## SYSMAC CP Series CP1L-L_D CPIL-M D

## CP1L CPU Unit

## OPERATION MANUAL

OInROn

# CP1L-L10D $\square-\square$ <br> CP1L-L14D■-■ <br> CP1L-L20D $\square-\square$ <br> CP1L-M30D $\square-\square$ <br> CP1L-M40D $-\square$ <br> CP1L-M60D $\square-\square$ <br> CP1L CPU Unit <br> <br> Operation Manual 

 <br> <br> Operation Manual}

Revised May 2010

## 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.

Indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury. Additionally, there may be severe property damage.

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 CX-Programmer 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, 2007
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 CP-series CPU Units

## Unit Versions

## Notation of Unit Versions on Products

A "unit version" has been introduced to manage CPU Units in the CP Series according to differences in functionality accompanying Unit upgrades.

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.


CX-Programmer version 7.3 or higher can be used to confirm the unit version of the CP1L CPU Unit with 10 I/O points.
CX-Programmer version 7.1 or higher can be used to confirm the unit version of the CP1L CPU Unit with 14, 20, 30, 40 or 60 I/O points.

Note CX-Programmer version 7.2 or lower cannot be used to confirm unit versions for CP1L CPU Units with 10 I/O points.
CX-Programmer version 7.0 or lower cannot be used to confirm unit versions for CP1L CPU Units with 14, 20, 30, 40 or 60 I/O points.

## - Confirmation Procedure

## Procedure When the Device Type and CPU Type Are Known

1,2,3... 1. Set the Device Type Field in the Change PLC Dialog Box to CP1L.

2. Click the Settings Button by the Device Type Field and, when the Device Type Settings Dialog Box is displayed, set the CPU Type Field to J, L, L10 or $M$.

Device Tgpe Settings [CP1L]
Gener:l|

-Expansion Memory Hone $\quad \square$ Read Orily


```
Timer / Clock
    F Installed
```

ake Iefault

3. Go online and select PLC - Edit - Information


The PLC Information Dialog Box will be displayed.

| PLC Information - NewPLC1 |  |  |  |
| :---: | :---: | :---: | :---: |
| Project PLC type: | CP1LM |  | Close |
| - Actual Characteristics |  |  |  |
| Type: | CP1LM |  |  |
| Unit Ver.: | 1.0 |  | Unit version |
| Program memory: | 11264 | Steps |  |
| Useable: | 10646 | Steps |  |
| Protected: | No |  |  |
| Memory type: | - |  |  |
| File/memory card: | No |  |  |
| Data memory: | 32768 | Words |  |
| Extension: | 0 | KWords |  |
| EM banks: | 0 |  |  |
| Bank size: | . | Words |  |
| 10 memory: | 11.5 | KWords |  |
| Timer/counters: | 8 | KWords |  |
| Manufacturing Details |  |  |  |
| Revision | A |  |  |
| PCB Revision | AA. |  |  |
| Software Revision | AA 0 |  |  |
| Lot Number | 070323 |  |  |
| Manufacturing | - |  |  |
| Serial Number | - |  |  |

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

## Procedure When the Device Type and CPU Type Are Not Known

This procedure is possible only when connected directly to the CPU Unit with a serial connection.
If you don't know the device type and CPU type that 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.
The PLC Information Dialog Box will be displayed and can be used to confirm the unit version of the CPU Unit.


## Using the Unit Version Labels

The following unit version labels are provided with the CPU Unit．
Ver． 1.0
Ver． 1.0
バージョンアップによるユニット
の搭載機能の差異を管理するため
のラベルです。
必要に応じて，製品の前面に貼り
付けてご使用ください。
These Labels can be used
to manage differences
in the available
functions among the Units．
Place the appropriate Iabel
onthe front ofthe Unit to
show what Unit
versionis actually being
used．

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

## Functions Supported by Unit Version for CP－series CPU Units

## Functions Supported by Unit Version 1.0 and 1.1

Functionality is the same as that for CS／CJ－series CPU Units with unit version 3．0．The functionality added for CS／CJ－series CPU Unit unit version 4.0 is not supported．

## CP1H CPU Units

－CX－Programmer version 6.11 or higher is required to use CP1H－X $\square \square \square \square$－ $\square / X A \square \square \square \square-\square$ with unit version 1.1 or 1.0 ．
－CX－Programmer version 6.20 or higher is required to use CP1H－Y $\square \square \square \square$－ $\square$ with unit version 1．1．

| CPU Unit |  | CP1H CPU Unit |  |  |
| :---: | :---: | :---: | :---: | :---: |
| Model |  | CP1H－ CP1H－XA （See note 1．） |  | CP1H－Y $\qquad$ （See note 2．） |
| Function | Unit version | Ver． 1.1 or later | Ver． 1.0 | Ver． 1.1 |
| Pulse outputs | Allocated built－ in I／O terminals | 4 axes at 100 kHz | 2 axes at 100 kHz <br> 2 axes at 30 kHz | 2 axes at 100 kHz |
|  | Special pulse output terminals | None |  | 2 axes at 1 MHz |

Note 1．The unit version for the CP1H－X $\square \square \square \square-\square / X A \square \square \square \square-\square$ begins at 1．0．
2．The unit version for the $\mathrm{CP} 1 \mathrm{H}-\mathrm{Y} \square \square \square-\square$ begins at 1．1．
3．CX－Programmer version 7.11 or higher is required to use CP1L CPU Units with unit version 1．0．
4．CX－Programmer version 7.3 or higher is required to use CP1L CPU Units with 10 I／O points．

## TABLE OF CONTENTS

PRECAUTIONS ..... xxiii
1 Intended Audience ..... xxiv
2 General Precautions ..... xxiv
3 Safety Precautions ..... xxiv
4 Operating Environment Precautions ..... xxvi
5 Application Precautions ..... xxvii
6 Conformance to EC Directives ..... xxx
SECTION 1
Features and System Configuration ..... 1
1-1 Features and Main Functions ..... 2
1-2 System Configuration ..... 15
1-3 Connecting the CX-Programmer ..... 23
1-4 Function Charts ..... 33
1-5 Function Blocks ..... 34
SECTION 2
Nomenclature and Specifications ..... 37
2-1 Part Names and Functions ..... 38
2-2 Specifications ..... 43
2-3 CP1L CPU Unit Operation ..... 73
2-4 CPU Unit Operation ..... 81
2-5 CPU Unit Operating Modes. ..... 85
2-6 Power OFF Operation ..... 87
2-7 Computing the Cycle Time ..... 89
SECTION 3 Installation and Wiring ..... 99
3-1 Fail-safe Circuits ..... 100
3-2 Installation Precautions ..... 101
3-3 Mounting ..... 103
3-4 Wiring CP1L CPU Units ..... 110
3-5 Wiring CPU Unit I/O ..... 118
3-6 CP/CPM1A-series Expansion I/O Unit Wiring ..... 130
SECTION 4
I/O Memory Allocation ..... 137
4-1 Overview of I/O Memory Area ..... 138
4-2 I/O Area and I/O Allocations ..... 145
4-3 1:1 Link Area ..... 152
4-4 Serial PLC Link Area ..... 153

## TABLE OF CONTENTS

4-5 Internal Work Area ..... 153
4-6 Holding Area (H) ..... 154
4-7 Auxiliary Area (A) ..... 155
4-8 TR (Temporary Relay) Area ..... 155
4-9 Timers and Counters ..... 156
4-10 Data Memory Area (D) ..... 160
4-11 Index Registers ..... 161
4-12 Data Registers ..... 169
4-13 Task Flags ..... 171
4-14 Condition Flags ..... 171
4-15 Clock Pulses ..... 173
SECTION 5
Pulse and Counter Functions. ..... 175
5-1 High-speed Counters ..... 176
5-2 Pulse Outputs ..... 197
5-3 Inverter Positioning ..... 280
SECTION 6
Advanced Functions ..... 337
6-1 Interrupt Functions ..... 338
6-2 Quick-response Inputs ..... 362
6-3 Serial Communications ..... 366
6-4 Analog Adjuster and External Analog Setting Input ..... 394
6-5 Battery-free Operation ..... 395
6-6 Memory Cassette Functions ..... 397
6-7 Program Protection ..... 405
6-8 Failure Diagnosis Functions ..... 414
6-9 Clock ..... 418
SECTION 7
Using Expansion Units and Expansion I/O Units ..... 421
7-1 Connecting Expansion Units and Expansion I/O Units. ..... 422
7-2 Analog Input Units ..... 423
7-3 Analog Output Units ..... 432
7-4 Analog I/O Units ..... 440
7-5 Temperature Sensor Units ..... 463
7-6 CompoBus/S I/O Link Units ..... 478
7-7 DeviceNet I/O Link Units ..... 484

## TABLE OF CONTENTS

SECTION 8
LCD Option Board ..... 491
8-1 Features ..... 492
8-2 Specifications ..... 493
8-3 Part Names ..... 494
8-4 Installation and Removing ..... 495
8-5 Basic Operation ..... 496
8-6 LCD Option Board Function ..... 501
8-7 Trouble Shooting ..... 550
SECTION 9
Ethernet Option Board ..... 553
9-1 Ethernet Option Board Function Guide ..... 554
9-2 Features ..... 557
9-3 System Configuration ..... 558
9-4 Specifications ..... 559
9-5 FINS Communications ..... 560
9-6 Part Names ..... 562
9-7 Comparison with Previous Models ..... 563
9-8 Installation and Initial Setup ..... 564
9-9 Memory Allocations ..... 571
9-10 Web Browser Setup and Display ..... 577
9-11 Trouble Shooting ..... 588
9-12 Sample Application ..... 592
SECTION 10
Program Transfer, Trial Operation, and Debugging ..... 597
10-1 Program Transfer ..... 598
10-2 Trial Operation and Debugging ..... 598
SECTION 11
Troubleshooting ..... 605
11-1 Error Classification and Confirmation ..... 606
11-2 Troubleshooting ..... 608
11-3 Error Log ..... 617
11-4 Troubleshooting Unit Errors ..... 618
SECTION 12
Inspection and Maintenance ..... 621
12-1 Inspections ..... 622
12-2 Replacing User-serviceable Parts ..... 624

## TABLE OF CONTENTS

Appendices ..... 629
A Standard Models ..... 629
B Dimensions Diagrams ..... 635
C Auxiliary Area Allocations by Function ..... 643
D Auxiliary Area Allocations by Address ..... 665
E Memory Map ..... 713
F Connections to Serial Communications Option Boards ..... 715
G PLC Setup ..... 741
H Specifications for External Power Supply Expansion ..... 765
Index. ..... 767
Revision History ..... 773

## About this Manual:

This manual describes installation and operation of the CP-series Programmable Controllers (PLCs) and includes the sections described below. The CP Series provides advanced package-type PLCs based on OMRON's advanced control technologies and vast experience in automated control.
Please read this manual carefully and be sure you understand the information provided before attempting to install or operate a CP-series PLC. Be sure to read the precautions provided in the following section.

## Definition of the CP Series

The CP Series is centered around the CP1H and CP1L CPU Units and is designed with the same basic architecture as the CS and CJ Series. Always use CP-series Expansion Units and CP-series Expansion I/O Units when expanding I/O capacity.
I/O words are allocated in the same way as the CPM1A/CPM2A PLCs, i.e., using fixed areas for inputs and outputs.


Precautions provides general precautions for using the Programmable Controller and related devices.
Section 1 introduces the features of the CP1L and describes its configuration. It also describes the Units that are available and connection methods for Programming Devices and other peripheral devices.

Section 2 describes the names and functions of CP1L parts and provides CP1L specifications.
Section 3 describes how to install and wire the CP1L.
Section 4 describes the structure and functions of the I/O Memory Areas and Parameter Areas.
Section 5 describes the CP1L's interrupt and high-speed counter functions.
Section 6 describes all of the advanced functions of the CP1L that can be used to achieve specific application needs.

Section 7 describes how to use CP-series Expansion Units and Expansion I/O Units.
Section 8 gives an outline of the LCD Option Board, explains how to install and remove the LCD Option Board, and describes the functions including how to monitor and make settings for the PLC. It also lists the errors during operation and provides probable causes and countermeasures for troubleshooting.
Section 9 gives an outline of the Ethernet Option Board, explains how to install and remove the Ethernet Option Board, and how to monitor and make settings required for operation. It also lists the errors during operation and provides countermeasures for troubleshooting.
Section 10 describes the processes used to transfer the program to the CPU Unit and the functions that can be used to test and debug the program.
Section 11 provides information on hardware and software errors that occur during CP1L operation.
Section 12 provides inspection and maintenance information.
Appendices provide product lists, dimensions, tables of Auxiliary Area allocations, and a memory map.

## Related Manuals

The following manuals are used for the CP1L CPU Units. Refer to these manuals as required.

| Cat. No. | Model numbers | Manual name | Description |
| :---: | :---: | :---: | :---: |
| W462 | $\begin{aligned} & \text { CP1L-L10D } \square-[ \\ & \text { CP1L-L14D } \square- \\ & \text { CP1L-L20D } \square- \\ & \text { CP1L-M30D } \square- \\ & \text { CP1L-M40D } \\ & \text { CP1L-M60D } \end{aligned}$ | SYSMAC CP Series CP1L CPU Unit Operation Manual | Provides the following information on the CP Series: <br> - Overview, design, installation, maintenance, and other basic specifications <br> - Features <br> - System configuration <br> - Mounting and wiring <br> - I/O memory allocation <br> - Troubleshooting <br> Use this manual together with the CP1L Programmable Controllers Programming Manual (W451). |
| W451 | CP1H-X40D $\square-\square$ <br> CP1H-XA40D <br> CP1H-Y20DT-D <br> CP1L-L10D $\square-\square$ <br> CP1L-L14D $\square-\square$ <br> CP1L-L20DD- $\square$ <br> CP1L-M30D $\square-\square$ $\square$ <br> CP1L-M40D $\square-\square$ <br> CP1L-M60D $\square-\square$ $\square$ | SYSMAC CP Series CP1H /CP1L CPU Unit Programming Manual | Provides the following information on programming the CP Series: <br> - Programming methods <br> - Tasks <br> - Programming instructions |
| W461 | $\begin{aligned} & \text { CP1L-L10D } \square-\square \\ & \text { CP1L-L14D }-\square \\ & \text { CP1L-L20D } \square \\ & \text { CP1L-M30DD- } \\ & \text { CP1L-M40D } \\ & \text { CP1L-M60D }- \end{aligned}$ | SYSMAC CP Series CP1L CPU Unit Introduction Manual | Describes basic setup methods of CP1L PLCs: <br> - Basic configuration and component names <br> - Mounting and wiring <br> - Programming, data transfer, and debugging using the CX-Programmer <br> - Application program examples |
| W446 | WS02-CXPC1-E-V73 | SYSMAC CX-Programmer <br> Ver. 7.2 Operation Manual | Provides information on installing and operating the CX-Programmer for all functions except for function blocks. |
| W447 | WS02-CXPC1-E-V73 | SYSMAC CX-Programmer Ver. 7.1 Operation Manual Function Blocks | Provides specifications and operating procedures for function blocks. Function blocks can be used with CX-Programmer Ver. 7.1 or higher and a CP1L CPU Unit. Refer to W446 for operating procedures for functions other than function blocks. |
| W463 | CXONE-AL $\square \square C-E V 2$ <br> CXONE-AL $\square \square D-E V 2$ | CX-One Setup Manual | Provides an overview of and describes how to install the CX-One FA Integrated Tool Package. |
| W464 |  | CX-Integrator Operation Manual | Describes operating the CX-Integrator, including operations to build networks (e.g., setting data links, routing tables, and Communications Units. |


| Cat. No. | Model numbers | Manual name | Description |
| :---: | :---: | :---: | :---: |
| W344 | WS02-PSTC1-E | CX-Protocol Operation Manual | Provides operating procedures for creating protocol macros (i.e., communications sequences) with the CX-Protocol and other information on protocol macros. <br> The CX-Protocol is required to create protocol macros for user-specific serial communications or to customize the standard system protocols. |
| W342 |  | SYSMAC CS/CJ/CP/ NSJ-series Communications Commands Reference Manual | Describes commands addressed to CS-series, CJseries, and CP-series CPU Units, including C-mode commands and FINS commands. <br> Note This manual describes on commands address to CPU Units regardless of the communications path. (CPU Unit serial ports, Serial Communications Unit/Board ports, and Communications Unit ports can be used.) Refer to the relevant operation manuals for information on commands addresses to Special I/O Units and CPU Bus Units. |

## 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 NONINFRINGEMENT, 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 CP-series Programmable Controllers (PLCs) and related devices. The information contained in this section is important for the safe and reliable application of Programmable Controllers. 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 ..... xxvi
5 Application Precautions. ..... xxvii
6 Conformance to EC Directives ..... xxx
6-1 Applicable Directives ..... xxx
6-2 Concepts ..... xxx
6-3 Conformance to EC Directives ..... xxx
6-4 Relay Output Noise Reduction Methods ..... XXX
6-5 Conditions for Meeting EMC Directives when Using CP/CPM1A Relay Expansion I/O Units ..... xxxii

## 1 Intended Audience

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

- Personnel in charge of installing FA systems.
- Personnel in charge of designing FA systems.
- Personnel in charge of managing FA systems and facilities.


## 2 General Precautions

The user must operate the product according to the performance specifications described in the operation manuals.
Before using the product under conditions which are not described in the manual or applying the product to nuclear control systems, railroad systems, aviation systems, vehicles, combustion systems, medical equipment, amusement machines, safety equipment, and other systems, machines, and equipment that may have a serious influence on lives and property if used improperly, consult your OMRON representative.
Make sure that the ratings and performance characteristics of the product are sufficient for the systems, machines, and equipment, and be sure to provide the systems, machines, and equipment with double safety mechanisms.
This manual provides information for programming and operating the Unit. Be sure to read this manual before attempting to use the Unit and keep this manual close at hand for reference during operation.

WARNING It is extremely important that a 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 touch any of the terminals or terminal blocks 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.

WARNING Provide safety measures in external circuits (i.e., not in the Programmable Controller), including the following items, to ensure safety in the system if an abnormality occurs due to malfunction of the PLC or another external factor affecting the PLC operation. Not doing so may result in serious accidents.

- Emergency stop circuits, interlock circuits, limit circuits, and similar safety measures must be provided in external control circuits.
- The PLC will turn OFF all outputs when its self-diagnosis function detects any error or when a severe failure alarm (FALS) instruction is executed. Unexpected operation, however, may still occur for errors in the I/O control section, errors in I/O memory, and errors that cannot be detected by the self-diagnosis function. As a countermeasure for all these errors, external safety measures must be provided to ensure safety in the system.
- The PLC or outputs may remain ON or OFF due to deposits on or burning of the output relays, or destruction of the output transistors. As a countermeasure for such problems, external safety measures must be provided to ensure safety in the system.
- When the 24-V DC output (service power supply to the PLC) is overloaded or short-circuited, the voltage may drop and result in the outputs being turned OFF. As a countermeasure for such problems, external safety measures must be provided to ensure safety in the system.

WARNING 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. Not doing so may result in serious accidents.

WARNING Do not apply the voltage/current outside the specified range to this unit. It may cause a malfunction or fire.

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 safety at the destination node before transferring a program to another node or editing the I/O area. Doing either of these without confirming safety may result in injury.

Caution Tighten the screws on the terminal block of the AC power supply to the torque specified in this manual. The loose screws may result in burning or malfunction.

4 Caution Do not touch anywhere near the power supply parts or I/O terminals while the power is ON, and immediately after turning OFF the power. The hot surface may cause burn injury.

Caution Pay careful attention to the polarities (+/-) when wiring the DC power supply. A wrong connection may cause malfunction of the system.

Caution When connecting the PLC to a computer or other peripheral device, either ground the 0 V side of the external power supply or do not ground the external power supply at all. Otherwise the external power supply may be shorted depending on the connection methods of the peripheral device. DO NOT ground the 24 V side of the external power supply, as shown in the following diagram.


Caution After programming (or reprogramming) using the IOWR instruction, confirm that correct operation is possible with the new ladder program and data before starting actual operation. Any irregularities may cause the product to stop operating, resulting in unexpected operation in machinery or equipment.

Caution The CP1L CPU Units automatically back up the user program and parameter data to flash memory when these are written to the CPU Unit. I/O memory (including the DM Area, counter present values and Completion Flags, and HR Area), however, is not written to flash memory. The DM Area, counter present values and Completion Flags, and HR Area can be held during power interruptions with a battery. If there is a battery error, the contents of these areas may not be accurate after a power interruption. If the contents of the DM Area, counter present values and Completion Flags, and HR Area are used to control external outputs, prevent inappropriate outputs from being made whenever the Battery Error Flag (A402.04) is ON.

## 4 Operating Environment Precautions

Caution Do not operate the control system in the following locations:

- 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.

1 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.

4 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. Make 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 $100 \Omega$ or less when installing the Units. Not connecting to a ground of $100 \Omega$ or less 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 Expansion Units or any other Units
- Connecting or removing the Memory Cassette or Option Board
- Setting DIP switches or rotary switches
- Connecting or wiring the cables
- Connecting or disconnecting the connectors

4 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.

- Install external breakers and take other safety measures against short-circuiting in external wiring. Insufficient safety measures against short-circuiting may result in burning.
- Mount the Unit only after checking the connectors and terminal blocks completely.
- Be sure that all the terminal screws and cable connector screws are tightened to the torque specified in the relevant manuals. Incorrect tightening torque may result in malfunction.
- Wire all connections correctly according to instructions in this manual.
- 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.
- 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.
- Use crimp terminals for wiring. Do not connect bare stranded wires directly to terminals. Connection of bare stranded wires may result in burning.
- Do not apply voltages to the input terminals in excess of the rated input voltage. Excess voltages may result in burning.
- Do not apply voltages or connect loads to the output terminals in excess of the maximum switching capacity. Excess voltage or loads may result in burning.
- Be sure that the terminal blocks, connectors, Option Boards, and other items with locking devices are properly locked into place. Improper locking may result in malfunction.
- Disconnect the functional ground terminal when performing withstand voltage tests. Not disconnecting the functional ground terminal may result in burning.
- Wire correctly and double-check all the wiring or the setting switches before turning ON the power supply. Incorrect wiring may result in burning.
- Check that the DIP switches and data memory (DM) are properly set before starting operation.
- Check the user program for proper execution before actually running it on the Unit. Not checking the program may result in an unexpected operation.
- Resume operation only after transferring to the new CPU Unit the contents of the DM, HR, and CNT Areas required for resuming operation. Not doing so 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 (including the setting of the startup operating mode).
- Force-setting/force-resetting any bit in memory.
- Changing the present value of any word or any set value in memory.
- 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 buildup. Not doing so may result in malfunction or damage.
- Do not touch the Expansion I/O Unit Connecting Cable while the power is being supplied in order to prevent malfunction due to static electricity.
- Do not turn OFF the power supply to the Unit while data is being transferred.
- When transporting or storing the product, cover the PCBs and the Units or put there in the antistatic bag with electrically conductive materials to prevent LSIs and ICs from being damaged by static electricity, and also keep the product within the specified storage temperature range.
- Do not touch the mounted parts or the rear surface of PCBs because PCBs have sharp edges such as electrical leads.
- Double-check the pin numbers when assembling and wiring the connectors.
- Wire correctly according to specified procedures.
- Do not connect pin $6(+5 \mathrm{~V})$ on the RS-232C Option Board (CP1W-CIF01) on the CPU Unit to any external device other than the NT-ALOO1 or CJ1W-CIF11 Conversion Adapter. The external device and the CPU Unit may be damaged.
- Use the dedicated connecting cables specified in this manual to connect the Units. Using commercially available RS-232C computer cables may cause failures in external devices or the CPU Unit.
- The user program and parameter area data in the CPU Unit is backed up in the built-in flash memory. The BKUP indicator will light on the front of the CPU Unit when the backup operation is in progress. Do not turn OFF the power supply to the CPU Unit when the BKUP indicator is lit. The data will not be backed up if power is turned OFF.
- Do not turn OFF the power supply to the PLC while the Memory Cassette is being written. Doing so may corrupt the data in the Memory Cassette. The BKUP indicator will light while the Memory Cassette is being written. Wait for the BKUP indicator to go out before turning OFF the power supply to the PLC.
- Before replacing the battery, supply power to the CPU Unit for at least 5 minutes and then complete battery replacement within 5 minutes of turn OFF the power supply. Memory data may be corrupted if this precaution is not observed.
- Always use the following size wire when connecting I/O terminals: AWG22 to AWG18 ( 0.32 to $0.82 \mathrm{~mm}^{2}$ ).
- Dispose of the product and batteries according to local ordinances as they apply.
Have qualified specialists properly dispose of used batteries as industrial waste.


- UL standards required that batteries be replaced only by experienced technicians. Do not allow unqualified persons to replace batteries. Also, always follow the replacement procedure provided in the manual.
- Never short-circuit the positive and negative terminals of a battery or charge, disassemble, heat, or incinerate the battery. Do not subject the battery to strong shocks or deform the barry by applying pressure. Doing any of these may result in leakage, rupture, heat generation, or ignition of the battery. Dispose of any battery that has been dropped on the floor or otherwise subjected to excessive shock. Batteries that have been subjected to shock may leak if they are used.
- Always construct external circuits so that the power to the PLC it turned ON before the power to the control system is turned ON. If the PLC power supply is turned ON after the control power supply, temporary errors may result in control system signals because the output terminals on DC Output Units and other Units will momentarily turn ON when power is turned ON to the PLC.
- Fail-safe measures must be taken by the customer to ensure safety in the event that outputs from Output Units remain ON as a result of internal circuit failures, which can occur in relays, transistors, and other elements.
- If the I/O Hold Bit is turned ON, the outputs from the PLC will not be turned OFF and will maintain their previous status when the PLC is switched from RUN or MONITOR mode to PROGRAM mode. Make sure that the external loads will not produce dangerous conditions when this occurs. (When operation stops for a fatal error, including those produced with the FALS(007) instruction, all outputs from Output Unit will be turned OFF and only the internal output status will be maintained.)


## 6 Conformance to EC Directives

## 6-1 Applicable Directives

- EMC Directives
- Low Voltage Directive


## 6-2 Concepts

## EMC Directives

OMRON devices that comply with EC Directives also conform to the related EMC standards so that they can be more easily built into other devices or the overall machine. The actual products have been checked for conformity to EMC standards (see the following note). Whether the products conform to the standards in the system used by the customer, however, must be checked by the customer.
EMC-related performance of the OMRON devices that comply with EC Directives will vary depending on the configuration, wiring, and other conditions of the equipment or control panel on which the OMRON devices are installed. The customer must, therefore, perform the final check to confirm that devices and the overall machine conform to EMC standards.

Note The applicable EMC (Electromagnetic Compatibility) standard is EN61131-2.

## Low Voltage Directive

Always ensure that devices operating at voltages of 50 to $1,000 \mathrm{~V} \mathrm{AC}$ and 75 to 1,500 V DC meet the required safety standards for the PLC (EN61131-2).

## 6-3 Conformance to EC Directives

The CP1L PLCs comply with EC Directives. To ensure that the machine or device in which the CP1L PLC is used complies with EC Directives, the PLC must be installed as follows:

1,2,3... 1. The CP1L PLC must be installed within a control panel.
2. You must use reinforced insulation or double insulation for the DC power supplies used for I/O Units and CPU Units requiring DC power. The output holding time must be 10 ms minimum for the DC power supply connected to the power supply terminals on Units requiring DC power.
3. CP1L PLCs complying with EC Directives also conform to EN61131-2. Radiated emission characteristics (10-m regulations) may vary depending on the configuration of the control panel used, other devices connected to the control panel, wiring, and other conditions. You must therefore confirm that the overall machine or equipment complies with EC Directives.

## 6-4 Relay Output Noise Reduction Methods

The CP1L PLCs conforms to the Common Emission Standards (EN61131-2) of the EMC Directives. However, noise generated by relay output switching may not satisfy these Standards. In such a case, a noise filter must be connected to the load side or other appropriate countermeasures must be provided external to the PLC.
Countermeasures taken to satisfy the standards vary depending on the devices on the load side, wiring, configuration of machines, etc. Following are examples of countermeasures for reducing the generated noise.

## Countermeasures

Countermeasures are not required if the frequency of load switching for the whole system with the PLC included is less than 5 times per minute.
Countermeasures are required if the frequency of load switching for the whole system with the PLC included is more than 5 times per minute.

Note Refer to EN61131-2 for more details.

## Countermeasure Examples

When switching an inductive load, connect an surge protector, diodes, etc., in parallel with the load or contact as shown below.

| Circuit | Current |  | Characteristic | Required element |
| :---: | :---: | :---: | :---: | :---: |
|  | AC | DC |  |  |
|  | Yes | Yes | If the load is a relay or solenoid, there is a time lag between the moment the circuit is opened and the moment the load is reset. <br> If the supply voltage is 24 or 48 V , insert the surge protector in parallel with the load. If the supply voltage is 100 to 200 V , insert the surge protector between the contacts. | The capacitance of the capacitor must be 1 to $0.5 \mu \mathrm{~F}$ per contact current of 1 A and resistance of the resistor must be 0.5 to $1 \Omega$ per contact voltage of 1 V . These values, however, vary with the load and the characteristics of the relay. Decide these values from experiments, and take into consideration that the capacitance suppresses spark discharge when the contacts are separated and the resistance limits the current that flows into the load when the circuit is closed again. <br> The dielectric strength of the capacitor must be 200 to 300 V . If the circuit is an AC circuit, use a capacitor with no polarity. |
| Diode method | No | Yes | The diode connected in parallel with the load changes energy accumulated by the coil into a current, which then flows into the coil so that the current will be converted into Joule heat by the resistance of the inductive load. <br> This time lag, between the moment the circuit is opened and the moment the load is reset, caused by this method is longer than that caused by the CR method. | The reversed dielectric strength value of the diode must be at least 10 times as large as the circuit voltage value. The forward current of the diode must be the same as or larger than the load current. <br> The reversed dielectric strength value of the diode may be two to three times larger than the supply voltage if the surge protector is applied to electronic circuits with low circuit voltages. |
|  | Yes | Yes | The varistor method prevents the imposition of high voltage between the contacts by using the constant voltage characteristic of the varistor. There is time lag between the moment the circuit is opened and the moment the load is reset. <br> If the supply voltage is 24 or 48 V , insert the varistor in parallel with the load. If the supply voltage is 100 to 200 V , insert the varistor between the contacts. | --- |

When switching a load with a high inrush current such as an incandescent lamp, suppress the inrush current as shown below.

## Countermeasure 1



Providing a dark current of approx. one-third of the rated value through an incandescent lamp

Countermeasure 2


Providing a limiting resistor

## 6-5 Conditions for Meeting EMC Directives when Using CP/CPM1A Relay Expansion I/O Units

EN61131-2 immunity testing conditions when using the CP1W-40EDR, CPM1A-40EDR, CP1W-32ER, CP1W-16ER, or CPM1A-16ER with a CP1WCN811 I/O Connecting Cable are given below.

## Recommended Ferrite Core

Ferrite Core (Data Line Filter): 0443-164151 manufactured by Nisshin Electric Minimum impedance: $90 \Omega$ at $25 \mathrm{MHz}, 160 \Omega$ at 100 MHz



Recommended Connection Method
1,2,3... 1. Cable Connection Method

2. Connection Method

As shown below, connect a ferrite core to each end of the CP1W-CN811 I/O Connecting Cable.


## SECTION 1 <br> Features and System Configuration

This section introduces the features of the CP1L and describes its configuration. It also describes the Units that are available and connection methods for the CX-Programmer and other peripheral devices.
1-1 Features and Main Functions ..... 2
1-1-1 CP1L Overview ..... 2
1-1-2 Features ..... 7
1-2 System Configuration ..... 15
1-2-1 Basic System ..... 15
1-2-2 System Expansion. ..... 18
1-2-3 Restrictions on System Configuration ..... 21
1-3 Connecting the CX-Programmer ..... 23
1-3-1 Connecting with a Commercially Available USB Cable ..... 24
1-3-2 Connecting to a Serial Port ..... 31
1-4 Function Charts ..... 33
1-5 Function Blocks ..... 34
1-5-1 Overview of Function Blocks ..... 34
1-5-2 Advantages of Function Blocks ..... 34

## 1-1 Features and Main Functions

## 1-1-1 CP1L Overview

The SYSMAC CP1L PLCs are the low end PLCs in the SYSMAC CP Series of package-type Programmable Controllers. They have the smallest program and I/O capacity. The CP1L PLCs are the same size as the CPM1A and CPM2A PLCs, but offer many more features and high performance.

CPU Units with 60 I/O Points: CP1L-M60D $\square$

- The CPU Unit has 36 inputs and 24 outputs built in.
- The PLC can be expanded to a maximum total of 180 I/O points by using CP-series Expansion I/O Units.


CPU Units with 40 I/O Points: CP1L-M40D

- The CPU Unit has 24 inputs and 16 outputs built in.
- The PLC can be expanded to a maximum total of 160 I/O points by using CP-series Expansion I/O Units.



## CPU Units with 30 I/O Points: CP1L-M30D $\square-\square$

- The CPU Unit has 18 inputs and 12 outputs built in.
- The PLC can be expanded to a maximum total of 150 I/O points by using CP-series Expansion I/O Units.

- Four high-speed counters for two axes and two pulse outputs for two axes can be used with the CPU Unit alone.
- Using CP-series Expansion Units also allows extra functions (such as temperature sensor inputs) to be added.
- Installing an Option Board enables RS-232C and RS-422A/485 communications for Programmable Terminals, Bar Code Readers, Inverters, etc.

Note Settings in the PLC Setup determine whether each input point is to be used as a normal input, interrupt input, quick-response input, or high-speed counter. The instruction used to control each output point determines whether it is used as a normal output, pulse output, or PWM output.

## CPU Units with 20 I/O Points: CP1L-L20D

- The CPU Unit has 12 inputs and 8 outputs built in.
- The PLC can be expanded to a maximum total of 60 I/O points by using CP-series Expansion I/O Units.


CPU Units with 14 I/O Points: CP1L-L14D $\square-\square$

- The CPU Unit has 8 inputs and 6 outputs built in.
- The PLC can be expanded to a maximum total of 54 I/O points by using CP-series Expansion I/O Units.



## CPU Units with 10 I/O Points: CP1L-L10D $\square-\square$

- The CPU Unit has 6 inputs and 4 outputs built in.
- The PLC cannot use CP-series Expansion I/O Units to expand the maximum total of I/O points.

- Four high-speed counters for two axes and two pulse outputs for two axes can be used with the CPU Unit alone.
- Using CP-series Expansion Units also allows extra functions (such as temperature sensor inputs) to be added.
- Installing an Option Board enables RS-232C and RS-422A/485 communications for Programmable Terminals, Bar Code Readers, Inverters, etc.

Note Settings in the PLC Setup determine whether each input point is to be used as a normal input, interrupt input, quick-response input, or high-speed counter. The instruction used to control each output point determines whether it is used as a normal output, pulse output, or PWM output.

## CP1L CPU Units

|  | Type | M CPU Units |  |  | L CPU Units |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Model |  | CP1L-M60DR-A CP1L-M60DR-D CP1L-M60DT-A CP1L-M60DT-D CP1L-M60DT1-D | CP1L-M40DR-A CP1L-M40DR-D CP1L-M40DT-A CP1L-M40DT-D CP1L-M40DT1-D | CP1L-M30DR-A <br> CP1L-M30DR-D <br> CP1L-M30DT-A <br> CP1L-M30DT-D <br> CP1L-M30DT1-D | CP1L-L20DR-A <br> CP1L-L20DR-D <br> CP1L-L20DT-A <br> CP1L-L20DT-D <br> CP1L-L20DT1-D | CP1L-L14DR-A <br> CP1L-L14DR-D <br> CP1L-L14DT-A <br> CP1L-L14DT-D <br> CP1L-L14DT1-D | CP1L-L10DR-A <br> CP1L-L10DR-D <br> CP1L-L10DT-A <br> CP1L-L10DT-D <br> CP1L-L10DT1-D |
| Power supply |  | ```Models with AC power (model numbers ending in "-A"): 100 to 240 V AC, \(50 / 60 \mathrm{~Hz}\) Models with DC power (model numbers ending in "-D"): 24 V DC``` |  |  |  |  |  |
| Program capacity |  | 10K steps |  |  | 5K steps |  |  |
| Maximum number of I/O points |  | $\begin{array}{\|l\|} \hline 180 \\ \text { (See note 1.) } \end{array}$ | $\begin{array}{\|l\|} \hline 160 \\ \text { (See note 1.) } \end{array}$ | $\begin{array}{\|l\|} \hline 150 \\ \text { (See note 1.) } \end{array}$ | $\begin{array}{\|l\|} \hline 60 \\ \text { (See note 2.) } \end{array}$ | $\begin{array}{\|l\|} \hline 54 \\ \text { (See note 2.) } \\ \hline \end{array}$ | $\begin{array}{\|l\|} \hline 10 \\ \text { (See note 3.) } \\ \hline \end{array}$ |
| Normal I/O | I/O points | 60 | 40 | 30 | 20 | 14 | 10 |
|  | Input points | 36 | 24 | 18 | 12 | 8 | 6 |
|  | Input specifications | 24 VDC |  |  |  |  |  |
|  | Interrupt or quick-response inputs | 6 max |  |  |  | 4 max | 2 max |
|  | Output points | 24 | 16 | 12 | 8 | 6 | 4 |
|  | Output specifications | Relay outputs: Model numbers with "R" before the final suffixTransistor outputs, sinking: Model numbers with "T" before the final suffixTransistor outputs, sourcing:Model numbers with "T1" before the final suffix |  |  |  |  |  |
| High-speed counter inputs |  | 4 counters/2 axes, 100 kHz (single-phase), <br> 100 kHz for up/down pulses or pulse plus direction, 50 kHz for differential phases |  |  |  |  |  |
| Pulse outputs |  | 2 axes, 100 kHz (transistor outputs) |  |  |  |  |  |

Note (1) Three Expansion I/O Units connected to a CP-series CPU Unit with 60, 40 or 30 I/O Points.
(2) One Expansion I/O Unit connected to a CP-series CPU Unit with 20 or 14 I/O Points.

## Interpreting CP1L CPU Unit Model Numbers



## 1-1-2 Features

This section describes the main features of the CP1L.

## Basic CP1L Configuration



## Positioning with an Inverter

Positioning can be controlled using an inverter. Previously, an internal pulse output with trapezoidal acceleration/deceleration is created using the PULSE OUTPUT instruction. The position offset is calculated using an error counter for the feedback pulse input from a rotary encoder connected to an inductive motor and the internal pulse output. The error counter is then used to output a speed command to the inverter to control positioning. This enables positioning with high-capacity motors, as well as low-cost positioning with small-capacity motors (in comparison to using a servo).


Note If high-precision positioning is required, we recommend using an inverter with vector control.

## Full Complement of High-speed Counter Functions

High-speed counter inputs can be enabled by connecting rotary encoders to the built-in inputs. The ample number of high-speed counter inputs makes it possible to control a multi-axis device with a single PLC.

- Four $100-\mathrm{kHz}$ (single phase) $/ 50-\mathrm{kHz}$ (differential phases) high-speed counter inputs (4 counters/2 axes) are provided as a standard feature. (See note.)


Note Settings in the PLC Setup determine whether each input point is to be used as a normal input, interrupt input, quick-response input, or high-speed counter.

Full Complement of Highspeed Counter Functions

Full Complement of Pulse Output Functions

## High-speed Processing for High-speed Counter Present Value (PV) Target Values or Range Comparison Interrupts

An interrupt task can be started when the count reaches a specified value or falls within a specified range.

## High-speed Counter Input Frequency (Speed) Monitoring

The input pulse frequency can be monitored using the PRV instruction (one point (counter 0) only, and you must select whether to use input frequency monitoring or counter 3; you cannot use both).
High-speed Counter PV Holding/Refreshing
It is possible to toggle between holding and refreshing the high-speed counter PV by turning ON and OFF the High-speed Counter Gate Flag from the ladder program.

Positioning and speed control by a pulse-input servo driver is enabled by outputting fixed duty ratio pulse output signals from the CPU Unit's built-in outputs.

- Pulse outputs for 2 axes at 100 kHz maximum are provided as standard features. (See note.)


Note The instruction used to control each output point determines whether it is used as a normal output, pulse output, or PWM output.

## Select CW/CCW Pulse Outputs or Pulse Plus Direction Outputs for the Pulse Outputs

The pulse outputs can be selected to match the pulse input specifications of the motor driver.

## Easy Positioning with Absolute Coordinate System Using Automatic Direction Setting

For operations in an absolute coordinate system (i.e., when the origin is established or when the PV is changed by the INI instruction), the CW/CCW direction can be automatically set when PULSE OUTPUT instructions are executed according to whether the specified number of output pulses is more or less than the pulse output PV.

## Triangular Control

If the amount of output pulses required for acceleration and deceleration (the target frequency times the time to reach the target frequency) exceeds the preset target number of output pulses during positioning (when the ACC instruction in independent mode or the PLS2 instruction is executed), the acceleration and deceleration will be shortened and triangular control will be executed instead of trapezoidal control. In other words, the trapezoidal pulse output will be eliminated, with no period of constant speed.

## Target Position Changes during Positioning (Multiple Start)

While positioning using a PULSE OUTPUT (PLS2) instruction is in progress, the target position, target speed, acceleration rate, and deceleration rate can be changed by executing another PLS2 instruction.

## Positioning Changes during Speed Control (Interrupt Feeding)

While speed control in continuous mode is in effect, it is possible to change to positioning in independent mode by executing a PULSE OUTPUT (PLS2) instruction. By this means, interrupt feeding (moving a specified amount) can be executed under specified conditions.

## Target Speed, Acceleration Rate, and Deceleration Rate Changes during Acceleration or Deceleration <br> When a PULSE OUTPUT instruction with trapezoidal acceleration and deceleration is executed (for speed control or positioning), the target speed and acceleration and deceleration rates can be changed during acceleration or deceleration.

## Lighting and Power Control by Outputting Variable Duty Ratio Pulses

Operations, such as lighting and power control, can be handled by outputting variable duty ratio pulse (PWM) output signals from the CPU Unit's built-in outputs.

## Origin Searches

## Input Interrupts

## Quick-response Inputs

## Origin Search and Origin Return Operations Using a Single Instruction

An accurate origin search combining all I/O signals (origin proximity input signal, origin input signal, positioning completed signal, error counter reset output, etc.) can be executed with a single instruction. It is also possible to move directly to an established origin using an origin return operation.

In direct mode, an interrupt task can be started when a built-in input turns ON or OFF. In counter mode, the rising or falling edges of built-in inputs can be counted, and an interrupt task started when the count reaches a specified value. The maximum number of interrupt input points is 6 for CPU Units with 20, 30, 40 or 60 I/O points and 4 for CPU Units with 14 I/O points and 2 for CPU Units with 10 I/O points.

Note For each input point, a selection in the PLC Setup determines whether it is to be used as a normal input, interrupt input, quick-response input, or highspeed counter. The interrupt input response frequency in counter mode must be 5 kHz or less total for all interrupts.

By using quick-response inputs, built-in inputs up to a minimum input signal width of $50 \mu \mathrm{~s}$ can be read regardless of the cycle time. The maximum number of quick-response input points is 6 for CPU Units with 20, 30, 40 or 60 I/O points and 4 for CPU Units with 14 I/O points and 2 for CPU Units with 10 I/O points.

Note For each input, a PLC Setup parameter determines whether it is to be used as a normal input, interrupt input, quick-response input, or high-speed counter.

## Analog Settings

Changing Settings Using Analog Adjustment

By adjusting the analog adjuster with a Phillips screwdriver, the value in the Auxiliary Area can be changed to any value between 0 and 255. This makes it easy to change set values such as timers and counters without Programming Devices.


External analog values of 0 to 10 V (resolution: 256) are converted to digital values and stored in a word in the AR Area. This enables applications that require on-site adjustment of settings that do not demand a particularly high degree of accuracy, such as for example, a setting based on changes in outdoor temperatures or potentiometer inputs.


## Connectability with Various Components

USB Port for Programming Devices

CX-One Support Software, such as the CX-Programmer, connects from the USB port on a computer to the CP1L built-in peripheral USB port via commercially available USB cable.


Personal computer
CX-One (ver. 2.0 or higher)
(e.g., CX-Programmer ver. 7.1 or higher)


Expansion Capability for Serial Ports

Up to two Serial Communications Boards each with one RS-232C port or one RS-422A/485 port can be added to a CPU Unit with 30,40 or 60 I/O points.

One Serial Communications Boards can be added to a CPU Unit with 20 or 14 I/O points. With a total of up to three ports, including the USB port, this makes it possible to simultaneously connect a computer, PT, CP1L, and/or various components, such as an Inverter, Temperature Controller, or Smart Sensor.

NS-series PT, personal computer, bar code reader, etc.


Note (1) The Modbus-RTU easy master (available for all models) makes it easy to control Modbus Slaves (such as Inverters) with serial communications. After the Modbus Slave address, function, and data have been preset in
a fixed memory area (DM), messages can be sent or received independently of the program by turning software switches.

(2) By using the serial PLC Links, a maximum of 10 words of data per CPU Unit can be shared independently of the program among a maximum of nine CPU Units (CP1L-CP1L-CP1H/CJ1M) using RS-422A/485 Option Boards.


## No-battery Operation

Programs, the PLC Setup, and other data can be automatically saved to the CPU Unit's built-in flash memory. Moreover, DM Area data can be saved to the flash memory and then used as initial data when the power is turned ON. This allows programs and initial values (such as recipe setup data) in the DM Area to be saved in the CPU Unit without the need to maintain a backup battery.

CP1L CPU Unit


## Memory Cassettes

## Security

Built-in flash memory data, such as programs and DM initial-value data, can be stored on a Memory Cassette (optional) as backup data. In addition, programs and initial-value data can be easily copied to another CPU Unit using the Memory Cassette to recreate the same system.


A password registration function is provided for the CPU Unit to prevent unauthorized copy of ladder programs. If an attempt is made to read a ladder program from a CX-Programmer, access to the program is denied if the password that is entered does not match the registered password. If incorrect passwords are entered for five consecutive attempts, the CPU Unit does not accept any more passwords for two hours.

## 1-2 System Configuration

## 1-2-1 Basic System





CPU Unit with 10 I/O Points


## Maximum Number of Normal I/O Points

| Type | I/O capacity | Power supply voltage | Model | Normal builtin inputs | Normal built-in outputs | Weight |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| M | 60 points | 100 to 240 VAC | CP1L-M60DR-A | 36 DC inputs | 24 relay outputs | 820 g max. |
|  |  | 24 VDC | CP1L-M60DR-D |  |  | 730 g max. |
|  |  | 100 to 240 VAC | CP1L-M60DT-A |  | 24 transistor (sinking) outputs | 765 g max. |
|  |  | 24 VDC | CP1L-M60DT-D |  |  | 680 g max. |
|  |  |  | CP1L-M60DT1-D |  | 24 transistor (sourcing) outputs | 675 g max . |
|  | 40 points | 100 to 240 VAC | CP1L-M40DR-A | 24 DC inputs | 16 relay outputs | 675 g max. |
|  |  | 24 VDC | CP1L-M40DR-D |  |  | 590 g max . |
|  |  | 100 to 240 VAC | CP1L-M40DT-A |  | 16 transistor (sinking) outputs | 645 g max . |
|  |  | 24 VDC | CP1L-M60DT-D |  |  | 550 g max. |
|  |  |  | CP1L-M40DT1-D |  | 16 transistor (sourcing) outputs | 550 g max. |
|  | 30 points | 100 to 240 VAC | CP1L-M30DR-A | 18 DC inputs | 12 relay outputs | 610 g max . |
|  |  | 24 VDC | CP1L-M30DR-D |  |  | 525 g max. |
|  |  | 100 to 240 VAC | CP1L-M30DT-A |  | 12 transistor (sinking) outputs | 590 g max. |
|  |  | 24 VDC | CP1L-M30DT-D |  |  | 495 g max. |
|  |  |  | CP1L-M30DT1-D |  | 12 transistor (sourcing) outputs | 495 g max. |


| Type | I/O capacity | Power supply voltage | Model | Normal builtin inputs | Normal built-in outputs | Weight |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| L | 20 points | 100 to 240 VAC | CP1L-L20DR-A | 12 DC inputs | 8 relay outputs | 380 g max. |
|  |  | 24 VDC | CP1L-L20DR-D |  |  | 350 g max. |
|  |  | 100 to 240 VAC | CP1L-L20DT-A |  | 8 transistor (sinking) outputs | 360 g max. |
|  |  | 24 VDC | CP1L-L20DT-D |  |  | 335 g max. |
|  |  |  | CP1L-L20DT1-D |  | 8 transistor (sourcing) outputs | 335 g max. |
|  | 14 points | 100 to 240 VAC | CP1L-L14DR-A | 8 DC inputs | 6 relay outputs | 380 g max. |
|  |  | 24 VDC | CP1L-L14DR-D |  |  | 350 g max. |
|  |  | 100 to 240 VAC | CP1L-L14DT-A |  | 6 transistor (sinking) outputs | 360 g max. |
|  |  | 24 VDC | CP1L-L14DT-D |  |  | 335 g max. |
|  |  |  | CP1L-L14DT1-D |  | 6 transistor (sourcing) outputs | 335 g max. |
|  | 10 points | 100 to 240 VAC | CP1L-L10DR-A | 6 DC inputs | 4 relay outputs | 300 g max. |
|  |  | 24 VDC | CP1L-L10DR-D |  |  | 275 g max. |
|  |  | 100 to 240 VAC | CP1L-L10DT-A |  | 4 transistor (sinking) outputs | 290 g max. |
|  |  | 24 VDC | CP1L-L10DT-D |  |  | 270 g max. |
|  |  |  | CP1L-L10DT1-D |  | 4 transistor (sourcing) outputs | 270 g max. |

## Optional Products

Serial
Communications
Expansion

| Item | Model | Specifications | Weight |
| :--- | :--- | :--- | ---: |
| Memory <br> Cassette | CP1W-ME05M | Can be used to store user programs in <br> flash memory, parameters, DM initial <br> values, comment memory, FB pro- <br> grams, and data in RAM. | 10 g max. |
| LCD Option <br> Board | CP1W-DAM011 | Can be used to monitor and change <br> user-specified messages, time or other <br> data of the PLC. | $20 \mathrm{~g} \mathrm{max}$. |
| Ethernet <br> Option <br> Board | CP1W-CIF41 | Can be used to communicate with these <br> units supported OMRON FINS/TCP, <br> FINS/UDP protocol. | 20 g max. |

When serial communications are required for a CP1L CPU Unit, an RS-232C or RS-422A/485 Option Board can be added.
Two Option Boards can be mounted with a CPU Units with 30, 40 or 60 I/O points and one Option Board can be mounted with a CPU Units with 20 or 14 I/O points.
This enables connection by serial communications to NS-series PTs, Bar Code Readers, components such as Inverters, and computers without USB ports (such as when using the CX-Programmer).

NS－series PT，personal computer，bar code reader，etc．


Option Boards for Serial Communications

| Appearance | Name | Model | Port | Serial communications modes |
| :---: | :---: | :---: | :---: | :---: |
|  | RS－232C Option Board | CP1W－CIF01 | One RS－232C port （D－Sub， 9 pins， female） | Host Link，NT Link（1：N or 1：1 Link Master，1：1 Link Slave）， No－protocol，Serial PLC Link Slave，Serial PLC Link Master， Serial Gateway（conversion to CompoWay／F，conversion to Mod－ bus－RTU），peripheral bus |
| $\square$ сомм <br> あぁあぁあ <br> ［回回回 | RS－422A／485 Option Board | CP1W－CIF11／CIF12 | One RS－422A／485 port（terminal block for ferrules） |  |

## 1-2-2 System Expansion

CP-series Expansion Units or Expansion I/O Units can be connected to a CP1L CPU Unit. Up to three Expansion Units or Expansion I/O Units can be connected to a CPU Unit with 30,40 or 60 I/O points and one Expansion Unit or Expansion I/O Unit can be connected to a CPU Unit with 20 or 14 I/O points. This allows for the expansion of various functions such as I/O points or temperature sensor inputs.


CP1L CPU Unit with 20 or 14 I/O Points


One CP-series Expansion I/O Unit or Expansion Unit can be added.

CP1L CPU Unit with 10 I/O Points


No CP-series Expansion I/O Unit or Expansion Unit can be added.

## Using I/O Connecting Cable

When using CP-series Expansion Units and Expansion I/O Units, it is possible to use CP1W-CN811 Connecting Cable to arrange the Units in upper and lower rows.

- I/O Connecting Cable can be used in one place only, and not in multiple places.



## Maximum I/O Points

Up to three Expansion Units or Expansion I/O Units can be connected to a CPU Unit with 30, 40 or 60 I/O points and one Expansion Unit or Expansion I/O Unit can be connected to a CPU Unit with 20 or 14 I/O points. The maximum I/O capacity is thus achieved by connecting either one or three Expansion Units or Expansion I/O Units.

| Type | I/O capacity | Model | Built-in inputs | Built-in outputs | Maximum number of Expansion I/O Units or Expansion Units | Maximum total I/O points |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| M | 60 points | $\begin{aligned} & \text { CP1L-M60DR-A } \\ & \text { CP1L-M60DR-D } \\ & \text { CP1L-M60DT-A } \\ & \text { CP1-M60DT-D } \\ & \text { CP1L-M60DT1-D } \end{aligned}$ | 36 | 24 | 3 Units max. Inputs: $24 \times 3$ Outputs: $16 \times 3$ | Max.: 180 points Inputs: 108 points Outputs: 72 points |
|  | 40 points | $\begin{aligned} & \text { CP1L-M40DR-A } \\ & \text { CP1L-M40DR-D } \\ & \text { CP1L-M40DT-A } \\ & \text { CP1-M40DT-D } \\ & \text { CP1L-M40DT1-D } \end{aligned}$ | 24 | 16 | 3 Units max. Inputs: $24 \times 3$ Outputs: $16 \times 3$ | Max.: 160 points Inputs: 96 points Outputs: 64 points |
|  | 30 points | CP1L-M30DR-A CP1L-M30DR-D CP1L-M30DT-A CP1L-M30DT-D CP1L-M30DT1-D | 18 | 12 | 3 Units max. Inputs: $24 \times 3$ Outputs: $16 \times 3$ | Max.: 150 points Inputs: 90 points Outputs: 60 points |
| L | 20 points | $\begin{aligned} & \text { CP1L-L20DR-A } \\ & \text { CP1L-L20DR-D } \\ & \text { CP1L-L20DT-A } \\ & \text { CP1L-20DT-D } \\ & \text { CP1L-L20DT1-D } \end{aligned}$ | 12 | 8 | 1 Unit max. Inputs: 24 Outputs: 16 | Max.: 60 points Inputs: 36 points Outputs: 24 points |
|  | 14 points | $\begin{aligned} & \text { CP1L-L14DR-A } \\ & \text { CP1L-L14DR-D } \\ & \text { CP1L-L14DT-A } \\ & \text { CP1L-L14DT-D } \\ & \text { CP1L-L14DT1-D } \end{aligned}$ | 8 | 6 | 1 Unit max. Inputs: 24 Outputs: 16 | Max.: 54 points Inputs: 32 points Outputs: 22 points |
|  | 10 points | CP1L-L10DR-A CP1L-L10DR-D CP1L-L10DT-A CP1L-L10DT-D CP1L-L10DT1-D | 6 | 4 | 0 Unit max. Inputs: 0 Outputs: 0 | Max.: 10 points Inputs: 6 points Outputs: 4 points |

## CP-series Expansion I/O Units

| Appearance | Model | Normal inputs | Normal outputs | Weight |
| :---: | :---: | :---: | :---: | :---: |
|  | CP1W-40EDR CPM1A-40EDR | 24 VDC: <br> 24 inputs | 16 relay outputs | 380 g max. |
|  | CP1W-40EDT <br> CPM1A-40EDT |  | 16 transistor outputs (sinking) | 320 g max. |
|  | CP1W-40EDT1 CPM1A-40EDT1 |  | 16 transistor outputs (sourcing) |  |
|  | CP1W-32ER | None | 32 relay outputs | 465 g max. |
|  | CP1W-32ET |  | 32 transistor outputs (sinking) | 325 g max. |
|  | CP1W-32ET1 |  | 32 transistor outputs (sourcing) |  |
|  | CP1W-20EDR1 CPM1A-20EDR1 | 24 VDC: 12 inputs | 8 relay outputs | 300 g max. |
|  | $\begin{aligned} & \hline \text { CP1W-20EDT } \\ & \text { CPM1A-20EDT } \end{aligned}$ |  | 8 transistor outputs (sinking) |  |
|  | CP1W-20EDT1 CPM1A-20EDT1 |  | 8 transistor outputs (sourcing) |  |
|  | CP1W-16ER CPM1A-16ER | None | 16 relay outputs | 280 g max. |
|  | CP1W-16ET |  | 16 transistor outputs (sinking) | 225 g max. |
|  | CP1W-16ET1 |  | 16 transistor outputs (sourcing) |  |


| Appearance | Model | Normal inputs | Normal outputs | Weight |
| :---: | :---: | :---: | :---: | :---: |
|  | $\begin{aligned} & \text { CP1W-8ED } \\ & \text { CPM1A-8ED } \end{aligned}$ | $24 \text { VDC: }$ $8 \text { inputs }$ | None | 200 g max. |
|  | CP1W-8ER CPM1A-8ER | None | 8 relay outputs | 250 g max . |
|  | $\begin{aligned} & \hline \text { CP1W-8ET } \\ & \text { CPM1A-8ET } \end{aligned}$ |  | 8 transistor outputs (sinking) |  |
|  | $\begin{array}{\|l} \text { CP1W-8ET1 } \\ \text { CPM1A-8ET1 } \end{array}$ |  | 8 transistor outputs (sinking) |  |

## CP-series Expansion Units

| Name and appearance | Model | Specifications |  |  | Weight |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Analog I/O Units | CPM1A-MAD01 | 2 analog inputs | $\begin{aligned} & 0 \text { to } 10 \mathrm{~V} / 1 \text { to } 5 \mathrm{~V} / 4 \\ & \text { to } 20 \mathrm{~mA} \end{aligned}$ | Resolution: 256 | 150 g max. |
|  |  | 1 analog output | 0 to $10 \mathrm{~V} /-10$ to $+10 \mathrm{~V} / 4$ to 20 mA |  |  |
|  | $\begin{aligned} & \text { CP1W-MAD11 } \\ & \text { CPM1A-MAD11 } \end{aligned}$ | 2 analog inputs | 0 to $5 \mathrm{~V} / 1$ to $5 \mathrm{~V} / 0$ to $10 \mathrm{~V} /-10$ to $+10 \mathrm{~V} / 0$ to $20 \mathrm{~mA} / 4$ to 20 mA | Resolution: 6,000 |  |
|  |  | 1 analog output | 1 to $5 \mathrm{~V} / 0$ to $10 \mathrm{~V} /-$ 10 to $+10 \mathrm{~V} / 0$ to 20 $\mathrm{mA} / 4$ to 20 mA |  |  |
| Analog Input Units | $\begin{aligned} & \text { CP1W-AD041 } \\ & \text { CPM1A-AD041 } \end{aligned}$ | 4 analog inputs | 0 to $5 \mathrm{~V} / 1$ to $5 \mathrm{~V} / 0$ to $10 \mathrm{~V} /-10$ to $+10 \mathrm{~V} /$ 0 to $20 \mathrm{~mA} /$ $4 \text { to } 20 \text { mA }$ | Resolution: 6,000 | 200 g max. |
| Analog Output Units | CP1W-DA021 | 2 analog outputs | $\begin{aligned} & 1 \text { to } 5 \mathrm{~V} / 0 \text { to } 10 \mathrm{~V} / \\ & -10 \text { to }+10 \mathrm{~V} / \\ & 0 \text { to } 20 \mathrm{~mA} / \\ & 4 \text { to } 20 \mathrm{~mA} \end{aligned}$ |  |  |
|  | $\begin{aligned} & \text { CP1W-DA041 } \\ & \text { CPM1A-DA041 } \end{aligned}$ | 4 analog outputs |  |  |  |
| Temperature Sensor Units | $\begin{aligned} & \text { CP1W-TS001 } \\ & \text { CPM1A-TS001 } \end{aligned}$ | 2 inputs | Thermocouple input K, J |  | 250 g max. |
|  | $\begin{aligned} & \text { CP1W-TS002 } \\ & \text { CPM1A-TS002 } \end{aligned}$ | 4 inputs |  |  |  |  |
|  | $\begin{aligned} & \text { CP1W-TS101 } \\ & \text { CPM1A-TS101 } \end{aligned}$ | 2 inputs | Platinum resistance thermometer input <br> Pt100, JPt100 |  |  |
|  | CP1W-TS102 CPM1A-TS102 | 4 inputs |  |  |  |  |


| Name and appearance | Model | Specifications | Weight |
| :---: | :---: | :---: | :---: |
| DeviceNet I/O Link Unit | CPM1A-DRT21 | As a DeviceNet Slave, 32 inputs and 32 outputs are allocated. | 200 g max. |
| CompoBus/S I/O Link Unit | CP1W-SRT21 CPM1A-SRT21 | As a CompoBus/S slave, 8 inputs and 8 outputs are allocated. | 200 g max. |

## 1-2-3 Restrictions on System Configuration

The following restrictions apply to the CP-series Expansion Units and CPseries Expansion I/O Units that can be connected to CP1L CPU Units.

## ■ Number of Expansion Units and Expansion I/O Units Connected

A maximum of three Units can be connected to a CPU Unit with 30,40 or 60 I/O points and one Unit can be connected to a CPU Unit with 20 or 14 I/O points.

## ■ Mounting Restriction

When connecting CP-series or CPM1A-series Expansion Units or Expansion I/O Units to a CPU Unit with AC power, provide a space of approximately 10 mm between the CPU Unit and the first Expansion Unit or Expansion I/O Unit.


If sufficient space cannot be provided between the CPU Unit and the first Expansion Unit or Expansion I/O Unit, use the PLC in an ambient temperature of 0 to $50^{\circ} \mathrm{C}$.

■ Restrictions in the External Power Supply Capacity
The following restrictions apply when using the external power supply from a CPU Unit with AC power.

## CPU Units with 30, 40 or 60 I/O Points and AC Power (CP1L-M $\square$ DR-A

 and CP1L-M $\square$ DT $\square$-A)When CP1W or CPM1A Expansion Units or Expansion I/O Units are connected to a CPU Unit with 30,40 or 60 I/O Points and AC Power (CP1LM $\quad$ DDR-A and CP1L-M $\square$ DT-A), it may not be possible to use the entire 300 mA from the external power supply due to restrictions in the power supply capacity. The entire 300 mA from the external power supply can be used if Expansion Units and Expansion I/O Units are not connected.

Calculation Examples of Restrictions in External Power Supply Capacity
Calculate the external power supply capacity using the following calculation example.

| Item | CPU Unit | Expansion Unit |  |  | Total | Restriction |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | 1st Unit | 2nd Unit | 3rd Unit |  |  |
|  | CP1L-M40DR-A | CP1W-DA041 | CP1W-DA041 | CP1W-DA041 |  |  |
| 5 V | 0.22 A | 0.08 A | 0.08 A | 0.08 A | 0.46 A |  |
| 24 V | 0.08 A | 0.124 A | 0.124 A | 0.124 A | 0.452 A |  |
| Power consumption | $\begin{aligned} & 5 \mathrm{~V} \times 0.46 \mathrm{~A}=2.3 \mathrm{~W} \\ & 24 \mathrm{~V} \times 0.452 \mathrm{~A}=10.848 \mathrm{~W} \end{aligned}$ |  |  |  | 13.148 W | $\leq 18.5$ W |
| Applicable external power | 18.5 W (total external power supply capacity) $-13.148 \mathrm{~W}=5.352 \mathrm{~W}$ 5.352 W/24V = 0.223 A |  |  |  | 0.223 A | $\leq 0.3 \mathrm{~A}$ |
| supply capacity | Note If the results exceeds 0.3 A, reduce the current consumption to 0.3 A or less. |  |  |  |  |  |

CPU Units with 14 or 20 I/O Points and AC Power (CP1L-L $\square D R-A$ and CP1L-L $\square$ DT $\square-A)$
When CP1W or CPM1A Expansion Units or Expansion I/O Units are connected to a CPU Unit with 14 or 20 I/O Points and AC Power (CP1L-L $\square$ DDRA and CP1L-L $\square \square D T-A$ ), the external power supply cannot be used. If no Expansion Units or Expansion I/O Units are connected, up to 200 mA can be used.

## CPU Units with DC Power

CPU Units with DC power do not have an external power supply.

## ■ Restrictions on the number of simultaneously ON output points

CP1W-32ER/32ET/32ET1's maximum number of simultaneously ON points is 24 (75\%).

## ■ Restrictions Imposed by Ambient Temperature

There are restrictions in the power supply voltage and output load current imposed by the ambient temperature for CPU Units with DC power. Use the CPU Unit within the following ranges of power supply voltage and output load current.

## CPU Units with Relay Outputs (CP1L- $\square \square D R-D)$

Relay Output Load Current Derating Curves for CPU Units and Expansion I/O Units


Note The above restrictions, apply to the relay output load current from the CPU Unit even if Expansion I/O Units are not connected.

Using CP1W-8ER/16ER/20EDR1/32ER/40EDR or CPM1A-8ER/16ER/20EDR1/40EDR Expansion I/O Units with CPU Units with Transistor Outputs (CP1L- $\square \square \square D T-D$ )
Relay Output Load Current Derating Curves for Expansion I/O Units


Note There are no restrictions on the transistor output load current from the CPU Unit.

## CPU Units with AC Power

There are no restrictions on the output load current from CPU Units with AC power.

## 1-3 Connecting the CX-Programmer

The CX-Programmer (version 7.3 or higher), which runs on Windows, can be used with CP-series CP1L PLCs with 10 I/O points. The CX-Programmer (version 7.1 or higher), which runs on Windows, can be used with CP-series CP1L PLCs with 14, 20, 30, 40 or 60 I/O points. Computers running Support Software (e.g., the CX-Programmer) can be connected to the USB port or to a serial port.

Note A Programming Console cannot be used with CP1L PLCs.

## 1-3-1 Connecting with a Commercially Available USB Cable

Connect the computer running the CX-One Support Software (e.g., the CXProgrammer) using a commercially available USB cable to the peripheral USB port on the CPU Unit.


Restrictions when Connecting by USB

In conformity with USB specifications, the following restrictions apply when connecting a computer running Support Software.

- A USB connection is possible for only one CP-series PLC from a single computer. It is not possible to connect multiple PLCs simultaneously.
- Do not disconnect the USB cable while the Support Software is connected online. Before disconnecting the USB cable, be sure to place the application in offline status. If the USB cable is disconnected while online, the situations described below will occur as a result of OS error.
- Windows Me, 2000, or XP:

The Support Software cannot be returned to online status by simply reconnecting the USB cable. First return the Support Software to offline status, and then reconnect the USB cable. Then perform the online connection procedure for the Support Software.

- Windows 98 :

If the USB cable is disconnected while online, an error message may be displayed on a blue screen. If that occurs, it will be necessary to reboot the computer.
The peripheral USB port (conforming to USB 1.1, B connector) is a dedicated port for connecting Support Software, such as the CX-Programmer.

## Items Required for USB Connection

| Operating system | Windows 98, Me, 2000, or XP |
| :--- | :--- |
| Support Software | CX-Programmer Ver. 6.1 (CX-One Ver. 1.1) |
| USB driver | Included with above Support Software. |
| USB cable | USB 1.1(or 2.0) cable (A connector-B connector), 5 m max. |

Installing the USB Driver
The procedure for first connecting a computer to the CP1L peripheral USB port is described below.
It is assumed that the Support Software has already been installed in the computer.

## Windows XP

Turn ON the power supply to the CP1L, and connect USB cable between the USB port of the computer and the peripheral USB port of the CP1L.

After the cable has been connected, the computer will automatically recognize the device and the following message will be displayed.


1,2,3... 1. If the following window appears, select the No, not this time Option and then click the Next Button. This window is not always displayed.
Found New Hardware Wizard
Welcome to the Found New
Hardware Wizard
Windows will search for current and updated software by
looking on your computer, on the hardware installation CD, or on
the Windows Update Web site (with your permission).
Read our privacy policy
Can Windows connect to Windows Update to search for
software?
OYes, this time only
OYes, now and every time I connect a device
©)No, not this times
2. The following window will be displayed. Select the Install from a list of specific location Option and then click the Next Button.

3. The following window will be displayed. Click the Browse Button for the Include this location in the search Field, specify C:\Program Files\} OMRONICX-ServerlUSBlwin2000_XP\Inf, and then click the Next Button. The driver will be installed. ("C:1" indicates the installation drive and may be different on your computer.)

4. Ignore the following window if it is displayed and click the Continue Anyway Button.

5. The following window will be displayed if the installation is completed normally. Click the Finish Button.


## Windows 2000

Turn ON the power supply to the CP1L, and connect USB cable between the USB port of the computer and the peripheral USB port of the CP1L.
After the cable has been connected, the computer will automatically recognize the device and the following message will be displayed.

Found New Hardware

## OMRON-PLC

1,2,3... 1. The following message will be displayed. Click the Next Button.

2. The following window will be displayed.

3. Select the Search for a suitable driver for the device (recommended) Option and then click the Next Button. The following window will be displayed. From the list in the window, select the Specify location Checkbox and then click the Next Button.

4. Click the Browse Button, specify C:\Program Files $\backslash O M R O N \backslash C X-S e r v-$ erlUSBlwin2000_XP\Inf, and then click the Next Button. ("C:\" indicates the installation drive and may be different on your computer.)

5. A search will be made for the driver and the following window will be displayed. Click the Next Button. The driver will be installed.

6. After the driver has been successfully installed, the following window will be displayed. Click the Finish Button.


## Connection Setup Using the CX-Programmer

1,2,3... 1. Select CP1L as the device type in the Change PLC Dialog Box and confirm that USB is displayed in the Network Type Field.

2. Click the OK Button to finish setting the PLC model. Then connect to the CP1L by executing the CX-Programmer's online connection command.

## Checking after Installation

1,2,3... 1. Display the Device Manager at the computer.
2. Click USB (Universal Serial Bus) Controller, and confirm that OMRON SYSMAC PLC Device is displayed.


## Re-installing the USB Driver

If the USB driver installation fails for some reason or is cancelled in progress, the USB driver must be reinstalled.

## Checking USB Driver Status

1,2,3... 1. Display the Device Manager on the computer.
2. If USB Device is displayed for Other devices, it means that the USB driver installation has failed.

| 玉 Device Manager | - $\square$ [] $x$ |
| :---: | :---: |
|  |  |
|  |  |

## Reinstalling the USB Driver

1,2,3... 1. Right-click USB Device and select Delete from the pop-up menu to delete the driver.
2. Reconnect the USB cable. The USB Driver Installation Window will be displayed.
3. Reinstall the USB driver.

## 1-3-2 Connecting to a Serial Port

Mounting a CP1W-CIF01 RS-232C Option Board in a CP1L Option Board slot makes it possible to connect Support Software with serial communications, just as with previous models.


Personal computer
CX-One (e.g., CX-Programmer)


Connect the CX-Programmer to the RS-232C port of the CP1W-CIF01 Option Board by XW2Z-200S-CV/500S-CV RS-232C cable.

Connection Method
Connect the Programming Device using the Connecting Cable that is appropriate for the serial communications mode of the computer and CPU Unit.

| Computer |  | Connecting Cable |  | CP1L CPU Unit |  |
| :---: | :--- | :--- | :--- | :--- | :--- |
| Model | Connector | Model | Length | Connector | Serial <br> communications <br> mode |
| IBM PC/AT or <br> compatible | D-Sub 9 pin, <br> male | XW2Z-200S-CV | 2 m | D-Sub 9 pin, female <br> (With a CP1W-CIF01 RS- <br> 232C Option Board <br> mounted in Option Board <br> Slot 1 or 2.) | Peripheral bus or Host <br> Link (SYSWAY) |
|  | XW2Z-500S-CV | 5 m |  |  |  |

## Serial Communications Mode

| Serial communications mode | Features | CPU Unit setting method |
| :---: | :---: | :---: |
| Peripheral bus (toolbus) | This is the faster mode, so it is generally used for CX-Programmer connections. <br> - Only 1: 1 connections are possible. <br> - When a CP1L CPU Unit is used, the baud rate is automatically detected by the Support Software. | Turn ON pins SW4 (Serial Port 1) and SW5 (Serial Port 2) on the DIP switch on the front panel of the CPU Unit. These settings enable connection by peripheral bus regardless of the serial port settings in the PLC Setup. |
| Host Link (SYSWAY) | A standard protocol for host computers with either 1: 1 or 1: N connections. <br> - Slower than the peripheral bus mode. <br> - Allows modem or optical adapter connections, or longdistance or 1: N connections using RS-422A/485. | Turn OFF pins SW4 (Serial Port 1) and SW5 (Serial Port 2) on the DIP switch on the front panel of the CPU Unit. <br> The mode will then be determined by the serial port settings in the PLC Setup. The default settings are for Host Link with a baud rate of 9,600 bits/s, 1 start bit, data length of 7 bits, even parity, and 2 stop bits. |

Note When a Serial Communications Option Board is mounted in Option Board Slot 1, it is called "Serial Port 1." When mounted in Option Board Slot 2, it is called "Serial Port 2."

## 1-4 Function Charts



## 1-5 Function Blocks

Function blocks can be used in programming SYSMAC CP-series PLCs.

## 1-5-1 Overview of Function Blocks

A function block is a basic program element containing a standard processing function that has been defined in advance. Once the function block has been defined, the user just has to insert the function block in the program and set the $\mathrm{I} / \mathrm{O}$ in order to use the function.
As a standard processing function, a function block is not created with actual physical addresses, but local variables. The user sets parameters (addresses or values) in those variables to use the function block. The addresses used for the variables themselves are automatically assigned by the system (CX-Programmer) each time they are placed in the program.
In particular, each function block is saved by the CX-Programmer as an individual file that can be reused with programs for other PLCs. This makes it possible to create a library of standard processing functions.


## 1-5-2 Advantages of Function Blocks

Function blocks allow complex programming units to be reused easily. Once standard program sections have been created as function blocks and saved in files, they can be reused just by placing a function block in a program and setting the parameters for the function block's I/O. Reusing standardized function blocks reduces the time required for programming/debugging, reduces coding errors, and makes programs easier to understand.

## Structured Programming

Easy-to-read "Block Box" Design

Structured programs created with function blocks have better design quality and required less development time.

