5. Execute Memory Clear.




- Press the **CLR** Key to return to the original display.

# 3-5 Buzzer Operation

| RUN | MONITOR | PROGRAM |
|---|---|---|
| OK | OK | OK |

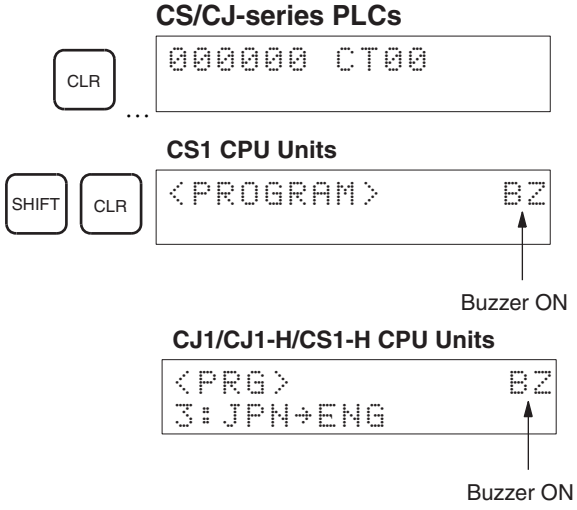
This operation is used to turn ON and OFF the buzzer that sounds when Programming Console keys are pressed. The buzzer ON/OFF operation is performed from the mode display.

### Key Sequence

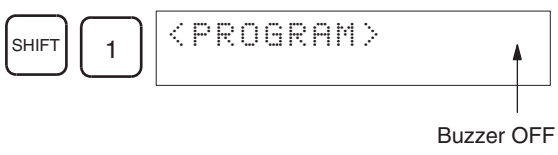



### Operation Example

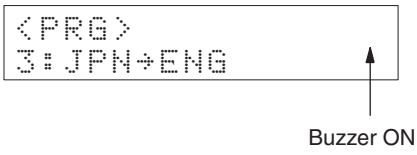
***1,2,3...*** 1. Access the buzzer mode display from the initial display.




2. Turn the buzzer ON or OFF as desired.



**CJ1/CJ1-H/CS1-H CPU Units**




Note a) When an error occurs, the buzzer will sound regardless of whether the buzzer has been turned ON or OFF.

b) The mode display is displayed immediately after the password has been entered or after the mode is changed.

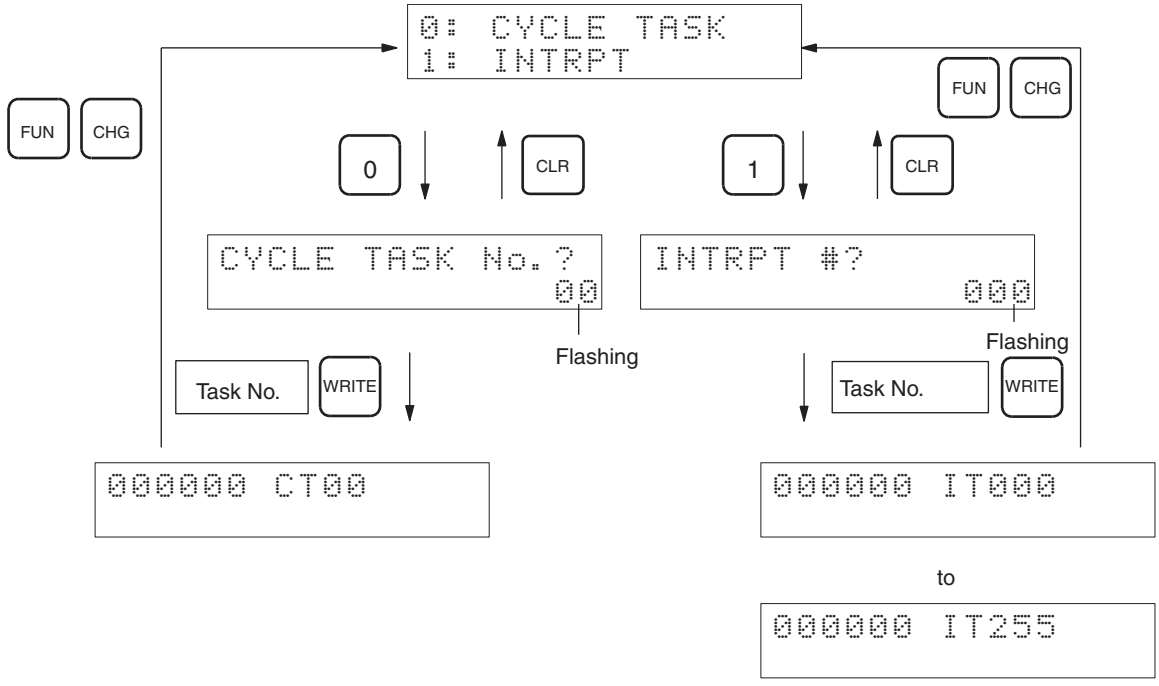
c) For CJ1/CJ1-H/CS1-H CPU Units, the display language can be switched by pressing the **3** Key while "3: JPN→ENG" or "3: ENG→JPN" is being displayed.

# 3-6 Selecting Tasks

| RUN | MONITOR | PROGRAM |
|---|---|---|
| OK | OK | OK |

This operation allows the user to select the task and the type of task (cyclic or interrupt.) From the Programming Console however, only interrupt task numbers 1,2, and 3 (IT100 to 131) can be newly created. Interrupt task numbers 0 and 4 to 99 (IT132 to 255) cannot be created.

## Outline of Key Sequence for Selecting Tasks



## Task Types and Task Numbers

| Task | | Task No. | Details | Created using the Programming Console |
|---|---|---|---|---|
| Cyclic tasks | | 0 to 31 (See note 1.) | Each enabled task is executed in order of task number from the lower number. Each enabled task is executed once each cycle from the first word to the END(001) instruction. | Task No. 0 only, cannot be created. (1 to 31 can be created.) |
| Interrupt tasks (See note 2.) | Power OFF interrupt | 1 | Executed when power to the CPU Unit is interrupted. | Possible |
| | Scheduled interrupt | 2, 3 | Executed at regular intervals according to the CPU Unit's internal timer. | Possible |
| | I/O interrupt | 100 to 131 | Executed when the Interrupt Input Unit input turns ON. | Possible |
| | External interrupt | 0 to 255 | Executed when there is a request from a Special I/O Unit, a CS1 CPU Bus Unit, or the Inner Board user program. | Task No. 1 to 3 and 100 to 131 possible. (0.4 to 99 and 132 to 255 not possible.) |

**Note**

1. The Programming Console cannot be used to create programs with more than one cyclic task. Use the CX-Programmer if more than one cyclic task is required.
2. If ***INT 0: NO*** is set when memory is cleared, only a cyclic task (CT00) can be created, and interrupt tasks cannot be created. Refer to page 39 for details.

## Cyclic Task

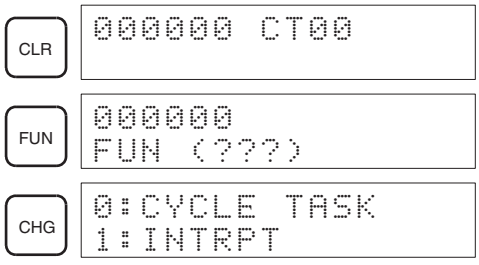
### Key Sequence



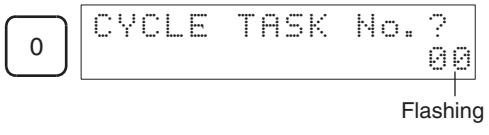

### Operation Example

***1,2,3...***

1. Access the task selection display from the initial display.



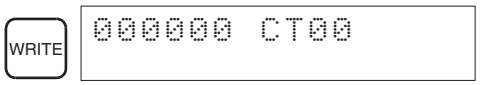

2. Select ***0: CYCLE TASK*** from the above display.



3. Input the task number.




4. Confirm the task number and then press the **WRITE** Key to set the task number and return to the initial display.



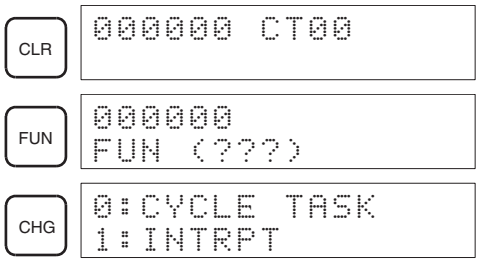

## Interrupt Tasks

### Key Sequence

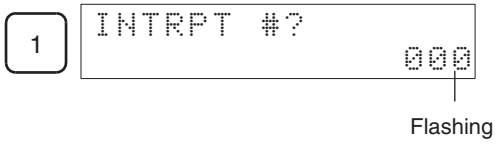



### Operation Example

***1,2,3...*** 1. Access the task selection display from the initial display.



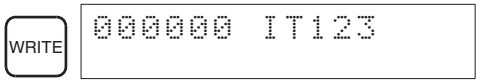

2. Select ***1: INTRPT*** from the above display.




3. Enter the number of the interrupt task.




4. Confirm the task number and then press the **WRITE** Key to set the number.



# 3-7 I/O Table Operations

## 3-7-1 I/O Table Create

| RUN | MONITOR | PROGRAM |
|---|---|---|
| No | No | OK |

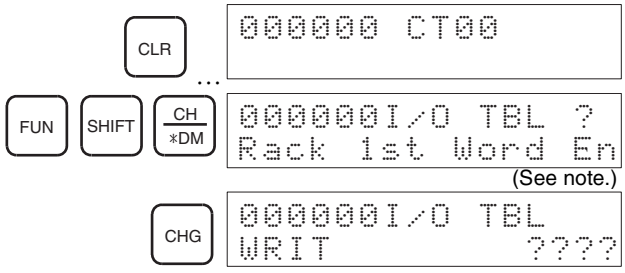
This operation is used to register the I/O tables in the CPU Unit. The I/O tables contain information on all Units mounted to the CPU and Expansion Racks. The information includes the unit types and positions.

**Key Sequence**




**Operation Example**

***1,2,3...*** 1. From the initial display, access the I/O Table Create display and start the I/O table creation process.




Note If the first word for a Rack has been set in advance from the CX-Programmer for a CS1-H or CJ1-H CPU Unit, “Rack 1st Word En” will be displayed to show that the first word has been set.

2. Enter the password (9713) and then press the **WRITE** Key.




Note “9713” is the password set by the system for I/O Table Create and Memory Card Format operations. This password cannot be changed.



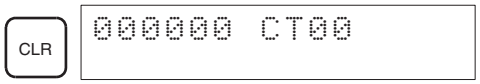

3. Select to either clear or keep CS1 CPU Bus Unit system information.




0 : Clears all CS/CJ CPU Bus Unit system information.

1 : Keeps all CS/CJ CPU Bus Unit system information.

4. Return to the initial display.

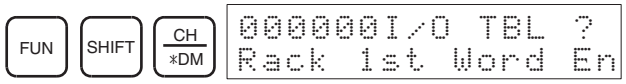



**Confirming Rack First Word Settings**

With a CS1-H or CJ1-H CPU Unit, the Programming Console can be used to check whether or not the first word has been set on a Rack. Use the following procedure.

***1,2,3...***

1. Press the **FUN**, **SHIFT**, and **CH** Keys to start the I/O table creation operation. If the first word for a Rack has been set, a message saying so will appear on the second line of the display.




   If nothing is displayed, then the first word has not been set.

2. Press the **CHG** Key, enter the password (9713), and then press the **WRITE** Key to continue creating the I/O tables, or press the **CLR** Key to cancel the operation and return to the initial display.

**Note Detailed Information on I/O Table Creation Errors**

The CS1-H and CJ1-H CPU Units will provide detailed information on errors that occur when creating I/O tables. For example, with a CS1-H CPU Unit, the contents of A261 will provide information on the Unit causing the error whenever one occurs when creating the I/O tables from the Programming Console.

For details, refer to the troubleshooting section in the *CS Series Programmable Controllers Operation Manual* or *CJ Series Programmable Controllers Operation Manual*.

**I/O Table Error Details**

| Name | Address | | Contents | When changing to RUN mode | At startup | Setting timing |
|---|---|---|---|---|---|---|
| | Word | Bit | | | | |
| CPU Bus Unit Setup Area Initialization Error Flag | A261 | 00 | ON: Error in CPU Bus Unit Setup<br>Turns OFF when I/O tables are generated normally. | Held | Cleared | When I/O tables are created |
| I/O Overflow Flag | | 02 | ON: Overflow in maximum number of I/O points.<br>Turns OFF when I/O tables are generated normally. | | | |
| Duplication Error Flag | | 03 | ON: The same unit number was used more than once.<br>Turns OFF when I/O tables are generated normally. | | | |
| I/O Bus Error Flag | | 04 | ON: I/O bus error<br>Turns OFF when I/O tables are generated normally. | | | |
| SYSMAC BUS Recognition Error Flag | | 06 | ON: SYSMAC BUS detection ended in an error.<br>Turns OFF when I/O tables are generated normally. | | | |
| Special I/O Unit Error Flag | | 07 | ON: Error in a Special I/O Unit<br>Turns OFF when I/O tables are generated normally. | | | |
| I/O Unconfirmed Error Flag | | 09 | ON: I/O detection has not been completed.<br>Turns OFF when I/O tables are generated normally. | | | |

## 3-7-2 I/O Table Verify

| RUN | MONITOR | PROGRAM |
|---|---|---|
| OK | OK | OK |

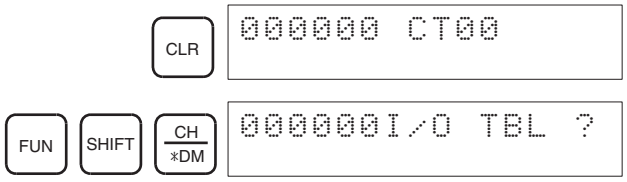
This operation is used to compare the I/O table registered in the PLC (registered I/O tables), with the actual I/O Units mounted to the PLC Racks.

**Key Sequence**

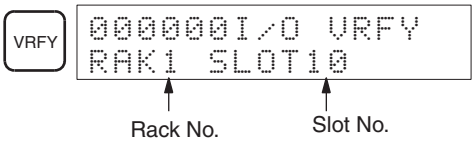


**Operation Example**

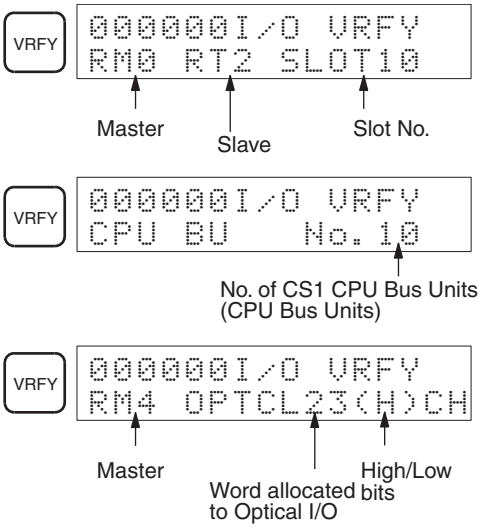
***1,2,3...*** 1. From the initial display, execute I/O Table Verify.



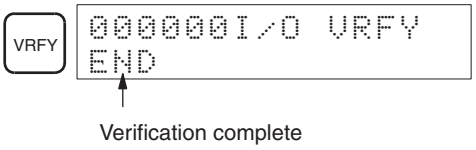

2. Confirm the location of any verification errors.




3. Any further errors will be displayed in sequence as the **VRFY** Key is pressed.



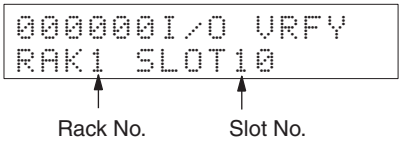

4. When all verification errors have been displayed, the following display will appear.




## Verification Errors

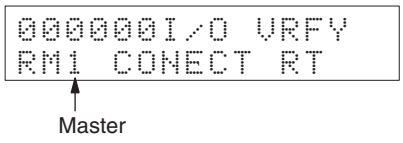
**Verification Error**

The following display indicates that the mounted I/O Units differ from those registered in the I/O table.



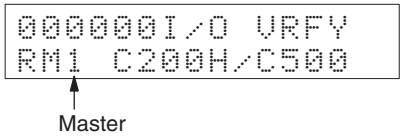
**No. of Remote Terminals Verification Error**

The following display indicates that the number of Slaves (RT) connected to the Master does not match the number registered in the I/O table.



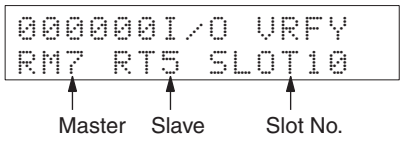

**C200H/C500 Master Verification Error**

The following display indicates that the Master Unit (RM) (C200H/C500) mounted to the PLC does not match the one registered in the I/O table.



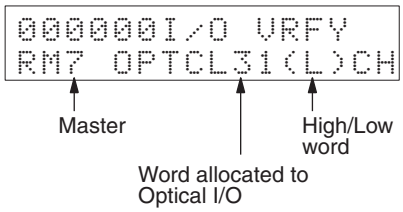

**Master-Slave Slot Verification Error**

The following display indicates that a Slave Unit connected to a Master does not match the Unit registered in the I/O table.



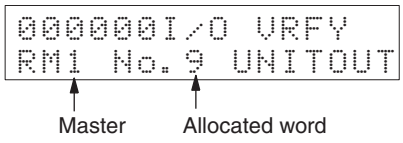

**Slaves Connected to Master Verification Error**

The following display indicates that the words or the high/low word specification allocated to an Optical I/O Units does not match the information registered in the I/O table.



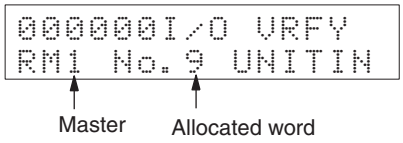

**No. of Output Words Verification Error**

The following display indicates that the number of output words connected to a Master mounted to the PLC does not match the number registered in the I/O table.



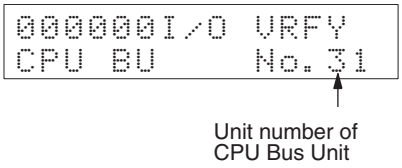

**No. of Input Words Verification Error**

The following display indicates that the number of input words connected to a Master mounted to the PLC does not match the number registered in the I/O table.



**CS1 CPU Bus Unit Verification Error**

The following display indicates that the type of a CS/CJ CPU Bus Unit (CPU Bus Unit) mounted to the PLC does not match the type registered in the I/O table.



# SECTION 4
# Writing and Editing Programs

This section describes how to create and edit programs from the Programming Console.


4-1 Inputting Instructions . . . . . 54
4-1-1 Procedure for Inputting Instructions . . . . . 54
4-1-2 Inputting Basic Instructions . . . . . 55
4-1-3 Inputting Timer/Counter Instructions: TIM/CNT . . . . . 59
4-1-4 Inputting Special Instructions . . . . . 62
4-2 Reading and Searching Programs . . . . . 67
4-2-1 Reading Designated Addresses . . . . . 67
4-2-2 Instruction Search . . . . . 68
4-2-3 Bit Search . . . . . 69
4-3 Editing Programs . . . . . 71
4-3-1 Instruction Change . . . . . 71
4-3-2 Instruction Insert . . . . . 72
4-3-3 Instruction Delete . . . . . 73
4-3-4 Bit Address Change . . . . . 74
4-3-5 N.O./N.C. Change . . . . . 74
4-3-6 Instruction Variation Change . . . . . 75
4-3-7 Timer/Counter SV Changes . . . . . 76

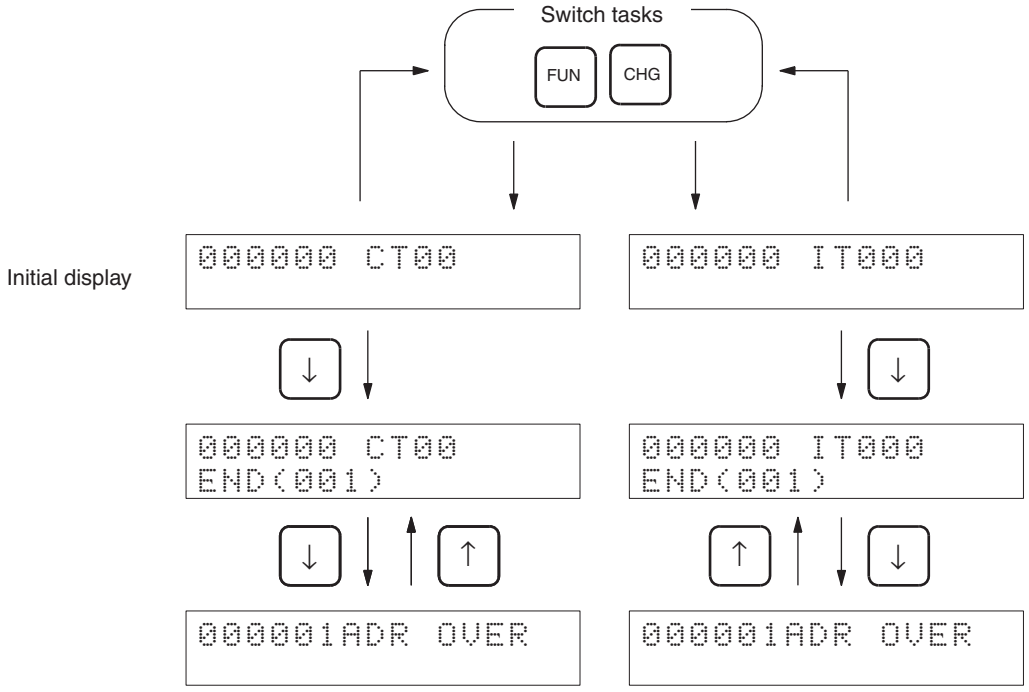
# 4-1 Inputting Instructions

## 4-1-1 Procedure for Inputting Instructions

| RUN | MONITOR | PROGRAM |
|---|---|---|
| No | No | OK |

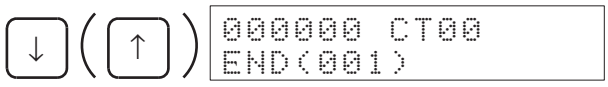
Programs can be input in PROGRAM mode only.

***1,2,3...*** 1. Press the **FUN** and **CHG** Keys to switch tasks and bring up the initial display.

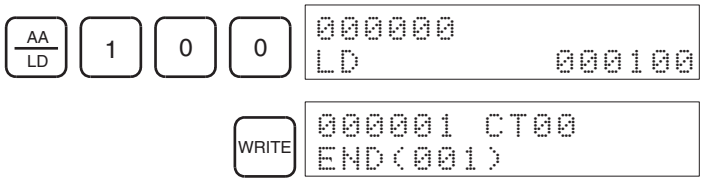



- The END(001) instruction will be displayed at program address 000000 the first time instructions are written into a task. When the END(001) instruction is displayed, all instructions input will be inserted before the END(001) instruction.
- If the address after the END(001) instruction is displayed, ***ADR OVER*** will appear. Instructions cannot be executed, even if they are input while ***ADR OVER*** is displayed, so press the **Up** Key to return to the END(001) instruction display.

2. Input instructions while the END(001) instruction is displayed.



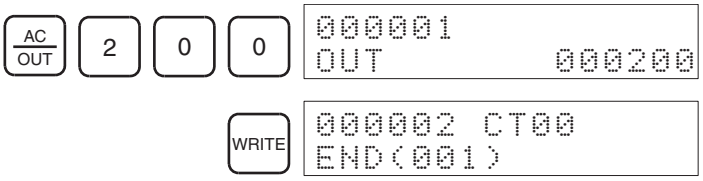

a) Input LD 000100.



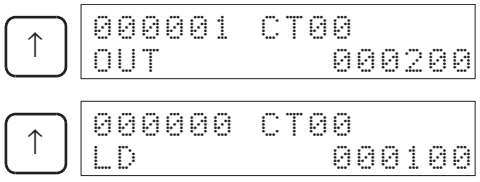
b) Press the **WRITE** Key.

***Write*** will be displayed temporarily. When inputting is complete, the address will move forward by one and the END(001) instruction will automatically be sent to the next address.

3. Input OUT 000200 while the END (001) instruction is displayed.




4. Check instructions that have been input correctly.




An instruction can be input when any program address is displayed. The instruction will overwrite the data at the current program address.

5. Return to the initial display.




## 4-1-2 Inputting Basic Instructions

| RUN | MONITOR | PROGRAM |
|---|---|---|
| No | No | OK |

Basic instructions can be input only when the CPU Unit is in PROGRAM mode.

**Key Sequence**

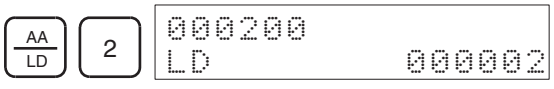



**Operation Example**

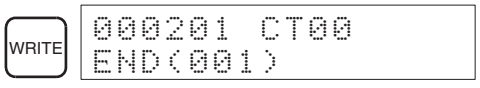
Use the following procedure to input the LOAD (LD) instruction.

***1,2,3...***

1. Designate the mnemonic and the operand.




2. Press the **WRITE** Key.




If a mistake is made inputting, press the **Up** Key to bring up the program address, and re-input the instruction. The data at the address will be overwritten.

### Inputting Basic Instructions

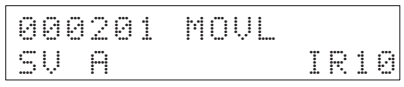
| Mnemonic | Key sequence | Mnemonic | Key sequence |
|---|---|---|---|
| LD | AA / LD | OR LD | ST / OR, AA / LD |
| LD NOT | AA / LD, NOT | AND LD | TN / AND, AA / LD |
| OR | ST / OR | OUT | AC / OUT |
| OR NOT | ST / OR, NOT | OUT NOT | AC / OUT, NOT |
| AND | TN / AND | SET | FUN, SET |
| AND NOT | TN / AND, NOT | RSET | FUN, RESET |

### Designating Bit Addresses

| Bit type | Key sequence | Display example |
|---|---|---|
| CIO bits | Number | Bit 15 of CIO 0000<br>000000<br>LD 000015 |
| Work bits (WR) | *EM / WR/LR, Number | Bit 15 of W000<br>000000<br>LD W00015 |
| Holding bits (HR) | AR / HR, Number | Bit 15 of H000<br>000000<br>LD H00015 |
| Auxiliary bits (AR) | SHIFT, AR / HR, Number | Bit 15 of A000<br>000000<br>LD A00015 |
| Timer bits | TIM, Number | T0015<br>000000<br>LD T0015 |
| Counter bits | TK / CNT, Number | C0015<br>000000<br>LD C0015 |

<table>
<tr><th>Bit type</th><th>Key sequence</th><th>Display example</th></tr>
<tr><td rowspan="2">Clock pulse/Condition Flag</td><td>SHIFT CF/SRCH</td><td>Always ON Flag<br></td></tr>
<tr><td>When clock pulses or Condition Flags are designated, the bits listed at the right can be designated, by using the Up and Down Keys.<br><br><table><tr><th>Display</th><th>Bit</th></tr><tr><td>ON</td><td>Always ON Flag</td></tr><tr><td>OFF</td><td>Always OFF Flag</td></tr><tr><td>AER</td><td>Access Error Flag</td></tr><tr><td>0.02s</td><td>0.02-s clock pulse</td></tr><tr><td>0.1s</td><td>0.1-s clock pulse</td></tr><tr><td>0.2s</td><td>0.2-s clock pulse</td></tr><tr><td>1s</td><td>1-s clock pulse</td></tr><tr><td>1min</td><td>1-min clock pulse</td></tr><tr><td>ER</td><td>Error Flag</td></tr><tr><td>CY</td><td>Carry Flag</td></tr><tr><td>></td><td>Greater Than Flag</td></tr><tr><td>=</td><td>Equals Flag</td></tr><tr><td><</td><td>Less Than Flag</td></tr><tr><td>N</td><td>Negative Flag</td></tr><tr><td>OF</td><td>Overflow Flag</td></tr><tr><td>UF</td><td>Underflow Flag</td></tr><tr><td>>=</td><td>Greater Than or Equals Flag</td></tr><tr><td><></td><td>Not Equal Flag</td></tr><tr><td><=</td><td>Less Than or Equals Flag</td></tr></table></td></tr>
<tr><td>Task Flag</td><td>SHIFT TK/CNT Number</td><td>TK0000<br></td></tr>
</table>

### Designating Index Register Addresses

<table>
<tr><th>Method</th><th>Key sequence</th><th>Display example</th></tr>
<tr><td>Direct addressing</td><td>DR/IR Number<br>(Absolute address)<br><br>Note Only some instructions such as MOVL are possible.</td><td>IR10<br></td></tr>
<tr><td>Indirect addressing</td><td>,IR+/,IR Number<br>(Relative address)</td><td>IR10<br></td></tr>
</table>

| Method | Key sequence | Display example |
|---|---|---|
| Constant offset addressing | ,IR+ / ,IR → Number (Relative address)<br>SET (+) → Number<br>RESET (Constant offset) (–) | +1234, IR10<br>000201<br>LD +1234, IR10<br>Positive offset<br>–1234, IR10<br>000201<br>LD –1234, IR10<br>Negative offset |
| DR offset addressing | ,IR+ / ,IR → Number (Relative address)<br>SHIFT, DR / IR → Number DR address | DR 01, IR10<br>000201<br>LD DR01, IR10 |
| Auto-increment | Auto-increment+<br>SHIFT, ,IR+ / ,IR → Number (Relative address) | IR10+<br>000201<br>LD , IR10+<br>Auto-increment (+1) |
| | Auto-increment+2<br>SHIFT, ,IR+ / ,IR, SHIFT, ,–IR / TR, Number<br>**Note** The display will toggle between auto-increment +1 and auto-increment +2. | IR10++<br>000201<br>LD , IR10++<br>Auto-increment (+2) |
| Auto-decrement | Auto-decrement<br>SHIFT, ,–IR / TR → Number (Relative address) | IR10–<br>000201<br>LD , –IR10<br>Auto-decrement (–1) |
| | Auto-decrement+2<br>SHIFT, ,–IR / TR, SHIFT, ,–IR / TR, Number<br>**Note** The display will toggle between auto-decrement –1 and auto-decrement –2. | IR10– –<br>000201<br>LD , – –IR10<br>Auto-decrement (–2) |

**Designating Instruction Variations**

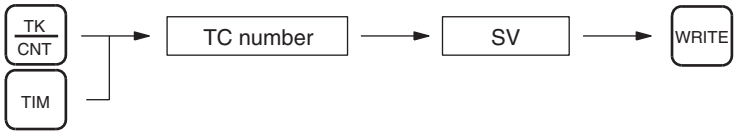
| Variation | Key sequence | Display example |
|---|---|---|
| Differentiate up | SHIFT, SET<br>**Note** The same key sequence can be used to reset. | 000123 CT00<br>@LD 000000<br>Symbol: Differentiate up |
| Differentiate down | SHIFT, RESET<br>**Note** The same key sequence can be used to reset. | 000123 CT00<br>%LD 000000<br>Symbol: Differentiate down |
| Immediate refresh | TEXT (!)<br>**Note** The same key sequence can be used to reset. | 000123 CT00<br>!LD 000000<br>Symbol: Immediate refresh<br>000123 CT00<br>!@LD 000000<br>Symbols: Immediate refresh/Differentiate up<br>000123 CT00<br>!%LD 000000<br>Symbols: Immediate refresh/Differentiate down |

## 4-1-3 Inputting Timer/Counter Instructions: TIM/CNT

| RUN | MONITOR | PROGRAM |
|---|---|---|
| No | No | OK |

Timer/counter instructions can be input when the CPU Unit is in PROGRAM mode.