The I/O operands are displayed as local variable names in the program, so the program is like a "black box" when entering or reading the program and no extra time is wasted trying to understand the internal algorithm.

Different Processes Easily Created from a Single Function Block

Reduced Coding Errors

## Data Protection

Improved Reusability through Programming with Variables

## Creating Libraries

## Nesting Multiple Languages

Many different processes can be created easily from a single function block by using input variables for the parameters (such as timer SVs, control constants, speed settings, and travel distances) in the standard process.

Coding mistakes can be reduced, because blocks that have already been debugged can be reused.
The local variables in the function block cannot be accessed directly from the outside, so the data can be protected. (Data cannot be changed unintentionally.)

The function block's I/O is entered as local variables, so the data addresses in the function block do not have to be changed as they do when copying and reusing a program section.

Processes that are independent and reusable (such as processes for individual steps, machinery, equipment, or control systems) can be saved as function block definitions and converted to library functions.
The function blocks are created with local variable names that are not tied to physical addresses, so new programs can be developed easily just by reading the definitions from the file and placing them in a new program.

Mathematical expressions can be entered in structured text (ST) language.
Nesting function blocks is supported for CX-Programmer Ver. 6.0 or higher. For example, it is possible to express only special operations in ST language within a function block in a ladder diagram.
Function block (ladder language)


For details on using function blocks, refer to the CX-Programmer Ver. 7.0 Operation Manual: Function Blocks (Cat. No. W447).

## SECTION 2 Nomenclature and Specifications

This section describes the names and functions of CP1L parts and provides CP1L specifications.

2-1 Part Names and Functions. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 38
2-1-1 CP1L CPU Units. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 38
2-1-2 CP1W-CIF01 RS-232C Option Boards . . . . . . . . . . . . . . . . . . . . . . . 41
2-1-3 CP1W-CIF11/CIF12 RS-422A/485 Option Boards . . . . . . . . . . . . . 42
2-2 Specifications . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43
2-2-1 CP1L CPU Units. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43
2-2-2 I/O Memory Details . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 50
2-2-3 I/O Specifications . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 51
2-2-4 CP/CPM1A-series Expansion I/O Unit I/O Specifications ....... 69
2-3 CP1L CPU Unit Operation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 73
2-3-1 Overview of CPU Unit Configuration . . . . . . . . . . . . . . . . . . . . . . . 73
2-3-2 Flash Memory Data Transfers . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 77
2-3-3 Memory Cassette Data Transfers . . . . . . . . . . . . . . . . . . . . . . . . . . . . 79
2-4 CPU Unit Operation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 81
2-4-1 General Flow. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 81
2-4-2 I/O Refreshing and Peripheral Servicing . . . . . . . . . . . . . . . . . . . . . . . . . 82
2-4-3 I/O Refresh Methods. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 82
2-4-4 Initialization at Startup . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 84
2-5 CPU Unit Operating Modes . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 85
2-5-1 Operating Modes . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 85
2-5-2 Status and Operations in Each Operating Mode. . . . . . . . . . . . . . . . 85
2-5-3 Operating Mode Changes and I/O Memory . . . . . . . . . . . . . . . . . . . . . . . . 86
2-5-4 Startup Mode Setting . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 86
2-6 Power OFF Operation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 87
2-6-1 Overview. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 87
2-6-2 Instruction Execution for Power Interruptions . . . . . . . . . . . . . . . . . . 88
2-7 Computing the Cycle Time . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 89
2-7-1 CPU Unit Operation Flowchart . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 89
2-7-2 Cycle Time Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 90
2-7-3 Functions Related to the Cycle Time . . . . . . . . . . . . . . . . . . . . . . . . . 91
2-7-4 I/O Refresh Times for PLC Units . . . . . . . . . . . . . . . . . . . . . . . . . . . . 93
2-7-5 Cycle Time Calculation Example . . . . . . . . . . . . . . . . . . . . . . . . . . . . 93
2-7-6 Online Editing Cycle Time Extension . . . . . . . . . . . . . . . . . . . . . . . . 94
2-7-7 I/O Response Time . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 95
2-7-8 Interrupt Response Times . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 96
2-7-9 Serial PLC Link Response Performance . . . . . . . . . . . . . . . . . . . . . 97
2-7-10 Pulse Output Start Time . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 98
2-7-11 Pulse Output Change Response Time. . . . . . . . . . . . . . . . . . . . . . . . . 98

## 2-1 Part Names and Functions

## 2-1-1 CP1L CPU Units


(1) Battery Cover

Covers the location where the battery is stored.
(2) Operation Indicators Show CP1L operation status.


| POWER (Green) | Lit | Power is ON. |
| :---: | :---: | :---: |
|  | Not lit | Power is OFF. |
| RUN (Green) | Lit | The CP1L is executing a program in either RUN or MONITOR mode. |
|  | Not lit | Operation is stopped in PROGRAM mode or due to a fatal error. |
| ERR/ALM (Red) | Lit | A fatal error (including FALS execution) or a hardware error (WDT error) has occurred. CP1L operation will stop and all outputs will be turned OFF. |
|  | Flashing | A non-fatal error has occurred (including FAL execution). CP1L operation will continue. |
|  | Not lit | Operation is normal. |
| INH (Yellow) | Lit | The Output OFF Bit (A500.15) has turned ON. All outputs will be turned OFF. |
|  | Not lit | Operation is normal. |
| PRPHL (Yellow) | Flashing | Communications (either sending or receiving) are in progress through the peripheral USB port. |
|  | Not lit | Other than the above. |
| BKUP (Yellow) | Lit | A user program, parameters, or Data Memory is being written or accessed in the built-in flash memory (backup memory). <br> A user program, parameters, Data Memory, DM initial values, or comment memory is being written or accessed in a Memory Cassette. <br> The BKUP indicator also lights while user programs, parameters, and Data Memory are being restored when the PLC power supply is turned ON. <br> Note Do not turn OFF the PLC power supply while this indicator is lit. |
|  | Not lit | Other than the above. |

(3) Peripheral USB Port

Used for connecting to a personal computer for programming and monitoring by the CX-Programmer.
(4) Analog Adjuster

By turning the analog adjuster, it is possible to adjust the value of A642 within a range of 0 to 255. (Refer to 6-4 Analog Adjuster and External Analog Setting Input.)
(5) External Analog Setting Input Connector By applying 0 to 10 V of external voltage, it is possible to adjust the value of A643 within a range of 0 to 256 . This input is not isolated. (Refer to 6 4 Analog Adjuster and External Analog Setting Input.)
(6) DIP Switch

CPU Units with 30, 40 or 60 I/O Points


| No. | Setting | Description | Application | Default |
| :--- | :--- | :--- | :--- | :--- |
| SW1 | ON | User memory write- <br> protected (See note.) | Used to prevent pro- <br> grams from being inad- <br> vertently overwritten. | OFF |
|  | OFF | User memory not <br> write-protected. | Data automatically <br> transferred from <br> Memory Cassette at <br> startup. | Used to enable pro- <br> grams, Data Memory, or <br> parameters saved on a <br> Memory Cassette to be <br> opened by the CPU Unit <br> at startup. |

Note The following data will be write-protected if pin SW1 is turned ON:

- The entire user program (all tasks)
- All data in parameter areas (such as the PLC Setup)

When SW1 is turned ON, the user program and the data in the parameter areas will not be cleared even if the All Clear operation is performed from a Peripheral Device (i.e., the CX-Programmer).
CPU Units with 10,14 or 20 I/O Points


| No. | Setting | Description | Application | Default |
| :---: | :--- | :--- | :--- | :---: |
| SW1 | ON | User memory write- <br> protected (See note.) | Used to prevent pro- <br> grams from being inad- <br> vertently overwritten. | OFF |
|  | OFF | User memory not <br> write-protected. |  |  |


| No. | Setting | Description | Application | Default |
| :--- | :--- | :--- | :--- | :--- |
| SW2 | ON | Data automatically <br> transferred from <br> Memory Cassette at <br> startup. | Used to enable pro- <br> grams, Data Memory, or <br> parameters saved on a <br> Memory Cassette to be <br> opened by the CPU Unit <br> at startup. | OFF |
|  | OFF | Data not transferred |  |  |
|  | ON | A395.12 ON | This pin enables control- <br> ling a bit in memory with- <br> out using an input relay. | OFF |
|  | OFF | A395.12 OFF | Used for peripheral <br> bus. | Used to enable a Serial <br> Communications Option <br> Board mounted in Option <br> Board Slot 1 to be used |
| SW4 | ON | By the peripheral bus. <br> This SW should be <br> always OFF for CPU <br> Units with 10 I/O points, <br> becording to PLC <br> Secause there is no <br> option slot in CPU Units <br> with 10 I/O points. |  |  |
|  | OFF |  |  |  |

Note The following data will be write-protected if pin SW1 is turned ON:

- The entire user program (all tasks)
- All data in parameter areas (such as the PLC Setup)

When SW1 is turned ON, the user program and the data in the parameter areas will not be cleared even if the All Clear operation is performed from a Peripheral Device (i.e., the CX-Programmer).
(7) Memory Cassette Slot

Used for mounting a CP1W-ME05M Memory Cassette. When mounting a Memory Cassette, remove the dummy cassette.
Data, such as CP1L CPU Unit programs, parameters, and data memory, can be transferred to the Memory Cassette to be saved.
(8) Power Supply, Ground, and Input Terminal Block

| Power supply ter- <br> minals | Used to provide a 100- to 240-VAC or 24-VDC power <br> supply. |
| :--- | :--- |
| Ground terminals | Functional ground ( $\quad$ ( ) : : <br> Connect this ground to strengthen noise immunity and to <br> prevent electric shock. <br> (AC power supply models only.) |
| Protective ground ( 它): <br> To prevent electric shock, ground to $100 \Omega$ or less. |  |
| Input terminals | Used to connect input devices. |

(9) Option Board Slots

The following Option Boards can be mounted in either slot 1 (left) or slot 2 (right).

- CP1W-CIF01 RS-232C Option Board
- CP1W-CIF11/CIF12 RS-422A/485 Option Board
- CP1W-DAM01 LCD Option Board
- CP1W-CIF41 Ethernet Option Board

Caution Always turn OFF the power supply to the PLC before mounting or removing an Option Board.
(10) Input Indicators

The input indicators light when input terminal contacts turn ON.
(11) Expansion I/O Unit Connector

CP-series Expansion I/O Units and Expansion Units (Analog I/O Units, Temperature Sensor Units, CompoBus/S I/O Link Units, or DeviceNet I/O Link Units) can be connected. Up to three Expansion Units or Expansion I/O Units can be connected to a CPU Unit with 30, 40 or 60 I/O points and one Expansion Unit or Expansion I/O Unit can be connected to a CPU Unit with 20 or 14 I/O points. (For details on using Expansion Units and Expansion I/O Units, refer to SECTION 7 Using Expansion Units and Expansion I/O Units.)
(12) Output Indicators

The output indicators light when output terminal contacts turn ON.
(13) External Power Supply and Output Terminal Block

| External power <br> supply terminals | CPU Units with AC power supply specifications have <br> external 24-VDC, 300-mA, power supply terminals. <br> (except for the CP1L-LПDR-A, which has a 200-mA <br> power supply terminals). They can be used as service <br> power supplies for input devices. |
| :--- | :--- |
| Output terminals | Used for connecting output devices. |

## 2-1-2 CP1W-CIF01 RS-232C Option Boards

An RS-232C Option Board can be mounted to an Option Board slot on the CPU Unit. With a CPU Unit with 30, 40 or 60 I/O points, either Option Board slot may be used.
When mounting an Option Board, first remove the slot cover. Grasp both of the cover's up/down lock levers at the same time to unlock the cover, and then pull the cover out.
Then to mount the Option Board, check the alignment and firmly press it in until it snaps into place.
\. Caution Always turn OFF the power supply to the PLC before mounting or removing an Option Board.


## RS-232C Connector



| Pin | Abbr. | Signal name | Signal direction |
| :--- | :--- | :--- | :--- |
| 1 | FG | Frame Ground | --- |
| 2 | SD (TXD) | Send Data | Output |
| 3 | RD (RXD) | Receive Data | Input |
| 4 | RS (RTS) | Request to Send | Output |
| 5 | CS (CTS) | Clear to Send | Input |
| 6 | 5V | Power Supply | --- |
| 7 | DR (DSR) | Data Set Retry | Input |
| 8 | ER (DTR) | Equipment Ready | Output |
| 9 | SG (OV) | Signal Ground | --- |
| Connector hood | FG | Frame Ground | --- |

## 2-1-3 CP1W-CIF11/CIF12 RS-422A/485 Option Boards

An RS-422A/485 Option Board can be mounted to an Option Board slot on the CPU Unit. With a CPU Unit with 30,40 or 60 I/O points, either Option Board slot may be used.
When mounting an Option Board, first remove the slot cover. Grasp both of the cover's up/down lock levers at the same time to unlock the cover, and then pull the cover out.
Then to mount the Option Board, check the alignment and firmly press it in until it snaps into place.

Caution Always turn OFF the power supply to the PLC before mounting or removing an Option Board.


## RS-422A/485 Terminal Block



Tighten the terminal block screws to a torque of $0.28 \mathrm{~N} \cdot \mathrm{~m}(2.5 \mathrm{Lb} \ln$.$) .$

## DIP Switch for Operation Settings



| Pin | Settings |  |  |
| :---: | :---: | :---: | :---: |
| 1 | ON | ON (both ends) | Terminating resistance selection |
|  | OFF | OFF |  |
| 2 | ON | 2-wire | 2-wire or 4-wire selection (See note 1.) |
|  | OFF | 4-wire |  |
| 3 | ON | 2-wire | 2-wire or 4-wire selection (See note 1.) |
|  | OFF | 4-wire |  |
| 4 | --- | --- | Not used. |
| 5 | ON | RS control enabled | RS control selection for RD (See note 2.) |
|  | OFF | RS control disabled (Data always received.) |  |
| 6 | ON | RS control enabled | RS control selection for SD (See note 3.) |
|  | OFF | RS control disabled (Data always sent.) |  |

Note (1) Set both pins 2 and 3 to either ON (2-wire) or OFF (4-wire).
(2) To disable the echo-back function, set pin 5 to ON (RS control enabled).
(3) When connecting to a device on the N side in a 1: N connection with the 4 -wire method, set pin 6 to ON (RS control enabled).
Also, when connecting by the 2 -wire method, set pin 6 to ON (RS control enabled).

## 2-2 Specifications

## 2-2-1 CP1L CPU Units

## General Specifications

| Power supply |  | AC power supply | DC power supply |
| :---: | :---: | :---: | :---: |
| Model numbers | 60 I/O points | CP1L-M60DR-A, CP1L-M60DT-A | CP1L-M60DR-D, CP1L-M60DT-D, or CP1L-M60DT1-D |
|  | 40 I/O points | CP1L-M40DR-A, <br> CP1L-M40DT-A | CP1L-M40DR-D, CP1L-M40DT-D, or CP1L-M40DT1-D |
|  | 30 I/O points | CP1L-M30DR-A, CP1L-M30DT-A | CP1L-M30DR-D, CP1L-M30DT-D, or CP1L-M30DT1-D |
|  | 20 I/O points | CP1L-L20DR-A, CP1L-L20DT-A | CP1L-L20DR-D, CP1L-L20DT-D, or CP1L-L20DT1-D |
|  | 14 I/O points | CP1L-L14DR-A, CP1L-L14DT-A | CP1L-L14DR-D, CP1L-L14DT-D, or CP1L-L14DT1-D |
|  | 10 I/O points | CP1L-L10DR-A, CP1L-L10DT-A | CP1L-L10DR-D, CP1L-L10DT-D, or CP1L-L10DT1-D |
| Power supply |  | $\begin{aligned} & 100 \text { to } 240 \mathrm{VAC} \\ & 50 / 60 \mathrm{~Hz} \end{aligned}$ | 24 VDC |
| Operating voltage range |  | 85 to 264 VAC | 20.4 to 26.4 VDC |
| Power consumption |  | 50 VA max. (CP1L-M $\square$ DR-A) 50 VA max. (CP1L-M $\square$ DT-A) 30 VA max. (CP1L-L $\square D R-A)$ 30 VA max. (CP1L-L $\square D T-A)$ | See note 3. <br> 20 W max. (CP1L-M $\square$ DR-D) <br> 20 W max. (CP1L-M $\square D T \square$-D) <br> 13 W max. (CP1L-LDDR-D) <br> 13 W max. (CP1L-L $\square D T \square-D)$ |


| Power supply |  | AC power supply | DC power supply |
| :---: | :---: | :---: | :---: |
| Model numbers | 60 I/O points | $\begin{aligned} & \text { CP1L-M60DR-A, } \\ & \text { CP1L-M60DT-A } \end{aligned}$ | CP1L-M60DR-D, CP1L-M60DT-D, or CP1L-M60DT1-D |
|  | 40 I/O points | CP1L-M40DR-A, <br> CP1L-M40DT-A | CP1L-M40DR-D, CP1L-M40DT-D, or CP1L-M40DT1-D |
|  | 30 I/O points | CP1L-M30DR-A, CP1L-M30DT-A | CP1L-M30DR-D, CP1L-M30DT-D, or CP1L-M30DT1-D |
|  | 20 I/O points | CP1L-L20DR-A, <br> CP1L-L20DT-A | CP1L-L20DR-D, CP1L-L20DT-D, or CP1L-L20DT1-D |
|  | 14 I/O points | CP1L-L14DR-A, CP1L-L14DT-A | CP1L-L14DR-D, CP1L-L14DT-D, or CP1L-L14DT1-D |
|  | 10 I/O points | CP1L-L10DR-A, CP1L-L10DT-A | CP1L-L10DR-D, CP1L-L10DT-D, or CP1L-L10DT1-D |
| Inrush current (See note 1.) |  | 100 to 120 VAC inputs: <br> 20 A max.(for cold start at room temperature.) 8 ms max. <br> 200 to 240 VAC inputs: <br> 40 A max.(for cold start at room temperature.) 8 ms max. | 30 A max.(for cold start.) 20 ms max. |
| External power supply (See note 2.) |  | 300 mA at 24 VDC (CP1L-M $\square$ DR-A) 300 mA at 24 VDC (CP1L-MDDT-A) 200 mA at 24 VDC (CP1L-L $\square$ DR-A) 200 mA at 24 VDC (CP1L-L $\square D T-A)$ | None |
| Insulation resistance |  | $20 \mathrm{M} \Omega \mathrm{min}$. (at 500 VDC ) between the external AC terminals and GR terminals | No insulation between primary and secondary DC power supplies. |
| Dielectric strength |  | 2,300 VAC $50 / 60 \mathrm{~Hz}$ for 1 min between the external AC and GR terminals, leakage current: 5 mA max. | No insulation between primary and secondary DC power supplies. |
| Noise resistance |  | Conforms to IEC 61000-4-4 2 kV (power supply line) |  |
| Vibration resistance |  | 10 to $57 \mathrm{~Hz}, 0.075-\mathrm{mm}$ amplitude, 57 to 150 Hz , acceleration: $9.8 \mathrm{~m} / \mathrm{s}^{2}$ in $\mathrm{X}, \mathrm{Y}$, and $Z$ directions for 80 minutes each (time coefficient of 8 minutes $\times$ coefficient factor of $10=$ total time of 80 minutes) |  |
| Shock resistance |  | $147 \mathrm{~m} / \mathrm{s}^{2}$ three times each in $\mathrm{X}, \mathrm{Y}$, and Z directions |  |
| Ambient operating |  | 0 to $55^{\circ} \mathrm{C}$ |  |
| Ambient humidity |  | 10\% to 90\% (with no condensation) |  |
| Atmosphere |  | No corrosive gas. |  |
| Ambient storage |  | -20 to $75^{\circ} \mathrm{C}$ (excluding battery) |  |
| Terminal screw size |  | M3 |  |
| Power interrupt time |  | 10 ms min . | $2 \mathrm{~ms} \mathrm{min}$. |
| Weight |  | CP1L-M60D $\square-\square: 820 \mathrm{~g}$ max. CP1L-M40D $\square-\square: 675 \mathrm{~g}$ max. CP1L-M30D $\square-\square: 610 \mathrm{~g}$ max. CP1L-L20D $\square-\square$ : 380 g max. CP1L-L14D $\square$ - $\square: 380 \mathrm{~g}$ max. CP1L-L10D $\square$ - $\square: 300 \mathrm{~g}$ max. |  |

Note
(1) The above values are for a cold start at room temperature for an AC power supply, and for a cold start for a DC power supply.

- A thermistor (with low-temperature current suppression characteristics) is used in the inrush current control circuitry for the AC power supply. The thermistor will not be sufficiently cooled if the ambient temperature is high or if a hot start is performed when the power supply has been OFF for only a short time, so in those cases the inrush current values may be higher (as much as two times higher) than those shown above.
- A capacitor delay circuit is used in the inrush current control circuitry for the DC power supply. The capacitor will not be charged if a hot start is performed when the power supply has been OFF for only a short time, so in those cases the inrush current values may be higher (as much as two times higher) than those shown above.
Always allow for this when selecting fuses and breakers for external circuits.
(2) Use the external power supply to power input devices. Do not use it to drive output devices.
(3) This is the rated value for the maximum system configuration. Use the following formula to calculate DC power consumption for CPU Units with DC power.


## Formula:

DC-powered CP1L power consumption $=5-\mathrm{V}$ current consumption $\times 5 \mathrm{~V} /$ $70 \%$ (CP1L internal power efficiency) $+24-\mathrm{V}$ current consumption $\times 1.1$ (current fluctuation factor)
Calculation Example

| System | CPU Unit | Expansion Unit or Expansion I/O Unit |  |  | Total |
| :--- | :--- | :--- | :--- | :--- | :---: |
|  |  | 1st Unit | 2nd Unit | 3rd Unit |  |
|  | CP1L-M40DR-D | CP1W-20EDT | CP1W-TS001 | CP1W-DA041 |  |
| 5 V | 0.220 A | 0.130 A | 0.040 A | 0.080 A | 0.470 A |
| 24 V | 0.080 A | 0.000 A | 0.059 A | 0.124 A | 0.263 A |

CP1L Power Consumption

$$
=(0.47 \mathrm{~A} \times 5 \mathrm{~V} / 70 \%+0.263 \mathrm{~A} \times 24 \mathrm{~V}) \times 1.1
$$

$$
=10.64 \mathrm{~W}
$$

The above calculation results show that a power supply with a capacity of 7 W or greater is required.

## Current Consumption

## CPU Units

| I/O capacity | Model | Current consumption |  | External power |
| :---: | :---: | :---: | :---: | :---: |
|  |  | 5 V DC | 24 V DC | 24 V DC |
| 60 I/O points | CP1L-M60DR-A | 0.25 A | 0.14 A | 0.3 A max. |
|  | CP1L-M60DR-D | 0.25 A | 0.14 A | --- |
|  | CP1L-M60DT-A | 0.39 A | 0.03 A | 0.3 A max. |
|  | CP1L-M60DT-D | 0.39 A | 0.03 A | --- |
|  | CP1L-M60DT1-D | 0.39 A | 0.03 A | --- |
| 40 I/O points | CP1L-M40DR-A | 0.22 A | 0.08 A | 0.3 A max. |
|  | CP1L-M40DR-D | 0.22 A | 0.08 A | --- |
|  | CP1L-M40DT-A | 0.31 A | 0.03 A | 0.3 A max. |
|  | CP1L-M40DT-D | 0.31 A | 0.03 A | --- |
|  | CP1L-M40DT1-D | 0.31 A | 0.03 A | --- |
| 30 I/O points | CP1L-M30DR-A | 0.21 A | 0.07 A | 0.3 A max. |
|  | CP1L-M30DR-D | 0.21 A | 0.07 A | --- |
|  | CP1L-M30DT-A | 0.28 A | 0.03 A | 0.3 A max. |
|  | CP1L-M30DT-D | 0.28 A | 0.03 A | --- |
|  | CP1L-M30DT1-D | 0.28 A | 0.03 A | --- |
| 20 I/O points | CP1L-L20DR-A | 0.20 A | 0.05 A | 0.2 A max. |
|  | CP1L-L20DR-D | 0.20 A | 0.05 A | --- |
|  | CP1L-L20DT-A | 0.24 A | 0.03 A | 0.2 A max. |
|  | CP1L-L20DT-D | 0.24 A | 0.03 A | --- |
|  | CP1L-L20DT1-D | 0.24 A | 0.03 A | --- |
| 14 I/O points | CP1L-L14DR-A | 0.18 A | 0.04 A | 0.2 A max. |
|  | CP1L-L14DR-D | 0.18 A | 0.04 A | --- |
|  | CP1L-L14DT-A | 0.21 A | 0.03 A | 0.2 A max. |
|  | CP1L-L14DT-D | 0.21 A | 0.03 A | --- |
|  | CP1L-L14DT1-D | 0.21 A | 0.03 A | --- |
| 10 I/O points | CP1L-L10DR-A | 0.16 A | 0.03 A | 0.2 A max. |
|  | CP1L-L10DR-D | 0.16 A | 0.03 A | --- |
|  | CP1L-L10DT-A | 0.18 A | 0.03 A | 0.2 A max. |
|  | CP1L-L10DT-D | 0.18 A | 0.03 A | --- |
|  | CP1L-L10DT1-D | 0.18 A | 0.03 A | --- |

Note (1) The current consumption of the CP1W-ME05M Memory Cassette and CP1W-CIF01/11 Option Boards are included in the current consumption of the CPU Unit.
(2) The current consumption of the following is not included with the current consumption of the CPU Unit: CP1W-CIF12.

| Unit | Model | Current consumption |  | External power <br> supply |
| :---: | :---: | :--- | :--- | :---: |
|  |  | 5 V DC | $\mathbf{2 4 ~ V ~ D C ~}$ |  |
| Interface Unit | CP1W-CIF12 | 0.075 A | --- | --- |

(3) CPU Units with DC power do not provide an external power supply.
(4) The current consumptions given in the following table must be added to the current consumption of the CPU Unit if an Expansion Unit or Expansion I/O Unit is connected.
(5) The external power supply cannot be used if an Expansion Unit or Expansion I/O Unit is connected to a CPU Unit with 14 or 20 I/O points.

Expansion Units and Expansion I/O Units

| Unit name |  | Model |  | Current consumption |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | 5 VDC | 24 VDC |
| Expansion I/O Units |  |  |  | 40 I/O points <br> 24 inputs <br> 16 outputs | CP1W/CPM1A-40EDR | 0.080 A | 0.090 A |
|  |  | CP1W/CPM1A-40EDT | 0.160 A |  | --- |
|  |  | CP1W/CPM1A-40EDT1 |  |  |  |
|  |  | 32 outputs | CP1W-32ER | 0.049 A | 0.131 A |
|  |  | CP1W-32ET | 0.113 A | --- |  |
|  |  | CP1W-32ET1 |  |  |  |
|  |  | 20 I/O points 12 inputs 8 outputs | CP1W/CPM1A-20EDR1 | 0.103 A | 0.044 A |
|  |  | CP1W/CPM1A-20EDT | 0.130 A | --- |  |
|  |  | CP1W/CPM1A-20EDT1 |  |  |  |
|  |  | 16 outputs | CP1W/CPM1A-16ER | 0.042 A | 0.090 A |
|  |  | CP1W-16ET | 0.076 A | --- |  |
|  |  | CP1W-16ET1 |  |  |  |
|  |  | 8 inputs | CP1W/CPM1A-8ED | 0.018 A | --- |
|  |  | 8 outputs | CP1W/CPM1A-8ER | 0.026 A | 0.044 A |
|  |  | CP1W/CPM1A-8ET | 0.075 A | --- |  |
|  |  | CP1W/CPM1A-8ET1 |  |  |  |
| Expansion Units | Analog Input Unit |  | 4 inputs | CP1W/CPM1A-AD041 | 0.100 A | 0.090 A |
|  | Analog Output Unit |  | 2 outputs | CP1W-DA021 | 0.040 A | 0.095 A |
|  |  | 4 outputs | CP1W/CPM1A-DA041 | 0.080 A | 0.124 A |
|  | Analog I/O Units | 2 inputs 1 output | CP1W/CPM1A-MAD01 | 0.066 A | 0.066 A |
|  |  |  | CP1W/CPM1A-MAD11 | 0.083 A | 0.110 A |
|  | Temperature Sensor Units | K or J thermocouples | CP1W/CPM1A-TS001 | 0.040 A | 0.059 A |
|  |  |  | CP1W/CPM1A-TS002 |  |  |
|  |  | Pt or JPt platinum resistance thermometers | CP1W/CPM1A-TS101 | 0.054 A | 0.073 A |
|  |  |  | CP1W/CPM1A-TS102 |  |  |
|  | CompoBus/S I/O Link Unit | 8 inputs 8 outputs | CP1W/CPM1A-SRT21 | 0.029 A | --- |
|  | DeviceNet I/O Link Unit | 32 inputs 32 outputs | CPM1A-DRT21 | 0.048 A | --- |

Note CP1W-32ER/32ET/32ET1's maximum number of simultaneously ON points is 24 (75\%).

## Characteristics

| Type | M CPU Units |  |  | L CPU Units |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Model | CP1L-M60DR-A <br> CP1L-M60DR-D <br> CP1L-M60DT-A <br> CP1L-M60DT-D <br> CP1L-M60DT1-D | CP1L-M40DR-A <br> CP1L-M40DR-D <br> CP1L-M40DT-A <br> CP1L-M40DT-D <br> CP1L-M40DT1-D | CP1L-M30DR-A CP1L-M30DR-D CP1L-M30DT-A CP1L-M30DT-D CP1L-M30DT1-D | CP1L-L20DR-A <br> CP1L-L20DR-D <br> CP1L-L20DT-A <br> CP1L-L20DT-D <br> CP1L-L20DT1-D | CP1L-L14DR-A CP1L-L14DR-D CP1L-L14DT-A CP1L-L14DT-D CP1L-L14DT1-D | CP1L-L10DR-A <br> CP1L-L10DR-D <br> CP1L-L10DT-A <br> CP1L-L10DT-D <br> CP1L-L10DT1-D |
| Program capacity | 10 Ksteps |  |  | 5 Ksteps |  |  |
| Control method | Stored program method |  |  |  |  |  |
| I/O control method | Cyclic scan with immediate refreshing |  |  |  |  |  |
| Program language | Ladder diagram |  |  |  |  |  |
| Function blocks | Maximum number of function block definitions: 128 <br> Maximum number of instances: 256 <br> Languages usable in function block definitions: Ladder diagrams, structured text (ST) |  |  |  |  |  |


| Type |  |  | M CPU Units |  |  | L CPU Units |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Model |  |  | CP1L-M60DR-A CP1L-M60DR-D CP1L-M60DT-A CP1L-M60DT-D CP1L-M60DT1-D | CP1L-M40DR-A CP1L-M40DR-D CP1L-M40DT-A CP1L-M40DT-D CP1L-M40DT1-D | CP1L-M30DR-A CP1L-M30DR-D CP1L-M30DT-A CP1L-M30DT-D CP1L-M30DT1-D | CP1L-L20DR-A CP1L-L20DR-D CP1L-L20DT-A CP1L-L20DT-D CP1L-L20DT1-D | CP1L-L14DR-A CP1L-L14DR-D CP1L-L14DT-A CP1L-L14DT-D CP1L-L14DT1-D | CP1L-L10DR-A <br> CP1L-L10DR-D <br> CP1L-L10DT-A <br> CP1L-L10DT-D <br> CP1L-L10DT1-D |
| Instruction length |  |  | 1 to 7 steps per instruction |  |  |  |  |  |
| Instructions |  |  | Approx. 500 (function codes: 3 digits) |  |  |  |  |  |
| Instruction execution time |  |  | Basic instructions: $0.61 \mu \mathrm{~s} \mathrm{~min}$. Special instructions: $4.1 \mu \mathrm{~s} \mathrm{~min}$. |  |  |  |  |  |
| Common processing time |  |  | 0.38 ms |  |  |  |  |  |
| Number of connectable Expansion Units and Expansion I/O Units |  |  | 3 Units (CP Series or CPM1A) |  |  | 1 Unit (CP Series or CPM1A) |  | 0 Unit |
| Maximum number of I/O points |  |  | 180 points (60 built in, $40 \times 3$ expansion) | 160 points (40 built in, $40 \times 3$ expansion) | 150 points (30 built in, $40 \times 3$ expansion) | 60 points (20 built in, $40 \times 1$ expansion) | 54 points (14 built in, $40 \times 1$ expansion) | 10 points (10 built in) |
| Built-in <br> terminals <br> (Func- <br> tions can <br> be <br> assigned.) | Built-in I/O |  | 60 terminals (36 inputs and 24 outputs) | 40 terminals (24 inputs and 16 outputs) | 30 terminals (18 inputs and 12 outputs) | 20 terminals (12 inputs and 8 outputs) | 14 terminals (8 inputs and 6 outputs) | 10 terminals (6 inputs and 4 outputs) |
|  | Interrupt inputs | Direct mode | 6 inputs |  |  |  | 4 inputs | 2 inputs |
|  |  |  | Response time: 0.3 ms |  |  |  |  |  |
|  |  | Counter mode | 6 inputs |  |  |  | 4 inputs | 2 inputs |
|  |  |  | Response frequency: 5 kHz total, 16 bits Incrementing counter or decrementing counter |  |  |  |  |  |
|  | Quickresponse inputs |  | 6 points |  |  |  | 4 points | 2 points |
|  |  |  | Min. input pulse width: $50 \mu \mathrm{~s}$ max. |  |  |  |  |  |
|  | High-speed counters |  | 4 inputs/2 axes (24 VDC) <br> - Single phase (pulse plus direction, up/down, increment), 100 kHz <br> - Differential phases (4×), 50 kHz <br> Value range: 32 bits, Linear mode or ring mode Interrupts: Target value comparison or range comparison |  |  |  |  |  |
| Pulse outputs (Transistor output models only) | Pulse outputs |  | 2 outputs, 1 Hz to 100 kHz <br> (CCW/CW or pulse plus direction) <br> Trapezoidal or S-curve acceleration and deceleration (Duty ratio: 50\% fixed) |  |  |  |  |  |
|  | PWM outputs |  | 2 outputs, 0.1 to $6,553.5 \mathrm{~Hz}$ or 1 to $32,800 \mathrm{~Hz}$ <br> Variable duty ratio: $0.0 \%$ to $100.0 \%$ (in increments of $0.1 \%$ or $1 \%$ ) <br> Accuracy: $+1 \% /-0 \%$ at 0.1 Hz to $10,000 \mathrm{~Hz}$ and $+5 \% /-0 \%$ at $10,000 \mathrm{~Hz}$ to $32,800 \mathrm{~Hz}$ |  |  |  |  |  |
| Analog settings | Analog adjuster |  | 1 (Setting range: 0 to 255) |  |  |  |  |  |
|  | External analog setting |  | 1 input (Resolution: $1 / 256$, Input range: 0 to 10 V ) |  |  |  |  |  |


|  | Type | M CPU Units |  |  | L CPU Units |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Model |  | CP1L-M60DR-A CP1L-M60DR-D CP1L-M60DT-A CP1L-M60DT-D CP1L-M60DT1-D | CP1L-M40DR-A <br> CP1L-M40DR-D <br> CP1L-M40DT-A <br> CP1L-M40DT-D <br> CP1L-M40DT1-D | CP1L-M30DR-A CP1L-M30DR-D CP1L-M30DT-A CP1L-M30DT-D CP1L-M30DT1-D | CP1L-L20DR-A <br> CP1L-L20DR-D <br> CP1L-L20DT-A <br> CP1L-L20DT-D <br> CP1L-L20DT1-D | CP1L-L14DR-A CP1L-L14DR-D CP1L-L14DT-A CP1L-L14DT-D CP1L-L14DT1-D | CP1L-L10DR-A <br> CP1L-L10DR-D <br> CP1L-L10DT-A <br> CP1L-L10DT-D <br> CP1L-L10DT1-D |
| Serial port | Peripheral USB port | Supported. (1-port USB connector, type B): Special for a Peripheral Device such as the CX-Programmer. (Set the network classification to USB in the Peripheral Device's PLC model setting.) <br> - Serial communications standard: USB 1.1 |  |  |  |  |  |
|  | $\begin{array}{\|l\|} \hline \text { RS-232C } \\ \text { port, } \\ \text { RS-422A/485 } \\ \text { port } \end{array}$ | Ports not provided as standard equipment. (M-type CPU Unit: 2 ports max., L-type CPU Unit: 1 port) <br> The following Option Boards can be mounted: <br> - CP1W-CIF01: One RS-232C port <br> - CP1W-CIF11/CIF12: One RS-422A/485 port <br> Applicable communications modes (same for all of the above ports): Host Link, NT Link (1: N mode), No-protocol, Serial PLC Link Slave, Serial PLC Link Master, Serial Gateway (conversion to CompoWay/F, conversion to Modbus-RTU), peripheral bus (See note.) |  |  |  |  | Not Support |
| Number of tasks |  | 288 (32 cycle execution tasks and 256 interrupt tasks) |  |  |  |  |  |
|  | Scheduled interrupt | 1 (interrupt task 2, fixed) |  |  |  |  |  |
|  | Input interrupt tasks | 6 (interrupt tasks 140 to 145, fixed) |  |  |  | 4 (interrupt tasks 140 to 143, fixed) | 2 (interrupt tasks 140 to 141, fixed) |
|  |  | (High-speed counter interrupts and interrupt tasks specified by external interrupts can also be executed.) |  |  |  |  |  |
| Maximum subroutine number |  | 256 |  |  |  |  |  |
| Maximum jump number |  | 256 |  |  |  |  |  |
| Scheduled interrupts |  | 1 |  |  |  |  |  |
| Clock function |  | Supported. <br> Accuracy (monthly deviation): 4.5 min to -0.5 min (ambient temperature: $55^{\circ} \mathrm{C}$ ), <br> -2.0 min to +2.0 min (ambient temperature: $25^{\circ} \mathrm{C}$ ), <br> -2.5 min to +1.5 min (ambient temperature: $0^{\circ} \mathrm{C}$ ) |  |  |  |  |  |
| Memory Backup | Built-in flash memory | User programs and parameters (such as the PLC Setup) are automatically saved to the flash memory. It is also possible to save and read data memory initial data. The data is automatically transferred to RAM when the power supply is turned ON. (Data memory initial data, however, may or may not be transferred, depending on the selection in the PLC Setup. |  |  |  |  |  |
|  | Battery backup | The HR Area, DM Area, and counter values (flags, PV) are backed up by a battery. Battery model: CJ1W-BAT01 (Built into the CP1L CPU Unit.) <br> Maximum battery service life: 5 years <br> Guaranteed (ambient temperature: $55^{\circ} \mathrm{C}$ ): 13,000 hours (approx. 1.5 years) <br> Effective value (ambient temperature: $25^{\circ} \mathrm{C}$ ): 43,000 hours (approx. 5 years) |  |  |  |  |  |
| Memory Cassette function |  | A CP1W-ME05M Memory Cassette (512K words, optional) can be mounted. It can be used to back up the following data on the CPU Unit's RAM and to transfer the data at startup. <br> - Data saved on Memory Cassette: User programs, parameters (such as the PLC Setup), DM Area, data memory initial data, comment memory (CX-Programmer conversion tables, comments, program indices), and FB program memory. <br> - Writing to Memory Cassette: By operations from the CX-Programmer. <br> - Reading from Memory Cassette: At startup, or by operations from the CX-Programmer. |  |  |  |  |  |

## Note Can be used as Modbus-RTU easy master function.

## 2-2-2 I/O Memory Details

|  | Type | M CPU Units |  |  | L CPU Units |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Model |  | CP1L-M60DR-A CP1L-M60DR-D CP1L-M60DT-A CP1L-M60DT-D CP1L-M60DT1-D | CP1L-M40DR-A <br> CP1L-M40DR-D <br> CP1L-M40DT-A <br> CP1L-M40DT-D <br> CP1L-M40DT1-D | CP1L-M30DR-A <br> CP1L-M30DR-D <br> CP1L-M30DT-A <br> CP1L-M30DT-D <br> CP1L-M30DT1-D | $\begin{aligned} & \text { CP1L-L20DR-A } \\ & \text { CP1L-L20DR-D } \\ & \text { CP1L-L20DT-A } \\ & \text { CP1L-L20DT-D } \\ & \text { CP1L-L20DT1-D } \end{aligned}$ | $\begin{aligned} & \text { CP1L-L14DR-A } \\ & \text { CP1L-L14DR-D } \\ & \text { CP1L-L14DT-A } \\ & \text { CP1L-L14DT-D } \\ & \text { CP1L-L14DT1-D } \end{aligned}$ | CP1L-L10DR-A <br> CP1L-L10DR-D <br> CP1L-L10DT-A <br> CP1L-L10DT-D <br> CP1L-L10DT1-D |
| I/O Areas | Input bits | 36 bits CIO 0.00 to CIO 0.11 CIO 1.00 to CIO 1.11 CIO 2.00 to CIO 2.11 | 24 bits <br> CIO 0.00 to <br> CIO 0.11 <br> CIO 1.00 to <br> CIO 1.11 | 18 bits CIO 0.00 to CIO 0.11 CIO 1.00 to CIO 1.05 | 12 bits CIO 0.00 to CIO 0.11 | 8 bits CIO 0.00 to CIO 0.07 | $\qquad$ $\text { CIO } 0.05$ |
|  | Output bits | 24 bits CIO 100.00 to CIO 100.07 CIO 101.00 to CIO 101.07 CIO 102.00 to CIO 102.07 | 16 bits <br> CIO 100.00 to <br> CIO 100.07 <br> CIO 101.00 to <br> CIO 101.07 | 12 bits CIO 100.00 to CIO 100.07 CIO 101.00 to CIO 100.03 | 8 bits CIO 100.00 to CIO 100.07 | 6 bits CIO 100.00 to CIO 100.05 | 4 bits CIO 100.00 to CIO 100.03 |
|  | 1:1 Link Bit Area | 256 bits (16 words): CIO 3000.00 to CIO 3015.15 (words CIO 3000 to CIO 3015 ) |  |  |  |  | Not Support |
|  | Serial PLC Link Area | 1,440 bits (90 words): ClO 3100.00 to CIO 3189.15 (words CIO 3100 to CIO 3189) |  |  |  |  | Not Support |
|  | Work bits | 4,800 bits ( 300 words): CIO 1200.00 to CIO 1499.15 (words CIO 1200 to CIO 1499) 6,400 bits ( 400 words): CIO 1500.00 to CIO 1899.15 (words CIO 1500 to CIO 1899) 15,360 bits ( 960 words): CIO 2000.00 to CIO 2959.15 (words CIO 2000 to CIO 2959) 9,600 bits ( 600 words): CIO 3200.00 to CIO 3799.15 (words CIO 3200 to CIO 3799) 37,504 bits ( 2,344 words): CIO 3800.00 to CIO 6143.15 (words CIO 3800 to CIO 6143 ) |  |  |  |  |  |
| Work bits |  | 8,192 bits (512 words): W000.00 to W511.15 (words W0 to W511) |  |  |  |  |  |
| TR Area |  | 16 bits: TR0 to TR15 |  |  |  |  |  |
| HR Area |  | 8,192 bits (512 words): H0.00 to H511.15 (words H0 to H511) |  |  |  |  |  |
| AR Area |  | Read-only (Write-prohibited) 7,168 bits (448 words): A0.00 to A447.15 (words A0 to A447) Read/Write 8,192 bits ( 512 words): A448.00 to A959.15 (words A448 to A959) |  |  |  |  |  |
| Timers |  | 4,096 bits: T0 to T4095 |  |  |  |  |  |
| Counters |  | 4,096 bits: C0 to C4095 |  |  |  |  |  |
| DM Area |  | 32 Kwords: D0 to D32767 <br> Note Initial data can be transferred to the CPU Unit's built-in flash memory using the data memory initial data transfer function. A setting in the PLC Setup can be used so that the data in flash memory is transferred to RAM at startup. <br> DM fixed allocation words for Modbus-RTU Easy Master D32200 to D32249 for Serial Port 1, D32300 to D32349 for Serial Port 2 |  |  | 10 Kwords: D0 to D9999 and D32000 to D32767 <br> Note Initial data can be transferred to the CPU Unit's built-in flash memory using the data memory initial data transfer function. A setting in the PLC Setup can be used so that the data in flash memory is transferred to RAM at startup. <br> DM fixed allocation words for Modbus-RTU Easy Master D32300 to D32349 for Serial Port 1 |  |  |
| Data Register Area |  | 16 registers (16 bits): DR0 to DR15 |  |  |  |  |  |
| Index Register Area |  | 16 registers (16 bits): IR0 to IR15 |  |  |  |  |  |
| Task Flag Area |  | 32 flags (32 bits): TK0 to TK31 |  |  |  |  |  |
| Trace Memory |  | 4,000 words (500 samples for the trace data maximum of 31 bits and 6 words.) |  |  |  |  |  |

## 2-2-3 I/O Specifications

## I/O Terminal Blocks of CPU Units with 60 I/O Points <br> Input Terminal Block (Top Block)

AC Power Supply Models


Inputs (CIO 0 )
Inputs (CIO 1)
Inputs (CIO 2)
DC Power Supply Models

| + |  | CO |  | 1 | 03 |  | 05 | 0 |  | 09 | 1 |  | 01 |  |  | 05 |  | 07 |  | 09 |  | 11 |  |  | 03 |  |  |  |  |  | 11 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | NC | $\stackrel{\square}{5}$ | 00 |  |  | 04 |  | 06 | 08 |  | 10 | 00 |  | 02 |  | 4 | 06 |  | 08 |  | 10 |  | 00 | 02 |  | 04 | 06 |  | 8 | 10 |  |

Inputs (CIO 0)
Inputs (CIO 1)
Inputs (CIO 2)
Setting Input Functions Using PLC Setup

| Address |  | Input operation settings |  |  | High-speed counters |  | Origin searches <br> Origin searches enabled for pulse outputs 0 and 1 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bit | Normal inputs | Interrupt inputs (See note.) | Quickresponse inputs | Operation settings: <br> High-speed counters enabled Phase-Z reset |  |  |
|  |  |  |  |  | Single-phase (increment pulse input) | Two-phase (differential phase x4, up/down, or pulse/direction) |  |
| CIO 0 | 00 | Normal input 0 | --- | --- | Counter 0, increment input | Counter 0, A phase, up, or count input | --- |
|  | 01 | Normal input 1 | --- | --- | Counter 1, increment input | Counter 0, B phase, down, or direction input | --- |
|  | 02 | Normal input 2 | --- | --- | Counter 2, increment input | Counter 1, A phase, up, or count input | --- |
|  | 03 | Normal input 3 | --- | --- | Counter 3, increment input | Counter 1, B phase, down, or direction input | --- |
|  | 04 | Normal input 4 | Interrupt input 0 | Quickresponse input 0 | Counter 0, phase-Z/reset input | Counter 0, phase-Z reset input | --- |
|  | 05 | Normal input 5 | Interrupt input 1 | Quickresponse input 1 | Counter 1, phase-Z reset input | Counter 1, phase-Z reset input | --- |
|  | 06 | Normal input 6 | Interrupt input 2 | Quickresponse input 2 | Counter 2, phase-Z reset input | --- | Pulse output 0: Origin input signal |
|  | 07 | Normal input 7 | Interrupt input 3 | Quickresponse input 3 | Counter 3, phase-Z reset input | --- | Pulse output 1: Origin input signal |
|  | 08 | Normal input 8 | Interrupt input 4 | Quickresponse input 4 | --- | --- | --- |
|  | 09 | Normal input 9 | Interrupt input 5 | Quickresponse input 5 | --- | --- | --- |
|  | 10 | Normal input 10 | --- | --- | --- | --- | Pulse output 0: Origin proximity input signal |
|  | 11 | Normal input 11 | --- | --- | --- | --- | Pulse output 1: Origin proximity input signal |


| Address |  | Input operation settings |  |  | High-speed counters |  | Origin searches <br> Origin searches enabled for pulse outputs 0 and 1 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bit | Normal inputs | Interrupt inputs (See note.) | Quickresponse inputs | Operation settings: <br> High-speed counters enabled Phase-Z reset |  |  |
|  |  |  |  |  | Single-phase (increment pulse input) | Two-phase (differential phase x4, up/down, or pulse/direction) |  |
| CIO 1 | 00 | Normal input 12 | --- | --- | --- | --- | --- |
|  | 01 | Normal input 13 | --- | --- | --- | --- | --- |
|  | 02 | Normal input 14 | --- | --- | --- | --- | --- |
|  | 03 | Normal input 15 | --- | --- | --- | --- | --- |
|  | 04 | Normal input 16 | --- | --- | --- | --- | --- |
|  | 05 | Normal input 17 | --- | --- | --- | --- | --- |
|  | 06 | Normal input 18 | --- | --- | --- | --- | --- |
|  | 07 | Normal input 19 | --- | --- | --- | --- | --- |
|  | 08 | Normal input 20 | --- | --- | --- | --- | --- |
|  | 09 | Normal input 21 | --- | --- | --- | --- | --- |
|  | 10 | Normal input 22 | --- | --- | --- | --- | --- |
|  | 11 | Normal input 23 | --- | --- | --- | --- | --- |
| CIO 2 | 00 | Normal input 24 | --- | --- | --- | --- | --- |
|  | 01 | Normal input 25 | --- | --- | --- | --- | --- |
|  | 02 | Normal input 26 | --- | --- | --- | --- | --- |
|  | 03 | Normal input 27 | --- | --- | --- | --- | --- |
|  | 04 | Normal input 28 | --- | --- | --- | --- | --- |
|  | 05 | Normal input 29 | --- | --- | --- | --- | --- |
|  | 06 | Normal input 30 | --- | --- | --- | --- | --- |
|  | 07 | Normal input 31 | --- | --- | --- | --- | --- |
|  | 08 | Normal input 32 | --- | --- | --- | --- | --- |
|  | 09 | Normal input 33 | --- | --- | --- | --- | --- |
|  | 10 | Normal input 34 | --- | --- | --- | --- | --- |
|  | 11 | Normal input 35 | --- | --- | --- | --- | --- |

## Output Terminal Block (Bottom Block)

AC Power Supply Models


CIO 100

CIO 102

DC Power Supply Models


CIO 100
CIO 101
CIO 102
Setting Output Functions Using Instructions and PLC Setup

| Address |  | When the instructions to the right are not executed | When a pulse output instruction (SPED, ACC, PLS2, or ORG) is executed |  | When origin searches are enabled in the PLC Setup, and an origin search is executed with ORG instruction | When the PWM instruction is executed |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bit | Normal outputs | Fixed duty ratio pulse output |  |  | Variable duty ratio pulse output |
|  |  |  | CW/CCW | Pulse plus direction | + When the origin search function is used | PWM output |
| CIO 100 | 00 | Normal output 0 | $\begin{aligned} & \text { Pulse output } 0 \\ & \text { (CW) } \end{aligned}$ | Pulse output 0 (pulse) | --- | --- |
|  | 01 | Normal output 1 | Pulse output 0 (CCW) | Pulse output 0 (direction) | --- | PWM output 0 |
|  | 02 | Normal output 2 | Pulse output 1 (CW) | Pulse output 1 (pulse) | --- | --- |
|  | 03 | Normal output 3 | Pulse output 1 (CCW) | Pulse output 1 (direction) | --- | PWM output 1 |
|  | 04 | Normal output 4 | --- | --- | Origin search 0 (Error counter reset output) | --- |
|  | 05 | Normal output 5 | --- | --- | Origin search 1 (Error counter reset output) | --- |
|  | 06 | Normal output 6 | --- | --- | --- | --- |
|  | 07 | Normal output 7 | --- | --- | --- | --- |
| CIO 101 | 00 | Normal output 8 | --- | --- | --- | --- |
|  | 01 | Normal output 9 | --- | --- | -- | --- |
|  | 02 | Normal output 10 | --- | --- | --- | --- |
|  | 03 | Normal output 11 | --- | --- | --- | --- |
|  | 04 | Normal output 12 | --- | --- | --- | --- |
|  | 05 | Normal output 13 | --- | --- | --- | --- |
|  | 06 | Normal output 14 | --- | --- | --- | --- |
|  | 07 | Normal output 15 | --- | --- | --- | --- |


| Address |  | When the instructions to the right are not executed | When a pulse output instruction (SPED, ACC, PLS2, or ORG) is executed |  | When origin searches are enabled in the PLC Setup, and an origin search is executed with ORG instruction | When the PWM instruction is executed |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bit | Normal outputs | Fixed duty ratio pulse output |  |  | Variable duty ratio pulse output |
|  |  |  | CW/CCW | Pulse plus direction | + When the origin search function is used | PWM output |
| CIO 102 | 00 | Normal output 16 | --- | --- | --- | --- |
|  | 01 | Normal output 17 | --- | --- | --- | --- |
|  | 02 | Normal output 18 | --- | --- | -- | --- |
|  | 03 | Normal output 19 | --- | --- | --- | --- |
|  | 04 | Normal output 20 | --- | --- | --- | --- |
|  | 05 | Normal output 21 | --- | --- | --- | --- |
|  | 06 | Normal output 22 | --- | --- | --- | --- |
|  | 07 | Normal output 23 | --- | --- | --- | --- |

## I/O Terminal Blocks of CPU Units with 40 I/O Points

## Input Terminal Block (Top Block)

AC Power Supply Models


Inputs (CIO 0 )
Inputs (CIO 1)
DC Power Supply Models


Inputs (CIO 0)
Inputs (CIO 1)

## Setting Input Functions Using PLC Setup

| Address |  | Input operation settings |  |  | High-speed counters |  | Origin searches <br> Origin searches enabled for pulse outputs 0 and 1 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bit | Normal inputs | Interrupt inputs (See note.) | Quickresponse inputs | Operation settings: <br> High-speed counters enabled Phase-Z reset |  |  |
|  |  |  |  |  | Single-phase (increment pulse input) | Two-phase (differential phase x4, up/down, or pulse/direction) |  |
| ClO 0 | 00 | Normal input 0 | --- | --- | Counter 0, increment input | Counter 0, A phase, up, or count input | --- |
|  | 01 | Normal input 1 | --- | --- | Counter 1, increment input | Counter 0, B phase, down, or direction input | --- |
|  | 02 | Normal input 2 | --- | --- | Counter 2, increment input | Counter 1, A phase, up, or count input | --- |
|  | 03 | Normal input 3 | --- | -- | Counter 3, increment input | Counter 1, B phase, down, or direction input | --- |
|  | 04 | Normal input 4 | Interrupt input 0 | Quickresponse input 0 | Counter 0, phase-Z/reset input | Counter 0, phase-Z reset input | --- |
|  | 05 | Normal input 5 | Interrupt input 1 | Quickresponse input 1 | Counter 1, phase-Z reset input | Counter 1, phase-Z reset input | --- |
|  | 06 | Normal input 6 | Interrupt input 2 | Quickresponse input 2 | Counter 2, phase-Z reset input | --- | Pulse output 0: Origin input signal |
|  | 07 | Normal input 7 | Interrupt input 3 | Quickresponse input 3 | Counter 3, phase-Z reset input | --- | Pulse output 1: Origin input signal |
|  | 08 | Normal input 8 | Interrupt input 4 | Quickresponse input 4 | --- | --- | --- |
|  | 09 | Normal input 9 | Interrupt input 5 | Quickresponse input 5 | --- | --- | --- |
|  | 10 | Normal input 10 | --- | --- | --- | --- | Pulse output 0: Origin proximity input signal |
|  | 11 | Normal input 11 | --- | --- | --- | --- | Pulse output 1: Origin proximity input signal |


| Address |  | Input operation settings |  |  | High-speed counters |  | Origin searches <br> Origin searches enabled for pulse outputs 0 and 1 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bit | Normal inputs | Interrupt inputs (See note.) | Quickresponse inputs | Operation settings: <br> High-speed counters enabled Phase-Z reset |  |  |
|  |  |  |  |  | Single-phase (increment pulse input) | Two-phase (differential phase x4, up/down, or pulse/direction) |  |
| CIO 1 | 00 | Normal input 12 | --- | --- | --- | --- | --- |
|  | 01 | Normal input 13 | --- | --- | --- | --- | --- |
|  | 02 | Normal input 14 | --- | --- | --- | --- | --- |
|  | 03 | Normal input 15 | --- | --- | --- | --- | --- |
|  | 04 | Normal input 16 | --- | --- | --- | --- | --- |
|  | 05 | Normal input 17 | --- | --- | --- | --- | --- |
|  | 06 | Normal input 18 | --- | --- | --- | --- | --- |
|  | 07 | Normal input 19 | --- | --- | --- | --- | --- |
|  | 08 | Normal input 20 | --- | --- | --- | --- | --- |
|  | 09 | Normal input 21 | --- | --- | --- | --- | --- |
|  | 10 | Normal input 22 | --- | --- | --- | --- | --- |
|  | 11 | Normal input 23 | --- | --- | --- | --- | --- |

## Output Terminal Block Arrangement (Bottom Block)

AC Power Supply Models


CIO 100
CIO 101
DC Power Supply Models


CIO 100
CIO 101

Setting Output Functions Using Instructions and PLC Setup

| Address |  | When the instructions to the right are not executed | When a pulse output instruction (SPED, ACC, PLS2, or ORG) is executed |  | When origin searches are enabled in the PLC Setup, and an origin search is executed with ORG instruction | When the PWM instruction is executed |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bit | Normal outputs | Fixed duty ratio pulse output |  |  | Variable duty ratio pulse output |
|  |  |  | CW/CCW | Pulse plus direction | + When the origin search function is used | PWM output |
| CIO 100 | 00 | Normal output 0 | $\begin{array}{\|l\|} \hline \text { Pulse output } 0 \\ \text { (CW) } \end{array}$ | Pulse output 0 (pulse) | -- | --- |
|  | 01 | Normal output 1 | Pulse output 0 (CCW) | Pulse output 0 (direction) | --- | PWM output 0 |
|  | 02 | Normal output 2 | Pulse output 1 (CW) | Pulse output 1 (pulse) | --- | --- |
|  | 03 | Normal output 3 | Pulse output 1 (CCW) | Pulse output 1 (direction) | --- | PWM output 1 |
|  | 04 | Normal output 4 | --- | --- | Origin search 0 (Error counter reset output) | --- |
|  | 05 | Normal output 5 | --- | --- | Origin search 1 (Error counter reset output) | --- |
|  | 06 | Normal output 6 | --- | --- | -- | --- |
|  | 07 | Normal output 7 | --- | --- | --- | --- |
| CIO 101 | 00 | Normal output 8 | --- | --- | --- | --- |
|  | 01 | Normal output 9 | --- | --- | --- | --- |
|  | 02 | Normal output 10 | --- | --- | --- | --- |
|  | 03 | Normal output 11 | --- | --- | --- | --- |
|  | 04 | Normal output 12 | --- | --- | --- | --- |
|  | 05 | Normal output 13 | --- | --- | --- | --- |
|  | 06 | Normal output 14 | --- | --- | --- | --- |
|  | 07 | Normal output 15 | --- | --- | --- | --- |

## I/O Terminal Blocks of CPU Units with 30 I/O Points

Input Terminal Block (Top Block)
AC Power Supply Models


Inputs (CIO 0 )
Inputs (CIO 1)
DC Power Supply Models


Setting Input Functions Using PLC Setup

| Address |  | Input operation settings |  |  | High-speed counters |  | Origin searches Origin searches enabled for pulse outputs 0 and 1 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bit | Normal inputs | Interrupt inputs (See note.) | Quickresponse inputs | Operation settings: <br> High-speed counters enabled Phase-Z reset |  |  |
|  |  |  |  |  | Single-phase (increment pulse input) | Two-phase (differential phase x4, up/down, or pulse/direction) |  |
| ClO 0 | 00 | Normal input 0 | --- | --- | Counter 0, increment input | Counter 0, A phase, up, or count input | --- |
|  | 01 | Normal input 1 | --- | --- | Counter 1, increment input | Counter 0, B phase, down, or direction input | --- |
|  | 02 | Normal input 2 | --- | --- | Counter 2, increment input | Counter 1, A phase, up, or count input | --- |
|  | 03 | Normal input 3 | --- | --- | Counter 3, increment input | Counter 1, B phase, down, or direction input | --- |
|  | 04 | Normal input 4 | Interrupt input 0 | Quickresponse input 0 | Counter 0, phase-Z/reset input | Counter 0, phase-Z reset input | --- |
|  | 05 | Normal input 5 | Interrupt input 1 | Quickresponse input 1 | Counter 1, phase-Z reset input | Counter 1, phase-Z reset input | --- |
|  | 06 | Normal input 6 | Interrupt input 2 | Quickresponse input 2 | Counter 2, phase-Z reset input | --- | Pulse output 0: Origin input signal |
|  | 07 | Normal input 7 | Interrupt input 3 | Quickresponse input 3 | Counter 3, phase-Z reset input | --- | Pulse output 1: Origin input signal |
|  | 08 | Normal input 8 | Interrupt input 4 | Quickresponse input 4 | --- |  | --- |
|  | 09 | Normal input 9 | Interrupt input 5 | Quickresponse input 5 | --- |  | --- |
|  | 10 | Normal input 10 | --- | --- | --- |  | Pulse output 0: Origin proximity input signal |
|  | 11 | Normal input 11 | --- | --- | --- |  | Pulse output 1: Origin proximity input signal |
| CIO 1 | 00 | Normal input 12 | --- | --- | --- |  |  |
|  | 01 | Normal input 13 | --- | --- | --- |  |  |
|  | 02 | Normal input 14 | --- | --- | --- |  |  |
|  | 03 | Normal input 15 | --- | --- | --- |  |  |
|  | 04 | Normal input 16 | --- | --- | --- |  |  |
|  | 05 | Normal input 17 | --- | --- | --- |  |  |

## Output Terminal Block (Bottom Block)

AC Power Supply Models


DC Power Supply Models


CIO 100
CIO 101

## Setting Output Functions Using Instructions and PLC Setup

| Address |  | When the instructions to the right are not executed | When a pulse output instruction (SPED, ACC, PLS2, or ORG) is executed |  | When origin searches are enabled in the PLC Setup, and an origin search is executed with ORG instruction | When the PWM instruction is executed |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bit | Normal outputs | Fixed duty ratio pulse output |  |  | Variable duty ratio pulse output |
|  |  |  | CW/CCW | Pulse plus direction | + When the origin search function is used | PWM output |
| CIO 100 | 00 | Normal output 0 | Pulse output 0 (CW) | Pulse output 0 (pulse) | --- | --- |
|  | 01 | Normal output 1 | Pulse output 0 (CCW) | Pulse output 0 (direction) | --- | PWM output 0 |
|  | 02 | Normal output 2 | Pulse output 1 (CW) | Pulse output 1 (pulse) | --- | --- |
|  | 03 | Normal output 3 | Pulse output 1 (CCW) | Pulse output 1 (direction) | --- | PWM output 1 |
|  | 04 | Normal output 4 | --- | -- | Origin search 0 (Error counter reset output) | --- |
|  | 05 | Normal output 5 | --- | --- | Origin search 1 (Error counter reset output) | --- |
|  | 06 | Normal output 6 | --- | --- | --- | --- |
|  | 07 | Normal output 7 | --- | --- | --- | --- |
| CIO 101 | 00 | Normal output 8 | --- | --- | --- | --- |
|  | 01 | Normal output 9 | --- | --- | --- | --- |
|  | 02 | Normal output 10 | --- | --- | --- | --- |
|  | 03 | Normal output 11 | --- | --- | --- | --- |

## I/O Terminal Blocks of CPU Units with 20 I/O Points

## Input Terminal Block (Top Block)

AC Power Supply Models


Inputs (CIO 0 )
DC Power Supply Models


Setting Input Functions Using PLC Setup

| Address |  | Input operation settings |  |  | High-speed countersOperation settings:High-speed counters enabledPhase-Z reset |  | Origin searches Origin searches enabled for pulse outputs 0 and 1 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bit | Normal inputs | Interrupt inputs (See note.) | Quickresponse inputs |  |  |  |
|  |  |  |  |  | Single-phase (increment pulse input) | Two-phase (differential phase x4, up/down, or pulse/direction) |  |
| CIO 0 | 00 | Normal input 0 | --- | --- | Counter 0, increment input | Counter 0, A phase, up, or count input | --- |
|  | 01 | Normal input 1 | --- | --- | Counter 1, increment input | Counter 0, B phase, down, or direction input | --- |
|  | 02 | Normal input 2 | --- | -- | Counter 2, increment input | Counter 1, A phase, up, or count input | --- |
|  | 03 | Normal input 3 | --- | --- | Counter 3, increment input | Counter 1, B phase, down, or direction input | --- |
|  | 04 | Normal input 4 | Interrupt input 0 | Quickresponse input 0 | Counter 0, phase-Z/reset input | Counter 0, phase-Z reset input | --- |
|  | 05 | Normal input 5 | Interrupt input 1 | Quickresponse input 1 | Counter 1, phase-Z reset input | Counter 1, phase-Z reset input | --- |
|  | 06 | Normal input 6 | Interrupt input 2 | Quickresponse input 2 | Counter 2, phase-Z reset input | --- | Pulse output 0: Origin input signal |
|  | 07 | Normal input 7 | Interrupt input 3 | Quickresponse input 3 | Counter 3, phase-Z reset input | --- | Pulse output 1: Origin input signal |
|  | 08 | Normal input 8 | Interrupt input 4 | Quickresponse input 4 | --- |  | --- |
|  | 09 | Normal input 9 | Interrupt input 5 | Quickresponse input 5 | --- |  | --- |
|  | 10 | Normal input 10 | --- | --- | --- |  | Pulse output 0: Origin proximity input signal |
|  | 11 | Normal input 11 | --- | --- | --- |  | Pulse output 1: Origin proximity input signal |

## Output Terminal Block (Bottom Block)

AC Power Supply Models


CIO 100


CIO 100

## Setting Output Functions Using Instructions and PLC Setup

| Address |  | When the instructions to the right are not executed | When a pulse output instruction (SPED, ACC, PLS2, or ORG) is executed |  | When origin searches are enabled in PLC Setup, and an origin search is executed with ORG instruction | When the PWM instruction is executed |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bit | Normal outputs | Fixed duty ratio pulse output |  |  | Variable duty ratio pulse output |
|  |  |  | CW/CCW | Pulse plus direction | + When the origin search function is used | PWM output |
| CIO 100 | 00 | Normal output 0 | $\begin{aligned} & \hline \text { Pulse output } 0 \\ & \text { (CW) } \end{aligned}$ | Pulse output 0 (pulse) | --- | --- |
|  | 01 | Normal output 1 | Pulse output 0 (CCW) | Pulse output 0 (direction) | --- | PWM output 0 |
|  | 02 | Normal output 2 | Pulse output 1 (CW) | Pulse output 1 (pulse) | --- | --- |
|  | 03 | Normal output 3 | Pulse output 1 (CCW) | Pulse output 1 (direction) | --- | PWM output 1 |
|  | 04 | Normal output 4 | --- | --- | Origin search 0 (Error counter reset output) | --- |
|  | 05 | Normal output 5 | --- | --- | Origin search 1 (Error counter reset output) | --- |
|  | 06 | Normal output 6 | --- | --- | --- | --- |
|  | 07 | Normal output 7 | --- | --- | --- | --- |

## I/O Terminal Blocks of CPU Units with 14 I/O Points

## Input Terminal Block (Top Block)

AC Power Supply Models


Inputs (CIO 0 )

DC Power Supply Models


Inputs (CIO 0)

Setting Input Functions Using PLC Setup

| Address |  | Input operation settings |  |  | High-speed counters |  | Origin searches <br> Origin searches enabled for pulse outputs 0 and 1 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bit | Normal inputs | Interrupt inputs (See note.) | Quickresponse inputs | Operation settings: <br> High-speed counters enabled Phase-Z reset |  |  |
|  |  |  |  |  | Single-phase (increment pulse input) | Two-phase (differential phase x4, up/down, or pulse/direction) |  |
| CIO 0 | 00 | Normal input 0 | --- | --- | Counter 0, increment input | Counter 0, A phase, up, or count input | --- |
|  | 01 | Normal input 1 | --- | --- | Counter 1, increment input | Counter 0, B phase, down, or direction input | --- |
|  | 02 | Normal input 2 | --- | --- | Counter 2, increment input | Counter 1, A phase, up, or count input | Pulse output 0: Origin proximity input signal |
|  | 03 | Normal input 3 | --- | --- | Counter 3, increment input | Counter 1, B phase, down, or direction input | Pulse output 1: Origin proximity input signal |
|  | 04 | Normal input 4 | Interrupt input 0 | Quickresponse input 0 | Counter 0, phase-Z/reset input | Counter 0, phase-Z or reset input | --- |
|  | 05 | Normal input 5 | Interrupt input 1 | Quickresponse input 1 | Counter 1, phase-Z reset input | Counter 1, phase-Z or reset input | --- |
|  | 06 | Normal input 6 | Interrupt input 2 | Quickresponse input 2 | Counter 2, phase-Z reset input | --- | Pulse output 0: Origin input signal |
|  | 07 | Normal input 7 | Interrupt input 3 | Quickresponse input 3 | Counter 3, phase-Z reset input | --- | Pulse output 1: Origin input signal |

## Output Terminal Block (Bottom Block)

AC Power Supply Models


CIO 100

DC Power Supply Models


CIO 100

Setting Functions Using Instructions and PLC Setup

| Address |  | When the instructions to the right are not executed | When a pulse output instruction (SPED, ACC, PLS2, or ORG) is executed |  | When origin searches are enabled in PLC Setup, and an origin search is executed with ORG instruction | When the PWM instruction is executed |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bit | Normal outputs | Fixed duty ratio pulse output |  |  | Variable duty ratio pulse output |
|  |  |  | CW/CCW | Pulse plus direction | + When the origin search function is used | PWM output |
| CIO 100 | 00 | Normal output 0 | $\begin{aligned} & \hline \text { Pulse output } \\ & 0 \text { (CW) } \end{aligned}$ | Pulse output 0 (pulse) | --- | --- |
|  | 01 | Normal output 1 | $\begin{array}{\|l} \hline \text { Pulse output } \\ 0 \text { (CCW) } \\ \hline \end{array}$ | Pulse output 0 (direction) | --- | PWM output 0 |
|  | 02 | Normal output 2 | $\begin{array}{\|l} \hline \text { Pulse output } \\ 1 \text { (CW) } \end{array}$ | Pulse output 1 (pulse) | --- | --- |
|  | 03 | Normal output 3 | Pulse output 1 (CCW) | Pulse output 1 (direction) | --- | PWM output 1 |
|  | 04 | Normal output 4 | --- | --- | Origin search 0 (Error counter reset output) | --- |
|  | 05 | Normal output 5 | --- | --- | Origin search 1 (Error counter reset output) | --- |

## I/O Terminal Blocks of CPU Units with 10 I/O Points

## Input Terminal Block (Top Block)

AC Power Supply Models


Inputs (CIO 0)

DC Power Supply Models


Inputs (CIO 0)

Setting Input Functions Using PLC Setup

| Address |  | Input operation settings |  |  | High-speed countersOperation settings:High-speed counters enabledPhase-Z reset |  | Origin searches Origin searches enabled for pulse outputs 0 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bit | Normal inputs | Interrupt inputs (See note.) | Quickresponse inputs |  |  |  |
|  |  |  |  |  | Single-phase (increment pulse input) | Two-phase (differential phase x4, up/down, or pulse/direction) |  |
| ClO 0 | 00 | Normal input 0 | --- | --- | Counter 0, increment input | Counter 0, A phase, up, or count input | --- |
|  | 01 | Normal input 1 | --- | --- | Counter 1, increment input | Counter 0, B phase, down, or direction input | --- |
|  | 02 | Normal input 2 | --- | --- | Counter 2, increment input | Counter 1, A phase, up, or count input | --- |
|  | 03 | Normal input 3 | --- | --- | Counter 3, increment input | Counter 1, B phase, down, or direction input | Pulse output 0: Origin proximity input signal |
|  | 04 | Normal input 4 | Interrupt input 0 | Quickresponse input 0 | Counter 0, phase-Z/reset input | Counter 0, phase-Z or reset input | --- |
|  | 05 | Normal input 5 | Interrupt input 1 | Quickresponse input 1 | Counter 1, phase-Z reset input | Counter 1, phase-Z or reset input | Pulse output 0: Origin input signal |

## I/O Terminal Blocks of CPU Units with 10 I/O Points

## Output Terminal Block (Bottom Block)



CIO 100

DC Power Supply Models


CIO 100

Setting Functions Using Instructions and PLC Setup

| Address |  | When the instructions to the right are not executed | When a pulse output instruction (SPED, ACC, PLS2, or ORG) is executed |  | When origin searches are enabled in PLC Setup, and an origin search is executed with ORG instruction | When the PWM instruction is executed |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bit | Normal outputs | Fixed duty ratio pulse output |  |  | Variable duty ratio pulse output |
|  |  |  | CW/CCW | Pulse plus direction | + When the origin search function is used | PWM output |
| CIO 100 | 00 | Normal output 0 | Pulse output 0 (CW) | Pulse output 0 (pulse) | --- | --- |
|  | 01 | Normal output 1 | Pulse output 0 (CCW) | Pulse output 0 (direction) | --- | PWM output 0 |
|  | 02 | Normal output 2 | Pulse output 1 (CW) | Pulse output 1 (pulse) | --- | --- |
|  | 03 | Normal output 3 | Pulse output 1 (CCW) | Pulse output 1 (direction) | Origin search 0 (Error counter reset output) | PWM output 1 |

## Input Specifications

Normal Inputs

| Item | Specification |  |  |
| :---: | :---: | :---: | :---: |
|  | High-speed Counter Inputs | Interrupt Inputs and Quick-response Inputs | Normal inputs |
|  | CIO 0.00 to ClO 0.03 | CIO 0.04 to CIO 0.09 (See note 1.) | CIO 0.10 to CIO 0.11 and CIO 1.00 to 1.11 (See note 2.) |
| Input voltage | $24 \mathrm{VDC}^{+10 \%} /_{-15 \%}$ |  |  |
| Applicable inputs | 2-wire and 3-wire sensors |  |  |
| Input impedance | $3.0 \mathrm{k} \Omega$ | $3.0 \mathrm{k} \Omega$ | $4.7 \mathrm{k} \Omega$ |
| Input current | 7.5 mA typical | 7.5 mA typical | 5 mA typical |
| ON voltage | 17.0 VDC min. | 17.0 VDC min. | 14.4 VDC min. |
| OFF voltage/current | 1 mA max. at 5.0 VDC max. | 1 mA max. at 5.0 VDC max. | 1 mA max. at 5.0 VDC max. |
| ON delay | $2.5 \mu$ s max. | $50 \mu$ s max. | $1 \mathrm{~ms} \mathrm{max}$. (See note 3.) |



Note (1) Hlgh-speed counter inputs, interrupt inputs, and quick-response inputs can also be used as normal inputs.
(2) The bits that can be used depend on the model of CPU Unit.
(3) The response time is the hardware delay value. The delay set in the PLC Setup ( 0 to 32 ms , default: 8 ms ) must be added to this value.

High-speed Counter Inputs

| Bit | Differential <br> phase mode | Pulse plus <br> direction input <br> mode | Up/down input <br> mode | Increment <br> mode |
| :--- | :--- | :--- | :--- | :--- |
| CIO 0.00, <br> CIO 0.02 | A-phase pulse <br> input | Pulse input | Increment pulse <br> input | Increment pulse <br> input |
| CIO 0.01, <br> CIO 0.03 | B-phase pulse <br> input | Direction input | Decrement <br> pulse input | Normal input |
| CIO 0.04, <br> CIO 0.05 | Z-phase pulse input or hardware reset input (Can be used as ordinary <br> inputs when high-speed counter is not being used.) |  |  |  |
| Max. count <br> frequency $50 \mathrm{kHz} \mathrm{(4} \mathrm{\times)}$ | 100 kHz |  |  |  |

Input Bits for High-speed Counters

| Counter | Single phase | Phase A | Phase B | Phase Z |
| :--- | :--- | :--- | :--- | :--- |
| High-speed counter 0 | CIO 0.00 | CIO 0.00 | CIO 0.01 | CIO 0.04 |
| High-speed counter 1 | CIO 0.01 | CIO 0.02 | CIO 0.03 | CIO 0.05 |
| High-speed counter 2 | CIO 0.02 | --- | --- | -- |
| High-speed counter 3 | CIO 0.03 | --- | --- | --- |

Pulse plus direction input mode, Increment mode Up/down input mode


Input bits: CIO 0.04 to ClO 0.09

Differential phase mode

$\mathrm{T}_{1}, \mathrm{~T}_{2}, \mathrm{~T}_{3}, \mathrm{~T}_{4}: 2.5 \mu \mathrm{~s} \mathrm{~min}$.


With CPU Units with 20, 30, 40 or 60 I/O points, the six input bits from CIO 0.04 to CIO 0.09 can be used as either normal inputs or as interrupt or quickresponse inputs depending on the settings in the PLC Setup. With CPU Units with 14 I/O points, the four input bits from CIO 0.04 to CIO 0.07 can be used as either normal inputs or as interrupt or quick-response inputs. With CPU Units with 10 I/O points, the two input bits from CIO 0.04 to CIO 0.05 can be used as either normal inputs or as interrupt or quick-response inputs.

| Input bit |  |  | Interrupt <br> inputs | Quick-response <br> inputs |
| :--- | :--- | :--- | :--- | :--- |
| CPU Units with <br> 20, 30, 40 or <br> 60 I/O points | CPU Units <br> with 14 I/O <br> points | CPU Units <br> with 10 I/O <br> points |  |  |
| CIO 0.04 | CIO 0.04 | CIO 0.04 | Interrupt input 0 | Quick-response input 0 |
| CIO 0.05 | CIO 0.05 | CIO 0.05 | Interrupt input 1 | Quick-response input 1 |
| CIO 0.06 | CIO 0.06 | --- | Interrupt input 2 | Quick-response input 2 |
| CIO 0.07 | CIO 0.07 | --- | Interrupt input 3 | Quick-response input 3 |
| CIO 0.08 | --- | --- | Interrupt input 4 | Quick-response input 4 |
| CIO 0.09 | --- | --- | Interrupt input 5 | Quick-response input 5 |

## Output Specifications

## Relay Outputs

| Item |  |  | Specification |
| :---: | :---: | :---: | :---: |
| Max. switching capacity |  |  | 2 A, 250 VAC $(\cos \phi=1)$ <br> $2 \mathrm{~A}, 24 \mathrm{VDC}$ (4 A/common) |
| Min. switching capacity |  |  | $10 \mathrm{~mA}, 5 \mathrm{VDC}$ |
| Service life of relay | Electrical | Resistive load | 100,000 operations (24 VDC) |
|  |  | Inductive load | 48,000 operations (250 VAC, $\cos \phi=0.4)$ |
|  | Mechanic |  | 20,000,000 operations |
| ON delay |  |  | 15 ms max. |


| Item | Specification |
| :---: | :---: |
| OFF delay | 15 ms max. |
| Circuit configuration |  |

(1) Under the worst conditions, the service life of output contacts is as shown above. The service life of relays is as shown in the following diagram as a guideline.

(2) There are restrictions imposed by the ambient temperature.

CPU Units with Relay Outputs (CP1L- $\square \square \square$ DR-D)
Relay Output Load Current Derating Curves for CPU Units and Expansion I/O Units


Note The above restrictions, apply to the relay output load current from the CPU Unit even if Expansion I/O Units are not connected.

## Transistor Outputs (Sinking or Sourcing)

## Normal Outputs

| Item | Specification |  |
| :---: | :---: | :---: |
|  | CIO 100.00 to CIO 100.03 | CIO 100.04 to CIO 100.07 (See note 3.) |
| Max. switching capacity | 4.5 to $30 \mathrm{VDC}, 300 \mathrm{~mA} /$ output, $0.9 \mathrm{~A} /$ common, | M60D $\square$-D 5.4 A/Unit M40D $\square$-D 3.6 A/Unit M30D $\square$-D 2.7 A/Unit L20D $\square$-D 1.8 A/Unit L14D $\square$-D 1.5 A/Unit (See note 2.) L10D $\square$-D 0.9 A/Unit (See note 2.) |
| Min. switching capacity | 4.5 to 30 VDC, 1 mA |  |
| Leakage current | 0.1 mA max. |  |
| Residual voltage | 0.6 V max. | 1.5 V max. |
| ON delay | 0.1 ms max. |  |
| OFF delay | 0.1 ms max. | 1 ms max. |
| Fuse | 1 fuse/output (See note 1.) |  |
| Circuit configuration | - Normal outputs CIO 100.00 to CIO 100.03 (Sinking Outputs) <br> - Normal outputs CIO 100.00 to CIO 100.03 (Sourcing Outputs) | - Normal outputs CIO 100.04 to CIO 101.07 (Sourcing Outputs) |

Note (1) The fuse cannot be replaced by the user.
(2) Also do not exceed 0.9 A for the total of CIO 100.00 to CIO 100.03 , which are different common.
(3) The bits that can be used depend on the model of the CPU Unit.

Caution Do not connect a load to an output terminal or apply a voltage in excess of the maximum switching capacity.

## Pulse Outputs (CIO 100.00 to CIO 100.03)

| Item | Specification |
| :---: | :---: |
| Max. switching capacity | $30 \mathrm{~mA} / 4.75$ to 26.4 VDC |
| Min. switching capacity | $7 \mathrm{~mA} / 4.75$ to 26.4 VDC |
| Max. output frequency | 100 kHz |
| Output waveform | OFF ON <br> The OFF and ON refer to the output transistor. The output transistor is ON at level " L ". |

Note (1) The load for the above values is assumed to be the resistance load, and does not take into account the impedance for the connecting cable to the load.
(2) Due to distortions in pulse waveforms resulting from connecting cable impedance, the pulse widths in actual operation may be smaller than the values shown above.

PWM Outputs (CIO 100.01 and CIO 100.03)

| Item | Specification |  |
| :--- | :--- | :---: |
| Max. switching capacity | $30 \mathrm{~mA} / 4.75$ to 26.4 VDC |  |
| Max. output frequency | 1 kHz |  |
| PWM output accuracy | For ON duty $+1 \%,-0 \%: 10 \mathrm{kHz}$ output |  |
| For ON duty $+5 \%,-0 \%: 0$ to 32.8 kHz output |  |  |

The OFF and ON refer to the output transistor. The output transistor is ON at level "L".

## 2-2-4 CP/CPM1A-series Expansion I/O Unit I/O Specifications Input Specifications (CP1W-40EDR/40EDT/40EDT1/20EDR1/20EDT/20EDT1/8ED and CPM1A-40EDR/40EDT/20EDR1/20EDT/20EDT1/8ED)

| Item | Specification |
| :--- | :--- |
| Input voltage | $24 \mathrm{VDC}{ }^{+10 \%}{ }_{-15 \%}$ |
| Input impedance | $4.7 \mathrm{k} \Omega$ |
| Input current | 5 mA typical |
| ON voltage | 14.4 VDC min. |
| OFF voltage | 5.0 VDC max. |
| ON delay | 1 ms max. (See note 1.) |
| OFF delay | 1 ms max. (See note 1.) |
| Circuit configuration |  |

(1) The response time is the hardware delay value. The delay set in the PLC Setup ( 0 to 32 ms , default: 8 ms ) must be added to this value. For the CP1W-40EDR/EDT/EDT1 and CPM1A-40EDR/EDT/EDT1, a fixed value of 16 ms must be added.
(2) Do not apply voltage in excess of the rated voltage to the input terminal.

## Output Specifications

Relay Outputs (CP1W-40EDR/32ER/20EDR1/16ER/8ER and CPM1A-40EDR/20EDR1/16ER/8ER)

| Item |  |  | Specification |
| :---: | :---: | :---: | :---: |
| Max. switching capacity |  |  | 2 A, 250 VAC $(\cos \phi=1)$, 2 A, 24 VDC (4 A/common) |
| Min. switching capacity |  |  | 5 VDC, 10 mA |
| Service life of relay (See note.) | Electrical | Resistive load | 150,000 operations (24 VDC) |
|  |  | Inductive load | 100,000 operations (240 VAC, $\cos \phi=0.4)$ |
|  | Mechanic |  | 20,000,000 operations |
| ON delay |  |  | 15 ms max. |
| OFF delay |  |  | 15 ms max. |
| Circuit configuration |  |  |  |

Note
(1) Under the worst conditions, the service life of output contacts is as shown above. The service life of relays is as shown in the following diagram as a guideline.

(2) With the CPM1A-16ER/CP1W-32ER/CP1W-16ER, the load current is restricted depending on the ambient temperature. Design the system considering the load current based on the following graph.

(3) CP1W-32ER's maximum number of simultaneously ON output points is 24 (75\%). Design the system considering the simultaneously ON points and load current based on the following curve.

(4) There are restrictions imposed by the ambient temperature.

Relay Output Load Current Derating Curves for Expansion I/O Units (CP1W8ER/16ER/20EDR1/32ER/40EDR and CPM1A-8ER/16ER/20EDR1/40EDR)


Transistor Outputs (Sinking or Sourcing)

| Item | Specification |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  | CP1W-40EDT CPM1A-40EDT CP1W-40EDT1 CPM1A-40EDT1 | CP1W-32ET CP1W-32ET1 | $\begin{aligned} & \text { CP1W-20EDT } \\ & \text { CPM1A-20EDT } \\ & \text { CP1W-20EDT1 } \\ & \text { CPM1A-20EDT1 } \end{aligned}$ | $\begin{aligned} & \hline \text { CP1W-16ET } \\ & \text { CP1W-16ET1 } \end{aligned}$ | $\begin{aligned} & \text { CP1W-8ET } \\ & \text { CPM1A-8ET } \\ & \text { CP1W-8ET1 } \\ & \text { CPM1A-8ET1 } \end{aligned}$ |
| Max. switching capacity (See note2.) | 4.5 to 30 VDC 0.3 A/output | 4.5 to 30 VDC 0.3 A/output | $\begin{aligned} & \text { 24 VDC }{ }^{10 \%} /-5 \% \\ & 0.3 \text { A/output } \end{aligned}$ | 4.5 to 30 VDC 0.3 A/output | - OUT00/01 4.5 to 30 VDC, 0.2 A/ output <br> - OUT02 to 074.5 to 30 VDC, 0.3 A/output |
|  | 0.9 A/common 3.6 A/Unit | 0.9 A/common 7.2 A/Unit | 0.9 A/common 1.8 A/Unit | 0.9 A/common 3.6 A/Unit | 0.9 A/common 1.8 A/Unit |
| Leakage current | 0.1 mA max. | 0.1 mA max. | 0.1 mA max. | 0.1 mA max. | 0.1 mA max. |
| Residual voltage | 1.5 V max. | 1.5 V max. | 1.5 V max. | 1.5 V max. | 1.5 V max. |
| ON delay | 0.1 ms max. | 0.1 ms max . | 0.1 ms . | 0.1 ms max . | 0.1 ms max. |
| OFF delay | $\begin{aligned} & 1 \mathrm{~ms} \max . \\ & 24 \mathrm{VDC}+10 \% /-5 \% \\ & 5 \text { to } 300 \mathrm{~mA} \end{aligned}$ | 1 ms max. 24 VDC ${ }^{+10 \%}{ }^{-5 \%}$ 5 to 300 mA | 1 ms max. 24 VDC ${ }^{+10 \%} /_{-5 \%}$ 5 to 300 mA | 1 ms max. 24 VDC ${ }^{+10 \%} /-5 \%$ 5 to 300 mA | 1 ms max. 24 VDC ${ }^{+10 \%} /-5 \%$ 5 to 300 mA |
| Max. number of Simultaneously ON Points of Output | 16 pts (100\%) | 24 pts (75\%) | 8 pts (100\%) | 16 pts (100\%) | 8 pts (100\%) |
| Fuse (See note 1.) | 1 fuse/common |  |  |  |  |
| Circuit configuration |  |  |  |  |  |

Note (1) The fuse cannot be replaced by the user.
(2) If the ambient temperature is maintained below $50^{\circ} \mathrm{C}$, up to $0.9 \mathrm{~A} / \mathrm{com}$ mon can be used.


Caution Do not connect a load to an output terminal or apply a voltage in excess of the maximum switching capacity.

## 2-3 CP1L CPU Unit Operation

## 2-3-1 Overview of CPU Unit Configuration

The CP1L CPU Unit memory consists of the following blocks.

(1)

- Data is backed up from RAM to the built-in flash memory when changes are made, e.g., from the CX-Programmer.
- When the power supply is turned ON, data is transferred from the builtin flash memory to RAM.
(2) • A CX-Programmer operation can be used to transfer DM Area initial values from RAM to the built-in flash memory.
- The PLC Setup can be set so that DM Area initial values are transferred from the built-in flash memory to RAM when the power supply is turned ON.
(3) - CX-Programmer operations can be used to transfer data from RAM to the Memory Cassette or from the built-in flash memory to the Memory Cassette.
- When the power supply is turned ON, data is transferred from the Memory Cassette to the built-in flash memory and RAM. Data can also be transferred from the Memory Cassette to the built-in flash memory and RAM using the CX-Programmer.

User Program

## Parameter Area

## PLC Setup

## I/O Memory

The user program consists of up to 288 tasks, including interrupt tasks. Each task is programmed from the CX-Programmer and then transferred to the CPU Unit.
There are two types of tasks: cyclic tasks and interrupt tasks. Cyclic tasks are executed once each cycle and interrupt tasks are executed only when the interrupt conditions are met. There can be up to 32 cyclic tasks and up to 256 interrupt tasks. Cyclic tasks are executed in the order of the task numbers.
Instructions programmed in the tasks are executed in order from the first instruction and then I/O memory is refreshed. When all cyclic tasks have been executed, I/O refreshing with PLC Units is performed and then the cyclic tasks are executed again starting from the one with the lowest task number. This is called the cyclic scan method.

The I/O memory area is a RAM area read and written by the user. Some parts of the I/O memory are cleared when the power is interrupted. Other parts are maintained. There are parts that used for data exchange with PLC Units and parts that are used internally.
There are two ways to refresh the parts of I/O memory used for data exchange with PLC Units: Once each program execution cycle and immediately when needed when executing specific instructions.

In addition to the I/O memory used as instructions operands by the user, there is also a separate memory area that can be manipulated only from the CXProgrammer. This area, called the parameter area, contains the following.

- PLC Setup
- Routing tables

The PLC Setup contains configuration parameters that can be set by the user to define the basic specifications of the CPU Unit. Included are serial port settings, a minimum cycle time setting, and other parameters. For details, refer to the CX-Programmer Operation Manual.

## Routing Tables

Tables specifying the communications paths from the Communications Units on the local PLC to remote PLCs connected on other networks must be registered in all the CPU Units in network PLCs to send and receive data between networks. These tables are called the routing tables. The routing tables consist of the relay network table and local network table.
Routing tables are created from the CX-Programmer or Support Software for Communications Units (e.g., CX-Integrator) and then transferred to each CPU Unit.


## Remote Network Table

The remote network tables lists the node number and network address of the first relay node that must be passed through to reach any remote network to which the PLC is not directly connected. Once the routing tables have been registered, any remote network can be reached by passing through relay nodes.

## Local Network Table

The local network table contains the unit number and network address of all Communications Units that are part of the local PLC.

## Built-in Flash Memory

Flash memory is built into the CP1L CPU Units. Data in the following areas is automatically backed up to the flash memory whenever it is written in any way other than by instructions in the user program, e.g., when the CX-Programmer or PT is used to transfer or edit data, edit the program online, or transfer data from a Memory Cassette.

- User program area
- Parameter area (PLC Setup and routing tables)

The next time the power supply is turned ON, the data in the built-in flash memory is automatically transferred to user memory (i.e., the user program area and parameter area).
It is also possible to save data from data areas in I/O memory in the built-in flash memory using operations from the CX-Programmer.
The symbol table, comment file, and program index file can be stored in the comment memory in flash memory. When the program is transferred from the CX-Programmer to the CPU Unit, function block program information is also stored automatically in flash memory.
Note The BKUP indicator on the front of the CPU Unit will light whenever the built-in flash memory is being written or the Memory Cassette is being accessed. Never turn OFF the power supply to the CPU Unit when the BKUP indicator is lit.

Memory Cassette

Memory Cassettes can be used as required in system operation and maintenance. For example, they can be used to save programs, data memory contents, PLC Setup data, or I/O comments from the CX-Programmer. The contents of a Memory Cassette can also be automatically transferred if desired.

## 2-3-2 Flash Memory Data Transfers

## Built-in Flash Memory

## Writing to Flash Memory

| Data | Transfer method |
| :--- | :--- |
| User program and <br> parameter data | This data is automatically transferred from RAM to flash mem- <br> ory when a project is transferred from the CX-Programmer, <br> when the data is written to RAM from a PT or other external <br> device, or when the data is transferred from a Memory Cas- <br> sette. |
| DM Area data | This data is transferred to flash memory only when the trans- <br> fer is specified from the CX-Programmer. |
| Comment memory <br> data | This data is written to flash memory when a project is trans- <br> ferred from the CX-Programmer and transferring comment <br> memory is specified. |
| Function block <br> source data | This data is written to flash memory when a project containing <br> one or more function blocks is transferred from the CX-Pro- <br> grammer. |



Write operation from CX-Programmer or automatic transfer from Memory Cassette at startup.

$\mathrm{FB}=$ Function block

Reading from Flash
Memory

| Data | Read method |
| :--- | :--- |
| User program and <br> parameter data | This data is automatically read to RAM when power is turned <br> ON. |
| DM Area data | Reading this data when power is turned ON can be enabled or <br> disabled in the PLC Setup. |
| Comment memory <br> data | When the project is transferred from the CX-Programmer, <br> comment memory can be specified as a destination to transfer <br> the comment memory data to built-in flash memory. |
| Function block <br> source data | When a project that contains function blocks is transferred <br> from the CX-Programmer, the function block source data is <br> transferred to built-in flash memory. |



FB = Function block

## 2-3-3 Memory Cassette Data Transfers

## Writing to a Memory Cassette

| Data | Method | Source |
| :--- | :--- | :--- |
| User program and <br> parameter data | Data is written to a Memory <br> Cassette using write opera- <br> tions from the CX-Program- <br> mer. | Data in the built-in flash mem- <br> ory is written to the Memory <br> Cassette. |
| Comment memory <br> and function block <br> source data | Either of both of the following <br> can be transferred to the <br> Memory Cassette. |  |
| DM Area data | - Data in the built-in flash <br> memory. <br> - Data in RAM. |  |



## Reading from a Memory Cassette

| Data | Method | Destination |
| :--- | :--- | :--- |
| User program and <br> parameter data | This data is transferred by <br> turning SW2 on the DIP <br> switch to ON and turning ON <br> the power supply. | Data in the Memory Cassette <br> is transferred to RAM and <br> then automatically transferred <br> to the built-in flash memory. |
|  | Data is transferred to the built- <br> in flash memory. |  |
| Comment memory <br> and function block <br> source data |  | DM Area data originally from <br> the built-in flash memory is <br> transferred back to the flash <br> memory and DM Area data <br> originally from RAM is trans- <br> ferred to RAM. |
| DM Area data |  |  |



## 2-4 CPU Unit Operation

## 2-4-1 General Flow

The following flowchart shows the overall operation of the CPU Unit. First the user program is executed and then I/O is refreshed and peripheral servicing is performed. These processes are then repeated in cyclic fashion.


## 2-4-2 I/O Refreshing and Peripheral Servicing

## I/O Refreshing

I/O refreshing involves cyclically transferring data with external devices using preset words in memory. I/O refreshing includes the following:

- Refreshing between I/O words in the CIO Area and CPU Unit built-in I/O, CP/CPM1A-series Expansion Units, and CP/CPM1A-series Expansion I/O Units
All I/O refreshing is performed in the same cycle (i.e., time slicing is not used). I/O refreshing is always performed after program execution.

| Units | Max. data exchange | Data exchange area |
| :--- | :--- | :--- |
| CPU Unit built-in I/O | 2 input words <br> 2 output words | I/O Bit Area |
| CP/CPM1A-series Expansion <br> Units and Expansion I/O Units | Fixed depending on Units | I/O Bit Area |

## Peripheral Servicing

Peripheral servicing involves servicing non-scheduled events for external devices. This includes both events from external devices and service requests to external devices.
Most peripheral servicing involves FINS commands. The specific amount of time set in the system is allocated to each type of servicing and executed every cycle. If all servicing cannot be completed within the allocated time, the remaining servicing is performed the next cycle.

| Service | Description |
| :---: | :---: |
| USB port servicing | - Non-scheduled servicing for FINS or Host Link commands received via a USB port or serial port from the CX-Programmer, PTs, or host computers (e.g., requests for program transfers, monitoring, forced-set/reset operations, or online editing) <br> - Non-scheduled servicing from the CPU Unit transmitted from a serial port (non-solicited communications) |
| Communications port servicing |  |
| Communications port servicing | - Servicing to execute network communications or serial communications for the SEND, RECV, CMND or PMCR instructions using communications ports 0 to 7 (internal logical ports) <br> - Servicing to execute background execution using communications ports 0 to 7 (internal logical ports) |
| Built-in flash memory access servicing | - Read/write processing for built-in flash memory |
| Memory Cassette access servicing | - Read/write processing for a Memory Cassette |
| Peripheral USB port, serial port, and communications port servicing is allocated $8 \%$ of the previous cycle time by default (the default can be changed) for each service. If servicing is separated over many cycles, delaying completion of the servicing, set the same allocated time (same time for all services) rather than a percentage under execute time settings in the PLC Setup. |  |

## 2-4-3 I/O Refresh Methods

I/O for CPU Unit built-in I/O and I/O on CP/CPM1A-series Expansion Units and Expansion I/O Units is performed at the following times.

1,2,3... 1. Cyclic refresh period
2. When instructions with an immediate refresh variation are executed
3. When IORF(097) is executed

## Cyclic Refreshing

Immediate Refreshing

Note

I/O is refreshed after all the instructions in executable tasks have been executed.


When the immediate refreshing variation of an instruction is specified and the instruction's operand is an input bit or word in the Built-in I/O Area, the word containing the bit or the word itself will be refreshed.

(1) Immediate refreshing is possible only for the Built-in I/O Area. Use IORF(097) for I/O on CP/CPM1A-series Expansion Units and Expansion I/O Units.
(2) Refreshing Range

- Bit Operands

The ON/OFF status of the 16 I/O points allocated to the word containing the specified bit will be refreshed.

- Word Operands

The ON/OFF status of the 16 I/O points allocated to the specified word will be refreshed.
(3) Refresh Timing

- Input or source operands are read just before the instruction is executed.
- Output or destination (results) operands are written just after the instruction is executed.
(4) Using instructions with the immediate refresh option, instruction execution time will be increased, increasing the overall cycle time. Be sure to confirm that this will not adversely affect system operation.

IORF(097) Refreshing
When IORF(097) (I/O REFRESH) is executed, the I/O bits in the specified range of words are refreshed. IORF(097) can be used for CP/CPM1A-series Expansion Units and CP/CPM1A-series Expansion I/O Units.


St: Starting word
E: End word
All the words from St to E , inclusive are refreshed.

Example

| IORF |
| :---: |
| 2 |
| 5 |

Here, the four words from CIO 2 to CIO 5 are refreshed.

If high-speed response is required from input to output, execute $\operatorname{IORF}(097)$ before and after the relevant instructions.

Note $\operatorname{IORF}(097)$ has a relatively long execution time which increase with the number of words being refreshed. Be sure to consider the affect of this time on the overall cycle time. Refer to the CP Series Programmable Controllers Programming Manual for instruction execution times.

## 2-4-4 Initialization at Startup

The following initializing processes will be performed once each time the power is turned ON.

- Confirm mounted Units and I/O allocations.
- Clear the non-holding areas of I/O memory according to the status of the IOM Hold Bit. (See note 1.)
- Clear forced status according to the status of the Forced Status Hold Bit. (See note 2.)
- Automatically transfer data from the Memory Cassette if one is mounted and automatic transfer at startup is specified.
- Perform self-diagnosis (user memory check).
- Restore the user program. (See note 3.)

Note (1) The I/O memory is held or cleared according to the status of the IOM Host Bit and the setting for IOM Hold Bit Status at Startup in the PLC Setup (read only when power is turned ON).

| Auxiliary bit <br> PLC Setup setting |  | IOM Hold Bit (A500.12) |  |
| :---: | :---: | :---: | :---: |
|  |  | Clear (OFF) | Hold (ON) |
| IOM Hold Bit Status at Startup | $\begin{aligned} & \hline \text { Clear } \\ & \text { (OFF) } \end{aligned}$ | At power ON: Clear At mode change: Clear | At power ON: Clear At mode change: Hold |
|  | Hold (ON) |  | At power ON: Hold At mode change: Hold |

Note When the mode is changed between PROGRAMMING mode and RUN or MONITOR mode, I/O memory initialization is according to the status of the IOM Hold Bit at that time.
(2) The forced status held or cleared according to the status of the Force Status Hold Bit and the setting for Forced Status Hold Bit Status at Startup in the PLC Setup (read only when power is turned ON).

| Auxiliary bit PLC Setup setting |  | Forced Status Hold Bit (A500.13) |  |
| :---: | :---: | :---: | :---: |
|  |  | Clear (OFF) | Hold (ON) |
| Forced Status Hold Bit Status at Startup | $\begin{aligned} & \text { Clear } \\ & \text { (OFF) } \end{aligned}$ | At power ON: Clear At mode change: Clear | At power ON: Clear At mode change: Hold |
|  | Hold (ON) |  | At power ON: Hold At mode change: Hold |

Note When the mode is changed between PROGRAMMING mode and RUN or MONITOR mode, forced status initialization is according to the status of the Forced Status Hold Bit at that time.
(3) User program recovery is performed if online editing is performed but the power supply to the PLC is turned OFF before the CPU Unit can complete backup processing. The BKUP indicator will light during backup processing.

## 2-5 CPU Unit Operating Modes

## 2-5-1 Operating Modes

The CPU Unit has three operating modes that control the entire user program and are common to all tasks.
PROGRAM: Programs are not executed and preparations, such as initializing the PLC Setup and other settings, transferring programs, checking programs, force-setting and force-resetting can be executed prior to program execution.
MONITOR: Programs are executed, but some operations, such as online editing, forced-set/reset, and changes to present values in I/O memory, are enabled for trial operation and other adjustments.
RUN: $\quad$ Programs are executed and some operations are disabled.

## 2-5-2 Status and Operations in Each Operating Mode

The following table lists status and operations for each mode.

| Operation |  |  | PROGRAM mode | RUN mode | MONITOR mode |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Program execution |  |  | Stopped | Executed | Executed |
| I/O refreshing |  |  | Executed | Executed | Executed |
| External I/O status |  |  | OFF | According to program | According to program |
| I/O memory | Non-holding memory |  | Cleared | According to program | According to program |
|  | Holding memory |  | Held |  |  |
| CX-Programmer operations | I/O memory monitoring |  | OK | OK | OK |
|  | Program monitoring |  | OK | OK | OK |
|  | Program transfers | From CPU Unit | OK | OK | OK |
|  |  | To CPU Unit | OK | X | X |
|  | Checking program |  | OK | X | X |
|  | Setting PLC Setup |  | OK | X | X |
|  | Changing program |  | OK | X | OK |
|  | Force-setting/resetting |  | OK | X | OK |
|  | Changing timer/counter SV |  | OK | X | OK |
|  | Changing timer/counter PV |  | OK | X | OK |
|  | Change I/O memory PV |  | OK | X | OK |

Note The following table shows the relationship of operating modes to tasks.

| Mode | Cyclic task status | Interrupt task <br> status |
| :---: | :--- | :--- |
| PROGRAM | Disabled status (INI) | Stopped |


| RUN | - Any task that has not yet been executed, will be in disabled status (INI). <br> - <br> - A task will go to READY status if the task is set to go to READY status at star- <br> tup or the TASK ON (TKON) instruction has been executed for it. <br> - A task in READY status will be executed (RUN status) when it obtains the <br> right to execute. <br> MONITOR <br> A status will go to Standby status (WAIT) if a READY task is put into Standby <br> status by a TASK OFF (TKOF) instruction. | Executed if inter- <br> rupt condition is <br> met. |
| :--- | :--- | :--- | :--- |

## 2-5-3 Operating Mode Changes and I/O Memory

## Operating Mode Changes and I/O Memory

| Mode Changes | Non-holding areas | Holding Areas |
| :---: | :---: | :---: |
|  | - I/O bits <br> - Data Link bits <br> - Work bits <br> - Timer PV/Completion Flags <br> - Index Registers <br> - Data Registers <br> - Task Flags <br> Auxiliary Area bits/words are holding or non-holding depending on the address. | - HR Area <br> - DM Area <br> - Counter PV and Completion Flags Auxiliary Area bits/words are holding or non-holding depending on the address. |
| RUN or MONITOR to PROGRAM | Cleared (See note 1.) | Held |
| PROGRAM to RUN or MONITOR | Cleared (See note 1.) | Held |
| RUN to MONITOR or MONITOR to RUN | Held (See note 2.) | Held |

Note 1. The following processing is performed if the I/O Memory Hold Bit is ON. Outputs from Output Units will be turned OFF when operation stops even if I/O bit status is held in the CPU Unit.
2. The cycle time will increase by approximately 10 ms when the operating mode is changed from MONITOR to RUN mode. This will not, however, cause an error for exceeding the maximum cycle time limit.

| I/O Memory Hold Bit status (A500.12) | I/O Memory |  |  | Output bits allocated to Output Units |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | Mode changed between PROGRAM and RUN/ MONITOR | Operation stopped |  | Mode changed between PROGRAM and RUN/ MONITOR | Operation stopped |  |
|  |  | Fatal error other than FALS | FALS executed |  | Fatal error other than FALS | FALS executed |
| OFF | Cleared | Cleared | Held | OFF | OFF | OFF |
| ON | Held | Held | Held | Held | OFF | OFF |

Note Refer to SECTION 4 I/O Memory Allocation.

## 2-5-4 Startup Mode Setting

This setting in the PLC Setup determines the operating mode that will be used by the CPU Unit when the power supply is turned ON.

## PLC Setup

| Name | Description | Settings | Default |
| :---: | :--- | :--- | :---: |
| Startup Mode | Specifies the <br> CPU Unit oper- <br> ating mode at <br> startup | • Program (See note.) <br> - Monitor <br> - Run <br> -Use programming console | Use program- <br> ming console <br> (See note.) |

A Programming Console cannot be connected to the CP1L.


Note A Programming Console cannot be connected to a CP1L CPU Unit. If Use programming console is set, the CPU Unit will start in RUN mode.

## 2-6 Power OFF Operation

## 2-6-1 Overview

The following processing is performed when CPU Unit power is turned OFF. Power OFF processing will be performed if the power supply voltage falls below the specified value while the CPU Unit is in RUN or MONITOR mode.

1,2,3... 1. The CPU Unit will stop.
2. Outputs from all Output Units will be turned OFF.

Note (1) All outputs will turn OFF despite the status of the I/O Memory Hold Bit or I/O Memory Hold Bit at power ON settings in the PLC Setup.
(2) AC Power
$85 \%$ of the rated voltage: 85 V or less for a 100 to 240 V AC system
(3) DC Power
$90 \%$ of rated voltage: 20.4 V DC or less
The following processing will be performed if power drops only momentarily (momentary power interruption).

1,2,3... 1. The system will continue to run unconditionally if the momentary power interruption lasts less than 10 ms for AC power or 2 ms for DC power, i.e., the time it takes the rated voltage at $85 \%$ or less to return to $85 \%$ or higher is less than 10 ms for $A C$ power or the time it takes the rated voltage at $90 \%$ or less to return to $90 \%$ or higher is less than 2 ms for DC power.
2. A momentary power interruption that lasts more than 10 ms for $A C$ power or more than 2 ms for DC power may or may not be detected.


The following timing chart shows the CPU Unit power OFF operation in more detail.

## Power OFF Timing Chart



Power OFF detection time:
The time from when the power supply voltages drops to $85 \%$ or less of the rated voltage for AC power or $90 \%$ for DC power until the power OFF condition is detected.

Holding time for 5 V internal power supply after power OFF detection:
The maximum time that the 5 V internal power supply voltage will be maintained after the power OFF condition is detected. The holding time is fixed at 1 ms .

## Description of Operation

Power OFF will be detected if the 100 to 240 V AC power supply falls below $85 \%$ of the rated voltage or the DC power supply falls below $90 \%$ of the rated voltage for the power OFF detection time ( 10 ms minimum for AC power and 2 ms minimum for DC power). The CPU reset signal will turn ON while the internal power supply is being held and the CPU Unit will be reset.

## 2-6-2 Instruction Execution for Power Interruptions

If power is interrupted and the interruption is detected when the CPU Unit is operating in RUN or MONITOR mode, the instruction currently being executed will be completed and then the CPU Unit will be reset.

## 2-7 Computing the Cycle Time

## 2-7-1 CPU Unit Operation Flowchart

The CPU Unit processes data in repeating cycles from the overseeing processing up to peripheral servicing as shown in the following diagram.


Peripheral servicing


## 2-7-2 Cycle Time Overview

The cycle time depends on the following conditions.

- Type and number of instructions in the user program (in all cyclic tasks that are executed during a cycle, and within interrupt tasks for which the execution conditions have been satisfied)
- Type and number of CP/CPM1A-series Expansion Units and Expansion I/ O Units
- Use of protocol macros and the largest communications message
- Fixed cycle time setting in the PLC Setup
- Use of USB and serial ports
- Fixed peripheral servicing time in the PLC Setup

Note 1. The cycle time is not affected by the number of tasks that are used in the user program. The tasks that affect the cycle time are those cyclic tasks that are READY in the cycle.
2. When the mode is switched from MONITOR mode to RUN mode, the cycle time will be extended by 10 ms (this will not, however, take the cycle time over its limit).
The cycle time is the total time required for the PLC to perform the five operations given in the following tables.
Cycle time $=(1)+(2)+(3)+(4)+(5)$
1: Overseeing

| Details | Processing time and fluctuation cause |
| :--- | :--- |
| Checks the I/O bus and user program memory, checks for <br> battery errors, etc. | 0.4 ms |

## Program Execution

| Details | Processing time and fluctuation cause |
| :--- | :--- |
| Executes the user program, and calculates the total time <br> time taken for the instructions to execute the program. | Total instruction execution time |

## 3: Cycle Time Calculation

| Details | Processing time and fluctuation cause |
| :--- | :--- |
| Waits for the specified cycle time to elapse when a minimum | When the cycle time is not fixed, the time for step 3 is <br> approximately 0. <br> (fixed) cycle time has been set in the PLC Setup. |
| When the cycle time is fixed, the time for step 3 is the preset <br> fixed cycle time minus the actual cycle time ((1) + (2) + (4) + <br> (5)). |  |

## 4: I/O Refreshing

| Details | Processing time and fluctuation cause |  |
| :--- | :--- | :--- |
| CPU Unit built- | Outputs from the CPU Unit to the actual | I/O refresh time for each Unit multiplied by the number of |
| in I/O and I/O | outputs are refreshed first for each Unit, | Units used. |
| on CP/ |  |  |
| CPM then inputs. |  |  |
| CPMA-series |  |  |
| Expansion |  |  |
| Units and |  |  |
| Expansion I/O |  |  |
| Units |  |  |

5: Peripheral Servicing

| Details | Processing time and fluctuation cause |
| :--- | :--- |
| Services USB port. | If a uniform peripheral servicing time hasn't been set in the PLC Setup for <br> this servicing, 8\% of the previous cycle's cycle time (calculated in step (3)) <br> will be allowed for peripheral servicing. <br> If a uniform peripheral servicing time has been set in the PLC Setup, servic- <br> ing will be performed for the set time. Servicing will be performed for at <br> least 0.1 ms, however, whether the peripheral servicing time is set or not. <br> If the ports are not connected, the servicing time is 0 ms. |
| Services serial ports | If a uniform peripheral servicing time hasn't been set in the PLC Setup for <br> this servicing, 8\% of the previous cycle's cycle time (calculated in step (3)) <br> will be allowed for peripheral servicing. |
| If a uniform peripheral servicing time has been set in the PLC Setup, servic- |  |
| ing will be performed for the set time. Servicing will be performed for at |  |
| least 0.1 ms, however, whether the peripheral servicing time is set or not. |  |
| If no communications ports are used, the servicing time is 0 ms. |  |, | If a uniform peripheral servicing time hasn't been set in the PLC Setup for |
| :--- |
| this servicing, 8\% of the previous cycle's cycle time (calculated in step (3)) |
| will be allowed for peripheral servicing. |
| If a uniform peripheral servicing time has been set in the PLC Setup, servic- |
| ing will be performed for the set time. Servicing will be performed for at |
| least 0.1 ms, however, whether the peripheral servicing time is set or not. |
| If there is no access, the servicing time is 0 ms. |

## 2-7-3 Functions Related to the Cycle Time

## Minimum Cycle Time

Set the minimum cycle time to a non-zero value to eliminate inconsistencies in I/O responses. A minimum cycle time can be set in the PLC Setup between 1 and $32,000 \mathrm{~ms}$ in 1 -ms increments.


This setting is effective only when the actual cycle time is shorter than the minimum cycle time setting. If the actual cycle time is longer than the minimum cycle time setting, the actual cycle time will remain unchanged.


## PLC Setup

| Name | Settings | Default |
| :---: | :--- | :---: |
| Minimum cycle time | 0000 to 7D00 hex <br> (1 to 32,000 ms in 1-ms incre- <br> ments) | 0000 hex: Variable cycle time |

## Watch Cycle Time

If the cycle time exceeds the watch (maximum) cycle time setting, the Cycle Time Too Long Flag (A401.08) will be turned ON and PLC operation will be stopped.

## PLC Setup

| Name | Settings | Default |
| :--- | :--- | :--- |
| Enable Watch Cycle 0: Default (1 s) <br> Time Setting0000 hex: Watch cycle time of <br> 1 s |  |  |
| Watch Cycle Time | O01 to FA0: 10 to $40,000 \mathrm{~ms}$ <br> (10-ms increments) |  |

## Related Flags

Cycle Time Monitoring

Related Words

| Name | Address | Description |
| :--- | :--- | :--- |
| Cycle Time Too Long <br> Flag | A401.08 | Turns ON if the present cycle time exceeds the <br> Watch Cycle Time set in the PLC Setup. |

The maximum cycle time is stored in A262 and A263 and the present cycle time is stored in A264 and A265 every cycle.

| Name | Addresses | Description |
| :--- | :--- | :--- |
| Maximum Cycle <br> Time | A262 and <br> A263 | These words contain the maximum cycle time in <br> increments of 0.1 ms. The time is updated every <br> cycle and is recorded in 32-bit binary (0 to FFFF <br> FFFF hex, or 0 to 429,496,729.5 ms). (A263 is <br> the leftmost word.) |
| Present Cycle Time | A264 and <br> A265 | These words contain the present cycle time in <br> increments of 0.1 ms. The time is updated every <br> cycle and is recorded in 32-bit binary (0 to FFFF <br> FFFF, or 0 to 429,496,729.5 ms). (A265 is the <br> leftmost word.) |

The average cycle time for the past eight cycles can be read from the CX-Programmer.

Note The following methods are effective in reducing the cycle time.

- Place tasks that do not need to be executed on standby.
- Use JMP-JME instructions to skip instructions that do not need to be executed.


## 2-7-4 I/O Refresh Times for PLC Units

## CP-series Expansion Unit and Expansion I/O Unit I/O Refresh Times

| Name | Model | I/O refresh time per Unit |
| :---: | :---: | :---: |
| Expansion I/O Units | CP1W-40EDR CPM1A-40EDR | 0.39 ms |
|  | CP1W-40EDT <br> CPM1A-40EDT | 0.39 ms |
|  | CP1W-40EDT1 <br> CPM1A-40EDT1 | 0.39 ms |
|  | CP1W-32ER | 0.33 ms |
|  | $\begin{aligned} & \text { CP1W-32ET } \\ & \text { CP1W-32ET1 } \end{aligned}$ | 0.33 ms |
|  | CP1W-20EDT CPM1A-20EDT | 0.18 ms |
|  | CP1W-20EDT1 CPM1A-20EDT1 | 0.18 ms |
|  | CP1W-16ER CPM1A-16ER | 0.25 ms |
|  | $\begin{aligned} & \text { CP1W-16ET } \\ & \text { CP1W-16ET1 } \end{aligned}$ | 0.25 ms |
|  | CP1W-8ED CPM1A-8ED | 0.13 ms |
|  | CP1W-8ER CPM1A-8ER | 0.08 ms |
|  | CP1W-8ET CPM1A-8ET | 0.08 ms |
|  | $\begin{aligned} & \text { CP1W-8ET1 } \\ & \text { CPM1A-8ET1 } \end{aligned}$ | 0.08 ms |
| Analog Input Units | $\begin{aligned} & \text { CP1W-AD041 } \\ & \text { CPM1A-AD041 } \end{aligned}$ | 0.61 ms |
| Analog Output Units | CP1W-DA021 | 0.33 ms |
|  | CP1W-DA041 CPM1A-DA041 | 0.33 ms |
| Analog I/O Units | CPM1A-MAD01 | 0.29 ms |
|  | CP1W-MAD11 CPM1A-MAD11 | 0.32 ms |
| Temperature Sensor Units | $\begin{array}{\|l\|} \hline \text { CP1W-TS001 } \\ \text { CPM1A-TS001 } \end{array}$ | 0.25 ms |
|  | $\begin{aligned} & \hline \text { CP1W-TS002 } \\ & \text { CPM1A-TS002 } \end{aligned}$ | 0.52 ms |
|  | $\begin{aligned} & \hline \text { CP1W-TS101 } \\ & \text { CPM1A-TS101 } \end{aligned}$ | 0.25 ms |
|  | CP1W-TS102 <br> CPM1A-TS102 | 0.52 ms |
| DeviceNet I/O Link Unit | CPM1A-DRT21 | 0.38 ms |
| CompoBus/S I/O Link Unit | CP1W-SRT21 CPM1A-SRT21 | 0.21 ms |

Note The I/O refresh time for CPU Unit built-in I/O is included in overhead processing.

## 2-7-5 Cycle Time Calculation Example

The following example shows the method used to calculate the cycle time when CP-series Expansion I/O Units only are connected to a CP1L CPU Unit.

## Conditions

| Item | Details |  |
| :--- | :--- | :--- |
| CP1L | CP1W-40EDR <br> $40-$ pt I/O Unit | 1 Unit |
| User program | 5 K steps | LD instructions: 2.5 Ksteps, <br> OUT instructions: 2.5 Ksteps |
| USB port connection | Yes and no |  |
| Fixed cycle time processing | No |  |
| Serial port connection | No |  |
| Other peripheral servicing | No |  |

## Calculation Example

| Process name | Calculation | Processing time |  |
| :--- | :--- | :--- | :--- |
|  |  | USB port <br> connected | USB port not <br> connected |
| (1) Overseeing | --- | 0.4 ms | 0.4 ms |
| (2) Program execution | $0.55 \mu \mathrm{~s} \times 2,500+1.1 \mu \mathrm{~s}$ <br> $\times 2,500$ | 4.1 ms | 4.1 ms |
| (3) Cycle time calculation | (Minimum cycle time not <br> set) | 0 ms | 0 ms |
| (4) I/O refreshing | 0.39 ms | 0.39 ms | 0.39 ms |
| (5) Peripheral servicing | (Only USB port con- <br> nected) | 0.1 ms | 0 ms |
|  |  |  |  |
| Cycle time | $(1)+(2)+(3)+(4)+(5)$ | 4.99 ms | 4.89 ms |

## 2-7-6 Online Editing Cycle Time Extension

When online editing is executed to change the program from the CX-Programmer while the CPU Unit is operating in MONITOR mode, the CPU Unit will momentarily suspend operation while the program is being changed. The period of time that the cycle time is extended is determined by the following conditions.

- Number of steps changed
- Editing operations (insert/delete/overwrite)
- Types of instructions

The cycle time extension for online editing is negligibly affected by the size of task programs. If the maximum program size for a task is 10 Ksteps , the online editing cycle time extension will be as follows:

| CPU Unit | Increase in cycle time for online editing |
| :--- | :--- |
| CP1L CPU Unit | Maximum: 16 ms, Normal: 12 ms <br> (for a program size of 10 Ksteps) |

When editing online, the cycle time will be extended by according to the editing that is performed. Be sure that the additional time will not adversely affect system operation.

Note When there is one task, online editing is processed all in the cycle time following the cycle in which online editing is executed (written). When there are multiple tasks (cyclic tasks and interrupt tasks), online editing is separated, so that for n tasks, processing is executed over n to $\mathrm{n} \times 2$ cycles max.

## 2-7-7 I/O Response Time

The I/O response time is the time it takes from when an input turns ON, the data is recognized by the CPU Unit, and the user program is executed, up to the time for the result to be output to an output terminal. The length of the I/O response time depends on the following conditions.

- Timing of Input Bit turning ON.
- Cycle time.


## Minimum I/O Response Time

The I/O response time is shortest when data is retrieved immediately before I/ O refresh of the CPU Unit. The minimum I/O response time is calculated as follows:
Minimum I/O response time = Input ON delay + Cycle time + Output ON delay
Note The input and output ON delays depend on the type of terminals used on the CPU Unit or the model number of the Unit being used.


## Maximum I/O Response Time

The I/O response time is longest when data is retrieved immediately after I/O refresh period of the CPU Unit. The maximum I/O response time is calculated as follows:
Maximum I/O response time $=$ Input ON delay $+($ Cycle time $\times 2)+$ Output ON delay


Input Response Times

## Calculation Example

| Conditions: | Input ON delay | 1 ms (normal input with input <br> constant set to 0 ms ) |
| :--- | :--- | :--- |
|  | Output ON delay | 0.1 ms (transistor output) |
|  | Cycle time | 20 ms |

Minimum I/O response time $=1 \mathrm{~ms}+20 \mathrm{~ms}+0.1 \mathrm{~ms}=21.1 \mathrm{~ms}$
Maximum I/O response time $=1 \mathrm{~ms}+(20 \mathrm{~ms} \times 2)+0.1 \mathrm{~ms}=41.1 \mathrm{~ms}$
Input response times can be set in the PLC Setup. Increasing the response time reduces the effects of chattering and noise. Decreasing the response time allows reception of shorter input pulses, (but the pulse width must be longer than the cycle time).


## PLC Setup

| Name | Description | Settings | Default |
| :---: | :--- | :--- | :---: |
| Input constants | Input response times | 00 hex: 8 ms | 00 hex (8 ms) |
|  |  | 10 hex: 0 ms |  |
|  |  | 11 hex: 0.5 ms |  |
|  |  | 12 hex: 1 ms |  |
|  |  | 13 hex: 2 ms |  |
|  |  | 14 hex: 4 ms |  |
|  |  | 15 hex: 8 ms |  |
|  |  | 16 hex: 16 ms |  |
|  |  | 17 hex: 32 ms |  |

## 2-7-8 Interrupt Response Times

Input Interrupt Tasks

The interrupt response time for I/O interrupt tasks is the time taken from when a built-in input has turned ON (or OFF) until the I/O interrupt task has actually been executed. The length of the interrupt response time for I/O interrupt tasks depends on the following conditions.

| Item | Interrupt response time | Counter interrupts |
| :--- | :--- | :--- |
| Hardware response | Rise time: $50 \mu \mathrm{~s}$ | --- |
|  | Fall time: $50 \mu \mathrm{~s}$ | --- |
| Software interrupt <br> response | Minimum: $134 \mu \mathrm{~s}$ | Minimum: $236 \mu \mathrm{~s}$ |
|  | Maximum: $234 \mu \mathrm{~s}+$ Wait <br> time (See note 1.) | Maximum: $336 \mu \mathrm{~s}+$ Wait time <br> (See note1.) |

(1) The wait time occurs when there is competition with other interrupts. As a guideline, the wait time will be 6 to $169 \mu \mathrm{~s}$.
(2) I/O interrupt tasks can be executed during execution of the user program (even while an instruction is being executed by stopping the execution of an instruction), I/O refresh, peripheral servicing, or overseeing. The interrupt response time is not affected by which of the above processing operations during which the interrupt inputs turns ON. I/O interrupts, however, are not executed during execution of other interrupt tasks even if the I/O interrupt conditions are satisfied. Instead, the I/O interrupts are


## Scheduled Interrupt Tasks

The interrupt response time of scheduled interrupt tasks is the time taken from after the scheduled time specified by the MSKS(690) instruction has elapsed until the interrupt task has actually been executed. The length of the interrupt response time for scheduled interrupt tasks is 1 ms max. There is also an error of $80 \mu \mathrm{~s}$ in the time to the first scheduled interrupt ( 0.5 ms min.).
Note Scheduled interrupt tasks can be executed during execution of the user program (even while an instruction is being executed by stopping the execution of an instruction), I/O refresh, peripheral servicing, or overseeing. The interrupt response time is not affected by which of the above processing operations during which the scheduled interrupt time occurs. Scheduled interrupts, however, are not executed during execution of other interrupt tasks even if the interrupt conditions are satisfied. Instead, the interrupts are executed in order of priority after the current interrupt task has completed execution and the software interrupt response time has elapsed.


## 2-7-9 Serial PLC Link Response Performance

The response times for CPU Units connected via a Serial PLC Link (master to slave or slave to master) can be calculated as shown below. If a PT is in the Serial PLC Link, however, the amount of communications data will not be fixed and the values will change.

- Maximum I/O response time (not including hardware delay) = Master cycle time + Communications cycle time + Slave cycle time +4 ms
- Minimum I/O response time (not including hardware delay) $=$ Slave communications time +0.8 ms

Here,

| Number of participating slave nodes | The number of slaves to which links have been established within the maximum unit number set in the master. |
| :---: | :---: |
| Number of non-participating slave nodes | The number of slaves not participating in the links within the maximum unit number set in the master |
| Communications cycle time (ms) | Slave communications time $\times$ Number of participating slave nodes $+10 \times$ Number of non-participating slave nodes |
| Slave communications time (ms) | - Communications time set to Standard $0.4+0.286 \times(($ No. of slaves +1$) \times$ No. of link words $\times 2+12)$ <br> - Communications time set to Fast $0.4+0.0955 \times(($ No. of slaves +1$) \times$ No. of link words $\times 2+$ 12) |

## 2-7-10 Pulse Output Start Time

The pulse output start time is the time required from executing a pulse output instruction until pulses are output externally. This time depends on the pulse output instruction that is used and operation that is performed.


| Pulse output instruction | Start time |
| :--- | :--- |
| SPED: continuous | $86 \mu \mathrm{~s}$ |
| SPED: independent | $98 \mu \mathrm{~s}$ |
| ACC: continuous | $103 \mu \mathrm{~s}$ |
| ACC: independent, trapezoidal | $122 \mu \mathrm{~s}$ |
| ACC: independent, triangular | $123 \mu \mathrm{~s}$ |
| PLS2: trapezoidal | $145 \mu \mathrm{~s}$ |
| PLS2: triangular | $146 \mu \mathrm{~s}$ |

## 2-7-11 Pulse Output Change Response Time

The pulse output change response time is the time for any change made by executing an instruction during pulse output to actually affect the pulse output operation.

| Pulse output instruction | Change response time |
| :--- | :--- |
| INI: immediate stop | $63 \mu \mathrm{~s}+1$ pulse output time |
| SPED: immediate stop | $106 \mu \mathrm{~s}+1$ pulse output time |
| ACC: deceleration stop | 1 control cycle (4 ms) minimum, |
| PLS2: deceleration stop | 2 control cycles ( 8 ms ) maximum |
| SPED: speed change |  |
| ACC: speed change |  |
| PLS2: target position change in <br> reverse direction |  |
| PLS2: target position change in <br> same direction at same speed |  |
| PLS2: target position change in <br> same direction at different speed |  |

## SECTION 3 <br> Installation and Wiring

This section describes how to install and wire the CP1L.
3-1 Fail-safe Circuits ..... 100
3-2 Installation Precautions ..... 101
3-2-1 Installation and Wiring Precautions ..... 101
3-3 Mounting ..... 103
3-3-1 Mounting in a Panel ..... 103
3-3-2 Connecting Expansion Units and Expansion I/O Units ..... 106
3-3-3 DIN Track Installation ..... 108
3-4 Wiring CP1L CPU Units ..... 110
3-4-1 Wiring Power Supply and Ground Lines ..... 111
3-4-2 Wiring Built-in I/O ..... 113
3-4-3 Wiring Safety and Noise Controls ..... 116
3-5 Wiring CPU Unit I/O ..... 118
3-5-1 I/O Wiring for CPU Units with 60 I/O Points ..... 118
3-5-2 I/O Wiring for CPU Units with 40 I/O Points ..... 119
3-5-3 I/O Wiring for CPU Units with 30 I/O Points ..... 121
3-5-4 I/O Wiring for CPU Units with 20 I/O Points ..... 122
3-5-5 I/O Wiring for CPU Units with 14 I/O Points ..... 124
3-5-6 I/O Wiring for CPU Units with 10 I/O Points ..... 125
3-5-7 Pulse Input Connection Examples ..... 127
3-5-8 Pulse Output Connection Examples ..... 127
3-6 CP/CPM1A-series Expansion I/O Unit Wiring ..... 130

## 3-1 Fail-safe Circuits

Supply Power to the CP1L CPU Unit before the Controlled<br>System

Managing CPU Unit Errors

Always set up safety circuits outside of the PLC to prevent dangerous conditions in the event of errors in the CP1L CPU Unit or external power supply. In particular, be careful of the following points.

If the PLC's power supply is turned ON after the controlled system's power supply, outputs in Units such as DC Output Units may malfunction momentarily. To prevent any malfunction, add an external circuit that prevents the power supply to the controlled system from going ON before the power supply to the PLC itself.

When any of the following errors occurs, PLC operation (program execution) will stop and all outputs from Output Units will be turned OFF.

- A CPU error (watchdog timer error) or CPU on standby
- A fatal error (memory error, I/O bus error, duplicate number error, too many I/O points error, I/O setting error, program error, cycle time too long error, or FALS(007) error) (See note.)
Always add any circuits necessary outside of the PLC to ensure the safety of the system in the event of an error that stops PLC operation.

Note When a fatal error occurs, all outputs from Output Units will be turned OFF even if the IOM Hold Bit has been turned ON to protect the contents of I/O memory. (When the IOM Hold Bit is ON, the outputs will retain their previous status after the PLC has been switched from RUN/MONITOR mode to PROGRAM mode.)

Managing Output Malfunctions

Interlock Circuits

It is possible for an output to remain ON due to a malfunction in the internal circuitry of the Output Unit, such as a relay or transistor malfunction. Always add any circuits necessary outside of the PLC to ensure the safety of the system in the event that an output fails to go OFF.

When the PLC controls an operation such as the clockwise and counterclockwise operation of a motor and if there is any possibility of an accident or mechanical damage due to faulty PLC operation, provide an external interlock such as the one shown below to prevent both the forward and reverse outputs from turning ON at the same time.
Example


This circuit prevents outputs MC1 and MC2 from both being ON at the same time even if both PLC outputs CIO 100.00 and CIO 100.01 are both ON, so the motor is protected even if the PLC is programmed improperly or malfunctions.

## 3-2 Installation Precautions

## 3-2-1 Installation and Wiring Precautions

Always consider the following factors when installing and wiring the PLC to improve the reliability of the system and make the most of the CP1L functions.

## Ambient Conditions

Installation in Cabinets or Control Panels

## Temperature Control

Do not install the PLC in any of the following locations.

- Locations subject to ambient temperatures lower than $0^{\circ} \mathrm{C}$ or higher than $55^{\circ} \mathrm{C}$.
- Locations subject to drastic temperature changes or condensation.
- Locations subject to ambient humidity lower than $10 \%$ or higher than 90\%.
- Locations subject to corrosive or flammable gases.
- Locations subject to excessive dust, salt, or metal filings.
- Locations that would subject the PLC to direct shock or vibration.
- Locations exposed to direct sunlight.
- Locations that would subject the PLC to water, oil, or chemical reagents.

Always enclose or protect the PLC sufficiently 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 lines.

When the CP1L is being installed in a cabinet or control panel, always provide proper ambient conditions as well as access for operation and maintenance.

The ambient temperature within the enclosure must be within the operating range of $0^{\circ} \mathrm{C}$ to $55^{\circ} \mathrm{C}$. When necessary, take the following steps to maintain the proper temperature.

- Provide enough space for good air flow.
- Do not install the PLC above equipment that generates a large amount of heat, such as heaters, transformers, or high-capacity resistors.
- If the ambient temperature exceeds $55^{\circ} \mathrm{C}$, install a cooling fan or air conditioner.


Accessibility for Operation and Maintenance

- To ensure safe access for operation and maintenance, separate the PLC as much as possible from high-voltage equipment and moving machinery.
- The PLC will be easiest to install and operate if it is mounted at a height of about 1,000 to $1,600 \mathrm{~mm}$.

Caution Do not touch the power supply or the area around the I/O terminals while power is being supplied or immediately after power has been turned OFF. Doing so may result in burns.

1 Caution After the power supply has been turned OFF, wait until the PLC has sufficiently cooled before touching it.

- Do not mount the PLC in a control panel containing high-voltage equipment.
- Install the PLC at least 200 mm from power lines.

Power lines


- Ground the mounting plate between the PLC and the mounting surface.
- The CP1L must be installed in the orientation shown below to ensure adequate cooling.

- Do not install the CP1L in any of the following orientations.



## 3-3 Mounting

## 3-3-1 Mounting in a Panel

When mounting the CP1L CPU Unit in a panel, use either surface installation or DIN Track installation.

Surface Installation

Even if a DIN Track is not used, a CP1L CPU Unit and CP/CPM1A-series Expansion Units or Expansion I/O Units can be mounted using M4 screws.
For restrictions on the number of Expansion Units and Expansion I/O Units that can be connected, refer to 1-2 System Configuration.


## DIN Track Installation

## Using I/O Connecting Cable

The CP1L CPU Unit, Expansion Units, and Expansion I/O Units can be mounted to DIN Track. Secure the DIN Track with screws in at least three places.


When using Expansion Units and Expansion I/O Units, it is possible to use CP1W-CN811 Connecting Cable to arrange the Units in upper and lower rows. The following restrictions apply:

- I/O Connecting Cable can be used in one place only, and not in multiple places.



## Wiring Ducts

Whenever possible, route I/O wiring through wiring ducts. Install the duct so that it is easy to wire from the I/O Units through the duct. It is handy to have the duct at the same height as the PLC.



Note Tighten terminal block screws and cable screws to the following torque.
M4: 1.2 N.m
M3: $0.5 \mathrm{~N} \cdot \mathrm{~m}$

Install the wiring ducts at least 20 mm between the tops of the PLC and any other objects, (e.g., ceiling, wiring ducts, structural supports, devices, etc.) to provide enough space for air circulation and replacement of Units.


## Dimensions

## External Dimensions



| Model | W1 | W2 |
| :--- | :--- | :--- |
| CP1L-L14D $\square-\square$ | 86 | 76 |
| CP1L-L10D $\square-\square$ | 66 | 56 |

Mounting Height

The mounting height is approximately 90 mm .
When a cable is connected to an Option Board, however, the additional height must be factored in. Always allow for the additional height when considering the depth of the control panel in which the PLC is to be mounted.

## 3-3-2 Connecting Expansion Units and Expansion I/O Units

Leave approximately 10 mm of space between the CPU Unit and the Expansion Units or Expansion I/O Units.


## Mounting Method



| Unit |  | A (mm) |
| :--- | :--- | :--- |
| CP1L CPU Unit | 60 I/O points | $185 \pm 0.5$ |
|  | 40 I/O points | $140 \pm 0.5$ |
|  | 30 I/O points | $120 \pm 0.5$ |
|  | 20 I/O points | $76 \pm 0.5$ |
|  | 14 I/O points | $76 \pm 0.5$ |
|  | 10 I/O points | $56 \pm 0.5$ |
| Analog I/O Unit | 40 I/O points | $140 \pm 0.2$ |
|  | 32 outputs | $140 \pm 0.2$ |
|  | 20 I/O points | $76 \pm 0.2$ |
|  | 16 outputs | $76 \pm 0.2$ |
|  | 8 inputs | $56 \pm 0.2$ |
|  | 8 outputs | $56 \pm 0.2$ |
| Temperature Sensor Unit | MAD01 | $56 \pm 0.2$ |
| CompoBus/S I/O Link Unit | MAD11 <br> AD041 <br> DA041 <br> DA021 | $76 \pm 0.2$ |
| DeviceNet I/O Link Unit |  | $76 \pm 0.2$ |

## Space between Units When Expansion I/O Units Are Connected



1,2,3... 1. Remove the cover from the CPU Unit's or the Expansion I/O Unit's expansion connector. Use a flat-blade screwdriver to remove the cover from the Expansion I/O Connector.

2. Insert the Expansion I/O Unit's connecting cable into the CPU Unit's or the Expansion I/O Unit's expansion connector.

3. Replace the cover on the CPU Unit's or the Expansion I/O Unit's expansion connector.


## 3-3-3 DIN Track Installation

1,2,3... 1. Use a screwdriver to pull down the DIN Track mounting pins from the back of the Units, and mount the Units to the DIN Track.

2. Lower the Units so that they catch on the top of the DIN Track, and then press them forward all the way to the DIN Track at the bottom.

3. Press in all of the DIN Track mounting pins to securely lock the Units in place.


## DIN Track

Mount the DIN Track in the control panel with screws in at least three places.

- DIN Track: PFP-50N (50 cm), PFP-100N (100 cm), or PFP-100N2 ( 100 cm )


Secure the DIN Track to the control panel using M4 screws separated by 210 mm (6 holes). The tightening torque is $1.2 \mathrm{~N} \cdot \mathrm{~m}$.


## 3-4 Wiring CP1L CPU Units

Note
(1) Do not remove the protective label from the top of the Unit until wiring has been completed. This label prevents wire strands and other foreign matter from entering the Unit during wiring procedures.
(2) Remove the label after the completion of wiring to ensure proper heat dissipation.

## 3-4-1 Wiring Power Supply and Ground Lines

## CPU Units with AC Power Supply

## Wiring the AC Power Supply and Ground Lines



- Wire a separate circuit for the power supply circuit so that there is no voltage drop from the inrush current that flows when other equipment is turned ON.
- When several CP1L PLCs are being used, it is recommended to wire the PLCs on separate circuits to prevent a voltage drop from the inrush current or incorrect operation of the circuit breaker.
- Use twisted-pair power supply cables to prevent noise from the power supply lines. Adding a 1:1 isolating transformer reduces electrical noise even further.
- Consider the possibility of voltage drops and the allowable current, and always use thick power lines.
- Use round crimp terminals for AC power supply wiring.

- AC Power Supply Provide a power supply of 100 to 240 VAC.
- Use a power supply within the following voltage fluctuation range.

| Power supply voltage | Allowable voltage fluctuation range |
| :---: | :---: |
| 100 to 240 VAC | 85 to 264 VAC |

Note (1) Before connecting the power supply, make sure that the CPU Unit requires an AC power supply and not a DC power supply. The CPU Unit's internal circuitry will be damaged if AC power is mistakenly supplied to a CPU Unit that requires a DC power supply.
(2) The power supply input terminals are at the top of the CPU Unit; the terminals at the bottom of the CPU Unit output 24-VDC power for external devices. The CPU Unit's internal circuitry will be damaged if AC power is mistakenly supplied to a CPU Unit's power supply output terminals.
$\triangle$ Caution Tighten the terminal block screws for the AC power supply to the torque of $0.5 \mathrm{~N} \cdot \mathrm{~m}$. Loose screws may result in fire or malfunction.

- Always ground the ground terminal to $100 \Omega$ or less to protect against electric shock and incorrect operation from electrical noise.
- If one phase of the power supply is grounded, connect the grounded phase to the L2/N terminal.
- The GR terminal is a ground terminal. To prevent electrical shock, use a dedicated ground line ( $2 \mathrm{~mm}^{2} \mathrm{~min}$.) of $100 \Omega$ or less.
- The line ground terminal (LG) is a noise-filtered neutral terminal. If noise is a significant source of errors or if electrical shocks are a problem, connect the line ground terminal (LG) to the ground terminal (GR) and ground both with a ground resistance of $100 \Omega$ or less.
- To prevent electrical shock when short-circuiting between the LG and GR terminals, always use a ground of $100 \Omega$ or less.
- Do not connect ground lines to other devices or to the frame of a building. Doing so will reverse the effectiveness of the ground and instead have a bad influence.


## Isolating Transformer

The PLC's internal noise control is sufficient for the general noise to which power supply lines are subjected. Ground noise can be further reduced by providing the power supply through a $1: 1$ isolating transformer. Leave the isolating transformer's secondary side ungrounded.

## CPU Units with DC Power Supply

## DC Power Supply Wiring



- Use crimp terminals or solid wire for wiring the power supply. Do not connect bare stranded wires directly to terminals.

- M3 self-rising terminal screws are used. Tighten the terminal screws to the torque of $0.5 \mathrm{~N} \cdot \mathrm{~m}$.
- To prevent noise, use a ground of $100 \Omega$ or less.


## DC Power Supply

- Provide a power supply of 20.4 to 26.4 VDC.
- The maximum current consumption is 20 W for CPU Units with 30,40 or 60 I/O points and 13 W for CPU Units with 10,14 or 20 I/O points.
- When the power supply is turned ON, the inrush current is approximately five times the normal current.
- The GR terminal is a ground terminal. To prevent electrical shock, use a dedicated ground line ( $2 \mathrm{~mm}^{2} \mathrm{~min}$.) of $100 \Omega$ or less.

Note (1) Never reverse the positive and negative leads when wiring the power supply terminals.
(2) Supply all power to the power supply terminals from the same source.

## 3-4-2 Wiring Built-in I/O

## Wiring Precautions

## Double-checking I/O

 Specifications
## Electric Wires

## Wiring

Double-check the specifications for the I/O Units. In particular, do not apply a voltage that exceeds the input voltage for Input Units or the maximum switching capacity for Output Units. Doing so may result in breakdown, damage, or fire.
When the power supply has positive and negative terminals, always wire them correctly.

- AWG22 to AWG18 ( 0.32 to $0.82 \mathrm{~mm}^{2}$ ) power lines are recommended. Use cable with a maximum diameter of 1.61 mm including the insulation covering.
- The current capacity of electric wire depends on factors such as the ambient temperature and insulation thickness, as well as the gauge of the conductor.
- M3 self-rising screws are used for all screw terminals including terminal screws for crimp terminal power supply wiring.
- Use crimp terminals or solid wire for wiring.
- Do not connect bare stranded wires directly to terminals.
- Tighten the terminal block screws to the torque of $0.5 \mathrm{~N} \cdot \mathrm{~m}$.
- Use crimp terminals (M3) having the dimensions shown below.

- Wire the Units so that they can be easily replaced.
- Make sure that the I/O indicators are not covered by the wiring.
- Do not place the I/O wiring in the same conduits or ducts as high-voltage or power lines. Inductive noise can cause errors or damage.
- Tighten the terminal screws to the torque of $0.5 \mathrm{~N} \cdot \mathrm{~m}$.

Note
(1) Never apply a voltage that exceeds the input voltage for Input Units or the maximum switching capacity for Output Units.
(2) When the power supply has positive and negative terminals, always wire them correctly.
(3) When required by EC Low Voltage Directive, use reinforced insulation or double insulation on the DC power supply connected to DC-power-supply CPU Units and I/O.
For the DC power supply connected to a DC-power-supply CPU Unit, use a power supply with a minimum output holding time of 10 ms .
(4) Do not pull on the cables or bend the cables beyond their natural limit. Doing either of these may break the cables.


- The circuit below should not be used for I/O devices with a voltage output.


Precautions when Connecting a Two-wire DC Sensor

When using a two-wire sensor with a 24-V DC input device, check that the following conditions have been met. Failure to meet these conditions may result in operating errors.

1,2,3... 1. Relation between voltage when the PLC is ON and the sensor residual voltage:
$\mathrm{V}_{\mathrm{ON}} \leq \mathrm{V}_{\mathrm{CC}}-\mathrm{V}_{\mathrm{R}}$
2. Relation between current when the PLC is ON and sensor control output (load current):
$\mathrm{I}_{\text {OUT }}(\min ) \leq \mathrm{I}_{\text {ON }} \leq \mathrm{I}_{\text {OUT }}(\max )$
$\mathrm{I}_{\mathrm{ON}}=\left(\mathrm{V}_{\mathrm{CC}}-\mathrm{V}_{\mathrm{R}}-1.5\right.$ [PLC internal residual voltage] $\left.{ }^{*}\right) / \mathrm{R}_{\mathrm{IN}}$
When $\mathrm{I}_{\mathrm{ON}}$ is smaller than $\mathrm{I}_{\mathrm{OUT}}(\mathrm{min})$, connect a bleeder resistor R. The bleeder resistor constant can be calculated as follows:
$\mathrm{R} \leq\left(\mathrm{V}_{\mathrm{CC}}-\mathrm{V}_{\mathrm{R}}\right) /\left(\mathrm{I}_{\mathrm{OUT}}(\min )-\mathrm{I}_{\mathrm{ON}}\right)$
Power $\mathrm{W} \geq\left(\mathrm{V}_{\mathrm{Cc}}-\mathrm{V}_{\mathrm{R}}\right)^{2} / \mathrm{R} \times 4$ [allowable margin]
3. Relation between current when the PLC is OFF and sensor leakage current:
$I_{\text {OFF }} \geq I_{\text {leak }}$
Connect a bleeder resistor if $\mathrm{I}_{\text {leak }}$ is greater than $\mathrm{I}_{\text {OFF }}$ Use the following equation to calculate the bleeder resistance constant.
$R \leq R_{\text {IN }} \times V_{\text {OFF }} /\left(I_{\text {leak }} \times R_{\text {IN }}-V_{\text {OFF }}\right)$
Power $\mathrm{W} \geq\left(\mathrm{V}_{\mathrm{CC}}-\mathrm{V}_{\mathrm{R}}\right)^{2} / \mathrm{R} \times 4$ (allowable margin)


Vcc: Power voltage
Von: PLC ON voltage
Voff: PLC OFF voltage
Ion: PLC ON current
loff: PLC OFF current
Rin: PLC input impedance
4. Precautions on Sensor Inrush Current

An incorrect input may occur due to sensor inrush current if a sensor is turned ON after the PLC has started up to the point where inputs are possible. Determine the time required for sensor operation to stabilize after the sensor is turned ON and take appropriate measures, such as inserting into the program a timer delay after turning ON the sensor.

## Program Example

In this example, the sensor's power supply voltage is provided to input bit CIO 0.00 and a $100-\mathrm{ms}$ timer delay (the time required for an OMRON Proximity Sensor to stabilize) is created in the program. After the Completion Flag for the timer turns ON, the sensor input on input bit CIO 0.01 will cause output bit CIO 100.00 to turn ON.


## Output Wiring Precautions

Output Short-circuit Protection

## Connecting to a TTL Circuit

## Inrush Current Considerations

If a load connected to the output terminals is short-circuited, output components and the printed circuit boards may be damaged. To guard against this, incorporate a fuse in the external circuit. Use a fuse with a capacity of about twice the rated output.

A TTL circuit cannot be connected directly to a transistor output because of the transistor's residual voltage. It is necessary to connect a pull-up resistor and a CMOS IC between the two.

When connecting a transistor or triac output to a load having a high inrush current (such as an incandescent lamp), steps must be taken to avoid damage to the transistor or triac. Use either of the following methods to reduce the inrush current.

Example Method 1


Use a dark current of approximately $1 / 3$ the rated current of the incandescent lamp.

Example Method 2


Install a limit resistance.

## 3-4-3 Wiring Safety and Noise Controls

## I/O Signal Wiring

Inductive Loads

Whenever possible, place I/O signal lines and power lines in separate ducts or conduits both inside and outside of the control panel.


If the I/O wiring and power wiring must be routed in the same duct, use shielded cables and connect the shields to the GR terminal to reduce noise.

When an inductive load is connected to an I/O Unit, connect a surge suppressor or diode in parallel with the load as shown below.


Note Use surge suppressors and diodes with the following specifications.

Surge Suppressor Specifications $\qquad$
Resistance: $50 \Omega$
Capacitance: $0.47 \mu \mathrm{~F}$
Voltage: 200 V

Diode Specifications
Breakdown voltage: 3 times load voltage min.
Mean rectification current: 1 A

Noise from External Wiring

Take the following points into account when externally wiring I/O, power supply, and power lines.

- When multi-conductor signal cable is being used, avoid combining I/O wires and other control wires in the same cable.
- If wiring racks are parallel, allow at least 300 mm between them.


Ground to $100 \Omega$ or less

- If the I/O wiring and power cables must be placed in the same duct, they must be shielded from each other using grounded steel sheet metal.


Ground to $100 \Omega$ or less

## 3-5 Wiring CPU Unit I/O

## 3-5-1 I/O Wiring for CPU Units with 60 I/O Points

## Input Wiring (Upper Terminal Block, Removable)

The input circuits have 36 points/common. Use power lines with sufficient current capacity for the COM terminals.