**Key Sequence**

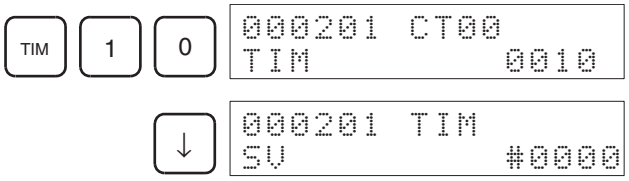



**Operation Example**

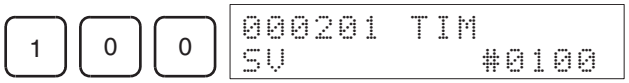
The following example describes how to input TIM 0010 with a SV of 10 s.

***1,2,3...***

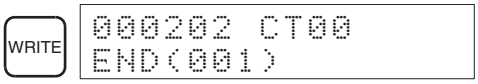
1. Press the **TIM** Key and input the timer number.




2. Input the set value. In the following example, the set value is #0100 (10 s).



3. Press the **WRITE** Key.




If a mistake is made inputting the set value, press the **Up** Key until the set value is displayed. Then press the **CONT/#** Key and enter the correct value.

### Inputting Timer/Counter Instructions

| Instruction | Key sequence |
|---|---|
| TIM | TIM |
| CNT | TK/CNT |

### Designating Timer/Counter Set Values (1)

#### Designating Constants

| Type | Key sequence | Display example |
|---|---|---|
| Constants | CONT/# Number | #0123<br>000201 TIM<br>SV #0123 |

#### Designating Words (16-bit data)

| Type | Key sequence | Display example |
|---|---|---|
| CIO words | SHIFT CH/*DM Number | CIO 0010<br>000000 TIM<br>SV 0010 |
| Work words (WR) | *EM/WR/LR Number | W010<br>000201 TIM<br>SV W010 |
| Holding words (HR) | AR/HR Number | H010<br>000201 TIM<br>SV H010 |
| Auxiliary words (AR) | SHIFT AR/HR Number | A010<br>000201 TIM<br>SV A010 |
| Data memory words (DM) | EM/DM Number | D00010<br>000201 TIM<br>SV D00010 |
| Extended DM (EM): With bank | *EM_/EM_/EXT Number<br>(Bank number, word address) | E0_00010<br>000201 TIM<br>SV E0_00010 |

| Type | Key sequence | Display example |
|---|---|---|
| Extended DM (EM): Current bank | SHIFT, EM/DM, Number (Word address) | E00010<br>000201 TIM<br>SV E00010 |
| Timer PVs | TIM, Number | T0010<br>000201 TIM<br>SV T0100 |
| Counter PVs | TK/CNT, Number | C0010<br>000201 TIM<br>SV C0010 |

**Designating Timer/Counter Set Values (2)**

**Designating Indirect DM/EM Addresses**

| Mode | Key sequence | Display example |
|---|---|---|
| Data Registers (DR) | SHIFT, DR/IR, Number | DR10<br>000201 TIM<br>SV DR10 |

**Indirect DM Addresses**

| Mode | Key sequence | Display example |
|---|---|---|
| BCD | CH/*DM, Number | *D00010<br>000202 TIM<br>SV *D00010 |
| Binary | CH/*DM, CH/*DM, Number | @D00010<br>000202 TIM<br>SV @D00010 |

**Indirect EM Addresses**

| Mode | Key sequence | Display example |
|---|---|---|
| BCD With bank | SHIFT, *EM_/EM_/EXT, Number | *E0_00010<br>000202 TIM<br>SV *E0_00010 |
| BCD Current bank | SHIFT, *EM/WR/LR, Number | *E00010<br>000202 TIM<br>SV *E00010 |
| Binary With bank | SHIFT, *EM_/EM_/EXT, SHIFT, *EM_/EM_/EXT, Number | @E0_00010<br>000202 TIM<br>SV @E0_00010 |
| Binary Current bank | SHIFT, *EM/WR/LR, SHIFT, *EM/WR/LR, Number | E00010<br>000202 TIM<br>SV @E00010 |

- In BCD mode, indirect DM/EM addresses can be specified from D00000 to D09999 and E00000 to E09999. Use binary mode to specify D10000 to D32767 and E10000 to E32767.
- If 8000 to FFFF Hex are specified as indirect DM addresses in binary mode, E00000 to E32767 in bank 0 of the EM Area will be designated.
- If 8000 to FFFF Hex are specified as indirect EM addresses in binary mode, E00000 to E32767 in the bank following the one specified will be designated.

## 4-1-4 Inputting Special Instructions

| RUN | MONITOR | PROGRAM |
|---|---|---|
| No | No | OK |

Special instructions can be input when the CPU Unit is in PROGRAM mode.

### Inputting Standard Special Instructions

**Key Sequence**

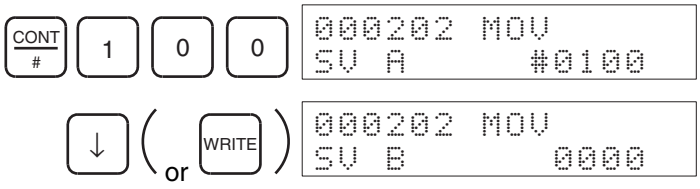



**Operation Example** The following example describes how to input the MOV(021) instruction.

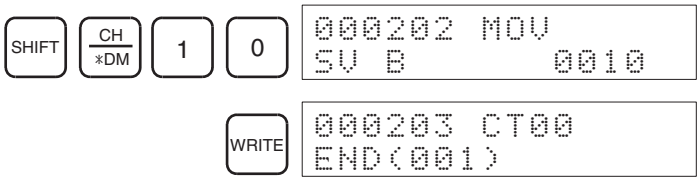
***1,2,3...*** 1. Press the **FUN** Key, enter the function code, and then press the **Down** Key. All 3 digits of the function code must be input. In this example, the function code for MOV is 021.




2. Input the first operand (SV A), and press the **Down** Key. In this example, constant #0100 (hexadecimal) is input.




3. Input the next operand (SV B), and press the **WRITE** Key. In this example, CIO 0010 is input.




- The function codes for some of the instructions, such as KEEP and STEP (which have operands), will not be displayed.

- For further information of the following instruction variations, refer to page 102.

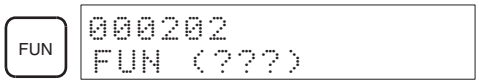
## Inputting Symbol Comparison Instructions

### Key Sequence




**Operation Example** The following example describes how to input the <= instruction.

***1,2,3...*** 1. Press the **FUN** Key and enter the function code. The function code for <= is 318.



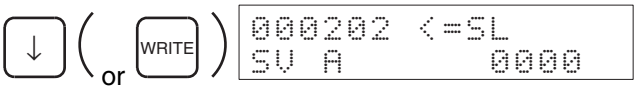




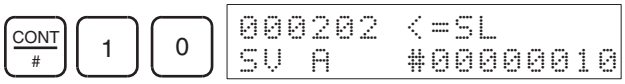
2. Specify the logic type. In this example, OR.

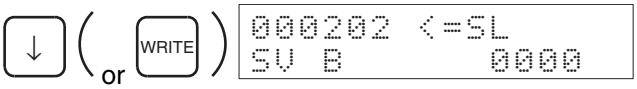



3. Press the **Down** Key.

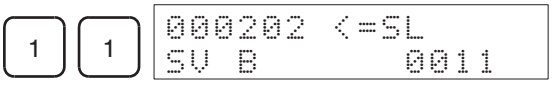



4. Input the first operand (SV A), and press the **Down** Key. In this example, constant #00000010.




5. Input the next operand (SV B), and press the **WRITE** Key. In this example, CIO 0011 is used.






### Designating Word (16-bit) Addresses

| Type | Key sequence | Display example |
|---|---|---|
| CIO words | (SHIFT) (CH / *DM) Number | CIO0010<br>000000 MOV<br>SV A 0010 |
| Work words (WR) | *EM / WR/LR Number | W010<br>000000 MOV<br>SV A W010 |
| Holding words (HR) | AR / HR Number | H010<br>000000 MOV<br>SV A H010 |
| Auxiliary words (AR) | SHIFT AR / HR Number | A010<br>000000 MOV<br>SV A A010 |
| Data memory words (DM) | EM / DM Number | D00010<br>000000 MOV<br>SV A D00010 |
| Extended DM (EM): With bank | *EM_ / EM_/EXT Number<br>(Bank number, word address) | E0_00010<br>000000 MOV<br>SV A E0_00010 |
| Extended DM (EM): Current bank | SHIFT EM / DM Number<br>(Word address) | E00010<br>000000 MOV<br>SV A E00010 |
| Timer PVs | TIM Number | T0010<br>000000 MOV<br>SV A T0100 |
| Counter PVs | TK / CNT Number | C0010<br>000000 MOV<br>SV A C0010 |
| Data Registers (DR) | SHIFT DR / IR Number | DR10<br>000000 MOV<br>SV A DR01 |
| Index Registers (IR) | DR / IR Number | IR1<br>000000 MOVL<br>SV A IR01 |

### Designating Indirect DM/EM Addresses

**Indirect DM Addresses**

| Mode | Key sequence | Display example |
|---|---|---|
| BCD | CH *DM, Number | *D00010<br>000202 MOV<br>SV A *D00010 |
| Binary | CH *DM, CH *DM, Number | @D00010<br>000202 MOV<br>SV A @D00010 |

**Indirect EM Addresses**

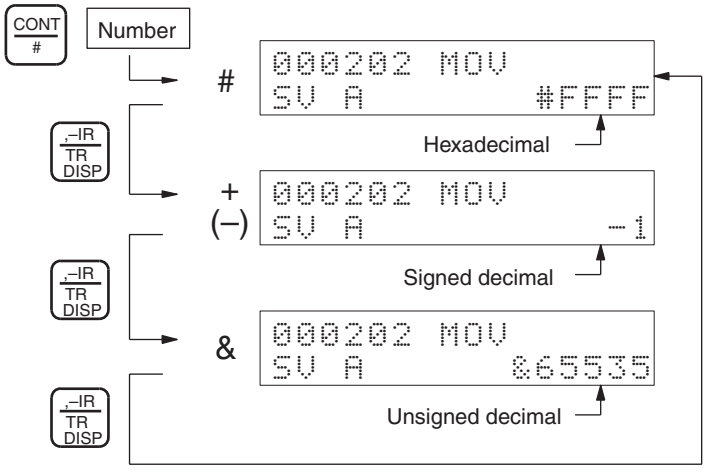
| Mode | Key sequence | Display example |
|---|---|---|
| BCD<br>With bank | SHIFT, *EM_ EM_/EXT, Number | *E0_00010<br>000202 MOV<br>SV A *E0_00010 |
| BCD<br>Current bank | SHIFT, *EM WR/LR, Number | *E00010<br>000202 MOV<br>SV A *E00010 |
| Binary<br>With bank | SHIFT, *EM_ EM_/EXT, SHIFT, *EM_ EM_/EXT, Number | @E0_00010<br>000202 MOV<br>SV A @E0_00010 |
| Binary<br>Current bank | SHIFT, *EM WR/LR, SHIFT, *EM WR/LR, Number | @E00010<br>000202 MOV<br>SV A @E00010 |

- In BCD mode, indirect DM/EM addresses can be specified from D00000 to D09999 and E00000 to E09999. Use binary mode to specify D10000 to D32767 and E10000 to E32767.
- If 8000 to FFFF Hex are specified as indirect DM addresses in binary mode, E00000 to E32767 in bank 0 of the EM Area will be designated.
- If 8000 to FFFF Hex are specified as indirect EM addresses in binary mode, E00000 to E32767 in the bank following the one specified will be designated.

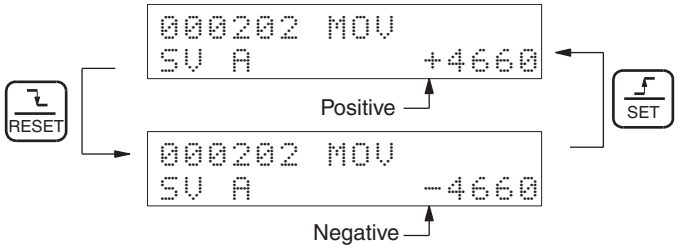
### Designating Constants

| Type | Key sequence | Display example |
|---|---|---|
| Single-word constants | CONT (#) Number<br>4-digit hexadecimal | 000202 MOV<br>SV A #1234<br>Hexadecimal |
| Long constants | CONT (#) Number<br>8-digit hexadecimal | 000202 MOVL<br>SV A #12345678<br>Hexadecimal |

The display format of constants can be changed by pressing the **,–IR/TR** Key.




When the signed decimal display is used, the value can be changed between positive and negative using the **SET/RESET** Keys.




**Note** Changes such as those illustrated above cannot be made for BCD data constants such as Timer/Counter set values, JMP/Subroutine numbers etc.

**Designating Instruction Variations**

| Variation | Key sequence | Display example |
|---|---|---|
| Differentiate up | SHIFT, SET<br>**Note** The same key sequence can be used to reset. | 000123 CT00<br>@MOV(021)<br>Symbol: Differentiate up |
| Differentiate down | SHIFT, RESET<br>**Note** The same key sequence can be used to reset. | 000123 CT00<br>%SET 000000<br>Symbol: Differentiate down |
| Immediate refresh | TEXT !<br>**Note** The same key sequence can be used to reset. | 000123 CT00<br>!MOV(021)<br>Symbol: Immediate refresh<br>000123 CT00<br>!@MOV(021)<br>Symbols: Immediate refresh/Differentiate up<br>000123 CT00<br>!%SET 000000<br>Symbols: Immediate refresh/Differentiate down |

# 4-2 Reading and Searching Programs

## 4-2-1 Reading Designated Addresses

| RUN | MONITOR | PROGRAM |
|---|---|---|
| OK | OK | OK |

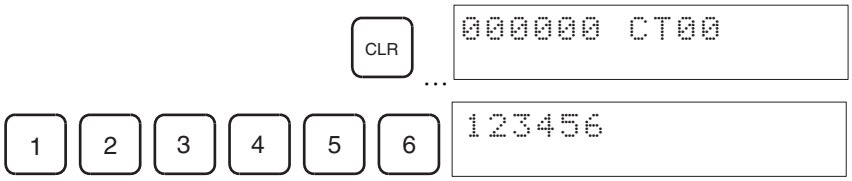
This operation is used to read programs from user memory. In RUN and MONITOR modes, the I/O status of the current bit will be displayed.

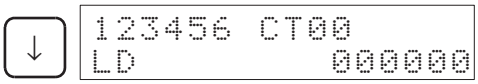
**Key Sequence**




**Operation Example** In the following example, address 123456 is read.

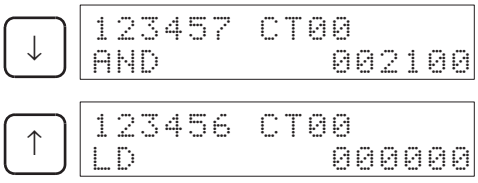
***1,2,3...*** 1. From the initial display, specify the address to be read. In this example, program address 123456 is read.






The program address is displayed when using either the Instruction Search or Operand Search operation.

2. Using the **Up** and **Down** Keys, read the previous and next program addresses.




When the program is finished, the following display will appear.




If you attempt to read program addresses after the END(001) instruction, (program addresses without instructions) ***ADR OVER*** will be displayed. When there is no END(001) instruction, the following display will appear.

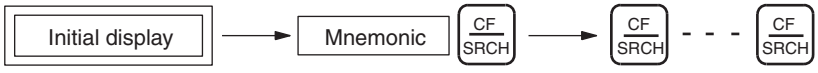



## 4-2-2 Instruction Search

| RUN | MONITOR | PROGRAM |
|---|---|---|
| OK | OK | OK |

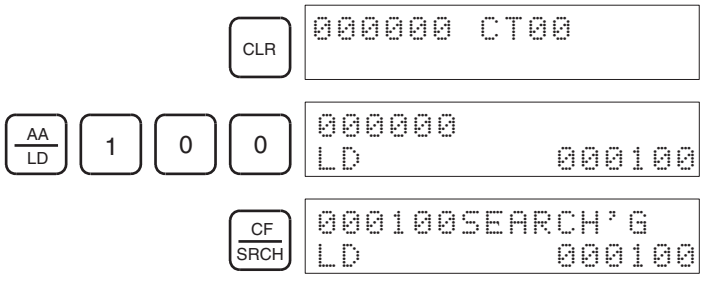
This operation is used to search the program for specified instructions. The program address from which to begin the search can be specified before starting.

**Key Sequence**




**Operation Example** The following example shows a search for the LD (LOAD) instruction.

***1,2,3...*** 1. From the initial display, specify the instruction to be searched for and press the **CF/SRCH** Key to execute. This example shows a search for LD 000100.



The next LD instruction after the beginning program address will be displayed.

```
000200 CT00 FIND
LD          000100
```

2. To continue searching for the same instruction, press the **CF/SRCH** Key. The following display will appear.




```
000300 CT00 FIND
LD          000100
```

3. When the search reaches the END(001) instruction, the following display will appear.




```
000400 CT00 FIND
END(001)
```

4. If the SRCH Key is pressed after the search has finished searching to the END(001) instruction, the following display will appear.




```
000400 CT00 FIND
NO END INST END
```

## 4-2-3 Bit Search

| RUN | MONITOR | PROGRAM |
|---|---|---|
| OK | OK | OK |

This operation searches for the program for a designated bit address. The program address from which to begin the search can be specified before starting.

**Key Sequence**




**Operation Example** In the following example, A20011 is searched for.

***1,2,3...*** 1. From the initial display, specify the bit address to be searched for and press the **CF/SRCH** Key to execute. This example shows a search for A20011.







```
000000 CT00
```

```
000000 CT00
BIT         A20011
```

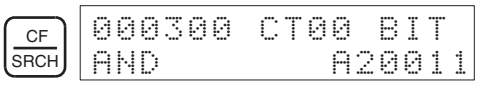



```
000000SEARCH'G
BIT         A20011
```

The following display will appear.

```
000020 CT00 BIT
LD          A20011
```

2. To continue searching for the same address, press the **CF/SRCH** Key. The following display will appear.







3. When the search reaches the END(001) instruction, the following display will appear.




4. If the SRCH Key is pressed after the search has finished searching to the END(001) instruction, the following display will appear.




### Designating Bit Addresses

| Bit type | Key sequence | Display example |
|---|---|---|
| CIO bits | SHIFT, CONT/#, Number | Bit 15 of CIO 0000<br>000000<br>BIT 000015 |
| Work bits (WR) | SHIFT, CONT/#, *EM/WR/LR, Number | Bit 15 of W000<br>000000<br>BIT W00015 |
| Holding bits (HR) | SHIFT, CONT/#, AR/HR, Number | Bit 15 of H000<br>000000<br>BIT H00015 |
| Auxiliary bits (AR) | SHIFT, CONT/#, SHIFT, AR/HR, Number | Bit 15 of A000<br>000000<br>BIT A00015 |
| Timer bits | SHIFT, CONT/#, TIM, Number | T00015<br>000000<br>BIT T0015 |
| Counter bits | SHIFT, CONT/#, TK/CNT, Number | C0015<br>000000<br>BIT C0015 |

<table>
<tr><th>Bit type</th><th>Key sequence</th><th>Display example</th></tr>
<tr><td rowspan="2">Clock pulse/ Condition Flag</td><td>SHIFT CF/SRCH</td><td>Always ON Flag<br>000000<br>BIT ON</td></tr>
<tr><td>When clock pulses or Condition Flags are designated, the bits listed at the left can be designated, by using the Up and Down Keys.<br><table><tr><th>Display</th><th>Bit</th></tr><tr><td>ON</td><td>Always ON Flag</td></tr><tr><td>OFF</td><td>Always OFF Flag</td></tr><tr><td>AER</td><td>Access Error Flag</td></tr><tr><td>0.02s</td><td>0.02-s clock pulse</td></tr><tr><td>0.1s</td><td>0.1-s clock pulse</td></tr><tr><td>0.2s</td><td>0.2-s clock pulse</td></tr><tr><td>1s</td><td>1-s clock pulse</td></tr><tr><td>1min</td><td>1-min clock pulse</td></tr><tr><td>ER</td><td>Error Flag</td></tr><tr><td>CY</td><td>Carry Flag</td></tr><tr><td>></td><td>Greater Than Flag</td></tr><tr><td>=</td><td>Equals Flag</td></tr><tr><td><</td><td>Less Than Flag</td></tr><tr><td>N</td><td>Negative Flag</td></tr><tr><td>OF</td><td>Overflow Flag</td></tr><tr><td>UF</td><td>Underflow Flag</td></tr><tr><td>>=</td><td>Greater Than or Equals Flag</td></tr><tr><td><></td><td>Not Equal Flag</td></tr><tr><td><=</td><td>Less Than or Equals Flag</td></tr></table></td></tr>
<tr><td>Task Flag</td><td>SHIFT TK/CNT Number</td><td>TK0000<br>000000<br>BIT TK0000</td></tr>
</table>

# 4-3 Editing Programs

## 4-3-1 Instruction Change

| RUN | MONITOR | PROGRAM |
|---|---|---|
| No | No | OK |

This operation is used to overwrite instructions in user programs.

**Key Sequence**

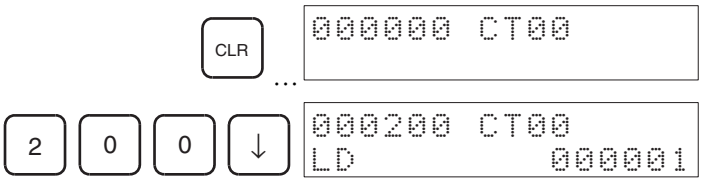
Program address display → Input mnemonic → WRITE

**Operation Example**

In the following instruction block, the instruction for address 000200 will be changed to LD NOT 000002.

| Address | Instruction | Operand | |
|---|---|---|---|
| : | : | : | |
| 000200 | LD | 000001 | ← Changed to LD NOT 000002. |
| 000201 | AND | 000200 | |
| : | : | : | |

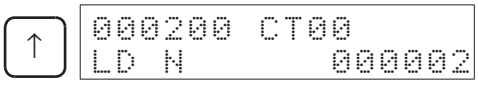
***1,2,3...*** 1. From the initial display, access the address where the instruction is to be changed.




2. Input the instruction and press the **WRITE** Key.




3. Press the **Up** Key to check the instruction.




## 4-3-2 Instruction Insert

| RUN | MONITOR | PROGRAM |
|---|---|---|
| No | No | OK |

This operation is used to insert instructions into the user program.

**Key Sequence**



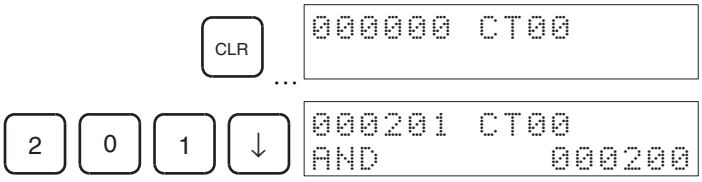

**Operation Example**

In the following instruction block, AND 000100 will be inserted at address 000201.

| | | |
|---|---|---|
| : | : | : |
| 000200 | LD | 000001 |
| 000201 | AND | 000200 |
| : | : | : |

← AND 000100 will be inserted.

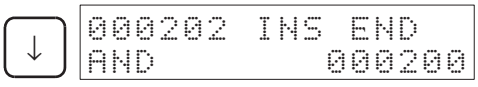
***1,2,3...*** 1. From the initial display, access the address where the instruction is to be inserted.




2. Specify the instruction to be inserted, and press the **WRITE** Key.

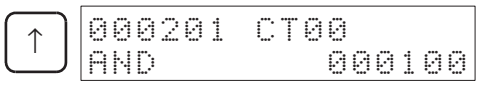



3. Press the **Down** Key to insert the designated instruction. The next program address will be displayed.



If the user attempts to insert an instruction at an address where no instruction exists, ***ADR OVER*** will be displayed, and the instruction will not be inserted.

4. Press the **Up** Key to check the instruction.

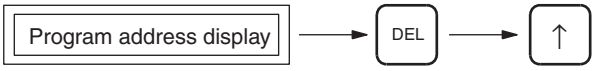



## 4-3-3 Instruction Delete

| RUN | MONITOR | PROGRAM |
|---|---|---|
| No | No | OK |

This operation is used to delete instructions from the user program.

**Key Sequence**



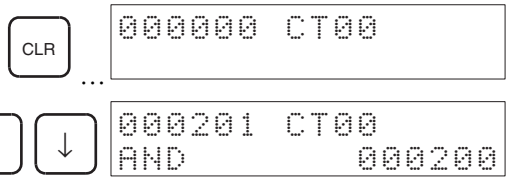

**Operation Example**

In the following instruction block, AND 000200 will be deleted from address 00201.

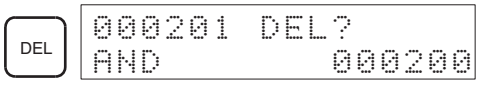
| Address | Instruction | Operand | |
|---|---|---|---|
| : | : | : | |
| 000200 | LD | 000001 | |
| 000201 | AND | 000200 | → Delete |
| 000202 | OUT | 000100 | |
| : | : | : | |

***1,2,3...***

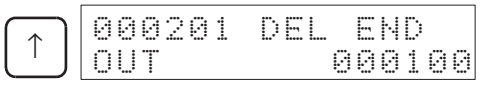
1. From the initial display, access the address of the instruction to be deleted.




2. Delete the instruction.




3. Press the **Up** Key to delete the instruction currently displayed. The next program address will move forward.




- If the user attempts to delete from an address where there is no instruction or when END(001) instruction is written, ***ADR OVER*** will be displayed.
- When multiple-operand instructions are deleted, the set values (operands) will also be deleted.

## 4-3-4 Bit Address Change

| RUN | MONITOR | PROGRAM |
|---|---|---|
| No | No | OK |

This operation is used to change bit addresses in operands in the user program.

**Key Sequence**

Program address display → Input operand address → WRITE

**Operation Example**

In the following instruction block, the operand address at 000200, CIO 000001, will be changed to CIO 000002.

| Address | Instruction | Operands | |
|---|---|---|---|
| : | : | : | |
| 000200 | LD | 000001 | ← The operand address will be changed to CIO 000002. |
| 000201 | AND | 000200 | |
| : | : | : | |

***1,2,3...***

1. From the initial display, access the address of the instruction containing the bit address to be changed.

CLR …

```
000000 CT00
```

2 0 0 ↓

```
000200 CT00
LD        000001
```

2. Specify the new operand, and press the **WRITE** Key.

2

```
000200 CT00
LD        000002
```

WRITE

```
000201 CT00
AND       000200
```

3. Press the **Up** Key to check the operand has been changed.

↑

```
000200 CT00
LD        000002
```

## 4-3-5 N.O./N.C. Change

| RUN | MONITOR | PROGRAM |
|---|---|---|
| No | No | OK |

This operation is used to change N.O. and N.C. conditions in the user program.

**Key Sequence**

Program address display → NOT - - - NOT → WRITE

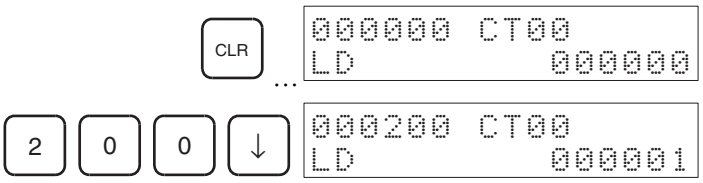
**Operation Example**

In the following instruction block, address 000200 will be changed from LD 000001 to LD NOT 000001.

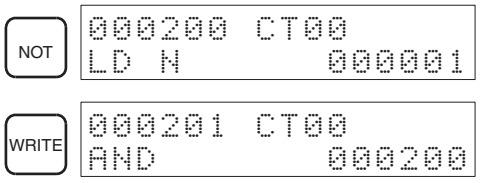
| Address | Instruction | Operands | |
|---|---|---|---|
| : | : | : | |
| 000200 | LD | 000001 | ← Changed to LD NOT 000001. |
| 000201 | AND | 000200 | |
| : | : | : | |

***1,2,3...***

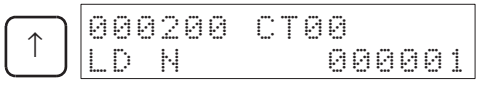
1. From the initial display, access the address containing the instruction to be changed.




2. Press the **NOT** Key and then the **WRITE** Key.




3. Use the **Up** Key to check that LD has been changed to LD NOT.

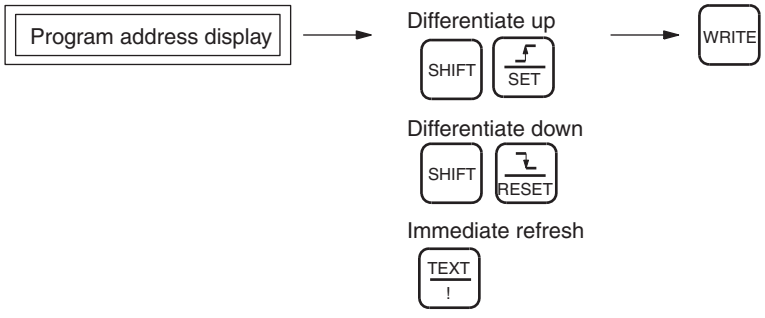



## 4-3-6 Instruction Variation Change

| RUN | MONITOR | PROGRAM |
|---|---|---|
| No | No | OK |

This operation is used to change the variation of an instruction in the user program. Refer to page 110 for further information on key sequences and displays in MONITOR mode.

**Key Sequence**



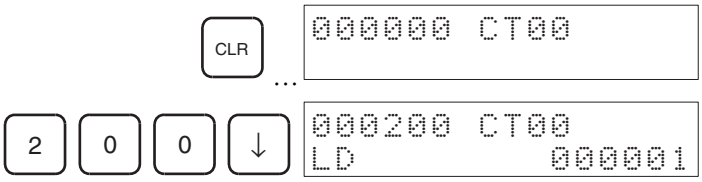
**Operation Example**

In the following instruction block address, 000200 (LD000001) will be changed to an upwardly differentiated LD instruction.

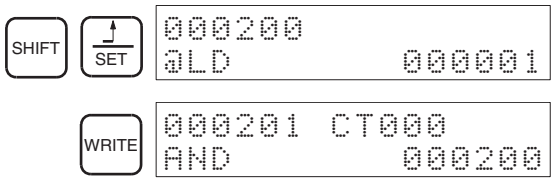
| Address | Instruction | Operands | |
|---|---|---|---|
| : | : | : | |
| 000200 | LD | 000001 | ← Changed to LD 000001. |
| 000201 | AND | 000200 | |
| : | : | : | |

***1,2,3...***

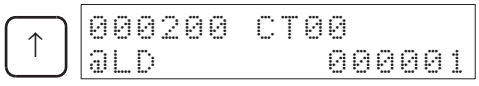
1. From the initial display, access the address of the instruction to be changed.




2. Using the following key sequence, specify the variation and press the **WRITE** Key.




3. Use the **Up** Key to check that the variation has changed.



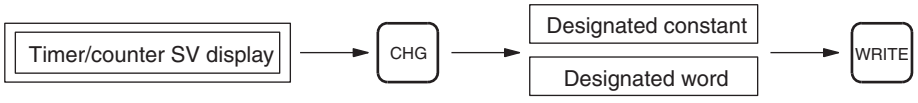

## 4-3-7 Timer/Counter SV Changes

| RUN | MONITOR | PROGRAM |
|---|---|---|
| No | No | OK |

### Timer/Counter SV Change 1: Constant or Word

This operation is used to change the set value of a timer/counter in the user program.

**Key Sequence**

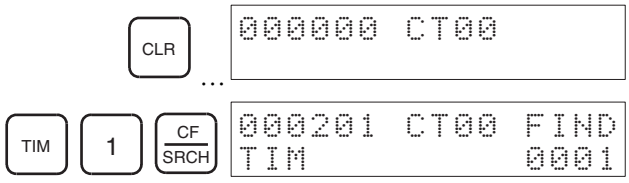


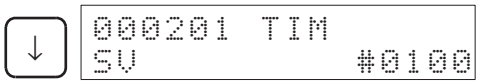

**Operation Example**

The following example describes how to change the set value for TIM 0001.

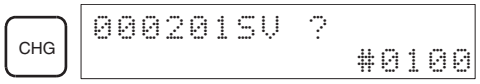
***1,2,3...***

1. From the initial display, access the timer/counter instruction for which the set values will be changed.





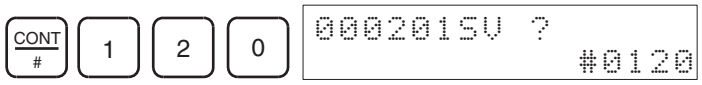

2. Star the set Timer/Counter SV Change 1 operation.

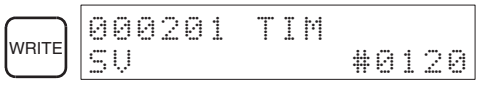



3. The designated word or constant specified for the set value can be changed from the above display.

**Changing a Constant**

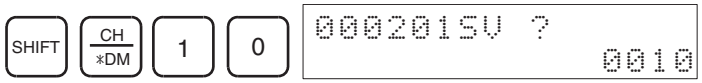
Input the constant as follows:

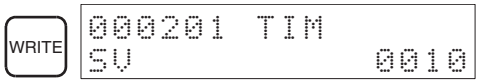



**Changing a Word Address**

Input the word address as follows:

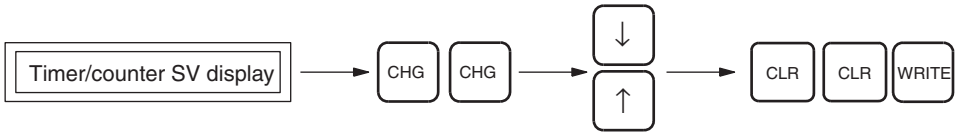



## Timer/Counter SV Change 2: Fine Adjustment of a Constant

This operation is used to increment or decrement a constant specified for the set value of a timer/counter in the user program.

**Key Sequence**

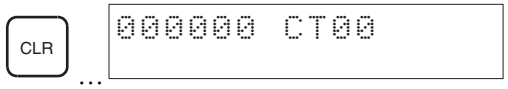


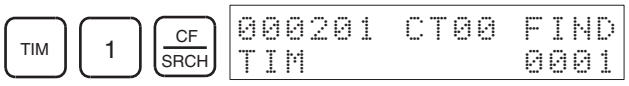

**Operation Example**

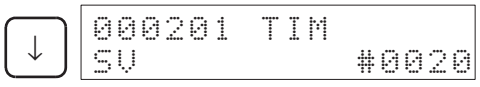
The following example describes how to adjust the constant specified for the TIM 0001 set value.

***1,2,3...***

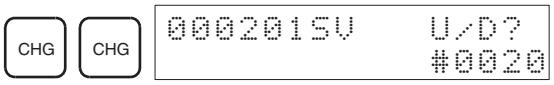
1. From the initial display, access the timer/counter instruction for which the set values will be changed.



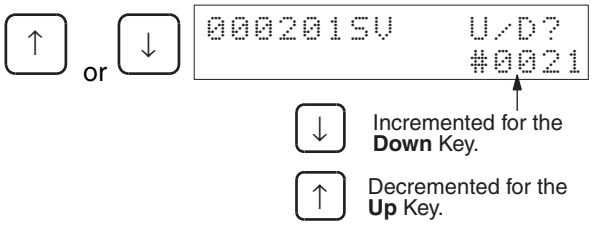



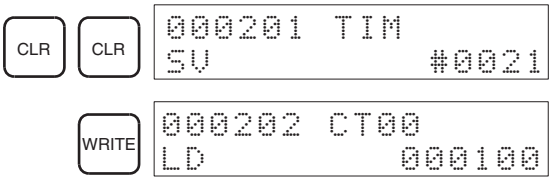

2. Start the Timer/Counter SV Change 2 operation.



3. Incremented or decremented the constant using the **Up** and **Down** Keys.




4. Exit the operation. The new set value will be displayed.



# SECTION 5
# Checking and Adjusting Programs

This section describes how to monitor programs in different display modes, for the purpose of checking and making adjustments. It also includes information on how to change values from various displays for adjusting programs.


5-1 Reading Program Execution Status . . . . . 80
5-2 Simple I/O Monitor . . . . . 81
5-2-1 Monitoring Specified Addresses in I/O Memory . . . . . 81
5-2-2 Monitoring Operands at Specified Program Addresses . . . . . 86
5-3 I/O Multipoint Monitor . . . . . 87
5-4 Monitor Display Format . . . . . 90
5-4-1 Word (Multipoint) Monitor . . . . . 91
5-4-2 Signed Decimal Monitor . . . . . 92
5-4-3 Unsigned Decimal Monitor . . . . . 93
5-4-4 ASCII Monitor . . . . . 94
5-5 Changing Word (16-bit) Present Values . . . . . 95
5-5-1 Basic Procedure for Changing PVs . . . . . 95
5-5-2 Changing PVs from Hexadecimal or BCD Display . . . . . 96
5-5-3 Changing PVs from Signed Decimal Monitor . . . . . 96
5-5-4 Changing PVs from Unsigned Decimal Monitor . . . . . 98
5-5-5 Changing PVs from the ASCII Monitor . . . . . 99
5-5-6 Changing PVs from the Word (Multiple-bit) Monitor . . . . . 100
5-6 Forcing Bits ON/OFF . . . . . 101
5-6-1 Continuous Force Set/Reset . . . . . 101
5-6-2 Temporary Force Set/Reset . . . . . 102
5-6-3 Clear Force Set/Reset . . . . . 103
5-7 Differential Monitor . . . . . 103
5-8 Online Editing . . . . . 104
5-8-1 Instruction Change . . . . . 104
5-8-2 Instruction Add . . . . . 105
5-8-3 Instruction Insert . . . . . 106
5-8-4 Instruction Delete . . . . . 107
5-8-5 Bit Address Change . . . . . 108
5-8-6 N.O./N.C. Change . . . . . 109
5-8-7 Instruction Variation Change . . . . . 110
5-8-8 Timer/Counter SV Change . . . . . 111

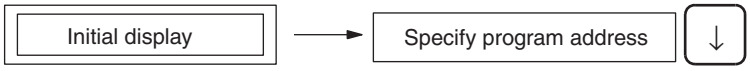
# 5-1 Reading Program Execution Status

## I/O Bit Status Monitor

| RUN | MONITOR | PROGRAM |
|---|---|---|
| OK | OK | No |

This operation is used to read the program execution status one address at a time. By monitoring the ON/OFF status of bits or timer/counter Completion Flags, the execution status of the program can be read from the Programming Console display. The key sequence is the same as that for reading the program.

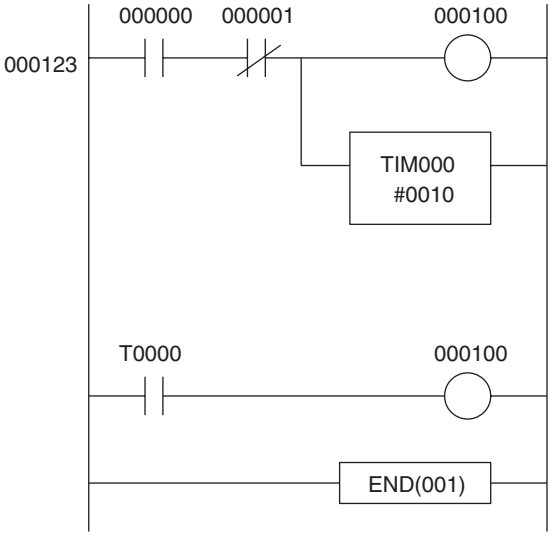
**Key Sequence**




**Operation Example**

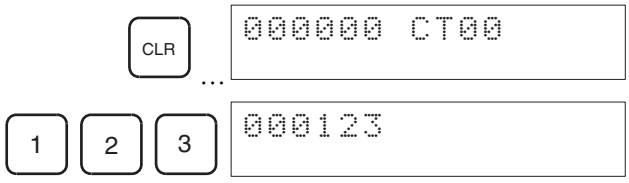
In the following example, the ON/OFF status of the bits in the program are read.

Program example

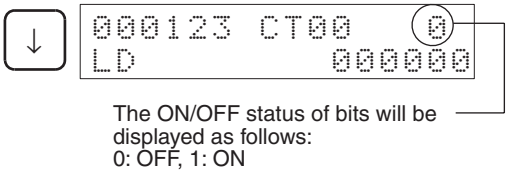


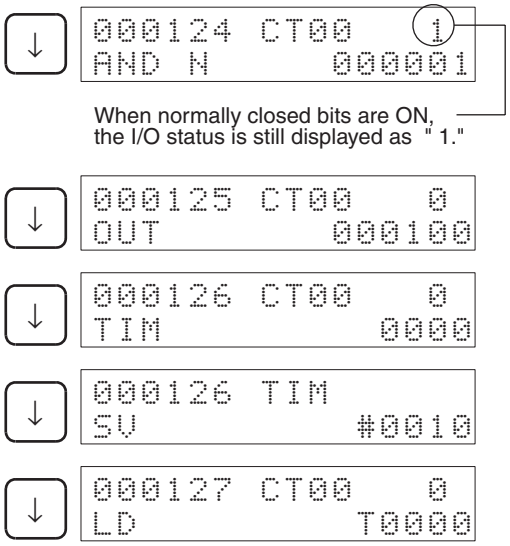

***1,2,3...***

1. From the initial display, specify the program address to read. In this example, program address 123 is specified.




2. Read the program. The ON/OFF status of bits will be displayed as “0” or “1.”






3. Press the **CLR** Key to end the operation.

Note When the **MON** Key is pressed during I/O bit status monitoring, the Programming Console will be switched to the Simple I/O Monitor operation. To switch back to the I/O Bit Status Monitor operation, press the **CLR** Key once more.

# 5-2 Simple I/O Monitor

| RUN | MONITOR | PROGRAM |
|---|---|---|
| OK | OK | OK |

This operation is used to monitor present values in I/O memory one bit or one word at a time.

- There are two methods for I/O monitoring, as follows:
  a) Monitoring specified addresses in I/O memory.
  b) Specifying program addresses and monitoring the operands of the corresponding instructions.
- Simple I/O Monitor is available in RUN, MONITOR and PROGRAM modes.
- When monitoring bit status, the ON/OFF status of the bit can be changed. Refer to *5-6 Forcing Bits ON/OFF* for details.
- When monitoring word data (16-bit data), the contents of the word can be changed. Refer to *5-5 Changing Word (16-bit) Present Values* for details.

## 5-2-1 Monitoring Specified Addresses in I/O Memory

### Monitoring ON/OFF Status of Specified Bits (Single-bit Data)

**Key Sequence**

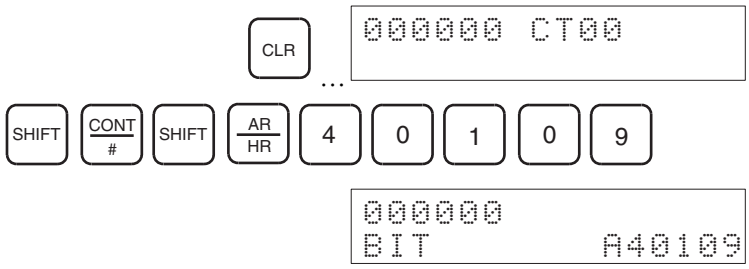


**Operation Example**

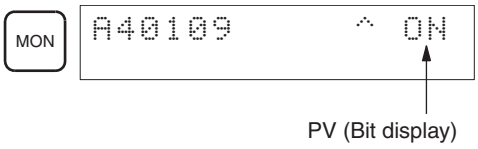
In the following example, auxiliary bit (AR) A40109 is monitored.

***1,2,3...***

1. From the initial display, input the bit address to be monitored. In this example, input A40109.




2. Start monitoring.




   - Press the **Up** and **Down** Keys to monitor either the previous or the next bit.
   - Bit ON/OFF status can be changed from this display.

3. Press the **CLR** Key to exit Simple I/O Monitor.

**Designating Bit Address for Monitoring**

| Type | Key sequence | Display example |
|---|---|---|
| CIO bits | SHIFT, CONT/#, Number | Bit 15 of CIO 0000<br>000000<br>BIT 000015 |
| Work bits (WR) | SHIFT, CONT/#, *EM/WR/LR, Number | Bit 15 of W000<br>000000<br>BIT W00015 |
| Holding bits (HR) | SHIFT, CONT/#, AR/HR, Number | Bit 15 of H000<br>000000<br>BIT H00015 |
| Auxiliary bits (AR) | SHIFT, CONT/#, SHIFT, AR/HR, Number | Bit 15 of A000<br>000000<br>BIT A00015 |

<table>
<tr><th>Type</th><th>Key sequence</th><th>Display example</th></tr>
<tr><td rowspan="2">Clock pulse/Condition Flag</td><td>SHIFT CF/SRCH</td><td>Always ON Flag<br>000000<br>BIT ON</td></tr>
<tr><td>When clock pulses or Condition Flags are designated, the bits listed at the right can be designated, by using the Up and Down Keys.<br><table><tr><th>Display</th><th>Bit</th></tr><tr><td>ON</td><td>Always ON Flag</td></tr><tr><td>OFF</td><td>Always OFF Flag</td></tr><tr><td>AER</td><td>Access Error Flag</td></tr><tr><td>0.02s</td><td>0.02-s clock pulse</td></tr><tr><td>0.1s</td><td>0.1-s clock pulse</td></tr><tr><td>0.2s</td><td>0.2-s clock pulse</td></tr><tr><td>1s</td><td>1-s clock pulse</td></tr><tr><td>1min</td><td>1-min clock pulse</td></tr><tr><td>ER</td><td>Error Flag</td></tr><tr><td>CY</td><td>Carry Flag</td></tr><tr><td>></td><td>Greater Than Flag</td></tr><tr><td>=</td><td>Equals Flag</td></tr><tr><td><</td><td>Less Than Flag</td></tr><tr><td>N</td><td>Negative Flag</td></tr><tr><td>OF</td><td>Overflow Flag</td></tr><tr><td>UF</td><td>Underflow Flag</td></tr><tr><td>>=</td><td>Greater Than or Equals Flag</td></tr><tr><td><></td><td>Not Equal Flag</td></tr><tr><td><=</td><td>Less Than or Equals Flag</td></tr></table></td></tr>
<tr><td>Task Flag</td><td>SHIFT TK/CNT Number</td><td>TK0000<br>000000<br>BIT TK0000</td></tr>
</table>

## Monitoring Timer/Counter Completion Flags and PVs

| Set value input range | Simple I/O Monitor present value display (initial status) |
|---|---|
| #0000 to #9999 | 0000 to 9999 |

**Example:** When the present value is 10 s, the values are as follows:

| Set value input range | Simple I/O Monitor present value display (initial status) |
|---|---|
| #0100 | 0100 |

**Key Sequence**

TIM Timer number MON

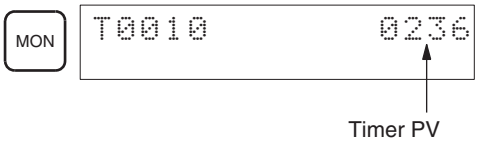
TK/CNT Counter number MON

**Operation Example** In the following example, TIM 0010 is monitored.

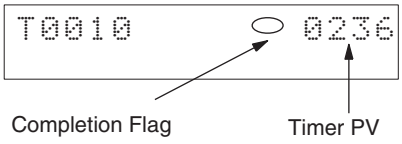
***1,2,3...*** 1. From the initial display, input the number of the timer instruction to be monitored. In this example, input TIM 0010.




2. Start monitoring.




- The Completion Flag and PV are monitored at the same time.
- When the timer/counter PV becomes "0," the Completion Flag will turn ON and a symbol will be displayed to the right of the PV.




3. Press the **CLR** Key to exit Simple I/O Monitor.

**Designating Timer/Counter Numbers for Monitoring**

| Type | Key sequence | Display example |
|---|---|---|
| Timer PVs | TIM Number | TIM0100<br>000000<br>TIM 0100 |
| Counter PVs | TK/CNT Number | C0010<br>000000<br>CNT 0010 |

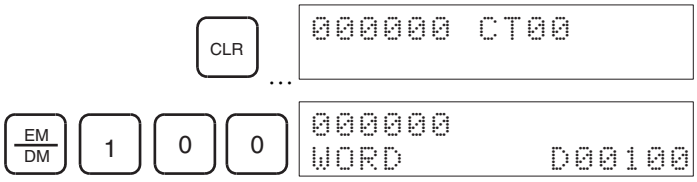
## Monitoring the PVs of Designated Words (16-bit Data)

**Key Sequence**

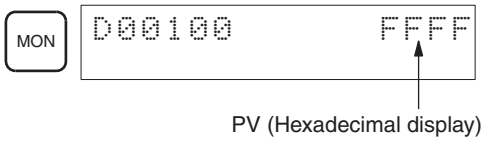



**Operation Example** In the following example, D00100 in data memory is monitored.

***1,2,3...*** 1. From the initial display, input the DM address to be monitored. In this example, input D00100.



2. Start monitoring.




- Press the **Up** and **Down** Keys to monitor either the previous or the next word.
- Word values can be changed from this display.

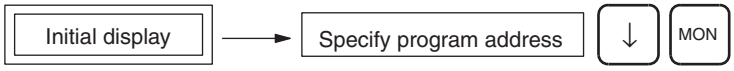
3. Press the **CLR** Key to exit Simple I/O Monitor.

**Designating Word Addresses (16-bit) for Monitoring**

| Type | Key sequence | Display example |
|---|---|---|
| CIO words | SHIFT, CH/*DM, Number | CIO0010<br>000000<br>WORD 0010 |
| Work words (WR) | SHIFT, CH/*DM, *EM/WR/LR, Number | W010<br>000000<br>WORD W010 |
| Holding words (HR) | SHIFT, CH/*DM, AR/HR, Number | H010<br>000000<br>WORD H010 |
| Auxiliary words (AR) | SHIFT, CH/*DM, SHIFT, AR/HR, Number | A010<br>000000<br>WORD A010 |
| Data memory words (DM) | EM/DM, Number | D00010<br>000000<br>WORD D00010 |
| Extended DM (EM): With bank | *EM/EM_/EXT, Number (Bank number, word address) | E0_00010<br>000000<br>WORD E0_00010 |
| Extended DM (EM): Current bank | SHIFT, EM/DM, Number (Word address) | E00010<br>000000<br>WORD E00010 |
| Data Registers (DR) | SHIFT, DR/IR, Number | DR1<br>000000<br>Regi DR01 |
| Index Registers (IR) | DR/IR, Number | IR1<br>000000<br>Regi IR01 |

## 5-2-2 Monitoring Operands at Specified Program Addresses

### Key Sequence



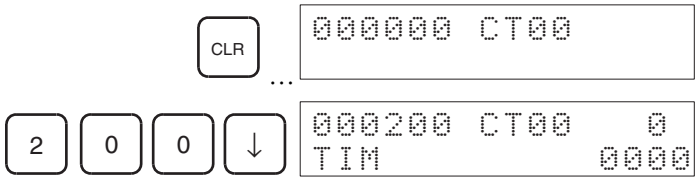

### One Operand

**Operation Example**

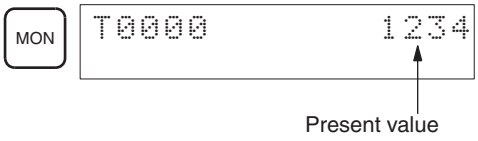
In the following example, the PV for the TIM instruction at program address 000200 is monitored.

***1,2,3...***

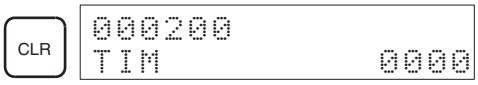
1. From the initial display, input the program address of the instruction whose operand is to be monitored. In this example, input program address 000200.




2. Start monitoring.




3. Press the **CLR** Key to exit Simple I/O Monitor.



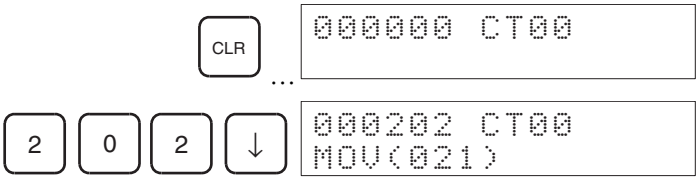

### Two or More Operands

**Operation Example**

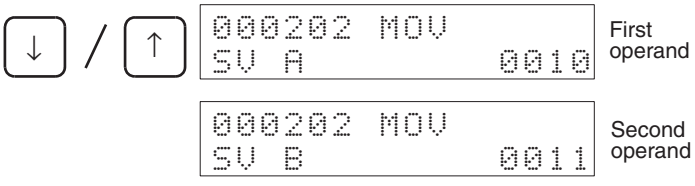
In the following example, operands for the MOV(021) instruction at program address 000202 is monitored

***1,2,3...***

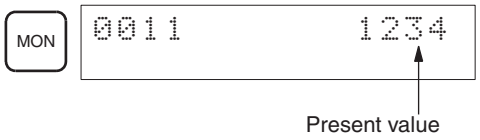
1. From the initial display, input the program address of the instruction whose operands to be monitored. In this example, input program address 000202.



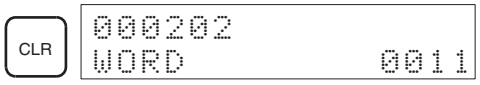

2. Select the operands to be monitored. The following example shows the display for the first and second operands.



3. Start monitoring.




4. Press the **CLR** Key to exit Simple I/O Monitor.



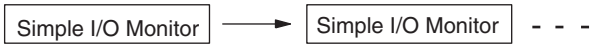

# 5-3 I/O Multipoint Monitor

| RUN | MONITOR | PROGRAM |
|---|---|---|
| OK | OK | OK |

This operation is used to monitor up to four bits or words simultaneously.

- Four bits and/or words can be monitored at the same time, but only 2 of these can be displayed simultaneously.