## Output Wiring (Lower Terminal Block, Removable)

Relay Outputs (CP1L-M60DR-A and CP1L-M60DR-D)


AC-power-supply models have a $24-$ VDC output terminals (+/-) on the lower terminal block. They can be used as a DC power supply for the input circuit.

## Sinking Transistor

 Outputs (CP1L-M60DT-A and CP1L-M60DT-D)

AC-power-supply models have a $24-\mathrm{VDC}$ output terminals (+/-) on the lower terminal block. They can be used as a DC power supply for the input circuit.

## Sourcing Transistor

## Outputs (CP1L-M60DT1-D)



## 3-5-2 I/O Wiring for CPU Units with 40 I/O Points

## Input Wiring (Upper Terminal Block, Removable)

The input circuits have 24 points/common. Use power lines with sufficient current capacity for the COM terminals.


## Output Wiring (Lower Terminal Block, Removable)

## Relay Outputs

 (CP1L-M40DR-A and CP1L-M40DR-D)
## Sinking Transistor

 Outputs (CP1L-M40DT-A and CP1L-M40DT-D)

AC-power-supply models have a $24-$ VDC output terminals (+/-) on the lower terminal block. They can be used as a DC power supply for the input circuit.

## Sourcing Transistor

## Outputs (CP1L-M40DT1-D)



## 3-5-3 I/O Wiring for CPU Units with 30 I/O Points

## Input Wiring (Upper Terminal Block, Removable)

The input circuits have 18 points/common. Use power lines with sufficient current capacity for the COM terminals.


Output Wiring (Lower Terminal Block, Removable)
Relay Outputs
(CP1L-M30DR-A and CP1L-M30DR-D)


AC-power-supply models have a $24-\mathrm{VDC}$ output terminals (+/-) on the lower terminal block. They can be used as a DC power supply for the input circuit.

## Sinking Transistor

 Outputs (CP1L-M30DT-A and CP1L-M30DT-D)

AC-power-supply models have a 24-VDC output terminals (+/-) on the lower terminal block. They can be used as a DC power supply for the input circuit.

## Sourcing Transistor

## Outputs (CP1L-M30DT1-D )



## 3-5-4 I/O Wiring for CPU Units with 20 I/O Points

## Input Wiring (Upper Terminal Block, not Removable)

The input circuits have 12 points/common. Use power lines with sufficient current capacity for the COM terminals.


## Output Wiring (Lower Terminal Block, not Removable)

## Relay Outputs

(CP1L-L20DR-A and CP1L-L20DR-D)

Sinking Transistor Outputs (CP1L-L20DT-A and CP1L-L20DT-D)

## Sourcing Transistor

 Outputs (CP1L-L20DT1-D)

AC-power-supply models have a $24-V D C$ output terminals (+/-) on the lower terminal block. They can be used as a DC power supply for the input circuit.


AC-power-supply models have a $24-\mathrm{VDC}$ output terminals (+/-) on the lower terminal block. They can be used as a DC power supply for the input circuit.


## 3-5-5 I/O Wiring for CPU Units with 14 I/O Points

## Input Wiring (Upper Terminal Block, not Removable)

The input circuits have 8 points/common. Use power lines with sufficient current capacity for the COM terminals.


## Output Wiring (Lower Terminal Block, not Removable)



AC-power-supply models have a 24-VDC output terminals (+/-) on the lower terminal block. They can be used as a DC power supply for the input circuit.

Sinking Transistor Outputs (CP1L-L14DT-A and CP1L-L14DT-D)


AC-power-supply models have a $24-$ VDC output terminals (+/-) on the lower terminal block. They can be used as a DC power supply for the input circuit.

## Sourcing Transistor

## Outputs (CP1L-L14DT1-D)



## 3-5-6 I/O Wiring for CPU Units with 10 I/O Points

## Input Wiring (Upper Terminal Block, not Removable)

The input circuits have 6 points/common. Use power lines with sufficient current capacity for the COM terminals.


## Output Wiring (Lower Terminal Block, not Removable)

## Relay Outputs

(CP1L-L10DR-A and CP1L-L10DR-D)

## Sinking Transistor

## Outputs (CP1L-L10DT-A

 and CP1L-L10DT-D)
## Sourcing Transistor

Outputs (CP1L-L10DT1-D)


## 3-5-7 Pulse Input Connection Examples

For a 24-VDC Opencollector Encoder

This example shows the connections to an encoder with phase-A, phase-B, and phase Z inputs.

(Do not use the same I/O power supply as other equipment.)


## 3-5-8 Pulse Output Connection Examples

This example shows a connection to a motor driver. Always check the specifications of the motor driver before actually connecting it.
For open-collector output, use a maximum of 3 m of wiring between the CP1L CPU Unit and the motor driver.
No pulses are output while the pulse output transistor is OFF. For a direction output, OFF indicates that CCW output is in progress.
Do not use the same power supply for both pulse output $24-\mathrm{VDC} / 5-\mathrm{VDC}$ power and other I/O power.


## CW/CCW Pulse Output and Pulse Plus Direction Output

Using a 24-VDC Photocoupler Input Motor Driver (CP1L- $\square \square$ DT-D)


Using a 5-VDC
Photocoupler Input Motor
Driver (CP1L- $\square \square \square$ DT-D)

## Connection Example 1



In this example, a 5-V input motor driver is used with a $24-\mathrm{VDC}$ power supply. Be careful to ensure that the Position Control Unit output current does not damage the input circuit at the motor driver and yet is sufficient to turn it ON.
Take into account the power derating for the $1.6-\mathrm{k} \Omega$ resistance.
Connection Example 2


## 3-6 CP/CPM1A-series Expansion I/O Unit Wiring

CP-series Expansion I/O Units

|  | Model | Inputs | Outputs |
| :---: | :---: | :---: | :---: |
| 40-point I/O Units | CP1W-40EDR CPM1A-40EDR | $\begin{aligned} & 24 \text { 24-VDC } \\ & \text { inputs } \end{aligned}$ | 16 relay outputs |
|  | $\begin{aligned} & \hline \text { CP1W-40EDT } \\ & \text { CPM1A-40EDT } \end{aligned}$ |  | 16 transistor outputs (sinking) |
|  | $\begin{aligned} & \text { CP1W-40EDT1 } \\ & \text { CPM1A-40EDT1 } \end{aligned}$ |  | 16 transistor outputs (sourcing) |
| 32-point Output Units | CP1W-32ER | None | 32 relay outputs |
|  | CP1W-32ET |  | 32 transistor outputs (sinking) |
|  | CP1W-32ET1 |  | 32 transistor outputs (sourcing) |
| $\begin{aligned} & \text { 20-point I/O } \\ & \text { Units } \end{aligned}$ | CP1W-20EDT1 CPM1A-20EDR1 | $\begin{aligned} & 1224-\mathrm{VDC} \\ & \text { inputs } \end{aligned}$ | 8 relay outputs |
|  | $\begin{aligned} & \text { CP1W-20EDT } \\ & \text { CPM1A-20EDT } \end{aligned}$ |  | 8 transistor outputs (sinking) |
|  | $\begin{aligned} & \text { CP1W-20EDT1 } \\ & \text { CPM1A-20EDT1 } \end{aligned}$ |  | 8 transistor outputs (sourcing) |
| 16-point Output Units | $\begin{aligned} & \text { CP1W-16ER } \\ & \text { CPM1A-16ER } \end{aligned}$ | None | 16 relay outputs |
|  | CP1W-16ET |  | 16 transistor outputs (sinking) |
|  | CP1W-16ET1 |  | 16 transistor outputs (sourcing) |
| 8-point Input Units | $\begin{aligned} & \text { CP1W-8ED } \\ & \text { CPM1A-8ED } \end{aligned}$ | $\begin{aligned} & 8 \text { 24-VDC } \\ & \text { inputs } \end{aligned}$ | None |
| 8-point Output Units | $\begin{aligned} & \text { CP1W-8ER } \\ & \text { CPM1A-8ER } \end{aligned}$ | None | 8 relay outputs |
|  | $\begin{aligned} & \text { CP1W-8ET } \\ & \text { CPM1A-8ET } \end{aligned}$ |  | 8 transistor outputs (sinking) |
|  | $\begin{aligned} & \text { CP1W-8ET1 } \\ & \text { CPM1A-8ET1 } \end{aligned}$ |  | 8 transistor outputs (sourcing) |

For details on wiring Expansion Units, refer to SECTION 7 Using Expansion Units and Expansion I/O Units.

## 40-point I/O Units (CP1W/CPM1A-40ED $\square \square$ ) (Terminal Block is not Removable)

 Input Wiring
$\mathrm{ClO} \mathrm{m}+1$
$\mathrm{ClO} \mathrm{m}+2$

## Output Wiring

## CP1W-40EDR/CPM1A-40EDR (Relay Outputs)



CP1W-40EDT/CPM1A-40EDT (Sinking Transistor Outputs)


## CP1W-40EDT1/CP1A-40EDT1 (Sourcing Transistor Outputs)



## 32-point Output Units (CP1W-32E $\square \square$ ) (Terminal Block is not Removable) <br> CP1W-32ER (Relay Outputs)

Upper Terminal Block


Lower Terminal Block


## Output Wiring

CP1W-32ET (Sinking Transistor Outputs)

Upper Terminal Block


Lower Terminal Block


Output Wiring

## CP1W-32ET1 (Sourcing Transistor Outputs)

Upper Terminal Block
Lower Terminal Block


## 20-point I/O Units (CP1W/CPM1A-20ED $\square \square$ ) (Terminal Block is not Removable)

Input Wiring
CP1W-20ED $\square \square / C P M 1 A-20 E D \square \square$

$\mathrm{ClO} \mathrm{m}+1$
Output Wiring
CP1W-20EDR1/CPM1A-20EDR1 (Relay Outputs)


CP1W-20EDT/CPM1A-20EDT (Sinking Transistor Outputs)


## CP1W-20EDT1/CP1A-20EDT1 (Sourcing Transistor Outputs)



## 16-point Output Units (CP1W/CPM1A-16E $\square \square$ ) (Terminal Block is not Removable) <br> Output Wiring <br> CP1W-16ER/CPM1A-16ER (Relay Outputs)



## Output Wiring

## Output Wiring

## CP1W-16ET1 (Sourcing Transistor Outputs)

Upper Terminal Block


Lower Terminal Block


## 8-point Input Units (CP1W/CPM1A-8ED) (Terminal Block is not Removable)

 Input Wiring

## 8-point Output Units (CP1W/CPM1A-8E $\square$ ) (Terminal Block is not Removable)

## Output Wiring

CP1W-8ER/CPM1A-8ER (Relay Outputs)
Unit Upper Terminal Block Unit Lower Terminal Block


Output Wiring

Output Wiring
CP1W-8ET1/CPM1A-8ET1 (Sourcing Transistor Outputs)
Unit Upper Terminal Block Unit Lower Terminal Block


## SECTION 4 <br> I/O Memory Allocation

This section describes the structure and functions of the I/O Memory Areas and Parameter Areas
4-1 Overview of I/O Memory Area ..... 138
4-1-1 I/O Memory Area ..... 138
4-1-2 Overview of the Data Areas ..... 139
4-1-3 Clearing and Holding I/O Memory ..... 143
4-1-4 Hot Start/Hot Stop Functions ..... 143
4-2 I/O Area and I/O Allocations ..... 145
4-2-1 I/O Bits Allocated to CPU Units ..... 146
4-2-2 I/O Bits Allocated to Expansion I/O Units ..... 148
4-2-3 I/O Allocation Examples with Expansion I/O Units ..... 150
4-2-4 I/O Word Allocations to Expansion Units ..... 151
4-3 1:1 Link Area ..... 152
4-4 Serial PLC Link Area ..... 153
4-5 Internal Work Area ..... 153
4-6 Holding Area (H) ..... 154
4-7 Auxiliary Area (A) ..... 155
4-8 TR (Temporary Relay) Area ..... 155
4-9 Timers and Counters ..... 156
4-9-1 Timer Area (T) ..... 156
4-9-2 Counter Area (C) ..... 157
4-9-3 Changing the BCD or Binary Mode for Counters and Timers ..... 159
4-10 Data Memory Area (D) ..... 160
4-11 Index Registers ..... 161
4-11-1 Using Index Registers ..... 165
4-11-2 Precautions for Using Index Registers ..... 167
4-12 Data Registers ..... 169
4-13 Task Flags ..... 171
4-14 Condition Flags ..... 171
4-15 Clock Pulses ..... 173

## 4-1 Overview of I/O Memory Area

## 4-1-1 I/O Memory Area

This region of memory contains the data areas that can be accessed as instruction operands. I/O memory includes the CIO Area, Work Area, Holding Area, Auxiliary Area, DM Area, Timer Area, Counter Area, Task Flag Area, Data Registers, Index Registers, Condition Flag Area, and Clock Pulse Area.


| Area |  |  | Size | Range | Task usage | Allocation | $\begin{array}{\|c\|} \hline \text { Bit } \\ \text { access } \end{array}$ | Word access | Access |  | Change from CXProgrammer | $\underset{\text { bit }}{\text { Forcing }}$ status |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  |  | Read |  |  |  |  |  | Write |  |  |
| $\mathrm{ClO}$Area | I/O Area | Input Area |  | $\begin{array}{\|l} 1,600 \text { bits } \\ \text { (100 } \\ \text { words) } \\ \hline \end{array}$ | $\begin{aligned} & \hline \mathrm{CIO} 0 \text { to } \\ & \mathrm{CIO} 99 \end{aligned}$ | Shared by all tasks | CP1L CPU Units and CPseries Expansion Units or Expansion I/O Units | OK | OK | OK | OK | OK | OK |
|  |  | Output Area | $\begin{array}{\|l} 1,600 \text { bits } \\ \text { (100 } \\ \text { words) } \\ \hline \end{array}$ | $\begin{array}{\|l} \hline \mathrm{CIO} 100 \\ \text { to } \mathrm{CIO} \\ 199 \\ \hline \end{array}$ | OK |  |  | OK | OK | OK | OK | OK |
|  | 1:1 Link Area |  | $\begin{aligned} & 256 \text { bits } \\ & \text { (16 words) } \end{aligned}$ | $\begin{array}{\|l} \hline \mathrm{CIO} 3000 \\ \text { to CIO } \\ 3015 \end{array}$ | 1:1 Links |  | OK | OK | OK | OK | OK | OK |
|  | Serial PLC Link Area |  | 1,440 bits (90 words) | $\begin{aligned} & \hline \text { CIO } 3100 \\ & \text { to CIO } \\ & 3189 \end{aligned}$ | Serial PLC Links |  | OK | OK | OK | OK | OK | OK |
|  | Work Area |  | $\begin{array}{\|l} \hline 14,400 \\ \text { bits (900 } \\ \text { words) } \\ \hline \end{array}$ | $\begin{array}{\|l} \hline \mathrm{CIO} 3800 \\ \text { to CIO } \\ 6143 \end{array}$ | --- |  | OK | OK | OK | OK | OK | OK |
| Work Area |  |  | 8,192 bits (512 words) | $\begin{aligned} & \text { W000 to } \\ & \text { W511 } \end{aligned}$ | --- |  | OK | OK | OK | OK | OK | OK |
| Holding Area |  |  | $\begin{array}{\|l} 8,192 \text { bits } \\ \text { (512 } \\ \text { words) } \\ \hline \end{array}$ | $\begin{aligned} & \text { H000 to } \\ & \text { H511 } \\ & \text { (Note 6) } \end{aligned}$ | --- |  | OK | OK | OK | OK | OK | OK |
| Auxiliary Area |  |  | 15,360 bits (960 words) | A000 to A959 | --- |  | OK | --- | OK | Note 1 | Note 1 | No |
| TR Area |  |  | 16 bits | $\begin{array}{\|l} \text { TR0 to } \\ \text { TR15 } \end{array}$ | --- |  | OK | OK | OK | OK | No | No |
| Data Memory Area |  |  | $32,768$ words | $\begin{aligned} & \text { D00000 } \\ & \text { to } \\ & \text { D32767 } \\ & \text { (Note 7) } \end{aligned}$ | --- |  | No (Note 2) | OK | OK | OK | OK | No |
| Timer Completion Flags |  |  | 4,096 bits | $\begin{array}{\|l} \hline \text { T0000 to } \\ \text { T4095 } \end{array}$ | --- |  | OK | --- | OK | OK | OK | OK |
| Counter Completion Flags |  |  | 4,096 bits | $\begin{array}{\|l\|l} \hline \text { C0000 to } \\ \text { C4095 } \\ \hline \end{array}$ | --- |  | OK | --- | OK | OK | OK | OK |
| Timer PVs |  |  | $\begin{aligned} & 4,096 \\ & \text { words } \end{aligned}$ | $\begin{array}{\|l} \text { T0000 to } \\ \text { T4095 } \\ \hline \end{array}$ | --- |  | --- | OK | OK | OK | OK | $\begin{array}{\|l\|} \hline \text { No } \\ \text { (Note 4) } \\ \hline \end{array}$ |
| Counter PVs |  |  | $\begin{array}{\|l\|l} \hline 4,096 \\ \text { words } \\ \hline \end{array}$ | $\begin{array}{\|l\|l\|} \hline \text { C0000 to } \\ \text { C4095 } \\ \hline \end{array}$ | --- |  | --- | OK | OK | OK | OK | $\begin{array}{\|l\|} \hline \text { No } \\ \text { (Note 5) } \\ \hline \end{array}$ |
| Task Flag Area |  |  | 32 bits | $\begin{array}{\|l\|l\|l\|} \text { TK0 to } \\ \text { TK31 } \end{array}$ | --- |  | OK | --- | OK | No | No | No |
| Index Registers |  |  | 16 registers | $\begin{aligned} & \text { IR0 to } \\ & \text { IR15 } \end{aligned}$ | Function separately in each task (Note 3) | --- | OK | OK | Indirect addressing only | Specific instructions only | No | No |
| Data Registers |  |  | 16 registers | $\begin{aligned} & \text { DR0 to } \\ & \text { DR15 } \end{aligned}$ |  | --- | No | OK | OK | OK | No | No |

## Note

1. A0 to A447 are read only and cannot be written. A448 to A959 are read/write.
2. Bits can be manipulated using TST(350), TSTN(351), SET, SETB(532), RSTB(533), and OUTB(534).
3. Index registers and data registers can be used either individually by task or they can be shared by all the tasks (the default is individual use by task).
4. Timer PVs can be refreshed indirectly by force-setting/resetting the Timer Completion Flags.
5. Counter PVs can be refreshed indirectly by force-setting/resetting the Counter Completion Flags.
6. H 512 to H 1535 are used as a Function Block Holding Area. These words can be used only for function block instances (internally allocated variable area).
7. Data Memory Area for CPU Units with 10, 14 or 20 I/O Points: D0 to D9999 and D32000 to D32767.

## 4-1-2 Overview of the Data Areas

## $\square$ CIO Area

It is not necessary to input the "CIO" acronym when specifying an address in the CIO Area. The CIO Area is generally used for data exchanges, such as I/O refreshing with PLC Units. Words that are not allocated to Units may be used as work words and work bits in the program.


Note The parts of the CIO Area that are labelled "not used" may be used in programming as work bits. In the future, however, unused CIO Area bits may be used when expanding functions. Always use Work Area bits first.

## I/O Area (Inputs: CIO 0 to CIO 99, Outputs: CIO 100 to CIO 199)

These words are allocated to built-in I/O terminals of CP1L CPU Units and CP-series Expansion Units or Expansion I/O Units. Input words and output bits that aren't allocated may be used in programming.

Work Area (W)

Holding Area (H)

## 1:1 Link Area

These bits are used by the 1:1 Link Master and Slave. They are used for data links between CP1L CPU Units and CPM2 $\square$ CPU Units.

## Serial PLC Link Area

These words are allocated for use for data links (Serial PLC Links) with other CP1L CPU Units or CP1H CPU Units. Addresses not used for Serial PLC Links can be used in programming.

## Internal I/O Area

These words can be used in programming; they cannot be used for I/O exchange with external I/O terminals. Be sure to use the work words provided in the Work Area before using words in the Internal I/O Area or other unused words in the CIO Area. It is possible that these words will be assigned to new functions in future versions of the CPU Units. The parts of the CIO Area that are labelled "Not used" are functionally identical to the Internal I/O Area.

Words in the Work Area can be used in programming; they cannot be used for I/O exchange with external I/O terminals. Use this area for work words and bits before any words in the CIO Area.


Note These words should be used first in programming be assigned to new functions in future versions of CP1L CPU Units.

Words in the Holding Area can be used in programming. These words retain their content when the PLC is turned ON or the operating mode is switched between PROGRAM mode and RUN or MONITOR mode.


Note H512 to H1535 are used as a Function Block Holding Area. These words can be used only for function block instances (internally allocated variable area). These words cannot be specified as instruction operands in the user program.

## Auxiliary Area (A)

## Temporary Relay Area

 (TR)Data Memory Area (D)

## Timer Area ( T )

## Counter Area (C)

These words are allocated to specific functions in the system.
Refer to Appendix C Auxiliary Area Allocations by Function and Appendix D Auxiliary Area Allocations by Address for details on the Auxiliary Area.


The TR Area contains bits that record the ON/OFF status of program branches. Refer to the CP1H/CP1L Programming Manual for details.

The DM Area is a multi-purpose data area that is normally accessed only in word-units. These words retain their content when the PLC is turned ON or the operating mode is switched between PROGRAM mode and RUN or MONITOR mode.


There are two parts to the Timer Area: the Timer Completion Flags and the timer Present Values (PVs). Up to 4,096 timers with timer numbers T0 to T4095 can be used.

## Timer Completion Flags

These flags are read as individual bits. A Completion Flag is turned ON by the system when the corresponding timer times out (i.e., when the set time elapses).

## Timer PVs

The PVs are read and written as words (16 bits). The PVs count up or down as the timer operates.

There are two parts to the Counter Area: the Counter Completion Flags and the Counter Present Values (PVs). Up to 4,096 counters with counter numbers C 0 to C 4095 can be used.

## Counter Completion Flags

These flags are read as individual bits. A Completion Flag is turned ON by the system when the corresponding counter counts out (i.e., when the set value is reached).

## Counter PVs

The PVs are read and written as words (16 bits). The PVs count up or down as the counter operates.

## Condition Flags

## Clock Pulses

Task Flag Area (TK)

Index Registers (IR)

Data Registers (DR)
These flags include the Arithmetic Flags, such as the Error Flag and Equals Flag, which indicate the results of instruction execution as well as the Always ON and Always OFF Flags. The Condition Flags are specified with symbols rather than addresses.
The Clock Pulses are turned ON and OFF by the CPU Unit's internal timer. These bits are specified with symbols rather than addresses.
A Task Flag will be ON when the corresponding cyclic task is in executable (RUN) status and OFF when the cyclic task hasn't been executed (INI) or is in standby (WAIT) status.
Index registers (IR0 to IR15) are used to store PLC memory addresses (i.e., absolute memory addresses in RAM) to indirectly address words in I/O memory. The Index Registers can be used separately in each task or they can be shared by all tasks.

Data registers (DR0 to DR15) are used together with Index Registers. When a Data Register is input just before an Index Register, the content of the Data Register is added to the PLC memory address in the Index Register to offset that address. The Data Registers can be used separately in each task or they can be shared by all tasks.

## 4-1-3 Clearing and Holding I/O Memory

| Area |  | Mode changed ${ }^{1}$ |  | Fatal error generated |  |  |  | PLC power turned ON |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Execution of FALS | Other fatal errors |  | PLC Setup set to clear IOM Hold Bit status ${ }^{2}$ |  | PLC Setup set to hold IOM Hold Bit status ${ }^{2}$ |  |
|  |  | IOM Hold Bit OFF | IOM Hold Bit ON | IOM Hold Bit OFF | IOM Hold Bit ON | IOM Hold Bit OFF | IOM Hold Bit ON | IOM Hold Bit OFF | IOM Hold Bit ON | IOM Hold Bit OFF | IOM Hold Bit ON |
| CIO Area | I/O Area |  |  | Cleared | Retained | Retained | Retained | Cleared | Retained | Cleared | Cleared | Cleared | Retained |
|  | Serial PC Link Area |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Internal I/O Area |  |  |  |  |  |  |  |  |  |  |  |  |
| Work Area (W) |  | Cleared | Retained | Retained | Retained | Cleared | Retained | Cleared | Cleared | Cleared | Retained |  |  |
| Holding Area (H) |  | Retained | Retained | Retained | Retained | Retained | Retained | Retained | Retained | Retained | Retained |  |  |
| Auxiliary Area (A) |  | Status treatment depends on address. |  |  |  |  |  |  |  |  |  |  |  |
| Data Memory Area (D) |  | Retained | Retained | Retained | Retained | Retained | Retained | Retained | Retained | Retained | Retained |  |  |
| Timer Completion Flags (T) |  | Cleared | Retained | Retained | Retained | Cleared | Retained | Cleared | Cleared | Cleared | Retained |  |  |
| Timer PVs (T) |  | Cleared | Retained | Retained | Retained | Cleared | Retained | Cleared | Cleared | Cleared | Retained |  |  |
| Counter Completion Flags (C) |  | Retained | Retained | Retained | Retained | Retained | Retained | Retained | Retained | Retained | Retained |  |  |
| Counter PVs (C) |  | Retained | Retained | Retained | Retained | Retained | Retained | Retained | Retained | Retained | Retained |  |  |
| Task Flags (TK) |  | Cleared | Cleared | Retained | Retained | Cleared | Cleared | Cleared | Cleared | Cleared | Cleared |  |  |
| Index Registers (IR) |  | Cleared | Retained | Retained | Retained | Cleared | Retained | Cleared | Cleared | Cleared | Retained |  |  |
| Data Registers (DR) |  | Cleared | Retained | Retained | Retained | Cleared | Retained | Cleared | Cleared | Cleared | Retained |  |  |

Note 1. Mode changed from PROGRAM to RUN/MONITOR or vice-versa.
2. The PLC Setup's IOM Hold Bit Status at Startup setting determines whether the IOM Hold Bit's status is held or cleared when the PLC is turned ON.

## 4-1-4 Hot Start/Hot Stop Functions

## Operating Mode Changes

## Hot Start

Turn ON the IOM Hold Bit to retain all data* in I/O memory when the CPU Unit is switched from PROGRAM mode to RUN/MONITOR mode to start program execution.


## Hot Stop

When the IOM Hold Bit is ON, all data* in I/O memory will also be retained when the CPU Unit is switched from RUN or MONITOR mode to PROGRAM mode to stop program execution.


Note *The following areas of I/O memory will be cleared during mode changes (between PROGRAM and RUN/MONITOR) unless the IOM Hold Bit is ON: the CIO Area (I/O Area, Data Link Area, CPU Bus Unit Area, Special I/O Unit Area, DeviceNet (CompoBus/D) Area, and Internal I/O Areas), Work Area, Timer Completion Flags, and Timer PVs.

Auxiliary Area Flags and Words

| Name | Address | Description |
| :---: | :---: | :--- |
| IOM Hold Bit | A500.12 | Specifies whether the I/O memory will be retained or not <br> when the CPU Unit operating mode is changed <br> (between PROGRAM and RUN/MONITOR) or when the <br> power is cycled. <br> OFF: I/O memory is cleared to 0 when the operating <br> mode is changed. |
| ON: I/O memory is retained when the operating mode |  |  |
| is changed between PROGRAM and RUN or |  |  |
| MONITOR. |  |  |

When the IOM Hold Bit is ON, all outputs from Output Units will be maintained when program execution stops. When the program starts again, outputs will have the same status that they had before the program was stopped and instructions will be executed. (When the IOM Hold Bit is OFF, instructions will be executed after the outputs have been cleared.)

## PLC Power ON

In order for all data* in I/O memory to be retained when the PLC is turned ON, the IOM Hold Bit must be ON and it must be protected in the PLC Setup using the IOM Hold Bit Status at Startup parameter.


## Auxiliary Area Flags and Words

| Name | Address | Description |
| :---: | :---: | :--- |
| IOM Hold Bit | A500.12 | Specifies whether the I/O memory will be retained or <br> not when the CPU Unit operating mode is changed <br> (between PROGRAM and RUN/MONITOR) or when <br> the power is cycled. |
|  | OFF: I/O memory is cleared to 0 when the operating <br> mode is changed. |  |
| ON:I/O memory is retained when the operating <br> mode is changed between PROGRAM and <br> RUN or MONITOR. |  |  |

## PLC Setup

| Name | Description | Setting | Default |
| :---: | :--- | :--- | :--- |
| IOM Hold | To retain all data in I/O <br> Bemory when the PLC <br> Bit Status <br> at Startup <br> is turned ON, set the <br> IOM Hold Bit at startup <br> parameter to hold the <br> status of the I/O Hold Bit. | OFF: The IOM Hold Bit is cleared <br> to O when power is cycled. <br> One status of the IOM Hold <br> Bit is retained when power is <br> cycled. | OFF <br> (Cleared) |

## 4-2 I/O Area and I/O Allocations

Input Bits: CIO 0.00 to CIO 99.15 (100 words)
Output Bits: CIO 100.00 to CIO 199.15 (100 words)
The starting words for inputs and outputs are predetermined for CP1L CPU Unit. Input bits in CIO 0 and CIO 1 and output bits in CIO 100 and CIO 101 are automatically allocated to the built-in I/O on the CPU Unit. CP-series Expansion Units and CP-series Expansion I/O Units are automatically allocated input bits in words starting from CIO 2 and output bits in words starting from CIO 102.

- Allocated Words and Number of Expansion Units and Expansion I/O Units

| CPU Unit | Allocated words |  | $\begin{gathered} \text { Number of } \\ \text { Expansion Units } \\ \text { and Expansion I/O } \\ \text { Units connected } \end{gathered}$ |
| :---: | :---: | :---: | :---: |
|  | Input bits | Output bits |  |
| CPU Unit with 10 I/O points | CIO 0 | CIO 100 | 0 |
| CPU Unit with 14 I/O points | ClO 0 | CIO 100 | 1 |
| CPU Unit with 20 I/O points | ClO 0 | CIO 100 | 1 |
| CPU Unit with 30 I/O points | CIO 0 and CIO 1 | CIO 100 and CIO 101 | 3 |
| CPU Unit with 40 I/O points | CIO 0 and CIO 1 | CIO 100 and CIO 101 | 3 |
| CPU Unit with 60 I/O points | $\begin{aligned} & \mathrm{CIO} 0, \mathrm{CIO} 1, \\ & \mathrm{CIO} 2 \end{aligned}$ | $\begin{aligned} & \text { CIO 100, CIO 101, } \\ & \text { CIO 102 } \end{aligned}$ | 3 |

For example, with a CPU Unit with 40 I/O points, the input bits in CIO 0 and CIO 1 and the outputs bits in CIO 100 and CIO 101 would be allocated to the built-in I/O of the CPU Unit. Input bits in CIO 2 and higher and outputs bits in CIO 102 and higher would be automatically allocated in order to any Expansion Units or Expansion I/O Units connected to the CPU Unit.
When the power to the CPU Unit is turned ON, the CPU Unit checks for any Expansion Units and Expansion I/O Units connected to it and automatically allocates I/O bits. If the order in which the Units are connected is changed, the the bits used in the ladder program will no longer match the bits allocated to the actual Units. Always review the ladder program whenever changing the order in which Units are connected.

## 4-2-1 I/O Bits Allocated to CPU Units

## CPU Unit with 10 I/O Points



## CPU Unit with 14 I/O Points



## CPU Unit with 20 I/O Points



## CPU Unit with 30 I/O Points



## CPU Unit with 40 I/O Points



## CPU Unit with 60 I/O Points



For a CPU Unit with 40 I/O points (shown above), a total of 24 input bits are allocated to the input terminal block. The bits that are allocated are input bits CIO 0.00 to CIO 0.11 (i.e., bits 00 to 11 in CIO 0 ) and input bits CIO 1.00 to CIO 1.11 (i.e., bits 00 to 11 in CIO 1).
In addition, a total of 16 output bits are allocated to the output terminal block. The bits that are allocated are output bits CIO 100.00 to CIO 100.07 (i.e., bits

00 to 07 in CIO 0 ) and output bits CIO 101.00 to CIO 101.07 (i.e., bits 00 to 07 in ClO 1 ).
The upper bits (bits 12 to 15) not used in the input words cannot be used as work bits. Only the bits not used in the output words can be used as work bits.

## 4-2-2 I/O Bits Allocated to Expansion I/O Units

There are Expansion I/O Units for expanding inputs, for expanding outputs, and for expanding both input and outputs. I/O bits starting from bit 00 in the next word after the word allocated to the previous Expansion Unit, Expansion I/O Unit, or CPU Unit are automatically allocated. This word is indicated as "CIO m" for input words and as "CIO n" for output words.

| Unit |  |  | Input bits |  |  | Output bits |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  |  | No. of bits | No. of words | Addresses | No. of bits | No. of words | Addresses |
| Unit with 8 inputs |  | $\begin{aligned} & \text { CP1W-8ED } \\ & \text { CPM1A-8ED } \end{aligned}$ | 8 bits | 1 word | CIO m (bits 00 to 07) | --- | None | None |
| Unit with 8 outputs | Relays | $\begin{aligned} & \text { CP1W-8ER } \\ & \text { CPM1A-8ER } \end{aligned}$ | --- | None | None | 8 bits | 1 word | CIO n (bits 00 to 07) |
|  | Sinking transistors | $\begin{aligned} & \text { CP1W-8ET } \\ & \text { CPM1A-8ET } \end{aligned}$ | --- | None | None | 8 bits | 1 word | ClO n (bits 00 to 07) |
|  | Sourcing transistors | $\begin{aligned} & \text { CP1W-8ET1 } \\ & \text { CPM1A-8ET1 } \end{aligned}$ | --- | None | None | 8 bits | 1 word | ClO n (bits 00 to 07) |
| Unit with 16 outputs | Relays | $\begin{aligned} & \text { CP1W-16ER } \\ & \text { CPM1A-16ER } \end{aligned}$ | --- | None | None | 16 bits | 2 words | CIO n (bits 00 to 07) <br> $\mathrm{ClO} \mathrm{n}+1$ (bits 00 to 07 ) |
|  | Sinking transistors | CP1W-16ET | --- | None | None | 16 bits | 2 words | CIO n (bits 00 to 07) $\mathrm{ClO} \mathrm{n}+1$ (bits 00 to 07) |
|  | Sourcing transistors | CP1W-16ET1 | -- | None | None | 16 bits | 2 words | CIO n (bits 00 to 07) $\mathrm{ClO} \mathrm{n}+1$ (bits 00 to 07) |
| Unit with 20 I/O | Relays | $\begin{aligned} & \text { CP1W-20EDR1 } \\ & \text { CPM1A-20EDR1 } \end{aligned}$ | 12 bits | 1 word | CIO m (bits 00 to 11) | 8 bits | 1 word | ClO n (bits 00 to 07) |
|  | Sinking transistors | $\begin{aligned} & \text { CP1W-20EDT } \\ & \text { CPM1A-20EDT } \end{aligned}$ | 12 bits | 1 word | CIO m (bits 00 to 11) | 8 bits | 1 word | ClO n (bits 00 to 07) |
|  | Sourcing transistors | CP1W-20EDT1 CPM1A-20EDT1 | 12 bits | 1 word | CIO m (bits 00 to 11) | 8 bits | 1 word | ClO n (bits 00 to 07) |
| Unit with 32 outputs | Relays | CP1W-32ER | --- | None | None | 32 bits | 4 words | CIO n (bits 00 to 07) CIO n+1 (bits 00 to 07) CIO n+2 (bits 00 to 07) $\mathrm{CIO} n+3$ (bits 00 to 07) |
|  | Sinking transistors | CP1W-32ET | --- | None | None | 32 bits | 4 words | CIO n (bits 00 to 07) $\mathrm{CIO} n+1$ (bits 00 to 07) $\mathrm{CIO} \mathrm{n}+2$ (bits 00 to 07) CIO n+3 (bits 00 to 07) |
|  | Sourcing transistors | CP1W-32ET1 | --- | None | None | 32 bits | 4 words | CIO n (bits 00 to 07) CIO n+1 (bits 00 to 07) CIO n+2 (bits 00 to 07) CIO n+3 (bits 00 to 07) |
| Unit with 40 I/O | Relays | CP1W-40EDR CPM1A-40EDR | 24 bits | 2 words | CIO m (bits 00 to 11) CIO m+1 (bits 00 to 11) | 16 bits | 2 words | CIO n (bits 00 to 07) $\mathrm{ClO} \mathrm{n}+1$ (bits 00 to 07) |
|  | Sinking transistors | $\begin{aligned} & \hline \text { CP1W-40EDT } \\ & \text { CPM1A-40EDT } \end{aligned}$ | 24 bits | 2 words | CIO m (bits 00 to 11) CIO m+1 (bits 00 to 11) | 16 bits | 2 words | CIO n (bits 00 to 07) $\mathrm{ClO} \mathrm{n}+1$ (bits 00 to 07) |
|  | Sourcing transistors | $\begin{aligned} & \text { CP1W-40EDT1 } \\ & \text { CPM1A-40EDT1 } \end{aligned}$ | 24 bits | 2 words | CIO m (bits 00 to 11) CIO m+1 (bits 00 to 11) | 16 bits | 2 words | CIO n (bits 00 to 07) $\mathrm{CIO} \mathrm{n}+1$ (bits 00 to 07 ) |

## ■ I/O Bit Addresses

Units 8 Input Points (CP1W-8ED/CPM1A-8ED)
Eight input bits are allocated in one word (bits 00 to 07 in CIO m).
Inputs


Only one word (8 bits) is allocated to an 8 -input Expansion Input Unit. No output words are allocated. Input bits 08 to 15 are always cleared by the system and cannot be used as work bits.

Units with 8 Output Points (CP1W-8E $\square \square / \mathrm{CPM} 1 \mathrm{~A}-8 \mathrm{E} \square \square$ )
Eight output bits are allocated in one word (bits 00 to 07 in $\mathrm{ClO} n+1$ ).

Outputs n


Only one word (8 bits) is allocated to an 8-output Expansion Output Unit. No input words are allocated. Output bits 08 to 15 can be used as work bits.

Units with 16 Output Points (CP1W-16E $\square$ /CPM1A-16E $\square \square$ )
Sixteen output bits in two words are allocated in two words (bits 00 to 07 in $\mathrm{ClO} n$ and bits 00 to 07 in $\mathrm{ClO} \mathrm{n}+1$ ).


Two words (16 bits) are allocated to a 16 -output Expansion Output Unit. No input words are allocated. Output bits 08 to 15 can be used as work bits.

Units with 20 I/O Points (CPM1A-20ED $\square \square / 20 E D T / 20 E D \square \square)$
Twelve input bits are allocated in one word (bits 00 to 11 in CIO m). Eight output bits are allocated in one word (bits 00 to 07 in ClOn ).


One input word (12 bits) and one output word (8 bits) are allocated for a 20point Expansion I/O Unit.
Input bits 12 to 15 are always cleared by the system and cannot be used as work bits. Output bits 08 to 15 , however, can be used as work bits.

## Units with 32 Output Points (CP1W-32E $\square$ )

Thirty-two output bits are allocated in four words (bits 00 to 07 in CIO n , bits 00 to 07 in $\mathrm{ClOn} n+1$, bits 00 to 07 in $\mathrm{ClOn} n+2$ and bits 00 to 07 in $\mathrm{CIO} n+3$ ).


Four words (32 bits) are allocated to a 32-output Expansion Output Unit. No input words are allocated. Output bits 08 to 15 can be used as work bits.

## Units with 40 I/O Points (CPM1A-40ED $\square$ (40EDT/40ED $\square \square)$

Twenty-four input bits in two words are allocated (bits 00 to 11 in CIO m and bits 00 to $11 \mathrm{CIO} \mathrm{m}+1$ ). Sixteen output bits in two words are allocated (bits 00 to 07 in CIO n and bits 00 to 07 in $\mathrm{CIO} \mathrm{n}+1$ ).


Two input words ( 24 bits) and two output words ( 16 bits) are allocated to a 40point Expansion I/O Unit. Input bits 12 to 15 cannot be used as work bits. Output bits 08 to 15 , however, can be used as work bits.

## 4-2-3 I/O Allocation Examples with Expansion I/O Units

## Example 1: Maximum I/O Capacity

The configuration shown in this example is for the maximum I/O capacity. It consists of a CPU Unit with 40 I/O points and three Expansion I/O Units, each with 40 I/O points. Up to three Expansion I/O Units can be connected to a CPU Unit with either 30, 40 or 60 I/O points.
When Expansion I/O Units with 40 I/O points are connected, control is possible for up to 160 I/O points, including 96 inputs and 64 outputs.


Output bits


## Example 2: Connecting Expansion I/O Units with Only Inputs or Only Outputs

If Expansion I/O Units with only inputs or only outputs are connected, the input or output word not used by an Expansion I/O Unit is allocated to the next Unit that requires it.


Output bits


## 4-2-4 I/O Word Allocations to Expansion Units

| Unit |  | Input words |  | Output words |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Analog I/O Units | CP1W-MAD11 <br> CPM1A-MAD11 <br> CPM1A-MAD01 | 2 words | ClO m to $\mathrm{ClO} \mathrm{m}+1$ | 1 word | ClO n |
| Analog Input Units | CP1W-AD041 CPM1A-AD041 | 4 words | ClO m to $\mathrm{ClO} \mathrm{m}+3$ | 2 words | ClO n to $\mathrm{ClO} \mathrm{n}+1$ |
| Analog Output Units | CP1W-DA021 | None | --- | 2 words | $\mathrm{CIO} n$ to $\mathrm{ClO} \mathrm{n}+1$ |
|  | CP1W-DA041 CPM1A-DA041 | None | --- | 4 words | $\mathrm{ClO} n$ to $\mathrm{ClO} \mathrm{n}+3$ |
| Temperature Sensor Units | $\begin{aligned} & \text { CP1W-TS001 } \\ & \text { CPM1A-TS001 } \end{aligned}$ | 2 words | ClO m to $\mathrm{ClO} \mathrm{m}+1$ | None | --- |
|  | CP1W-TS002 CPM1A-TS002 | 4 words | ClO m to $\mathrm{ClO} \mathrm{m}+3$ | None | --- |
|  | CP1W-TS101 CPM1A-TS101 | 2 words | ClO m to $\mathrm{ClO} \mathrm{m}+1$ | None | --- |
|  | CP1W-TS102 CPM1A-TS102 | 4 words | ClO m to $\mathrm{ClO} \mathrm{m}+3$ | None | --- |
| DeviceNet I/O Link Units | CPM1A-DRT21 | 2 words | $\mathrm{ClO} \mathrm{m} \mathrm{to} \mathrm{ClO} \mathrm{m}+1$ | 2 words | ClO n to $\mathrm{ClO} \mathrm{n}+1$ |
| CompoBus/S I/O Link Units | CP1W-SRT21 CPM1A-SRT21 | 1 word | CIO m | 1 word | ClO n |

m : Indicates the next input word after the input word allocated to the Expansion Unit, Expansion I/O Unit, or CPU Unit to the left of the current Unit.
n : Indicates the next output word after the output word allocated to the Expansion Unit, Expansion I/O Unit, or CPU Unit to the left of the current Unit.

## ■ I/O Word Allocations to Expansion Units

CPU Unit with 40 I/O Points + TS002 + DA041 + 40ED



## 4-3 1:1 Link Area

The 1:1 Link Area contains 1,024 bits ( 64 words) with addresses ranging from CIO 3000.00 to CIO 3015.15 (CIO 3000 to ClO 3015 ).
These bits are used to create $1: 1$ links (i.e., shared data link areas) by connecting the RS-232C ports of two PLCs, including the CP1L, CPM1A, CPM2A, CPM2B, CPM2C, SRM1(-V2), CQM1H, and C200HX/HG/HE(-Z).

1:1 Links


Refer to 6-3-6 1:1 Links for information on using 1:1 links.

## 4-4 Serial PLC Link Area

The Serial PLC Link Area contains 1,440 bits (90 words) with addresses ranging from CIO 3100.00 to CIO 3189.15 (CIO 3100 to CIO 3189).
Words in the Serial PLC Link Area can be used for data links with other PLCs.
Serial PLC Links exchange data among CPU Units via the built-in RS-232C ports, with no need for special programming.
The Serial PLC Link allocations are set automatically by means of the following PLC Setup in the Polling Unit.

- Serial PLC Link Mode
- Number of Serial PLC Link transfer words
- Maximum Serial PLC Link unit number


Addresses not used for Serial PLC Links can be used in programming, the same as the Work Area.

Bits in the Serial PLC Link Area can be force-set and force-reset.
The contents of the Serial PLC Link Area will be cleared in the following cases:

1. When the operating mode is changed from PROGRAM mode to RUN/MONITOR mode or vice-versa and the IOM Hold Bit is OFF
2. When the power is cycled
3. When the Serial PLC Link Area is cleared from the CX-Programmer
4. When PLC operation is stopped when a fatal error other than an FALS(007) error occurs (The contents of the Serial PLC Link Area will be retained when FALS(007) is executed.)

## 4-5 Internal Work Area

The Internal Work Area contains 512 words with addresses ranging from W0 to W511. These words can be used in programming as work words.
There are unused words in the CIO Area (CIO 3800 to CIO 6143 ) that can also be used in the program, but use any available words in the Work Area first because the unused words in the CIO Area may be allocated to other applications when functions are expanded.

Bits in the Work Area can be force-set and force-reset.
The contents of the Work Area will be cleared in the following cases:

1. When the operating mode is changed from PROGRAM to RUN or MONITOR mode or vice-versa and the IOM Hold Bit is OFF
2. When the power is cycled
3. When the Work Area is cleared from the CX-Programmer.
4. When PLC operation is stopped when a fatal error other than an FALS(007) error occurs. (The contents of the Work Area will be retained when FALS(007) is executed.)

## 4-6 Holding Area (H)

The Holding Area contains 512 words with addresses ranging from HO to H511 (bits H0.00 to H511.15). These words can be used in programming.

Holding Area Initialization

## Self-maintaining Bits

Data in the Holding Area is not cleared when the power is cycled or the PLC's operating mode is changed from PROGRAM mode to RUN or MONITOR mode or vice-versa.
A Holding Area bit will be cleared if it is programmed between IL(002) and ILC(003) and the execution condition for IL(002) is OFF. To keep a bit ON even when the execution condition for IL(002) is OFF, turn ON the bit with the SET instruction just before IL(002).

When a self-maintaining bit is programmed with a Holding Area bit, the selfmaintaining bit won't be cleared even when the power is reset.

Note 1. If a Holding Area bit is not used for the self-maintaining bit, the bit will be turned OFF and the self-maintaining bit will be cleared when the power is reset.
2. If a Holding Area bit is used but not programmed as a self-maintaining bit as in the following diagram, the bit will be turned OFF by execution condition A when the power is reset.

3. H 512 to H 1535 are used as a Function Block Holding Area. These words can be used only for function block instances (internally allocated variable area). These words cannot be specified as instruction operands in the user program.

When a Holding Area bit is used in a KEEP(011) instruction, never use a normally closed condition for the reset input if the input device uses an AC power supply. When the power supply goes OFF or is temporarily interrupted, the input will go OFF before the PLC's internal power supply and the Holding Area bit will be reset.


Instead, use a configuration like the one shown below.


There are no restrictions in the order of using bit address or in the number of N.C. or N.O. conditions that can be programmed.

## 4-7 Auxiliary Area (A)

The Auxiliary Area contains 960 words with addresses ranging from A0 to A959). These words are preassigned as flags and control bits to monitor and control operation.
A0 through A447 are read-only, but A448 through A959 can be read or written from the program or the CX-Programmer.
Refer to Appendix C Auxiliary Area Allocations by Function and Appendix D Auxiliary Area Allocations by Address for Auxiliary Area functions.

## Forcing Bit Status

Read/write bits in the Auxiliary Area cannot be force-set and force-reset continuously.

## 4-8 TR (Temporary Relay) Area

The TR Area contains 16 bits with addresses ranging from TR0 to TR15. These temporarily store the ON/OFF status of an instruction block for branching and are used only with mnemonics. TR bits are useful when there are several output branches and interlocks cannot be used.
The TR bits can be used as many times as required and in any order required as long as the same TR bit is not used twice in the same instruction block.
TR bits can be used only with the OUT and LD instructions. OUT instructions (OUT TR0 to OUT TR15) store the ON OFF status of a branch point and LD instructions recall the stored ON OFF status of the branch point.

TR bits cannot be changed from the CX-Programmer.
In this example, a TR bit is used when two outputs have been directly connected to a branch point.


| Instruction | Operand |
| :--- | :--- |
| LD | 0.00 |
| OR | 0.01 |
| OUT | TR 0 |
| AND | 0.02 |
| OUT | 0.03 |
| LD | TR 0 |
| AND | 0.04 |
| OUT | 0.05 |

In this example, a TR bit is used when an output is connected to a branch point without a separate execution condition.


| Instruction | Operand |
| :--- | :--- |
| LD | 0.00 |
| OUT | TR 0 |
| AND | 0.01 |
| OUT | 0.02 |
| LD | TR 0 |
| OUT | 0.03 |

Note A TR bit is not required when there are no execution conditions after the branch point or there is an execution condition only in the last line of the instruction block.


| Instruction | Operand |
| :--- | :--- |
| LD | 0.00 |
| OUT | 0.01 |
| OUT | 0.02 |


| Instruction | Operand |
| :--- | :--- |
| LD | 0.00 |
| OUT | 0.01 |
| AND | 0.02 |
| OUT | 0.03 |

## 4-9 Timers and Counters

## 4-9-1 Timer Area (T)

The 4,096 timer numbers (T0000 to T 4095 ) are shared by the TIM, TIMX(550), $\mathrm{TIMH}(015)$, $\operatorname{TIMHX}(551)$, TMHH(540), $\operatorname{TIMHHX}(552)$, TTIM(087), TTIMX(555), TIMW(813), TIMWX(816), TMHW(815), and TIMHWX(817) instructions. Timer Completion Flags and present values (PVs) for these instructions are accessed with the timer numbers.
The TIML(542), TIMLX(553), MTIM(543), and MTIMX(554) instructions do not use timer numbers.
When a timer number is used in an operand that requires bit data, the timer number accesses the Completion Flag of the timer. When a timer number is used in an operand that requires word data, the timer number accesses the PV of the timer. Timer Completion Flags can be used as often as necessary as normally open and normally closed conditions and the values of timer PVs can be read as normal word data.
The refresh method for timer PVs can be set from the CX-Programmer to either BCD or binary.

Note It is not recommended to use the same timer number in two timer instructions because the timers will not operate correctly if they are timing simultaneously. (If two or more timer instructions use the same timer number, an error will be generated during the program check, but the timers will operate as long as the instructions are not executed in the same cycle.)

The following table shows when timers will be reset or maintained.

| Instruction name | Effect on PV and Completion Flag |  |  | Operation in Jumps and Interlocks |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  | Mode change ${ }^{1}$ | PLC start-up ${ }^{2}$ | $\begin{aligned} & \hline \text { CNR(545)/CN } \\ & \text { RX(547) } \end{aligned}$ | Jumps (JMP-JME) or Tasks on standby ${ }^{4}$ | Interlocks (IL-ILC) |
| TIMER: TIM/TIMX(550) | $\begin{aligned} & \hline \mathrm{PV} \rightarrow 0 \\ & \text { Flag } \rightarrow \text { OFF } \end{aligned}$ | $\begin{aligned} & \hline \mathrm{PV} \rightarrow 0 \\ & \text { Flag } \rightarrow \text { OFF } \end{aligned}$ | $\begin{aligned} & \text { PV } \rightarrow 9999 \\ & \text { Flag } \rightarrow \text { OFF } \end{aligned}$ | PVs refreshed in operating timers | $\begin{aligned} & \hline \text { PV } \rightarrow \text { SV } \\ & \text { (Reset to SV.) } \\ & \text { Flag } \rightarrow \text { OFF } \end{aligned}$ |
| HIGH-SPEED TIMER: TIMH(015)/TIMHX(551) |  |  |  |  |  |
| ONE-MS TIMER: <br> TMHH(540)/TMHHX(552) |  |  |  |  |  |
| ACCUMULATIVE TIMER: <br> TTIM(087)/TTIMX(555) |  |  |  | PV Maintained | PV Maintained |
| TIMER WAIT: TIMW(813)TIMWX(816) |  |  |  | PVs refreshed in operating timers | --- |
| HIGH-SPEED TIMER WAIT: TMHW(815)/TMHWX(817) |  |  |  |  | --- |

Note 1. If the IOM Hold Bit (A500.12) is ON, the PV and Completion Flag will be retained when a fatal error occurs (including execution of FALS instructions) or the operating mode is changed from PROGRAM mode to RUN or MONITOR mode or vice-versa. The PV and Completion Flag will be cleared when power is cycled.
2. If the IOM Hold Bit (A50012) is ON and the PLC Setup's IOM Hold Bit Status at Startup setting is set to protect the IOM Hold Bit, the PV and Completion Flag will be retained when the PLC's power is cycled.
3. Since the $\operatorname{TIML}(542)$, $\operatorname{TIMLX}(553)$, $\operatorname{MTIM}(543)$, and $\operatorname{MTIMX(554)~instruc-~}$ tions do not use timer numbers, they are reset under different conditions. Refer to the descriptions of these instructions for details.
4. The present value of $\operatorname{TIM}, \operatorname{TIMX}(550), \operatorname{TIMH}(015), \operatorname{TIMHX}(551)$, TMHH(540), TMHHX(552), TIMW(813), TIMWX(816), TMHW(815) and TMHWX(817) timers programmed with timer numbers 0000 to 2047 will be updated even when jumped between JMP and JME instructions or when in a task that is on standby. The present value of timers programmed with timer numbers 2048 to 4095 will be held when jumped or when in a task that is on standby.

## Forcing Bit Status

## Restrictions

Timer Completion Flags can be force-set and force-reset.
Timer PVs cannot be force-set or force-reset, although the PVs can be refreshed indirectly by force-setting/resetting the Completion Flag.

There are no restrictions in the order of using timer numbers or in the number of N.C. or N.O. conditions that can be programmed. Timer PVs can be read as word data and used in programming.

## 4-9-2 Counter Area (C)

The 4,096 counter numbers ( C 0000 to C4095) are shared by the CNT, CNTX(546), CNTR(012), CNTRX(548), CNTW(814), and CNTWX(818) instructions. Counter Completion Flags and present values (PVs) for these instructions are accessed with the counter numbers.
When a counter number is used in an operand that requires bit data, the counter number accesses the Completion Flag of the counter. When a counter number is used in an operand that requires word data, the counter number accesses the PV of the counter.

The refresh method for counter PVs can be set from the CX-Programmer to either BCD or binary. (Refer to the previous page).
It is not recommended to use the same counter number in two counter instructions because the counters will not operate correctly if they are counting simultaneously. If two or more counter instructions use the same counter number, an error will be generated during the program check, but the counters will operate as long as the instructions are not executed in the same cycle.
The following table shows when counter PVs and Completion Flags will be reset.

| Instruction name | Effect on PV and Completion Flag |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | Reset | Mode change | PLC startup | Reset Input | $\begin{aligned} & \text { CNR(545)/CN } \\ & \text { RX(547) } \end{aligned}$ | Interlocks (IL-ILC) |
| COUNTER: CNT/CNTX(546) | $\begin{aligned} & \mathrm{PV} \rightarrow 0 \\ & \text { Flag } \rightarrow \text { OFF } \end{aligned}$ | Maintained | Maintained | Reset | Reset | Maintained |
| REVERSIBLE COUNTER: <br> CNTR(012)/CNTRX(548) |  |  |  |  |  |  |
| COUNTER WAIT: <br> CNTW(814)/CNTWX(818) |  |  |  |  |  |  |

## Forcing Bit Status

## Restrictions

Counter Completion Flags can be force-set and force-reset.
Counter PVs cannot be force-set or force-reset, although the PVs can be refreshed indirectly by force-setting/resetting the Completion Flag.

There are no restrictions in the order of using counter numbers or in the number of N.C. or N.O. conditions that can be programmed. Counter PVs can be read as word data and used in programming.

## 4-9-3 Changing the BCD or Binary Mode for Counters and Timers

The refresh method for set values and present values for timers and counters can be changed from BCD mode (0000 to 9999) to binary method (0000 to FFFF) using the CX-Programmer
This setting is made in common for all tasks for all timers and counters.

1. Right-click New PLC in the project tree and select Properties.

2. Select the Execute Timer/Counter as Binary Option in the PLC Properties Dialog Box. The timers and counters for all tasks will be executed in binary mode.


## 4-10 Data Memory Area (D)

CPU Units with 30, 40 or 60 I/O points: D0 to D32767
CPU Units with 10, 14 or 20 I/O points: D0 to D9999 and D32000 to D32767


This data area is used for general data storage and manipulation and is accessible only by word.
Data in the DM Area is retained when the PLC's power is cycled or the PLC's operating mode is changed from PROGRAM mode to RUN/MONITOR mode or vice-versa.
Although bits in the DM Area cannot be accessed directly, the status of these bits can be accessed with the BIT TEST instructions, TST(350) and TSTN(351).

Bits in the DM Area cannot be force-set or force-reset.
Words in the DM Area can be indirectly addressed in two ways: binary-mode and BCD-mode.

## Binary-mode Addressing (@D)

When a "@" character is input before a DM address, the content of that DM word is treated as binary and the instruction will operate on the DM word at that binary address. The entire DM Area (D0 to D32767) can be indirectly addressed with hexadecimal values 0000 to 7FFF.

## @D100



## BCD-mode Addressing (*D)

When a " $*$ " character is input before a DM address, the content of that DM word is treated as BCD and the instruction will operate on the DM word at that BCD address. Only part of the DM Area (D0 to D09999) can be indirectly addressed with BCD values 0000 to 9999.


Note
(1) If an address between D10000 and D31999 is specified as an operand for a CPU Unit with 10, 14 or 20 I/O Points, an illegal area access error will occur.

DM Fixed Allocation Words for Modbus-RTU Easy Master
(2) If two-word data is accessed from the last address in the DM Area (D9999 for the CP1L-L $\square \mathrm{D} \square-\square$ and D32767 for other CPU Units), the Access Error Flag (P_AER) will turn ON and the data at D9999 or D32767 will not be read or written.

The following DM area words are used as command and response storage areas for the Modbus-RTU Easy Master function.
D32200 to D32299: Serial port 1 on CP1L CPU Unit with M CPU type D32300 to D32399: Serial port 2 on CP1L CPU Unit with M CPU type and serial port 1 on CP1L CPU Unit with L CPU type
For use of these areas, refer to 6-3-3 Modbus-RTU Easy Master Function.

## 4-11 Index Registers

The sixteen Index Registers (IR0 to IR15) are used for indirect addressing. Each Index Register can hold a single PLC memory address, which is the absolute memory address of a word in I/O memory. Use MOVR(560) to convert a regular data area address to its equivalent PLC memory address and write that value to the specified Index Register. (Use MOVRW(561) to set the PLC memory address of a timer/counter PV in an Index Register.)
Note Refer to Appendix E Memory Map for more details on PLC memory addresses.
When an Index Register is used as an operand with a "," prefix, the instruction will operate on the word indicated by the PLC memory address in the Index Register, not the Index Register itself. Basically, the Index Registers are I/O memory pointers.

- All addresses in I/O memory (except Index Registers, Data Registers, and Condition Flags) can be specified seamlessly with PLC memory addresses. It isn't necessary to specify the data area. I/O memory addresses for IR, DR, and Condition Flags, however, cannot be held.
- In addition to basic indirect addressing, the PLC memory address in an Index Register can be offset with a constant or Data Register, auto-incremented, or auto-decremented. These functions can be used in loops to read or write data while incrementing or decrementing the address by one each time that the instruction is executed.
With the offset and increment/decrement variations, the Index Registers can be set to base values with MOVR(560) or MOVRW(561) and then modified as pointers in each instruction.


Note (1) It is possible to specify regions outside of I/O memory and generate an Illegal Access Error when indirectly addressing memory with Index Registers. Refer to Appendix E Memory Map for details on the limits of PLC memory addresses.
(2) When an Instruction Execution Error or an Illegal Access Error is generated during the execution of a certain instruction, the auto-increment/decrement for the rest Index Registers of the instruction will not execute.
(3) An Illegal Access Error will be generated when indirectly addressing memory in D10000 to D31999 with Index Registers for CPU Units with 10,14 or $20 \mathrm{I} / \mathrm{O}$ Points.
The following table shows the variations available when indirectly addressing I/O memory with Index Registers. (IR $\square$ represents an Index Register from IRO to IR15.)

| Variation | Function | Syntax | Example |  |
| :---: | :---: | :---: | :---: | :---: |
| Indirect addressing | The content of IR $\square$ is treated as the PLC memory address of a bit or word. | ,IR $\square$ | LD , IR0 | Loads the bit at the PLC memory address contained in IRO. |
| Indirect addressing with constant offset | The constant prefix is added to the content of IR $\square$ and the result is treated as the PLC memory address of a bit or word. <br> The constant may be any integer from $-2,048$ to 2,047 . | Constant ,IR $\square$ (Include a + or in the constant.) | LD +5, IR0 | Adds 5 to the contents of IRO and loads the bit at that PLC memory address. |
| Indirect addressing with DR offset | The content of the Data Register is added to the content of IR $\square$ and the result is treated as the PLC memory address of a bit or word. | DR $\square, \mathrm{IR} \square$ | $\begin{array}{\|l\|} \hline \text { LD } \\ \text { DRO,IRO } \end{array}$ | Adds the contents of DRO to the contents of IRO and loads the bit at that PLC memory address. |
| Indirect addressing with auto-increment | After referencing the content of IR $\square$ as the PLC memory address of a bit or word, the content is incremented by 1 or 2 . | Increment by 1 : , IR $\square+$ <br> Increment by 2 : <br> ,IR $\square++$ | LD , IR0++ | Loads the bit at the PLC memory address contained in IRO and then increments the content of IRO by 2. |
| Indirect addressing with auto-decrement | The content of IR $\square$ is decremented by 1 or 2 and the result is treated as the PLC memory address of a bit or word. | Decrement by 1 : ,-IR $\square$ <br> Decrement by 2 : ,--IR | LD , --IR0 | Decrements the content of IRO by 2 and then loads the bit at that PLC memory address. |

## Example

Index Register Initialization

This example shows how to store the PLC memory address of a word (CIO 2) in an Index Register (IR0), use the Index Register in an instruction, and use the auto-increment variation.

| MOVR(560) | 2 | IR0 | Stores the PLC memory address of <br> CIO 2 in IR0. |
| :--- | :--- | :--- | :--- |
| MOV(021) | $\# 0001$ | , IR0 | Writes \#0001 to the PLC memory ad- <br> dress contained in IR0. |
| $M O V(021)$ | $\# 0020$ | +1, IR0Reads the content of IR0, adds 1, <br> and writes \#0020 to that PLC memo- <br> ry address. |  |



Note The PLC memory addresses are listed in the diagram above, but it isn't necessary to know the PLC memory addresses when using Index Registers.

Since some operands are treated as word data and others are treated as bit data, the meaning of the data in an Index Register will differ depending on the operand in which it is used.

1,2,3... 1. Word Operand:

| $\operatorname{MOVR}(560)$ | 0000 | IR2 |
| :--- | :--- | :--- |
| $\operatorname{MOV}(021)$ | D0 | , IR2 |

When the operand is treated as a word, the contents of the Index Register are used "as is" as the PLC memory address of a word.
In this example $\operatorname{MOVR}(560)$ sets the PLC memory address of CIO 2 in IR2 and the $\mathrm{MOV}(021)$ instruction copies the contents of DO to CIO 2.
2. Bit Operand:

MOVR(560) 000013 ,IR2
SET +5, IR2
When the operand is treated as a bit, the leftmost 7 digits of the Index Register specify the word address and the rightmost digit specifies the bit number. In this example, MOVR(560) sets the PLC memory address of CIO 13 (0C00D hex) in IR2. The SET instruction adds +5 from bit 13 ( D hex) to this PLC memory address, so it turns ON bit CIO 1.02.

The Index Registers will be cleared in the following cases:

1. When the operating mode is changed from PROGRAM to RUN or MONITOR mode or vice-versa
2. When the power is cycled

## Setting Index Registers

Direct Addressing

Always set the required value in an index register before using it. The contents of an index register will be unpredictable if it is not set in advance.
The contents of an index register is also unpredictable after an interrupt task is started. When using index registers inside an interrupt task, use $\operatorname{MOVR}(560)$ (for anything but timer/counter PVs) or MOVRW(561) (for timer/counter PVs) to set the required value.

When an Index Register is used as an operand without a "," prefix, the instruction will operate on the contents of the Index Register itself (a two-word or "double" value). Index Registers can be directly addressed only in the instructions shown in the following table. Use these instructions to operate on the Index Registers as pointers.
The Index Registers cannot be directly addressed in any other instructions, although they can usually be used for indirect addressing.

| Instruction group | Instruction name | Mnemonic |
| :---: | :---: | :---: |
| Data Movement Instructions | MOVE TO REGISTER | MOVR(560) |
|  | MOVE TIMER/COUNTER PV TO REGISTER | MOVRW(561) |
|  | DOUBLE MOVE | MOVL(498) |
|  | DOUBLE DATA EXCHANGE | XCGL(562) |
| Table Data Processing Instructions | SET RECORD LOCATION | SETR(635) |
|  | GET RECORD NUMBER | GETR(636) |
| Increment/Decrement Instructions | DOUBLE INCREMENT BINARY | ++L(591) |
|  | DOUBLE DECREMENT BINARY | --L(593) |
| Comparison Instructions | DOUBLE EQUAL | =L(301) |
|  | DOUBLE NOT EQUAL | < >L(306) |
|  | DOUBLE LESS THAN | < L(311) |
|  | DOUBLE LESS THAN OR EQUAL | < = L(316) |
|  | DOUBLE GREATER THAN | >L(321) |
|  | DOUBLE GREATER THAN OR EQUAL | > = L(326) |
|  | DOUBLE COMPARE | CMPL(060) |
| Symbol Math Instructions | DOUBLE SIGNED BINARY ADD WITHOUT CARRY | +L(401) |
|  | DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY | -L(411) |

The SRCH(181), MAX(182), and MIN(183) instructions can output the PLC memory address of the word with the desired value (search value, maximum, or minimum) to IRO. In this case, IRO can be used in later instructions to access the contents of that word.

## 4-11-1 Using Index Registers

Processing of multiple (identical) instructions such as consecutive addresses for table data can be merged into one instruction by combining repetitive processing (e.g., FOR(513) and NEXT(514)instructions) with indirect addressing using Index Registers, thereby simplifying programming.


The Index operation uses the following procedure.

1. PLC memory addresses for the addresses in the Index Registers are stored using a MOVR instruction.
2. Operation is then executed by indirectly addressing Index Registers to the operand for Instruction A.
3. The addresses are moved using processing such as adding, subtracting, incrementing, or decrementing the Index Register (see note).
4. Steps 2 and 3 are processed repeatedly until the conditions are met.

Note Adding, subtracting incrementing, or decrementing for the Index Register is performed using one of the following methods.

- Each Type of Indirect Addressing for Index Registers:

Auto-increment (,IR $\square+$ or ,IR $\square++$ ), auto-decrement (,-IR $\square$ or ,--IR $\square$ ), constant offset (constant ,IR $\square$ ), and DR offset (DR $\square, I R \square$ ) for Index Registers

- Instructions for Direct Addressing of Index Registers:

DOUBLE SIGNED BINARY ADD WITHOUT CARRY (+L), DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY (-L), DOUBLE INCREMENT BINARY (++L), DOUBLE DECREMENT BINARY (--L)
Example:


If, for example, instruction A above is a comparison instruction, table data could be read from start to the end of the table to compare all of the data with a specific value. In this way, blocks of user-defined processing can be freely created depending by applying Index Registers.

## ■ Example Using Index Registers

In the following example, TIM instructions for timer numbers 0 to 99 use set values in D100 to D199. This can be achieved by using one TIM instruction, using an index register for the timer number, using another index register for the Completion Flags, and repeatedly executing the TIM instruction to start the timers.

The PLC memory addresses for each TO's PV, Completion Flag, and W0.00 are set in Index Registers IR0, IR1, and IR2 using a MOVRW or MOVR instruction.



## 4-11-2 Precautions for Using Index Registers

## Precautions

Do not use a Index Register until a PLC memory address has been set in the register. The pointer operation will be unreliable if the registers are used without setting their values.
The values in Index Registers are unpredictable at the start of an interrupt task. When an Index Register will be used in an interrupt task, always set a PLC memory address in the Index Register with MOVR(560) or MOVRW(561) before using the register in that task.
Each Index Register task is processed independently, so they do not affect each other. For example, IRO used in Task 1 and IRO used in Task 2 are different. Consequently, each Index Register task has 16 Index Registers.

## Limitations when Using Index Registers

- It is only possible to read the Index Register for the last task executed within the cycle from the CX-Programmer. If using Index Registers with the same number to perform multiple tasks, it is only possible with the CXProgrammer to read the Index Register value for the last task performed within the cycle from the multiple tasks, nor is it possible to write the Index Register value from the CX-Programmer.
- It is not possible to either read or write to the Index Registers using Host Link commands or FINS commands.
- A setting can be made from the CX-Programmer to share Index Registers between tasks. This setting will be enabled uniformly for all Index Registers and Data Registers.


## Sharing Index Registers

The following setting can be made from the PLC Properties Dialog Box on the CX-Programmer to control sharing Index and Data Registers between tasks.


## Monitoring Index Registers

It is possible to monitor Index Registers as follows:
To use the Programming Devices to monitor the final Index Register values for each task, or to monitor the Index Register values using Host Link commands or FINS commands, write a program to store Index Register values from each task to another area (e.g., DM area) at the end of each task, and to read Index Register values from the storage words (e.g., DM area) at the beginning of each task. The values stored for each task in other areas (e.g., DM area) can then be edited using the CX-Programmer, Host Link commands, or FINS commands.

Note Be sure to use PLC memory addresses in Index Registers.


## 4-12 Data Registers

The sixteen Data Registers (DR0 to DR15) are used to offset the PLC memory addresses in Index Registers when addressing words indirectly.
The value in a Data Register can be added to the PLC memory address in an Index Register to specify the absolute memory address of a bit or word in I/O memory. Data Registers contain signed binary data, so the content of an Index Register can be offset to a lower or higher address.

## Forcing Bit Status

## Examples

## Range of Values

## Data Register Initialization

IOM Hold Bit Operation

Normal instructions can be used to store data in Data Registers.
Bits in Data Registers cannot be force-set and force-reset.


The following examples show how Data Registers are used to offset the PLC memory addresses in Index Registers.

LD DRO ,IRO

MOV(021) \#0001 DR0 ,IR1 Adds the contents of DRO to the contents of IR1 and writes \#0001 to that PLC memory address.

The contents of data registers are treated as signed binary data and thus have a range of $-32,768$ to 32,767 .

| Hexadecimal content | Decimal equivalent |
| :--- | :--- |
| 8000 to $F F F F$ | $-32,768$ to -1 |
| 0000 to $7 F F F$ | 0 to 32,767 |

The Data Registers will be cleared in the following cases:

1. When the operating mode is changed from PROGRAM mode to RUN/MONITOR mode or vice-versa and the IOM Hold Bit is OFF
2. When the power is cycled and the IOM Hold Bit is OFF or not protected in the PLC Setup

If the IOM Hold Bit (A500.12) is ON, the Data Registers won't be cleared when a FALS error occurs or the operating mode is changed from PROGRAM mode to RUN/MONITOR mode or vice-versa.

If the IOM Hold BIt (A500.12) is ON and the PLC Setup's "IOM Hold Bit Status at Startup" setting is set to protect the IOM Hold Bit, the Data Registers won't be cleared when the PLC's power supply is reset (ON $\rightarrow$ OFF $\rightarrow$ ON).

Data Registers are normally local to each task. For example, DRO used in task 1 is different from DRO used in task 2. (A PLC Setup setting can be made from the CX-Programmer to share Data Registers between tasks.)
The content of Data Registers cannot be accessed (read or written) from the CX-Programmer.
Do not use Data Registers until a value has been set in the register. The register's operation will be unreliable if they are used without setting their values.
The values in Data Registers are unpredictable at the start of an interrupt task. When a Data Register will be used in an interrupt task, always set a value in the Data Register before using the register in that task.

## 4-13 Task Flags

Task Flags range from TK00 to TK31 and correspond to cyclic tasks 0 to 31. A Task Flag will be ON when the corresponding cyclic task is in executable (RUN) status and OFF when the cyclic task hasn't been executed (INI) or is in standby (WAIT) status.

Note These flags indicate the status of cyclic tasks only, they do not reflect the status of interrupt tasks.

Task Flag Initialization

Forcing Bit Status

The Task Flags will be cleared in the following cases, regardless of the status of the IOM Hold Bit.

1. When the operating mode is changed from PROGRAM mode to RUN/MONITOR mode or vice-versa
2. When the power is cycled.

The Task Flags cannot be force-set and force-reset.

## 4-14 Condition Flags

These flags include the Arithmetic Flags, such as the Error Flag and Equals Flag, which indicate the results of instruction execution.
The Condition Flags are specified with symbols, such as P_CY and P_ER, rather than addresses. The status of these flags reflects the results of instruction execution, but the flags are read-only; they cannot be written directly from instructions or the CX-Programmer.

Note The CX-Programmer treats condition flags as global symbols beginning with P_.

All Condition Flags are cleared when the program switches tasks, so the status of the ER and AER flags are maintained only in the task in which the error occurred.