**Key Sequence**

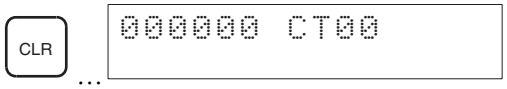



**Operation Example**

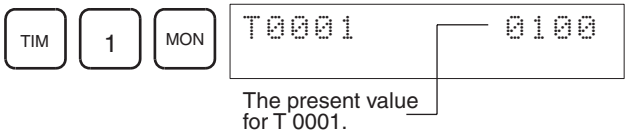
In the following example, T0001, CIO 000001, D00000, and CIO 1225 are monitored.

***1,2,3...***

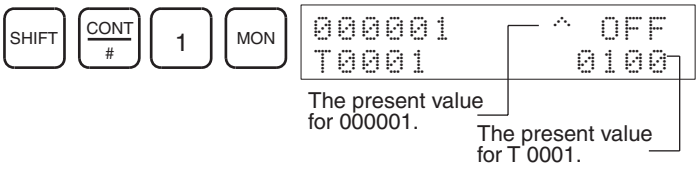
1. Specify the bits and words for monitoring, in succession as shown in the following steps. The status/present values of the specified bits and words will appear at the top of the display. Methods for specifying bits or words are the same as for Simple I/O Monitor.



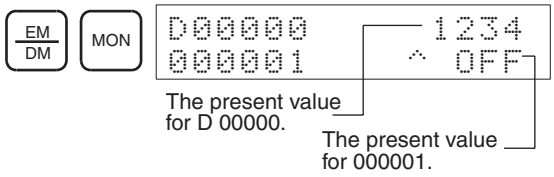

a) Specify T0001 and the present value will appear at the top of the display.



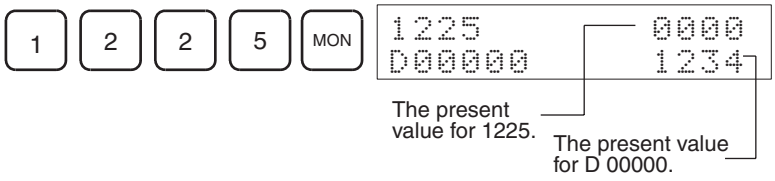

b) Specify CIO bit 000001 and the status will appear at the top of the display.



c) Specify D000000 and the present value will appear at the top of the display.

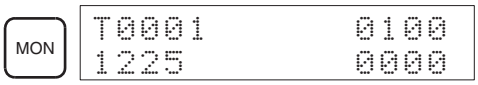



d) Specify CIO 1225 and the present value will appear at the top of the display.

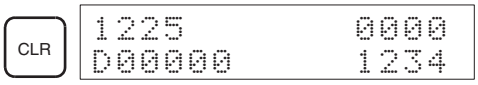



- A total of up to 4 bits and words can be specified, but only a maximum of 2 of these will be displayed at any one time. If more than 4 bits/words are specified, old ones will be cleared from the monitor operation.

2. Press the **MON** Key to display bits and words that were not visible.



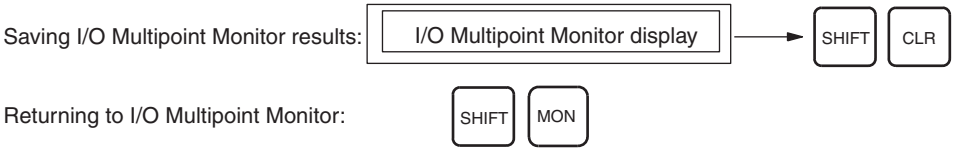

3. Press the **CLR** Key and bit/word being monitored on the top line will be cleared from monitor operation.




## Saving I/O Multipoint Monitor Results

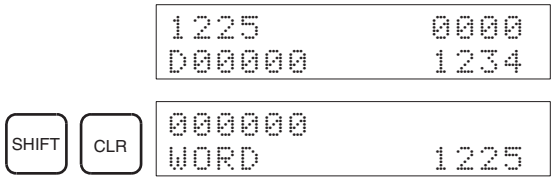
This operation is used to save the current bits/words specified for the I/O Multipoint Monitor operation while performing other operations.

### Key Sequence

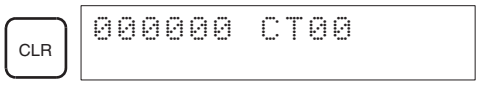



### Operation Example

***1,2,3...*** 1. From the I/O Multipoint Monitor display, press the **SHIFT** then **CLR** Keys to return to the initial display.



2. From the initial display, perform any operation other than I/O Multipoint Monitor.




3. When you have finished performing other operations, return to the saved monitoring status by accessing the I/O Multipoint Monitor display.




## Clearing all Monitor Displays

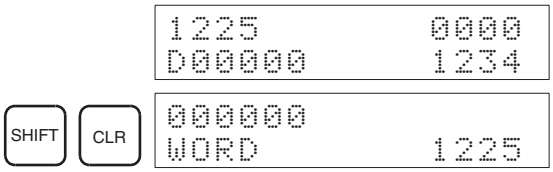
This operation is used to clear from the display all bits/words being monitored in I/O Multipoint Monitor. The specification of the bits/words being monitoring will be saved and the Multipoint I/O Monitor operation can be returned to using the procedure described in the preceding section.

**Key Sequence**



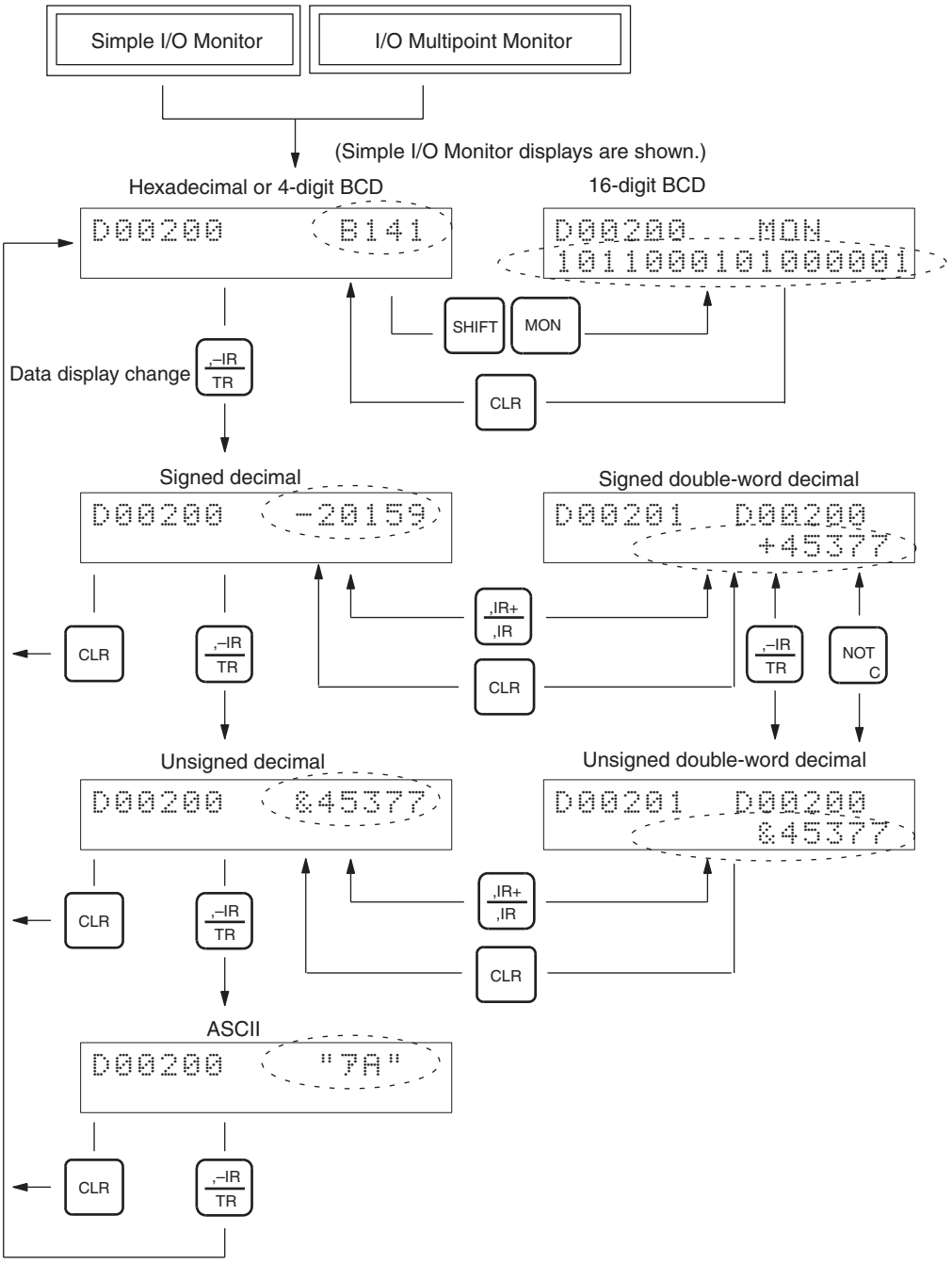

**Operation Example**

From the I/O Multipoint Monitor display, press **SHIFT** then **CLR** Keys.



# 5-4 Monitor Display Format

The display format for words (16-bit data) in Simple I/O Monitor or I/O Multipoint Monitor can be changed to display formats other then binary (i.e., 4-digit and 2-digit hexadecimal), such as signed decimal, unsigned decimal, and ASCII. The word (16-bit data) display format can be changed using the following key combinations.



## 5-4-1 Word (Multipoint) Monitor

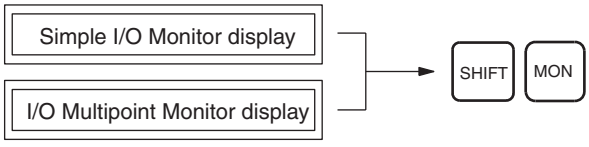
| RUN | MONITOR | PROGRAM |
|---|---|---|
| OK | OK | OK |

This operation monitors I/O memory bit status in one word (16 bits) at a time.

- The following words can be monitored:
  - I/O bits
  - Holding bits
  - Data memory
  - Data registers
  - Work bits
  - Auxiliary bits
  - Extended data memory
  - Index registers
- The status of the 16 bits will be shown on the bottom line of the display using 1, 0, S, and R as follows:

  1: ON S: Force Set
  0: OFF R: Force Reset

**Key Sequence**

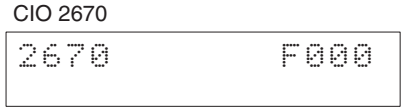



**Operation Example**

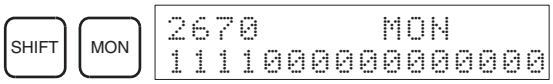
In the following example, CIO 2670 is monitored using a word (multipoint) display.

***1,2,3...***

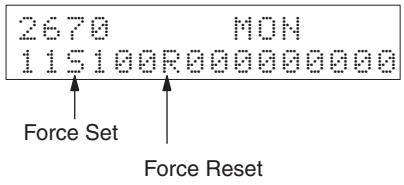
1. Access the Simple I/O Monitor display and display the word for monitoring.




2. Switch to word monitoring.




- Use the **Up** or **Down** Key to display the next or previous word.
- From this display, word values can be changed one bit at a time. Refer to page *5-5 Changing Word (16-bit) Present Values*.
- The Force Set/Reset status of bits will also be displayed. (S: Force Set, R: Force Reset)



3. Press the **CLR** Key to stop word monitoring and return to the Simple I/O Monitor display.



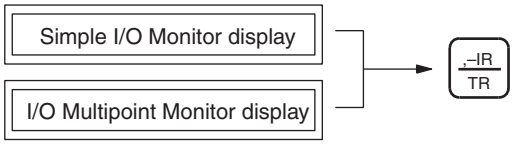

## 5-4-2 Signed Decimal Monitor

| RUN | MONITOR | PROGRAM |
|---|---|---|
| OK | OK | OK |

This operation converts the hexadecimal contents of words from 2's complement to a signed decimal data display.

This operation can be performed during Simple I/O Monitor or I/O Multipoint Monitor.

**Key Sequence**

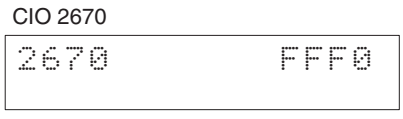



**Operation Example**

In the following example, CIO 2670 is monitored using a signed decimal data display.

***1,2,3...***

1. Access the Simple I/O Monitor or I/O Multipoint Monitor display and display the word for monitoring.

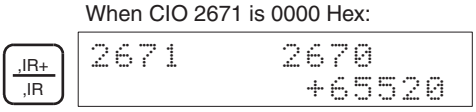



During I/O Multipoint monitoring, the word at the top of the screen will be changed.

2. Convert the display format to signed decimal.




3. Convert the display to a double-word display.




From this display, word data can be converted to signed decimal. (Refer to page 96.)

4. Press the **CLR** Key to exit the double-word display and return to Signed Decimal Monitor.



5. Press the **CLR** Key again to return the display from Signed Decimal Monitor to either Simple I/O Monitor or I/O Multipoint Monitor.



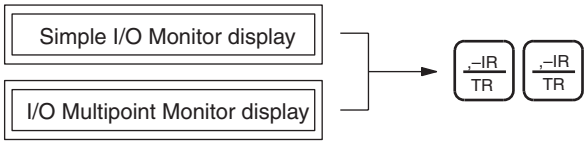

## 5-4-3 Unsigned Decimal Monitor

| RUN | MONITOR | PROGRAM |
|---|---|---|
| OK | OK | OK |

This operation converts the hexadecimal contents of word data into unsigned decimal data for display.

This operation can be performed during Simple I/O Monitor or I/O Multipoint Monitor.

**Key Sequence**

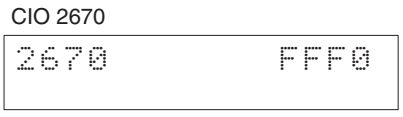



**Operation Example**

In the following example, CIO 2670 is monitored using an unsigned decimal display.

***1,2,3...***

1. Access the Simple I/O Monitor or I/O Multipoint Monitor display and display the word for monitoring.

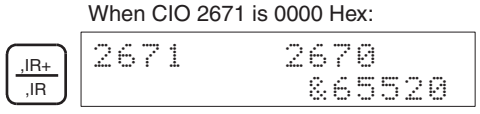



During I/O Multipoint Monitor, the word at the top of the screen will be changed.

2. Convert the display format to unsigned decimal.




3. Convert the display to a double-word display.




From this display, word data can be converted to signed decimal. (Refer to page 98.)

4. Press the **CLR** Key to exit double-word display and return to Unsigned Decimal Monitor.



5. Press the **CLR** Key again to return the display from Unsigned Decimal Monitor to either Simple I/O Monitor or I/O Multipoint Monitor.



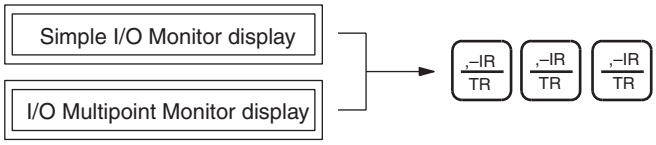

## 5-4-4 ASCII Monitor

| RUN | MONITOR | PROGRAM |
|---|---|---|
| OK | OK | OK |

This operation reads the hexadecimal contents of words as ASCII data, and displays them as ASCII text.

- This operation can be performed during Simple I/O Monitor or I/O Multipoint Monitor.
- For details on ASCII characters which can be displayed, refer to Appendix B.

**Key Sequence**

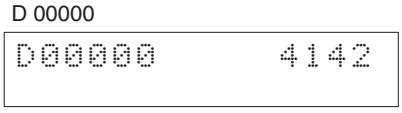



**Operation Example**

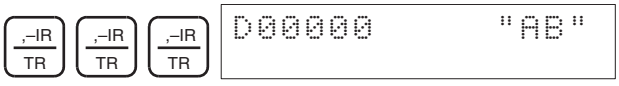
In the following example, D00000 is monitored using an ASCII display.

***1,2,3...***

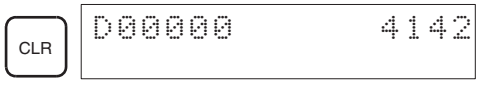
1. Access the Simple I/O Monitor or I/O Multipoint Monitor display and display the word for which the display will be changed.




2. Convert the display format to ASCII display.




3. Press the **CLR** Key to exit ASCII display and return to either Simple I/O Monitor or I/O Multipoint Monitor.



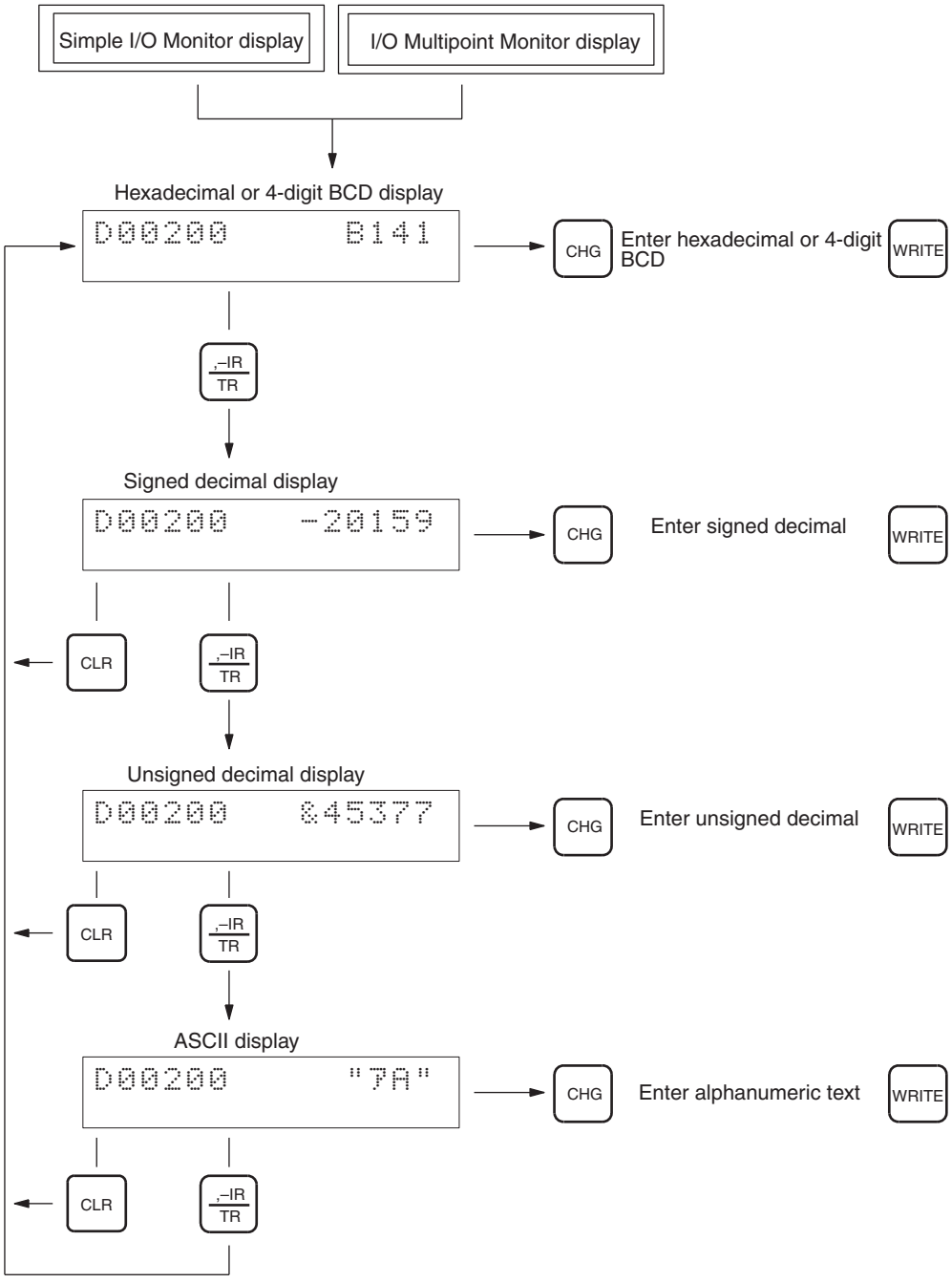
# 5-5 Changing Word (16-bit) Present Values

## 5-5-1 Basic Procedure for Changing PVs

| RUN | MONITOR | PROGRAM |
|---|---|---|
| No | OK | OK |

The operations used to change word data are illustrated below.

⚠ **Caution** Always confirm safety before changing data even when working in PROGRAM mode. The CPU Unit will refresh I/O even in PROGRAM mode. If the status of a bit allocated to an Output Unit, Special I/O Unit, or CPU Bus Unit is changed, the load connected to the Unit may operate unexpectedly.



## 5-5-2 Changing PVs from Hexadecimal or BCD Display

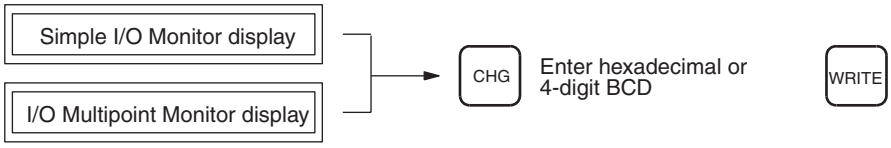
| RUN | MONITOR | PROGRAM |
|---|---|---|
| No | OK | OK |

This operation changes word (16-bit) present values using hexadecimal or 4-digit BCD.

This operation can be performed during Simple I/O Monitor or I/O Multipoint Monitor. From I/O Multipoint Monitor, the top value on the display will be changed.

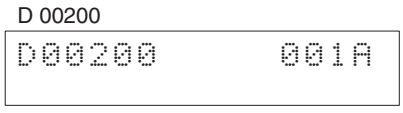
⚠ **Caution** Always confirm safety before changing data even when working in PROGRAM mode. The CPU Unit will refresh I/O even in PROGRAM mode. If the status of a bit allocated to an Output Unit, Special I/O Unit, or CPU Bus Unit is changed, the load connected to the Unit may operate unexpectedly.

**Key Sequence**

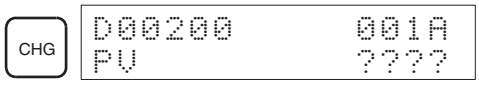



**Operation Example** The following example changes the present value of D00200.

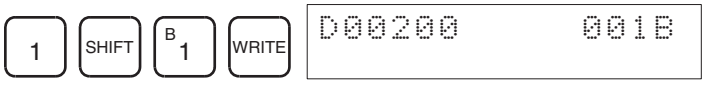
***1,2,3...*** 1. Access the Simple I/O Monitor or I/O Multipoint Monitor and bring up the word for changing.




2. Specify changing the value from the hexadecimal display.




3. Input the new value and return to Simple I/O Monitor or Multipoint I/O Monitor.




## 5-5-3 Changing PVs from Signed Decimal Monitor

| RUN | MONITOR | PROGRAM |
|---|---|---|
| No | OK | OK |

This operation changes present values for word data using signed decimal (single-word: –32768 to 32767, double-word: –214783648 to 214783647). The 2's complement expression is automatically converted to hexadecimal.

This operation can be performed during Simple I/O Monitor or I/O Multipoint Monitor. From I/O Multipoint Monitor, the top value on the display will be changed.

⚠ **Caution** Always confirm safety before changing data even when working in PROGRAM mode. The CPU Unit will refresh I/O even in PROGRAM mode. If the status of a bit allocated to an Output Unit, Special I/O Unit, or CPU Bus Unit is changed, the load connected to the Unit may operate unexpectedly.

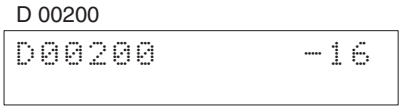
**Key Sequence**



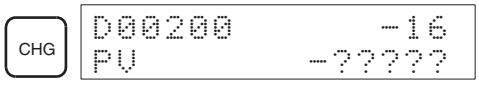

## Signed Decimal Displays

**Operation Example** The following example changes the present value of D00200.

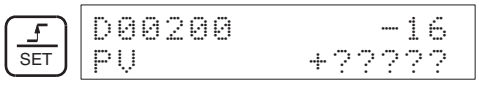
***1,2,3...*** 1. Access the Simple I/O Monitor or I/O Multipoint Monitor as signed decimal display, and bring up the word for changing.




2. Specify changing the value from the signed decimal display.

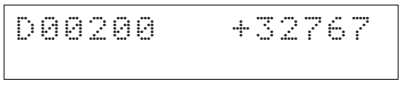



3. Input the new value and return to Signed Decimal Monitor.







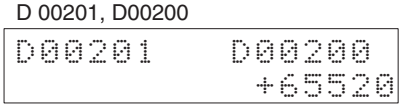



- When inputting positive numbers, press the **SET** Key; when inputting negative numbers, press the **RESET** Key.
- When an error is made inputting, press the **CLR** Key to return to the previous display, then enter the value correctly.

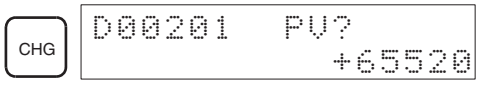
## Double-word Displays

**Operation Example** The following example changes the present values of D00201 and D00200.

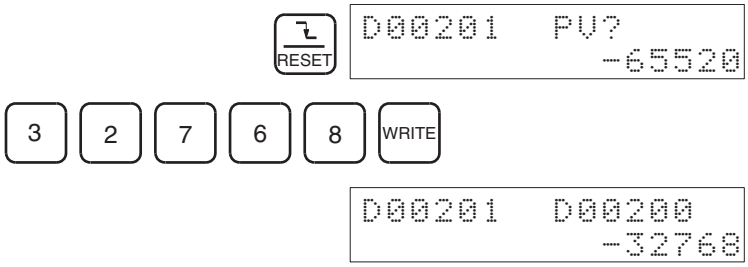
***1,2,3...*** 1. Access the Simple I/O Monitor or I/O Multipoint Monitor as signed decimal display, and bring up the words for changing.




2. Specify changing the values from the double-word signed decimal display.



3. Input the new values and return to Signed Decimal Monitor.




- When inputting positive numbers, press the **SET** Key; when inputting negative numbers, press the **RESET** Key.
- When an error is made inputting, press the **CLR** Key to return to the previous display, then enter the value correctly.

## 5-5-4 Changing PVs from Unsigned Decimal Monitor

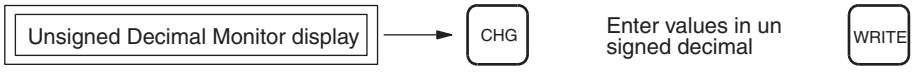
| RUN | MONITOR | PROGRAM |
|---|---|---|
| No | OK | OK |

This operation changes present values for word data using unsigned decimal (single-word data: 0 to 65535, double-word: 0 to 4294967295). Conversions to hexadecimal are performed automatically.

This operation can be performed during Simple I/O Monitor or I/O Multipoint Monitor. From I/O Multipoint Monitor, the top value on the display will be changed.

⚠ **Caution** Always confirm safety before changing data even when working in PROGRAM mode. The CPU Unit will refresh I/O even in PROGRAM mode. If the status of a bit allocated to an Output Unit, Special I/O Unit, or CPU Bus Unit is changed, the load connected to the Unit may operate unexpectedly.

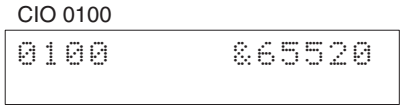
**Key Sequence**




### Unsigned Decimal Displays

**Operation Example** The following example changes the present value of CIO 0100.

***1,2,3...*** 1. Access the Simple I/O Monitor or I/O Multipoint Monitor as unsigned decimal display, and bring up the word for changing.




2. Specify changing the value from the unsigned decimal display.




3. Input the new value and return to Unsigned Decimal Monitor.






When an error is made inputting, press the **CLR** Key to return to the previous display, then enter the value correctly.

### Double-word Display

***1,2,3...*** 1. Access the Simple I/O Monitor or I/O Multipoint Monitor as unsigned decimal display, and bring up the words for changing.

CIO 0101, CIO 0100

```
0101      0100
            &65520
```

2. Specify changing the values from the double-word unsigned decimal display.

CHG

```
0101      PV?
            &65520
```

3. Input the new values and return to Unsigned Decimal Monitor.

6 5 5 3 5 WRITE

```
0101      0100
            &65535
```

When an error is made inputting, press the **CLR** Key to return to the previous display, then enter the value correctly.

## 5-5-5 Changing PVs from the ASCII Monitor

| RUN | MONITOR | PROGRAM |
|---|---|---|
| No | OK | OK |

This operation changes present values for word data using ASCII.

- This operation can be performed during Simple I/O Monitor or I/O Multipoint Monitor. From I/O Multipoint Monitor, the top value on the display will be changed.
- For details on ASCII characters which can be displayed, refer to Appendix B.

⚠ **Caution** Always confirm safety before changing data even when working in PROGRAM mode. The CPU Unit will refresh I/O even in PROGRAM mode. If the status of a bit allocated to an Output Unit, Special I/O Unit, or CPU Bus Unit is changed, the load connected to the Unit may operate unexpectedly.

**Key Sequence**

ASCII Monitor display → CHG Enter values as numerals. WRITE

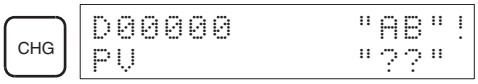
**Operation Example** The following example changes the present value for D00000.

***1,2,3...*** 1. Access the Simple I/O Monitor or I/O Multipoint Monitor as an ASCII display, and bring up the word for changing.

D 00000

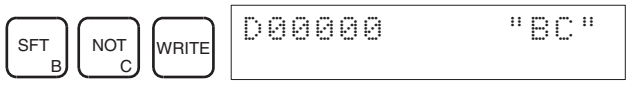
```
D00000      "AB"
```

2. Specify changing the text from the ASCII display.




When " " is displayed, the text on the lower-left of the keys will be input when keys are pressed.

3. Input the new value and return to Simple I/O Monitor or I/O Multipoint Monitor.



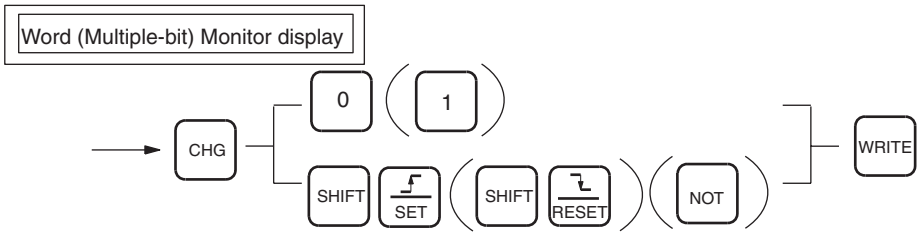

## 5-5-6 Changing PVs from the Word (Multiple-bit) Monitor

| RUN | MONITOR | PROGRAM |
|---|---|---|
| No | OK | OK |

This operation converts word (16-bit) present values to multiple-bit (16-bit binary) format. This operation can be used during Word (Multiple-bit) Monitor.

**⚠Caution** Always confirm safety before changing data even when working in PROGRAM mode. The CPU Unit will refresh I/O even in PROGRAM mode. If the status of a bit allocated to an Output Unit, Special I/O Unit, or CPU Bus Unit is changed, the load connected to the Unit may operate unexpectedly.

**Key Sequence**

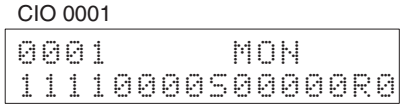



**Operation Example**

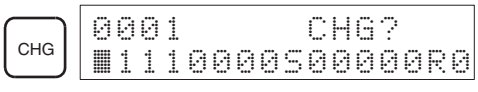
The following example changes the present value of CIO 0001.

***1,2,3...***

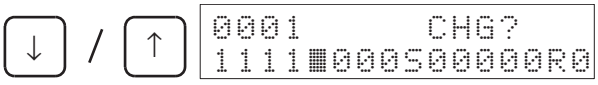
1. Access the Word (Multiple-bit Monitor) and bring up the word to be changed.



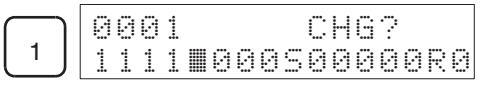

2. Press the **CHG** Key to specify changing the present value.




3. Press the **Up** and **Down** Keys to move the cursor to the bit to be changed.




4. Specify the new value.



The cursor position after inputting a new value varies depending on the last key pressed for moving the cursor.
Press the **Up** and **Down** Keys to confirm the new value.

Press the **Down** Key

```
0001      CHG?
11111■00S00000R0
```

Press the **Up** Key

```
0001      CHG?
111■1000S00000R0
```

The procedure for inputting the new value is as follows:

0 : OFF

1 : ON

SHIFT SET : Force Set (S) (Not possible in data memory)

SHIFT RESET : Force Reset (R) (Not possible in data memory)

NOT : Clear Force Set/Reset

5. Input the new values and return to the Word Monitor.

WRITE

```
0001      MON
11111000S00000R0
```

# 5-6 Forcing Bits ON/OFF

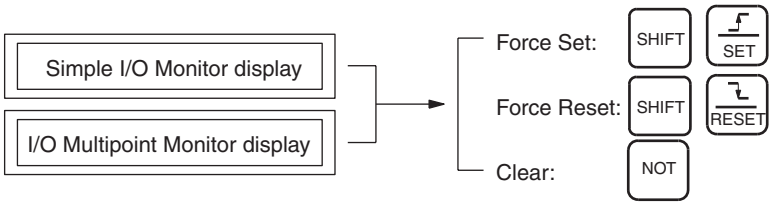
| RUN | MONITOR | PROGRAM |
|---|---|---|
| No | OK | OK |

This operation is used to force bits ON/OFF regardless of any external inputs or results of user program execution. This operation can be performed during Simple I/O Monitor or I/O Multipoint Monitor.

⚠ **Caution** Always confirm safety before setting or resetting bits even when working in PROGRAM mode. The CPU Unit will refresh I/O even in PROGRAM mode. If the status of a bit allocated to an Output Unit, Special I/O Unit, or CPU Bus Unit is changed, the load connected to the Unit may operate unexpectedly.

## 5-6-1 Continuous Force Set/Reset

**Key Sequence**



**Operation Example**

The following example uses Force Set/Reset to reset bit 00 of CIO 0001.

***1,2,3...***

1. Access the Simple I/O Monitor or the I/O Multipoint Monitor and bring up the bit to be Force Set/Reset. The following example shows the I/O Multipoint Monitor.

Bit 00 of CIO 0001

```
000100      ^ OFF
000001      ^ ON
```

From the I/O Multipoint Monitor, the bit at the top of the display will be changed.

2. When a bit has been Force Set, an "S" will appear and when it has been Force Reset, an "R" will appear on the display. The forced ON/OFF status of the bit will not be affected by external input changes or the result of executing a command.




```
000100      S ON
000001      ^ ON
```

Key sequence:



: Continuous Force Set.



: Continuous Force Reset.



: Continuous Force Set/Reset clear.

Continuous Force Set/Reset can be cleared by pressing the NOT Key to perform the Clear Force Set/Reset operation. Force Set/Reset is cleared when the following operations are performed:

a) When the Clear Force Set/Reset operation is executed.

b) When the operating mode of the PLC is changed. If the Forced Status Hold Bit is ON, however, Force Set/Reset bits will not be cleared when the PLC mode is changed from PROGRAM to MONITOR.

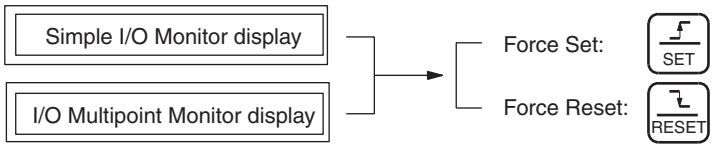
c) When operation stops as the result of a fatal error.

d) When operation stops as a result of power interruption.

## 5-6-2 Temporary Force Set/Reset

This operation force-sets/resets a bit only while the key is held down.

**Key Sequence**



**Operation Example** The following example uses Force Set/Reset to reset bit 00 of CIO 0001.

***1,2,3...*** 1. Access the Simple I/O Monitor or the I/O Multipoint Monitor and bring up the bit to be Force Set/Reset. The following example shows the I/O Multipoint Monitor.

Bit 00 of CIO 0001

```
000100      ^ OFF
000001      ^ ON
```

From the I/O Multipoint Monitor, the bit at the top of the display will be changed.

2. The bit will be force-set/reset only while the key is being pressed. While the key is being pressed, a black square will be displayed.




```
000100      ▩ OFF
000001      ^ ON
```

Key sequence



: Force Set (Only while the key is pressed.)



: Force Reset (Only while the key is pressed.)

## 5-6-3 Clear Force Set/Reset

This operation clears all bits that are being force-set/reset.

**Key Sequence**




**Operation Example** Use the following key strokes to clear all force-set/reset bits.




```
000000REL FORCED
?
```




```
000000REL FORCED
END
```

**Note** Press the **CLR** Key to cancel the operation.

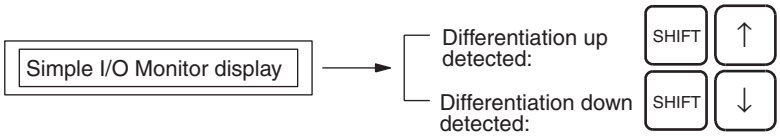
# 5-7 Differential Monitor

| RUN | MONITOR | PROGRAM |
|---|---|---|
| OK | OK | No |

This operation monitors an OFF to ON (upward differentiation) or ON to OFF (downward differentiation) change in the status of a specific bit. When the change is detected, the status will be displayed and the buzzer will sound.

This operation can be used during Simple I/O Monitor or I/O Multipoint Monitor.

**Key Sequence**

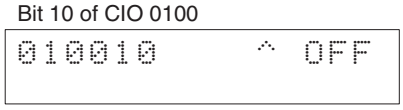



**Operation Example**

The following example detects the differentiate up/differentiate down status of bit 10 of CIO 0100.

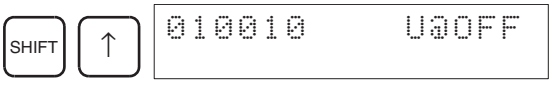
***1,2,3...***

1. Access the Simple I/O Monitor or I/O Multipoint Monitor display, and bring up the bit to be monitored.




From the I/O Multipoint Monitor, the bit at the top of the display will be monitored.

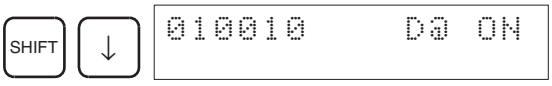
2. Set detection of the differentiate up status. In this example, an OFF to ON change is detected in the status of CIO 010010.




3. When the specified bit turns ON, the display will change and the buzzer will sound.




4. Set detection of differentiate down status. In this example, an ON to OFF change is detected in the status of bit CIO 010010.




5. When the specified bit turns OFF, the display will change and the buzzer will sound.




To exit the Differential Monitor operation, press the **CLR** Key.

# 5-8 Online Editing

## 5-8-1 Instruction Change

| RUN | MONITOR | PROGRAM |
|---|---|---|
| No | OK | OK |

This operation is used when the CPU Unit is in MONITOR mode to overwrite instructions in user programs.

**Key Sequence**



**Operation Example**

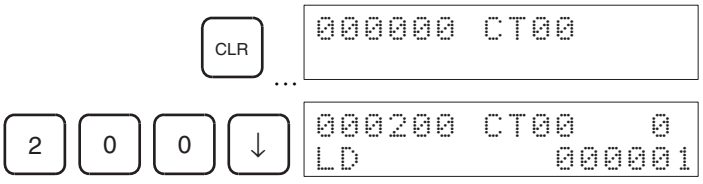
In the following instruction block, the instruction at address 00200 will be changed to LD NOT 000002.

| Address | Instruction | Operand |
|---|---|---|
| : | : | : |
| 000200 | LD | 000001 |
| 000201 | AND | 000200 |
| : | : | : |

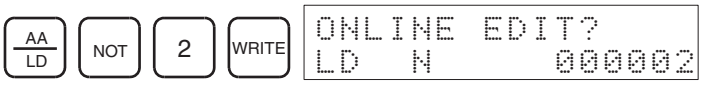
← Changed to LD NOT 000002.

***1,2,3...***

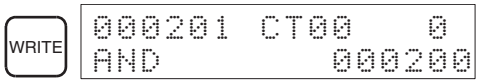
1. From the initial display, access the program address containing the instruction to be changed.




2. Input the new instruction and press the **WRITE** Key.

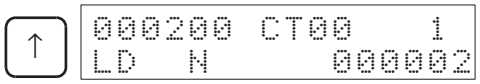



3. Press the **WRITE** Key once more.




After pressing the **WRITE** Key the second time, the altered program will be executed from the next scan.

4. Press the **Up** Key to check the instruction.



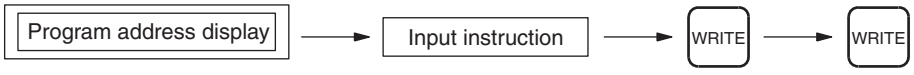

## 5-8-2 Instruction Add

| RUN | MONITOR | PROGRAM |
|---|---|---|
| No | OK | See note. |

This operation is used when the CPU Unit is in MONITOR mode to add an instruction to the end of the user program.

**Note** Refer to *4-1 Inputting Instructions* for details on adding instructions in PROGRAM mode.

**Key Sequence**



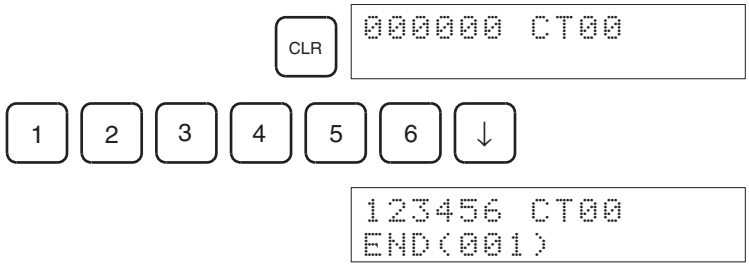

**Operation Example**

In the following instruction block, LD 000002 and OUT 000100 are added before the END(001) instruction.

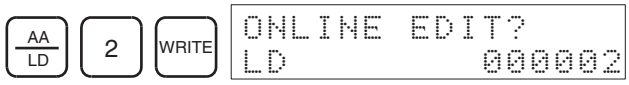
| Address | Instruction | Operand |
|---|---|---|
| : | : | : |
| 123456 | END | |

← LD 000002 and OUT 000100 are added.

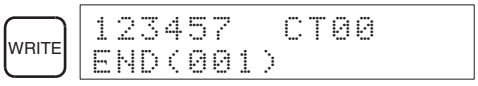
***1,2,3...*** 1. From the initial display, access the last program address.




2. Specify the instruction to be added and press the **WRITE** Key.



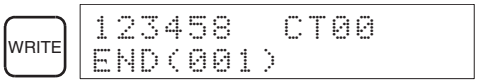

3. Press the **WRITE** Key once more.




4. Continue to specify the next instruction to be added and press the **WRITE** Key after each one.

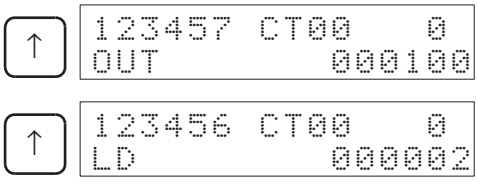



5. Press the **WRITE** Key once more.




After pressing the **WRITE** Key the second time, the altered program will be executed from the next scan.

6. Use the **Up** Key to check that the instruction has been added.



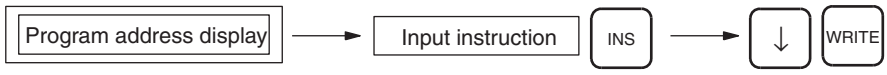

## 5-8-3 Instruction Insert

| RUN | MONITOR | PROGRAM |
|---|---|---|
| No | OK | See note. |

This operation is used when the CPU Unit is in MONITOR mode to insert instructions into the middle of a user program.

**Note** Refer to *4-3-2 Instruction Insert* for details on inserting instructions in PROGRAM mode.

**Key Sequence**



**Operation Example**

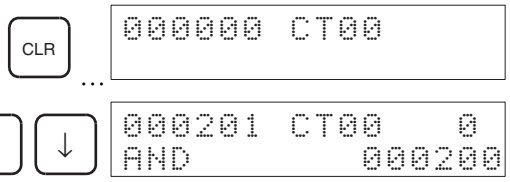
In the following instruction block, AND 000100 will be inserted in front of address 00201.

| Address | Instruction | Operand |
|---|---|---|
| : | : | : |
| 000200 | LD | 000001 |
| 000201 | AND | 000200 |
| : | : | : |

← AND 000100 will be inserted.

***1,2,3...***

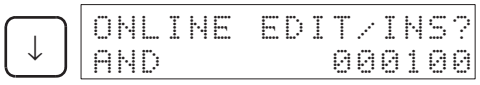
1. From the initial display, access the program where the address to be inserted.



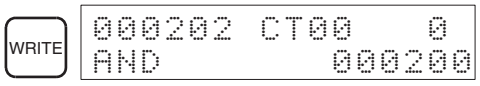

2. Specify the instruction to be inserted, and press the **INS** Key.




3. Press the **Down** Key. A confirmation message will be displayed, asking whether or not to execute the instruction insert using Online Edit.

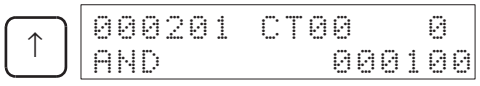



4. If the instruction requires more than one operand, specify the set values (operands), and press the **WRITE** Key after each.
5. To execute, press the **WRITE** Key.




After pressing the **WRITE** Key the second time, the altered program will be executed from the next scan.

6. Press the **Up** Key to confirm that the instruction has been inserted.



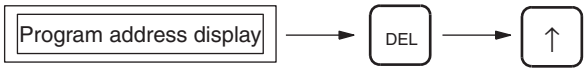

## 5-8-4 Instruction Delete

| RUN | MONITOR | PROGRAM |
|---|---|---|
| No | OK | See note. |

This operation is used when the CPU Unit is in MONITOR mode to delete instructions from the user program.

**Note** Refer to *4-3-3 Instruction Delete* for details on deleting instructions in PROGRAM mode.

**Key Sequence**



**Operation Example**

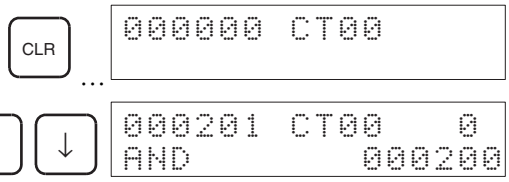
In the following instruction block, AND 000200 will be deleted from address 00201.

| Address | Instruction | Operand |
|---|---|---|
| : | : | : |
| 000200 | LD | 000001 |
| 000201 | AND | 000200 |
| 000202 | OUT | 000100 |
| : | : | : |

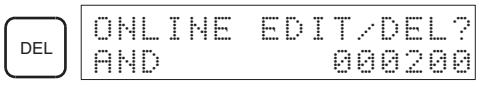
→ Delete

***1,2,3...***

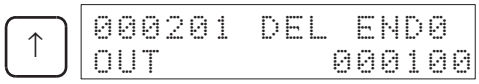
1. From the initial display, access the program address of the instruction to be deleted.




2. Delete the address.




3. To execute, press the **Up** Key. The next program address will be shifted forwards.




- After pressing the **Up** Key the second time, the altered program will be executed from the next scan.
- When a multiple-operand instruction is deleted, the set values (operands) will also be deleted.

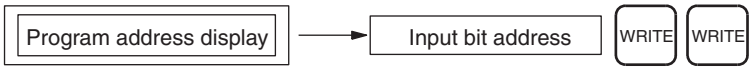
## 5-8-5 Bit Address Change

| RUN | MONITOR | PROGRAM |
|---|---|---|
| No | OK | See note. |

This operation is used when the CPU Unit is in MONITOR mode to change bit addresses used as operands in the user program.

**Note** Refer to *4-3-4 Bit Address Change* for details on changing bit addresses in PROGRAM mode.

**Key Sequence**



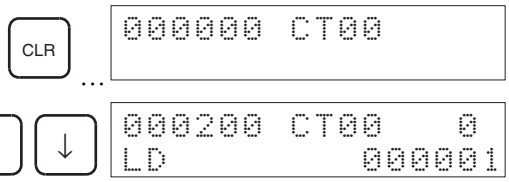

**Operation Example**

In the following instruction block, the bit address in the instruction at program address 00200 (LD 000001) will be changed to CIO 000002.

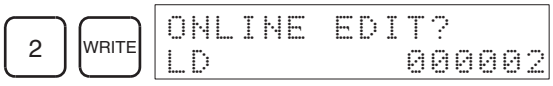
| Address | Instruction | Operand |
|---|---|---|
| : | : | : |
| 000200 | LD | 000001 |
| 000201 | AND | 000200 |
| : | : | : |

← The bit address will be changed to CIO 000002.

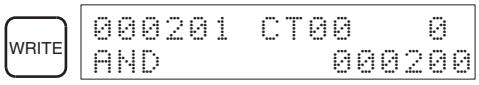
***1,2,3...*** 1. From the initial display, access the program address containing the bit address to be changed.




2. Specify the new bit address, and press the **WRITE** Key.

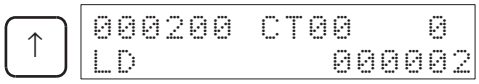



3. Press the **WRITE** Key once more.




After pressing the **WRITE** Key the second time, the altered program will be executed from the next scan.

4. Press the **Up** Key to check that the address of the bit operand has been changed.




## 5-8-6 N.O./N.C. Change

| RUN | MONITOR | PROGRAM |
|---|---|---|
| No | OK | See note. |

This operation is used when the CPU Unit is in MONITOR mode to change N.O. and N.C. conditions in the user program.

**Note** Refer to *4-3-5 N.O./N.C. Change* for details on changing bit addresses in PROGRAM mode.

**Key Sequence**

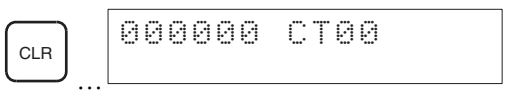


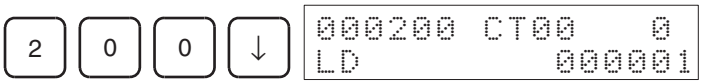

**Operation Example**

In the following instruction block, the instruction at program address 000200 is changed from LD 000001 to LD NOT 000001.

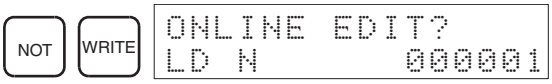
| Address | Instruction | Operand | |
|---|---|---|---|
| : | : | : | |
| 000200 | LD | 000001 | ← Changed to LD NOT 000001. |
| 000201 | AND | 000200 | |
| : | : | : | |

***1,2,3...*** 1. From the initial display, access the program address containing the instruction to be changed.

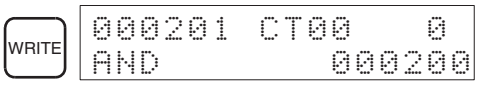





2. Press the **NOT** Key and then the **WRITE** Key.

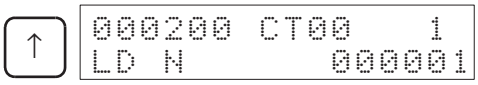



3. Press the **WRITE** Key once more.




After pressing the **WRITE** Key the second time, the altered program will be executed from the next scan.

4. Use the **Up** Key to check that LD has been changed to LD NOT.



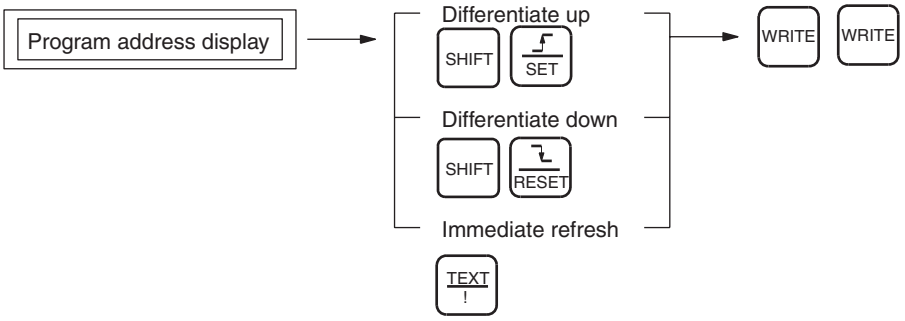

## 5-8-7 Instruction Variation Change

| RUN | MONITOR | PROGRAM |
|---|---|---|
| No | OK | See note. |

This operation is used when the CPU Unit is in MONITOR mode to change the variation of an instruction in the user program.

**Note** Refer to *4-3-6 Instruction Variation Change* for details on changing bit addresses in PROGRAM mode.

**Key Sequence**



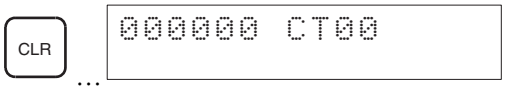

**Operation Example**

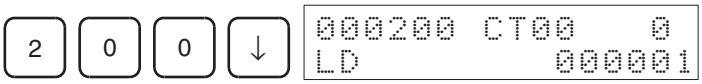
In the following instruction block, the instruction at address 000200 (LD 000001) will be changed to an upwardly differentiated LD instruction.

| Address | Instruction | Operand |
|---|---|---|
| : | : | : |
| 000200 | LD | 000001 |
| 000201 | AND | 000200 |
| : | : | : |

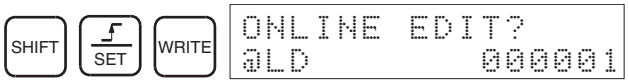
← Changed to upwardly differentiated @LD 000001.

***1,2,3...*** 1. From the initial display, access the program address containing the instruction to be changed.

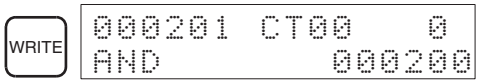





2. Specify the variation and press the **WRITE** Key.

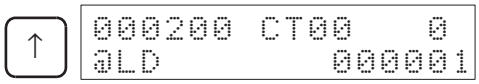



3. Press the **WRITE** Key once more.




After pressing the **WRITE** Key the second time, the altered program will be executed from the next scan.

4. Use the **Up** Key to check the variation has changed.




## 5-8-8 Timer/Counter SV Change

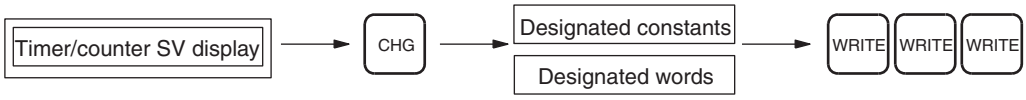
| RUN | MONITOR | PROGRAM |
|---|---|---|
| No | OK | See note. |

**Note** Refer to *4-3-7 Timer/Counter SV Changes* for details on changing timer/counter set values in PROGRAM mode.

### Timer/Counter SV Change 1: Constant or Word

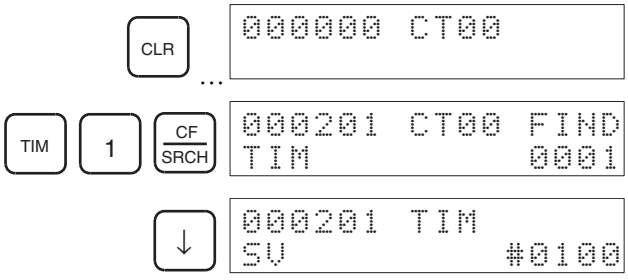
This operation is used to change the set value of a timer/counter in the user program.

**Key Sequence**

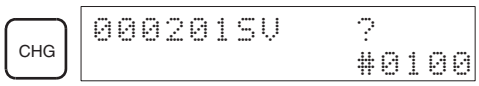



**Operation Example** The following example describes how to change the set value of TIM 0001.

***1,2,3...*** 1. From the initial display, access the timer/counter for which the set value will be changed.




2. Specify changing the set value from the above display.




3. The set value can be changed from the above display.

**Changing Constants**

Input the constant from the above display.




```
ONLINE EDIT?
SV          #0120
```




```
000202 CT00
LD          000100
```

**Changing Word Address**

Input the word address from the above display.




```
ONLINE EDIT?
SV            0010
```

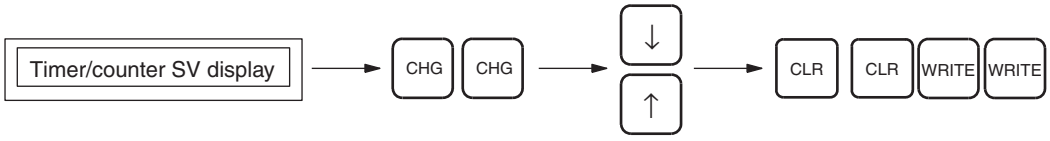



```
000202 CT00
LD          000100
```

## Timer/Counter SV Change 2: Fine Adjustment of Constant

This operation is used to increment or decrement the set value of a timer/counter with a constant specified for the set value in the user program.

**Key Sequence**




**Operation Example**

The following example describes how to adjust the constant specified as the set value for TIM 0001.