The Condition Flags cannot be force-set and force-reset.
The following table summarizes the functions of the Condition Flags, although the functions of these flags will vary slightly from instruction to instruction. Refer to the description of the instruction for complete details on the operation of the Condition Flags for a particular instruction.

| Name | Symbol | Function |
| :--- | :--- | :--- |
| Error Flag | P_ER | Turned ON when the operand data in an instruction is incorrect (an <br> instruction processing error) to indicate that an instruction ended <br> because of an error. <br> When the PLC Setup is set to stop operation for an instruction error <br> (Instruction Error Operation), program execution will be stopped and <br> the Instruction Processing Error Flag (A29508) will be turned ON <br> when the Error Flag is turned ON. |
| Access Error Flag | P_AER | Turned ON when an Illegal Access Error occurs. The Illegal Access <br> Error indicates that an instruction attempted to access an area of <br> memory that should not be accessed. <br> When the PLC Setup is set to stop operation for an instruction error <br> (Instruction Error Operation), program execution will be stopped and <br> the Instruction Processing Error Flag (A429510) will be turned ON <br> when the Access Error Flag is turned ON. |


| Name | Symbol |  |
| :--- | :--- | :--- |
| Carry Flag | P_CY | Turned ON when there is a carry in the result of an arithmetic opera- <br> tion or a "1" is shifted to the Carry Flag by a Data Shift instruction. <br> The Carry Flag is part of the result of some Data Shift and Symbol <br> Math instructions. |
| Greater Than Flag | P_GT | Turned ON when the first operand of a Comparison Instruction is <br> greater than the second or a value exceeds a specified range. |
| Equals Flag | P_EQ | Turned ON when the two operands of a Comparison Instruction are <br> equal the result of a calculation is 0. |
| Less Than Flag | P_LT | Turned ON when the first operand of a Comparison Instruction is less <br> than the second or a value is below a specified range. |
| Negative Flag | P_N | Turned ON when the most significant bit (sign bit) of a result is ON. |
| Overflow Flag | P_OF | Turned ON when the result of calculation overflows the capacity of the <br> result word(s). |
| Underflow Flag | P_UF | Turned ON when the result of calculation underflows the capacity of <br> the result word(s). |
| Greater Than or <br> Equals Flag | P_GE | Turned ON when the first operand of a Comparison Instruction is <br> greater than or equal to the second. |
| Not Equal Flag | P_NE | Turned ON when the two operands of a Comparison Instruction are <br> not equal. |
| Less Than or <br> Equals Flag | P_LE | Turned ON when the first operand of a Comparison Instruction is less <br> than or equal to the second. |
| Always ON Flag | P_On | Always ON. (Always 1.) |
| Always OFF Flag | P_Off | Always OFF. (Always 0.) |

## Using the Condition Flags

The Condition Flags are shared by all of the instructions, so their status may change often in a single cycle. Be sure to read the Condition Flags immediately after the execution of instruction, preferably in a branch from the same execution condition.


| Instruction | Operand |
| :--- | :--- |
| LD |  |
| Instruction A |  |
| AND | $=$ |
| Instruction B |  |

Since the Condition Flags are shared by all of the instructions, program operation can be changed from its expected course by interruption of a single task. Be sure to consider the effects of interrupts when writing the program. Refer to SECTION 2 Programming of CS/CJ Series Programming Manual (W394) for more details.
The Condition Flags are cleared when the program switches tasks, so the status of a Condition Flag cannot be passed to another task. For example the status of a flag in task 1 cannot be read in task 2.

## Saving and Loading Condition Flag Status

The CP1-H CPU Units support instructions to save and load the Condition Flag status (CCS(282) and CCL(283)). These can be used to access the status of the Condition Flags at other locations in a task or in a different task.
The following example shows how the Equals Flag is used at a different location in the same task.

Task


## 4-15 Clock Pulses

The Clock Pulses are flags that are turned ON and OFF at regular intervals by the system.

| Name | Symbol | Operation |  |
| :---: | :---: | :---: | :---: |
| 0.02 s Clock Pulse | P_0_02_s |  | ON for 0.01 s OFF for 0.01 s |
| 0.1 s Clock Pulse | P_0_1s |  | ON for 0.05 s OFF for 0.05 s |
| 0.2 s Clock Pulse | P_0_2s |  | ON for 0.1 s OFF for 0.1 s |
| 1 s Clock Pulse | P_1s |  | ON for 0.5 s OFF for 0.5 s |
| 1 min Clock Pulse | P_1min |  | ON for 30 s OFF for 30 s |

The Clock Pulses are specified with symbols rather than addresses.
Note The CX-Programmer treats condition flags as global symbols beginning with P.

The Clock Pulses are read-only; they cannot be overwritten from instructions or the CX-Programmer.
The Clock Pulses are cleared at the start of operation.
Using the Clock Pulses
The following example turns CIO 100.00 ON and OFF at 0.5 s intervals.


| Instruction | Operand |
| :--- | :--- |
| LD | 1 s |
| OUT | 100.00 |

100.00


## SECTION 5 Pulse and Counter Functions

This section describes the CP1L's interrupt and high-speed counter functions.
5-1 High-speed Counters ..... 176
5-1-1 Overview ..... 176
5-1-2 High-speed Counter Specifications ..... 177
5-1-3 Procedure ..... 182
5-1-4 PLC Setup ..... 183
5-1-5 High-speed Counter Terminal Allocation ..... 184
5-1-6 Pulse Input Connection Examples ..... 190
5-1-7 Ladder Program Example ..... 190
5-1-8 Additional Capabilities and Restrictions ..... 193
5-2 Pulse Outputs ..... 197
5-2-1 Overview ..... 197
5-2-2 Pulse Output Specifications ..... 200
5-2-3 Pulse Output Terminal Allocations ..... 201
5-2-4 Pulse Output Patterns ..... 208
5-2-5 Origin Search and Origin Return Functions ..... 220
5-2-6 Origin Return ..... 237
5-2-7 Pulse Output Procedures ..... 239
5-2-8 Instructions Used for Pulse Outputs ..... 241
5-2-9 Variable Duty Factor Pulse Outputs (PWM(891) Outputs) ..... 250
5-2-10 Example Pulse Output Applications ..... 250
5-3 Inverter Positioning ..... 280
5-3-1 Features ..... 280
5-3-2 System Configuration ..... 283
5-3-3 Functional Overview ..... 284
5-3-4 Specifications ..... 286
5-3-5 Application Procedure for Inverter Positioning ..... 288
5-3-6 Instruction Specifications ..... 289
5-3-7 Determining the Internal Pulse Output Frequency ..... 295
5-3-8 PLC Setup ..... 297
5-3-9 Automatic Calculation of Inverter Frequency Command Value ..... 302
5-3-10 Memory Allocations ..... 305
5-3-11 Application Example with Serial Communications ..... 317
5-3-12 Application Example with an Analog Output ..... 327
5-3-13 Supplemental Information ..... 336

## 5-1 High-speed Counters

## 5-1-1 Overview

- A rotary encoder can be connected to a built-in input to produce a highspeed pulse input.
- The $\operatorname{PRV}(881)$ instruction can be used to measure the input pulse frequency (one input only).
- The high-speed counter PVs can be maintained or refreshed.
- The High-speed Counter Gate Bit can be turned ON/OFF from the ladder program to select whether the high-speed counter PVs will be maintained or refreshed.
- Any one of the following input signals can be selected as the counter input mode.
Response Frequencies for 24 VDC Inputs to High-speed Counters 0 and 1:
- Differential phase inputs (4x): 50 kHz
- Pulse + direction inputs: 100 kHz
- Up/Down pulse inputs: 100 kHz
- Increment pulse inputs: 100 kHz
- The counting mode can be set to linear mode or circular (ring) mode.
- The counter reset method can be set to $Z$ phase signal + software reset, software reset, $Z$ phase signal + software reset (continue comparing), or software reset (continue comparing).


## Pulse Input Functions

| Purpose | Function used | Description |
| :---: | :---: | :---: |
| Receive incremental rotary encoder inputs to calculate length or position. | High-speed counter function | Built-in input terminals can be used for high-speed counter inputs. <br> The PV for the high-speed counters are stored in the Auxiliary Area. <br> The counters can be operated in ring mode or linear mode. |
| Measure a workpiece's length or position. <br> (Start counting when a certain condition is established or pause counting when a certain condition is established.) | High-speed Counter Gate Bit | The high-speed counter can be started or stopped (PV held) from the Unit's program by turning ON/OFF the High-speed Counter Gate Bit when the desired condition is met. |
| Measure a workpiece's speed from its position data (frequency measurement.) | PRV(881) HIGHSPEED COUNTER PV READ | The PRV(881) instruction can be used to measure the pulse frequency. <br> - Range with differential phase inputs: 0 to 50 kHz (Y models: 0 to 500 kHz ) <br> - Range with all other input modes: 0 to 100 kHz (Y models: 0 to 1 MHz ) |
|  | PRV2(883) PULSE FREQUENCY CONVERT | PRV2(883) reads the pulse frequency and converts it to a rotational speed (r/min) or it converts the counter PV to a total number of rotations. Results are calculated by the number of pulses/ rotation. |

## 5-1-2 High-speed Counter Specifications

## Specifications

|  | Item | Specification |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Number of high-speed counters |  | 2 (High-speed counters 0 and 1) |  |  | 4 (High-speed counters 0 to 3 ) |
| Pulse input modes (Selected in the PLC Setup) |  | Differential phase inputs | Up/down inputs | Pulse + direction inputs | Increment inputs |
| Input terminal allocation |  | Phase-A input | Increment pulse input | Pulse input | Increment pulse input |
|  |  | Phase-B input | Decrement pulse input | Direction input | --- |
|  |  | Phase-Z input | Reset input | Reset input | Reset input |
| Input method |  | Differential phase, 4 x <br> (Fixed) | Two single-phase inputs | Single-phase pulse + direction inputs | Single-phase input |
| Response frequency |  | 50 kHz | 100 kHz | 100 kHz | 100 kHz |
| Counting mode |  | Linear mode or circular (ring) mode (Select in the PLC Setup.) |  |  |  |
| Count values |  | Linear mode: 80000000 to 7FFF FFFF hex <br> Ring mode: 00000000 to Ring SV <br> (The Ring SV (Circular Max. Count) is set in the PLC Setup and the setting range is 00000001 to FFFFFFFF hex.) |  |  |  |
| High-speed counter PV storage locations |  | High-speed counter 0: A271 (leftmost 4 digits) and A270 (rightmost 4 digits) High-speed counter 1: A273 (leftmost 4 digits) and A272 (rightmost 4 digits) High-speed counter 2: A317 (leftmost 4 digits) and A316 (rightmost 4 digits) High-speed counter 3: A319 (leftmost 4 digits) and A318 (rightmost 4 digits) <br> Target value comparison interrupts or range comparison interrupts can be executed based on these PVs. <br> Note The PVs are refreshed in the overseeing processes at the start of each cycle. Use PRV(881) to read the most recent PVs. |  |  |  |
|  |  | Data format: 8 digit hexadecimal <br> Range in linear mode: 80000000 to 7FFF FFFF hex Range in ring mode: 00000000 to Ring SV (Circular Max. Count) |  |  |  |
| Control method | Target value comparison | Up to 48 target values and corresponding interrupt task numbers can be registered. |  |  |  |
|  | Range comparison | Up to 8 ranges can be registered, with a separate upper limit, lower limit, and interrupt task number for each range. |  |  |  |
| Counter reset method |  | Select one of the following methods in the PLC Setup. <br> -Phase-Z + Software reset <br> The counter is reset when the phase-Z input goes ON while the Reset Bit is ON. <br> - Software reset <br> The counter is reset when the Reset Bit goes ON. <br> (Set the counter reset method in the PLC Setup.) <br> Note Operation can be set to stop or continue the comparison operation when the high-speed counter is reset. |  |  |  |

## Auxiliary Area Data Allocation

| Function |  | High-speed counter number |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | 0 | 1 | 2 | 3 |
| PV storage words | Leftmost 4 digits | A271 | A273 | A317 | A319 |
|  | Rightmost 4 digits | A270 | A272 | A316 | A318 |
| Range Comparison Condition Met Flags | Range 1 Comparison Condition Met Flag | A274.00 | A275.00 | A320.00 | A321.00 |
|  | Range 2 Comparison Condition Met Flag | A274.01 | A275.01 | A320.01 | A321.01 |
|  | Range 3 Comparison Condition Met Flag | A274.02 | A275.02 | A320.02 | A321.02 |
|  | Range 4 Comparison Condition Met Flag | A274.03 | A275.03 | A320.03 | A321.03 |
|  | Range 5 Comparison Condition Met Flag | A274.04 | A275.04 | A320.04 | A321.04 |
|  | Range 6 Comparison Condition Met Flag | A274.05 | A275.05 | A320.05 | A321.05 |
|  | Range 7 Comparison Condition Met Flag | A274.06 | A275.06 | A320.06 | A321.06 |
|  | Range 8 Comparison Condition Met Flag | A274.07 | A275.07 | A320.07 | A321.07 |
| Comparison In-progress Flags | ON when a comparison operation is being executed for the high-speed counter. | A274.08 | A275.08 | A320.08 | A321.08 |
| Overflow/Underflow Flags | ON when an overflow or underflow has occurred in the high-speed counter's PV. (Used only when the counting mode is set to Linear Mode.) | A274.09 | A275.09 | A320.09 | A321.09 |
| Count Direction Flags | 0: Decrementing <br> 1: Incrementing | A274.10 | A275.10 | A320.10 | A321.10 |

## Counter Input Modes

Differential Phase Mode (4x)

The differential phase mode uses two phase signals (phase A and phase B) and increments/decrements the count according to the status of these two signals.


Conditions for Incrementing/Decrementing the Count

| Phase A | Phase B | Count value |
| :--- | :--- | :--- |
| $\uparrow$ | L | Increment |
| H | $\uparrow$ | Increment |
| $\downarrow$ | H | Increment |
| L | $\downarrow$ | Increment |
| L | $\uparrow$ | Decrement |
| $\uparrow$ | H | Decrement |
| H | $\downarrow$ | Decrement |
| $\downarrow$ | L | Decrement |

Pulse + Direction Mode
The pulse + direction mode uses a direction signal input and pulse signal input. The count is incremented or decremented depending on the status (ON or OFF) of the direction signal.


Conditions for Incrementing/Decrementing the Count

| Direction <br> signal | Pulse <br> signal | Count value |
| :--- | :--- | :--- |
| $\uparrow$ | L | No change |
| H | $\uparrow$ | Increment |
| $\downarrow$ | H | No change |
| L | $\downarrow$ | No change |
| L | $\uparrow$ | Decrement |
| $\uparrow$ | H | No change |
| H | $\downarrow$ | No change |
| $\downarrow$ | L | No change |

- The count is incremented when the direction signal is ON and decremented when it is OFF.
- Only up-differentiated pulses (rising edges) can be counted.


## Up/Down Mode

The up/down mode uses two signals, an increment pulse input and a decrement pulse input.


Conditions for Incrementing/Decrementing the Count

| Decrement <br> pulse | Increment <br> pulse | Count value |
| :--- | :--- | :--- |
| $\uparrow$ | L | Decrement |
| H | $\uparrow$ | Increment |
| $\downarrow$ | H | No change |
| L | $\downarrow$ | No change |
| L | $\uparrow$ | Increment |
| $\uparrow$ | H | Decrement |
| H | $\downarrow$ | No change |
| $\downarrow$ | L | No change |

- The count is incremented for each increment pulse input and decremented for each decrement pulse input.
- Only up-differentiated pulses (rising edges) can be counted.

Increment Mode
The increment mode counts single-phase pulse signal inputs. This mode only increments the count.


| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |

Conditions for Incrementing/Decrementing the Count

| Pulse | Count value |
| :--- | :--- |
| $\uparrow$ | Increment |
| $H$ | No change |
| $\downarrow$ | No change |
| L | No change |

- Only up-differentiated pulses (rising edges) can be counted.

Note The count of the high-speed counter can be monitored to see if it is currently being incremented or decremented. The count in the current cycle is compared with the count in the previous cycle to determine if it is being incremented or decremented. The results are reflected in the High-speed Counter Count Direction Flags (A274.10 for high-speed counter 0, A275.10 for highspeed Counter 1, A320.10 for high-speed counter 2, and A321.10 for highspeed counter 3.)

## Counting Modes

## Linear Mode

## Circular (Ring) Mode

Input pulses can be counted in the range between the lower limit and upper limit values. If the pulse count goes beyond the lower/upper limit, an underflow/overflow will occur and counting will stop.

## Lower and Upper Limits of the Range

The following diagrams show the lower limit and upper limit values for increment mode and up/down mode.


Input pulses are counted in a loop within the set range. The loop operates as follows:

- If the count is incremented from the max. ring count, the count will be reset to 0 automatically and incrementing will continue.
- If the count is decremented from 0 , the count will be set to the max. ring count automatically and decrementing will continue.
Consequently, underflows and overflows cannot occur when ring mode is used.



## Max. Ring Count

Use the PLC Setup to set the max. ring count (Circular Max. Count), which is the max. value of the input pulse counting range. The max. ring count can be set to any value between 00000001 and FFFFFFFF hex.

## Restrictions

- There are no negative values in ring mode.
- If the max. ring count is set to 0 in the PLC Setup, the counter will operate with a max. ring count of FFFFFFFF hex.


## Reset Methods

Phase-Z Signal + Software Reset

The high-speed counter's PV is reset when the phase-Z signal (reset input) goes from OFF to ON while the corresponding High-speed Counter Reset Bit is ON .

The CPU Unit recognizes the ON status of the High-speed Counter Reset Bit only at the beginning of the PLC cycle during the overseeing processes. Consequently, when the Reset Bit is turned ON in the ladder program, the phase$Z$ signal does not become effective until the next PLC cycle.


The high-speed counter's PV is reset when the corresponding High-speed Counter Reset Bit goes from OFF to ON.
The CPU Unit recognizes the OFF-to-ON transition of the High-speed Counter Reset Bit only at the beginning of the PLC cycle during the overseeing processes. Reset processing is performed at the same time. The OFF-toON transition will not be recognized if the Reset Bit goes OFF again within the same cycle.


Note The comparison operation can be set to stop or continue when a high-speed counter is reset. This enables applications where the comparison operation can be restarted from a counter PV of 0 when the counter is reset.

## 5-1-3 Procedure



PLC Setup settings


## $\downarrow$

|  |
| :--- |
|  |
|  |
|  |
|  |
|  |

- High-speed counters 0 to 3: 24 VDC input, Response frequency: 50 kHz for single-phase, 100 kHz for differential phase
- Pulse input methods: Differential phase (4x), Pulse + direction, Up/Down, or Increment
- Reset methods: Phase-Z + Software reset, Software reset, Phase-Z + Software reset (continuing comparing), Software reset (continuing comparing)
- Counting ranges: Linear mode or Ring mode
- Enable/disable interrupts
- Target value comparison interrupt
- Range comparison interrupt
- Connect to the terminals (24 VDC input or line-driver)
- High-speed Counters 0 to 3 Enable/Disable:
- High-speed Counters 0 to 3Pulse Input Mode:

Differential phase (4x)
Pulse + direction
Up/Down
Increment

- High-speed Counters 0 to 3 Reset Method: Phase-Z + Software reset, Software reset, Phase-Z + Software reset (continuing comparing), Software reset (continuing comparing)
- High-speed Counters 0 to 3 Counting Mode: Linear mode Ring mode
- Program the interrupt task (with any interrupt number between 0 and 255) to be executed when using a target value comparison or range comparison interrupts.
- Register a target value comparison table and start the comparison.
- Register a range comparison table and start the comparison.
- Register a target value comparison table without starting the comparison.
- Register a range comparison table without starting the comparison.
- Change the counter PV.
- Start comparison with the registered target value comparison table or range comparison table.
- Read the high-speed counter PVs, read the status of the highspeed counter comparison operation, or read the rangecomparison results.
- Turn ON the High-speed Counter Gate Bit to stop counting input pulses.


## 5-1-4 PLC Setup

The settings for high-speed counters 0 to 3 are located in the Built-in Input Tab of the CX-Programmer's PLC Settings Window.


Settings in the Builtin Input Tab

| Item | Setting |
| :--- | :--- |
| Use high speed counter 0 to 3 | Use counter |
| Counting mode | Linear mode |
|  | Circular mode (ring mode) |
|  | 0 to 4,294,967,295 (0 to FFFF FFFF hex) |
|  | Phase Z and software reset |
|  | Software reset |
|  | Phase Z and software reset (continue comparing) |
|  | Software reset (continue comparing) |
| Input Setting | Differential phase inputs (4x) |
|  | Pulse + direction inputs |
|  | Up/Down inputs |
|  | Increment pulse input |

## 5-1-5 High-speed Counter Terminal Allocation

Differential Phases, Up/ Down, or Pulse + Direction

The following diagrams show the input terminals that can be used for highspeed counters in each CPU Unit.
Input Terminal Arrangement for CPU Units with 10 I/O Points


## Input Terminal Arrangement for CPU Units with 30 I/O Points



Input Terminal Arrangement for CPU Units with 40 I/O Points


Input Terminal Arrangement for CPU Units with 60 I/O Points


Input Terminal Arrangement for CPU Units with 10 I/O Points


Input Terminal Arrangement for CPU Units with 14 I/O Points


Input Terminal Arrangement for CPU Units with 20 I/O Points


Input Terminal Arrangement for CPU Units with 30 I/O Points


Input Terminal Arrangement for CPU Units with 40 I/O Points


Input Terminal Arrangement for CPU Units with 60 I/O Points


## Input Function Settings in the PLC Setup

The CPU Unit's built-in inputs can be set for use as high-speed counter inputs in the PLC Setup's Built-in Input Tab using the CX-Programmer. (When an input is set for use as a high-speed counter input, the corresponding words and bits cannot be used for general-purpose (normal) inputs, input interrupts, or quick-response inputs.)
CPU Units with 10 I/O Points

| Address |  | Default setting | High-speed counter operation settings |  | Origin searches |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bit |  | Single-phase (increment pulse input) | Two-phase (differential phases x4, up/down, or pulse/direction) | Origin searches enabled for pulse outputs 0 |
| CIO 0 | 00 | Normal input 0 | Counter 0, increment input | Counter 0, A phase, up, or count input | --- |
|  | 01 | Normal input 1 | Counter 1, increment input | Counter 0, B phase, down, or direction input | --- |
|  | 02 | Normal input 2 | Counter 2, increment input | Counter 1, A phase, up, or count input | --- |
|  | 03 | Normal input 3 | Counter 3, increment input | Counter 1, B phase, down, or direction input | Pulse output 0: Origin proximity input signal |
|  | 04 | Normal input 4 | Counter 0, phase-Z/reset input | Counter 0, phase-Z reset input | --- |
|  | 05 | Normal input 5 | Counter 1, phase-Z reset input | Counter 1, phase-Z reset input | Pulse output 0: Origin input signal |

CPU Units with 14 I/O Points

| Input terminal block |  | Default setting | High-speed counter operation settings |  | Origin searches |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bit |  | Single-phase (increment pulse input) | Two-phase (differential phases x4, up/down, or pulse/direction) | Origin searches enabled for pulse outputs 0 and 1 |
| CIO 0 | 00 | Normal input 0 | High-speed counter 0 (Increment) | High-speed counter 0 (Phase A, Increment, or Count input) | --- |
|  | 01 | Normal input 1 | High-speed counter 1 (Increment) | High-speed counter 0 (Phase B, Decrement, or Direction input) | --- |
|  | 02 | Normal input 2 | High-speed counter 2 (Increment) | High-speed counter 1 (Phase A, Increment, or Count input) | Pulse output 0: Origin proximity input signal |
|  | 03 | Normal input 3 | High-speed counter 3 (Increment) | High-speed counter 1 (Phase B, Decrement, or Direction input) | Pulse output 1: Origin proximity input signal |
|  | 04 | Normal input 4 | High-speed counter 0 (Phase Z or reset input) | High-speed counter 0 (Phase Z or reset input) | --- |
|  | 05 | Normal input 5 | High-speed counter 1 (Phase Z or reset input) | High-speed counter 1 (Phase Z or reset input) | --- |
|  | 06 | Normal input 6 | High-speed counter 2 (Phase Z or reset input) | --- | Pulse output 0: Origin input signal |
|  | 07 | Normal input 7 | High-speed counter 3 (Phase Z or reset input) | --- | Pulse output 1: Origin input signal |

CPU Units with 20, 30, 40 or 60 I/O Points

| Address |  | Default setting |  |  |  | High-speed counter operation settings: |  | Origin |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bit | CPU Units with $601 / 0$ Points | CPU Units with $401 / 0$ Points | CPU Units with 30 I/O Points | CPU Units with 20 I/O Points | Single-phase (increment pulse input) | Two-phase (differential phases x4, up/down, or pulse/direction) | Origin searches enabled for pulse outputs 0 and 1 |
| CIO 0 | 00 | Normal input 0 | Normal input 0 | Normal input 0 | Normal input 0 | Counter 0, increment input | Counter 0, A phase, up, or count input | --- |
|  | 01 | Normal input 1 | Normal input 1 | Normal input 1 | Normal input 1 | Counter 1, increment input | Counter 0, B phase, down, or direction input | --- |
|  | 02 | Normal input 2 | Normal input 2 | Normal input 2 | Normal input 2 | Counter 2, increment input | Counter 1, A phase, up, or count input | --- |
|  | 03 | Normal input 3 | Normal input 3 | Normal input 3 | Normal input 3 | Counter 3, increment input | Counter 1, B phase, down, or direction input | --- |
|  | 04 | Normal input 4 | Normal input 4 | Normal input 4 | Normal input 4 | Counter 0, phase-Z reset input | Counter 0, phase-Z reset input | --- |
|  | 05 | Normal input 5 | Normal input 5 | Normal input 5 | Normal input 5 | Counter 1, phase-Z reset input | Counter 1, phase-Z reset input | --- |
|  | 06 | Normal input 6 | Normal input 6 | Normal input 6 | Normal input 6 | Counter 2, phase-Z reset input | --- | Pulse output 0: Origin input signal |
|  | 07 | Normal input 7 | Normal input 7 | Normal input 7 | Normal input 7 | Counter 3, phase-Z reset input | --- | Pulse output 1: Origin input signal |
|  | 08 | Normal input 8 | Normal input 8 | Normal input 8 | Normal input 8 | --- | --- | --- |
|  | 09 | Normal input 9 | Normal input 9 | Normal input 9 | Normal input 9 | --- | --- | --- |
|  | 10 | Normal input 10 | Normal input 10 | Normal input 10 | Normal input 10 | -- | --- | Pulse output 0: Origin proximity input signal |
|  | 11 | Normal input 11 | Normal input 11 | Normal input 11 | Normal input 11 | --- | --- | Pulse output 1: Origin proximity input signal |
| CIO 1 | $\begin{aligned} & \hline 00 \text { to } \\ & 05 \end{aligned}$ | Normal input 12 to 17 | Normal input 12 to 17 | Normal input 12 to 17 | --- | --- | --- | --- |
|  | $\begin{aligned} & \hline 06 \text { to } \\ & 11 \end{aligned}$ | Normal input 18 to 23 | Normal input 18 to 23 | --- | --- | --- | --- | --- |
| CIO 2 | $\begin{aligned} & \hline 00 \text { to } \\ & 11 \end{aligned}$ | Normal input 24 to 35 | --- | --- | --- | --- | --- | --- |

## 5-1-6 Pulse Input Connection Examples

## Encoders with 24 VDC Open-collector Outputs

This example shows how to connect an encoder that has phase-A, phase-B, and phase-Z outputs.


## 5-1-7 Ladder Program Example

Inspecting a Dimension by Counting Pulse Inputs

- This example is for a CPU Unit with 40 I/O Points.
- High-speed counter 0 is used.
- When the edge of the workpiece is detected, the counter PV is reset by a phase-Z pulse.
- The workpiece is passes inspection if the final count is between 30,000 and 30,300 , otherwise the workpiece fails.
- If the workpiece passes, output CIO 100.00 is turned ON by an interrupt and the indicator PL1 is lit. If the workpiece fails, output CIO 100.01 is turned ON by an interrupt and indicator PL2 is lit.
- The interrupt program is interrupt task 10.


## ■ I/O Allocations

Input Terminals

| Input terminal |  | Usage |  |
| :---: | :--- | :--- | :---: |
| Word | Bit |  |  |
| CIO 0 | 00 | High-speed counter 0 phase-A input (See note.) |  |
|  | 01 | High-speed counter 0 phase-B input (See note.) |  |
|  | 02 | Start measurement by pushbutton switch (normal input). |  |
|  | 03 | Detect trailing edge of measured object (normal input). |  |
|  | 04 | Detect leading edge of measured object for high-speed counter 0 <br> phase-Z/reset input (see note). Bit status is reflected in A531.00. |  |
|  | 05 to 11 | Not used. (normal input) |  |
| CIO 1 | 00 to 11 | Not used. (normal input) |  |

Note The high-speed counter inputs are enabled when the Use high speed counter 0 Option is selected in the PLC Setup's Built-in Input Tab.

## Output Terminals

| Output terminal |  | Usage |  |
| :---: | :--- | :--- | :--- |
| Word | Bit |  |  |
| CIO 100 | 00 | Normal input | PL1: Dimension pass output |
|  | 01 | Normal input | PL2: Dimension fail output |
|  | 02 to 07 | Normal input | Not used. |
| CIO 101 | 00 to 07 | Normal input | Not used. |

## Auxiliary Area Addresses for High-speed Counter 0

| Function |  | Address |
| :--- | :--- | :--- |
| PV storage words | Leftmost 4 digits | A271 |
|  | Rightmost 4 digits | A270 |
| Range Comparison <br> Condition Met Flag | Range 1 Comparison Condition Met Flag | A274.00 |
| Comparison In- <br> progress Flag | ON when a comparison operation is being exe- <br> cuted for the high-speed counter. | A274.08 |
| Overflow/Underflow <br> Flag | ON when an overflow or underflow has occurred <br> in the high-speed counter's PV. (Used only when <br> the counting mode is set to Linear Mode.) | A274.09 |
| Count Direction Flag | 0: Decrementing <br> 1: Incrementing | A274.10 |
| Reset Bit | Used for the PV software reset. | A531.00 |
| High-speed Counter <br> Gate Bit | When ON, the counter's PV will not be changed <br> even if pulse inputs are received for the counter. | A531.08 |

## Range Comparison Table

The range comparison table is stored in D10000 to D10039.

## - PLC Setup

Select the Use high speed counter 0 Option in the PLC Setup's Built-in Input Tab.

| Item | Setting |
| :--- | :--- |
| High-speed counter 0 | Use high speed counter 0 |
| Counting mode | Linear mode |
| Circular Max. Count | --- |
| Reset method | Software reset |
| Input Setting | Up/Down inputs |

## - I/O Wiring

## Input Wiring



## Output Wiring



## ■ Range Comparison Table Settings

The inspection standards data is set in the DM Area with the CX-Programmer. Even though range 1 is the only range being used, all 40 words must still be dedicated to the range comparison table.

| Word | Setting | Function |  |
| :---: | :---: | :---: | :---: |
| D10000 | 7430 | Rightmost 4 digits of range 1 lower limit | Lower limit value:30,000 |
| D10001 | 0000 | Leftmost 4 digits of range 1 lower limit |  |
| D10002 | 765C | Rightmost 4 digits of range 1 upper limit | Upper limit value: 30,300 |
| D10003 | 0000 | Leftmost 4 digits of range 1 upper limit |  |
| D10004 | 000A | Range 1 interrupt task number = 10 (A hex) |  |
| $\begin{array}{\|l} \hline \text { D10005 to } \\ \text { D10008 } \\ \hline \end{array}$ | All 0000 | Range 2 lower and upper limit values (Not used and don't need to be set.) | Range 2 settings |
| D10009 | FFFF | Disables range 2. |  |
| ~ |  |  |  |


| Word | Setting | Function |  |
| :---: | :---: | :---: | :---: |
| D10014 <br> D10019 <br> D10024 <br> D10029 <br> D10034 | FFFF | Set the fifth word for ranges 3 to 7 (listed at left) to FFFF to disable those ranges. |  |
| . |  |  |  |
| $\begin{array}{\|l} \hline \text { D10035 to } \\ \text { D10038 } \end{array}$ | All 0000 | Range 8 lower and upper limit values (Not used and don't need to be set.) | Range 8 settings |
| D10039 | FFFF | Disables range 8. |  |

## - Creating the Ladder Program

## Programming in Cyclic Task

Use CTBL(882) to start the comparison operation with high-speed counter 0 and interrupt task 10.


## Programming in Interrupt Task 10

Create the processing performed by interrupt task 10.


## 5-1-8 Additional Capabilities and Restrictions

Restrictions on Highspeed Counter Inputs

- The Phase-Z signal + Software reset method cannot be used when the high speed counters are operating in Differential Phase or Pulse + Direction Input Modes and the origin search function is enabled for the pulse output (in the PLC Setup). The Phase-Z signal + Software reset method can be used when the high speed counters are operating in Incrementing or Up/Down Input Modes.
- When a high-speed counter is being used (enabled in the PLC Setup), the input cannot be used as a general-purpose (normal) input, interrupt input, or quick-response input.


## Starting Interrupt Tasks based on Comparison Conditions

Data registered in advance in a comparison table can be compared with the actual counter PVs during operation. The specified interrupt tasks (registered in the table) will be started when the corresponding comparison condition is met.
There are two comparison methods available: Target value comparison and range comparison.

- Use the CTBL(882) instruction to register the comparison table.
- Use either the CTBL(882) instruction or $\operatorname{INI}(880)$ instruction to start the comparison operation.
- Use the $\mathrm{INI}(880)$ instruction to stop the comparison operation.


## Target Value Comparison

The specified interrupt task is executed when the high-speed counter PV matches a target value registered in the table.

- The comparison conditions (target values and counting directions) are registered in the comparison table along with the corresponding interrupt task number. The specified interrupt task will be executed when the highspeed counter PV matches the registered target value.
- Up to 48 target values (between 1 and 48 ) can be registered in the comparison table.
- A different interrupt task can be registered for each target value.
- The target value comparison is performed on all of the target values in the table, regardless of the order in which the target values are registered.
- If the PV is changed, the changed PV will be compared with the target values in the table, even if the PV is changed while the target value comparison operation is in progress.



## Restrictions

A comparison condition (target value and count direction) cannot appear in the table more than once. An error will occur if a comparison condition is specified two or more times.

Note When the count direction (incrementing/decrementing) changes at a PV that matches a target value, the next target value cannot be matched in that direction.

Set the target values so that they do not occur at the peak or trough of count value changes.


## Range Comparison

The specified interrupt task is executed when the high-speed counter PV is within the range defined by the upper and lower limit values.

- The comparison conditions (upper and lower limits of the range) are registered in the comparison table along with the corresponding interrupt task number. The specified interrupt task will be executed once when the highspeed counter PV is in the range (Lower limit $\leq \mathrm{PV} \leq$ Upper limit).
- A total of 8 ranges (upper and lower limits) are registered in the comparison table.
- The ranges can overlap.
- A different interrupt task can be registered for each range.
- The counter PV is compared with the 8 ranges once each cycle.
- The interrupt task is executed just once when the comparison condition goes from unmet to met.


## Restrictions

When more than one comparison condition is met in a cycle, the first interrupt task in the table will be executed in that cycle. The next interrupt task in the table will be executed in the next cycle.


Note The range comparison table can be used without starting an interrupt task when the comparison condition is met. The range comparison function can be useful when you just want to know whether or not the high-speed counter PV is within a particular range.
Use the Range Comparison Condition Met Flags to determine whether the high-speed counter PV is within a registered range.

## Pausing Input Signal Counting (Gate Function)

If the High-speed Counter Gate Bit is turned ON, the corresponding highspeed counter will not count even if pulse inputs are received and the counter PV will be maintained at its current value. Bits A53108 to A53111 are the High-speed Counter Gate Bits for high-speed counters 0 to 3.

When the High-speed Counter Gate Bit is turned OFF again, the high-speed counter will resume counting and the counter PV will be refreshed.

## Restrictions

- The Gate Bit will be disabled if the high-speed counter's reset method is set to Phase-Z signal + Software reset and the Reset Bit is ON (waiting for the phase- Z input to reset the counter PV.)


## High-speed Counter Frequency Measurement

This function measures the frequency of the high-speed counter (input pulses.)
The input pulse frequency can be read by executing the $\operatorname{PRV}(881)$ instruction. The measured frequency is output in 8-digit hexadecimal and expressed in Hz . The frequency measurement function can be used with high-speed counter 0 only.
The frequency can be measured while a high-speed counter 0 comparison operation is in progress. Frequency measurement can be performed at the same time as functions such as the high-speed counter and pulse output without affecting the performance of those functions.

## Procedure

## Restrictions

> 1,2,3... 1. High-speed Counter Enable/Disable Setting (Required) Select the Use high speed counter O Option in the PLC Setup.
> 2. Pulse Input Mode Setting (Required)
> Set the High-speed Counter O Pulse Input Mode (Input Setting) in the PLC Setup.
> 3. Counting Mode Setting (Required)
> Set the High-speed Counter 0 Counting Mode in the PLC Setup. If ring mode counting is selected, set the High-speed Counter 0 Circular Max. Count (max. ring count) in the PLC Setup.
> 4. Reset Method Setting (Required)
> Set the High-speed Counter 0 Reset Method in the PLC Setup.
> 5. PRV(881) Instruction Execution (Required)
> N: Specify the high-speed counter number. (High-speed counter 0: 0010)
> C: 0003 (Read frequency)
> D: Destination word for frequency data
> - The frequency measurement function can be used with high-speed counter 0 only.

## Specifications

| Item | Specifications |
| :--- | :--- |
| Number of frequency <br> measurement inputs | 1 input (high-speed counter 0 only) |
| Frequency measurement <br> range | High-speed counter 0: <br> Differential phase inputs: 0 to 50 kHz <br> All other input modes: 0 to 100 kHz <br> Note: If the frequency exceeds the maximum value, the <br> maximum value will be stored. |
| Measurement method | Execution of the PRV(881) instruction |
| Output data range | Units: Hz <br> Range: <br> Differential phase input: 00000000 to 00030 D 40 hex <br> (Y models: 0000 0000 to 0007 A120 hex) <br> All other input modes: 00000000 to 0001 86AO hex <br> (Y models: 0000 0000 to 000F 4240 hex) |

## Pulse Frequency Conversion

The pulse frequency input to a high-speed counter can be converted to a rotational speed ( $\mathrm{r} / \mathrm{min}$ ) or the PV of the counter can be converted to the total number of rotations. The converted value is output as 8 -digit hexadecimal. This function is supported only for high-speed counter 0 .

## Frequency-Rotational Speed Conversion

The rotational speed in r/min is calculated from the pulse frequency input to a high-speed counter and the number of pulses per rotation.

## Counter PV-Total Number of Rotations Conversion

The total number of rotations is calculated from the present value of the counter and the number of pulses per rotation.

## Procedure

$$
\begin{aligned}
& \text { 1,2,3... 1. High-speed Counter Enable/Disable Setting (Required) } \\
& \text { Select the Use high speed counter O Option in the PLC Setup. } \\
& \text { 2. Pulse Input Mode Setting (Required) } \\
& \text { Set the High-speed Counter 0 Pulse Input Mode (Input Setting) in the PLC } \\
& \text { Setup. } \\
& \text { 3. Counting Mode Setting (Required) } \\
& \text { Set the High-speed Counter 0 Counting Mode in the PLC Setup. } \\
& \text { If ring mode counting is selected, set the Circular Max. Count (max. ring } \\
& \text { count) in the PLC Setup. } \\
& \text { 4. Reset Method Setting (Required) } \\
& \text { Set the High-speed Counter 0 Reset Method in the PLC Setup. } \\
& \text { 5. Execute PRV2(883) as described below (required). } \\
& \text { Converting the Frequency to a Rotational Speed } \\
& \text { Execute PRV2(883) with the following operands. } \\
& \text { C: Control data (Set to 0000 for frequency-rotational speed conversion.) } \\
& \text { P: Coefficient (pulses/rotation (hex)) } \\
& \text { D: First word for result } \\
& \text { Converting the Counter PV to the Total Number of Rotations } \\
& \text { Execute PRV2(883) with the following operands. } \\
& \text { C: Control data (Set to 0001 for counter PV-total number of rotations con- } \\
& \text { version.) } \\
& \text { P: Coefficient (pulses/rotation (hex)) } \\
& \text { D: First word for result } \\
& \text { Pulse frequency conversion is possible only for high-speed counter 0. }
\end{aligned}
$$

## Restrictions

## 5-2 Pulse Outputs

## 5-2-1 Overview

Fixed duty factor pulses can be output from the CPU Unit's built-in outputs to perform positioning or speed control with a servo driver that accepts pulse inputs.

■ CW/CCW Pulse Outputs or Pulse + Direction Outputs
The pulse output mode can be set to match the motor driver's pulse input specifications.

## ■ Automatic Direction Selection for Easy Positioning with Absolute Coordinates

When operating in absolute coordinates (origin defined or PV changed with the INI(880) instruction), the CW/CCW direction will be selected automatically when the pulse output instruction is executed. (The CW/CCW direction is selected by determining whether the number of pulses specified in the instruction is greater than or less than the pulse output PV.)

## - Triangular Control

Triangular control (trapezoidal control without a constant-speed plateau) will be performed during positioning executed by an ACC(888) instruction (independent) or PLS2(887) instruction if the number of output pulses required for acceleration/deceleration exceeds the specified target pulse Output Amount.

- Change Target Position during Positioning (Multiple Start) When positioning was started with a PULSE OUTPUT (PLS2(887)) instruction and the positioning operation is still in progress, another PLS2(887) instruction can be executed to change the target position, target speed, acceleration rate, and deceleration rate.


## Switch from Speed Control to Positioning (Fixed Distance Feed Interrupt)

A PLS2(887) instruction can be executed during a speed control (continuous mode) operation to change to positioning mode (independent mode). This feature allows a fixed distance feed interrupt (moving a specified amount) to be executed when specific conditions occur.
$\square$ Change Target Speed and Acceleration/Deceleration Rate during Acceleration or Deceleration
When trapezoidal acceleration/deceleration is being executed according to a pulse output instruction (speed control or positioning), the target speed and acceleration/deceleration rate can be changed during acceleration or deceleration.

■ Use Variable Duty Factor Pulse Outputs for Lighting, Power Control, Etc. The PULSE WITH VARIABLE DUTY FACTOR instruction (PWM(891)) can be used to output variable duty factor pulses from the CPU Unit's built-in outputs for applications such as lighting and power control.

## Controlling Pulse Outputs

| Purpose | Function | Description |
| :---: | :---: | :---: |
| Perform simple positioning by outputting pulses to a motor driver that accepts pulse-train inputs. | Pulse output functions <br> - Single-phase pulse output without acceleration/deceleration Controlled by SPED. <br> - Single-phase pulse output with acceleration/deceleration (equal acceleration and deceleration rates for trapezoidal form) Controlled by ACC. <br> - Single-phase pulse output with trapezoidal acceleration/deceleration (Supports a startup frequency and different acceleration/ deceleration rates.) Controlled by PLS2(887). | Built-in outputs can be used as pulse outputs 0 and 1. <br> Target frequency ranges: 1 Hz to 100 kHz <br> Duty factor: 50\% <br> The pulse output mode can be set to CW/CCW pulse control or Pulse plus direction control, but the same output mode must be used for pulse outputs 0 and 1 . <br> Note The pulse output PVs are stored in the Auxiliary Area. |


| Purpose | Function | Description |
| :--- | :--- | :--- |
| $\begin{array}{l}\text { Perform origin search } \\ \text { and origin return opera- } \\ \text { tions. }\end{array}$ | $\begin{array}{l}\text { Origin functions (Origin search and } \\ \text { origin return) }\end{array}$ | $\begin{array}{l}\text { Origin search and origin return operations can be exe- } \\ \text { cuted through pulse outputs. } \\ \text { Origin search: } \\ \text { To start the origin search, set the PLC Setup to }\end{array}$ |
| enable the origin search operation, set the various |  |  |
| origin search parameters, and execute the ORIGIN |  |  |
| SEARCH instruction (ORG(889)). The Unit will deter- |  |  |
| mine the location of the origin based on the Origin |  |  |
| Proximity Input Signal and Origin Input Signal. The |  |  |
| coordinates of the pulse output's PV will automatically |  |  |
| be set as the absolute coordinates. |  |  |$\}$

## 5-2-2 Pulse Output Specifications

## Specifications

| Item | Specifications |
| :--- | :--- |
| Output mode | Continuous mode (for speed control) or independent mode (for position control) |
| Positioning (independent mode) <br> instructions | PULS(886) and SPED(885), PULS(886) and ACC(888), or PLS2(887) |
| Speed control (continuous mode) <br> instructions | SPED(885) or ACC(888) |
| Origin (origin search and origin <br> return) instructions | ORG(889) |
| Output frequency | Pulse outputs 0, 1: 1 Hz to 100 kHz (1 Hz units) |
| Frequency acceleration and decel- <br> eration rates | Set in 1 Hz units for acceleration/deceleration rates from 1 Hz to 65,635 Hz (every 4 <br> ms). The acceleration and deceleration rates can be set independently only with <br> PLS2(887). |
| Changing SVs during instruction <br> execution | The target frequency, acceleration/deceleration rate, and target position can be <br> changed. |
| Duty factor | Fixed at 50\% |
| Pulse output method | CW/CCW inputs or Pulse + direction inputs <br> The method is selected with an instruction operand. The same method must be used <br> for pulse outputs 0 and 1. |
| Number of output pulses | Relative coordinates: 0000 0000 to 7FFF FFFF hex <br> (Each direction accelerating or decelerating: 2,147,483,647) <br> Absolute coordinates: 8000 0000 to 7FFF FFFF hex <br> (-2147483648 to 2147483647) |
| Pulse output PV's relative/absolute <br> coordinate specification | Absolute coordinates are specified automatically when the origin location has been <br> determined by setting the pulse output PV with INI(880) or performing an origin <br> search with ORG(889). Relative coordinates are used when the origin location is <br> undetermined. |
| Pulse output PV's storage location |  |
| Absolute pulse specification | The pulse type can be specified with an operand in PULS(886) or PLS2(887). <br> Note The absolute pulse specification can be used when absolute coordinates are specified for <br> the pulse output PV, i.e. the origin location has been determined. <br> The absolute pulse specification cannot be used when relative coordinates are specified, <br> Pulse output 0: A277 (leftmost 4 digits) and A276 (rightmost 4 digits) <br> Pulse output 1: A279 (leftmost 4 digits) and A278 (rightmost 4 digits) <br> The PVs are refreshed during regular I/O refreshing. |
| Acceleration/deceleration curve <br> specification | Trapezoidal or S-curve acceleration/deceleration |

## Pulse Output Modes

There are two pulse output modes. In independent mode the number of output pulses is specified and in continuous mode the number of output pulses is not specified.

| Mode | Description |
| :--- | :--- |
| Independent mode | This mode is used for positioning. <br> Operation stops automatically when the preset num- <br> ber of pulses has been output. It is also possible to <br> stop the pulse output early with INI(880). |
| Continuous mode | This mode is used for speed control. <br> The pulse output will continue until it is stopped by <br> executing another instruction or switching the PLC to <br> PROGRAM mode. |

## 5-2-3 Pulse Output Terminal Allocations

The following diagrams show the terminals that can be used for pulse outputs in each CPU Unit.

## ■ CPU Unit with 10 I/O Points

Lower Terminal Block
(Example: Transistor Outputs)

$\square$ CPU Unit with 14 I/O Points
Lower Terminal Block
(Example: Transistor Outputs)


CPU Unit with 20 I/O Points
Lower Terminal Block
(Example: Transistor Outputs)


## CPU Unit with 30 I/O Points



## ■ CPU Unit with 40 I/O Points



CPU Unit with 60 I/O Points


Setting Functions Using Instructions and PLC Setup

| Output terminal block |  | When the instructions to the right are not executed | When a pulse output instruction (SPED, ACC, PLS2, or ORG) is executed |  | When the origin search function is enabled in the PLC Setup, and an origin search is executed by the ORG instruction | When the PWM instruction is executed |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bit | Normal output | Fixed duty factor pulse output |  |  | Variable duty factor pulse output |
|  |  |  | CW/CCW | Pulse plus direction | When the origin search function is used | PWM output |
| $\begin{array}{\|l\|} \hline \mathrm{CIO} \\ 100 \end{array}$ | 00 | Normal output 0 | Pulse output 0 (CW) fixed | Pulse output 0 (pulse) fixed | --- | --- |
|  | 01 | Normal output 1 | Pulse output 0 (CCW) fixed | Pulse output 0 (direction) fixed | --- | PWM output 0 |
|  | 02 | Normal output 2 | Pulse output 1 (CW) fixed | Pulse output 1 (pulse) fixed | --- | --- |
|  | 03 | Normal output 3 | Pulse output 1 (CCW) fixed | Pulse output 1 (direction) fixed | --- | PWM output 1 |
|  | 04 | Normal output 4 | --- | --- | Origin search 0 (Error counter reset output) | --- |
|  | 05 | Normal output 5 | --- | --- | Origin search 1 (Error counter reset output) | --- |
|  | 06 | Normal output 6 | --- | --- | --- | --- |
|  | 07 | Normal output 7 | --- | --- | --- | --- |
| $\begin{array}{\|l\|} \hline \mathrm{CIO} \\ 101 \\ \hline \end{array}$ | $00 \text { to }$ $07$ | Normal output 8 to 15 | --- | --- | --- |  |

Input Terminal Block Arrangements
CPU Unit with 10 I/O Points


CPU Unit with 14 I/O Points


CPU Unit with 20 I/O Points


CPU Unit with 30 I/O Points


CPU Unit with 60 I/O Points


- Setting Functions Using Instructions and PLC Setup

CPU Units with 10 I/O Points

| Address |  | Default setting | High-speed counter operation settings |  | Origin searches |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bit |  | Single-phase (increment pulse input) | Two-phase (differential phases x4, up/down, or pulse/direction) | Origin searches enabled for pulse outputs 0 |
| ClO 0 | 00 | Normal input 0 | Counter 0, increment input | Counter 0, A phase, up, or count input | --- |
|  | 01 | Normal input 1 | Counter 1, increment input | Counter 0, B phase, down, or direction input | --- |
|  | 02 | Normal input 2 | Counter 2, increment input | Counter 1, A phase, up, or count input | --- |
|  | 03 | Normal input 3 | Counter 3, increment input | Counter 1, B phase, down, or direction input | Pulse output 0: Origin proximity input signal |
|  | 04 | Normal input 4 | Counter 0, phase-Z/reset input | Counter 0, phase-Z reset input | --- |
|  | 05 | Normal input 5 | Counter 1, phase-Z reset input | Counter 1, phase-Z reset input | Pulse output 0: Origin input signal |

CPU Units with 14 I/O Points

| Input terminal block |  | Default setting | High-speed counter operation settings |  | Origin searches |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bit |  | Single-phase (increment pulse input) | Two-phase (differential phases x4, up/down, or pulse/direction) | Origin searches enabled for pulse outputs 0 and 1 |
| CIO 0 | 00 | Normal input 0 | High-speed counter 0 (Increment) | High-speed counter 0 (Phase A, Increment, or Count input) | --- |
|  | 01 | Normal input 1 | High-speed counter 1 (Increment) | High-speed counter 0 (Phase B, Decrement, or Direction input) | --- |
|  | 02 | Normal input 2 | High-speed counter 2 (Increment) | High-speed counter 1 (Phase A, Increment, or Count input) | Pulse output 0: Origin proximity input signal |
|  | 03 | Normal input 3 | High-speed counter 3 (Increment) | High-speed counter 1 (Phase B, Decrement, or Direction input) | Pulse output 1: Origin proximity input signal |
|  | 04 | Normal input 4 | High-speed counter 0 (Phase Z or reset input) | High-speed counter 0 (Phase Z or reset input) | --- |
|  | 05 | Normal input 5 | High-speed counter 1 (Phase Z or reset input) | High-speed counter 1 (Phase Z or reset input) | --- |
|  | 06 | Normal input 6 | High-speed counter 2 (Phase Z or reset input) | --- | Pulse output 0: Origin input signal |
|  | 07 | Normal input 7 | High-speed counter 3 (Phase Z or reset input) | --- | Pulse output 1: Origin input signal |

CPU Units with 20, 30, 40 or 60 I/O Points

| Address |  | Default setting |  |  |  | High-speed counter operation settings: |  | Origin |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bit | CPU Units with 60 I/O Points | CPU Units with 40 I/O Points | CPU Units with 30 I/O Points | CPU Units with 20 I/O Points | Single-phase (increment pulse input) | Two-phase (differential phases x4, up/down, or pulse/direction) | Origin searches enabled for pulse outputs 0 and 1 |
| CIO 0 | 00 | Normal input 0 | Normal input 0 | Normal input 0 | Normal input 0 | Counter 0, increment input | Counter 0, A phase, up, or count input | --- |
|  | 01 | Normal input 1 | Normal input 1 | Normal input 1 | Normal input 1 | Counter 1, increment input | Counter 0, B phase, down, or direction input | --- |
|  | 02 | Normal input 2 | Normal input 2 | Normal input 2 | Normal input 2 | Counter 2, increment input | Counter 1, A phase, up, or count input | --- |
|  | 03 | Normal input 3 | Normal input 3 | Normal input 3 | Normal input 3 | Counter 3, increment input | Counter 1, B phase, down, or direction input | --- |
|  | 04 | Normal input 4 | Normal input 4 | Normal input 4 | Normal input 4 | Counter 0, phase-Z reset input | Counter 0, phase-Z reset input | --- |
|  | 05 | Normal input 5 | Normal input 5 | Normal input 5 | Normal input 5 | Counter 1, phase-Z reset input | Counter 1, phase-Z reset input | --- |
|  | 06 | Normal input 6 | Normal input 6 | Normal input 6 | Normal input 6 | Counter 2, phase-Z reset input | --- | Pulse output 0: Origin input signal |
|  | 07 | Normal input 7 | Normal input 7 | Normal input 7 | Normal input 7 | Counter 3, phase-Z reset input | --- | Pulse output 1: Origin input signal |
|  | 08 | Normal input 8 | Normal input 8 | Normal input 8 | Normal input 8 | --- | --- | --- |
|  | 09 | Normal input 9 | Normal input 9 | Normal input 9 | Normal input 9 | --- | --- | --- |
|  | 10 | Normal input 10 | Normal input 10 | Normal input 10 | Normal input 10 | --- | --- | Pulse output 0: Origin proximity input signal |
|  | 11 | Normal input 11 | Normal input 11 | Normal input 11 | Normal input 11 | --- | --- | Pulse output 1: Origin proximity input signal |
| CIO 1 | $\begin{array}{\|l} \hline 00 \text { to } \\ 05 \end{array}$ | Normal input 12 to 17 | Normal input 12 to 17 | Normal input 12 to 17 | --- | --- | --- | --- |
|  | $\begin{aligned} & \hline 06 \text { to } \\ & 11 \end{aligned}$ | Normal input 18 to 23 | Normal input 18 to 23 | --- | --- | --- | --- | --- |
| CIO 2 | $\begin{aligned} & \hline 00 \text { to } \\ & 11 \end{aligned}$ | Normal input 24 to 35 | --- | --- | --- | --- | --- | --- |

## Auxiliary Area Data Allocation

| Function |  | Pulse output number |  |
| :--- | :--- | :--- | :--- |
|  | $\mathbf{0}$ | $\mathbf{1}$ |  |
| Pulse output PV storage words <br> PV range: 8000 0000 to 7FFF FFFF hex <br> $(-2,147,483,648$ <br> to 2,147,483,647) | Leftmost 4 digits | A277 | A279 |
| Reset Bits <br> The pulse output PV will be cleared when this bit is <br> turned from OFF to ON. <br> Rightmost 4 digits | 0: Not cleared. <br> 1: Clear PV. | A276 | A278 |


| Function |  | Pulse output number |  |
| :---: | :---: | :---: | :---: |
|  |  | 0 | 1 |
| CW Limit Input Signal Flags <br> This is the CW limit input signal, which is used in the origin search. | ON when turned ON from an external input. | A540.08 | A541.08 |
| CCW Limit Input Signal Flags <br> This is the CCW limit input signal, which is used in the origin search. | ON when turned ON from an external input. | A540.09 | A541.09 |
| Positioning completed input signals <br> This is the positioning completed input signal, which is used in the origin search. | ON when turned ON from an external input. | A540.10 | A541.10 |
| Accel/Decel Flags <br> ON when pulses are being output according to an ACC(888) or PLS2(887) instruction and the output frequency is being changed in steps (accelerating or decelerating). | 0: Constant speed <br> 1: Accelerating or decelerating | A280.00 | A281.00 |
| Overflow/Underflow Flags <br> ON when an overflow or underflow has occurred in the pulse output PV. | 0: Normal <br> 1: Overflow or underflow | A280.01 | A281.01 |
| Output Amount Set Flags <br> ON when the number of output pulses has been set with the PULS instruction. | 0 : No setting <br> 1: Setting made | A280.02 | A281.02 |
| Output Completed Flags <br> ON when the number of output pulses set with the PULS(886)/PLS2(887) instruction has been output. | 0: Output not completed. <br> 1: Output completed. | A280.03 | A281.03 |
| Output In-progress Flags <br> ON when pulses are being output from the pulse output. | 0: Stopped <br> 1: Outputting pulses. | A280.04 | A281.04 |
| No-origin Flags <br> ON when the origin has not been determined for the pulse output. | 0: Origin established. <br> 1: Origin not established. | A280.05 | A281.05 |
| At-origin Flags <br> ON when the pulse output PV matches the origin (0). | 0: Not stopped at origin. <br> 1: Stopped at origin. | A280.06 | A281.06 |
| Output Stopped Error Flags <br> ON when an error occurred while outputting pulses in the origin search function. | 0: No error <br> 1: Stop error occurred. | A280.07 | A281.07 |
| Stop Error Codes <br> When a Pulse Output Stop Error occurs, the error code is stored in that pulse outputs corresponding Stop Error Code word. | --- | A444 | A445 |

## 5-2-4 Pulse Output Patterns

The following tables show the kinds of pulse output operations that can be performed by combining various pulse output instructions.

## Continuous Mode (Speed Control)

Starting a Pulse Output

| Operation | Example application | Frequency changes |  | Description | Procedure |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  |  |  | Instruction | Settings |
| Output with specified speed | Changing the speed (frequency) in one step |  |  |  | Outputs pulses at a specified frequency. | SPED(885) (Continuous) | - Port "CW/CCW" or "Pulse + direction" <br> - Continuous <br> - Target frequency |
| Output with specified acceleration and speed | Accelerating the speed (frequency) at a fixed rate |  |  | Outputs pulses and changes the frequency at a fixed rate. | ACC(888) (Continuous) | - Port <br> -"CW/CCW" or "Pulse + direction" <br> -Continuous <br> - Accelera-tion/deceleration rate <br> - Target frequency |

Changing Settings

| Operation | Example application | Frequency changes |  | Description | Procedure |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  |  |  | Instruction | Settings |
| Change speed in one step | Changing the speed during operation | Pulse freq <br> Target frequency <br> Present frequency | quency |  | Changes the frequency (higher or lower) of the pulse output in one step. | SPED(885) (Continuous) $\downarrow$ SPED(885) (Continuous) | - Port <br> - Continuous <br> - Target frequency |
| Change speed smoothly | Changing the speed smoothly during operation |  |  | Changes the frequency from the present frequency at a fixed rate. The frequency can be accelerated or decelerated. | ACC(888) or SPED(885) (Continuous) $\downarrow$ <br> ACC(888) (Continuous) | - Port <br> -Continuous <br> - Target frequency <br> - Accelera-tion/deceleration rate |
|  | Changing the speed in a polyline curve during operation |  |  | Changes the acceleration or deceleration rate during acceleration or deceleration. | ACC(888) (Continuous) $\downarrow$ <br> ACC(888) (Continuous) | - Port <br> -Continuous <br> -Target frequency <br> - Accelera-tion/deceleration rate |
| Change direction | Not supported. |  |  |  |  |  |
| Change pulse output method | Not supported. |  |  |  |  |  |

## Stopping a Pulse Output

| Operation | Example application | Frequency changes | Description | Procedure |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  |  |  | Instruction | Settings |
| Stop pulse output | Immediate stop |  | Stops the pulse output immediately. | $\begin{array}{\|l} \hline \text { SPED(885) } \\ \text { or ACC(888) } \\ \text { (Continu- } \\ \text { ous) } \\ \downarrow \\ \text { INI(880) } \end{array}$ | - Port <br> - Stop pulse output |
| Stop pulse output | Immediate stop |  | Stops the pulse output immediately. | SPED(885) $\downarrow$ <br> SPED(885) (Continuous) | - Port <br> - Continuous <br> - Target frequency=0 |
| Stop pulse output smoothly | Decelerate to a stop |  | Decelerates the pulse output to a stop. <br> Note If $\mathrm{ACC}(888)$ <br> started the operation, the original acceleration/ deceleration rate will remain in effect. <br> If SPED(885) started the operation, the acceleration/ deceleration rate will be invalid and the pulse output will stop immediately. | SPED(885) or ACC(888) (Continuous) $\downarrow$ <br> ACC(888) (Continuous) | - Port <br> - Continuous <br> - Target frequency=0 |

## Independent Mode (Positioning)

Starting a Pulse Output


## Note Triangular Control

If the specified number of pulses is less than the number required just to reach the target frequency and return to zero, the function will automatically reduce the acceleration/deceleration time and perform triangular control (acceleration and deceleration only.) An error will not occur.


Changing Settings

| Operation | Example application | Frequency changes | Description | Procedure |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  |  |  | Instruction | Settings |
| Change speed in one step | Changing the speed in one step during operation |  | SPED(885) can be executed during positioning to change (raise or lower) the pulse output frequency in one step. <br> The target position (specified number of pulses) is not changed. | PULS(886) <br> SPED(885) <br> (Independent) <br> $\downarrow$ <br> SPED(885) (Independent) | - Number of pulses <br> - Relative or absolute pulse specification <br> - Port <br> -"CW/CCW" or "Pulse + direction" <br> - Independent <br> - Target frequency |
| Change speed smoothly (with acceleration rate = deceleration rate) | Changing the target speed (frequency) during positioning (acceleration rate = deceleration rate) |  | ACC(888) can be executed during positioning to change the acceleration/deceleration rate and target frequency. <br> The target position (specified number of pulses) is not changed. | PULS(886) <br> ACC(888) or SPED(885) (Independent) $\downarrow$ <br> ACC(888) (Independent) <br> PLS2(887) $\downarrow$ <br> ACC(888) (Independent) | - Number of pulses <br> - Relative or absolute pulse specification <br> - Port <br> -"CW/CCW" or "Pulse + direction" <br> - Independent <br> - Acceleration and deceleration rate <br> - Target frequency |
| Change speed smoothly (with unequal acceleration and deceleration rates) | Changing the target speed (frequency) during positioning (different acceleration and deceleration rates) |  | PLS2(887) can be executed during positioning to change the acceleration rate, deceleration rate, and target frequency. <br> Note To prevent the target position from being changed intentionally, the original target position must be specified in absolute coordinates. | PULS(886) <br> ACC(888) <br> (Independent) <br> $\downarrow$ <br> PLS2(887) <br> PLS2(887) <br> $\downarrow$ <br> PLS2(887) | - Number of pulses <br> - Relative or absolute pulse specification <br> - Port <br> -"CW/CCW" or "Pulse + direction" <br> - Acceleration rate <br> - Deceleration rate <br> -Target frequency <br> - Starting frequency |


| Operation | Example application | Frequency changes | Description | Procedure |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  |  |  | Instruction | Settings |
| Change target position | Change the target position during positioning (multiple start function) |  | PLS2(887) can be executed during positioning to change the target position (number of pulses). <br> Note When the target position cannot be changed without maintaining the same speed range, an error will occur and the original operation will continue to the original target position. | PULS(886) <br> $\downarrow$ <br> ACC(888) <br> (Indepen- <br> dent) <br> $\downarrow$ <br> PLS2(887) <br> PLS2(887) <br> $\downarrow$ <br> PLS2(887) <br> PLS2(887) <br> $\downarrow$ <br> PLS2(887) | - Number of pulses <br> - Relative or absolute pulse specification <br> - Port <br> -"CW/CCW" or "Pulse + direction" <br> - Acceleration rate <br> -Deceleration rate <br> - Target frequency <br> - Starting frequency |
| Change target position and speed smoothly | Change the target position and target speed (frequency) during positioning (multiple start function) |  | Note When the settings cannot be changed without maintaining the same speed range, an error will occur and the original operation will continue to the original target position. | $\begin{aligned} & \text { PULS(886) } \\ & \downarrow \\ & \text { ACC(888) } \\ & \text { (Indepen- } \\ & \text { dent) } \\ & \downarrow \\ & \text { PLS2(887) } \end{aligned}$ | - Number of pulses <br> - Relative or absolute pulse specification <br> - Port <br> -"CW/CCW" or "Pulse + direction" <br> - Acceleration rate <br> - Deceleration rate <br> - Target frequency <br> - Starting frequency |
|  | Change the acceleration and deceleration rates during positioning (multiple start function) |  | PLS2(887) can be executed during positioning (acceleration or deceleration) to change the acceleration rate or deceleration rate. | PULS(886) <br> $\downarrow$ <br> ACC(888) <br> (Indepen- <br> dent) <br> $\downarrow$ <br> PLS2(887) <br> PLS2(887) <br> $\downarrow$ <br> PLS2(887) | - Number of pulses <br> - Acceleration rate <br> -Deceleration rate |


| Operation | Example application | Frequency changes |  |  | Description | Procedure |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  |  |  |  | Instruction | Settings |
| Change direction | Change the direction during positioning | Pulse freque <br> Target frequency |  | cified <br> mber of <br> ses <br> Change of direction at the specified deceleration rate <br> Execution of <br> PLS2(887) |  | PLS2(887) can be executed during positioning with relative pulse specification to change to absolute pulses and reverse direction. | $\begin{aligned} & \hline \text { PULS(886) } \\ & \downarrow \\ & \text { ACC(888) } \\ & \text { (Indepen- } \\ & \text { dent) } \\ & \downarrow \\ & \text { PLS2(887) } \\ & \hline \text { PLS2(887) } \\ & \downarrow \\ & \text { PLS2(887) } \end{aligned}$ | - Number of pulses <br> - Absolute pulse specification <br> - Port <br> -"CW/CCW" or "Pulse + direction" <br> - Acceleration rate <br> - Deceleration rate <br> - Target frequency <br> - Starting frequency |
| Change pulse output method | Not supported. |  |  |  |  |  |  |

Stopping a Pulse Output

| Operation | Example application | Frequency changes | Description | Procedure |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  |  |  | Instruction | Settings |
| Stop pulse output (Number of pulses setting is not preserved.) | Immediate stop |  | Stops the pulse output immediately and clears the number of output pulses setting. | PULS(886) <br> $\downarrow$ <br> ACC(888) or <br> SPED(885) <br> (Indepen- <br> dent) <br> $\downarrow$ <br> INI(880) <br> PLS2(887) <br> $\downarrow$ <br> INI(880) | - Stop pulse output |
| Stop pulse output (Number of pulses setting is not preserved.) | Immediate stop |  | Stops the pulse output immediately and clears the number of output pulses setting. | $\begin{aligned} & \text { PULS(886) } \\ & \downarrow \\ & \text { SPED(885) } \\ & \text { (Indepen- } \\ & \text { dent) } \\ & \downarrow \\ & \text { SPED(885) } \end{aligned}$ | - Port <br> - Independent <br> - Target frequency $=0$ |
| Stop sloped pulse output smoothly. (Number of pulses setting is not preserved.) | Decelerate to a stop | Pulse frequency | Decelerates the pulse output to a stop. <br> Note If ACC(888) <br> started the operation, the original acceleration/ deceleration rate will remain in effect. If SPED(885) started the operation, the acceleration/ deceleration rate will be invalid and the pulse output will stop immediately. | PULS(886) <br> ACC(888) or SPED(885) (Independent) <br> ACC(888) (Independent) <br> PLS2(887) <br> $\downarrow$ <br> ACC(888) (Independent) | - Port <br> - Independent <br> - Target frequency $=0$ |

Switching from Continuous Mode (Speed Control) to Independent Mode (Positioning)

| Example application | Frequency changes | Description | Procedure |  |
| :---: | :---: | :---: | :---: | :---: |
|  |  |  | Instruction | Settings |
| Change from speed control to fixed distance positioning during operation | Outputs the number of pulses specified in PLS2(887) (Both relative and absolute pulse specification can be used.) | PLS2(887) can be executed during a speed control operation started with ACC(888) to change to positioning operation. <br> Note An error will occur if a constant speed cannot be achieved after switching the mode. If this happens, the instruction | $\begin{aligned} & \text { ACC(888) } \\ & \text { (Continu- } \\ & \text { ous) } \\ & \downarrow \\ & \text { PLS2(887) } \end{aligned}$ | - Port <br> - Acceleration rate <br> - Deceleration rate <br> - Target frequency <br> - Number of pulses <br> Note The starting frequency is ignored. |
| Fixed distance feed interrupt | Pulse frequency Present frequency | be ignored and the previous operation will be continued. |  |  |

## Relative Pulse Outputs and Absolute Pulse Outputs

## Selecting Relative or Absolute Coordinates

The pulse output PV's coordinate system (absolute or relative) is selected automatically, as follows:

- When the origin is undetermined, the system operates in relative coordinates.
- When the origin has been determined, the system operates in absolute coordinates.

| Conditions | Origin has been <br> determined by an ori- <br> gin search | Origin has been <br> determined by exe- <br> cuting INI(880) to <br> change the PV | Origin not estab- <br> lished (Origin search <br> has not been per- <br> formed and PV has <br> not been changed <br> with INI(880).) |
| :--- | :--- | :--- | :--- |
| Pulse output <br> PV's coordi- <br> nate system | Absolute coordinates | Relative coordinates |  |

Relationship between the Coordinate System and Pulse Specification

The following table shows the pulse output operation for the four possible combinations of the coordinate systems (absolute or relative) and the pulse output (absolute or relative) specified when PULS(886) or PLS2(887) is executed.

| Pulse output specified in PULS(886) or PLS2(887) | Coordinate system |  |
| :---: | :---: | :---: |
|  | Relative coordinate system | Absolute coordinate syste |
|  | Origin not established: <br> The No-origin Flag will be ON in this | Origin established: The No-origin Flag will be OFF in this case. |
| Relative pulse specification | Positions the system to another position relative to the current position. Number of movement pulses = number of pulses setting |  |
|  | The pulse output PV after instruction execution = Number of movement pulses $=$ Number of pulses setting <br> Note The pulse output PV is reset to 0 just before pulses are output. After that, the specified number of pulses is output. <br> The following example shows the number of pulses setting = 100 counterclockwise. <br> Pulse output PV range: <br> 80000000 to 7FFF FFFF hex <br> Number of pulses setting range: <br> 00000000 to 7FFF FFFF hex | The pulse output PV after instruction execution = PV + Number of movement pulses. <br> The following example shows the number of pulses setting = 100 counterclockwise. <br> Pulse output PV range: 80000000 to 7FFF FFFF hex <br> Number of pulses setting range: 00000000 to 7FFF FFFF hex |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |


| Pulse output specified in PULS(886) or PLS2(887) | Coordinate system |  |
| :---: | :---: | :---: |
|  | Relative coordinate system | Absolute coordinate system |
|  | Origin not established: <br> The No-origin Flag will be ON in this case. | Origin established: <br> The No-origin Flag will be OFF in this case. |
| Absolute pulse specification | The absolute pulse specification cannot be used when the origin location is undetermined, i.e., when the system is operating in the relative coordinate system. An instruction execution error will occur. | Positions the system to an absolute position relative to the origin. |
|  |  | The number of movement pulses and movement direction are calculated automatically from the current position (pulse output PV) and target position. |
|  |  | The following example shows the number of pulses setting $=+100$. |
|  |  | Number of pulses setting II Number of movement pulses |
|  |  | Pulse |
|  |  |  |
|  |  | Number of movement pulses $=$ Number of pulses setting - Pulse output PV when instruction is executed |
|  |  | The movement direction is determined automatically. |
|  |  | Pulse output PV when instruction is executed $=$ Number of pulses setting |
|  |  | Pulse output PV range: <br> 80000000 to 7FFF FFFF hex |
|  |  | Number of pulses setting range: 80000000 to 7FFF FFFF hex |

## Operations Affecting the Origin Status (Established/Not Established Status)

The following table shows the operations that can affect the origin status (origin established or no-origin), such as changing the operating mode and executing certain instructions.
The No-origin Flag will be ON when the corresponding pulse output's origin is not established and OFF when the origin is established.

| Current status |  | PROGRAM mode |  | RUN mode or MONITOR <br> mode |  |
| :--- | :--- | :--- | :--- | :--- | :--- |
| Operation | Origin <br> established | Origin not <br> established | Origin <br> established | Origin not <br> established |  |
| Operat- <br> ing mode <br> change | Switch to <br> RUN or <br> MONITOR | Status <br> changes to <br> "Origin not <br> established." | "Origin not <br> established" <br> status contin- <br> ues. | --- | --- |
|  | Switch to <br> PROGRAM | --- | --- | "Origin <br> established" <br> status contin- <br> ues. | "Origin not <br> established" <br> status contin- <br> ues. |


| Current status |  | PROGRAM mode |  | RUN mode or MONITOR <br> mode |  |
| :--- | :--- | :--- | :--- | :--- | :--- |
| Operation | Origin <br> established | Origin not <br> established | Origin <br> established | Origin not <br> established |  |
| Instruc- <br> tion exe- <br> cution | Originsearch <br> performed by <br> ORG(889) | --- | --- | Status <br> changes to <br> "Origin <br> established." | Status <br> changes to <br> "Origin <br> established." |
|  |  |  | PV changed |  |  |
| by INI(880) | --- | "Origin <br> established" <br> status contin- <br> ues. | Status <br> changes to <br> "Origin <br> established." |  |  |
| The Pulse Output Reset <br> Bit (A54000 or A54100) <br> goes from OFF to ON. | Status <br> changes to <br> "Origin not <br> established." | "Origin not <br> established" <br> status contin- <br> ues. | Status <br> changes to <br> "Origin not <br> established." | "Origin not <br> established" <br> status contin- <br> ues. |  |

## Movement Direction when Using Absolute Pulse Specification

When operating with the absolute pulse specification, the movement direction is selected automatically based on the relationship between the pulse output PV when the instruction is executed and the specified target position. The direction (CW/CCW) specified in an ACC(888) or SPED(885) instruction is not effective.

## Using CW/CCW Limit Inputs for Pulse Output Functions Other than Origin Searches

Pulse outputs will stop when either the CW or CCW limit input signals turns ON. It is also possible to select whether or not the established origin will be cleared when a CW or CCW limit input signal turns ON for an origin search or other pulse output function.

## S-curve Acceleration/Deceleration

S-curve acceleration/deceleration can be used for pulse output instructions involving acceleration/deceleration. When there is leeway in the maximum allowable speed, S-curve accelerations/decelerations will help control shock and vibration by reducing the initial acceleration rate in comparison with linear acceleration/deceleration.

Note The setting for S-curve acceleration/deceleration applies to all pulse outputs.

## Output Pattern

The output pattern for S-curve acceleration/deceleration is shown below.

## Example for PLS2(887)



The same type of S-curve acceleration/deceleration can be used for ACC(888) as well.

Note The curve for S-curve acceleration/deceleration is formed by applying a cubic equation to the straight line of the set acceleration/deceleration rates (a cubic polynomial approximation). The curve's parameters cannot be changed. The maximum acceleration will be 1.5 times that of trapezoidal acceleration/ deceleration for the same acceleration/deceleration rate.