***1,2,3...***

1. From the initial display, access the timer/counter for which the set value will be adjusted.




```
000000 CT00
```




```
000201 CT00 FIND
TIM           0001
```



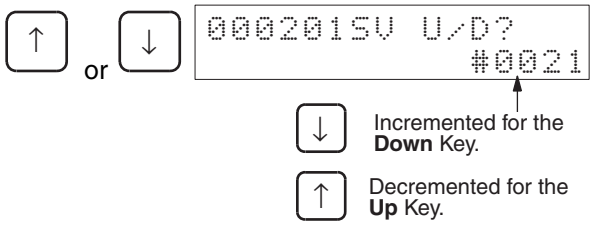

```
000201 TIM
SV          #0020
```

2. Specify changing the set values from the above display.

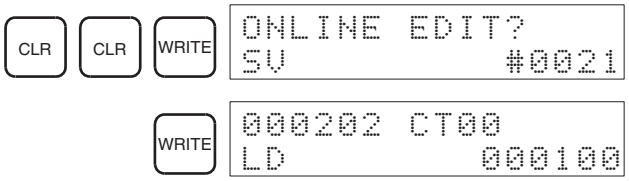



```
000201SV    U/D?
            #0020
```

3. Incremented or decremented the constant using the **Up** and **Down** Keys.




4. Exit the operation. The new set value will be displayed.



# SECTION 6
# Maintenance Operations

This section includes information on reading and setting the clock, reading cycle time, and reading and clearing error messages.


6-1 Clock Read/Change. . . . . 116
6-2 Cycle Time Read. . . . . 117
6-3 Reading/Clearing Error Messages . . . . . 118

# 6-1 Clock Read/Change

| RUN | MONITOR | PROGRAM |
|---|---|---|
| See note | OK | OK |

Note When the CPU Unit is in RUN mode, the clock can be read but cannot be changed.

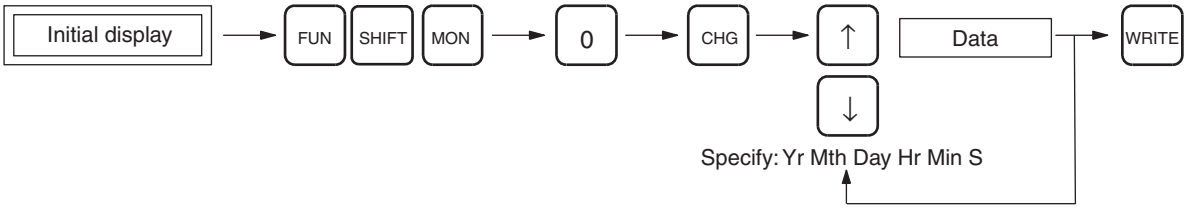
This operation is used to read or set the CPU Unit's internal clock.

- A battery is not installed in CS1-series PLCs when they are shipped. When the battery is installed in the CPU Unit, the CPU Unit clock will read as shown below. 00–01–00 = day, month, year; 00:00:00 = hours, minutes, seconds; SUN(0) = Sunday.

```
TIM      00-00-00
00:00:00  SUN(0)
```

- The CPU Unit's internal clock will start when the following Clock Read/Change operation is executed, when the time is read/changed from another Peripheral Device, when a FINS command is used to read/change the time, or when the DATE instruction is executed from the user program.
- The time data for the CPU Unit's internal clock is stored in the Calendar/Clock Area (A351 to 354) in the Auxiliary Area.

**Key Sequence**

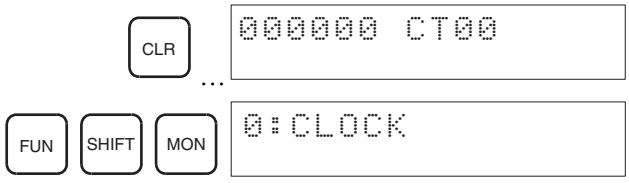



**Operation Example**

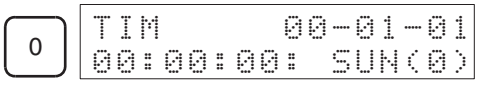
The following example shows how to set clock data for the first time.

***1,2,3...***

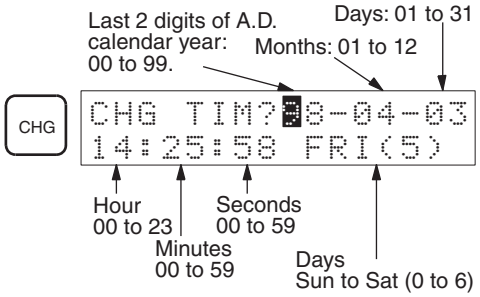
1. From the initial display, access the special function display.




2. Select ***0: TIM***.




3. To change the settings, press the **CHG** Key. When the **CHG** Key is pressed, the cursor will be displayed on the digit to be changed.



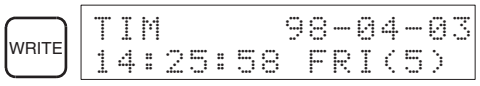

**Key Functions**

↓ :Moves the cursor to the right.

↑ :Moves the cursor to the left.

0 to 9 : Input numeric values.

4. Press the **WRITE** Key when the time and date have been set.



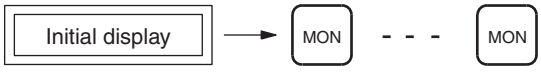

5. Press the **CLR** Key to exit.

# 6-2 Cycle Time Read

| RUN | MONITOR | PROGRAM |
|---|---|---|
| OK | OK | No |

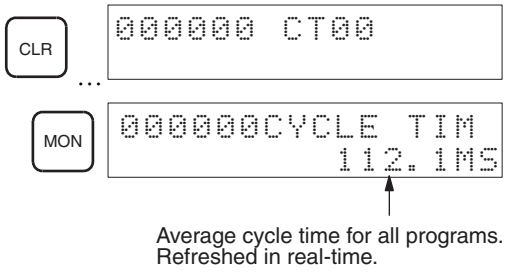
This operation is used to display the average cycle time for all programs (the total of all tasks) that are being executed.

**Key Sequence**




**Operation Example**

***1,2,3...*** 1. From the initial display, access the cycle time display.

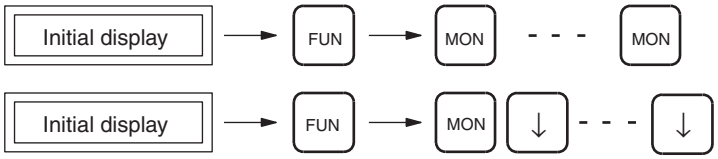


# 6-3 Reading/Clearing Error Messages

| RUN | MONITOR | PROGRAM |
|---|---|---|
| OK | OK | OK |

This operation is used to read fatal and non-fatal error messages, to read messages generated by MSG(046), and to reset error displays.

**Key Sequence**



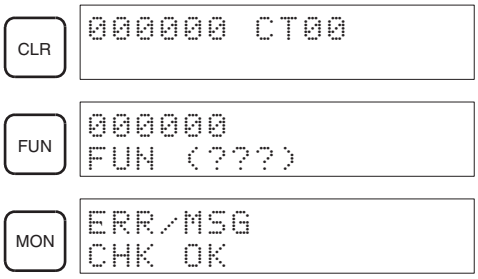

**Note**

1. Fatal operating errors can only be displayed and reset in PROGRAM mode.
2. If the source of the error is not removed, the error will not be reset and the error message will continue to be displayed.
3. When errors are read using the **Down** Key, the error will not be reset and current errors can be confirmed.

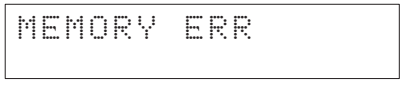
**Operation Example**

***1,2,3...***

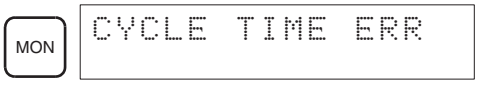
1. If there is no error, ***CHK OK*** will be displayed (and the error read operation will be finished).




2. If there is an error, the following will be displayed. The example below shows a memory error.




3. If there are several errors, the next most serious error will be displayed when the **MON** Key is pressed. (The order of display is given on the following pages.)

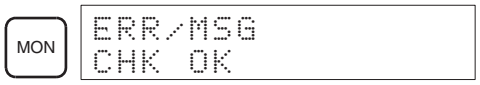



4. If the Message Display Instruction MSG(046) has been executed to record error messages, the recorded messages will be displayed when the **MON** Key is pressed. These messages will be displayed only after all other error messages have been displayed.

Note If the FAL instruction has being executed, messages recorded for the MSG(046) instruction will not be displayed until the cause of FAL execution has been removed.

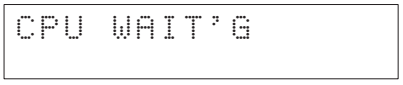



5. When all error messages and MSG(046) messages have been cleared, the following message will be displayed.




## Error Messages

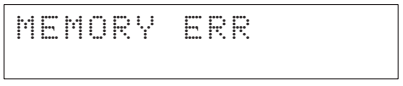
When CPU Unit operation is not possible, the following message will be displayed.



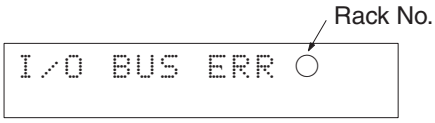

When several errors have occurred at the same time, the error messages will be displayed in order from 1 to 21, as listed below. (In order from the most serious to least serious.)

### Fatal Operating Errors

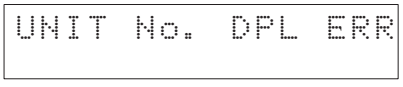
***1,2,3...*** 1. Memory Error

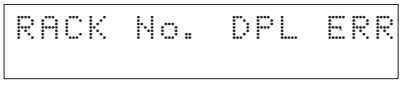



2. I/O Bus Error

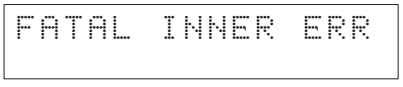



3. Unit Number Duplication Error

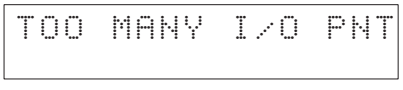


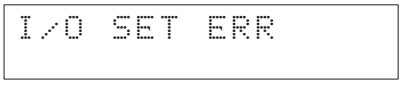

4. Fatal Inner Board Error



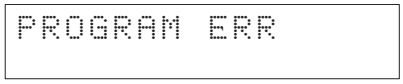

5. Too Many I/O Points



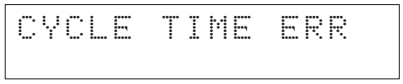

6. I/O Setting Error



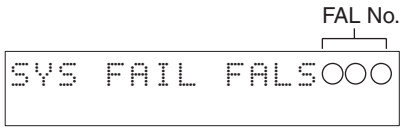
7. Program Error




8. Cycle Time Error

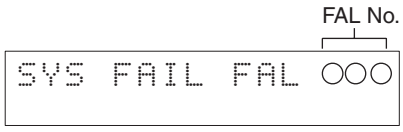



9. System Error (FALS)

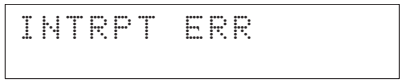



**Non-fatal Operating Errors**

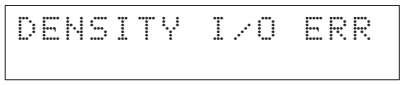
10. System Error (FAL)



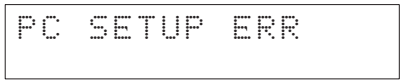

11. Interrupt Error



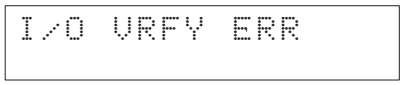

12. High-density I/O Error



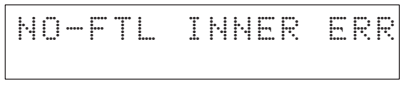

13. PLC Setup Error



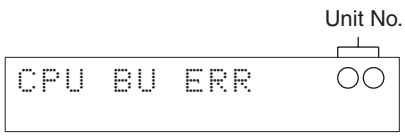

14. I/O Verification Error



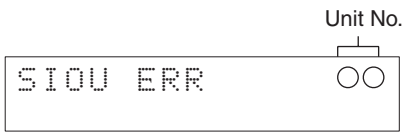

15. Non-fatal Inner Board Error



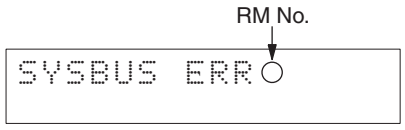

16. CS1 CPU Bus Unit Error



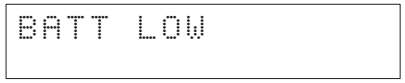

17. Special I/O Unit Error



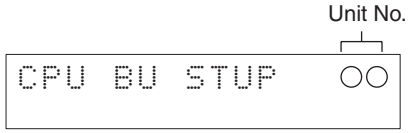
18. SYSMAC Bus Unit Error




19. Battery Error




20. CS1 CPU Bus Unit Setup Error




21. Special I/O Unit Setup Error

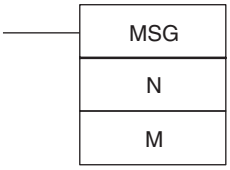



## MSG(046) Messages

The MSG(046) instruction can be used to record up to 8 messages for display on the Programming Console. These messages will be displayed when the **MON** Key is pressed after all other error messages have been displayed. If more than one MSG(046) instruction has been executed, the following methods can be used to display the other MSG(046) messages.

- Press the **MON** Key continuously. The currently displayed message will be cleared and the next message will be displayed.
- Press the **Down** Key continuously. The currently displayed message will not be cleared and the next message will be displayed.

### Using the MSG(046) Instruction




**Operands**

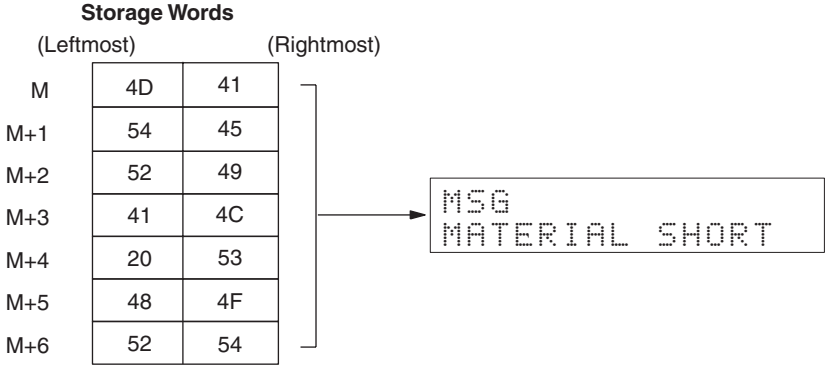
N: Message Number
0 to 7

M: 1st Message Word
The address of the first word containing the message to be displayed.

Specify a constant (0000 to FFFF Hex) to clear the message.

**MSG(046) Instruction**

The MSG(195) instruction reads sixteen words of extended ASCII contained in 16 words from M to M+15 and displays the message on a Peripheral Device (e.g., the Programming Console). The displayed message can be up to 32 characters long, i.e., each ASCII character code requires eight bits (two digits).




Note Data will be displayed in order from the leftmost byte to the rightmost byte on the Programming Console display.

**Note** Error messages will not be displayed while the error is occurring.

# SECTION 7
# Memory Card Operations

This section provides information on how to format Memory Cards before use, and procedures for transferring data between Memory Cards and the CPU Unit.


7-1 File Memory Operations . . . . . 124
7-2 Memory Card Format . . . . . 125
7-3 File Write . . . . . 126
7-4 File Read . . . . . 130
7-5 File Verify . . . . . 134
7-6 File Delete . . . . . 137

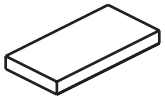
# 7-1 File Memory Operations

File memory operations read/write CPU Unit file memory (Memory Card or EM File Memory. It is possible to read/write the following types of data.

- All user programs
- I/O memory areas
- Parameter Areas (e.g., PLC Setup)

**Note** I/O name table and block comments cannot be saved to file memory.

### File Memory Types

| Memory type | Data recognized by CPU Unit | Memory capacity | Model |
|---|---|---|---|
| Memory Card (flash memory)  | 1. UM Area<br>2. I/O memory within specified ranges<br>3. Parameter Areas (e.g., PLC Setup)<br>4. I/O name table (See note)<br>5. Block comments (See note) | 8 Mbytes | HMC-EF861 |
| | | 15 Mbytes | HMC-EF171 |
| | | 30 Mbytes | HMC-EF371 |
| EM File Memory (RAM)<br>EM area<br>Bank 0<br>:<br>Bank n<br>:<br>Bank C<br>EM File Memory (From bank n onwards.) | 1. UM Area<br>2. I/O memory within specified ranges<br>3. Parameter Areas (e.g., PLC Setup)<br>4. I/O name table (See note)<br>5. Block comments (See note) | From the specified EM bank to the last bank (as specified in PLC Setup). | --- |

**Note** I/O name table files (extension .SBL) and block comment files (extension .RGL), cannot be handled from the Programming Console.

For details on how to specify the starting bank for EM area file memory, refer to *Section Section 8 PLC Setup Procedure*.

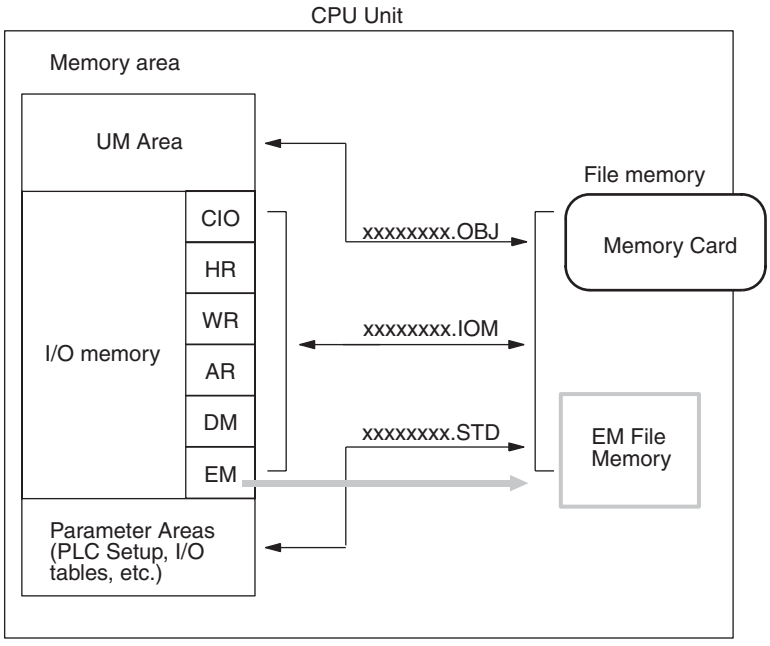
### Overview of Memory Card/EM File Memory Operations

| Operation | Details | |
|---|---|---|
| Memory Card Format | Formats Memory Cards or EM File Memory. | --- |
| File Write | Transfers files from the PLC to a Memory Card or EM File Memory. | See note. |
| File Read | Transfers files from a Memory Card or EM File Memory to the PLC. | |
| File Verify | Compares Memory Card or EM File Memory data with CPU Unit internal data. | |
| File Delete | Deletes the contents of Memory Card or EM File Memory data. | |

**Note** The memory areas that can be transferred or deleted are as follows:

| | | |
|---|---|---|
| UM Area | | |
| I/O memory | CIO Area | Specify the transfer start address and end address for each memory area.<br>**Note** One area = 1 file |
| | Work Area (WR) | |
| | Holding Bit Area (HR) | |
| | Auxiliary Bit Area (AR) | |
| | DM Area (DM) | |
| | EM Area (EM) | |
| Parameter Areas (e.g., PLC Setup) | | |

With CS1-series PLCs, Memory Cards and the specified range of EM File Memory can be used as file memory. Regardless of whether it's the UM Area, I/O memory or the Parameter Areas (e.g., PLC Setup), all data is saved as files.

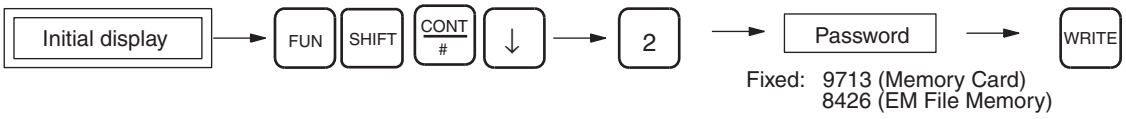



# 7-2 Memory Card Format

| RUN | MONITOR | PROGRAM |
|---|---|---|
| OK | OK | OK |

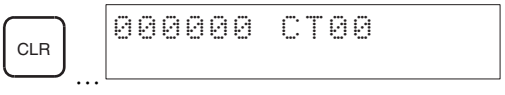
This operation is used to format Memory Cards or EM File Memory.

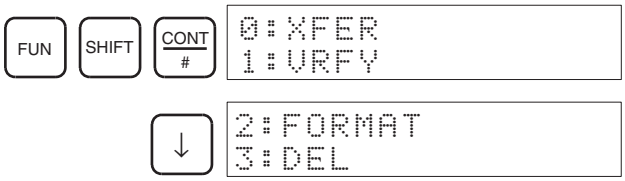
**Key Sequence**



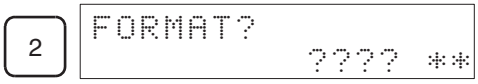

**Operation Example**

***1,2,3...*** 1. From the initial display, access the Memory Card operations menu.





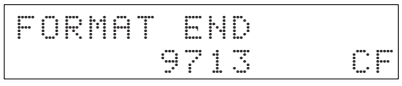

2. Select ***2: FORMAT***.




3. To format a Memory Card, enter the password.




Enter the password. (See note.)

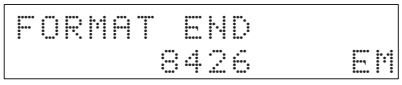



Note "9713" is a password set by the system for I/O Table Create and Memory Card Format. It cannot be changed.

4. To format EM File Memory, enter the password.




Enter the password. (See note.)



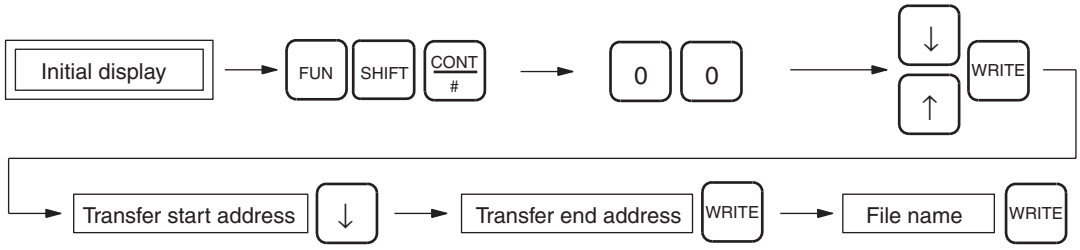

Note "8426" is a password set by the system for EM File Memory Format. It cannot be changed.

# 7-3 File Write

| RUN | MONITOR | PROGRAM |
|---|---|---|
| OK | OK | OK |

This operation transfers the UM Area, specified I/O memory area range, or PLC Setup data from the CPU Unit to a Memory Card or EM File Memory.

**Key Sequence**

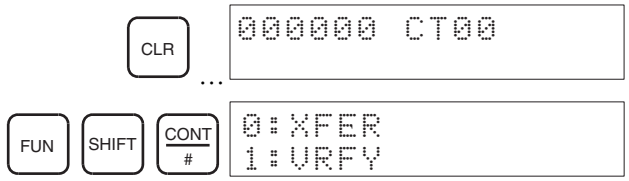


**Operation Example**

In the following example, CIO 2345 to CIO 2446 in the CIO Area of the CPU Unit are transferred to a Memory Card as the file “SAMPLE.IOM.”

**Note** For creation (and transfer to a Memory Card) of an AUTOEXEC file (a file that automatically transfers data when the power is turned ON), refer to page 130.

***1,2,3...*** 1. From the initial display, access the Memory Card operations menu.



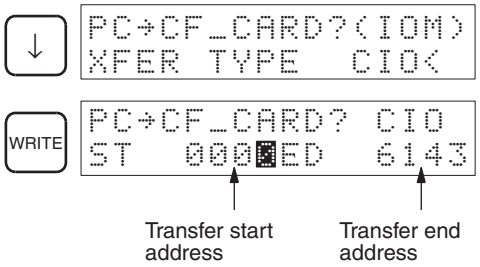

2. Select ***0:XFER*** to bring up the Memory Card operations menu.




3. Select ***0: PC → CF_CARD***.




4. Select CIO as the type of data to be transferred.




5. Use the **Up** and **Down** Keys to select the type of data to be transferred.

UM Area




CIO Area




Holding Bit Area




Work Bit Area



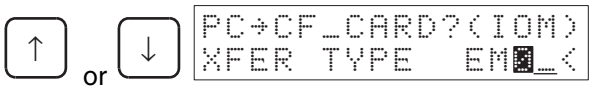

Auxiliary Bit Area



DM Area




EM Area




Parameter Areas (PLC Setup)




Note a) When saving I/O memory areas, each area (CIO Area, Holding Bit Area, Work Area, Auxiliary Area, DM Area or EM Area) is saved to a separate file (extension .IOM) within the specified word range.

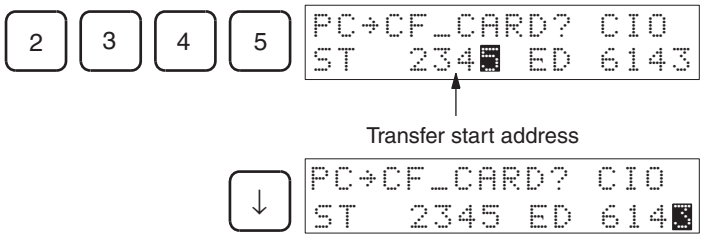
b) Always select ***DM<*** when creating a data file that is automatically transferred when the power is turned ON (AUTOEXEC.IOM).

c) If ***EM0_*** is displayed, input the number of the EM bank from 0 to C.

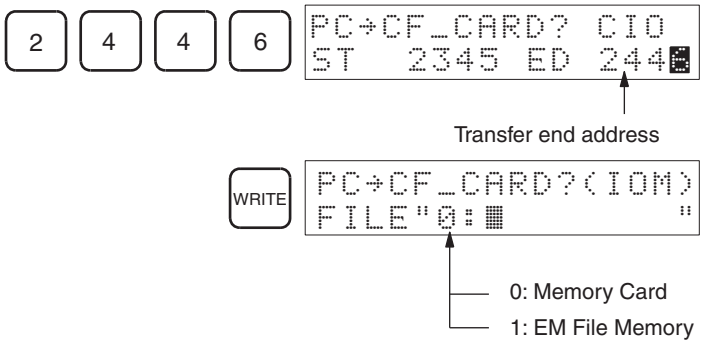
d) The Parameter Areas include the PLC Setup, registered I/O tables, routing tables, and CPU Bus Unit Setup.

6. When CIO, HR, WR, AR, DM, or EM□ data is being transferred, specify the transfer start word address and the transfer end word address.

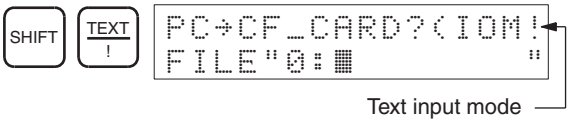
Transfer start address: CIO 2345



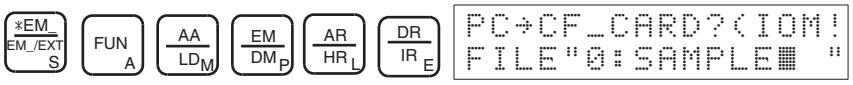

Transfer end address: CIO 2446




7. Switch to text input mode to enter the file name.



8. When saving data to a Memory Card, input the file name. When inputting the name from the Programming Console, A to Z and 0 to 9 are available. The file name may be a maximum of 8 characters. In the following example, the file name is SAMPLE.




It is not necessary to input the file extension (.OBJ for UM Area, .IOM for memory areas within I/O memory, and .STD for Parameter Areas). The extension will be automatically added.

| Transfer type | Extension |
|---|---|
| OBJ (UM area) | .OBJ |
| CIO, HR, AR, DM, EM□_ | .IOM |
| STD (PLC Setup) | .STD |

- Press the **Up** and **Down** Keys to move the cursor.
- Change the "0" in the file name to "1" to transfer the file to EM File Memory instead of to the Memory Card.
- Press the **CF/SRCH** Key to display the file names already on the Memory Card or EM File Memory. In this case steps 7, 8, and 9 are not necessary.




9. Return to normal input mode.



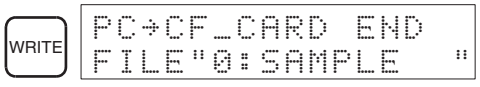

10. Execute the save by pressing the **WRITE** Key.




When a file with the same extension and file name already exists on the Memory Card, the following display will appear.




11. Use the following procedure to overwrite the existing file on the Memory Card.




- If you do not want to overwrite the existing file, press the **CLR** Key and input a different file name.
- If the volume label or directory name is specified by mistake, a message will ask if you want to overwrite. If this happens, press the **CLR** Key and input the proper file name.

### Precautions when Creating AUTOEXEC Files (Files that automatically transfer data when the power is turned ON.)

**1. The filename must be "AUTOEXEC."**

The following files can be created from the Programming Console and saved to a Memory Card. These files automatically transfer data when the power is turned ON. The filename however, must always be AUTOEXEC.

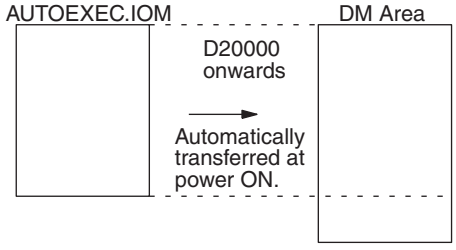
Program file: AUTOEXEC.OBJ
Parameter file: AUTOEXEC.STD
Data file: AUTOEXEC.IOM

**2. For AUTOEXEC.IOM, the transfer type must be "DM" and the start address must be 20000.**

When creating AUTOEXEC.IOM, Data Memory (DM) must be selected as the transfer type before transfer. Also, the start address must be set to 20000.

Operation: Select "DM<" as the transfer type and "20000" as the transfer start address.

When the power is turned ON, AUTOEXEC.IOM files will be read from the Memory Card and all data from the start of the file will be transferred to the Data Memory area, starting at D20000.




**Note** ***DM<*** must be selected as the transfer type.
**20000** must be selected as the start address.

**Note** When creating an AUTOEXEC.IOM file, be sure to set the start address to D20000.

It is possible to create AUTOEXEC.IOM files without setting the start address to D20000. The actual destination of data transferred from the Memory Card when the power is turned ON, however, will be the (fixed) address starting at D20000. Do not set the area type or the start address, therefore, to anything other than D20000.

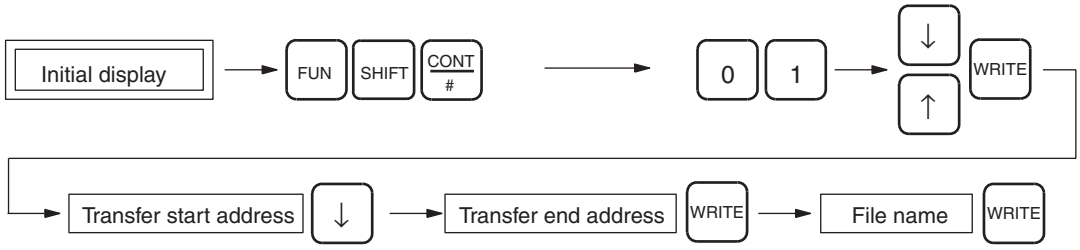
## 7-4 File Read

| RUN | MONITOR | PROGRAM |
|---|---|---|
| No | See note | OK |

This operation is used to transfer UM Area, specified I/O memory areas, or PLC Setup data from a Memory Card or EM File Memory to the CPU Unit.

⚠ **Caution** Always confirm safety before reading I/O memory data from a Memory Card even when working in PROGRAM mode. The CPU Unit will refresh I/O even in PROGRAM mode. If the status of a bit allocated to an Output Unit, Special I/O Unit, or CPU Bus Unit is changed, the load connected to the Unit may operate unexpectedly.

**Key Sequence**

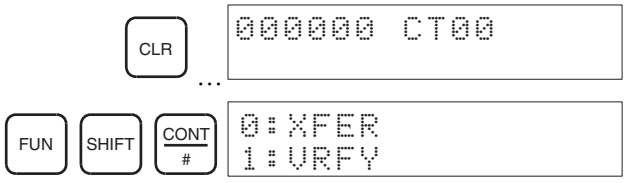



**Operation Example**

In the following example, a file on the Memory Card, SAMPLE.IOM, is transferred to CIO 2345 to CIO 2446 in the CPU Unit.

***1,2,3...***

1. From the initial display, access the Memory Card operations menu.



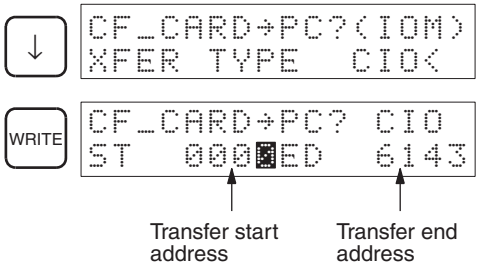

2. Select ***0: XFER*** from the Memory Card operations menu.




3. Select ***1: CF_CARD → PC***.




4. Select CIO as the type of data to be transferred.




5. Use the **Up** and **Down** Keys to select the type of data to be transferred.

UM Area




CIO Area




Holding Bit Area



Work Bit Area




Auxiliary Bit Area




DM Area




EM Area (See note)



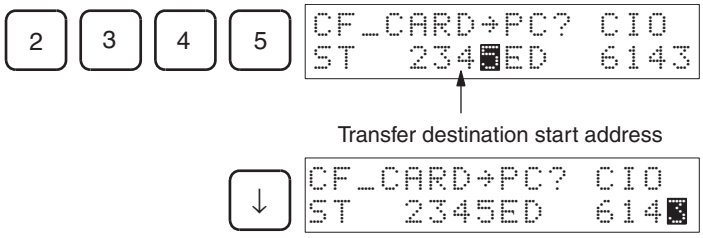

Parameter Areas



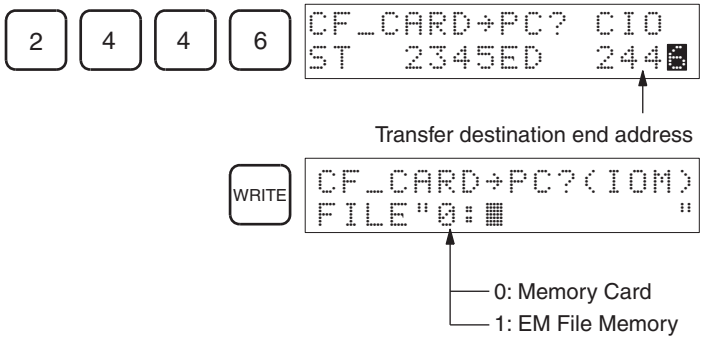

Note If ***EM0_*** is displayed, input the number of the EM bank from 0 to C.

6. When CIO, HR, WR, AR, DM, or EM□ data is being transferred, specify the transfer start word address and the transfer end word address. The start and end words are specified only when I/O memory is being transferred.

Transfer destination start address: CIO 2345



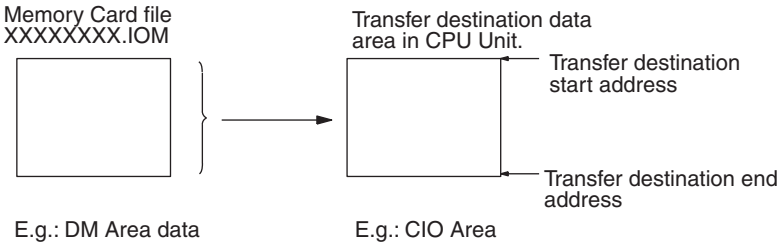

Transfer destination end address: CIO 2446




7. Switch to text input mode to enter the name of the file to transfer to the CPU Unit. (See note.)

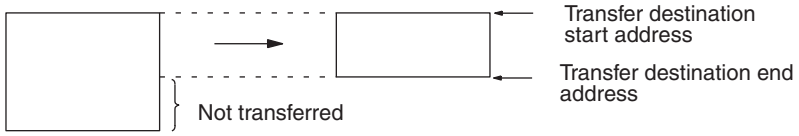


**Note** The data area to which data is being written does not necessarily have to be the same as the area from which data was originally read. The data will be transferred.

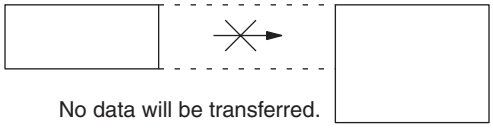



It is essential, however, that the number of words in the original transfer file be greater than or equal to the number of words in the transfer destination area, as shown below. Any words which exceed the transfer destination area will not be transferred.

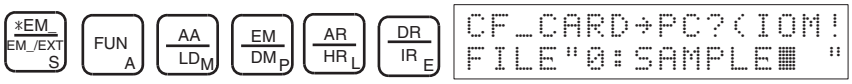
No. of words in Memory Card original transfer file ≥ No. of words in CPU Unit transfer destination area




If the number of words in the original transfer file is less than the number of words in the specified transfer destination area, no data will be transferred.




8. Enter the file name. In the following example, the file name is SAMPLE.

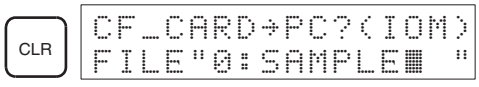



- Press the **Up** and **Down** Keys to move the cursor.
- Change the “0” in the file name to “1” to transfer the file from EM File Memory instead of the Memory Card.

Note Press the **CF/SRCH** Key to display the file names on the Memory Card or EM File Memory. If this function is used, steps 7., 8., and 9. are not necessary.




9. Return to normal input mode.

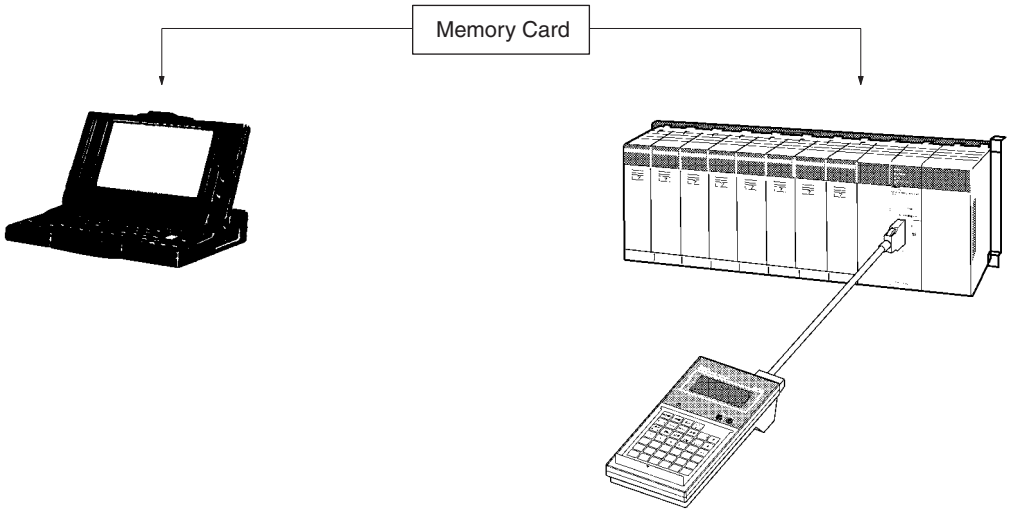



10. Execute the transfer by pressing the **WRITE** Key.



Reading Memory Card data using the Programming Console is useful, for example, to make changes to a program by taking only a Memory Card and Programming Console on-site.

1. Save the program file (XXXXXXXX.OBJ) using the CX-Programmer.
2. Take the Memory Card on-site and install it in the CPU Unit.
3. Connect the Programming Console, and transfer the program file from the Memory Card to the CPU Unit.

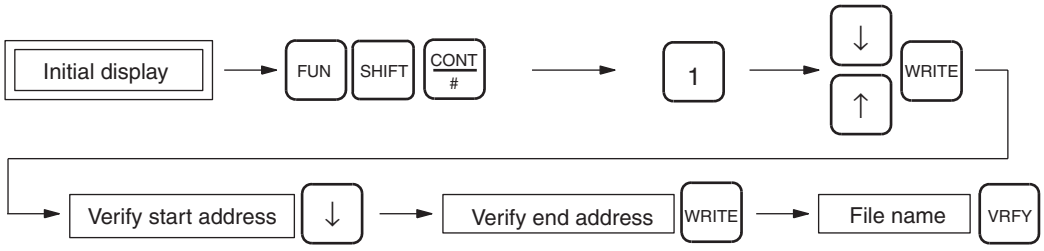



## 7-5 File Verify

| RUN | MONITOR | PROGRAM |
|---|---|---|
| OK | OK | OK |

This operation verifies data on a Memory Card or EM File Memory (UM Area, specified I/O memory areas, or PLC Setup) with internal CPU Unit data.

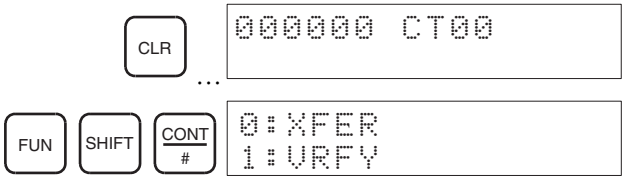
**Key Sequence**



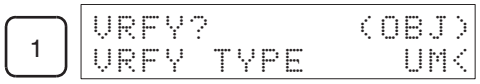

**Operation Example**

In the following example, CIO 2345 to CIO 2446 in the CPU Unit are verified against the Memory Card file “SAMPLE.IOM.”

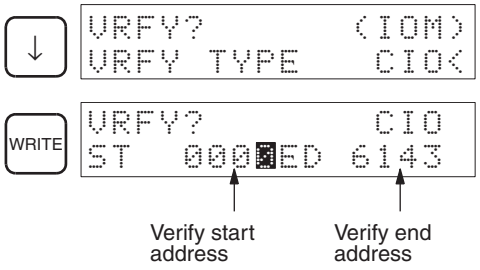
***1,2,3...*** 1. From the initial display, access the Memory Card operations menu.



2. From the Memory Card operations menu, select ***1: VRFY***.

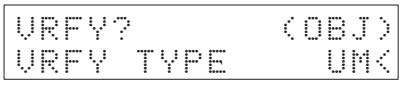



3. Specify CIO as the type of memory to be verified.

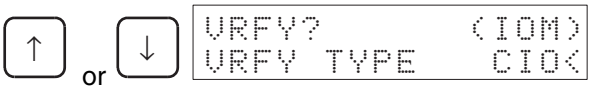



4. Use the **Up** and **Down** Keys to select the type of data to be verified.

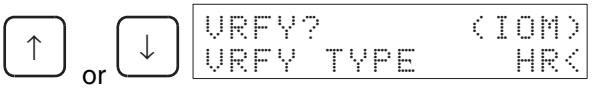
UM Area



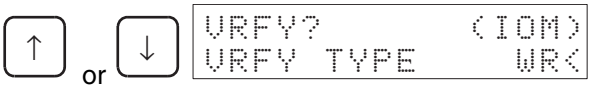

CIO Area



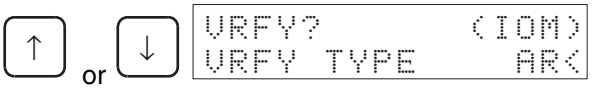

Holding Bit Area



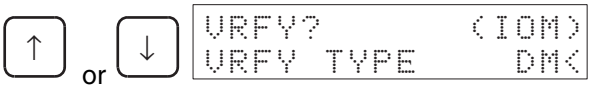

Work Bit Area




Auxiliary Bit Area



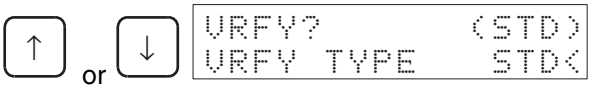

DM Area




EM Area




Parameter Areas

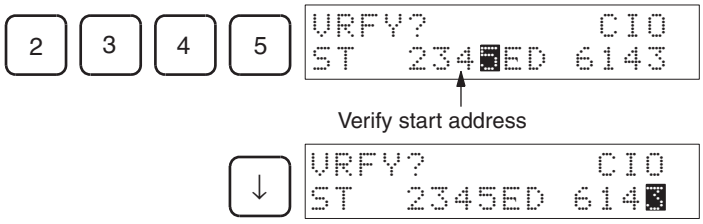



Note If ***EM0_*** is displayed, input the number of the EM bank from 0 to C.

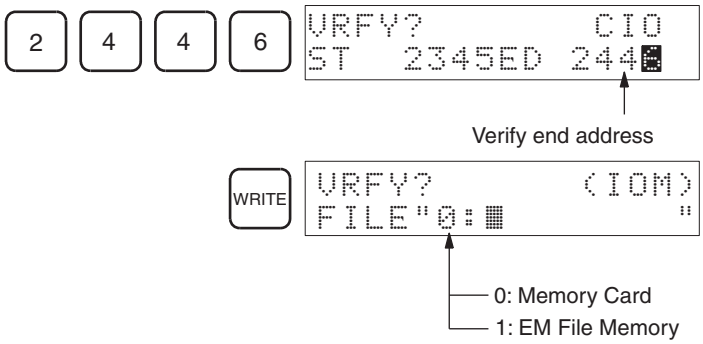
5. When CIO, HR, WR, AR, DM, or EM□ data is being transferred, specify the transfer start word address and the transfer end word address. The

start and end words are specified only when I/O memory is being transferred.

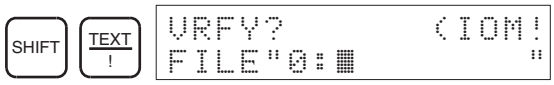
Verify start address: CIO 2345



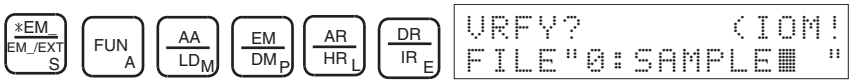

Verify end address: CIO 2446




6. Switch to text input mode to enter the file name.

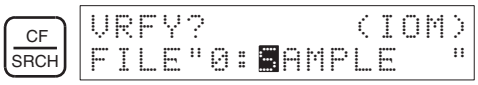



7. Enter the file name. In this example the file name is 0: SAMPLE.




- Press the **Up** and **Down** Keys to move the cursor.
- Change the "0" in the file name to "1" to verify the file to a file in EM File Memory instead of the Memory Card.

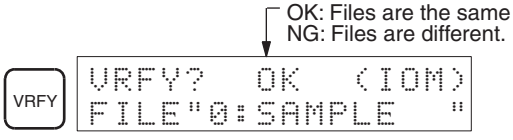
Note Press the **CF/SRCH** Key to display the file names on the Memory Card or EM File Memory. If this function is used, steps 6., 7., and 8. are not necessary.




8. Return to normal input mode.




9. Execute the verification by pressing the **VRFY** Key.

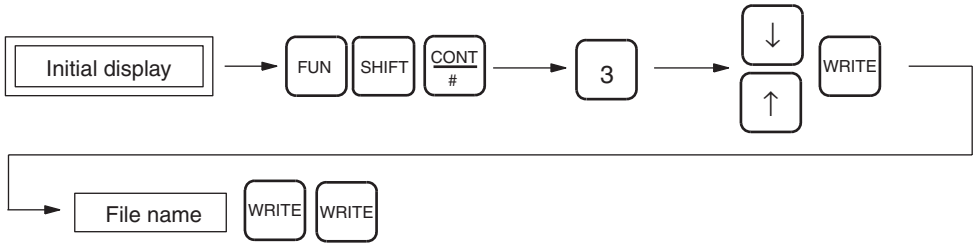


# 7-6 File Delete

| RUN | MONITOR | PROGRAM |
|---|---|---|
| OK | OK | OK |

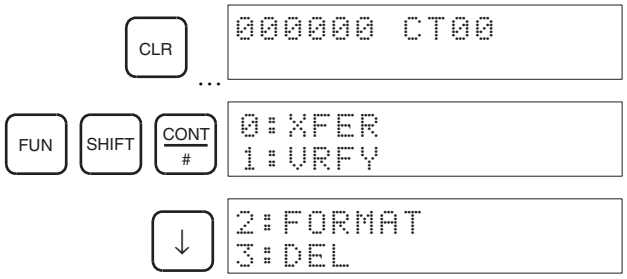
The operation deletes data from a Memory Card or EM File Memory (UM Area, specified I/O memory areas, PLC Setup).

**Key Sequence**

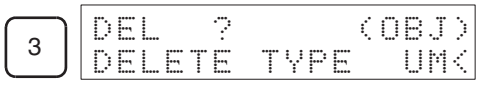



**Operation Example** In the following example, the Memory Card file “SAMPLE.IOM” is deleted.

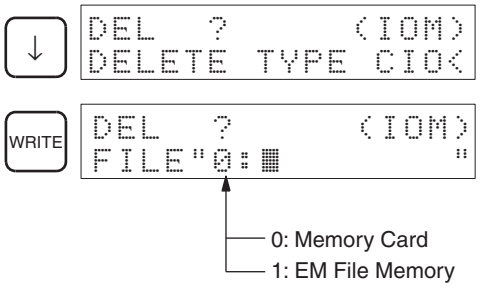
***1,2,3...*** 1. From the initial display, access the Memory Card operations menu.




2. From the Memory Card operations menu, select ***3:DEL***.

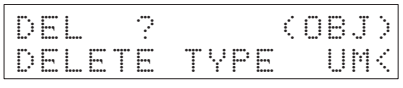



3. Select CIO as the type of data to be deleted.

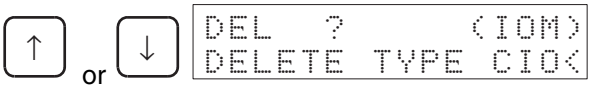



4. Use the **Up** and **Down** Keys to select the type of data to be deleted.

UM Area




CIO Area



Holding Bit Area

↑ or ↓

```
DEL  ?      (IOM)
DELETE TYPE  HR<
```

Work Bit Area

↑ or ↓

```
DEL  ?      (IOM)
DELETE TYPE  WR<
```

Auxiliary Bit Area

↑ or ↓

```
DEL  ?      (IOM)
DELETE TYPE  AR<
```

DM Area

↑ or ↓

```
DEL  ?      (IOM)
DELETE TYPE  DM<
```

EM Area

↑ or ↓

```
DEL  ?      (IOM)
DELETE TYPEEM0_<
```

Parameter Areas

↑ or ↓

```
DEL  ?      (STD)
DELETE TYPE STD<
```

Note If ***EM0_*** is displayed, input the number of the EM bank from 0 to C.

5. Switch to text input mode to enter the file name.

SHIFT, TEXT (!)

```
DEL  ?      (IOM!
FILE"0:▒           "
```

6. Enter the file name. (See note.)

*EM_ (EM_/EXT S), FUN (A), AA (LD M), EM (DM P), AR (HR L), DR (IR E)

```
DEL  ?      (IOM!
FILE"0:SAMPLE▒  "
```

- Press the **Up** and **Down** Keys to move the cursor.
- Change the “0” in the file name to “1” to delete a file from EM File Memory instead of the Memory Card.

Note Press the **CF/SRCH** Key to display the file names on the Memory Card or EM File Memory. If this function is used, steps 5., 6., and 7. are not necessary.

File name search:

CF (SRCH)

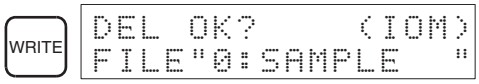
```
DEL  ?      (IOM)
FILE"0:▒AMPLE   "
```

7. Return to normal input mode.

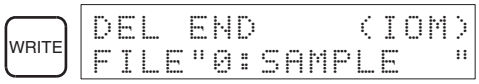
CLR

```
DEL  ?      (IOM)
FILE"0:SAMPLE▒  "
```

8. Confirm the file to be deleted.




9. Execute the deletion by pressing the **WRITE** Key.



# SECTION 8
# PLC Setup Procedure

This section lists the settings in the PLC Setup.

8-1 PLC Setup Procedure . . . . . 142

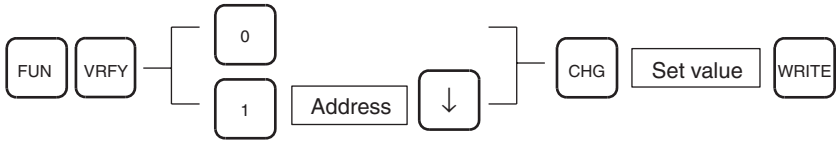
# 8-1 PLC Setup Procedure

| RUN | MONITOR | PROGRAM |
|---|---|---|
| See note | See note | OK |

**Note** When in RUN or MONITOR mode, contents of settings can be read, but setting cannot be performed.

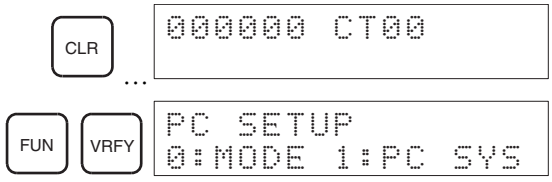
This operation is used to read and write data from and to the PLC Setup Area in the CPU Unit.

**Key Sequence**

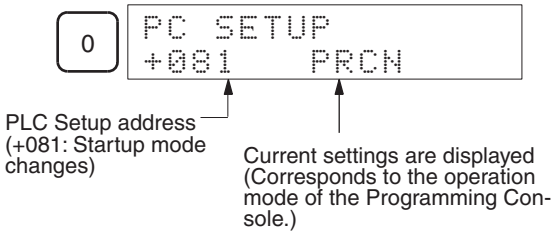



**Operation Example 1** In the following example, the Startup mode setting (+081) is changed.

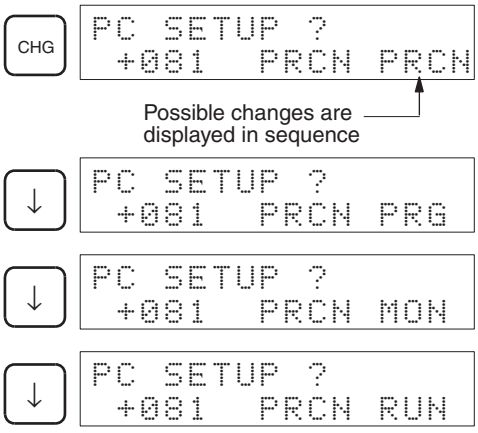
***1,2,3...*** 1. From the initial display, access the PLC Setup display.



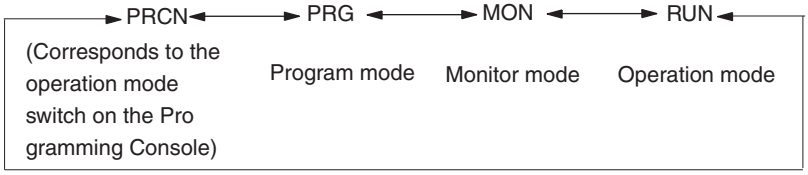

2. It is not necessary to input the Startup mode setting address, just select ***0:MODE***.




3. Make changes to the settings.



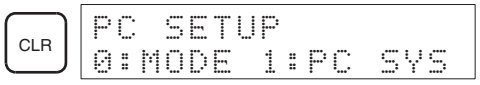
Press the **Up** or **Down** Keys to display each mode in sequence.







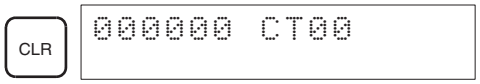
4. Press the **CLR** Key to return to the PLC Setup display.

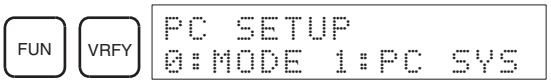



**Operation Example 2**

In the following example, the cycle time monitoring time is changed to 2,000 ms (2 s).

1. From the initial display, access the PLC Setup display.

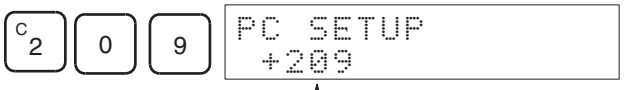



2. To make changes to all settings except the Startup mode, select ***1:PC SETUP***, specify the address (from + onwards), and read the set value.

   Example: Changing cycle time monitoring time




PC Setup address

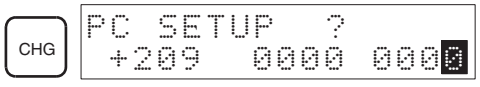
(+209: Cycle time monitoring time)



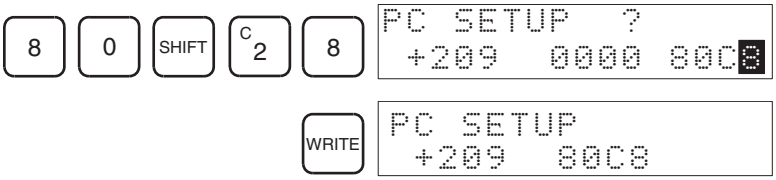

Set value

   When the set value is displayed, pressing the **Up** or **Down** Keys will display the previous or next address.

3. Use the following procedure to change the set word.



Example: Changing the value to 2,000 ms.