## Procedure

## Restrictions

Precautions when using the Pulse Output Function

Make the following settings in the PLC Setup.
Pulse Output 0 to 3

| Speed Curve | Trapezium | When a pulse output is executed with accelera- <br> tion/deceleration, this setting determines <br> whether the acceleration/deceleration rate is lin- <br> ear (trapezium) or S-shaped. |
| :--- | :--- | :--- |
|  | S-shaped | (tan |

The following restrictions apply when using S-curve acceleration/deceleration.

## Starting Frequency

The starting frequency must be 100 Hz or greater. If the starting frequency is set to less than 100 Hz , it will automatically be increased to 100 Hz if S-curve acceleration/deceleration is set.


## Target Frequency

S-curve acceleration/deceleration will not be performed if the target frequency is less than 100 Hz .


The CP1L CPU Unit's pulse output frequency is determined by dividing the source clock frequency by an integer ratio. (The source clock frequency for ports 0 and 1 is 20 MHz and the frequency for ports 2 and 3 is 16.4 MHz .) Consequently, there may be a slight difference between the set frequency and the actual frequency, and that difference increases as the frequency increases. The actual frequency can be calculated from the following equations.

## Pulse Output System



## Equations

Actual frequency $(\mathrm{Hz})=\frac{\text { Source clock frequency }}{\text { Dividing ratio }}$
Dividing ratio $=\operatorname{INT}\left(\frac{(\text { Clock frequency } \times 2)+\text { Set frequency }}{\text { Set frequency }(\mathrm{Hz}) \times 2}\right)$
The INT function extracts an integer from the fraction. The non-integer remainder is rounded.

## Differences between Set Frequencies and Actual Frequencies

- Source clock frequency: 16.4 MHz

| Set frequency (kHz) | Actual frequency (kHz) |
| :---: | :---: |
| 99.696 to 100.000 | 100.000 |
| 99.093 to 99.696 | 99.393 |
| 98.498 to 99.093 | 98.795 |
| $:$ | $:$ |
| 50.076 to 50.229 | 50.152 |
| 49.923 to 50.076 | 50.000 |
| 49.772 to 49.923 | 49.848 |
| $:$ | $:$ |
| 20.012 to 20.036 | 20.024 |
| 19.987 to 20.012 | 20.000 |
| 19.963 to 19.987 | 19.975 |
| $:$ | 10.006 |
| 10.003 to 10.009 | 10.000 |
| 9.996 to 10.003 | 9.993 |
| 9.990 to 9.996 | $:$ |
| $:$ | 5.001 |
| 5.000 to 5.002 | 5.000 |
| 4.999 to 5.000 | 4.998 |
| 4.997 to 4.999 | $:$ |
| $:$ | 3.001 |
| 3.001 to 3.001 | 2.999 |
| 3.000 to 3.000 |  |
| 2.998 to 2.999 |  |

## 5-2-5 Origin Search and Origin Return Functions

The CP1L CPU Units have two functions that can be used to determine the machine origin for positioning.

1,2,3... 1. Origin Search
The ORG instruction outputs pulses to turn the motor according to the pattern specified in the origin search parameters. As the motor turns, the origin search function determines the machine origin from the following 3 kinds of position input signals.

- Origin input signal
- Origin proximity input signal
- CW limit input signal and CCW limit input signal

2. Changing the Pulse Output PV

When you want to set the current position as the origin, execute $\mathrm{INI}(880)$ to reset the pulse output PV to 0 .
The origin location can be determined after using either method.
The CP1L CPU Units are also equipped with the origin return function, which can be executed to return the system to the origin after the origin location has been determined by one of the methods above.

- Origin Return

If the motor is stopped, ORG(889) can be executed to perform an origin return operation that moves the motor back to the origin position. The origin position must be determined in advance by performing an origin search or changing the pulse output PV.

Note The motor can be moved even if the origin position has not been determined, but positioning operations will be limited as follows:

- Origin return: Cannot be used.
- Positioning with absolute pulse specification: Cannot be used.
- Positioning with relative pulse specification: Outputs the specified number of pulses after setting the current position to 0 .


## 5-2-5-1 Origin Search

When ORG(889) executes an origin search, it outputs pulses to actually move the motor and determines the origin position using the input signals that indicate the origin proximity and origin positions.
The input signals that indicate the origin position can be received from the servomotor's built-in phase-Z signal or external sensors such as photoelectric sensors, proximity sensors, or limit switches.
Several origin search patterns can be selected.
In the following example, the motor is started at a specified speed, accelerated to the origin search high speed, and run at that speed until the origin proximity position is detected. After the Origin Proximity Input is detected, the motor is decelerated to the origin search low speed and run at that speed until the origin position is detected. The motor is stopped at the origin position.


## Procedure

| Wire the pulse output |
| :---: |
| and input signals. |
| PLC Setup settings |
| $\downarrow$ |

- Output: Connect the outputs using the CW/CCW method or pulse + direction method. The same method must be used for all of the pulse outputs. Power supply for outputs: 24 V DC
- Inputs: Connect the Origin input Signal, Near Origin Input Signal, and Positioning Complete Signal to the built-in input terminals allocated to the pulse output being used.
The limit inputs must be connected to available normal input terminals or terminals and output from the ladder program.
- Enable the origin search function for pulse output 0 to 3 by setting the Origin Search Function Enable/Disable setting to 1.
- Limit Input Signal Settings

Limit Input Signal Operation and Undefine Origin Settings

- Acceleration/Deceleration Curve Setting
- Other Parameter Settings

1. Operation Mode

- Set the best operation mode for the driver being used (servomotor or stepping motor.)
- Set "mode 0" when driving a stepping motor. Set "mode 1" or "mode 2" when driving a servomotor.

2. Set the origin search operation setting.
3. Set the origin detection method.
4. Set the origin search direction (CW or CCW.)
5. Set the origin search speeds:

Initial speed for origin search/origin return, origin search high speed, origin search proximity speed, origin search acceleration rate, and origin search deceleration rate
6. Origin Compensation

After the origin has been determined, the origin compensation can
be set to compensate for a shift in the Proximity Sensor's ON
position, motor replacement, or other change.
7. Set the Origin Proximity Input Signal type, Origin Input Signal
type, and Limit Input Signal type.
8. Set the Positioning Monitor Time.

- Output the status of the Limit Signal Inputs and Positioning Completed Signal to Auxiliary Area bits.
- Execute ORG(889).

Specify the origin search operation by setting the third operand to 0000.

## Restrictions

- The Phase-Z signal + Software reset method cannot be used for a highspeed counter when the origin search function has been enabled in the PLC Setup.


## PLC Setup

## Origin Search Function Enable/Disable Settings

These PLC Setup indicate whether or not the origin search function will be used for each pulse output.

## ■ Limit Input Signal Setting

Specify in the following PLC Setup whether to use the CW/CCW limit input signals only for origin searches or for all pulse output functions. These settings affect all pulse outputs.
(This setting is called the Limited Input Signal Operation setting.)

## - Pulse Output 0 Undefined Origin Setting

## - Acceleration/Deceleration Curve Settings

Note The acceleration/deceleration curve setting applies to all pulse outputs, not just to origin searches. Refer to S-curve Acceleration/Deceleration on page 217 for details.

## Origin Search Parameters

The various origin search parameters are set in the PLC Setup.

| Name | Settings | Time when <br> read |
| :--- | :--- | :--- |
| Operating mode | Operating mode 0, 1, or 2 | Start of <br> operation |
| Origin search operation <br> setting | 0: Reversal mode 1 <br> 1: Reversal mode 2 | Start of <br> operation |
| Origin detection method | 0: Read the Origin Input Signal after the <br> Origin Proximity Input Signal goes <br> from OFF $\rightarrow$ ON $\rightarrow$ OFF. <br> 1: Read the Origin Input Signal after the <br> Origin Proximity Input Signal goes <br> from OFF $\rightarrow$ ON. | Start of <br> operation |
| 2: Just read the Origin Input Signal with- |  |  |
| out using the Origin Proximity Input |  |  |
| Signal. |  |  |\(\quad\left\{\begin{array}{l}Origin search direction <br>

0: CW direction <br>
1: CCW direction\end{array}\right.\)

Note An origin search will not be started unless the origin search proximity speed is less than the origin search high speed and unless the origin search/return initial speed is less than the origin search proximity speed.

## Explanation of the Origin Search Parameters

Operating Mode
The operating mode parameter specifies the kind of I/O signals that are used in the origin search. The 3 operating modes indicate whether the Error Counter Reset Output and Positioning Completed Input are used.

| Operating mode | I/O signal |  |  | Remarks |
| :---: | :---: | :---: | :---: | :---: |
|  | Origin Input Signal | Error Counter Reset Output | Positioning Completed Input | Operation when the origin is detected during deceleration from the origin search's high speed |
| 0 | The origin position is determined when the Origin Input Signal goes from OFF to ON. | Not used. <br> The origin search operation ends after the origin is detected. | Not used. | The Origin Input Signal will be detected during deceleration. An Origin Input Signal Error (error code 0202) will occur and the motor will decelerate to a stop. |
| 1 |  | Goes ON for 20 to 30 ms when the origin is detected. |  | The Origin Input Signal will not be detected during deceleration. When the Origin Input Signal is detected after the motor has reached the proximity speed for origin search, the motor will be stopped and the origin search operation will end. |
| 2 |  |  | After the origin is detected, the origin search will not be end until the Positioning Completed Input is received from the driver. |  |

The following table shows the proper operating mode settings for different drivers and applications.

| Driver | Remarks | Operating mode |
| :--- | :--- | :--- |
| Stepping motor driver (See note.) | 0 |  |
| Servo driver | Use this mode when you want to <br> reduce the processing time, even at the <br> expense of positioning accuracy. (The <br> Servo Driver's positioning complete <br> signal is not used.) | 1 |
| Use this mode when you want high <br> positioning accuracy. (The Servo <br> Driver's positioning complete signal is <br> used.) | 2 |  |

Note There are stepping motor drivers that are equipped with a positioning completed signal like a Servo driver. Operating modes 1 and 2 can be used with these stepping motor drivers.
■ Remarks: Operations Detecting the Origin During Deceleration from High Speed

## Operating Mode 0 (without Error Counter Reset Output, without Positioning Completed Input)

Connect the sensor's open collector output signal to the Origin Input Signal. The Origin Input Signal's response time is 0.1 ms when set as a NO contact.
When the Origin Proximity Input Signal is received, the motor will begin decelerating from the origin search high speed to the origin search proximity speed. In this operating mode, the Origin Input Signal will be detected if it is received during this deceleration and an Origin Input Signal Error (error code 0202) will be generated. In this case, the motor will decelerate to a stop.


Operating Mode 1 (with Error Counter Reset Output, without Positioning Completed Input)
Connect the phase-Z signal from the Servo Driver to the Origin Input Signal.
When the Origin Input Signal is received, the pulse output will be stopped and the Error Counter Reset Signal will be output for about 20 to 30 ms .


When the Origin Proximity Input Signal is received, the motor will begin decelerating from the origin search high speed to the origin search proximity speed. In this operating mode, the motor will stop at the Origin Input Signal after deceleration is completed.

## Operating Mode 1 with Origin Proximity Input Signal Reverse (Origin Detection Method Setting = 0)

When the deceleration time is short, the Origin Input Signal can be detected immediately after the Origin Proximity Input Signal goes from ON to OFF. Set
a Origin Proximity Input Signal dog setting that is long enough (longer than the deceleration time.)


Note: The Origin Input Signal can be detected immediately after the Origin Proximity Input Signal goes from ON to OFF if the deceleration time is short, e.g., starting from within the Origin Proximity Input Signal.

## Operating Mode 1 without Origin Proximity Input Signal Reverse (Origin Detection Method Setting = 1)

Depending on the length of the deceleration time, the stopping position may change when the Origin Input Signal is detected during deceleration.


## Operating Mode 2 (with Error Counter Reset Output, with Positioning Completed Input)

This operating mode is the same as mode 1, except the Positioning Completed Signal (INP) from the Servo Driver is used. Connect the Positioning Completed Signal from the Servo Driver to a normal input (origin search 0 to 3 input).

Error Counter
Reset Output


Origin Search Operation Setting

Origin Detection Method

If origin compensation is not being applied, the Positioning Completed Signal is checked after the Error Counter Reset Output. If origin compensation is being applied, the Positioning Completed Signal is checked after the compensation operation is completed.


Select either of the following two reverse modes for the origin search operation pattern.

| Setting | Description |
| :---: | :--- |
| 0: Reversal mode 1 | When the limit input signal is received in the origin search <br> direction, reverse and continue operation. |
| 1: Reversal mode 2 | When the limit input signal is received in the origin search <br> direction, generate an error and stop operation. |

The origin detection method depends on the Origin Proximity Input Signal settings. Select one of the following three methods in each port's parameters.

| Setting | Description |
| :--- | :--- |
| 0: Origin Proximity Input Signal <br> reversal required. | Reads the first Origin Input Signal after the Ori- <br> gin Proximity Input Signal goes <br> OFF $\rightarrow$ ON $\rightarrow$ OFF. |
| 1: Origin Proximity Input Signal <br> reversal not required. | Reads the first Origin Input Signal after the Ori- <br> gin Proximity Input Signal goes OFF $\rightarrow$ ON. |
| 2: Origin Proximity Input Signal not <br> used. | Just read the Origin Input Signal without using <br> the Origin Proximity Input Signal. |

## Origin Detection Method 0: Origin Proximity Input Signal Reversal Required



## Origin Detection Method 1: Origin Proximity Input Signal Reversal Not Required



Origin Detection Method 2: Origin Proximity Input Signal Reversal Not Used

Origin Search Operating Mode and Origin Detection Method Settings

The following examples explain how the operation patterns are affected by the origin search operation and origin detection method settings.
These examples have a CW origin search direction. (The search direction and limit input signal direction would be different for an origin search in the CCW direction.)

## Using Reversal Mode 1

| Origin search operation <br> Origin detection method | $0:$ Reversal mode 1 |
| :---: | :---: |
| 0: Origin Proximity Input Signal reversal required. | Note When the limit input signal is received, the motor stops without deceleration, reverses direction, and accelerates. |
| 1: Origin Proximity Input Signal reversal not required. | Note When the limit input signal is received, the motor stops without deceleration, reverses direction, and accelerates. |
| 2: Origin Proximity Input Signal not used. | Note When the direction of operation is reversed, it is reversed immediately without deceleration or acceleration. |

## Using Reversal Mode 2

| $\left.$\begin{tabular}{\|l|}
\hline
\end{tabular}Origin search <br> operation\right\rvert\,Origin detection <br> method | 1: Reversal mode 2 |
| :---: | :---: |
| 0: Origin Proximity Input Signal reversal required. | Note When the limit input signal is received, the motor stops without deceleration. |
| 1: Origin Proximity Input Signal reversal not required. |  <br> Note When the limit input signal is received, the motor stops without deceleration. |


| Origin search operation <br> Origin detection method | 1: Reversal mode 2 |
| :---: | :---: |
| 2: Origin Proximity Input Signal not used. |  <br> Note When the limit input signal is received, the motor stops without deceleration. |

## Specifying the Origin Search Direction (CW or CCW Direction)

## Origin Search Speed

Note

Sets the direction to move when detecting the Origin Input Signal.
Typically, the origin search is performed so that the Origin Input Signal's rising edge is detected when moving in the origin search direction.

| Setting | Description |
| :--- | :--- |
| 0 | CW direction |
| 1 | CCW direction |

These are the motor speed settings used in the origin search.
The origin search will not be performed in these cases:
Origin search high speed $\leq$ Origin search proximity speed
Origin search proximity speed $\leq$ Origin search initial speed

## Origin Search/Return Initial Speed

Sets the motor's starting speed when the origin search is executed. Specify the speed in the number of pulses per second (pps).

## Origin Search High Speed

Sets the motor's target speed when the origin search is executed. Specify the speed in the number of pulses per second (pps).

## Origin Search Proximity Speed

Sets the motor's speed after the Origin Proximity Input Signal is detected. Specify the speed in the number of pulses per second (pps).

## Origin Search Acceleration Rate

Sets the motor's acceleration rate when the origin search is executed. Specify the amount to increase the speed (Hz) per 4-ms interval.

## Origin Search Deceleration Rate

Sets the motor's acceleration rate when the origin search function is decelerating. Specify the amount to decrease the speed (Hz) per 4-ms interval.
Origin Compensation

After the origin has been determined, the origin compensation can be set to compensate for a shift in the Proximity Sensor's ON position, motor replacement, or other change.

## I/O Settings

Positioning Monitor Time
Once the origin has been detected in an origin search, the number of pulses specified in the origin compensation is output, the current position is reset to 0 , and the pulse output's No-origin Flag is turned OFF.
Setting range: 80000000 to 7FFF FFFF hex ( $-2,147,483,648$ to $2,147,483,647$ ) pulses

## Limit Input Signal Type (NC/NO)

Specifies the type of input signal (normally closed or normally open) being used for the limit inputs.
0 : NC
1: NO

## Origin Proximity Input Signal Type (NC/NO)

Specifies the type of input signal (normally closed or normally open) being used for the Origin Proximity Input Signal.
0 : NC
1: NO

## Origin Input Signal Type (NC/NO)

Specifies the type of input signal (normally closed or normally open) being used for the Origin Input Signal.
0 : NC
1: NO
When the operating mode is set to mode 2, this setting specifies how long to wait (in ms ) for the Positioning Completed Signal after the positioning operation has been completed, i.e., the pulse output has been completed. A Positioning Timeout Error (error code 0300) will be generated if the motor driver's Positioning Completed Signal does not come ON within the specified time.
Setting range: 0000 to 270F hex ( 0 to $9,999 \mathrm{~ms}$ )
The actual monitoring time will be the Positioning Monitor Time rounded up to the nearest $10-\mathrm{ms}$ unit +10 ms max.
If the Positioning Monitor Time is set to 0 , the function will be disabled and the Unit will continue waiting for the Positioning Completed Signal to come ON. (A Positioning Timeout Error will not be generated.)

## Executing an Origin Search

Execute ORG(889) in the ladder program to perform an origin search with the specified parameters.


P: Port specifier
Pulse output 0: \#0000
Pulse output 1: \#0001
C: Control data; Origin search and CW/CCW method: \#0000
Origin search and pulse + direction method: \#0001

## Restrictions

The motor can be moved even if the origin position has not been determined with the origin search function, but positioning operations will be limited as follows:

| Function |  | Operation |
| :--- | :--- | :--- |
| Origin return | Cannot be used. |  |


| Function | Operation |
| :--- | :--- |
| Positioning with absolute <br> pulse specification | Cannot be used. |
| Positioning with relative <br> pulse specification | Outputs the specified number of pulses after setting the <br> current position to 0. |

An origin search will not be started unless the origin search proximity speed is less than the origin search high speed and unless the origin search/return initial speed is less than the origin search proximity speed.

## Origin Search Error Processing

The CP1L CPU Unit's pulse output function performs a basic error check before starting to output pulses (when the instruction is executed) and will not output pulses if the settings are incorrect. There are other errors that can occur with the origin search function during pulse output, which may stop the pulse output.
If an error occurs that stops pulse output, the pulse output's Output Stopped Error Flag will be turned ON and the Pulse Output Stop Error Code will be written to Error Code word. Use these flags and error codes to identify the cause of the error.

The Pulse Output Stop Errors will not affect the CPU Unit's operating status. (The Pulse Output Stop Errors do not cause a fatal or non-fatal error in the CPU Unit.)

Related Auxiliary Area Flags

| Function | Pulse output number |  |  |
| :--- | :--- | :--- | :--- |
|  | $\mathbf{0}$ | $\mathbf{1}$ |  |
| Output Stopped Error Flags <br> ON when an error occurred while outputting pulses in the ori- <br> gin search function. | 0: No error <br> $1:$ Stop error occurred. | A280.07 | A281.07 |
| Stop Error Codes <br> When a Pulse Output Stop Error occurs, the error code is stored in that pulse outputs corre- <br> sponding Stop Error Code word. | A444 | A445 |  |

## Pulse Output Stop Error Codes

| Error name | Error code | Likely cause | Corrective action | Operation after <br> error |
| :--- | :--- | :--- | :--- | :--- |
| CW Limit Stop Input <br> Signal | 0100 | Stopped due to a CW limit signal <br> input. | Move in the CCW direction. | Immediate stop, <br> No effect on <br> other port |
| CCW Limit Stop <br> Input Signal | 0101 | Stopped due to a CCW limit sig- <br> nal input. | Move in the CW direction. |  |
| No Origin Proximity <br> Input Signal | 0200 | The parameters indicate that the <br> Origin Proximity Input Signal is <br> being used, but a Origin Proxim- <br> ity Input Signal was not received <br> during the origin search. | Check the wiring of the Origin <br> Proximity Input Signal as well as <br> the PLC Setup's Origin Proxim- <br> ity Input Signal Type setting (NC <br> or NO) and execute the origin <br> search again. Turn the power <br> supply OFF and then ON if the <br> signal type setting was changed. | No effect on <br> other port |
| No Origin Input Sig- <br> nal | 0201 | The Origin Input Signal was not <br> received during the origin <br> search. | Check the wiring of the Origin <br> Input Signal as well as the PLC <br> Setup's Origin Input Signal Type <br> setting (NC or NO) and execute <br> the origin search again. Turn the <br> power supply OFF and then ON <br> if the signal type setting was <br> changed. |  |


| Error name | Error code | Likely cause | Corrective action | Operation after error |
| :---: | :---: | :---: | :---: | :---: |
| Origin Input Signal Error | 0202 | During an origin search in operating mode 0, the Origin Input Signal was received during the deceleration started after the Origin Proximity Input Signal was received. | Take one or both of the following steps so that the Origin Input Signal is received after deceleration is completed. <br> - Increase the distance between the Origin Proximity Input Signal sensor and Origin Input Signal sensor. <br> -Decrease the difference between the origin search's high speed and proximity speed settings. | Decelerates to a stop, <br> No effect on other port |
| Limit Inputs in Both Directions | 0203 | The origin search cannot be performed because the limit signals for both directions are being input simultaneously. | Check the wiring of the limit signals in both directions as well as the PLC Setup's Limit Signal Type setting (NC or NO) and execute the origin search again. Turn the power supply OFF and then ON if the signal type setting was changed. | Operation will not start. <br> No effect on other port |
| Simultaneous Origin Proximity and Limit Inputs | 0204 | The Origin Proximity Input Signal and the Limit Input Signal in the search direction are being input simultaneously during an origin search. | Check the wiring of the Origin Proximity Input Signal and the Limit Input Signal. Also check the PLC Setup's Origin Proximity Input Signal Type and Limit Signal Type settings (NC or NO) and then execute the origin search again. Turn the power supply OFF and then ON if a signal type setting was changed. | Immediate stop, No effect on other port |
| Limit Input Signal Already Being Input | 0205 | -When an origin search in one direction is being performed, the Limit Input Signal is already being input in the origin search direction. <br> -When a non-regional origin search is being performed, the Origin Input Signal and the Limit Input Signal in the opposite direction (from the search direction) are being input simultaneously. | Check the wiring of the Limit Input Signal and the PLC Setup's I/O settings. Also check the PLC Setup's Limit Signal Type setting (NC or NO) and then execute the origin search again. Turn the power supply OFF and then ON if the signal type setting was changed. | Immediate stop, No effect on other port |
| Origin Proximity Input Signal Origin Reverse Error | 0206 | -When an origin search with reversal at the limit is being performed, the Limit Input Signal in the search direction was input while the Origin Proximity Input Signal was reversing. <br> -When an origin search with reversal at the limit is being performed and the Origin Proximity Input Signal is not being used, the Limit Input Signal in the search direction was input while the Origin Input Signal was reversing. | Check the installation positions of the Origin Proximity Input Signal, Origin Input Signal, and Limit Input Signal as well as the PLC Setup's I/O settings. Also check the PLC Setup's Signal Type settings (NC or NO) for each input signal and then execute the origin search again. Turn the power supply OFF and then ON if a signal type setting was changed. | Immediate stop, <br> No effect on other port |
| Positioning Timeout Error | 0300 | The Servo Driver's Positioning Completed Signal does not come ON within the Positioning Monitor Time specified in the PLC Setup. | Adjust the Positioning Monitor Time setting or Servo system gain setting. Check the Positioning Completed Signal wiring, correct it if necessary, and then execute the origin search again. | Decelerates to a stop, <br> No effect on other port |

## Origin Search Examples

## Operation

Connect a Servo Driver and execute an origin search based on the Servomotor's built-in encoder phase-Z signal and a Origin Proximity Input Signal.

## Conditions

- Operating mode: 1
(Uses the Servomotor encoder's phase-Z signal as the Origin Input Signal.)
- Origin search operation setting: 0
(Sets reverse mode 1. Reverses direction when the limit input signal is input in the origin search direction.)
- Origin detection method: 0
(Reads the Origin Input Signal after the Origin Input Signal goes OFF $\rightarrow$ ON $\rightarrow$ OFF.)
- Origin search direction: 0 (CW direction)


## System Configuration



[^0]ORG(889)

## I/O Allocations

(Example: CP1L-M40/30
DT $\square$-D, CP1L-L20D $\square$-D
Units)
$\square$ Inputs

| Input terminal |  |  |
| :---: | :--- | :--- |
| Word | Bit |  |
| CIO 0 | 00 | CW limit detection sensor |
|  | 01 | CCW limit detection sensor |
|  | 06 | Pulse Output 0 Origin Input Signal |
|  | 10 | Pulse Output 0 Origin Proximity Input Signal |


| Word | Bit |  |
| :---: | :--- | :--- |
| A540 | 08 | Pulse Output 0 CW Limit Input Signal |
|  | 09 | Pulse Output 0 CCW Limit Input Signal |

Outputs

| Output terminal |  |  |
| :---: | :--- | :--- |
| Word | Bit |  |
| CIO 100 | 00 | Pulse Output 0 CW output |
|  | 01 | Pulse Output 0 CCW output |

## Operation



## PLC Setup

| Function |  |
| :--- | :--- |
| Pulse Output 0 Origin Search Function Enable/Disable | 1 hex: Enabled |
| Pulse Output 0 Origin Search Operating Mode | 1 hex: Mode 1 |
| Pulse Output 0 Origin Search Operation Setting | 0 hex: Reverse mode 1 |
| Pulse Output 0 Origin Detection Method | 0 hex: Origin detection method 0 |
| Pulse Output 0 Origin Search Direction Setting | 0 hex: CW direction |
| Pulse Output 0 Origin Search/Return Initial Speed | 0064 hex (100 pps) |
|  | 0000 hex |
| Pulse Output 0 Origin Search High Speed | 07 DO hex (2,000 pps) |
|  | 0000 hex |


| Function | Setting (example) |
| :--- | :--- |
| Pulse Output 0 Origin Search Proximity Speed | $03 \mathrm{E} 8 \mathrm{hex}(1,000 \mathrm{pps})$ |
|  | 0000 hex |
| Pulse Output 0 Origin Compensation | 0000 hex |
|  | 0000 hex |
| Pulse Output 0 Origin Search Acceleration Rate | $0032 \mathrm{hex}(50 \mathrm{~Hz} / 4 \mathrm{~ms})$ |
| Pulse Output 0 Origin Search Deceleration Rate | $0032 \mathrm{hex}(50 \mathrm{~Hz} / 4 \mathrm{~ms})$ |
| Pulse Output 0 Limit Input Signal Type | $1: \mathrm{NO}$ |
| Pulse Output 0 Origin Proximity Input Signal Type | $1: \mathrm{NO}$ |
| Pulse Output 0 Origin Input Signal Type | $1: \mathrm{NO}$ |

## Ladder Program



## 5-2-6 Origin Return

## Overview

Moves the motor to the origin position from any other position. The origin return operation is controlled by ORG(889).
The origin return operation returns the motor to the origin by starting at the specified speed, accelerating to the target speed, moving at the target speed, and then decelerating to a stop at the origin position.


## Procedure



## PLC Setup

The various origin return parameters are set in the PLC Setup.

## Origin Return Parameters

| Name | Settings | Remarks |
| :---: | :---: | :---: |
| Origin search/return initial speed | 00000000 to 000186A0 hex ( 0 Hz to 100 kHz ) | Start of operation |
| Origin return target speed | 00000001 to 000186A0 hex ( 1 Hz to 100 kHz ) |  |
| Origin return acceleration rate | 0001 to FFFF hex ( 1 to $65,535 \mathrm{~Hz} / 4 \mathrm{~ms}$ ) |  |
| Origin return deceleration rate | 0001 to FFFF hex $(1$ to $65,535 \mathrm{~Hz} / 4 \mathrm{~ms})$ |  |

## Explanation of the Origin Return Parameters

Origin Search/Return Initial Speed

Origin Return Target Speed

Origin Return
Acceleration Rate
Origin Return
Deceleration Rate

Sets the motor's starting speed when the origin return is executed. Specify the speed in the number of pulses per second (pps).

Sets the motor's target speed when the origin return is executed. Specify the speed in the number of pulses per second (pps).

Sets the motor's acceleration rate when the origin return operation starts. Specify the amount to increase the speed (Hz) per 4-ms interval.

Sets the motor's acceleration rate when the origin return function is decelerating. Specify the amount to decrease the speed (Hz) per 4-ms interval.

## Executing an Origin Return



P: Port specifier (Pulse output 0: \#0000, Pulse output 1: \#0001)
Pulse output 0: \#0000
Pulse output 1: \#0001
C: Control data
(Origin return and CW/CCW method: \#1000, Origin search and pulse + direction method: \#1100)

Note An instruction execution error will occur if the origin is not determined (relative coordinate system) when $\operatorname{ORG}(889)$ is executed to perform an origin return operation.

## 5-2-7 Pulse Output Procedures

## Single-phase Pulse Output without Acceleration/Deceleration

The number of output pulses setting cannot be changed during positioning.

- PULS(886) and SPED(885)



## Single-phase Pulse Output with Acceleration/Deceleration

- PULS(886) and ACC(888)

- Pulse output method
- CW/CCW inputs
- Pulse + direction inputs
- Output frequency: 1 Hz to 100 kHz (1 Hz units)
- Enable/disable the origin search function. Set the various origin search parameters if the origin search function is enabled.
- PULS(886): Specify port number and set the number of output pulses.
- ACC(888): Specify port number and set the output method (CW/CCW method or Pulse + direction method) and pulse output control with acceleration/deceleration (the same rate is used for both acceleration and deceleration.)
- $\mathrm{INI}(880)$ : Specify port number and stop pulse output when necessary.
- PRV(881): Specify port number and read pulse output PV when necessary.


## Pulse Output with Trapezoidal Acceleration/Deceleration (Using PLS2(887))



## 5-2-8 Instructions Used for Pulse Outputs

The pulse output functions can be used by executing the pulse control instructions in the ladder program. For some instructions, the PLC Setup must be set in advance. The following instructions can be combined for positioning and speed control.

## Supported Pulse Instructions

The following table shows the kinds of pulse outputs controlled by each instruction.

| Instruction | Function | Positioning (independent mode) |  |  | Speed control (continuous mode) |  | Origin search |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Pulse output without acceleration/ deceleration | Pulse output with acceleration/deceleration |  | Pulse output without acceleration/ deceleration | Pulse output with acceleration/ deceleration |  |
|  |  |  | Trapezoidal, equal acceleration/ deceleration rates | Trapezoi- dal, sepa- rate accelera- tion/ decel- eration rates |  |  |  |
| PULS(886) SET PULSES | Sets the number of pulses to be output. | Used | --- | --- | --- | --- | --- |
| SPED(885) <br> SPEED OUTPUT | Performs pulse output control without acceleration or deceleration. <br> (When positioning, the number of pulses must be set in advance with PULS(886).) | Used | --- | --- | Used | --- | --- |
| ACC(888) <br> ACCELERATION CONTROL | Performs pulse output control with acceleration and deceleration. <br> (When positioning, the number of pulses must be set in advance with PULS(886).) | --- | Used | --- | --- | Used | --- |
| PLS2(887) PULSE OUTPUT | Performs pulse output control with independent acceleration and deceleration rates. <br> (Also sets the number of pulses.) | --- | --- | Used | --- | --- | --- |
| ORG(889) ORIGIN SEARCH | Actually moves the motor with pulse outputs and determines the machine origin based on the Origin Proximity Input and Origin Input signals | --- | --- | --- | --- | --- | Used |
| INI(880) <br> MODE CONTROL | Stops the pulse output. Changes the pulse output PV. (This operation determines the origin location.) | Used | Used | Used | Used | Used | --- |
| PRV(881) <br> HIGH-SPEED COUNTER PV READ | Reads the pulse output PV. | Used | Used | Used | Used | Used | --- |
| PWM(891) <br> PULSE WITH VARIABLE DUTY FACTOR | Performs pulse output control with variable duty factor pulse output. | --- | --- | --- | --- | --- | --- |

SET PULSES: PULS(886)
PULS(886) is used to set the pulse output amount (number of output pulses) for pulse outputs that are started later in the program using SPED(885) or ACC(888) in independent mode.

| PULS(886) |  |
| :---: | :---: |
| P | P: Port specifier |
| T | T: Pulse type |
| N | N: Number of pulses |


| Operand |  | Contents |
| :--- | :--- | :--- |
| P | Port specifier | 0000 hex: Pulse output 0 <br> 0001 hex: Pulse output 1 |
| T | Pulse type | 0000 hex: Relative pulse output <br> 0001 hex: Absolute pulse output |
| N | First number <br> of pulses word | N and N+1 contain the number of pulses setting. (N contains <br> the rightmost 4 digits and N+1 contains the leftmost 4 digits.) <br> Relative pulse output: <br> 0000 0000 to 7FFF FFFF hex (0 to 2,147,483,647) |

## SPEED OUTPUT: <br> SPED(885)

SPED(885) can be used to perform pulse output without acceleration or deceleration. Either independent mode positioning or continuous mode speed control is possible. For independent mode positioning, the number of pulses is set using PULS(886).
SPED(885) can also be executed during pulse output to change the output frequency, creating stepwise changes in the speed.

| $\operatorname{SPED}(885)$ |
| :---: |
| P |
| T |
| F |

P: Port specifier
T: Output mode
F: First pulse frequency word

| Operand |  |  | Contents |
| :---: | :---: | :---: | :---: |
| P | Port specifier |  | 0000 hex: Pulse output 0 0001 hex: Pulse output 1 |
| T | Output mode | Bits 0 to 3 | Mode <br> 0 hex: Continuous <br> 1 hex: Independent |
|  |  | Bits 4 to 7 | Direction <br> 0 hex: CW <br> 1 hex: CCW |
|  |  | Bits 8 to 11 | Pulse output method (See note.) <br> 0 hex: CW/CCW <br> 1 hex: Pulse + direction |
|  |  | Bits 12 to 15 | Not used. (Always 0 hex.) |
| F | First pul word | se frequency | $F$ and $F+1$ contain the pulse frequency setting, in units of 1 Hz . (F contains the rightmost 4 digits and $F+1$ contains the leftmost 4 digits.) 00000000 to 000186A0 hex ( 0 Hz to 100 kHz ) |

ACCELERATION CONTROL: ACC(888)

PULSE OUTPUT: PLS2(887)

Use ACC(888) to set the target frequency and acceleration and deceleration rate and output pulses with acceleration and deceleration. (Acceleration rate is the same as the deceleration rate.)
Either independent mode positioning or constant mode speed control is possible when used in combination with PULS(886). ACC(888) can also be executed during pulse output to change the target frequency or acceleration/ deceleration rate, enabling smooth (sloped) speed changes.

| $A C C(888)$ |
| :---: |
| $P$ |
| $M$ |
| $S$ |

P: Port specifier
M: Output mode
S: First word of settings table

| Operand |  |  | Contents |
| :---: | :---: | :---: | :---: |
| P | Port specifier |  | 0000 hex: Pulse output 0 0001 hex: Pulse output 1 |
| M | Output mode | Bits 0 to 3 | Mode <br> 0 hex: Continuous <br> 1 hex: Independent |
|  |  | Bits 4 to 7 | Direction <br> 0 hex: CW <br> 1 hex: CCW |
|  |  | Bits 8 to 11 | Pulse output method (See note.) <br> 0 hex: CW/CCW <br> 1 hex: Pulse + direction |
|  |  | Bits 12 to 15 | Not used. (Always 0 hex.) |
| S | First settings table word | S | Acceleration/deceleration rate: <br> 0001 to FFFF hex ( 1 to $65,535 \mathrm{~Hz}$ ) <br> Specify the increase or decrease in the frequency per pulse control period (4 ms). |
|  |  | $\begin{aligned} & \text { S+1 and } \\ & \text { S+2 } \end{aligned}$ | S and S+1 contain the target frequency setting, in units of 1 Hz . (S+1 contains the rightmost 4 digits and $\mathrm{S}+2$ contains the leftmost 4 digits.) 00000000 to 000186A0 hex ( 0 Hz to 100 kHz ) |

Use PLS2(887) to set the startup frequency, acceleration rate, and deceleration rate, and output a specified number of pulses. Only independent mode positioning is supported.
PLS2(887) can also be executed during pulse output to change the number of output pulses, target frequency, acceleration rate, or deceleration rate.


P: Port specifier
M: Output mode
S: First word of settings table
F: First word of starting frequency

| Operand |  | Contents |
| :--- | :--- | :--- |
| P | Port specifier | 0000 hex: Pulse output 0 <br> 0001 hex: Pulse output 1 |


| Operand |  |  | Contents |
| :---: | :---: | :---: | :---: |
| M | Output mode | Bits 0 to 3 | Mode <br> 0000 hex: Relative pulse output <br> 0001 hex: Absolute pulse output |
|  |  | Bits 4 to 7 | Direction <br> 0 hex: CW <br> 1 hex: CCW |
|  |  | Bits 8 to 11 | Pulse output method (See note.) <br> 0 hex: CW/CCW <br> 1 hex: Pulse + direction |
|  |  | Bits 12 to 15 | Not used. (Always 0 hex.) |
| S | First settings table word | S | Acceleration rate: <br> 0001 to FFFF hex ( 1 to $65,535 \mathrm{~Hz}$ ) <br> Specify the increase or decrease in the frequency per pulse control period (4 ms). |
|  |  | S+1 | Deceleration rate: <br> 0001 to FFFF hex ( 1 to $65,535 \mathrm{~Hz}$ ) <br> Specify the increase or decrease in the frequency per pulse control period ( 4 ms ). |
|  |  | $\begin{aligned} & \text { S+2 and } \\ & \text { S+3 } \end{aligned}$ | S+2 and S+3 contain the target frequency setting, in units of 1 Hz . ( $\mathrm{S}+2$ contains the rightmost 4 digits and $\mathrm{S}+3$ contains the leftmost 4 digits.) 00000001 to 000186A0 hex ( 0 Hz to 100 kHz ) |
|  |  | $\begin{aligned} & \text { S+4 and } \\ & \text { S+5 } \end{aligned}$ | S+4 and S+5 contain the number of pulses setting. (S+4 contains the rightmost 4 digits and $\mathrm{S}+5$ contains the leftmost 4 digits.) <br> Relative pulse output: 00000000 to 7FFF FFFF hex (0 to $2,147,483,647$ ) <br> Absolute pulse output: <br> 80000000 to 7FFF FFFF hex ( $-2,147,483,648$ to <br> 2,147,483,647) |
| F | First sta quency | ting freword | $F$ and $F+1$ contain the starting frequency setting, in units of 1 Hz . ( $F$ contains the rightmost 4 digits and $F+1$ contains the leftmost 4 digits.) $00000000 \text { to 000186A0 hex ( } 0 \mathrm{~Hz} \text { to } 100 \mathrm{kHz} \text { ) }$ |

ORIGIN SEARCH: ORG(889)

ORG(889) performs an origin search or origin return operation. The required PLC Setup parameters must be set before performing an origin search or origin return operation.

## Origin Search

Positions the system to the origin based on the origin proximity input and origin input signals.

## Origin Return

Returns the system from its present position to the pre-established origin.
P: Port specifier
C: Control data

| Operand |  |  | Contents |
| :---: | :---: | :---: | :---: |
| P | Port specifier |  | 0000 hex: Pulse output 0 0001 hex: Pulse output 1 |
| C | Control data | Bits 0 to 3 | Not used. (Always 0 hex.) |
|  |  | Bits 4 to 7 | Not used. (Always 0 hex.) |
|  |  | Bits 8 to 11 | Pulse output method (See note.) <br> 0 hex: CW/CCW <br> 1 hex: Pulse + direction |
|  |  | Bits 12 to 15 | Mode <br> 0 hex: Origin search <br> 1 hex: Origin return |

MODE CONTROL: INI(880)

Note
This section explains the functions related to pulse outputs only. For details on the $\operatorname{INI}(880)$ instruction's high-speed counter or interrupt functions, refer to 6-1 Interrupt Functions or 5-1 High-speed Counters.

| $\mathrm{INI}(880)$ |
| :---: |
| P |
| C |
| NV |

P: Port specifier
C: Control data
NV: First word of new PV

| Operand |  | Contents |
| :--- | :--- | :--- |
| P | Port specifier | 0000 hex: Pulse output 0 <br> 0001 hex: Pulse output 1 <br> 1000 hex: PWM output 0 <br> 1001 hex: PWM output 1 |
| C | Control data | 0002 hex: Change the PV. <br> 0003 hex: Stop pulse output. |
| NV | First word of new PV | NV and NV+1 contain the new PV when changing the <br> PV. (N contains the rightmost 4 digits and N+1 con- <br> tains the leftmost 4 digits.) <br> 0000 0000 to FFFFFFFF hex |

In addition to its interrupt and high-speed counter functions, PRV(881) can be used to read the pulse output PV or pulse output status information.
The status of the following flags is read as status information:

- Pulse Output Status Flag
- PV Underflow/Overflow Flag
- Pulse Output Amount Set Flag
- Pulse Output Completed Flag
- Pulse Output Flag
- No-origin Flag
- At Origin Flag
- Pulse Output Stopped Error Flag

| PRV(881) | P: Port specifier |
| :---: | :---: |
| P |  |
| C | C: Control data |
| D | D: First destination word |

Note This section explains the functions related to pulse outputs only. For details on the PRV(881) instruction's high-speed counter or interrupt functions, refer to 6-1 Interrupt Functions or 5-1 High-speed Counters.

| Operand |  |  | Contents |  |
| :---: | :---: | :---: | :---: | :---: |
| P | Port specifier |  | 0000 hex: Pulse output 0 0001 hex: Pulse output 1 1000 hex: PWM output 0 1001 hex: PWM output 1 |  |
| C | Control data |  | $\begin{aligned} & 0000 \text { hex: Read the PV. } \\ & 0001 \text { hex: Read the status. } \\ & 0003 \text { hex: Read the pulse output frequency. } \end{aligned}$ |  |
| D | First destination word | Reading PV <br> (D and D+1) | After the pulse output PV is read, the 8-digit hexadecimal data is stored in D and $\mathrm{D}+1$. ( D contains the rightmost 4 digits and D+1 contains the leftmost 4 digits.) |  |
|  |  | Reading pulse output status <br> (D) | Bit 0 | Pulse Output Status Flag <br> 0: Constant speed <br> 1: Accelerating/decelerating |
|  |  |  | Bit 1 | PV Underflow/Overflow Flag <br> 0: Normal <br> 1: Error |
|  |  |  | Bit 2 | Pulse Output Amount Set Flag <br> 0: Not set <br> 1: Set |
|  |  |  | Bit 3 | Pulse Output Completed Flag <br> 0: Output not completed <br> 1: Output completed |
|  |  |  | Bit 4 | Pulse Output Flag <br> 0: Stopped <br> 1: Outputting pulses |
|  |  |  | Bit 5 | No-origin Flag <br> 0: Origin established <br> 1: Origin not established |
|  |  |  | Bit 6 | At Origin Flag <br> 0: Not stopped at origin <br> 1: Stopped at origin |
|  |  |  | Bit 7 | Pulse Output Stopped Error Flag <br> 0 : No error <br> 1: Pulse output stopped due to error |
|  |  |  | Bits 8 to 15 | Not used. |
|  |  | Reading PWM output status (D) | Bit 0 | PWM Output Flag <br> 0: Stopped <br> 1: Outputting pulses |
|  |  |  | Bits 1 to 15 | Not used. |

PULSE WITH VARIABLE DUTY FACTOR: PWM(891)

PWM(891) is used to output pulses with the specified duty factor.

| PWM |  |
| :---: | :--- |
| $P$ | P: Port specifier |
| $F$ | F: Frequency |
| $D$ | D: Duty factor |


| Operand |  | Contents |
| :---: | :---: | :---: |
| P | Port specifier | 0000 hex: Pulse output 0 (duty factor set in $1 \%$ units, frequency 0.1 Hz units) <br> 0001 hex: Pulse output 1 (duty factor set in $1 \%$ units, frequency 0.1 Hz units) <br> 1000 hex: Pulse output 0 (duty factor set in $0.1 \%$ units, frequency 0.1 Hz units) <br> 1001 hex: Pulse output 1 (duty factor set in $0.1 \%$ units, frequency 0.1 Hz units) <br> 0100 hex: Pulse output 0 (duty factor set in $1 \%$ unit, frequency 1 Hz units) <br> 0101 hex: Pulse output 1 (duty factor set in $1 \%$ unit, frequency 1 Hz units) <br> 1100 hex: Pulse output 0 (duty factor set in $0.1 \%$ unit, frequency 1 Hz units) <br> 1101 hex: Pulse output 1 (duty factor set in $0.1 \%$ unit, frequency 1 Hz units) |
| T | Frequency | 0001 to FFFF hex ( 0.1 to 6553.5 Hz , in 0.1 Hz units) 0001 to 8020 hex ( 1 to $32,800 \mathrm{~Hz}$, in 1 Hz units) |
| S | Duty factor | Specify the duty factor of the pulse output, i.e., the percentage of time that the output is ON. <br> 0000 to 03E8 hex: $0.0 \%$ to $100.0 \%$ (in 0.1 units) 0000 to 0064 hex: $0.0 \%$ to $100 \%$ (in $1 \%$ units) |

## Combinations of Pulse Control Instructions

The following tables show when a second pulse control instruction can be started if a pulse control operation is already being executed.
Generally, a second independent-mode positioning instruction can be started if an independent-mode positioning instruction is being execute, and a second continuous-mode speed control instruction can be started if a continuousmode speed control instruction is being executed. Operation cannot be switched between the independent and continuous modes, although PLS2(887) can be started while $\mathrm{ACC}(888)$ (continuous mode) is being executed.
It is possible to start another operation during acceleration/deceleration and start another positioning instruction during positioning.

| Instruction being executed |  | ( $O$ : Can be executed., $x$ : Instruction instruction Error occurs and Error Flag goes ON) |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | INI(880) | SPED(885) (Independent) | SPED(885) (Continuous) | $\mathrm{ACC}(888)$ (Independent) | ACC(888) (Continuous) | PLS2(887) | ORG(889) |
| SPED(885) (Independent) |  | $\bigcirc$ | O (note 1) | $\times$ | O (note 3) | $\times$ | $\times$ | $\times$ |
| SPED(885) (Continuous) |  | $\bigcirc$ | $\times$ | O (note 2) | $\times$ | O (note 5) | $\times$ | $\times$ |
| $\begin{aligned} & \hline \begin{array}{l} \text { ACC(888) } \\ \text { (Independent) } \end{array} \end{aligned}$ | Steady speed | $\bigcirc$ | $\times$ | $\times$ | O (note 4) | $\times$ | O (note 6) | $\times$ |
|  | Accelerating or decelerating | $\bigcirc$ | $\times$ | $\times$ | O (note 4) | $\times$ | O (note 6) | $\times$ |
| ACC(888) (Continuous) | Steady speed | $\bigcirc$ | $\times$ | $\times$ | $\times$ | O (note 5) | O (note 7) | $\times$ |
|  | Accelerating or decelerating | $\bigcirc$ | $\times$ | $\times$ | $\times$ | O (note 5) | O (note 7) | $\times$ |
| PLS2(887) | Steady speed | $\bigcirc$ | $\times$ | $\times$ | O (note 4) | $\times$ | O (note 8) | $\times$ |
|  | Accelerating or decelerating | $\bigcirc$ | x | $\times$ | O (note 4) | $\times$ | O (note 8) | $\times$ |
| ORG(889) | Steady speed | $\bigcirc$ | $\times$ | $\times$ | $\times$ | $\times$ | $\times$ | $\times$ |
|  | Accelerating or decelerating | $\bigcirc$ | $\times$ | $\times$ | $\times$ | $\times$ | $\times$ | $\times$ |
| PWM |  | $\bigcirc$ | $\times$ | $\times$ | $\times$ | $\times$ | $\times$ | $\times$ |

Note (1) SPED(885) (Independent) to SPED(885) (Independent)

- The number of pulses cannot be changed.
- The frequency can be changed.
- The output mode and direction cannot be switched.
(2) SPED(885) (Continuous) to SPED(885) (Continuous)
- The frequency can be changed.
- The output mode and direction cannot be switched.
(3) SPED(885) (Independent) to ACC(888) (Independent)
- The number of pulses cannot be changed.
- The frequency can be changed.
- The acceleration/deceleration rate can be changed.
- The output mode and direction cannot be switched.
(4) ACC(888) (Independent) to ACC(888) (Independent) or PLS2(887) to ACC(888) (Independent)
- The number of pulses cannot be changed.
- The frequency can be changed.
- The acceleration/deceleration rate can be changed. (The rate can even be changed during acceleration or deceleration.)
- The output mode and direction cannot be switched.
(5) SPED(885) (Continuous) to ACC(888) (Continuous)
or ACC(888) (Continuous) to ACC(888) (Continuous)
- The frequency can be changed. (The target frequency can even be changed during acceleration or deceleration.)
- The acceleration/deceleration rate can be changed. (The rate can even be changed during acceleration or deceleration.)
- The output mode and direction cannot be switched.
(6) ACC(888) (Independent) to PLS2(887)
- The number of pulses can be changed. (The setting can even be changed during acceleration or deceleration.)
- The frequency can be changed. (The target frequency can even be changed during acceleration or deceleration.)
- The acceleration/deceleration rate can be changed. (The rate can even be changed during acceleration or deceleration.)
- The output mode and direction cannot be switched.
(7) ACC(888) (Continuous) to PLS2(887)
- The frequency can be changed. (The target frequency can even be changed during acceleration or deceleration.)
- The acceleration/deceleration rate can be changed. (The rate can even be changed during acceleration or deceleration.)
- The output mode and direction cannot be switched.
(8) PLS2(887) to PLS2(887)
- The number of pulses can be changed. (The setting can even be changed during acceleration or deceleration.)
- The frequency can be changed. (The target frequency can even be changed during acceleration or deceleration.)
- The acceleration/deceleration rate can be changed. (The rate can even be changed during acceleration or deceleration.)
- The output mode and direction cannot be switched.


## 5-2-9 Variable Duty Factor Pulse Outputs (PWM(891) Outputs)

## Overview

PWM (Pulse Width Modulation) pulse outputs can be output with a specified duty factor. The duty factor is the ratio of the pulse's ON time and OFF time in one pulse cycle. Use the PWM(891) instruction to generate variable duty factor pulses from a built-in output.
The duty factor can be changed while pulses are being output.

## Bit Allocations

| Word | Bit | Function |
| :---: | :--- | :--- |
| CIO 100 | 01 | PWM output 0 |
|  | 03 | PWM output 1 |

## Procedure



## Specifications

| Item | Specifications |
| :--- | :--- |
| Duty factor | ( $0.0 \%$ to $100.0 \%$ in $0.1 \%$ increments <br> (Duty factor accuracy is $+1 \% /-0 \%$ at10 kHz, $+5 \% /-0 \%$ at <br> 10 to 32.8 kHz.$)$ |
| Frequency | 0.1 Hz to $6,553.5 \mathrm{~Hz}$ <br> Set in 0.1 Hz units. (See note.) |
| Output mode | Continuous mode |
| Instruction | PWM(891) |

Note The frequency can be set up to 6553.5 Hz in the PWM(891) instruction, but the duty factor accuracy declines significantly at high frequencies because of limitations in the output circuit at high frequencies.

## 5-2-10 Example Pulse Output Applications

## Outputting Pulses after a Preset Delay

This example program waits for a preset time ( 0.5 ms ) after the interrupt input (CIO 0.04) goes ON and then outputs 100,000 pulses at 100 kHz from pulse output 0 .

Input interrupt task 0 (interrupt task number 140) starts a scheduled interrupt with a scheduled time of 0.5 ms . The scheduled interrupt task executes the pulse output instructions and stops the scheduled interrupt.


Instructions Used
MSKS(690) Enables the I/O interrupt. Starts the scheduled interrupt.
PULS(886) Sets the number of output pulses.
SPED(885) Starts the pulse output.

## Preparation

## - PLC Setup

## Built-in Input Settings

| PLC Setup setting details |
| :--- |
| Use built-in input 0.04 as the interrupt input. |



## Pulse Output 0 Settings

| PLC Setup setting details |
| :--- |
| Do not use high-speed counter 0. |
| Do not use the pulse output 0 origin search function. |



## Scheduled Interrupt Time Unit Setting

| PLC Setup setting details | Data |
| :--- | :---: |
| Set the scheduled interrupt time units to 0.1 ms. | 0002 hex |


| PPLC Settings - NewPLC1 |  |  | - - $\square^{\text {x }}$ |
| :---: | :---: | :---: | :---: |
| File Options Help |  |  |  |
| Watch Cycle Time (default 1000 ms ) $\square$ <br> Cycle Time (No Setting) $\square$ ms <br> Scheduled Interrupt Interval ms |  |  |  |
|  |  |  |  |

## Cyclic Task (Task 0)



## Built-in Input 0 Interrupt Task (Interrupt Task 140)



* Select 0.1 ms for the setting units in the PLC Setup.


## Scheduled Interrupt Task 0 (Interrupt Task 2)



## Positioning (Trapezoidal Control)

## Specifications and Operation

When the start input (0.00) goes ON, this example program outputs 600,000 pulses from pulse output 0 and turns the motor.


PLS2(887)

## Preparation

## DM Area Settings

## - PLC Setup

There are no settings that need to be made in the PLC Setup.

## PLS2(887) Settings (D00000 to D00007)

| Setting details | Address | Data |
| :--- | :--- | :--- |
| Acceleration rate: $300 \mathrm{~Hz} / 4 \mathrm{~ms}$ | D0 | 012 C |
| Deceleration rate: $200 \mathrm{~Hz} / 4 \mathrm{~ms}$ | D1 | 00 C 8 |
| Target frequency: $50,000 \mathrm{~Hz}$ | D2 | C350 |
|  | D3 | 0000 |
|  | D4 | 27 C 0 |
|  | D5 | 0009 |
| Starting frequency: 100 Hz | D6 | 0064 |
|  | D7 | 0000 |

## Ladder Program



## Remarks

- Absolute pulses can be specified when the origin position has been determined.
- If a target frequency that cannot be reached has been set, the target frequency will be reduced automatically, i.e., triangular control will be performed. In some cases where the acceleration rate is substantially greater than the deceleration rate, the operation won't be true triangular control. The motor will be operated at a constant speed for a short time between the acceleration and deceleration.


## Jog Operation

## Specifications and Operation

## Instructions Used

## Preparation

## DM Area Settings

- Low-speed jog operation (CW) will be executed from pulse output 1 while input 0.00 is ON .
- Low-speed jog operation (CCW) will be executed from pulse output 1 while input 0.01 is ON .

- High-speed job operation (CW) will be executed from pulse output 1 while input 0.04 is ON .
- High-speed jog operation (CCW) will be executed from pulse output 1 while input 0.05 is ON .


SPED(885) Starts and stops (immediate stop) the low-speed jog operations.
ACC(888) Starts and stops (decelerate to a stop) the high-speed jog operations.

## - PLC Setup

There are no settings that need to be made in the PLC Setup.
Settings to Control Speed while Jogging (D0 to D1 and D10 to D15)

| Setting details | Address | Data |
| :--- | :--- | :--- |
| Target frequency (low speed): $1,000 \mathrm{~Hz}$ | D0 | 03 E 8 |
|  | D1 | 0000 |
| Acceleration rate: $100 \mathrm{~Hz} / 4 \mathrm{~ms}$ | D10 | 0064 |


| Setting details | Address | Data |
| :--- | :--- | :--- |
| Target frequency (high speed): 100,000 Hz | D011 | 86 A 0 |
|  | D12 | 0001 |
| Deceleration rate: $100 \mathrm{~Hz} / 4 \mathrm{~ms}$ (Not used.) | D13 | 0064 |
| Target frequency (stop): 0 Hz | D14 | 0000 |
|  | D15 | 0000 |

## Ladder Program




## Remarks

PLS2(887) can be used to set a starting frequency or unequal acceleration and deceleration rates, but there are limitations on the operating range because the end point must be specified in PLS2(887).

## Cutting Long Material Using Fixed Feeding

Specifications and Operation

## Outline

In this example, first jogging is used to position the material and then fixeddistance positioning is used to feed the material.


## System Configuration



Built-in I/O other than pulse outputs are used.

## Operation

$1,2,3 . . . \quad$ 1. The workpiece is set at the starting position using the Jogging Switch Input (IN 0.00).
2. The workpiece is feed the specified distance (relative) using the Positioning Switch Input (IN 0.01).
3. When feeding has been completed, the cutter is activated using the Cutter Start Output (OUT 100.02).
4. Feeding is started again when the Cutter Finished Input (IN 0.02) turns ON.
5. The feeding/cutting operation is repeated for the number of times specified for the counter (C0, 100 times).
6. When the operation has been completed, the Cutting Operation Finished Output (OUT 100.03). is turned ON.
The feeding operation can be canceled and operation stopped at any point using the Emergency Switch Input (IN 0.03).

## Instructions Used

SPED(885)
PLS2(887)

## Preparation

## - PLC Setup

There are no settings that need to be made in the PLC Setup.

- DM Area Settings


## Speed Settings for Jogging (D0 to D3)

| Setting details | Address | Data |
| :--- | :--- | :--- |
| Target frequency: $1,000 \mathrm{~Hz}$ | D0 | 03 E 8 |
|  | D1 | 0000 |
| Target frequency: 0 Hz | D2 | 0000 |
|  | D3 | 0000 |

Settings for PLS2(887) for Fixed-distance Feeding (D10 to D20)

| Setting details | Address | Data |
| :--- | :--- | :--- |
| Acceleration rate: $1,000 \mathrm{~Hz} / 4 \mathrm{~ms}$ | D10 | 03 E 8 |
| Deceleration rate: $1,000 \mathrm{~Hz} / 4 \mathrm{~ms}$ | D11 | 03 E 8 |
| Target frequency: $10,000 \mathrm{~Hz}$ | D12 | 2710 |
|  | D13 | 0000 |
| Number of output pulses: 50,000 pulses | D14 | C350 |
|  | D15 | 0000 |
|  | D16 | 0000 |
|  | D17 | 0000 |
| Counter setting: 100 times | D20 | 0100 |

## Ladder Program



## Remarks

1,2,3... 1. PLS22(887) used a relative pulse setting. This enables operation even if the origin is not defined. The present position in A276 (lower 4 digits) and A277 (upper 4 digits) is set to 0 before pulse output and then contains the specified number of pulses.
2. $A C C(888)$ can be used instead of $\operatorname{SPED}(885)$ for the jog operation. If ACC(888) is used, acceleration/deceleration can be included in the jog operation.

## Vertically Conveying PCBs (Multiple Progressive Positioning)

## Specifications and Operation

## Outline

1,2,3... 1. PCBs with components mounted are stored in a stocker.
2. When a stocker becomes full, it is moved to the conveyance point.

## Positioning Operation for Vertical Conveyor

(2)



## - Operation Pattern

1,2,3... 1. An origin search is performed.
2. Fixed-distance positioning is repeated.
3. The system is returned to the original position.


## Wiring Example Using SmartStep A-series Servo Driver



## Operation

1,2,3... 1. An origin search is performed using the Origin Search Switch (CIO 0.00).
2. When the origin search is finished, the PCB Storage Enabled Output (CIO 100.03) is turned ON.
3. When a PCB has been stored, the stocker is raised (relative positioning) using the PCB Storage Completed Input (CIO 0.03).
4. Storing PCBs is repeated until the stocker is full.
5. The number of PCBs in the stocker is counted with counter CO by counting the number of times the stocker is raised.
6. When the stocker is full, it is moved (CIO 100.02) and only the conveyor is lowered (absolute positioning) when stoker movement is completed (CIO 0.04).

The operation can be canceled and pulse output stopped at any point using the Emergency Switch Input (CIO 0.01).

## Preparation

- PLC Setup

| Setting details |
| :--- |
| Enable origin search function for pulse output 0. |

Note The origin search enable setting is read when the power supply is turned ON.


DM Area Settings
Settings for PLS2(887) for Fixed-distance Positioning (D0 to D7)

| Setting details | Address | Data |
| :--- | :--- | :--- |
| Acceleration rate: $1,000 \mathrm{~Hz} / 4 \mathrm{~ms}$ | D0 | 03 E 8 |
| Deceleration rate: $1,000 \mathrm{~Hz} / 4 \mathrm{~ms}$ | D1 | 03 E 8 |
| Target frequency: $50,000 \mathrm{~Hz}$ | D2 | C350 |
|  | D3 | 0000 |
|  | D4 | 2710 |
|  | D5 | 0000 |
| Starting frequency: 0 Hz | D6 | 0000 |
|  | D7 | 0000 |

Settings for PLS2(887) to Return to Start (D10 to D17)

| Setting details | Address | Data |
| :--- | :--- | :--- |
| Acceleration rate: $300 \mathrm{~Hz} / 4 \mathrm{~ms}$ | D10 | 012C |
| Deceleration rate: $200 \mathrm{~Hz} / 4 \mathrm{~ms}$ | D11 | 00C8 |
| Target frequency: $50,000 \mathrm{~Hz}$ | D12 | C350 |
|  | D13 | 0000 |


| Setting details | Address | Data |
| :--- | :--- | :--- |
| Number of output pulses: $10,000 \times 15$ pulses | D14 | 49 FO |
|  | D15 | 0002 |
| Starting frequency: 100 Hz | D16 | 0000 |
|  | D17 | 0000 |

Number of Repeats of Fixed-distance Positioning Operation (D20)

| Setting details | Address | Data |
| :--- | :--- | :--- |
| Number of repeats of fixed-distance positioning operation <br> (number of PCBs in stocker) | D20 | 0015 |

## Ladder Program



Positioning
Lift 10,000 pulses (relative) at a time


Counter for Number of Lifts (Number of PCBs stored)



## Palletize: Two-axis Multipoint Positioning

## Specifications and

 Operation■ Outline


## - Operation Pattern

1,2,3... 1. An origin search is performed.
2. A workpiece is grasped and moved to position A.
3. The workpiece is grasped at one position and moved back and forth to several assembly positions.


Note The X and Y axes are moved independently, i.e., interpolation is not performed.

## Wiring Example Using SmartStep A-series Servo Driver



## X Axis

CP1L-M60/40/30DT-D, CP1L-L20DT-D
SMARTSTEP A-series Servo Driver


## Y Axis

CP1L-M60/40/30DT-D, CP1L-L20DT-D


## Operation

1,2,3... 1. An origin search is performed using the Origin Search Switch (CIO 0.00).
2. When the origin search is finished, the following operations are performed continuously.
Move to A.
Move to $B$ and return to $A$.
Move to $C$ and return to $A$.
Move to $D$ and return to $A$.
3. An emergency stop can be performed using the Emergency Stop Input (CIO 0.01)

## Preparation

## - PLC Setup

| Setting details |
| :--- |
| Enable origin search function for pulse output 0 and 1. |

Note The origin search enable setting is read when the power supply is turned ON.



## DM Area Settings

## Starting Frequency

| Setting details | Address | Data |
| :--- | :--- | :--- |
| X-axis starting frequency | D0 | 0000 |
| Y-axis starting frequency | D2 | 0000 |

PLS2(887) Settings to Move from Origin to Position A

| Setting details |  | Address | Data |
| :---: | :---: | :---: | :---: |
| X axis | Acceleration rate: $2,000 \mathrm{~Hz} / 4 \mathrm{~ms}$ | D10 | 07D0 |
|  | Deceleration rate: $2,000 \mathrm{~Hz} / 4 \mathrm{~ms}$ | D11 | 07D0 |
|  | Target frequency: 100,000 Hz | D12 | 86A0 |
|  |  | D13 | 0001 |
|  | Number of output pulses: 5,000 pulses | D14 | 1388 |
|  |  | D15 | 0000 |
| Y axis | Acceleration rate: $2,000 \mathrm{~Hz} / 4 \mathrm{~ms}$ | D20 | 07D0 |
|  | Deceleration rate: $2,000 \mathrm{~Hz} / 4 \mathrm{~ms}$ | D21 | 07D0 |
|  | Target frequency: $100,000 \mathrm{~Hz}$ | D22 | 86A0 |
|  |  | D23 | 0001 |
|  | Number of output pulses: 5,000 pulses | D24 | 1388 |
|  |  | D25 | 0000 |

PLS2(887) Settings to Move from Position A to Position B

| Setting details |  | Address | Data |
| :--- | :--- | :--- | :--- |
| X axis | Acceleration rate: $2,000 \mathrm{~Hz} / 4 \mathrm{~ms}$ | D30 | 07D0 |
|  | Deceleration rate: $2,000 \mathrm{~Hz} / 4 \mathrm{~ms}$ | D31 | 07D0 |
|  | Target frequency: $100,000 \mathrm{~Hz}$ | D32 | $86 \mathrm{A0}$ |
|  |  | D33 | 0001 |
|  | Number of output pulses: 25,000 pulses | D34 | 61 A 8 |
|  |  | D35 | 0000 |


| Setting details |  | Address | Data |
| :--- | :--- | :--- | :--- |
| Y axis | Acceleration rate: $2,000 \mathrm{~Hz} / 4 \mathrm{~ms}$ | D40 | 07D0 |
|  | Deceleration rate: $2,000 \mathrm{~Hz} / 4 \mathrm{~ms}$ | D41 | 07D0 |
|  | Target frequency: $100,000 \mathrm{~Hz}$ | D42 | $86 \mathrm{A0}$ |
|  |  | D43 | 0001 |
|  | Number of output pulses: 50,000 pulses | D44 | C350 |
|  |  | D45 | 0000 |

PLS2(887) Settings to Move from Position A to Position C

| Setting details |  | Address | Data |
| :--- | :--- | :--- | :--- |
| X axis | Acceleration rate: $2,000 \mathrm{~Hz} / 4 \mathrm{~ms}$ | D50 | 07D0 |
|  | Deceleration rate: $2,000 \mathrm{~Hz} / 4 \mathrm{~ms}$ | D51 | 07D0 |
|  | Target frequency: $100,000 \mathrm{~Hz}$ | D52 | $86 \mathrm{A0}$ |
|  |  | D53 | 0001 |
|  | Number of output pulses: 35,000 pulses | D54 | 88 B 8 |
|  |  | D55 | 0000 |
| Y axis | Acceleration rate: $2,000 \mathrm{~Hz} / 4 \mathrm{~ms}$ | D60 | $07 \mathrm{D0}$ |
|  | Deceleration rate: $2,000 \mathrm{~Hz} / 4 \mathrm{~ms}$ | D61 | $07 \mathrm{D0}$ |
|  | Target frequency: $100,000 \mathrm{~Hz}$ | D62 | $86 \mathrm{A0}$ |
|  | Number of output pulses: 50,000 pulses | D63 | 0001 |
|  |  | D64 | C350 |
|  |  |  | 0000 |

PLS2(887) Settings to Move from Position A to Position D

| Setting details |  | Address | Data |
| :--- | :--- | :--- | :--- |
| X axis | Acceleration rate: $2,000 \mathrm{~Hz} / 4 \mathrm{~ms}$ | D70 | 07D0 |
|  | Deceleration rate: $2,000 \mathrm{~Hz} / 4 \mathrm{~ms}$ | D71 | 07D0 |
|  | Target frequency: $100,000 \mathrm{~Hz}$ | D72 | $86 \mathrm{A0}$ |
|  |  | D73 | 0001 |
|  | Number of output pulses: 25,000 pulses | D74 | $61 \mathrm{A8}$ |
|  |  | D75 | 0000 |
| Y axis | Acceleration rate: $2,000 \mathrm{~Hz} / 4 \mathrm{~ms}$ | D80 | 07 D |
|  | Deceleration rate: $2,000 \mathrm{~Hz} / 4 \mathrm{~ms}$ | D81 | $07 \mathrm{D0}$ |
|  | Target frequency: $100,000 \mathrm{~Hz}$ | D82 | $86 \mathrm{A0}$ |
|  | Number of output pulses: 30,000 pulses | D83 | 0001 |
|  |  | D84 | 7530 |
|  |  | D85 | 0000 |

## Ladder Program







## Feeding Wrapping Material: Interrupt Feeding

## Specifications and Operation

Feeding Wrapping Material in a Vertical Pillow Wrapper


## Operation Pattern

Speed control is used to feed wrapping material to the initial position. When the marker sensor input is received, fixed-distance positioning is performed before stopping.


## Operation

1,2,3... 1. Speed control is used to feed wrapping material to the initial position when the Start Switch (CIO 0.00) is activated.
2. When the Marker Sensor Input (0.04) is received, PLS2(887) is executed in interrupt task 140.
3. Fixed-distance positioning is executed with PLS2(887) before stopping.
4. An emergency stop is executed to stop pulse output with the Emergency Stop input (0.01).

## Preparation

- PLC Setup

| Setting details |
| :--- |
| Enable using built-in input IN0 as an interrupt input. |

Note The interrupt input setting is read when the power supply is turned ON.


■ DM Area Settings
Speed Control Settings to Feed Wrapping Material to Initial Position

| Setting details | Address | Data |
| :--- | :--- | :--- |
| Acceleration rate: $1,000 \mathrm{~Hz} / 4 \mathrm{~ms}$ | D0 | 03 E 8 |
| Target frequency: $10,000 \mathrm{~Hz}$ | D1 | 2710 |
|  | D2 | 0000 |

## Positioning Control Settings for Wrapping Material

| Setting details | Address | Data |
| :--- | :--- | :--- |
| Acceleration rate: $500 \mathrm{~Hz} / 4 \mathrm{~ms}$ | D10 | 01 F 4 |
| Deceleration rate: $500 \mathrm{~Hz} / 4 \mathrm{~ms}$ | D11 | 01 F 4 |
| Target frequency: $10,000 \mathrm{~Hz}$ | D12 | 2710 |
|  | D13 | 0000 |
| Number of output pulses: 5,000 pulses | D 14 | 1388 |
|  | D15 | 0000 |
| Starting frequency: 0 Hz | D 16 | 0000 |
|  | D17 | 0000 |

## Ladder Program

## Cyclic Task Program

## (Executed at Startup)



Program for Interrupt Task
140


## 5-3 Inverter Positioning

## 5-3-1 Features

## Feedback Control with Error Counter

Positioning can be achieved using an inverter. This enables a far more economical positioning system than with a servomotor.

A position error counter built into the CP1L CPU Unit enables high-precision positioning with an Inverter using feedback control. The PULSE OUTPUT instruction is used in the ladder program in the CP1L CPU Unit to output internal pulses to a built-in error counter.
The error counter calculates the position error from the number of input internal pulses and the number of feedback pulses from the rotary encoder, and sends speed commands to the inverter so that the position error goes to zero.


Reducing Positioning Time

With traditional inverter positioning, positioning patterns are created in which set positions are detected to trigger changes in the speed. Pulses are read from the encoder and compared to set values during positioning to enable determining when a position requiring a speed change has been reached. This results in positioning errors at speed-change points when stopping at high speed, reducing stopping precision. To ensure high-precision positioning, sufficient deceleration was required before stopping, but this increases the positioning time.
With the CP1L's inverter positioning function, feedback pulses are used so that the prevent position is always known, increasing positioning accuracy. And because preset positioning patterns are used for deceleration and stopping, positioning time is reduced.

## Traditional Inverter Positioning

The PLC counts the feedback pulses from the encoder using a high-speed counter. When a deceleration point is reached, the speed is changed to control the stop position. If the precision of the stop position must be increased, the stop position must also be detected to control positioning.



Stopping from a low speed is necessary to prevent positioning error.

The speed of high-speed operation produces error, requiring tha the deceleration start position be calculated and adjusted so that a low speed is achieved near the stopping point

The need for low-speed operation near the stopping position increases the positioning time. PLC

## Inverter Positioning with the CP1L

With the CP1L＇s inverter positioning function，feedback is constantly read for the positioning data while controlling the position．


」几几亿


Note
（1）The CP1L＇s inverter positioning function is designed to increase position－ ing speed and stopping precision by reading position information and us－ ing a feedback loop with an error counter to switch speeds．It does not increase the response，stopping precision，or speed change rate of the inverter and motor．These are characteristics of the inverter and motor． Refer to user documentation on your inverter and motor for details．
（2）The corresponding pulse output number（0 or 1）cannot be used for the PULSE WITH VARIABLE DUTY FACTOR instruction（PWM）if inverter positioning 0 or 1 is used．The high－speed counter of the same number （0 or 1 ）is used to input the feedback pulse．

## 5-3-2 System Configuration

Speed Commands Using Serial Communications or Analog Outputs

There are two ways to send speed commands to the inverter: serial communications and analog outputs.


Note (1) The inverter positioning function uses either serial communications or an analog output, and is thus possible with a CP1L CPU Unit with either transistor or relay outputs.
(2) The inverter positioning function does not use external pulse outputs. Normal outputs are used for commands to the inverter (e.g., forward/reverse commands).

## ■ Precaution for Inverter Settings

- Set the stop time to 0 second.
- Use Modus-RTU communication when the send delay setting is above 10 ms .
However, if the send delay time is too long, the inverter response to the command from the PLC will be slow.


## 5-3-3 Functional Overview

## Operation



1. To use inverter positioning, the motor and encoder specifications, feedback gain, and other parameters must be set in the PLC Setup. The highspeed counter and inverter must also be set.
2. Pulse output instructions, such as PLS2 or PULS with SPED, are used to execute positioning. Although normally the pulse output instructions are used to output pulses from CP1L output contacts, when inverter positioning 0 or inverter positioning 1 is enabled in the PLC Setup, the internal position error counter (called simply the "error counter") is enabled and the pulse output instruction will output internal pulses to the error counter. Both error counters 0 and 1 can be used at the same time.
3. For the number of pulses (i.e., the amount of movement) set in the pulse output instruction, use the number of feedback pulses from the encoder. For the pulse frequency set in the pulse output instruction, use the motor power supply frequency converted to the feedback pulse frequency from the encoder. (Refer to 5-3-7 Determining the Internal Pulse Output Frequency for details.)
4. Specify an inverter positioning port for the pulse output instruction (port 0: 0020, port 1: 0021). The internal pulses will be output to the error counter for the specified port.
5. The number of pulses remaining in the error counter is converted to a power supply frequency command for the inverter according to a value set in the PLC Setup and output to a word in the Auxiliary Area in increments of 0.01 Hz .
6. The frequency command value output to the Auxiliary Area is output to the inverter from the ladder program according to the inverter command method (i.e., RS-485 communications or analog output). (Refer to 5-3-9 Automatic Calculation of Inverter Frequency Command Value for details.)
7. When a speed command is sent to the inverter, the motor will turn at the command speed and feedback pulses (i.e., the amount of movement) from the encoder will be returned to a high-speed counter of the CP1L. The CP1L will continue to send a speed command to the inverter until the error counter (i.e., the position error) goes to zero, i.e., until positioning has been completed.
8. When the error counter goes to zero, the speed command to the inverter will also go to zero. Even after the completion of internal pulse output (i.e., position command) from the pulse output instruction, the CP1L will maintain the error counter so that is remains at zero.
9. The status of the error counter (such as the command direction and in-position status) will be stored in the Auxiliary Area. This status can be read from the user program to enable controlling output of commands to the inverter.
For example, if a change in the load causes the motor shaft to turn, feedback pulses from the encoder will enter the error counter, the value in the error counter will be reduced, and the Reverse Command Flag in the Auxiliary Area will turn ON. By writing the ladder program to output a reverse operation command to the inverter for the Reverse Command Flag, a command in the opposite direction of motor shaft movement will be output from the CP1L to the inverter, causing the motor to return to its original position. This compensating operation to continuously maintain the current stop position is called a servo lock.

## Other Functions

## Servo Locks with Vector Control Inverters

## Clearing the Error Counter for Errors

The servo lock on an inverter can be used to stop positioning. By using the inverter's servo lock, the inverter positioning function and the output command to the inverter can be stopped from the user program without using feedback control even if the error counter value is not zero. This enables servo locks when using an inverter with vector control.

If the motor shaft is moved manually for error stops or when the inverter is stopped, feedback pulses will accumulate in the error counter. This can be very dangerous because it may cause the motor to suddenly return to the original position at high speed when operation is started again. To prevent such problems, an error counter error output can be produced when more than a set number of pulses accumulated in the error counter when positioning operations are stopped.

## 5-3-4 Specifications

## Inverter Positioning

## Low-speed Operation Using Minimum Output Setting

## Absolute Positioning

Note The absolute position will change if the present value of the high-speed counter is changed or the high-speed counter is reset.
An inductive motor driven with an inverter is different from a servomotor in that the torque at low speeds is so low that it may not be possible to turn the motor shaft at the minimum frequency. The CP1L provide a minimum output setting the ensure a minimum output to enable positioning at low speeds even when there are extremely few pulses in the error counter.

The amount of movement (i.e., amount of rotation) is input to the high-speed counter as feedback pulses. During inverter positioning, the present value of the high-speed counter can be used as an absolute position.


## Specifications

| Item | Specification |
| :--- | :--- |
| Applicable inverters | Inverter that receives frequency commands from an analog <br> input or via Modbus-RTU communications. (Control method: <br> V/f control, vector control, etc.). |
| Applicable motors | Depends on the inverter (e.g., squirrel-cage inductive motor) |
| Number of position- <br> ing ports and <br> response frequency | Two ports at 100 kHz (within the speed command range of the <br> pulse output instructions) |
| Inverter command <br> output method | Modbus-RTU communications commands or analog output <br> (from ladder program) |
| Present value coor- <br> dinate system | With origin: Absolute coordinate system <br> Without origin: Relative coordinate system |
| Present value range | 32 bits: 8000 000 to 7FFF FFFF hex (range of position com- <br> mand values and present values for pulse output instructions) |
| Output modes | Continuous output (Number of pulses not specified.) <br> Independent mode (Number of pulses specified.) |
| Acceleration/decel- <br> eration control | Trapezoidal or S-curve acceleration/deceleration |


| Item | Specification |
| :--- | :--- |
| Specifications of <br> number of pulses | Relative positions: 00000000 to 7FFF FFFF hex <br> $(2,147,483,647$ incrementing and decrementing) <br> Absolute positions: 80000000 to 7FFF FFFF hex <br> $(-2,147,483,648$ to 2,147,483,647) <br> (Ranges of position command values and present values for <br> pulse output instructions) |
| Origin searches | Motor driver and signal wire modes: 3 modes <br> Origin search modes: 2 modes <br> Origin detection methods: 3 methods |
| Feedback pulse <br> input ports | High-speed counter 0 and high-speed counter 1 (fixed) <br> Maximum response frequency: 100 kHz |
| Present value range <br> for feedback pulses | 32 bits: 8000 000 to 7FFF FFFF hex |
| Error counter range | 8000 to 7FFF hex (signed) |
| Error counter calcu- <br> lation cycle | 4 to 1,020 ms (x4) |

## Note

High-speed Counter Specifications for Inverter Positioning
(1) If inverter positioning 0 is used, pulse output 0 and PWMO cannot be used. If inverter positioning 1 is used, pulse output 1 and PWM1 cannot be used.
(2) If inverter positioning 1 is used with a CPU Unit with 14 I/O Points, origin searches cannot be used.
(3) If the continuous output mode is specified (i.e., if the number of pulses is not specified), be sure to use the high-speed counter (linear mode) so that it does not overflow.

| Item | Specification |
| :--- | :--- |
| Response frequency and num- <br> ber of counters | Two 2-phase counters at 50 kHz and two single- <br> phase counters at 100 kHz |
| Counting mode | Differential-phase inputs (x4), up/down pulse <br> inputs, or pulse plus direction inputs |
| Numeric range mode | Linear mode <br> Note Always set linear mode when using inverter <br> positioning. |
| Numeric range | 32 bits (-2,147,483,648 to 2,147,483,647) |
| Reset method | Phase Z signal (reset input) + software reset, or <br> software reset |
| Interrupts <br> (See note.) | Target value <br> matching |
|  | Up to 48 target values and interrupt task numbers <br> can be registered. |
| son compari- |  | | Up to 8 sets of upper values, lower values, and |
| :--- |
| interrupt task numbers can be registered. |

Note Target value matching and zone comparisons can be used for high-speed counters with a feedback pulse input from an encoder even when using inverter positioning.

## 5-3-5 Application Procedure for Inverter Positioning



- MODE CONTROL (INI)

Port: Inverter positioning, stopping inverter positioning

- HIGH-SPEED COUNTER PV READ (PRV) Port: Inverter positioning, Operation: Reading error counter, inverter positioning status, or error counter present value


## ■ Automatic Calculation of Inverter Frequency Commands

- For either serial communications or an analog output, the power supply frequency per motor revolutions/s, the number of encoder pulses per motor revolution, and the error counter calculation cycle can be set in the PLC Setup to automatically calculate the inverter frequency command values and store it in A23/A33 in increments of 0.01 Hz .
- For serial communications, the ladder program is used to output the value in A23/A33 to the inverter using serial communications.
- For analog output, the value in A23/A33 can be scaled to analog output values and output from the Analog Unit to the inverter.


## ■ Forward/Reverse Operation Commands, Stopping Commands, Etc.

- The Forward Command Flag (A26.01/A36.01) and Reverse Command Flag (A26.02/A36.02) can be used as input conditions for forward and reverse operation commands.
- The Operation Command Flag (A26.00/A36.00) and In-position Flag (A26.03/A36.03) can be used as input conditions to execute scaling to inverter frequency commands and to execute stop commands.


## 5-3-6 Instruction Specifications

The normal pulse output instructions are used (PLS2, PULS + SPED, or PULS + ACC). One of the inverter positioning ports is specified as the port for the instruction. Just like pulses are output externally for the normal pulse output instructions, error counter pulses are accumulated in the internal error counter when executing inverter positioning.

When executing pulse output instructions or status read instructions for inverter positioning, a port number for inverter positioning is specified for the port operand of the instruction. The following values are used.
0020 hex: Inverter positioning 0
0021 hex: Inverter positioning 1
When reading the present value of inverter positioning, use the following values to specified the port number for inverter positioning.
0030 hex: Inverter positioning 0 (signed)
0031 hex: Inverter positioning 1 (signed)

| Set value | Specified port | Applicable instructions |
| :--- | :--- | :--- |
| 0000 | Pulse output 0 | --- |
| 0001 | Pulse output 1 | --- |
| 0002 | Pulse output 2 | --- |
| 0003 | Pulse output 3 | --- |
| 0010 | High-speed counter input 0 | --- |
| 0011 | High-speed counter input 1 | --- |
| 0012 | High-speed counter input 2 | --- |
| 0013 | High-speed counter input 3 | --- |
| 0020 | Inverter positioning 0 | SPED, PULS, ACC, PLS2, INI, PRV, ORG |
| 0021 | Inverter positioning 1 | SPED, PULS, ACC, PLS2, INI, PRV, ORG |


| Set value | Specified port | Applicable instructions |
| :--- | :--- | :--- |
| 0030 | Error counter 0 (signed) | PRV |
| 0031 | Error counter 1 (signed) | PRV |
| 0100 | Interrupt input 0 (counter <br> mode) | --- |
| $:$ | $:$ | $:$ |
| 0107 | Interrupt input 7 (counter <br> mode) | --- |
| 1000 | PWM output 0 | --- |
| 1001 | PWM output 1 | --- |

## Applicable Instructions

The following seven instructions can be used to execute inverter positioning. The relationship between the instructions and internal pulse outputs is as follows:

| Instruction | Overview | Positioning (Independent Mode) |  |  | Origin searches |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Pulse output with no acceleration/ deceleration | Pulse output with acceleration/ deceleration |  |  |
|  |  |  | Trapezoid, same rate for acceleration/ deceleration | Trapezoid, different rates for acceleration/ deceleration |  |
| PULS(886) SET PULSES | Sets the number of internal pulses to output. | Applicable | --- | --- | --- |
| SPED(885) SPEED OUTPUT | Controls pulse output without acceleration or deceleration. (The number of internal pulses must be set in advance with PULS(886).) | Applicable | --- | --- | --- |
| ACC(888) <br> ACCELERATION CONTROL | Controls pulse output with acceleration or deceleration using the same rate for both. (The number of internal pulses must be set in advance with PULS(886).) | --- | Applicable | --- | --- |
| PLS2(882) PULSE OUTPUT | Controls pulse output with acceleration or deceleration using a different rate for each (The number of internal pulses is also set.) | -- | --- | Applicable | --- |
| ORG(889) ORIGIN SEARCH | Actually moves the motor to establish the origin using origin proximity input, origin input, etc. | --- | --- | --- | Applicable |
| $\begin{aligned} & \text { INI(880) } \\ & \text { MODE CONTROL } \end{aligned}$ | Used to stop internal pulse output and inverter positioning. It can also be used to change the present value of pulse output (thus establishing the origin). | Applicable | Applicable | Applicable | --- |
| $\begin{aligned} & \text { PRV(881) } \\ & \text { HIGH-SPEED } \\ & \text { COUNTER PV } \\ & \text { READ } \end{aligned}$ | Reads the present value of the internal pulse output or error counter. | Applicable | Applicable | Applicable | --- |

SET PULSES: PULS(886)

## SPEED OUTPUT: SPED(885)

ACCELERATION CONTROL: ACC(888)

PULS(886) is used to set the pulse output amount (number of output pulses) for pulse outputs that are started later in the program using $\operatorname{SPED}(885)$ or ACC(888) in independent mode.

| PULS(886) |  |
| :---: | :---: |
| P | --- P: Port specifier |
| T | ----- T: Pulse type |
| N | -- N: Number of pulses |


| Operand |  | Description |
| :--- | :--- | :--- |
| P | Port specifier | 0020 hex: Inverter positioning 0 <br> 0021 hex: Inverter positioning 1 |
| T | Pulse type | 0000 hex: Relative <br> 0001 hex: Absolute |
| N | Number of <br> pulses | N (lower 4 digits) |
|  | N+1 Relative pulses: 00000000 to 7FFF FFFF <br> digits) | hex (0 to 2,147,489,647) <br> - Absolute pulses: 8000 0000 to 7FFF FFFF <br> hex (-2,147,489,648 to 2,147,489,647) |

SPED(885) is used to start pulse output without acceleration or deceleration. It is used together with PULS(886). SPED(885) can also be executed during pulse output to change the output frequency.


| Operand |  |  | Description |
| :---: | :---: | :---: | :---: |
| P | Port specifier |  | 0020 hex: Inverter positioning 0 0021 hex: Inverter positioning 1 |
| M | Output mode | Bits 0 to 3 | Mode <br> 0 hex: Continuous <br> 1 hex: Independent |
|  |  | Bits 4 to 7 | Direction <br> 0 hex: CW <br> 1 hex: CCW |
|  |  | Bits 8 to 11 | Not used: Always set to 0 hex. |
|  |  | Bits 9 to 15 | Not used: Always set to 0 hex. |
| F | First pulse frequency word | F (lower 4 digits) | Output Frequency in Hz <br> Pulse output 0 or 1: 00000000 to 0001 86A0 hex (0 to 100 kHz ) |
|  |  | F+1 (upper 4 digits) |  |

ACC(888) outputs pulses to the specified output port at the specified frequency using the specified acceleration and deceleration rate. (Acceleration rate is the same as the deceleration rate.) For positioning, $\operatorname{ACC}(888)$ is used in combination with PULS(886). ACC(888) can also be executed during pulse output to change the target frequency or acceleration/deceleration rate.


| Operand |  | Description |
| :--- | :--- | :--- |
| P | Port specifier | 0020 hex: Inverter positioning 0 <br>  |


| Operand |  |  | Description |
| :---: | :---: | :---: | :---: |
| M | Output mode | Bits 0 to 3 | Mode <br> 1 hex: Independent |
|  |  | Bits 4 to 7 | Direction <br> 0 hex: CW <br> 1 hex: CCW |
|  |  | Bits 8 to 11 | Not used: Always set to 0 hex. |
|  |  | Bits 9 to 15 | Not used: Always set to 0 hex. |
| S | First word of settings table | S | Acceleration/Deceleration Rate 1 to $65,535 \mathrm{~Hz}$ ( 0001 to FFFF hex) |
|  |  | S+1 (lower 4 digits) | Target Frequency in Hz <br> Pulse output 0 to 3: 00000000 to 0001 86A0 |
|  |  | $\begin{aligned} & \hline \begin{array}{l} \text { S+2 (upper } 4 \\ \text { digits) } \end{array} \\ & \hline \end{aligned}$ | hex (0 to 100 kHz ) |

## PULSE OUTPUT: PLS2(887)

PLS2(887) outputs a specified number of pulses to the specified port. Pulse output starts at a specified startup frequency, accelerates to the target frequency at a specified acceleration rate, decelerates at the specified deceleration rate, and stops at approximately the same frequency as the startup frequency. Only independent mode positioning is supported.
PLS2(887) can also be executed during pulse output to change the number of output pulses, target frequency, acceleration rate, or deceleration rate. PLS2(887) can thus be used for sloped speed changes with different acceleration and deceleration rates, target position changes, target and speed changes, or direction changes.


| Operand |  |  | Description |
| :---: | :---: | :---: | :---: |
| P | Port specifier |  | 0020 hex: Inverter positioning 0 0021 hex: Inverter positioning 1 |
| M | Output mode | Bits 0 to 3 | Mode <br> 0 hex: Relative pulses <br> 1 hex: Absolute pulses |
|  |  | Bits 4 to 7 | Direction <br> 0 hex: CW <br> 1 hex: CCW |
|  |  | Bits 8 to 11 | Not used: Always set to 0 hex. |
|  |  | Bits 9 to 15 | Not used: Always set to 0 hex. |
| S | First word of settings table | S1 | Acceleration rate Specify the increase or <br> 0001 to FFFF hex (1 to decrease in the fre- <br> $65,535 \mathrm{~Hz}$ ) quency in Hz per pulse |
|  |  | S1+1 | Deceleration rate 0001 to FFFF hex (1 to $65,535 \mathrm{~Hz}$ ) |
|  |  | $\begin{array}{\|l} \hline \begin{array}{l} \text { S1+2 (lower } 4 \\ \text { digits) } \end{array} \\ \hline \end{array}$ | Target Frequency in Hz <br> Pulse output 0 or 1: 00000000 to 0001 86A0 hex (0 to 100 kHz ) |
|  |  | $\begin{array}{\|l} \hline \begin{array}{l} \text { S1+3 (upper } 4 \\ \text { digits) } \end{array} \\ \hline \end{array}$ |  |
|  |  | S1+4 (lower 4 digits) | Number of Pulses <br> - Relative pulses: 00000000 to 7FFF FFFF hex (0 to 2,147,489,647) <br> - Absolute pulses: 80000000 to 7FFF FFFF hex $(-2,147,489,648$ to $2,147,489,647)$ |
|  |  | $\begin{aligned} & \begin{array}{l} \text { S1+5 (upper } 4 \\ \text { digits) } \end{array} \\ & \hline \end{aligned}$ |  |

ORIGIN SEARCH: ORG(889)

| Operand |  | Description |  |
| :--- | :--- | :--- | :--- |
| F | First word of <br> starting fre- <br> quency | F (lower 4 dig- <br> its) | Starting Frequency in Hz <br> Pulse output 0 or 1:0000 0000 to 0001 86A0 <br> hex (0 to 100 kHz) |
|  | F+1 (upper 4 <br> digits) |  |  |

ORG(889) performs an origin search or origin return operation.

- Origin Search:

Pulses are output to establish the origin based on origin proximity input and origin input signals.

- Origin Return:

The positioning system is returned to the origin.
The parameters for pulse output 0 or pulse output 1 must be set in advance in the PLC Setup to perform either an origin search or origin return operation.


| Operand |  | Description |  |
| :--- | :--- | :--- | :--- |
| P | Port specifier | 0020 hex: Inverter positioning 0 <br> 0021 hex: Inverter positioning 1 |  |
| C | Control data | Bits 0 to 3 | Not used: Always set to 0 hex |
|  |  | Bits 4 to 7 | Not used: Always set to 0 hex |
|  | Bits 8 to 11 | Not used: Always set to 0 hex |  |
|  | Bits 9 to 15 | Mode <br> 0 hex: Origin search <br> 1 hex: Origin return |  |

MODE CONTROL: INI(880)
$\mathrm{INI}(880)$ changes the present value of inverter positioning or stops positioning.


P: Port specifier
---- C: Control data
---- NV: First word with new PV

| Operand |  | Description |
| :--- | :--- | :--- |
| P | Port specifier | 0020 hex: Inverter positioning 0 <br> 0021 hex: Inverter positioning 1 |
| C | Control data | 0002 hex: Changes the PV of the internal pulse <br> output. <br> 0003 hex: Stops internal pulse output. Position- <br> ing will continue and the output value will not be <br> cleared. <br> 0004 hex: Stops inverter positioning. Internal <br> pulse output will be stopped, positioning will be <br> stopped, and the output value will be cleared. <br> The next operation will not be accepted until the <br> error counter is cleared. |
| NP | First word <br> with new PV | NP (lower 4 <br> digits) |
|  | NP+1 (upper <br> 4 digits) | New PV <br> 0000 0000 to FFFF FFFF hex |

HIGH-SPEED COUNTER PV READ: PRV(881)
$\mathrm{PRV}(881)$ is used to read the present value and status of inverter positioning. The following status can be read.

- Operation Command Flag
- Internal Pulse Acceleration/
- Forward Command Flag
- Reverse Command Flag
- In-position Flag
- Internal Pulse Output Flag Deceleration Flag
- Error Counter Error Flag
- Error Counter Alarm Flag
- Error Counter Sign Flag

| Operand |  | Description |
| :--- | :--- | :--- |
| P | Port specifier | 0020 hex: Inverter positioning 0 |
|  |  | 0021 hex: Inverter positioning 1 |
|  |  | 0030 hex: Error counter 0 |
|  |  | 0031 hex: Error counter 1 |
| C | Control data | 0000 hex: Read present value. <br>  |


| Operand |  |  | Description |  |
| :---: | :---: | :---: | :---: | :---: |
| D | First destination word for present value | D | Lower 4 digits <br> Upper 4 digits | When a present value is read, the following data is stored in D and $\mathrm{D}+1$ as an 8-digit hexadecimal value. <br> P = \#0020/\#0021: The actual movement from the internal pulse origin. <br> $\mathrm{P}=$ \#0030/\#0031: The present value of the error counter. |
|  | Destination word for inverter positioning status ( $\mathrm{P}=\# 0020$ or \#0021) | D | Bit 0 | Operation Command Flag <br> ON: Operation command in progress OFF: Stopped |
|  |  |  | Bit 1 | Forward Command Flag ON: Forward command in progress OFF: Reverse command in progress or stopped |
|  |  |  | Bit 2 | Reverse Command Flag ON: Reverse command in progress OFF: Forward command in progress or stopped |
|  |  |  | Bit 3 | In-position Flag ON: In position OFF: Not in position |
|  |  |  | Bit 4 | Error Counter Error Flag <br> ON: Error occurred in error counter OFF: No error |
|  |  |  | Bit 5 | Internal Pulse Output Flag <br> ON: Pulses being output <br> OFF: Pulse output stopped |
|  |  |  | Bit 6 | Internal Pulse Acceleration/Deceleration Flag ON: Acceleration/deceleration in progress for internal pulse output (i.e., frequency being changed) <br> OFF: Constant frequency for internal pulse output |
|  |  |  | Bit 7 | Error Counter Alarm Flag <br> ON: Alarm occurred for error counter OFF: No alarm |
|  |  |  | Bit 15 | Error Counter Sign Flag ON: Positive OFF: Negative |

## 5-3-7 Determining the Internal Pulse Output Frequency

Use the following formula to calculate the internal pulse frequency $(\mathrm{Hz})$ to output from the pulse output instruction (e.g., PLS2) based on the power supply frequency $(\mathrm{Hz})$ to be output from the inverter to the motor.

(1) Calculate the power supply frequency for one revolution per second from the motor specifications. For example, with a $1,800-\mathrm{r} / \mathrm{min}(60-\mathrm{Hz})$ motor $(30 \mathrm{r} / \mathrm{s})$, the power supply frequency for one revolution per second would be calculated as follows: $60[\mathrm{~Hz}] \div 30[\mathrm{r} / \mathrm{s}]=2[\mathrm{~Hz}]$.
(2) The encoder resolution times the counter multiplier times the gear ratio equals the number of pulses output by the encoder for one motor shaft revolution.


## Example of <br> Calculating Conversion Factor

## Conditions

- Frequency for 1 revolution/s for inductive motor: 2 Hz (motor specification)
- Rotary encoder resolution: 1,000 pulses/revolution (encoder specification)
- High-speed counter multiplier: x4 (PLC Setup)
- Gear ratio between motor and encoder shafts: 1/4 (machine specification)


## Calculations

The factor goes into the formula as shown below.

| Frequency of <br> internal pulse <br> output (Hz) | $=$$1000 \times 4 \times 1 / 4$ <br> 2 |
| ---: | :--- |
|  | $=500 \times$ Power supply frequency to motor $(\mathrm{Hz})$ |

For example, to output a power supply frequency of 10 Hz to the motor:
Frequency of internal pulse output $=500 \times 10 \mathrm{~Hz}=5,000 \mathrm{~Hz}=5 \mathrm{kHz}$
Therefore, set a pulse output frequency of 5 kHz in the pulse output instructions (e.g., PLS2).

## 5-3-8 PLC Setup

## Basic Settings

The following settings must be made in advance when using inverter positioning 0 or 1 .

The following settings are required to use inverter positioning.

## Inverter Positioning

 Function| Setting | Description | Set value | Default | Application | Refresh timing |
| :--- | :--- | :--- | :--- | :--- | :--- |
| Use <br> inverter <br> positioning | Select this option to use inverter <br> positioning. High-speed counter 0 <br> will be allocated to inverter posi- <br> tioning 0 and high-speed counter 1 <br> will be allocated to inverter posi- <br> tioning 1. The high-speed counter <br> mode that is set will be used. | Use/Do not use | Do not use | --- | When CPU Unit <br> power is turned <br> ON |
|  | Note If inverter positioning 1 is <br> used with a CPU Unit with <br> 14 I/O Points, origin <br> searches cannot be used. <br> (Origin searches are possi- <br> ble even if inverter position- <br> ing 0 is used.) |  |  |  |  |

## Gain

| Setting | Description | Set value | Default | Application | Refresh timing |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Gain | The error counter present value times the gain setting will be used as the output command to the inverter. <br> Note The setting is made in increments of 0.1 . The gain will thus be $1 / 10$ of the set value. For example, if 50 is set, the gain will be 5 . <br> It's best to initially try a gain of from 5 to 10 (settings of 50 to 100 ) and then adjust from there. | 1 to 65,535 (0.1 increments) 0 sets a value of 10 (0.1 increments) | 0: 10 (0.1 <br> incre- <br> ments) <br> This will <br> set a <br> gain of 1 | Adjusting the following characteristic of the motor | When CPU Unit power is turned ON |

In-position Range

| Setting | Description | Set value | Default | Application | Refresh timing |
| :--- | :--- | :--- | :--- | :--- | :--- |
| In-position <br> range | The In-position Flag (A26.03) will turn <br> ON when pulse output to the error <br> counter has been completed and the <br> error counter present value is less equal <br> to or less than the in-position range. | 1 to 65,535 <br> Setting 0 is <br> the same <br> as setting <br> 1. | $0: 1$ | When using the <br> inverter's servo lock, the <br> command value to the <br> inverter is set to zero <br> during in-position status. | When CPU Unit <br> power is turned <br> ON |

## Minimum Output Value

| Setting | Description | Set value | Default | Application | Refresh timing |
| :--- | :--- | :--- | :--- | :--- | :--- |
| Min. <br> output <br> value | If the error counter present value times the gain <br> setting is less than the minimum output value, <br> the minimum output value will be output. <br> Set the minimum output value so that it is equal <br> to or smaller than the maximum output value. | 1 to 65,535 <br> Setting 0 is <br> the same <br> as setting <br> 1. | $0: 1$ | A minimum output <br> value can be set to <br> ensure an output of <br> a specified size even <br> when the error <br> counter present <br> value is very small. | When CPU Unit <br> power is turned <br> ON |

## Maximum Output Value

| Setting | Description | Set value | Default | Application | Refresh timing |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Max. output value | If the error counter present value times the gain setting is greater than the maximum output value, the maximum output value will be output. Set the maximum output value so that it is equal to or greater than the minimum output value. | $\begin{array}{\|l\|} \hline 1 \text { to } \\ 4,294,967,29 \\ 5 \\ \text { Setting } 0 \text { is } \\ \text { the same as } \\ \text { setting } \\ 2,000,000 \text {. } \end{array}$ | 0: 2,000,000 | A maximum output value can be set to prevent the output value from becoming too large. | When CPU Unit power is turned ON |

## Error Counter Overflow

Detection Value

| Setting | Description | Set value | Default | Application | Refresh timing |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Error counter overflow detection value | If the absolute value of the error counter present value is greater than the error counter overflow detection value, the Error Counter Error Flag (A26.03) will turn ON. | 1 to 32,767 <br> Setting 0 is the same as setting 10,000. | 0: 10,000 | Provides notification of excessive pulses in the error counter, e.g., when manually moving the motor shaft while positioning is stopped. | When CPU Unit power is turned ON |

## Error Counter Alarm

## Detection Value

| Setting | Description | Set value | Default | Application | Refresh timing |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Error counter alarm detection value | If the absolute value of the error counter present value is greater than the error counter alarm detection value, the Error Counter Alarm Flag (A26.08) will turn ON. | 1 to 32,767 <br> Setting 0 is the same as setting 10,000. | 0: 10,000 | Provides notification of excessive pulses in the error counter, e.g., when encoder wiring breaks during positioning. | When CPU Unit power is turned ON |

## Error Counter Cycle

| Setting | Description | Set value | Default | Application | Refresh timing |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Error counter cycle | The calculation cycle of the error counter can be set. If the cycle is too short when using a motor with a slow response, pulses may easily accumulate in the error counter. Change the error counter cycle according to the machine load and motor response. <br> Note The setting is made in increments of 4 ms . The error counter cycle will thus be the set value times 4 ms . For example, if the set value is 10 , the error counter cycle will be 40 ms . | 1 to 255 (in 4ms increments) Setting 0 is the same as setting 3 (4-ms increments) | 0: 3 (4-ms increments) The error counter cycle will be 12 ms . | Set when using a motor with a slow response. | When CPU Unit power is turned ON |

## Power Supply Frequency

for One Motor Revolution
per Second

| Setting | Description | Set value | Default | Application | Refresh timing |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Power Supply Freq. for One Motor Revolution per Sec. | Calculate the power supply frequency for one revolution per second from the motor specifications. For example, with a $1,800-\mathrm{r} / \mathrm{min}(60-\mathrm{Hz})$ motor ( $30 \mathrm{r} / \mathrm{s}$ ), the power supply frequency for one revolution per second would be calculated as follows: $60[\mathrm{~Hz}] \div 30[\mathrm{r} / \mathrm{s}]=2[\mathrm{~Hz}]$. <br> Note The setting is made in increments of 0.1 Hz . The frequency will thus be the set value times 0.1 Hz . For example, if the set value is 20 , the frequency will be 2 Hz . | 0 to $65,535 \mathrm{~Hz}$ (0.1-Hz increments) | $0(0.1-\mathrm{Hz}$ <br> increments) | This setting is used when converting the output value to an inverter frequency command. | When CPU Unit power is turned ON |

Number of Encoder
Pulses for One Motor
Revolution

| Setting | Description | Set value | Default | Application | Refresh timing |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Number of Encoder Pulses for One Motor Revolution | Calculate the number of encoder pulses for one motor revolution from the encoder resolution (pulses/revolution), high-speed counter's multiplier, and motor-encoder shaft gear ratio. For example, if the encoder resolution is 1,000 , the highspeed counter multiplier is 4 , and the gear ratio is $1 / 4$, the number of encoder pulses for one motor revolution is $1,000 \times 4 \times(1 / 4)$ $=1,000$. | 0 to 65,535 | 0 | This setting is used when converting the output value to an inverter frequency command. | When CPU Unit power is turned ON |

## Operation Adjustment Settings

Use the following settings if the gain adjustment in the basic settings does not produce stable operation.

Limit Output during
Acceleration and
Constant Speed

| Setting | Description | Set value | Default | Application | Refresh timing |
| :--- | :--- | :--- | :--- | :--- | :--- |
| Limit output during <br> acceleration and <br> constant speed | Select this option to limit the upper and <br> lower values of the output value based on <br> the pulse output value during internal <br> pulse output acceleration or constant <br> speed. | Use/Do <br> not use | Do not <br> use | Use this setting <br> when positioning <br> precision is bad. | When CPU Unit <br> power is turned <br> ON |

## Limit Output during <br> Deceleration and When <br> Stopped

| Setting | Description | Set value | Default | Application | Refresh timing |
| :--- | :--- | :--- | :--- | :--- | :--- |
| Limit output during <br> deceleration and <br> when stopped | Select this option to multiply the error of <br> the output value by a coefficient during <br> internal pulse output deceleration or <br> after output has been completed. | Use/Do <br> not use | Do not <br> use | Use this setting <br> when positioning <br> precision is bad. | When CPU Unit <br> power is turned <br> ON |

Output Coefficient during
Acceleration and

## Constant Speed

| Setting | Description | Set value | Default | Application | Refresh timing |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Output coefficient during acceleration and constant speed | Upper and lower limits are placed on the output value by multiplying the pulse output value by a coefficient during internal pulse output acceleration or constant speed. <br> Output Upper Limit = Internal pulse output value + Internal pulse output value $\times$ Output coefficient <br> Output Lower Limit = Internal pulse output value - Internal pulse output value $\times$ Output coefficient <br> Note The setting is made in increments of 0.01 . The coefficient will thus be the set value times 0.01 . For example, if the set value is 10 , the coefficient will be 0.1 ms . | 1 to 255 (0.01 increments) <br> Setting 0 is the same as setting 6 (0.01 increments). | $0: 6(0.01$ <br> increments) | This coefficient can be used to restrict the output range to prevent excessive values, based on the internal pulse output value when the motor response is slow even if a large error is produced. | When CPU Unit power is turned ON |

## Output Coefficient during

Deceleration

| Setting | Description | Set value | Default | Application | Refresh timing |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Output coefficient during deceleration | The output value can be changed by multiplying the value in the error counter by a coefficient during deceleration of internal pulse output. <br> Output value = <br> Error $\times$ Error counter cycle <br> (s) $\times$ Gain $\times$ Coefficient <br> Note The setting is made in increments of 0.01 . The coefficient will thus be the set value times 0.01. For example, if the set value is 10 , the coefficient will be 0.1 ms . | 1 to 255 (0.01 increments) Setting 0 is the same as setting 96 (0.01 increments). | $\begin{array}{\|l\|} \hline 0: 96 \\ (0.01 \\ \text { incre- } \\ \text { ments }) \end{array}$ | This coefficient can be used to reduce the output value when the motor response is slow and the target position is exceeded when stopping. | When CPU Unit power is turned ON |

Output Coefficient after
Pulse Output

| Setting | Description | Set value | Default | Application | Refresh timing |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Output coefficient after pulse output | The output value can be changed by multiplying the value in the error counter by a coefficient after deceleration of internal pulse output. <br> Output value = <br> Error $\times$ Error counter cycle (s) <br> $\times$ Gain $\times$ Coefficient <br> Note The setting is made in increments of 0.01 . The coefficient will thus be the set value times 0.01. For example, if the set value is 10 , the coefficient will be 0.1 ms . | 1 to 255 (0.01 increments) Setting 0 is the same as setting 50 (0.01 increments). | $\begin{aligned} & \hline 0: 50 \\ & (0.01 \\ & \text { incre- } \\ & \text { ments }) \end{aligned}$ | This coefficient can be used to reduce the output value when it the value in the error counter is too large after completing internal pulse output. | When CPU Unit power is turned ON |

## 5-3-9 Automatic Calculation of Inverter Frequency Command Value

Set the Power Supply Frequency for One Motor Revolution per Second, Number of Encoder Pulses for One Motor Revolution, and Error Counter Cycle in the PLC Setup to automatically calculate the inverter frequency command value and store it in A23 for inverter positioning 0 and A33 for inverter positioning 1.

Note The inverter frequency command values are stored in A23 and A33 in increments of 0.01 Hz . Divide the value in A23 or A33 by 100 to obtain the value in hertz.
The values stored in A23 and A33 can be used in converting the output value to the frequency command value for the inverter. This value can be output to the inverter from the program using serial communications or an Analog Output Unit.

Note The following formula is used inside the PLC to automatically calculate the inverter frequency command value from the output value (i.e., the error counter present value multiplied by the gain). (The output value is stored in A20 and A21 for inverter positioning 0 and in A30 and A31 for inverter positioning 1.)


Note: The inverter frequency command value is stored in A23/A33 in increments of 0.01 Hz .
Note (1) Calculate the power supply frequency for one revolution per second from the motor specifications. For example, with a $1,800-\mathrm{r} / \mathrm{min}(60-\mathrm{Hz})$ motor $(30 \mathrm{r} / \mathrm{s})$, the power supply frequency for one revolution per second would be calculated as follows: $60[\mathrm{~Hz}] \div 30[\mathrm{r} / \mathrm{s}]=2[\mathrm{~Hz}]$.
(2) The encoder resolution times the counter multiplier times the gear ratio equals the number of pulses output by the encoder for one motor shaft revolution.

## Example of Calculating Conversion Factor

## Serial Communications



## Conditions

- Power Supply Frequency for One Motor Revolution per Second: 2 Hz (PLC Setup)
- Number of Encoder Pulses for One Motor Revolution: 1,000 (PLC Setup)
- Rotary encoder resolution: 1,000 pulses/revolution (encoder specification)
- High-speed counter multiplier: x4 (PLC Setup)
- Gear ratio between motor and encoder shafts: 1/4 (machine specification)
- Error Counter Cycle: 12 ms (PLC Setup)


## Calculation

The calculation performed inside the PLC is as shown below.


The command value calculated above is used in the Modbus-RTU command frame, adjusting for the frequency unit. (See note.)


Refer to 6-3-3 Modbus-RTU Easy Master Function and to the inverter manual for details on Modbus-RTU communications.

Note If the frequency command unit set in the inverter is 0.1 Hz , divide the command frequency in A23 or A33 by 10.

Analog Output

The following example is for the CP1W-DA041/CP1W-DA021.
The analog output resolution is 6,000 , so the command value calculated above is multiplied by 6,000 divided by the inverter's maximum output frequency.

| Stored analog |
| :--- |
| output value |$=$| Inverter frequency |
| :--- |
| command value (Hz) |$\quad \times \frac{6,000}{\text { Inverter's max. output frequency (Hz) }}$



Refer to 7-3 Analog Output Units for operating procedures for the Analog Output Unit.

## - Calculation Example

## Conditions

Inverter's maximum output frequency: 60 Hz

## Calculation

The stored analog output value is calculated as follows:

$$
\begin{aligned}
\begin{array}{l}
\text { Stored analog } \\
\text { output value }
\end{array} & =\left[\begin{array}{l}
\text { Auxiliary Area } \\
\text { A23/A33 } \\
\text { (Unit: } 0.01 \mathrm{~Hz})
\end{array}\right] \times \frac{1}{100} \times \frac{6,000}{60} \\
& =\left[\begin{array}{l}
\text { Auxiliary Area } \\
\text { A23/A33 } \\
\text { (Unit: } 0.01 \mathrm{~Hz})
\end{array}\right] \times 1
\end{aligned}
$$

## 5-3-10 Memory Allocations

## Built-in Input Area

| Input terminal block |  | Default | Pulse output origin searches enabled | Inverter positioning enabled |
| :---: | :---: | :---: | :---: | :---: |
| Word | Bit | Normal inputs | Origin search |  |
| CIO 0 (See note 1.) | 00 | Normal input 0 | --- | High-speed counter 0: Phase A |
|  | 01 | Normal input 1 | --- | High-speed counter 0: Phase B |
|  | 02 | Normal input 2 | Pulse output 0: Origin proximity input signal (CPU Units with 14 I/O (See note 3.)) | High-speed counter 1: Phase A |
|  | 03 | Normal input 3 | Pulse output 1: Origin proximity input signal (CPU Units with 14 I/O (See note 3.)) Pulse output 0: Origin proximity input signal (CPU Units with 10 I/O (See note 3.)) | High-speed counter 1: Phase B |
|  | 04 | Normal input 4 | --- | --- |
|  | 05 | Normal input 5 | Pulse output 0: Origin input signal (CPU Units with $10 \mathrm{I} / \mathrm{O}$ (See note 3.)) | --- |
|  | 06(See note 2.) | Normal input 6 | Pulse output 0: Origin input signal | --- |
|  | 07(See note 2.) | Normal input 7 | Pulse output 1: Origin input signal | --- |
|  | 08 (See note 2.) | Normal input 8 | --- | --- |
|  | 09 (See note 2.) | Normal input 9 | --- | --- |
|  | 10 (See note 2.) | Normal input 10 | Pulse output 0: Origin proximity input signal (CPU Units with 20, 30,40 or 60 I/O) | --- |
|  | 11 (See note 2.) | Normal input 11 | Pulse output 1: Origin proximity input signal (CPU Units with $20,30,40$ or $60 \mathrm{I} / \mathrm{O}$ ) | --- |

Note (1) The above table shows only allocations related to inverter positioning.
(2) Bits 08 to 11 are not supported by CPU Units with 14 I/O Points. Bits 06 to 11 are not supported by CPU Units with 10 I/O Points.
(3) If inverter positioning 1 is used with a CPU Unit with 14 I/O Points, origin searches (i.e., the origin proximity input signal) cannot be used.

## Built-in Output Area

This area is not used for inverter positioning.
When inverter positioning is enabled, bits 00 to 03 in CIO 100 can be used as normal outputs 0 to 3 . The corresponding pulse output and PWM output cannot be used.

## Auxiliary Area

## Read Area

## ■ Inverter Positioning 0

Use one of the following for the inverter frequency command.

| Word | Bits | Function | Data range | Refresh timing | Application examples |
| :---: | :---: | :---: | :---: | :---: | :---: |
| A20 | 00 to 15 | Lower 4 digits of present value of unsigned output value (output value = present value of error counter $\times$ error counter cycle (s) $\times$ gain) <br> Note The maximum and minimum output values are applied. | $\begin{aligned} & 00000000 \text { to } 8000 \\ & 0000 \text { hex } \\ & (0 \text { to } \\ & 2,147,483,648) \end{aligned}$ | Cleared to zero at following times: <br> - When power to CPU Unit is turned ON <br> - At start of operation <br> - When an error counter error occurs <br> Updated at following times: <br> - Cyclically according to error counter cycle | This value can be used when not using automatic frequency command calculations and instead to convert the output value provided here in the user program for output to the inverter. <br> This value is used when signed data is not required, i.e., when using communications or normal I/O to specify the direction. |
| A21 | 00 to 15 | Upper 4 digits of present value of unsigned output value (output value = present value of error counter $\times$ error counter cycle (s) $\times$ gain) <br> Note The maximum and minimum output values are applied. |  |  |  |


| Word | Bits | Function | Data range | Refresh timing | Application examples |
| :---: | :---: | :---: | :---: | :---: | :---: |
| A23 | 00 to 15 | Inverter frequency command value ( $0.01-\mathrm{Hz}$ increments, unsigned) <br> Note Set the Power Supply Frequency for One Motor Revolution per Second, Number of Encoder Pulses for One Motor Revolution, and Error Counter Cycle in the PLC Setup before using this value. | $\begin{aligned} & 0000 \text { to FFFF hex } \\ & (0.00 \text { to } \\ & 655.35 \mathrm{~Hz}) \end{aligned}$ | Cleared to zero at following times: <br> - When power to CPU Unit is turned ON <br> - At start of operation <br> - When an error counter error occurs Updated at following times: <br> Cyclically according to error counter cycle | These words contain the automatically calculated frequency command value for the inverter. (This value is normally used.) <br> For example, if the frequency setting unit of the inverter is 0.01 Hz , this value can be used as it in serial communications with the inverter. When converting to an analog output (0 to $5 \mathrm{~V}, 1$ to $5 \mathrm{~V}, 0$ to 10 $\mathrm{V}, 0$ to 20 mA , or 4 to 20 mA ), this value can be used to simplify the conversion. <br> This value is used when signed data is not required, i.e., when using communications or normal I/O to specify the direction. |
| A24 | 00 to 15 | Lower 4 digits of present value of signed output value (output value = present value of error counter $\times$ error counter cycle (s) $\times$ gain) <br> Note The maximum and minimum output values are applied. | 80000000 to 7FFF FFFF hex (-214,748,348 to 214,748,347) | Cleared to zero at following times: <br> - When power to CPU Unit is turned ON <br> - At start of operation <br> - When an error counter error occurs <br> Updated at following times: <br> - Cyclically according to error counter cycle | This value can be used when not using automatic frequency command calculations and instead to convert the output value provided here in the user program for output to the inverter. <br> This value is used when signed data |
| A25 | 00 to 15 | Upper 4 digits of present value of signed output value (output value = present value of error counter $\times$ error counter cycle (s) $\times$ gain) <br> Note The maximum and minimum output values are applied. |  |  | when outputting the frequency command with an analog output from -10 to 10 V . |

Use the following for inverter positioning status and the workpiece position.

| Word | Bits | Function | Data range | Refresh timing | Application examples |
| :---: | :---: | :---: | :---: | :---: | :---: |
| A26 | 00 | Operation Command Flag | ON: Operation command executed. <br> OFF: Stop command executed. | Turned ON at following times: <br> - When inverter positioning is started <br> Turned OFF at following times: <br> - When power to CPU Unit is turned ON <br> - At start of operation <br> - When CPU Unit operation stops <br> - When inverter positioning is stopped using INI instruction | This flag is used as a NO input condition when calculating the frequency command value in the user program. It is also used as a NC input condition when clearing the frequency command value to zero. |
|  | 01 | Forward Operation Command Flag | ON: Forward command in progress OFF: Reverse command in progress or stopped | Turned ON at following times: <br> - When error counter present value is greater than 0 (i.e., positive) <br> Turned OFF at following times: <br> - When error counter present value is less than 0 (i.e., negative) or zero <br> - When power to CPU Unit is turned ON <br> - When CPU Unit operation starts <br> - When CPU Unit operation stops | This flag is used as a NO input condition when outputting a forward operation command to the inverter from the user program It is also used as a NC input condition when outputting a reverse command to the inverter. |
|  | 02 | Reverse Operation Command Flag | ON: Reverse command in progress OFF: Forward command in progress or stopped | Turned ON at following times: <br> - When error counter present value is less than 0 (i.e., negative) <br> Turned OFF at following times: <br> - When error counter present value is greater than 0 (i.e., positive) or zero <br> - When power to CPU Unit is turned ON <br> - When CPU Unit operation starts <br> - When CPU Unit operation stops | This flag is used as a NO input condition when outputting a reverse operation command to the inverter from the user program It is also used as a NC input condition when outputting a forward command to the inverter. |


| Word | Bits | Function | Data range | Refresh timing | Application examples |
| :---: | :---: | :---: | :---: | :---: | :---: |
| A26 | 03 | In-position Flag | ON: In position OFF: Not in position | Turned ON at following times: <br> - When pulse output to error counter is stopped and absolute value of error counter present value is less than inposition range <br> Turned OFF at following times: <br> - When pulses are being output to error counter <br> - When absolute value of error counter present value is greater than in-position range. <br> - When power to CPU Unit is turned ON <br> - When CPU Unit operation starts <br> - When CPU Unit operation stops | This flag is used as an NO condition when clearing the frequency command value to zero from the user program. |
|  | 04 | Error Counter Error Flag | ON: Error counter error <br> OFF: No error | Turned ON at following times: <br> - When pulse output to error counter is stopped and absolute value of error counter present value is greater than or equal to error counter error detection value <br> Turned OFF at following times: <br> - When error counter error is reset <br> - When power to CPU Unit is turned ON <br> - When CPU Unit operation starts <br> - When CPU Unit operation stops | This flag can be used to provide notification of excessive pulses in the error counter, e.g., when manually moving the motor shaft while positioning is stopped. |
|  | 05 | Error Counter Pulse Output Flag | ON: Pulses being output <br> OFF: Pulse output stopped | Turned ON at following times: <br> - When pulse output to error counter is started <br> Turned OFF at following times: <br> - When pulse output to error counter is stopped (including immediate stops and deceleration stops) <br> - When power to CPU Unit is turned ON <br> - When CPU Unit operation starts <br> - When CPU Unit operation stops | This flag is used to determine whether pulses are being output to the error counter. <br> This flag can be used to determine when internal pulse output has been completed and start the next instruction. |
|  | 06 | Error Counter Pulse Output Acceleration/ Deceleration Flag | ON: Pulse output to the error counter is accelerating or decelerating (i.e., the frequency is changing) <br> OFF: Pulse output to the error counter is constant | Turned ON at following times: <br> - When pulse output frequency to error counter is changed by ACC or PLS2 instruction <br> Turned OFF at following times: <br> - During output of a constant pulse frequency to error counter <br> - When pulse output to error counter is stopped (including immediate stops and deceleration stops) <br> - When power to CPU Unit is turned ON <br> - When CPU Unit operation starts <br> - When CPU Unit operation stops | This flag is used to detect changes in the output frequency when the frequency is changed stepwise for internal pulses are being output by the ACC or PLS2 instruction. It can be used as a condition for executing ACC or PLS2 during internal pulse output. |


| Word | Bits | Function | Data range | Refresh timing <br> Application <br> examples |  |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| A26 | 07 | Error Counter Alarm <br> Flag | ON: Error counter <br> alarm <br> OFF: No error <br> counter alarm | Turned ON at following times: <br> $\bullet$ When pulse output to error counter is <br> stopped and absolute value of error <br> counter present value is greater than <br> or equal to error counter alarm detec- <br> tion value <br> Turned OFF at following times: <br> - When error counter alarm is reset <br> - When power to CPU Unit is turned ON | This flag can be <br> used to provide <br> notification of <br> excessive pulses in <br> the error counter, <br> e.g., when encoder <br> wiring breaks dur- <br> ing positioning. |

Use the following for the present values of the internal pulse and error counter of inverter positioning.

| Word | Bits | Function | Data range | Refresh timing | Application examples |
| :---: | :---: | :---: | :---: | :---: | :---: |
| A22 | 00 to 15 | Error counter 0 present value (signed) | $\begin{aligned} & 8000 \text { to 7FFF hex } \\ & (-32,768 \text { to } \\ & 32,767) \end{aligned}$ | Cleared to zero at following times: <br> - When power to CPU Unit is turned ON <br> - At start of operation <br> - When an error counter error occurs Updated at following times: <br> - Cyclically according to error counter cycle <br> Held at following times: <br> - When Error Counter Disable Bit (A562.01) is turned ON. | Use to monitor the difference between the target value and the present value. |
| A28 | 00 to 15 | Lower 4 digits of present value of pulse output to inverter (relative value) | 80000000 to 7FFF FFFF hex$(-2,147,483,648 \text { to }$$2,147,483,647)$ | Contains relative internal pulse output value when pulses are output to error counter. <br> Cleared to zero at following times: <br> - When power to CPU Unit is turned ON <br> - When operation is started <br> - When pulse output to error counter is started <br> Updated at following times: <br> - Cyclically on error counter cycle | These values can be used to monitor the present value of internal pulse output. |
| A29 | 00 to 15 | Upper 4 digits of present value of pulse output to inverter (relative value) |  |  |  |


| Word | Bits | Function | Data range | Refresh timing | Application examples |
| :---: | :---: | :---: | :---: | :---: | :---: |
| A276 | 00 to 15 | Lower 4 digits of the present value of the internal pulse output (absolute value for absolute coordinates) | 80000000 to 7FFF FFFF hex$(-2,147,483,648 \text { to }$$2,147,483,647)$ | Contains absolute movement value from the internal pulse origin when pulses are output to error counter. <br> Cleared to zero at following times: <br> - When power to CPU Unit is turned ON <br> - When operation is started <br> Updated at following times: <br> - Cyclically on error counter cycle | This value can be used to monitor the present value of the internal pulse output as an absolute value when using absolute coordinates. |
| A277 | 00 to 15 | Upper 4 digits of the present value of the internal pulse output (absolute value for absolute coordinates) |  |  |  |

$■$ Inverter Positioning 1
Use one of the following for the inverter frequency command.

| Word | Bits | Function | Data range | Refresh timing | Application examples |
| :---: | :---: | :---: | :---: | :---: | :---: |
| A30 | 00 to 15 | Lower 4 digits of present value of unsigned output value (output value $=$ present value of error counter $\times$ error counter cycle (s) $\times$ gain) <br> Note The maximum and minimum output values are applied. | $\begin{aligned} & 00000000 \text { to } 8000 \\ & 0000 \text { hex } \\ & \text { (0 to } \\ & 2,147,483,648) \end{aligned}$ | Cleared to zero at following times: <br> - When power to CPU Unit is turned ON <br> - At start of operation <br> - When an error counter error occurs <br> Updated at following times: <br> - Cyclically according to error counter cycle | This value can be used when not using automatic frequency command calculations and instead to convert the output value provided here in the user program for output to the inverter. <br> This value is used when signed data |
| A31 | 00 to 15 | Upper 4 digits of present value of unsigned output value (output value = present value of error counter $\times$ error counter cycle (s) $\times$ gain) <br> Note The maximum and minimum output values are applied. |  |  | when using communications or normal I/O to specify the direction. |


| Word | Bits | Function | Data range | Refresh timing | Application examples |
| :---: | :---: | :---: | :---: | :---: | :---: |
| A33 | 00 to 15 | Inverter frequency command value (0.01-Hz increments, unsigned) <br> Note Set the Power Supply Frequency for One Motor Revolution per Second, Number of Encoder Pulses for One Motor Revolution, and Error Counter Cycle in the PLC Setup before using this value. | 0000 to FFFF hex (0.00 to $655.35 \mathrm{~Hz})$ | Cleared to zero at following times: <br> - When power to CPU Unit is turned ON <br> - At start of operation <br> - When an error counter error occurs Updated at following times: <br> Cyclically according to error counter cycle | These words contain the automatically calculated frequency command value for the inverter. (This value is normally used.) For example, if the frequency setting unit of the inverter is 0.01 Hz , this value can be used as it in serial communications with the inverter. When converting to an analog output (0 to $5 \mathrm{~V}, 1$ to $5 \mathrm{~V}, 0$ to 10 V , 0 to 20 mA , or 4 to 20 mA ), this value can be used to simplify the conversion. <br> This value is used when signed data is not required, i.e., when using communications or normal I/O to specify the direction. |
| A34 | 00 to 15 | Lower 4 digits of present value of signed output value (output value = present value of error counter $\times$ error counter cycle (s) $\times$ gain) <br> Note The maximum and minimum output values are applied. | $\begin{array}{\|l} 80000000 \text { to } 7 \text { FFF } \\ \text { FFFF hex } \\ (-214,748,348 \text { to } \\ 214,748,347) \end{array}$ | Cleared to zero at following times: <br> - When power to CPU Unit is turned ON <br> - At start of operation <br> - When an error counter error occurs Updated at following times: <br> - Cyclically according to error counter cycle | This value can be used when not using automatic frequency command calculations and instead to convert the output value provided here in the user program for output to the inverter. <br> This value is used when signed data |
| A35 | 00 to 15 | Upper 4 digits of present value of signed output value (output value = present value of error counter $\times$ error counter cycle (s) $\times$ gain) <br> Note The maximum and minimum output values are applied. |  |  | is required, i.e., when outputting the frequency command with an analog output from -10 to 10 V . |

Use the following for inverter positioning status and the workpiece position.

| Word | Bits | Function | Data range | Refresh timing | Application examples |
| :---: | :---: | :---: | :---: | :---: | :---: |
| A36 | 00 | Operation Command Flag | ON: Operation command executed. <br> OFF: Stop command executed. | Turned ON at following times: <br> - When inverter positioning is started <br> Turned OFF at following times: <br> - When power to CPU Unit is turned ON <br> - At start of operation <br> - When CPU Unit operation stops <br> - When inverter positioning is stopped using INI instruction | This flag is used as a NO input condition when calculating the frequency command value in the user program. It is also used as a NC input condition when clearing the frequency command value to zero. |
|  | 01 | Forward Operation Command Flag | ON: Forward command in progress OFF: Reverse command in progress or stopped | Turned ON at following times: <br> - When error counter present value is greater than 0 (i.e., positive) <br> Turned OFF at following times: <br> - When error counter present value is less than 0 (i.e., negative) or zero <br> - When power to CPU Unit is turned ON <br> - When CPU Unit operation starts <br> - When CPU Unit operation stops | This flag is used as a NO input condition when outputting a forward operation command to the inverter from the user program It is also used as a NC input condition when outputting a reverse command to the inverter. |
|  | 02 | Reverse Operation Command Flag | ON: Reverse command in progress OFF: Forward command in progress or stopped | Turned ON at following times: <br> - When error counter present value is less than 0 (i.e., negative) <br> Turned OFF at following times: <br> - When error counter present value is greater than 0 (i.e., positive) or zero <br> - When power to CPU Unit is turned ON <br> - When CPU Unit operation starts <br> - When CPU Unit operation stops | This flag is used as a NO input condition when outputting a reverse operation command to the inverter from the user program It is also used as a NC input condition when outputting a forward command to the inverter. |


| Word | Bits | Function | Data range | Refresh timing | Application examples |
| :---: | :---: | :---: | :---: | :---: | :---: |
| A36 | 03 | In-position Flag | ON: In position OFF: Not in position | Turned ON at following times: <br> - When pulse output to error counter is stopped and absolute value of error counter present value is less than inposition range <br> Turned OFF at following times: <br> - When pulses are being output to error counter <br> - When absolute value of error counter present value is greater than in-position range. <br> - When power to CPU Unit is turned ON <br> - When CPU Unit operation starts <br> - When CPU Unit operation stops | This flag is used as an NO condition when clearing the frequency command value to zero from the user program. |
|  | 04 | Error Counter Error Flag | ON: Error counter error <br> OFF: No error | Turned ON at following times: <br> - When pulse output to error counter is stopped and absolute value of error counter present value is greater than or equal to error counter error detection value <br> Turned OFF at following times: <br> - When error counter error is reset <br> - When power to CPU Unit is turned ON <br> - When CPU Unit operation starts <br> - When CPU Unit operation stops | This flag can be used to provide notification of excessive pulses in the error counter, e.g., when manually moving the motor shaft while positioning is stopped. |
|  | 05 | Error Counter Pulse Output Flag | ON: Pulses being output <br> OFF: Pulse output stopped | Turned ON at following times: <br> - When pulse output to error counter is started <br> Turned OFF at following times: <br> - When pulse output to error counter is stopped (including immediate stops and deceleration stops) <br> - When power to CPU Unit is turned ON <br> - When CPU Unit operation starts <br> - When CPU Unit operation stops | This flag is used to determine whether pulses are being output to the error counter. <br> This flag can be used to determine when internal pulse output has been completed and start the next instruction. |
|  | 06 | Error Counter Pulse Output Acceleration/ Deceleration Flag | ON: Pulse output to the error counter is accelerating or decelerating (i.e., the frequency is changing) <br> OFF: Pulse output to the error counter is constant | Turned ON at following times: <br> - When pulse output frequency to error counter is changed by ACC or PLS2 instruction <br> Turned OFF at following times: <br> - During output of a constant pulse frequency to error counter <br> - When pulse output to error counter is stopped (including immediate stops and deceleration stops) <br> - When power to CPU Unit is turned ON <br> - When CPU Unit operation starts <br> - When CPU Unit operation stops | This flag is used to detect changes in the output frequency when the frequency is changed stepwise for internal pulses are being output by the ACC or PLS2 instruction. It can be used as a condition for executing ACC or PLS2 during internal pulse output. |


| Word | Bits | Function | Data range | Refresh timing | Application examples |
| :---: | :---: | :---: | :---: | :---: | :---: |
| A36 | 07 | Error Counter Alarm Flag | ON: Error counter alarm OFF: No error counter alarm | Turned ON at following times: <br> - When pulse output to error counter is stopped and absolute value of error counter present value is greater than or equal to error counter alarm detection value <br> Turned OFF at following times: <br> - When error counter alarm is reset <br> - When power to CPU Unit is turned ON <br> - When CPU Unit operation starts <br> - When CPU Unit operation stops | This flag can be used to provide notification of excessive pulses in the error counter, e.g., when encoder wiring breaks during positioning. |
|  | 08 to 14 | Not used. |  |  |  |
|  | 15 | Inverter Positioning Output Value Sign Flag | ON: OFF: | Turned ON at following times: <br> - When signed output value is between 00000000 and 7FFF FFFF hex. <br> Turned OFF at following times: <br> - When signed output value is between FFFF FFFF and 80000000 hex. | This flag can be used as a direction signal. |
| A272 | 00 to 15 | Lower 4 digits of the present value of the internal pulse output (absolute value for absolute coordinates) | $\begin{array}{\|l} \hline 80000000 \text { to } 7 \text { FFF } \\ \text { FFFF hex } \\ (-2,147,483,648 \text { to } \\ 2,147,483,647) \end{array}$ | Contains absolute movement value when pulses are output to error counter. <br> Cleared to zero at following times: <br> - When power to CPU Unit is turned ON <br> - When operation is started <br> Updated at following times: <br> - Cyclically on error counter cycle | This value can be used to monitor the present value of the internal pulse output as an absolute value when using absolute coordinates. |
| A273 | 00 to 15 | Upper 4 digits of the present value of the internal pulse output (absolute value for absolute coordinates) |  |  |  |

Use the following for the present values of the internal pulse and error counter of inverter positioning.

| Word | Bits | Function | Data range | Refresh timing | Application examples |
| :---: | :---: | :---: | :---: | :---: | :---: |
| A32 | 00 to 15 | Error counter 0 present value (signed) | $\begin{aligned} & 8000 \text { to 7FFF hex } \\ & (-32,768 \text { to } \\ & 32,767) \end{aligned}$ | Cleared to zero at following times: <br> - When power to CPU Unit is turned ON <br> - At start of operation <br> - When an error counter error occurs <br> Updated at following times: <br> - Cyclically according to error counter cycle <br> Saved at following times: <br> - When Error Counter Disable Bit (A562.01) is turned ON. | Use to monitor the difference between the target value and the present value. |
| A38 | 00 to 15 | Lower 4 digits of present value of pulse output to inverter (relative value) | $\begin{aligned} & 80000000 \text { to } 7 \text { FFF } \\ & \text { FFFF hex } \\ & (-2,147,483,648 \text { to } \\ & 2,147,483,647) \end{aligned}$ | Contains relative internal pulse output value when pulses are output to error counter. <br> Cleared to zero at following times: | These values can be used to monitor the present value of internal pulse output. |
| A39 | 00 to 15 | Upper 4 digits of present value of pulse output to inverter (relative value) |  | - When operation is started <br> - When pulse output to error counter is started <br> Updated at following times: <br> - Cyclically on error counter cycle |  |


| Word | Bits | Function | Data range | Refresh timing | Application examples |
| :---: | :---: | :---: | :---: | :---: | :---: |
| A278 | 00 to 15 | Lower 4 digits of the present value of the internal pulse output (absolute value for absolute coordinates) | $\begin{aligned} & \hline 80000000 \text { to } 7 \text { FFF } \\ & \text { FFFF hex } \\ & (-2,147,483,648 \text { to } \\ & 2,147,483,647) \end{aligned}$ | Contains absolute movement value from the internal pulse origin when pulses are output to error counter. <br> Cleared to zero at following times: <br> - When power to CPU Unit is turned ON <br> - When operation is started <br> Updated at following times: <br> - Cyclically on error counter cycle | This value can be used to monitor the present value of the internal pulse output as an absolute value when using absolute coordinates. |
| A279 | 00 to 15 | Upper 4 digits of the present value of the internal pulse output (absolute value for absolute coordinates) |  |  |  |

## Read/Write Area

| Word | Bits | Function |  | Data range | Refresh | Application |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| A562 | 00 | Inverter positioning 0 | Error Counter Reset Bit | Turned ON: Error counter 0 present value (A22) reset and Error Counter Error Flag cleared. | --- | Turn ON this bit to clear the error counter error status. |
|  | 01 |  | Error Counter Disable Bit | While ON: Error counter value held. | --- | Turn ON this bit, for example, to disable accumulating pulses in the error counter when stopping positioning and moving the motor shaft manually. |
|  | 02 to 15 | Not used. |  |  |  |  |
| A563 | 00 | Inverter positioning 1 | Error Counter Reset Bit | Turned ON: Error counter 0 present value (A32) reset and Error Counter Error Flag cleared. | --- | Turn ON this bit to clear the error counter error status. |
|  | 01 |  | Error Counter Disable Bit | While ON: Error counter value held. | --- | Turn ON this bit, for example, to disable accumulating pulses in the error counter when stopping positioning and moving the motor shaft manually. |
|  | 02 to 15 | Not used. |  |  |  |  |

## Note Present Values of High-speed Counter and Pulse Outputs

The present value of the high-speed counter when inverter positioning is used is stored in the same memory location as for normal high-speed counter application. This value can be used as the present value of feedback pulses from the encoder, i.e., as the absolute position of inverter positioning. Target value and range comparisons for high-speed counters are also valid.
The present value of the pulse output (A276/A277 or A278/A279), i.e., the pulse output value to the error counter, is an absolute position if an absolute coordinate system is specified and is a relative position if a relative coordinate system is specified.


## 5-3-11 Application Example with Serial Communications

## Positioning with <br> Trapezoidal Control

Specifications and Operation

When start input CIO 1.04 turns ON, 600,000 pulses are output internally for inverter positioning 0 to turn the motor shaft.

Note Refer to 5-3-7 Determining the Internal Pulse Output Frequency for the formula to convert the frequency and use the converted internal pulse frequency. The number of output pulses is calculated from the encoder specifications and the high-speed counter multiplier.


## System Configuration



## Instructions Used PLS2(887)

## Terminal Allocations

■ Error Counter


■ RS-422A/485 Communications (CP1W-CIF11/CIF12)

| $O$ | $O$ | $O$ | $O$ | $O$ |
| :--- | :--- | :--- | :--- | :--- |
|  |  |  |  |  |
| RDA- | RDB + | SDA- | SDB + | $F G$ |


| SW |  |
| :---: | :--- |
| 1 | $\square$ |
| 2 | $\square$ |
| 3 | $\square$ |
| 4 | $\square$ |
| 5 | $\square$ |
| 6 | $\square$ |

■ Inverter (3G3MV)

|  |  | S 5 | S 6 | S 7 | P 1 | P 2 | $\mathrm{R}+$ | $\mathrm{R}-$ | FS | FR |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | FC .



■ Encoder


## Connection Example

## ■ Encoder (24 VDC) Connections to High-speed Counter 0



■ RS-422A/485 (CP1W-CIF11/CIF12) Connections to Inverter

| $O$ | $O$ | $\bigcirc$ | $\bigcirc$ | $\bigcirc$ |
| :--- | :--- | :--- | :--- | :--- |
| RS-422A/485 |  |  |  |  |
| RDA- | RDB $+\mid$ | SDA- | SDB + | FG |

CP1W-CIF11/CIF12


Inverter


## ■ Inverter Connections to Motor



Parameter Settings
for 3G3MV Inverter

When connecting the Inverter to the PLC, communications parameters must be set in the Inverter. The settings of parameters n 152 to n 157 cannot be changed while communications are in progress. Always set them before starting communications.

Example settings of 3G3MV parameters are listed below. Refer to the User's Manual of the Inverter for details on the parameters.

| Parameter No. | Name | Description | Default | Setting |
| :---: | :---: | :---: | :---: | :---: |
| n003 | RUN command selection | 0: The RUN Key and STOP/RESET Key on the Digital Operator are enabled. <br> 1: Multi-function input is enabled through the control circuit terminals. <br> 2: RS-422A/485 communications are enabled. <br> 3: Input is enabled from the optional Communications Unit. | 0 | 2 |
| n004 | Frequency reference selection | 0: Frequency reference adjustment <br> 1: Frequency reference 1 (n024) <br> 2: Frequency reference control terminal (0 to 10 V) <br> 3: Frequency reference control terminal (4 to 20 mA) <br> 4: Frequency reference control terminal (0 to 20 mA ) <br> 5: Pulse train reference control terminal <br> 6: Frequency reference through RS-422A/RS-485 <br> 7: Multi-function analog voltage input ( 0 to 10 V ) <br> 8: Multi-function analog current input (4 to 20 mA ) <br> 9: Frequency reference input through optional Communications Unit. | 0 | 6 |
| n005 | Stopping method selection | 0: Decelerates to stop <br> 1: Coasts to stop | 0 | 0 |
| n006 | Reverse rotation-prohibit selection | 0: Reverse enabled <br> 1: Reverse disabled | 0 | 0 |
| n011 | Maximum frequency (FMAX) | 50.0 to 400.0 Hz (0.1-Hz increments) | 60.0 Hz | 60.0 Hz (Depends on machine configuration.) |
| n016 | Minimum output frequency (FMIN) | 0.1 Hz to 10.0 Hz (0.1-Hz increments) | 1.5 Hz | 0.1 Hz |
| n018 | Acceleration/deceleration time setting unit | $\begin{aligned} & \hline 0: 0.1 \mathrm{~s} \\ & 1: 0.01 \mathrm{~s} \end{aligned}$ | 0 | 0 |
| n019 | Acceleration time 1 | 0 to 6,000 s | 10.0 s | 0 |
| n020 | Deceleration time 1 | 0 to 6,000 s | 10.0 s | 0 |
| n151 | RS-422A/485 communications timeover detection selection (The time between receiving PLC signals is monitored, Timeout time: 2 s .) | 0: Detects time-over, fatal error, and the Inverter coasts to a stop. <br> 1: Detects time-over, detects fatal error, and the Inverter decelerates to a stop in deceleration time 1. <br> 2: Detects time-over, detects fatal error, and the Inverter decelerates to a stop in deceleration time 2. <br> 3: Detects time-over, detects nonfatal error warning, and the Inverter continues operating. <br> 4: No time-over is detected. | 0 | 0 |
| n152 | RS-422A/485 communications frequency reference/display unit selection | $\begin{aligned} & \text { 0: } 0.1 \mathrm{~Hz} \\ & \text { 1: } 0.01 \mathrm{~Hz} \end{aligned}$ <br> 2: Converted value based on 30,000 decimal as maximum frequency <br> 3: 0.1\% (Maximum frequency: 100\%) | 0 | 1 |


| Parameter <br> No. | Name | Description | Default | Setting |
| :--- | :--- | :--- | :--- | :--- |
| n153 | RS-422A/485 communications <br> Slave address | Setting range: 0 to 32 <br> 00: Communications disabled <br> 01 to 32: Slave address | 0 | 1 |
| n154 | RS-422A/485 baud rate selec- <br> tion | 0: 2,400 bps <br> $1: 4,800$ bps <br> $2: 9,600$ bps <br> $3: 19,200$ bps | 2 | 2 |
| n155 | RS-422A/485 parity selection | 0: Even <br> $1:$ Odd <br> 2: No parity | 0 | 0 |
| n156 | RS-422A/485 send wait time | Set value: 10 to 65 ms <br> Setting unit: 1 ms | 10 ms | 10 ms |
| n157 | RS-422A/485 RTS control <br> selection | 0: RTS control enabled <br> 1: RTS control disabled | 0 | 0 |

## PLC Setup

## ■ Serial Port Communications Settings



Note (1) Set the baud rate and parity check settings to the same value as for the Inverter communications parameters.
(2) Set the serial port to the serial gateway communications mode.

## ■ High-speed Counter Settings (on Built-in Input Tab Page)



Note (1) Set high-speed counter 0 when using inverter positioning 0 . Set highspeed counter 1 when using inverter positioning 1.
(2) Use linear mode for inverter positioning.

■ Inverter Positioning Settings (on Inverter Positioning 0 or 1 Tab Page)


Ladder Program
The following Modbus-RTU communications parameters are used.

| Baud rate | 9,600 bits/s |
| :--- | :--- |
| Format | $8,1, \mathrm{E}$ |
| Serial communications mode | Serial Gateway |

## Starting Inverter

Serial port 1 is used for communications with the Inverter.

## Positioning

Stopping Internal Pulse Output to the Error Counter

| 0.05 |  |  |  |
| :---: | :---: | :---: | :---: |
|  | @PLS2(887) |  |  |
| Start input | \#0020 $\leftarrow$ | $\leftarrow$ Inverter positioning 1 <br> $\leftarrow \mathrm{CW}$, relative pulses <br> $\leftarrow$ Target frequency, No. of output pulses <br> $\leftarrow$ Starting frequency |  |
|  | \#0000 $\leftarrow$ |  |  |
|  | D200 $\leftarrow$ |  |  |
|  | D300 $\leftarrow$ |  |  |
| PLS2(887) Settings |  |  |  |
| Setting details |  | Address | Data |
| Acceleration rate: $100 \mathrm{~Hz} / 4 \mathrm{~ms}$ |  | D200 | 0064 |
| Deceleration rate: $80 \mathrm{~Hz} / 4 \mathrm{~ms}$ |  | D201 | 0050 |
| Target frequency: 20,000 Hz |  | D202 | 4E20 |
|  |  | D203 | 0000 |
| Number of output pulses: 600,000 pulses |  | D204 | 27C0 |
|  |  | D205 | 0009 |
| Starting frequency: 100 Hz |  | D300 | 0064 |
|  |  | D301 | 0000 |

- High-speed counter 0 (i.e., error counter 0 ) is used for the feedback pulse input port.

- Internal pulse output is stopped immediately.
- Inverter positioning (i.e., the error counter) will continue to function.



## Stopping Inverter

 Positioning

- Internal pulse output is stopped immediately.
- The output value will remain at 0 until the error counter is reset.
- Pulse outputs will not be accepted until the error counter is reset. (Execut-

Referencing the Automatically Calculated Inverter Frequency Command Value
ing a pulse output instruction will cause an error.)


If the following settings are made in the PLC Setup, the inverter frequency command value will be calculated automatically and set in A23 in the Auxiliary Area. These settings are on the Inverter Positioning 0 Tab Page in the PLC Setup.

- Power Supply Frequency for One Motor Revolution per Second ( $0.1-\mathrm{Hz}$ increments)
- Number of Encoder Pulses for One Motor Revolution
- Error Counter Cycle (x 4 ms)

The inverter frequency command value in A23 is accessed. The value is stored in $0.01-\mathrm{Hz}$ increments.


■ Internal Work Addresses

| Address | Usage |
| :--- | :--- |
| D1 | Bits 00 to 03: Run/Stop Command |
| D2 | Bits 00 to 15: Frequency Command Value |
| D10 | Bits 00 to 03: Forward/Reverse Command |

## Setting Modbus

Communications

## Registers

| A641.00 |  | Slave address: 01 hex |
| :---: | :---: | :---: |
| $\square$ | MOV |  |
| Modbus simple master function not active | D32200 |  |
|  | MOV | Function code: <br> 10 hex (write data) |
|  | \#0010 |  |
|  | D32201 |  |
|  | MOV | Number of communications data bytes: 09 hex ( 9 bytes) |
|  | \#0009 |  |
|  | D32202 |  |
|  | MOV | Register number of write start data: 0001 hex |
|  | \#0001 |  |
|  | D32203 |  |
|  | MOV | Number of write data registers: 0002 hex |
|  | \#0002 |  |
|  | D32204 |  |
|  | MOV | Number of attached data bytes: 04 hex ( 4 bytes) |
|  | \#0400 |  |
|  | D32205 |  |
|  | XFRB | Moves bits 00 to 03 of D1 (Run/Stop Command) to D32206 (register 0001) bits 08 to 11. |
|  | \#0480 |  |
|  | D1 |  |
|  | D32206 |  |
|  | XFRB | Moves bit 00 of D10 (Forward/Reverse Command) to bit 09 of D15. |
|  | \#0190 |  |
|  | D10 |  |
|  | D15 |  |
|  | ORW | ORs D15 and D32206 (register 0001) and stores the result in D32206. (Reflect bit 09 of D15 in D32206 (register 0001). |
|  | D15 |  |
|  | D32206 |  |
|  | D32206 |  |
|  | XFRB | Move bits 08 to 15 of D2 (frequency command value) to bits 00 to 07 of D32206 (register 0001). |
|  | \#0808 |  |
|  | D2 |  |
|  | D32206 |  |
|  | XFRB | Move bits 00 to 07 of D2 (frequency command value) to bits 08 to 15 of D32207 (register 0002). |
|  | \#0880 |  |
|  | D2 |  |
|  | D32207 |  |

## ■ Internal Work Addresses

| Address | Usage |
| :--- | :--- |
| D1 | Bits 00 to 03: Run/Stop Command |
| D2 | Bits 00 to 15: Frequency Command Value |
| D10 | Bits 00 to 03: Forward/Reverse Command |
| D15 | Bit 09: Forward/Reverse Command |

## ■ Settings Addresses

| Address | Usage | Data |
| :--- | :--- | :--- |
| D32200 | Bits 00 to 07: Slave address | 01 |
| D32201 | Bits 00 to 07: Function code | 10 |
| D32202 | Bits 00 to 07: Number of communications data bytes | 09 |
| D32203 | Bits 00 to 15: Register number of write start data | 0001 |
| D32204 | Bits 00 to 15: Number of data registers to write | 0002 |
| D32205 | Bits 08 to 15: Number of attached data bytes | 04 |
| D32206 | Bits 00 to 07: Upper bytes of frequency command value in D2 <br> Bit 08: Run/Stop Command <br> Bit 09: Forward/Reverse Command | --- |
| D32207 | Bits 08 to 15: Lower bytes of frequency command value in D2 | --- |

## Modbus Communications



Add the above instructions to the end of the program as a starting condition for the ladder programming example. For error processing, refer to the ladder program in 6-3-3 Modbus-RTU Easy Master Function and to the inverter's manual.

## 5-3-12 Application Example with an Analog Output

## Positioning with <br> Trapezoidal Control

## Specifications and Operation

When start input CIO 1.04 turns ON, 600,000 pulses are output internally for inverter positioning 0 to turn the motor shaft.

Note Refer to 5-3-7 Determining the Internal Pulse Output Frequency for the formula to convert the frequency and use the converted internal pulse frequency. The number of output pulses is calculated from the encoder specifications and the high-speed counter multiplier.


## System Configuration



## Instructions Used

PLS2(887)

## Terminal Allocations

■ Error Counter


## ■ Built-in Outputs

| + | 00 |  | 1 | 02 |  | 3 | 04 | 0 |  | 00 |  | 04 |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | COM COM |  |  | COM |  | COM |  | 05 | 07 | COM | 02 | COM | 05 | 07 |

■ CP1W/CPM1A-DA041


■ Inverter (3G3MV)

| S5 |  | S6 | S7 |  | P1 | P2 |  | R+ | R- |  | FS | FR | FC |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| S1 | S2 | S |  | S4 |  | SC | PC | S |  | S- | AM | AC | RP |




## ■ Encoder



## Connection Example

■ Encoder (24 VDC) Connections to High-speed Counter 0


■ Output Terminal Connections to Inverter


■ CP1W/CPM1A-DA041 (Current Output) Connections to Inverter


## ■ Inverter Connections to Motor



## Parameter Settings for 3G3MV Inverter

When connecting the Inverter to the PLC, communications parameters must be set in the Inverter.

Example settings of 3G3MV parameters are listed below. Refer to the User's Manual of the Inverter for details on the parameters.

| Parameter No. | Name | Description | Default | Setting |
| :---: | :---: | :---: | :---: | :---: |
| n003 | RUN command selection | 0: The RUN Key and STOP/RESET Key on the Digital Operator are enabled. <br> 1: Multi-function input is enabled through the control circuit terminals. <br> 2: RS-422A/485 communications are enabled. <br> 3: Input is enabled from the optional Communications Unit. | 0 | 1 |
| n004 | Frequency reference selection | 0: Digital Operator <br> 1: Frequency reference 1 (n024) <br> 2: Frequency reference control terminal ( 0 to 10 V ) <br> 3: Frequency reference control terminal (4 to 20 mA) <br> 4: Frequency reference control terminal (0 to 20 mA) <br> 5: Pulse train reference control terminal <br> 6: Frequency reference through RS-422A/RS-485 <br> 7: Multi-function analog voltage input ( 0 to 10 V ) <br> 8: Multi-function analog current input ( 4 to 20 mA ) <br> 9: Frequency reference input through optional Communications Unit. | 0 | 4 |
| n050 | Multi-function input 1 | 1 to 25 | 1 | 1 |
| n051 | Multi-function input 2 | 1 to 25 | 2 | 2 |
| n060 | Frequency reference gain | 0\% to 255\% (1\% increments) | 100\% | 100\% |
| n061 | Frequency reference bias | -100\% to 100\% (1\% increments) | 0\% | 0\% |
| n005 | Stopping method selection | 0: Decelerates to stop <br> 1: Coasts to stop | 0 | 0 |
| n006 | Reverse rotation-prohibit selection | 0: Reverse enabled <br> 1: Reverse disabled | 0 | 0 |
| n011 | Maximum frequency (FMAX) | 50.0 to 400.0 Hz (0.1-Hz increments) | 60.0 Hz | 60.0 Hz (Depends on machine configuration.) |
| n016 | Minimum output frequency (FMIN) | 0.1 Hz to 10.0 Hz (0.1-Hz increments) | 1.5 Hz | 0.1 Hz |
| n018 | Acceleration/deceleration time setting unit | $\begin{aligned} & \hline 0: 0.1 \mathrm{~s} \\ & 1: 0.01 \mathrm{~s} \end{aligned}$ | 0 | 0 |
| n019 | Acceleration time 1 | 0 to 6,000 s | 10.0 s | 0 |
| n020 | Deceleration time 1 | 0 to 6,000 s | 10.0 s | 0 |

## PLC Setup

## ■ High－speed Counter Settings（on Built－in Input Tab Page）



Note（1）Set high－speed counter 0 when using inverter positioning 0 ．Set high－ speed counter 1 when using inverter positioning 1.
（2）Use linear mode for inverter positioning．
■ Inverter Positioning Settings（on Inverter Positioning 0 or 1 Tab Page）

| PLC Settings－NewPLC1 $\quad$－ $\mid$ 可 x |  |  |  |
| :---: | :---: | :---: | :---: |
| File Options Help |  |  |  |
|  |  |  |  |
| －Inverter Positioning <br> Use inverter positioning． <br> Gain（default 0：10＂0．1） <br> In－position range（default 0：1） <br> Min．output value（default 0：1］ <br> Max．output value <br> （default 0：2000000） <br> Error counter overflow detection value <br> （default 0：10000） <br> Error counter alarm detection value （default 0：10000） <br> Error counter cycle（default 0： 12 ms ） <br> Power Supply Freq．for One Motor Revolution per Sec． <br> Number of Encoder Pulses for One Motor Revolution  $\square$ $\square$ $\square$ ${ }^{*} 0.1 \mathrm{~Hz}$ <br> Limit output during acceleration and constant speed． <br> Output coefficient during acceleration and constant speed <br> （default 0： $6^{\circ} 0.01$ ） $\square$ ${ }^{*} 0.01$ <br> Limit output during deceleration and when stopped． <br> Output coefficient during deceleration （default 0： $96 \times 0.01$ ） $\square$ ${ }^{*} 0.01$ <br> Output coefficient after pulse output （default 0： $50^{\circ} 0.01$ ） $\square$ ${ }^{*} 0.01$ |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Ladder Program

## Starting Inverter

 Positioning
## Stopping Internal Pulse

 Output to the Error Counter

- Internal pulse output is stopped immediately.
- Inverter positioning (i.e., the error counter) will continue to function.



## Stopping Inverter

 Positioning

- Internal pulse output is stopped immediately.
- The output value will remain at 0 until the error counter is reset.
- Pulse outputs will not be accepted until the error counter is reset. (Executing a pulse output instruction will cause an error.)


Referencing the Automatically Calculated Inverter Frequency Command Value

If the following settings are made in the PLC Setup, the inverter frequency command value will be calculated automatically and set in A23 in the Auxiliary Area. These settings are on the Inverter Positioning 0 Tab Page in the PLC Setup.

- Power Supply Frequency for One Motor Revolution per Second ( $0.1-\mathrm{Hz}$ increments)
- Number of Encoder Pulses for One Motor Revolution
- Error Counter Cycle (x 4 ms)

The inverter frequency command value in A23 is accessed and converted to an analog output signal. The CP1W/CPM1A-DA041 has a resolution of 6,000, so the conversion to an analog signal is performed as follows:
$6,000 \div 60 \mathrm{~Hz}$ (inverter's maximum output frequency) $\div 100=1$
The conditions are as follows:


In this example, the results of *U and /UL are 1, so the value in A23 is moved directly to D102 with MOV.

## ■ Internal Work Addresses

| Address | Usage |
| :--- | :--- |
| D1010 | Holds the frequency command value converted for the analog <br> output resolution. |
| D1011 | Holds the frequency command value converted from $0.01-\mathrm{Hz}$ <br> increments to hertz. |
| D1020 | Analog conversion trigger |
| T0 |  |

## ■ Settings Addresses

| Address | Usage |
| :--- | :--- |
| D102 | Bits 00 to 15: Analog output value |
| CIO 100.02 | Forward (external output) |
| CIO 100.03 | Reverse (external output) |

CP1W/CPM1A-DA041 Analog Output Settings

Analog output 1 is used in this example. It is set to a range of 4 to 20 mA . The scaled value is set in the analog conversion area of the Analog Output Unit.


Refer to the first line in the programming example in Referencing the Automatically Calculated Inverter Frequency Command Value on page 334 for a timer for the time required for the Analog Output Unit's data to be valid (analog conversion trigger: TO).

## ■ Internal Work Addresses

| Address | Usage |
| :--- | :--- |
| D102 | Bits 00 to 15: Analog output value |
| T0 | Analog conversion trigger |

## ■ Settings Addresses

| Address | Usage |
| :--- | :--- |
| CIO 102 | Bits 00 to 15: Analog conversion area |
| CIO 103 | Bits 00 to 15: Analog conversion area |

## 5-3-13 Supplemental Information

## Restrictions

## Precautions

- Inverter positioning 0 and inverter positioning 1 each use one high-speed counter and one serial port (except that a serial port is not used when an Analog Output Unit is used). (High-speed counter 0 is allocated to inverter positioning 0 and high-speed counter 1 is allocated to inverter positioning 1.)
- When inverter positioning 0 or 1 is used, the corresponding pulse output (0 or 1 ) and the corresponding PWM command (pulse output 0 or 1) cannot be used.
- Determine the in-position range based on the mechanical system. Use a smaller range if positioning precision is required. If the range is too small, however, time may be required when stopping. If stopping quickly is more important than precision, increase the in-position range.
- The error counter cycle also affects the conversion between the output value and the inverter frequency command value. Refer to 5-3-9 Automatic Calculation of Inverter Frequency Command Value for details.
- If inverter positioning does not end normally, adjust the following settings. Reduce the acceleration/deceleration rates. Lower rates will stabilize operation at the end of acceleration/deceleration.
Reduce the target frequency.
Change the error counter cycle. Increasing the error counter cycle improve stopping precision, but it may also cause unstable speeds during operation.
Adjust the gain.
Increasing the gain will improve stopping precision, but it may also cause unstable speeds during operation.


## SECTION 6 Advanced Functions

This section describes all of the advanced functions of the CP1L that can be used to achieve specific application needs.
6-1 Interrupt Functions ..... 338
6-1-1 Overview of CP1L Interrupt Functions ..... 338
6-1-2 Input Interrupts (Direct Mode) ..... 342
6-1-3 Input Interrupts (Counter Mode) ..... 347
6-1-4 Scheduled Interrupts ..... 350
6-1-5 High-speed Counter Interrupts ..... 353
6-2 Quick-response Inputs ..... 362
6-3 Serial Communications ..... 366
6-3-1 Overview ..... 366
6-3-2 No-protocol Communications ..... 368
6-3-3 Modbus-RTU Easy Master Function ..... 371
6-3-4 Communications: Smart Active Parts and Function Blocks ..... 374
6-3-5 Serial PLC Links ..... 377
6-3-6 1:1 Links ..... 386
6-3-7 1:N NT Links ..... 387
6-3-8 1:1 NT Links ..... 388
6-3-9 Host Link Communications ..... 389
6-4 Analog Adjuster and External Analog Setting Input ..... 394
6-4-1 Analog Adjuster ..... 394
6-4-2 External Analog Setting Input ..... 394
6-5 Battery-free Operation ..... 395
6-5-1 Overview ..... 395
6-5-2 Using Battery-free Operation ..... 396
6-6 Memory Cassette Functions ..... 397
6-6-1 Overview ..... 397
6-6-2 Mounting and Removing a Memory Cassette ..... 398
6-6-3 Operation Using the CX-Programmer ..... 400
6-6-4 Memory Cassette Data Transfer Function ..... 401
6-6-5 Procedures for Automatic Transfer from the Memory Cassette at Startup ..... 404
6-7 Program Protection ..... 405
6-7-1 Read Protection ..... 405
6-7-2 Write Protection ..... 411
6-7-3 Protecting Program Execution Using the Lot Number ..... 413
6-8 Failure Diagnosis Functions ..... 414
6-8-1 Failure Alarm Instructions: $\operatorname{FAL}(006)$ and FALS(007) ..... 414
6-8-2 Failure Point Detection: FPD(269) ..... 415
6-8-3 Simulating System Errors ..... 416
6-8-4 Output OFF Bit ..... 417
6-9 Clock ..... 418

## 6-1 Interrupt Functions

## 6-1-1 Overview of CP1L Interrupt Functions

The CP1L CPU Unit's processing is normally cyclical (overseeing processing $\rightarrow$ program execution $\rightarrow$ I/O refreshing $\rightarrow$ peripheral servicing), with cyclic tasks executed in the program execution stage of the cycle. The interrupt functions can be used to temporarily interrupt this cyclic processing and execute a particular program when a predefined condition occurs.

## Types of Interrupt <br> Functions

```
Input Interrupts (Direct Mode)
Input Interrupts (Counter
Mode)
```


## Scheduled Interrupts

High-speed Counter Interrupts

When one of the CPU Unit's built-in inputs goes from OFF to ON (or ON to OFF), the corresponding interrupt task is executed. Interrupt tasks 140 to 145 are allocated to the 8 input terminals used for the input interrupts.
This function counts input pulses at one of the CPU Unit's built-in inputs and executes the corresponding interrupt task when the count reaches the SV.
The maximum input response frequency for input interrupts (in counter mode) is 5 kHz .

This function executes an interrupt task at a fixed time interval measured by the CPU Unit's built-in timer. The time interval units can be set to $10 \mathrm{~ms}, 1 \mathrm{~ms}$, or 0.1 ms . The minimum timer SV is 0.5 ms .
Interrupt task 2 is allocated to scheduled interrupt.
This function counts input pulses with the CPU Unit's built-in high-speed counter and executes an interrupt task when the count reaches the preset value or falls within a preset range (target-value or zone comparison). An interrupt task between 0 and 255 can be allocated with an instruction.
Refer to 5-1 High-speed Counters for details on high-speed counters.
Note Power OFF interrupts cannot be used with CP1L CPU Units.

## Creating an Interrupt

Task Program
1,2,3... 1. Right-click NewPLC1 [CP1L] Offline in the project workspace and select Insert Program from the pop-up menu. A new program called NewProgram2 (unassigned) will be inserted in the project workspace.

2. Right-click NewProgram2 (unassigned) and select Properties from the pop-up menu to display the Program Properties Window.

3. Set the Task type in the Program Properties Window. In this example, interrupt task 140 was allocated to NewProgram2.


If you click the $\mathbf{X}$ Button in the upper-right corner of the window, you can create the program that will be executed as interrupt task 140.
The programs allocated to each task are independent and an END(001) instruction must be input at the end of each program.

## Interrupt Task Priority

## Duplicate Processing in Cyclic and Interrupt Tasks

The input interrupts (direct mode and counter mode), high-speed counter interrupts, scheduled interrupts, and external interrupts all have the same priority. If interrupt task A (an input interrupt, for example) is being executed when interrupt task B (a scheduled interrupt, for example) is called, task A processing will not be interrupted. Task B processing will be started when task A is completed.
If two different types of interrupt occur simultaneously, they are executed in the following order:

| Input interrupt <br> (direct mode or <br> counter mode) | $>$High-speed <br> counter inter- <br> rupt |
| :--- | :--- | :--- |

If two of the same type interrupt occur simultaneously, the task with the lower interrupt task number is executed first.

Note If a user program is likely to generate multiple interrupts simultaneously, the interrupt tasks will be executed in the order shown above, so it may take some time from the occurrence of the interrupt condition to the actual execution of the corresponding interrupt task. In particular, it is possible that scheduled interrupts will not be executed in the preset time, so the program must be designed to avoid interrupt conflicts if necessary.

If a memory address is processed both by a cyclic task and an interrupt task, an interrupt mask must be set to disable interrupts.

When an interrupt occurs, execution of the cyclic task will be interrupted immediately, even during execution of a cyclic task's instruction, and the partially processed data is saved. After the interrupt task is completed, processing returns to the cyclic task and the interrupted processing restarts with the data saved before the interrupt processing. If the interrupt task overwrites a memory address used by one of the interrupted instruction's operands, that overwrite may not be reflected after the saved data is restored as processing returns to the cyclic task.
To prevent an instruction from being interrupted during processing, enter $\mathrm{DI}(693)$ just before the instruction to disable interrupts and $\mathrm{El}(694)$ just after the instruction to enable interrupts again.
a. The following example shows duplicate processing by an interrupt task, which interrupts processing of $a+B$ instruction between the first and third operands and overwrites the same memory address.


Flow of Processing


The interrupt occurs during processing of the +B instruction and the result is saved temporarily without being written to the destination word (D0).
The interrupt task transfers the value of \#0010 to D0, but the saved result of the +B instruction (1235) is written to D0 when processing returns to the cyclic task. In the end, the interrupt task's processing has no effect.

Prevention of Duplicate Processing

b. The following example shows duplicate processing by an interrupt task, which interrupts processing while BSET is writing to a block of words and yields an incorrect comparison result.


Flow of Processing


Since the interrupt occurs during BSET(071) processing and before \#1234 is set in D10, the content of D0 and D10 do not match when the comparison is made in the interrupt task (*1) and output A remains OFF.
In the end (*2), the D0 and D10 both contain \#1234 and match, but the correct comparison result is not reflected in comparison result output $A$.

Prevention of Duplicate Processing


## 6-1-2 Input Interrupts (Direct Mode)

This function executes an interrupt task when the corresponding input signal (up or down differentiated) is received.

Input Interrupt Bit and Terminal Allocations

Input Terminal Block of CPU Units with 10 I/O Points

The following diagrams show the input bits and terminals that are used for the input interrupt function in each CPU Unit.

The 2 input bits CIO 0.04 to CIO 0.05 can be used for input interrupts.


The 4 input bits CIO 0.04 to CIO 0.07 can be used for input interrupts.


Input Terminal Block of CPU Units with 20 I/O Points

Input Terminal Block of CPU Units with 30 I/O Points

Input Terminal Block of CPU Units with 40 I/O Points

Input Terminal Block of CPU Units with 60 I/O Points

The 6 input bits CIO 0.04 to CIO 0.09 can be used for input interrupts.


The 6 input bits CIO 0.04 to CIO 0.09 can be used for input interrupts.


The 6 input bits CIO 0.04 to CIO 0.09 can be used for input interrupts.
Upper Terminal Block
(Example: CPU Unit
with AC Power Supply)


The 6 input bits CIO 0.04 to CIO 0.09 can be used for input interrupts.
Upper Terminal Block
(Example: AC Power
Supply Modules)


## Setting the Input Functions in the PLC Setup

Normally, bits CIO 0.04 to CIO 0.09 are used as normal inputs. When using these inputs for input interrupts, use the CX-Programmer to change the input's setting in the PLC Setup.

| Input terminal block |  | CPU Unit |  |  |  |  |  | Input interrupt | Task number |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bit | CPU Units with 60 I/O Points | CPU Units with 40 I/O Points | CPU Units with 30 I/O Points | CPU Units with 20 I/O Points | CPU Units with 14 I/O Points | CPU Units with 10 I/O Points |  |  |
| CIO 0 | 00 | Normal input 0 | Normal input 0 | Normal input 0 | Normal input 0 | Normal input 0 | Normal input 0 | --- | --- |
|  | 01 | Normal input 1 | Normal input 1 | Normal input 1 | Normal input 1 | Normal input 1 | Normal input 1 | --- | --- |
|  | 02 | Normal input 2 | Normal input 2 | Normal input 2 | Normal input 2 | Normal input 2 | Normal input 2 | --- | --- |
|  | 03 | Normal input 3 | Normal input 3 | Normal input 3 | Normal input 3 | Normal input 3 | Normal input 3 | --- | --- |
|  | 04 | Normal input 4 | Normal input 4 | Normal input 4 | Normal input 4 | Normal input 4 | Normal input 4 | Input interrupt 0 | Interrupt task 140 |
|  | 05 | Normal input 5 | Normal input 5 | Normal input 5 | Normal input 5 | Normal input 5 | Normal input 5 | Input interrupt 1 | Interrupt task 141 |
|  | 06 | Normal input 6 | Normal input 6 | Normal input 6 | Normal input 6 | Normal input 6 | --- | Input interrupt 2** | Interrupt task 142** |
|  | 07 | Normal input 7 | Normal input 7 | Normal input 7 | Normal input 7 | Normal input 7 | --- | Input interrupt 3** | Interrupt task 143** |
|  | 08 | Normal input 8 | Normal input 8 | Normal input 8 | Normal input 8 | --- | --- | Input interrupt 4* | Interrupt task 144* |
|  | 09 | Normal input 9 | Normal input 9 | Normal input 9 | Normal input 9 | --- | --- | Input interrupt 5* | Interrupt task 145* |
|  | 10 | Normal input 10 | Normal input 10 | Normal input 10 | Normal input 10 | --- | --- | --- | --- |
|  | 11 | Normal input 11 | Normal input 11 | Normal input 11 | Normal input 11 | --- | --- | --- | --- |
| CIO 1 | $00 \text { to }$ $05$ | Normal input 12 to 17 | Normal input 12 to 17 | Normal input 12 to 17 | --- | --- | --- | --- | --- |
|  | $\begin{aligned} & 06 \text { to } \\ & 11 \end{aligned}$ | Normal inputs 18 to 23 | Normal inputs 18 to 23 | --- | --- | --- | --- | --- | --- |
| CIO 2 | $\begin{array}{\|l} 00 \text { to } \\ 11 \end{array}$ | Normal inputs 24 to 35 | --- | --- | --- | --- | --- | --- | --- |

Note *Input interrupts 4 and 5 are not supported by CPU Units with 10 or 14 I/O Points.
**Input interrupts 2 and 3 are not supported by CPU Units with 10 I/O Points.

## Procedure

| Select the input interrupts. | - Determine the inputs to be used for input interrupts and corresponding task numbers. |
| :---: | :---: |
| $\downarrow$ |  |
| Wire the inputs. | -Wire the inputs. |
| $\downarrow$ |  |
| Set the PLC Setup. | - Use the CX-Programmer to select the interrupt inputs in the PLC Setup. |
| $\downarrow$ |  |
| Write the ladder program. | - Write the programs for the corresponding interrupt task numbers. <br> - Use MSKS(690) to specify up-differentiation or down-differentiation. <br> - Use MSKS(690) to enable input interrupts (in direct mode). |

## PLC Setup

## Note

## Writing the Ladder

 ProgramMSKS(690) Settings

Click the Built-in Input Tab to display the Interrupt Input settings (at the bottom of the tab). Set the input function to Interrupt for each input that will be used as an input interrupt.

(1) Interrupt Input settings IN0 to IN7 correspond to input interrupt numbers 0 to 7.
(2) When using an input as a general-purpose (normal) input, set the input function to Normal.

The MSKS(690) instruction must be executed in order to use input interrupts. The settings made with MSKS(690) are enabled with just one execution, so in general execute MSKS(690) in just one cycle using an up-differentiated condition.
MSKS(690) has the following two functions and two of the instructions are used in combination. If an up-differentiated input interrupt is being used, the first MSKS(690) instruction can be omitted since the input is set for up-differentiation by default.


MSKS(690) Operands

Writing the Interrupt Task's Program

Input Interrupt Settings and Operation

| Input interrupt number | $\begin{gathered} \text { Interrupt } \\ \text { task } \\ \text { number } \end{gathered}$ | 1. Up-differentiation or Down-differentiation |  | 2. Enabling/Disabling the input interrupt |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | N | S | N | S |
|  |  | Input interrupt number | Execution condition | Input interrupt number | Enable/ Disable |
| Input interrupt 0 | 140 | 110 (or 10) | \#0: Up-differentiated \#1: Down-differentiated | 100 (or 6) | \#0: Enable interrupt <br> \#1: Disable interrupt |
| Input interrupt 1 | 141 | 111 (or 11) |  | 101 (or 7) |  |
| Input interrupt 2** | 142 | 112 (or 12) |  | 102 (or 8) |  |
| Input interrupt 3** | 143 | 113 (or 13) |  | 103 (or 9) |  |
| Input interrupt 4* | 144 | 114 |  | 104 |  |
| Input interrupt 5* | 145 | 115 |  | 105 |  |

Note *Input interrupts 4 and 5 are not supported by the CPU Units with 10 or 14 I/O Points.
${ }^{* *}$ Input interrupts 2 and 3 are not supported by the CPU Units with 10 I/O Points.

Create programs for interrupt tasks 140 to 145, which are executed by the corresponding input interrupt. Always put an END(001) instruction at the last address of the program.

This example shows how to execute interrupt task 140 when input CIO 0.04 goes ON.

## Settings

1,2,3... 1. Connect an input device to input 0.04.
2. Use the CX-Programmer to set input 0 as an input interrupt in the PLC Setup.
3. Use the CX-Programmer to create the program to use for interrupt processing and allocate the program to interrupt task 140.
4. Use the CX-Programmer to write MSKS(690) in the program.

W0.00
(Execution condition)


When execution condition W0.00 goes ON, MSKS(690) is executed to enable CIO 0.04 as an up-differentiated input interrupt.
If CIO 0.04 goes from OFF to ON (up-differentiation), processing of the cyclic task that is currently being executed will be interrupted and processing of interrupt task 140 will start. When the interrupt task processing is completed, processing of the interrupted ladder program will restart.


Inputs cannot be used for input interrupts when they are being used as gen-eral-purpose (normal) inputs or quick-response inputs.

## 6-1-3 Input Interrupts (Counter Mode)

## Overview

This function counts up-differentiated or down-differentiated input signals and executes an interrupt task when the count reaches the set value.

- The counter-mode input interrupts use the same input terminals as the direct-mode input interrupts. Refer to 6-1-2 Input Interrupts (Direct Mode) for details.
- The counter input mode can be set to up or down (incrementing or decrementing) with MSKS(690).
- The counter-mode input interrupts start the same interrupt tasks (140 to 145) as the direct-mode input interrupts.
- The maximum input response frequency is 5 kHz total for all countermode input interrupts.

Relationship of Input Bits, Task Numbers, and Counters

| Input bits | Function |  | Counter words |  |
| :--- | :--- | :--- | :--- | :--- |
|  | Input interrupt <br> number | Interrupt task <br> number | SV <br> (0000 to FFFF) | PV |
| 0.04 | Input interrupt 0 | 140 | A532 | A536 |
| 0.05 | Input interrupt 1 | 141 | A533 | A537 |
| $0.06^{\star *}$ | Input interrupt 2* | 142 | A534 | A538 |
| $0.07^{* *}$ | Input interrupt 3** | 143 | A535 | A539 |
| $0.08^{*}$ | Input interrupt 4* | 144 | A544 | A548 |
| $0.09^{*}$ | Input interrupt 5* | 145 | A545 | A549 |

Note *Input interrupts 4 and 5 are not supported by CPU Units with 10 or 14 I/O Points.
**Input interrupts 2 and 3 are not supported by CPU Units with 10 I/O Points.

## Procedure

## PLC Setup

## Writing the Ladder Program

MSKS(690) Settings



Note The input interrupt (counter mode) function is one of the input interrupt functions and executes an interrupt based on the pulse count. If the input pulse frequency is too high, interrupts will occur too frequently and prevent normal cyclic task processing. In this case, cycle time too long errors may occur or the pulse input may not be read.

The maximum total frequency of the counter-mode interrupt inputs is 5 kHz . Even in this case, the high frequencies may adversely affect other devices' operation or the system load, so check the system's operation thoroughly before using the counters at high frequencies.

The procedures for using the CX-Programmer to set the PLC Setup are the same as the procedures for input interrupts (direct mode). Refer to 6-1-2 Input Interrupts (Direct Mode) for details.

The MSKS(690) instruction must be executed in order to use input interrupts. The settings made with MSKS(690) are enabled with just one execution, so in general execute MSKS(690) in just one cycle using an up-differentiated condition.
MSKS(690) has the following two functions and three of the instructions are used in combination. If up-differentiated input pulses are being used, the first MSKS(690) instruction can be omitted since the input is set for up-differentiation by default.


MSKS(690) Operands

| Input interrupt number | Interrupt task muber Count | 1. Up-differentiation or Down-differentiation |  | 2. Enabling/Disabling the input interrupt |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | N | S | N | S |
|  |  | Input interrupt number | Count trigger | Input interrupt number | Enable/ Disable |
| Input interrupt 0 | 140 | 110 (or 10) | \#0: Up-differentiated pulses \#1: Down-differentiated pulses | 100 (or 6) | \#2: Start counting down (decrementing) and enable interrupts <br> \#3: Start counting up (incrementing) and enable interrupts |
| Input interrupt 1 | 141 | 111 (or 11) |  | 101 (or 7) |  |
| Input interrupt 2** | 142** | 112 (or 12) |  | 102 (or 8) |  |
| Input interrupt $3^{* *}$ | 143** | 113 (or 13) |  | 103 (or 9) |  |
| Input interrupt 4* | 144* | 114 |  | 104 |  |
| Input interrupt 5* | 145* | 115 |  | 105 |  |

## Writing the Interrupt Task's Program

Input Interrupt
Settings and Operation
*Input interrupts 4 and 5 are not supported by CPU Units with 10 or 14 I/O Points.
${ }^{* *}$ Input interrupts 2 and 3 are not supported by CPU Units with 10 I/O Points.
Create programs for interrupt tasks 140 to 145, which are executed by the corresponding input interrupt. Always put an END(001) instruction at the last address of the program.

This example shows how to execute interrupt task 141 when 200 up-differentiated pulses have been counted at input CIO 0.05. (The counter is an incrementing counter.)

## Settings

1,2,3... 1. Connect an input device to input 0.05 .
2. Use the CX-Programmer to set input 0.05 as an input interrupt in the PLC Setup.
3. Use the CX-Programmer to create the program to use for interrupt processing and allocate the program to interrupt task 141.
4. Use the CX-Programmer to set a high-speed counter SV of 00C8 hex (200 decimal) in A533.
5. Use the CX-Programmer to write MSKS(690) in the program.

## Operation



When execution condition W0.00 goes ON, MSKS(690) is executed to enable operation of the input interrupt in counter mode.

When CIO 0.05 goes from OFF to ON 200 times, processing of the cyclic task that is currently being executed will be interrupted and processing of interrupt task 141 will start. When the interrupt task processing is completed, processing of the interrupted ladder program will restart.


## Restrictions

Inputs cannot be used for input interrupts when they are being used as gen-eral-purpose (normal) inputs or quick-response inputs.

## 6-1-4 Scheduled Interrupts

This function executes an interrupt task at a fixed time interval measured by the CPU Unit's built-in timer. Interrupt task 2 is allocated to scheduled interrupt.

## Procedure



## PLC Setup

Click the Timings Tab and set the input function to Scheduled Interrupt Interval (the scheduled interrupt timer's units). The timing units can be set to 10 $\mathrm{ms}, 1 \mathrm{~ms}$, or 0.1 ms . The scheduled interrupt timer SV is calculated by multiplying this interval setting by the timer SV set with MSKS(690).

## Scheduled Interrupt Interval Setting



Note
(1) Set a scheduled interrupt time (interval) that is longer than the time required to execute the corresponding interrupt task.
(2) If the scheduled time interval is too short, the scheduled interrupt task will be executed too frequently, which may cause a long cycle time and adversely affect the cyclic task processing.
(3) If an interrupt task is being executed for another interrupt (input interrupt, high-speed counter interrupt, or external interrupt) when the scheduled interrupt occurs, the scheduled interrupt will not be executed until the other interrupt task is completed.
When different kinds of interrupts are being used, design the program to handle multiple interrupts smoothly. Even if two interrupts occur at the same time, the scheduled interrupts will continue as programmed, so the scheduled interrupt tasks will continue to occur at the scheduled times even if specific scheduled interrupts are delayed.

## Writing the Ladder

 ProgramMSKS(690) Settings

The MSKS(690) instruction must be executed in order to use the scheduled interrupt. The settings made with MSKS(690) are enabled with just one execution, so in general execute MSKS(690) in just one cycle using an up-differentiated condition.


MSKS(690) Operands

| Operand |  | Interrupt time interval (period) |  |
| :---: | :---: | :---: | :---: |
| N | S | Time units set in | Scheduled time |
| Scheduled interrupt number | Interrupt time | PLC Setup | interv |
| Scheduled interrupt 0 | \#0000 to \#270F | 10 ms | 10 to $99,990 \mathrm{~ms}$ |
| (interrupt task 2) | (0 to 9999) | 1 ms | 1 to $9,999 \mathrm{~ms}$ |
| 14: Reset start 4: Start without reset |  | 0.1 ms | 0.5 to 999.9 ms |

Writing the Scheduled Interrupt Task's Program

Create the program for interrupt task 2 (scheduled interrupt 0 ), which is executed by the input interrupt. Always put an END(001) instruction at the last address of the program.

## Selecting the Scheduled Interrupt Task



Input Interrupt Settings and Operation

## Settings

1,2,3... 1. Use the CX-Programmer to set the scheduled interrupt time units to 0.1 ms .
2. Use the CX-Programmer to create the interrupt program allocated to interrupt task 2.


When execution condition W0.00 goes ON, MSKS(690) is executed to enable the scheduled interrupt with the reset start specified. The timer is reset and timing starts.

Scheduled interrupt 2 is executed every 30.5 ms .


## 6-1-5 High-speed Counter Interrupts

This function executes the specified interrupt task ( 0 to 255) when the CP1L CPU Unit's built-in high-speed counter PV matches a pre-registered value (target value comparison) or lies within a pre-registered range (range comparison).

- CTBL(882) is used to register the comparison table.
- Either CTBL(882) or $\mathrm{INI}(880)$ can be used to start comparison.
- $\mathrm{INI}(880)$ is used to stop comparison.

For details on the built-in high-speed counter, refer to 5-1 High-speed Counters.

## Procedure



## PLC Setup

Click the Built-in Input Tab to and set the high-speed counters that will be used for interrupts.


## Settings

## Input Function Settings According the PLC Setup Setting

| Item | Setting |
| :--- | :--- |
| Use high speed counter 0 to 3 | Use counter |
| Counting mode | Linear mode |
|  | Circular mode (ring mode) |
| Circular Max. Count | 0 to FFFF FFFF hex <br> (When circular (ring) mode is selected as the count- <br> ing mode, set maximum ring value here.) |
|  | Phase Z and software reset |
|  | Software reset |
|  | Phase Z and software reset (continue comparing) |
|  | Software reset (continue comparing) |
|  | Differential phase inputs (4x) |
|  | Pulse + direction inputs |
|  | Up/Down inputs |
|  | Increment pulse input |

If the built-in inputs are set to be used as high-speed counters 0 to 3 , the function of the input bits will change as shown in the following table. If a highspeed counter is set to be used, the bits in CIO 0 and CIO 1 can no longer be used for normal inputs, input interrupts, or quick-response inputs.

CPU Units with 20,30,40 or 60 I/O Points

| Address |  | Default setting |  |  |  | High-speed counter operation settings: |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bit | CPUUnits with 60 I/O Points | CPU Units with 40 I/O Points | CPU Units with $301 / 0$ Points | CPU Units with 20 I/O Points | Single-phase (increment pulse input) | Two-phase (differential phases x4, up/down, or pulse/direction) | Origin searches |
| CIO 0 | 00 | Normal input 0 | Normal input 0 | Normal input 0 | Normal input 0 | Counter 0, increment input | Counter 0, A phase, up, or count input | --- |
|  | 01 | Normal input 1 | Normal input 1 | Normal input 1 | Normal input 1 | Counter 1, increment input | Counter 0, B phase, down, or direction input | --- |
|  | 02 | Normal input 2 | Normal input 2 | Normal input 2 | Normal input 2 | Counter 2, increment input | Counter 1, A phase, up, or count input | --- |
|  | 03 | Normal input 3 | Normal input 3 | Normal input 3 | Normal input 3 | Counter 3, increment input | Counter 1, B phase, down, or direction input | --- |
|  | 04 | Normal input 4 | Normal input 4 | Normal input 4 | Normal input 4 | Counter 0, phase-Z reset input | Counter 0, phase-Z reset input | --- |
|  | 05 | Normal input 5 | Normal input 5 | Normal input 5 | Normal input 5 | Counter 1, phase-Z reset input | Counter 1, phase-Z reset input | --- |
|  | 06 | Normal input 6 | Normal input 6 | Normal input 6 | Normal input 6 | Counter 2, phase-Z reset input | --- | Pulse output 0: Origin input signal |
|  | 07 | Normal input 7 | Normal input 7 | Normal input 7 | Normal input 7 | Counter 3, phase-Z reset input | --- | Pulse output 1: Origin input signal |
|  | 08 | Normal input 8 | Normal input 8 | Normal input 8 | Normal input 8 | --- | --- | --- |
|  | 09 | Normal input 9 | Normal input 9 | Normal input 9 | Normal input 9 | --- | --- | --- |
|  | 10 | Normal input 10 | Normal input 10 | Normal input 10 | Normal input 10 | -- | --- | Pulse output 0: Origin proximity input signal |
|  | 11 | Normal input 11 | Normal input 11 | Normal input 11 | Normal input 11 | -- | --- | Pulse output 1: Origin proximity input signal |
| CIO 1 | $\begin{aligned} & \hline 00 \text { to } \\ & 05 \end{aligned}$ | Normal input 12 to 17 | Normal input 12 to 17 | Normal input 12 to 17 | -- | --- | --- | --- |
|  | $\begin{array}{\|l\|} \hline 06 \text { to } \\ 11 \end{array}$ | Normal input 18 to 23 | Normal input 18 to 23 | --- | --- | --- | --- | --- |
| CIO 2 | $\begin{aligned} & \hline 00 \text { to } \\ & 11 \end{aligned}$ | Normal input 24 to 35 | --- | --- | --- | --- | --- | --- |

CPU Units with 14 I/O Points

| Input terminal block |  | Default setting | High-speed counter settings |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bit |  | Single-phase (increment pulse input) | Two-phase (differential phase $\times 4$, up/down, or pulse/direction) | Origin searches |
| CIO 0 | 00 | Normal input 0 | High-speed counter 0: Increment input | High-speed counter 0: Phase A, Increment, or Count input | --- |
|  | 01 | Normal input 1 | High-speed counter 1: Increment input | High-speed counter 0: Phase B, Decrement, or Direction input | --- |
|  | 02 | Normal input 2 | High-speed counter 2: Increment input | High-speed counter 1: Phase A, Increment, or Count input | Pulse output 0: Origin proximity input signal |
|  | 03 | Normal input 3 | High-speed counter 3: Increment input | High-speed counter 1: Phase B, Decrement, or Direction input | Pulse output 1: Origin proximity input signal |
|  | 04 | Normal input 4 | High-speed counter 0: Phase Z or reset input | High-speed counter 0: Phase Z or reset input | --- |
|  | 05 | Normal input 5 | High-speed counter 1: Phase Z or reset input | High-speed counter 1: Phase Z or reset input | --- |
|  | 06 | Normal input 6 | High-speed counter 2: Phase Z or reset input | --- | Pulse output 0: Origin input signal |
|  | 07 | Normal input 7 | High-speed counter 3: Phase Z or reset input | --- | Pulse output 1: Origin input signal |

CPU Units with 10 I/O Points

| Address |  | Default setting | High-speed counter Operation settings: |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bit |  | Single-phase (increment pulse input) | Two-phase (differential phase $\times 4$, up/down, or pulse/direction) | Origin searches |
| ClO 0 | 00 | Normal input 0 | Counter 0, increment input | Counter 0, A phase, up, or count input | --- |
|  | 01 | Normal input 1 | Counter 1, increment input | Counter 0, B phase, down, or direction input | --- |
|  | 02 | Normal input 2 | Counter 2, increment input | Counter 1, A phase, up, or count input | --- |
|  | 03 | Normal input 3 | Counter 3, increment input | Counter 1, B phase, down, or direction input | Pulse output 0: Origin proximity input signal |
|  | 04 | Normal input 4 | Counter 0, phase-Z reset input | Counter 0, phase-Z reset input | --- |
|  | 05 | Normal input 5 | Counter 1, phase-Z reset input | Counter 1, phase-Z reset input | Pulse output 0: Origin input signal |

## High-speed Counter

 Memory Areas| Content |  | High-speed counter |  |
| :---: | :---: | :---: | :---: |
|  |  | 0 | 1 |
| PV | Leftmost 4 digits | A271 | A273 |
|  | Rightmost 4 digits | A270 | A272 |
| Range Comparison Condition Met Flags | ON for match in range 1 | A274.00 | A275.00 |
|  | ON for match in range 2 | A274.01 | A275.01 |
|  | ON for match in range 3 | A274.02 | A275.02 |
|  | ON for match in range 4 | A274.03 | A275.03 |
|  | ON for match in range 5 | A274.04 | A275.04 |
|  | ON for match in range 6 | A274.05 | A275.05 |
|  | ON for match in range 7 | A274.06 | A275.06 |
|  | ON for match in range 8 | A274.07 | A275.07 |
| Comparison In-progress Flags | ON while the comparison is in progress. | A274.08 | A275.08 |
| Overflow/Underflow Flags | ON if a PV overflow or underflow occurred while operating in linear mode. | A274.09 | A275.09 |
| Count Direction Flags | 0: Decrementing <br> 1: Incrementing | A274.10 | A275.10 |

Note The comparison table and comparison conditions 1 to 8 are different for tar-get-value comparison and range comparison operations. For details, refer to next page.

REGISTER COMPARISON TABLE Instruction: CTBL(882)

## Contents of the Comparison Table

CTBL(882) compares the PV of a high-speed counter (0 to 3 ) to target values or target value ranges and executes the corresponding interrupt task ( 0 to 255 ) when the specified condition is met.


| Operand |  | Settings |  |
| :--- | :--- | :--- | :--- |
| P | High-speed <br> counter number | $\# 0000$ | High-speed counter 0 |
|  | C | Control data | $\# 0001$ |
| High-speed counter 1 |  |  |  |\(\left|\begin{array}{l}Registers a target-value comparison table and <br>

starts the comparison operation.\end{array}\right|\)

## Target-value Comparison Table

Depending on the number of target values in the table, the target-value comparison table requires a continuous block of 4 to 145 words.


## Range Comparison Table

The range comparison table requires a continuous block of 40 words because comparison conditions 1 to 8 require 5 words each ( 2 words for the upper range value, 2 words for the lower range value, and one word for the interrupt task number).

MODE CONTROL Instruction: INI(880)


Note Always set the upper limit greater than or equal to the lower limit in each range.
$\mathrm{INI}(880)$ can be used to start/stop comparison with the high-speed counter's comparison table, change the high-speed counter's PV, change the PV of interrupt inputs in counter mode, and control the pulse output functions.


| Operand |  | Settings |  |
| :---: | :---: | :---: | :---: |
| P | Port specifier | \#0000, \#0001 | Pulse outputs 0 or 1 |
|  |  | \#0010 | High-speed counter 0 |
|  |  | \#0011 | High-speed counter 1 |
|  |  | \#0100 to \#0105 | Input interrupts 0 to 5 (in counter mode) |
|  |  | \#1000 or \#1001 | PWM(891) output 0 or 1 |
| C | Control data | \#0000 | Start comparison. |
|  |  | \#0001 | Stop comparison. |
|  |  | \#0002 | Change the PV. |
|  |  | \#0003 | Stop pulse output. |
| NV | First word of new PV | NV and NV+1 contain the new PV when C is set to \#0002 (change the PV). |  |

New PV Setting in NV and NV+1


## Ladder Program <br> Examples

Example 1: High-speed Counter (Linear Mode)

In this example, high-speed counter 0 operates in linear mode and starts interrupt task 10 when the PV reaches 30,000 ( 00007530 hex).

1,2,3... 1. Set high-speed counter 0 in the PLC Setup's Built-in Input Tab.

| Item | Setting |
| :--- | :--- |
| High-speed counter 0 | Use counter |
| Counting mode | Linear mode |
| Circular Max. Count | --- |
| Reset method | Software reset |
| Input Setting | Up/Down inputs |

2. Set the target-value comparison table in words D10000 to D10003.

| Word | Setting | Function |  |  |
| :---: | :--- | :--- | :--- | :---: |
| D10000 | $\# 0001$ | Number of target values = 1 |  |  |
| D10001 | $\# 7530$ | Rightmost 4 digits of the target value 1 data | Target value $=$ <br> 30,000 <br> $(0000 ~ 7530 ~ h e x) ~$ |  |
| D10002 | $\# 0000$ | Leftmost 4 digits of the target value 1 data | Bit 15: 0 (incrementing) <br> Bits 0 to 7: A hex (interrupt task number 10) |  |
| D10003 | $\# 000$ A |  |  |  |

3. Create the program for interrupt task 10. Always put an END(001) instruction at the program's last address.
4. Use CTBL(882) to start the comparison operation with high-speed counter 0 and interrupt task 10.

5. Operation

When execution condition W0.00 goes ON, the comparison starts with high-speed counter 0.
When the PV of high speed counter 0 reaches 30,000 , cyclic task processing is interrupted, and interrupt task 10 is processed. When interrupt task 10 processing is completed, processing of the interrupted cyclic task resumes.


## Example 2: High-speed Counter (Ring Mode)

In this example, high-speed counter 1 operates in circular (ring) mode and starts interrupt task 12 when the PV is between 25,000 (0000 61A8 hex) and 25,500 (0000 639C hex).
The maximum ring count is set at 50,000 (0000 C350 hex).
1,2,3... 1. Set high-speed counter 1 in the PLC Setup's Built-in Input Tab.

| Item | Setting |
| :--- | :--- |
| High-speed counter 1 | Use counter |
| Counting mode | Circular mode |
| Circular Max. Count | 50,000 |
| Reset method | Software reset (continue comparing) |
| Input Setting | Up/Down inputs |

2. Set the range comparison table starting at word D20000. Even though range 1 is the only range being used, all 40 words must still be dedicated to the range comparison table.

| Word | Setting | Function |  |  |
| :--- | :--- | :--- | :--- | :--- |
| D20000 | \#61A8 | Rightmost 4 digits of range 1 lower limit | Lower limit value: |  |
| D20001 | \#0000 | Leftmost 4 digits of range 1 lower limit |  |  |
| D20002 | \#639C | Rightmost 4 digits of range 1 upper limit | Upper limit value: <br> 25,500 |  |
| D20003 | \#0000 | Leftmost 4 digits of range 1 upper limit |  |  |


| Word | Setting | Function |  |
| :---: | :---: | :---: | :---: |
| D20004 | \#000C | Range 1 interrupt task number = 12 ( C hex) |  |
| $\begin{aligned} & \hline \text { D20005 to } \\ & \text { D20008 } \end{aligned}$ | $\begin{aligned} & \text { All } \\ & \text { \#0000 } \end{aligned}$ | Range 2 lower and upper limit values (Not used and don't need to be set.) | Range 2 settings |
| D20009 | \#FFFF | Disables range 2. |  |
| $\sim$ |  |  |  |
| $\begin{aligned} & \hline \text { D20014 } \\ & \text { D20019 } \\ & \text { D20024 } \\ & \text { D20029 } \\ & \text { D20034 } \end{aligned}$ | \#FFFF | Set the fifth word for ranges 3 to 7 (listed at left) to \#FFFF to disable those ranges. |  |
| ~ |  |  |  |
| $\begin{aligned} & \hline \text { D20035 to } \\ & \text { D20038 } \end{aligned}$ | $\begin{aligned} & \hline \text { All } \\ & \# 0000 \end{aligned}$ | Range 8 lower and upper limit values (Not used and don't need to be set.) | Range 8 settings |
| D20039 | \#FFFF | Disables range 8. |  |

3. Create the program for interrupt task 12. Always put an END(001) instruction at the program's last address.
4. Use CTBL(882) to start the comparison operation with high-speed counter 1 and interrupt task 12.

5. Operation

When execution condition W0.00 goes ON, the comparison starts with high-speed counter 1.
When the PV of high speed counter 1 is between 25,000 and 25,500, cyclic task processing is interrupted, and interrupt task 12 is processed. When interrupt task 12 processing is completed, processing of the interrupted cyclic task resumes.


## 6-2 Quick-response Inputs

## Overview

## PLC Setup

Bit Allocation for Quick-Response Inputs

CPU Units with 10 I/O Points

The quick-response inputs can read pulses with an ON time shorter than the cycle time (as short as $50 \mu \mathrm{~s}$ ). Use the quick-response inputs to read signals shorter than the cycle time, such as inputs from photomicrosensors.

Use the CX-Programmer to set a built-in input as a quick-response input in the PLC Setup. Click the Built-in Input Tab to display the Interrupt Input settings (at the bottom of the tab). Set the input function from Normal to Quick for each input that will be used as a quick-response input.


The following diagrams show the input bits and terminals that can be used for quick-response inputs in each CPU Unit.

The 2 input bits CIO 0.04 to CIO 0.05 can be used as quick-response inputs.


CPU Units with 14 I/O Points

CPU Units with 20 I/O Points

## CPU Units with 30 I/O

 PointsUpper Terminal Block (CPU Unit with AC Power Supply)

Upper Terminal Block (CPU Unit with AC Power Supply)

CPU Units with 40 I/O Points

Upper Terminal Block (CPU Unit with AC Power Supply)

The 4 input bits CIO 0.04 to CIO 0.07 can be used as quick-response inputs.


The 6 input bits CIO 0.04 to CIO 0.09 can be used as quick-response inputs.


The 6 input bits CIO 0.04 to CIO 0.09 can be used as quick-response inputs.


The 6 input bits CIO 0.04 to CIO 0.09 can be used as quick-response inputs.


## CPU Units with 60 I/O Points

The 6 input bits CIO 0.04 to CIO 0.09 can be used as quick-response inputs.

## Upper Terminal Block

(CPU Unit with AC
Power Supply )


## Setting the Input Functions in the PLC Setup

Normally, bits CIO 0.04 to CIO 0.09 are used as normal inputs. When using these inputs as quick-response inputs, use the CX-Programmer to change the input's setting in the PLC Setup.

| Input terminal block |  | CPU Unit |  |  |  |  |  | Quickresponse inputs |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bit | CPU Units with 60 I/O Points | CPU Units with 40 I/O Points | CPU Units with 30 I/O Points | CPU Units with 20 I/O Points | CPU Units with 14 I/O Points | CPU Units with 10 I/O Points |  |
| CIO 0 | 00 | Normal input 0 | Normal input 0 | Normal input 0 | Normal input 0 | Normal input 0 | Normal input 0 | --- |
|  | 01 | Normal input 1 | Normal input 1 | Normal input 1 | Normal input 1 | Normal input 1 | Normal input 1 | --- |
|  | 02 | Normal input 2 | Normal input 2 | Normal input 2 | Normal input 2 | Normal input 2 | Normal input 2 | --- |
|  | 03 | Normal input 3 | Normal input 3 | Normal input 3 | Normal input 3 | Normal input 3 | Normal input 3 | --- |
|  | 04 | Normal input 4 | Normal input 4 | Normal input 4 | Normal input 4 | Normal input 4 | Normal input 4 | Quick-response input 0 |
|  | 05 | Normal input 5 | Normal input 5 | Normal input 5 | Normal input 5 | Normal input 5 | Normal input 5 | Quick-response input 1 |
|  | 06 | Normal input 6 | Normal input 6 | Normal input 6 | Normal input 6 | Normal input 6 | --- | Quick-response input 2** |
|  | 07 | Normal input 7 | Normal input 7 | Normal input 7 | Normal input 7 | Normal input 7 | --- | Quick-response input 3** |
|  | 08 | Normal input 8 | Normal input 8 | Normal input 8 | Normal input 8 | --- | --- | Quick-response input 4* |
|  | 09 | Normal input 9 | Normal input 9 | Normal input 9 | Normal input 9 | --- | --- | Quick-response input 5* |
|  | 10 | Normal input 10 | $\begin{array}{\|l} \hline \begin{array}{l} \text { Normal input } \\ 10 \end{array} \\ \hline \end{array}$ | $\begin{aligned} & \text { Normal input } \\ & 10 \end{aligned}$ | Normal input 10 | --- | --- | --- |
|  | 11 | $\begin{aligned} & \text { Normal input } \\ & 11 \end{aligned}$ | Normal input 11 | Normal input 11 | Normal input 11 | --- | --- | --- |
| CIO 1 | 00 to 05 | Normal input 12 to 17 | $\begin{array}{\|l\|} \hline \text { Normal input } \\ 12 \text { to } 17 \\ \hline \end{array}$ | $\begin{array}{\|l\|} \hline \text { Normal input } \\ 12 \text { to } 17 \end{array}$ | --- | --- | --- | --- |
|  | 06 to 11 | Normal input 18 to 23 | Normal input 18 to 23 | --- | --- | --- | --- | --- |
| CIO 2 | 00 to 11 | $\begin{aligned} & \text { Normal input } \\ & 24 \text { to } 35 \end{aligned}$ | --- | --- | --- | --- | --- | --- |

Note *Input interrupts 4 and 5 are not supported by CPU Units with 10 or 14 I/O Points.
**Input interrupts 2 and 3 are not supported by CPU Units with 10 I/O Points.

## Interrupt Input and

Quick-response Input Specifications

| Item | Specification |  |
| :--- | :--- | :--- |
| ON delay | $30 \mu \mathrm{~s}$ max. |  |
| OFF delay | $150 \mu \mathrm{~s}$ max. |  |
| Response pulse | ON |  |
|  | OFF |  |

## Procedure



## Restrictions

Inputs cannot be used as quick-response inputs when they are being used as general-purpose (normal) inputs, input interrupts, or high-speed counter inputs.

## 6-3 Serial Communications

## 6-3-1 Overview

The CP1L CPU Units support the following serial communications functions.

| Protocol | Connected devices | Description | Serial port 1 | Serial port 2 |
| :---: | :---: | :---: | :---: | :---: |
| No-protocol | Standard devices supporting serial communications | Communicates with standard devices with an RS-232C or RS-422A/485 port without a command-response format. Instead the TXD(236) and RXD(235) instructions are executed from the program to transmit data from the transmission port or read data in the reception port. The frame headers and end codes can be specified. | OK | OK |
| Serial gate- <br> way (to <br> Compo- <br> Way/F or <br> Modbus- <br> RTU) | OMRON components supporting CompoWay/F or Mod-bus-RTU slave devices | Converts received FINS commands into CompoWay/F or Modbus-RTU commands and transfers them on the serial communications path. | OK | OK |


| Protocol | Connected devices | Description | Serial port 1 | Serial port 2 |
| :---: | :---: | :---: | :---: | :---: |
| Serial PLC Link |  | Up to ten words per Unit can be shared by up to nine CPU Units, including one Polling Unit and eight Polled Units. <br> An RS-422A/485 Option Boards (CP1W-CIF11/CIF12) are used to communicate via RS-422A/485, or RS-232C Option Boards (CP1W-CIF01) can be used to communicate between two CPU Units via an RS-232C connection. <br> CJ1M CPU Units can also be included in Serial PLC Links, and the Serial PLC Links can also include PTs as Polled Units via 1:N NT Links. <br> Note Serial PLC Links can be created on serial port 1 or serial port 2, but not on both ports at the same time. | OK | OK |
| 1:N NT links (1:N NT Links are also used for 1:1 connections.) | OMRON PTs (Programmable Terminals) | Data can be exchanged with PTs without using a communications program in the CPU Unit. | OK | OK |
| Host Link | Host computer or OMRON PT (Programmable Terminal) | 1) Various control commands such as reading and writing I/O memory, changing the operating mode, and forcesetting/resetting bits can be executed by sending Cmode host link commands or FINS commands from the host computer to the CPU Unit. <br> 2) It is also possible to send FINS commands from the CPU Unit to the host computer to send data or information. <br> Use Host Link communications to monitor data, such as operating status, error information, and quality data in the PLC or send data, such as production planning information, to the PLC. | OK | OK |


| Protocol | Connected devices | Description | Serial port 1 | Serial port 2 |
| :---: | :---: | :---: | :---: | :---: |
| Peripheral bus (toolbus) |  | Provides high-speed communications with the CX-Programmer. | OK | OK |
| 1:1 NT Links | OMRON PTs (Programmable Terminals) | Enables data exchange with a PT without communications programming in the CPU Unit. (The 1:N NT Link protocol is used for communications even for 1:1 connections.) | OK | OK |
| 1:1 Links |  | Enables linking data in a 64word Link Area between two PLCs connected by an RS232C cable. | OK | OK |

## 6-3-2 No-protocol Communications

No-protocol communications enable sending and receiving data using the TRANSMIT (TXD(236)) and RECEIVE (RXD(235)) instructions without using a protocol and without data conversion (e.g., no retry processing, data type conversion, or process branching based on received data). The communications mode for the serial port must be set for no-protocol communications in the PLC Setup.
No-protocol communications are used to send data in one direction to or from standard devices that have an RS-232C or RS-422A/485 port using TXD(236) or $\operatorname{RXD}(235)$.


For example, simple (non-protocol) communications can be used to input data from a barcode reader or output data to a printer.
The following table lists the no-protocol communication functions supported by CP1L PLCs.

| Transfer direction | Method | Max. amount of data | Frame format |  | Other functions |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  |  | Start code | End code |  |
| Data transmission (PLC $\rightarrow$ External device) | Execution of TXD(236) in the program | 256 bytes | Yes: 00 to FF <br> No: None | Yes: <br> 00 to FF or CR+LF <br> No: None <br> (The amount of data to receive is specified between 1 and 256 bytes when no end code is specified.) | - Send delay time (delay between TXD(236) execution and sending data from specified port): 0 to 99,990 ms (unit: 10 ms ) <br> - Controlling RS and ER signals |
| Data reception <br> (External device $\rightarrow$ PLC) | Execution of RXD(235) in the program | 256 bytes |  |  | Monitoring CS and DR signals |

## Procedure

## Message Frame Formats



Data can be placed between a start code and end code for transmission by TXD(236) and data between a start code and end code can be received by RXD(235). When transmitting with TXD(236), data from I/O memory is transmitted, and when receiving with $\operatorname{RXD}(235)$, the data (without start/end codes) is stored in I/O memory. Up to 256 bytes (including the start and end codes) can be transferred in no-protocol mode.
The start and end codes are set in the PLC Setup.
The following table shows the message formats that can be set for transmissions and receptions in no-protocol mode.

| Start code | End code |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | No | Yes |  |  | CR+LF |  |  |
| No |  | $\qquad$ |  | ata <br> bytes max | ED |  | data <br> bytes max. | $C R+F$ |
| Yes | ST | data <br> 256 bytes max. |  |  | $\int E D$ | ST | data <br> 256 bytes max | CR+LF |

- When more than one start code is used, the first start code will be effective.
- When more than one end code is used, the first end code will be effective.
- If the data being transferred contains the end code, the data transfer will be stopped midway. In this case, change the end code to CR+LF.

Note A setting can be made to delay the transmission of data after the execution of TXD(236).


Execution of TXD(236)
Refer to the SYSMAC CP Series CP1L CPU Unit Programming Manual (W451) for more details on TXD(236) and RXD(235).

## 6-3-3 Modbus-RTU Easy Master Function

## Overview

If an RS-232C or RS-422A/485 Option Board is used, the CP1L CPU Unit can function as a Modbus-RTU Master to send Modbus-RTU commands by manipulating software switches. This enables easily controlling Modbus-compliant slaves, such as Inverters, through serial communications.
The following OMRON Inverters support Modbus-RTU slave operation: 3G3JV, 3G3MV, and 3G3RV.
The communications mode in the PLC Setup must be set to the Gateway Mode to enable this functionality.


Modbus-RTU commands can be set simply by turning ON a software switch after setting the Modbus slave address, function, and data in the DM fixed allocation words for the Modbus-RTU Easy Master. The response when received is also store in the DM fixed allocation words for the Modbus-RTU Easy Master.


DM Fixed Allocation
Words for the Modbus-RTU Easy Master

The Modbus-RTU command is stored in the following words in the DM Area.

- M-type CPU Units

Serial port 1: D32200 to D32249
Serial port 2: D32300 to D32349

- L-type CPU Units Serial port 1: D32300 to D32349
When a response is received after turning ON the Modbus-RTU Master Execution Bit, it is sotred in the following words in the DM Area.
- M-type CPU Units

Serial port 1: D32250 to D32299
Serial port 2: D32350 to D32399

- L-type CPU Units Serial port 1: D32350 to D32399

| Words |  | Bits | Contents |  |
| :---: | :---: | :---: | :---: | :---: |
| Serial port 1 on M-type CPU Unit | Serial port 2 on M-type CPU Unit or Serial port 1 on L-type CPU Unit |  |  |  |
| D32200 | D32300 | 00 to 07 | Command | Slave address (00 to F7 hex) |
|  |  | 08 to 15 |  | Reserved (Always 00.) |
| D32201 | D32301 | 00 to 07 |  | Function code |
|  |  | 08 to 15 |  | Reserved (Always 00.) |
| D32202 | D32302 | 00 to 15 |  | Number of communications data bytes ( 0000 to 005E hex) |
| $\begin{array}{\|l} \hline \text { D32203 to } \\ \text { D32249 } \end{array}$ | $\begin{array}{\|l} \hline \text { D32303 to } \\ \text { D32349 } \end{array}$ | 00 to 15 |  | Communications data (94 bytes maximum) |


| Words |  | Bits | Contents |  |
| :---: | :---: | :---: | :---: | :---: |
| Serial port 1 on M-type CPU Unit | Serial port 2 on M-type CPU Unit or Serial port 1 on L-type CPU Unit |  |  |  |
| D32250 | D32350 | 00 to 07 | Response | Slave address (00 to F7 hex) |
|  |  | 08 to 15 |  | Reserved (Always 00.) |
| D32251 | D32351 | 00 to 07 |  | Function code |
|  |  | 08 to 15 |  | Reserved |
| D32252 | D32352 | 00 to 07 |  | Error code |
|  |  | 08 to 15 |  | Reserved (Always 00.) |
| D32253 | D32353 | 00 to 15 |  | Number of response bytes (0000 to 03EA hex) |
| $\begin{aligned} & \hline \text { D32254 to } \\ & \text { D32299 } \end{aligned}$ | $\begin{aligned} & \hline \text { D32354 to } \\ & \text { D32399 } \end{aligned}$ | 00 to 15 |  | Response data (92 bytes maximum) |

Error Codes
The following error codes are stored in an allocated DM Area word when an error occurs in Modbus-RTU Easy Master function execution.

| Code | Name | Description |
| :--- | :--- | :--- |
| $0 \times 00$ | Normal end | Not an error. |
| $0 \times 01$ | Illegal address | The slave address specified in the parameter <br> is illegal (248 or higher). |
| $0 \times 02$ | Illegal function code | The function code specified in the parameter is <br> illegal. |
| $0 \times 03$ | Data length overflow | There are more than 94 data bytes. |
| $0 \times 04$ | Serial communica- <br> tions mode error | The Modbus-RTU Easy Master function was <br> executed when the serial communications <br> mode was not the Serial Gateway Mode. |
| $0 \times 80$ | Response timeout | A response was not received from the Servo. |
| $0 \times 81$ | Parity error | A parity error occurred. |
| $0 \times 82$ | Framing error | A framing error occurred. |
| $0 \times 83$ | Overrun error | An overrun error occurred. |
| $0 \times 84$ | CRC error | A CRC error occurred. |
| $0 \times 85$ | Incorrect confirmation <br> address | The slave address in the response is differ- <br> ence from the one in the request. |
| $0 \times 86$ | Incorrect confirmation <br> function code | The function code in the response is difference <br> from the one in the request. |
| $0 \times 87$ | Response size over- <br> flow | The response frame is larger than the storage <br> area (92 bytes). |
| $0 \times 88$ | Exception response | An exception response was received from the <br> slave. |
| $0 \times 89$ | Service being exe- <br> cuted | A service is already being executed (reception <br> traffic congestion). |
| $0 \times 8 \mathrm{~A}$ | Execution canceled | Executing the service has been canceled. |
| $0 \times 8 f$ | Other error | Other FINS response code was received. |

## Auxiliary Area Flags and Bits

The Modbus-RTU command set in the DM fixed allocation words for the Mod-bus-RTU Easy Master is automatically sent when the Modbus-RTU Master Execution Bit is turned ON. The results (normal or error) will be given in corresponding flags.

| Word | Bit | Port | Contents |
| :---: | :---: | :---: | :---: |
| A640 | 00 | M-type CPU Units: Serial port 2 L-type CPU Units: Serial port 1 | Modbus-RTU Master Execution Bit Turned ON: Execution started ON: Execution in progress. OFF: Not executed or execution completed. |
|  | 01 |  | Modbus-RTU Master Execution Normal Flag ON: Execution normal. OFF: Execution error or still in progress. |
|  | 02 |  | Modbus-RTU Master Execution Error Flag ON: Execution error. OFF: Execution normal or still in progress. |
| A641 | 00 | M-type CPU Unit: Serial port 1 | Modbus-RTU Master Execution Bit <br> Turned ON: Execution started <br> ON: Execution in progress. <br> OFF: Not executed or execution completed. |
|  | 01 |  | Modbus-RTU Master Execution Normal Flag ON: Execution normal. <br> OFF: Execution error or still in progress. |
|  | 02 |  | Modbus-RTU Master Execution Error Flag ON: Execution error. <br> OFF: Execution normal or still in progress. |

## 6-3-4 Communications: Smart Active Parts and Function Blocks

## Overview

OMRON components that support CompoWay/F communications or ModbusRTU slave functionality (such as Temperature Controllers) can be easily accessed from a CP1L CPU Unit equipped with an RS-422A/485 or RS-232C Option Board using Smart Active Parts (SAPs) on an NS-series PT or using function blocks in the ladder program in the CP1L CPU Unit.
The communications mode in the PLC Setup must be set to the Gateway Mode to enable this functionality.

## System Configuration



Note Refer to OMRON's Smart Library website for the most recent information on using SAPs and function blocks.

## Serial Gateway Function

When a FINS command is received, it is automatically converted to the protocol corresponding to the message and sent on the serial communications path. Responses are also converted in the same way.
Note Serial ports 1 and 2 on the CP1L CPU Unit can be used to convert to the following protocols.

- CompoWay/F
- Modbus-RTU

This functionality is enabled when the serial communications mode is set to Serial Gateway.


## Contents of FINS Header

- Destination network address (DNA)
a. When the routing table for network control of serial communication channel is developed:
It is the network address that corresponds to serial communication port according to the routing table.
b. When the routing table for networking serial communication channel is not developed:
It is the network address when actual destination PLC is specified.
- Destination node address (DA1)
a. When the routing table for network control of serial communication channel is developed:
00 Hex (means PLC internal communication)
b. When the routing table for network control of serial communication channel is not developed:
It is the node address when actual destination PLC is specified.
- Destination model address (DA2) It needs to be the model address of serial communication port. CP1L CPU Unit with 30,40 or 60 I/O Points


| Serial communication port of CP1L | Model address of serial <br> communication port |
| :--- | :--- |
| Serial port 1 | FD Hex (decimal 253) |
| Serial port 2 | FC Hex (decimal 252) |

CP1L CPU Unit with 14 or 20 I/O Points


| Serial communication port of CP1L | Model address of serial <br> communication port |
| :--- | :---: |
| Serial port 1 | FC Hex (decimal 252) |

CPU Unit Serial Gateway Function Specifications

| Item | Specification |
| :--- | :--- |
| Pre-conversion data | FINS (via FINS network, Host Link FINS, toolbus, NT Link, <br> or CPU bus) |
| Conversion functions | FINS commands addressed to serial port 1 or 2 on the CPU <br> Unit are converted to CompoWay/F commands (after <br> removing the header) if the FINS command code is 2803 <br> hex and to Modbus-RTU commands (after removing the <br> header) if the FINS command code is 2804 hex. |
| Post-conversion data | CompoWay/F command or Modbus-RTU command |
| Serial communications <br> method | 1:N half-duplex |
| Maximum number of <br> nodes | 31 |
| Enabling serial commu- <br> nications mode | Serial Gateway Mode |
| Response timeout | The time from when a message converted to a different pro- <br> tocol is set until a response is received is monitored by the <br> serial gateway function. <br> Default: 5 s, User setting: 0.1 to 25.5 s <br> Note A FINS response code of 0205 hex (response time- <br> out) is sent to the source of the FINS command if a <br> timeout occurs. |
| Send delay function | None |

Note If a CJ-series Serial Communications Unit is connected via a CJ Unit Adapter, messages can also be converted to Modbus-ASCII or Host Link FINS. Refer to the SYSMAC CS/CJ Series Serial Communications Boards/Units Operation Manual (W336) for details.

## 6-3-5 Serial PLC Links

## Overview

Serial PLC Links can be used to allow data to be exchanged among CP1L and CJ1M CPU Units via the RS-422A/485 or RS-232C Option Boards mounted to the CPU Units without requiring special programming. The communications mode in the PLC Setup must be set to the Serial PLC Link Mode to enable this functionality.

- Either serial port 1 or 2 can be used. (See note.)
- Words are allocated in memory in the Serial PLC Link Words (CIO 3100 to CIO 3199).
- A maximum of 10 words can be transferred by each CP1L CPU Unit, but the number of linked words can be set to fewer words. (The size must be the same for all CP1L CPU Units.)

Note Serial PLC Links cannot be used on serial ports 1 and 2 at the same time. If one port is set as a Serial PLC Link slave or master, it will not be possible to set the other port for a Serial PLC Link. A PLC Setup error will occur if an attempt is made to set both ports for Serial PLC Links.

## Configuration

1:N Connections between CP1L/CJ1M CPU Units (8 Nodes Maximum)


1:1 Connections between CP1L/CJ1M CPU Units


## Specifications

## Data Refresh Methods

## Complete Link Method

| Item | Specifications |
| :--- | :--- |
| Applicable serial <br> ports | Serial port 1 or 2. Both ports cannot be used for PLC Links at <br> the same time. If both ports are set for PLC Links (either as <br> polling node or polled node), a PLC Setup setting error (non- <br> fatal error) will occur and the PLC Setup Setting Error Flag <br> (A402.10) will turn ON. |
| Connection method | RS-422A/485 or RS-232C connection via RS-422A/485 or <br> RS-232C Option Board. |
| Allocated data area | Serial PLC Link Words: <br> CIO 3100 to CIO 3199 (Up to 10 words can be allocated for <br> each CPU Unit.) |
| Number of Units | 9 Units max., comprising 1 Polling Unit and 8 Polled Units (A <br> PT can be placed on the same network in an 1:N NT Link, but <br> it must be counted as one of the 8 Polled Units.) |
| Link methods (data <br> refresh methods) | Complete link method or Polling Unit link method |

The following two methods can be used to refresh data.

- Complete link method
- Polling Unit link method

The data from all nodes in the Serial PLC Links are reflected in both the Poll- ing Unit and the Polled Units. (The only exceptions are the address allocated
to the connected PT's unit number and the addresses of Polled Units that are not present in the network. These data areas are undefined in all nodes.)

## Example: Complete Link Method, Highest Unit Number: 3

In the following diagram, Polled Unit No. 2 is either a PT or is a Unit not present in the network, so the area allocated for Polled Unit No. 2 is undefined in all nodes.


## Example: Complete Link Method, Number of Link Words: 10

Each CPU Unit (either CP1L or CJ1M) sends data to the same words in all other CPU Units for the Polling Unit and all Polled Units. The Polling Unit is a CP1L CPU Unit in the following example, but it could also be a CJ1M CPU Unit.


Polling Unit Link Method

The data for all the Polled Units in the Serial PLC Links ar reflected in the Polling Unit only, and each Polled Unit reflects the data of the Polling Unit only. The advantage of the Polling Unit link method is that the addresses allocated for the local Polled Unit data are the same in each Polled Unit, allowing data to be accessed using common ladder programming. The areas allocated for the unit numbers of the PT or Polled Units not present in the network are undefined in the Polling Unit only.

## Example: Polling Unit Link Method, Highest Unit Number: 3

In the following diagram, Polled Unit No. 2 is a PT or a Unit not participating in the network, so the corresponding area in the Polling Unit is undefined.

| Polling Unit | Polled Unit No. | Polled Unit No. | Polled Unit No. 3 |
| :---: | :---: | :---: | :---: |
| Local area | Polling Unit | $\rightarrow$ Polling Unit | $\rightarrow$ Polling Unit |
| Polled Unit No. 0 | Local area | Local area | Local area |
| Polled Unit No. 1 | (Not used.) | (Not used.) | (Not used.) |
| Undefined | (Not used.) | (Not used.) | (Not used.) |
| Polled Unit No. 3 | (Not used.) | (Not used.) | (Not used.) |
| (Not used.) | (Not used.) | (Not used.) | (Not used.) |
| (Not used.) | (Not used.) | (Not used.) | (Not used.) |
| (Not used.) | (Not used.) | (Not used.) | (Not used.) |
| (Not used.) | (Not used.) | (Not used.) | (Not used.) |

Example: Polling Unit Link Method, Number of Link Words: 10
The CPU Unit that is the Polling Unit (either CP1L or CJ1M) sends its data (CIO 3100 to CIO 3109 ) to the same words (CIO 3100 to CIO 3109 ) in all other CPU Units. The Polled Units send their data (CIO 3110 to CIO 3119 ) to consecutive sets of 10 words in the Polling Unit. The Polling Units is a CP1L CPU Unit in the following example, but it could also be a CJ1M CPU Unit. (Only the first three Polled Units are shown below.)


## Allocated Words

Complete Link Method
Address
CIO 3100

CIO 3199
Serial PLC
Link Words

| Link words | 1 word | 2 words | 3 words | to | 10 words |
| :--- | :--- | :--- | :--- | :--- | :--- |
| Polling Unit | CIO 3100 | CIO 3100 to <br> CIO 3101 | CIO 3100 to <br> CIO 3102 |  | CIO 3100 to <br> CIO 3109 |
| Polled Unit No. 0 | CIO 3101 | CIO 3102 to <br> CIO 3103 | CIO 3103 to <br> CIO 3105 |  | CIO 3110 to <br> CIO 3119 |
| Polled Unit No. 1