4. Press the **CLR** Key to return to the PLC Setup display.

# Appendix A
## ASCII Coding Sheet

The following list shows the range of characters that can be displayed on the Programming Console. Symbols and Japanese “katakana” characters however, cannot be input from the Programming Console.

| Right-most bit | Leftmost bit | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0, 1, 8, 9 | 2 | 3 | 4 | 5 | 6 | 7 | A | B | C | D | E | F |
| 0 | | | 0 | @ | P | ` | p | | ー | タ | ミ | α | p |
| 1 | | ! | 1 | A | Q | a | q | 。 | ア | チ | ム | ä | q |
| 2 | | " | 2 | B | R | b | r | 「 | イ | ツ | メ | ß | θ |
| 3 | | # | 3 | C | S | c | s | 」 | ウ | テ | モ | ε | ∞ |
| 4 | | $ | 4 | D | T | d | t | 、 | エ | ト | ヤ | μ | Ω |
| 5 | | % | 5 | E | U | e | u | ・ | オ | ナ | ユ | σ | ü |
| 6 | | & | 6 | F | V | f | v | ヲ | カ | ニ | ヨ | ρ | Σ |
| 7 | | ' | 7 | G | W | g | w | ァ | キ | ヌ | ラ | q | π |
| 8 | | ( | 8 | H | X | h | x | ィ | ク | ネ | リ | ﾌ | 𝑥̄ |
| 9 | | ) | 9 | I | Y | i | y | ゥ | ケ | ノ | ル | ⁻¹ | y |
| A | | * | : | J | Z | j | z | ェ | コ | ハ | レ | i | 千 |
| B | | + | ; | K | [ | k | { | ォ | サ | ヒ | ロ | ˣ | 万 |
| C | | , | < | L | ¥ | l | \| | ャ | シ | フ | ワ | ¢ | 円 |
| D | | - | = | M | ] | m | } | ュ | ス | ヘ | ン | £ | ÷ |
| E | | . | > | N | ^ | n | → | ョ | セ | ホ | ゛ | ñ | |
| F | | / | ? | O | _ | o | ← | ッ | ソ | マ | ゜ | ö | ▒ |

# Appendix B
## Error Messages

Error messages will be displayed on the Programming Console, whenever an operating error is made. Refer to the following table for operating errors and their remedies.

| Error message | Remedy |
|---|---|
| ***CHK MEM*** (ROM) | Pin 1 on the CPU Unit's DIP switch is set to ON (write protect). Reset this pin to OFF (write enabled). |
| ***PRGM OVER*** | The last address is not NOP, therefore no more of the program can be input. Clear the program after the END instruction. |
| ***ADR OVER*** | The set address exceeds the last address in the UM Area. Reset the address. |
| ***I/O No. ERR*** | I/O data has been input outside the permissible range. Reconfirm the input ranges for all instructions and input again correctly. |

Refer to *6-3 Reading/Clearing Error Messages* for information on displaying the error status of the PLC.

# Appendix C
## PLC Setup Coding Sheets for Programming Console

Use the following coding sheets when setting the PLC Setup from a Programming Console.

Address
10 □□□□

| Value (Hex) | Rack 0, Slot 0 I/O response time |
|---|---|
| 00 | 8 ms |
| 10 | No filter |
| 11 | 0.5 ms |
| 12 | 1 ms |
| 13 | 2 ms |
| 14 | 4 ms |
| 15 | 8 ms |
| 16 | 16 ms |
| 17 | 32 ms |

| Value (Hex) | Rack 0, Slot 0 I/O response time |
|---|---|
| 00 to 17 | As above |

Address
11 □□□□

| Value (Hex) | Rack 0, Slot 2 I/O response time |
|---|---|
| 00 to 17 | As above |

| Value (Hex) | Rack 0, Slot 3 I/O response time |
|---|---|
| 00 to 17 | As above |

Address
49 □□□□

| Value (Hex) | Rack 7, Slot 8 I/O response time |
|---|---|
| 00 to 17 | As above |

| Value (Hex) | Rack 7, Slot 9 I/O response time |
|---|---|
| 00 to 17 | As above |

Note For the CJ-series PLCs, address go only to address +29 (3 Racks only).

Address
80 □□□□

| Value (Hex) | IOM Hold Bit status at startup | Forced Status Hold Bit at startup |
|---|---|---|
| C000 | Retained | Retained |
| 8000 | Retained | Cleared |
| 4000 | Cleared | Retained |
| 0000 | Cleared | Cleared |

Address
81 □□□□

| Value | Startup mode |
|---|---|
| PRCN | Mode on Programming Console's mode switch |
| PRG | PROGRAM mode |
| MON | MONITOR mode |
| RUN | RUN mode |

**Note** PRCN, PRG, MON, and RUN can be selected using the **Up** and **Down** Keys.

Address
83 □□□□ (CS1D CPU Units for Single-CPU Systems and CS1-H/CJ1-H CPU Units only)

| Value (Hex) | RUN enable setting |
|---|---|
| 8000 | Enabled (Operation will start.) |
| 0000 | Disabled (Operation will not start.) |

**Note** These settings are not supported for CS1D Duplex-CPU Systems.

Address
84 □□□□ (CS1-H CPU Units only)

| Value (Hex) | Special Inner Board Setting |
|---|---|
| 8000 | Special Inner Board Setting enabled. |
| 0000 | Special Inner Board Setting disabled. |

Address
95□□□□ (CS1D CPU Units only)

| Value (Hex) | Primary/Secondary Duplex Communications Unit settings | |
|---|---|---|
| 0000 | Not duplex for unit numbers 0 to 15. | Bits 00 to 14 correspond to unit numbers 0 to 14. |
| 0001 | Unit number 0 set as primary Unit. | |
| to | | |
| 4000 | Unit number 14 set as primary Unit. | |

**Note 1** The secondary unit number is calculated by adding 1 to the primary unit number.
**Note 2** These settings are supported only for CS1D Duplex-CPU Systems and only for CSID CPU Unit Ver. 1.1 or later.

Address
96□□□□ (CS1D Duplex-CPU Systems only)

| Value (Hex) | Duplex transfer settings | |
|---|---|---|
| | Program transfer | EM Area transfer |
| 0000 | Transfer program | Transfer together |
| 8000 | Do not transfer program | Transfer together |
| 40□□ | Transfer program | Transfer over multiple scans **Note:** □□ is the division size (words). |
| C0□□ | Do not transfer program | Transfer over multiple scans **Note:** □□ is the division size (words). |

**Note** The above settings assume that both bit 11 (Inner Board parameter area) and bit 10 (Inner Board variable area) are set to 0.

Address
121□□□□ (CS1D CPU Units only)

| Value (Hex) | Active/Standby Duplex Communications Unit settings | |
|---|---|---|
| 0000 | Not duplex for unit numbers 0 to 15. | Bits 00 to 15 correspond to unit numbers 0 to 15. |
| 0001 | Duplex for unit number 0 only. | |
| to | | |
| 8000 | Duplex for unit number 15 only. | |

**Note** Up to three bits (unit numbers of duplex units) may be turned ON at the same time.

Address
122 □□□□ (CS1D CPU Units only)

| Value (Hex) | Multiple Unit online replacement |
|---|---|
| 0000 | Allow only one Unit to be replaced online at a time. |
| 8000 | Allow simultaneous online replacement of more than one Unit. |

Address
123□□□□ (CS1D Duplex-CPU Systems only)

| Value (Hex) | Operation during duplex initialization | Automatic recovery to duplex operation |
|---|---|---|
| 0000 | Do not run during initialization | Do not recover automatically |
| 4000 | Run during initialization | Do not recover automatically |
| 8000 | Do not run during initialization | Recover automatically |
| C000 | Run during initialization | Recover automatically |

Address
127 □□□□ (CS1D Duplex-CPU Systems only)

| Value (Hex) | Standby CPU Unit RS-232C port setting |
|---|---|
| 0000 | Disable use of RS-232C port on standby CPU Unit. |
| 5AA5 | Enable use of RS-232C port on standby CPU Unit. |

Address
128□□□□

| Value (Hex) | Detect Low Battery | Detect Interrupt Task Error |
|---|---|---|
| C000 | Do not detect. | Do not detect. |
| 8000 | Do not detect. | Detect. |
| 4000 | Detect. | Do not detect. |
| 0000 | Detect. | Detect. |

**Note** Interrupt task error detection cannot be set for CS1D Duplex-CPU Systems.

Address
129 □□□□ (CS1-H/CJ1-H/CS1D CPU Units only)

| Value (Hex) | Error History Storage of FAL |
|---|---|
| 8000 | Do not store FAL in error history. |
| 0000 | Store FAL in error history. |

Address
130 □□□□ (CS1D Duplex-CPU Systems only)

Not used.

| Value (Hex) | Duplex Memory Card settings |
|---|---|
| 0 | Disabled. |
| 8 | Duplex Memory Card check enabled. |

Address
136□□□□

| Value (Hex) | EM file memory settings |
|---|---|
| 0000 | None |
| 0080 | EM File Memory Enabled: Bank No. 0 |
| 0081 | EM File Memory Enabled: Bank No. 1 |
| to | |
| 008C | EM File Memory Enabled: Bank No. C |

Address
144□□□□

Peripheral port

| Value (Hex) | Data bits | Stop bits | Parity |
|---|---|---|---|
| 00 | 7 bits | 2 bits | Even |
| 01 | 7 bits | 2 bits | Odd |
| 02 | 7 bits | 2 bits | None |
| 04 | 7 bits | 1 bits | Even |
| 05 | 7 bits | 1 bits | Odd |
| 06 | 7 bits | 1 bits | None |
| 08 | 8 bits | 2 bits | Even |
| 09 | 8 bits | 2 bits | Odd |
| 0A | 8 bits | 2 bits | None |
| 0C | 8 bits | 1 bits | Even |
| 0D | 8 bits | 1 bits | Odd |
| 0E | 8 bits | 1 bits | None |

| Value (Hex) | Communications mode |
|---|---|
| 00 | Default (The above 3 columns are disabled) |
| 80 | Host link |
| 82 | NT link |
| 84 | Peripheral bus |
| 85 | Host link |

Address
145 □□□□

Peripheral port

| Value (Hex) | Baud rate |
|---|---|
| 0000 | 9,600 bps |
| 0001 | 300 bps |
| 0002 | 600 bps |
| 0003 | 1,200 bps |
| 0004 | 2,400 bps |
| 0005 | 4,800 bps |
| 0006 | 9,600 bps |
| 0007 | 19,200 bps |
| 0008 | 38,400 bps |
| 0009 | 57,600 bps |
| 000A | 115,200 bps |

Note For NT Links: 0000 to 0009 Hex = Normal NT Link, 000A = High-speed NT Link.

Address
147□□□□

Peripheral port

| Value (Hex) | Host link Unit No. |
|---|---|
| 0000 | No. 0 |
| 0001 | No. 1 |
| to | to |
| 001F | No. 31 |

Address
150□□□□

Peripheral port

| Value (Hex) | Maximum Unit No. in NT Link Mode |
|---|---|
| 0000 | No. 0 |
| 0001 | No. 1 |
| to | to |
| 0007 | No. 7 |

Address
160□□□□

RS-232C Port

| Value (Hex) | Data bits | Stop bits | Parity |
|---|---|---|---|
| 00 | 7 bits | 2 bits | Even |
| 01 | 7 bits | 2 bits | Odd |
| 02 | 7 bits | 2 bits | None |
| 04 | 7 bits | 1 bits | Even |
| 05 | 7 bits | 1 bits | Odd |
| 06 | 7 bits | 1 bits | None |
| 08 | 8 bits | 2 bits | Even |
| 09 | 8 bits | 2 bits | Odd |
| 0A | 8 bits | 2 bits | None |
| 0C | 8 bits | 1 bits | Even |
| 0D | 8 bits | 1 bits | Odd |
| 0E | 8 bits | 1 bits | None |

| Value (Hex) | Communications mode |
|---|---|
| 00 | Default (The above 3 columns are disabled) |
| 80 | Host link |
| 82 | NT link |
| 83 | No-protocol |
| 84 | Peripheral bus |
| 85 | Host link |

Address
161□□□□

RS-232C port

| Value (Hex) | Baud rate |
|---|---|
| 0000 | 9,600 bps |
| 0001 | 300 bps |
| 0002 | 600 bps |
| 0003 | 1,200 bps |
| 0004 | 2,400 bps |
| 0005 | 4,800 bps |
| 0006 | 9,600 bps |
| 0007 | 19,200 bps |
| 0008 | 38,400 bps |
| 0009 | 57,600 bps |
| 000A | 115,200 bps |

Note For NT Links: 0000 to 0009 Hex = Normal NT Link, 000A = High-speed NT Link.

Address
162□□□□

RS-232C port

| Value (Hex) | No-protocol mode delay |
|---|---|
| 0000 | 0 ms |
| 0001 | 10 ms |
| to | to |
| 270F | 99,990 ms |

Address
163□□□□

RS-232C port

| Value (Hex) | Host link Unit No. |
|---|---|
| 0000 | No. 0 |
| 0001 | No. 1 |
| 0002 | No. 2 |
| to | to |
| 001F | No. 31 |

Address
164□□□□

RS-232C Port

| Value (Hex) | No-protocol Mode end code |
|---|---|
| 00 | 00 |
| to | to |
| FF | FF |

| Value (Hex) | No-protocol Mode end code |
|---|---|
| 00 | 00 |
| to | to |
| FF | FF |

Address
165□□□□

RS-232C Port

| Value (Hex) | No-protocol Mode reception data volume |
|---|---|
| 00 | 256 |
| 01 | 1 |
| to | to |
| FF | 256 |

| Value (Hex) | No-protocol Mode end code setting |
|---|---|
| 0 | None (Specify the amount of data being received) |
| 1 | Yes (Specify the end code) |
| 2 | End code is set to CR+LF |

| Value (Hex) | No-protocol Mode start code setting |
|---|---|
| 0 | None |
| 1 | Yes |

Address
166□□□□

RS-232C port

| Value (Hex) | Maximum Unit No. in NT Link Mode |
|---|---|
| 0000 | No. 0 |
| 0001 | No. 1 |
| to | to |
| 0007 | No. 7 |

Address
195□□□□

| Value (Hex) | Scheduled interrupt time units |
|---|---|
| 0000 | 10 ms |
| 0001 | 1.0 ms |

**Note** These settings are not supported for CS1D Duplex-CPU Systems.

Address
197□□□□

| Value (Hex) | Instruction error operation |
|---|---|
| 0000 | Continue operation |
| 8000 | Stop operation |

Address
198□□□□ (CS1-H/CJ1-H CPU Units only)

| Value (Hex) | Com Port for background processing |
|---|---|
| 0 to 7 | Communications port 0 to 7 (logic port) |

| Value (Hex) | Background processes | | |
|---|---|---|---|
| | Table data instructions | Text string instructions | Data shifting instructions |
| E00 | Yes | Yes | Yes |
| C00 | Yes | Yes | No |
| A00 | Yes | No | Yes |
| 800 | Yes | No | No |
| 600 | No | Yes | Yes |
| 400 | No | Yes | No |
| 100 | No | No | Yes |
| 000 | No | No | No |

**Note** These settings are not supported for CS1D Duplex-CPU Systems.

Address
208□□□□

| Value (Hex) | Minimum cycle time |
|---|---|
| 0000 | Cycle time not fixed |
| 0001 | Cycle time fixed: 1 ms |
| to | |
| 7D00 | Cycle time fixed: 32,000 ms |

Address
209□□□□

| Value (Hex) | Watch cycle time |
|---|---|
| 0000 | Default: 1,000 ms (1 s) |
| 0001 | 10 ms |
| to | |
| 0FA0 | 40,000 ms |

Address
218□□□□

| Value (Hex) | Fixed peripheral servicing time |
|---|---|
| 0000 | Default (4% of the cycle time) |
| 8000 | 00 ms |
| 8001 | 0.1 ms |
| to | to |
| 80FF | 25.5 ms |

Address
219□□□□

Time Slice Peripheral Servicing Time

| Value (Hex) | Setting |
|---|---|
| 00 | Do not use Peripheral Servicing Priority Mode. |
| 01 to FF | Time Slice Peripheral Servicing Time<br>(0.1 to 25.5 ms in 0.1-ms increments) |

Time Slice Instruction Execution Time

| Value (Hex) | Setting |
|---|---|
| 00 | Do not use Peripheral Servicing Priority Mode. |
| 05 to FF | Time Slice Instruction Execution Time<br>(5 to 255 ms in 1-ms increments) |

Parallel Processing Mode (See note.)

| Value (Hex) | Setting |
|---|---|
| 00 | Do not use Parallel Processing Mode. |
| 01 | Synchronous memory access |
| 02 | Asynchronous memory access |

**Note** These settings are not supported for CS1D Duplex-CPU Systems.

Address
220□□□□

Special Peripheral Servicing Unit Numbers

| Value (Hex) | Setting |
|---|---|
| 00 | No special servicing |
| 10 to 1F | CPU Bus Units 0 to 15 (unit number + 10 Hex) |
| 20 to 7F | Special I/O Units 0 to 95 (unit number + 20 Hex) |
| E1 | Inner Board |
| FC | RS-232C port |
| FD | Peripheral port |

| Value (Hex) | Setting |
|---|---|
| 00 | No special servicing |
| 10 to 1F | CPU Bus Units 0 to 15 (unit number + 10 Hex) |
| 20 to 7F | Special I/O Units 0 to 95 (unit number + 20 Hex) |
| E1 | Inner Board |
| FC | RS-232C port |
| FD | Peripheral port |

**Note** These settings are not supported for CS1D Duplex-CPU Systems.

Address
221□□□□

Special Peripheral Servicing Unit Numbers

| Value (Hex) | Setting |
|---|---|
| 00 | No special servicing |
| 10 to 1F | CPU Bus Units 0 to 15 (unit number + 10 Hex) |
| 20 to 7F | Special I/O Units 0 to 95 (unit number + 20 Hex) |
| E1 | Inner Board |
| FC | RS-232C port |
| FD | Peripheral port |

| Value (Hex) | Setting |
|---|---|
| 00 | No special servicing |
| 10 to 1F | CPU Bus Units 0 to 15 (unit number + 10 Hex) |
| 20 to 7F | Special I/O Units 0 to 95 (unit number + 20 Hex) |
| E1 | Inner Board |
| FC | RS-232C port |
| FD | Peripheral port |

Address
222□□□□

Not used.

Special Peripheral Servicing Unit Numbers

| Value (Hex) | Setting |
|---|---|
| 00 | No special servicing |
| 10 to 1F | CPU Bus Units 0 to 15 (unit number + 10 Hex) |
| 20 to 7F | Special I/O Units 0 to 95 (unit number + 20 Hex) |
| E1 | Inner Board |
| FC | RS-232C port |
| FD | Peripheral port |

Address
225□□□□

| Value (Hex) | Power-OFF interrupt task | Power-OFF detection delay time |
|---|---|---|
| 0000 | Disabled | 0 ms |
| 0001 | | 1 ms |
| to | | to |
| 000A | | 10 ms |
| 8000 | Enabled | 0 ms |
| 8001 | | 1 ms |
| to | | to |
| 800A | | 10 ms |

**Note** These settings are not supported for CS1D Duplex-CPU Systems.

Address
226□□□□

| Value (Hex) | Special I/O Unit cyclic refreshing 0: Yes 1: No | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Unit number | | | | | | | | | | | | | | | |
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0001 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0002 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0003 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0004 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0005 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| to | | | | | | | | | | | | | | | | |
| FFFF | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Note Address +227 through to +231 are the same as +226

Address
448□□□□

FINS write protection over network and excluded nodes

| Value (Hex) | Nodes excluded from write protection |
|---|---|
| 00 to 20 | Sets the number of nodes to be excluded from write protection set in Programming Console addresses 449 to 480. The values 00 to 20 Hex correspond to the number of nodes (0 to 32). |

| Value (Hex) | FINS write protection over network |
|---|---|
| 80 | Enabled |
| 00 | Disabled |

**Note** These settings are supported only for CS-series CPU Unit Ver. 2.0 or later.




Nodes Excluded from Write Protection (32 Nodes Max.)

| Value (Hex) | FINS command source node address |
|---|---|
| 01 to FE | Node address of node excluded from write protection |
| FF | All nodes in specified network |

| Value (Hex) | FINS command source network address |
|---|---|
| 00 to 7F | Network address of node excluded from write protection. |

**Note** These settings are supported only for CS-series CPU Unit Ver. 2.0 or later.

# Index

## A–B


All Clear, 39
ASCII coding sheet, 145
AUTOEXEC
  precautions, 130
bit address change, 74
bit addresses
  for inputting basic instructions, 56
bit search, 69
  using bit addresses, 70
buzzer operation, 42
buzzer volume, 9


## C


CJ Series
  definition, xvii
clock
  read/change, 116
connection, 14, 19
  panel mounting, 15
CS Series
  definition, xvii
CX-Programmer, 3
cyclic tasks, 44


## D


DIP switch
  settings, 18
display
  double-word, 97, 99
  hexadecimal or BCD, 96
  signed decimal, 97
  unsigned decimal, 98
display contrast, 9
display format, 90


## E


editing, 71
  bit address change, 74
  instruction change, 71
  instruction delete, 73
  instruction insert, 72
  instruction variation change, 75
  N.O./N.C. change, 74
  online, 104
  Timer/Counter SV changes, 76
error messages, 147, 149
  fatal operating errors, 119
  non fatal operating errors, 120
  reading/clearing, 118
errors
  verification errors, 49


## F


force set/reset
  clear, 103
  continuous, 101
  temporary, 102
functions
  All Clear, 5
  buzzer operation, 5, 42
  cycle time read, 5
  I/O Table Create, 46
  I/O Table Verify, 48
  instruction variation change, 5
  Memory Clear, 5
  monitoring, 6
  Program Edit, 5
  Program Read, 5
  Program Write, 5
  Programming Console, 5
  reading/clearing error messages, 5
  selecting tasks, 5, 43
  special functions, 6
  Table Create, 5
  Table Verify, 5
  Timer/Counter SV change, 5
  unsupported, 7


## I–J


I/O bit status monitor, 80
I/O Table Create, 46
I/O table operations, 46
I/O Table Verify, 48
index register addresses
  for inputting basic instructions, 57
initial display, 32, 33
inputting instructions
  basic instructions, 55
  special instructions, 62

standard special instructions, 62
symbol comparison instructions, 63
Timer/Counter, 59
installation, 1
instruction change, 71
instruction delete, 73
instruction insert, 72
instruction search, 68
instruction variation
change, 75
instruction variations, 59
interrupt tasks, 45
Japanese messages
change to English, 18


## K–L


key input modes, 36
Normal Input, 36
Shift Input, 36
Text Input, 36
keys
functions, 11, 36
keypad layout, 37
layout, 13
names, 11
ladder programming, 5


## M


maintenance, 115
cycle time read, 117
memory areas
All Clear, 37, 39
clearing memory, 37
Memory Clear, 37, 41
specifying areas, 38
memory cards
file delete, 137
file memory operations, 124
file read, 130
file verify, 134
file write, 126
formatting, 125
memory types, 124
operations, 123
Memory Clear, 41
mode switch, 9, 35
operating modes, 10
models
C200H and CQM1, 2
monitoring
ASCII monitor, 94
bit addresses, 82
clearing displays, 89
designated words, 84
differential monitor, 103
display format, 90
I/O multipoint monitor, 87
I/O multipoint monitor results, 88
operands at specified program addresses, 86
signed decimal monitor, 92
simple I/O monitor, 81
specified bits, 81
Timer/Counter completion flags, 83
unsigned decimal monitor, 93
word addresses, 85
word monitor, 91
MSG(046), 121


## N–O


N.O./N.C. change, 74
online editing, 104
bit address change, 108
instruction add, 105
instruction change, 104
instruction delete, 107
instruction insert, 106
instruction variation change, 110
N.O./N.C. change, 109
Timer/Counter SV change, 111
operation
changing operating modes, 35
initial display, 32, 33
starting, 32


## P


panel mounting, 15
password, 32, 33
PLC Setup
coding sheets, 149
PLC setup
procedure, 142
setting sheet, 149
precautions

general, xxiii
handling, 7
operating environment, 7
previous models
comparison with, 16
programming, 18
checking program operation, 26
inputting basic instructions, 55
inputting instructions, 54
inputting programs, 22
one cyclic task, 3
writing and editing programs, 53
Programming Console
connection, 19
features, 3
functions, 5
operation, 31
setup, 17
Programming Consoles
PLC Setup coding sheets, 149
programs
bit search, 69
checking and adjusting, 79
editing, 71
instruction search, 68
reading and searching, 67
reading designated addresses, 67
reading execution status, 80
PVs
changing from signed decimal monitor, 96
changing from the ASCII monitor, 99
changing from the word monitor, 100
changing from unsigned decimal monitor, 98

## S

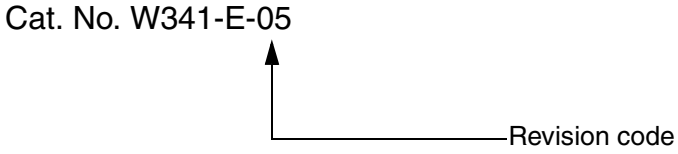
selecting tasks, 43
simple I/O monitor, 81
standard special instructions
inputting, 62
symbol comparison instructions, 63
bit addresses, 64
constants, 66
indirect DM addresses, 65
indirect EM addresses, 65
instruction variations, 67

## T–W

tasks
cyclic tasks, 44
interrupt tasks, 44
Timer/Counter set values, 60
change, 76
constants, 60
indirect DM addresses, 61
indirect EM addresses, 61
words, 60
word PVs
changing, 95

# Revision History

A manual revision code appears as a suffix to the catalog number on the front cover of the manual.




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 |
|---|---|---|
| 01 | March 1999 | Original production |
| 02 | May 2001 | Information on CJ-series PCs, the CQM1H-PRO01-E Programming Console, and PLC Setup addresses 219 to 222 added. |
| 03 | September 2001 | Information on new CPU Units.<br>**Page ix:** Addition to introduction at top of page and model numbers added to manual names.<br>**Page 2:** Model numbers added to bottom table and information added after table.<br>**Pages 9, 15, 18, 20, 32 to 36, and 42:** “CS Series” changed to “CS1” and “CJ Series” changed to “CJ1/CJ1-H/CS1-H.”<br>**Page 46:** Note added toward top of page and section and note added to bottom.<br>**Pages 138 to 145:** Deleted.<br>**Pages 148 to 154:** Moved to *Appendix C.*<br>**Pages 148, 150, and 151:** Note added.<br>**Pages 149 and 152:** Information for new addresses added.<br>**Page 151:** Information for following addresses moved to correct order: 163, 166. |
| 04 | November 2003 | The abbreviation for Programmable Logic Controller was changed from PC to PLC throughout the manual. Information on CS1D Duplex-CPU Systems and CPU Units for Duplex-CPU Systems was added, including the following changes.<br>**Page ix:** Changed introduction at top of page, added model numbers to manual names, and added extra manuals.<br>**Page 2:** Changed information on applicable CPU Units.<br>**Page 8:** Changed diagrams to show Key Sheets in English.<br>**Pages 145 to 152:** Removed “+” symbol from all addresses, added new address settings, and added notes indicating whether settings are supported in Duplex-CPU Systems. |
| 05 | October 2005 | **Page v:** Information on general precautions notation added.<br>**Page xix:** Information on liability and warranty added. |

**OMRON Corporation**
**Control Devices Division H.Q.**
Shiokoji Horikawa, Shimogyo-ku,
Kyoto, 600-8530 Japan
Tel: (81)75-344-7109/Fax: (81)75-344-7149

**Regional Headquarters**

**OMRON EUROPE B.V.**
Wegalaan 67-69, NL-2132 JD Hoofddorp
The Netherlands
Tel: (31)2356-81-300/Fax: (31)2356-81-388

**OMRON ELECTRONICS LLC**
1 East Commerce Drive, Schaumburg, IL 60173
U.S.A.
Tel: (1)847-843-7900/Fax: (1)847-843-8568

**OMRON ASIA PACIFIC PTE. LTD.**
83 Clemenceau Avenue,
#11-01, UE Square,
Singapore 239920
Tel: (65)6835-3011/Fax: (65)6835-2711

**OMRON (CHINA) CO., LTD.**
Room 2211, Bank of China Tower,
200 Yin Cheng Zhong Road,
PuDong New Area, Shanghai, 200120 China
Tel: (86)21-5037-2222/Fax: (86)21-5037-2200

OMRON

**Authorized Distributor:**

Cat. No. W341-E1-05 Note: Specifications subject to change without notice Printed in Japan

This manual is printed on 100% recycled paper.

**Artisan Technology Group** is an independent supplier of quality pre-owned equipment

## Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

## We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

## Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP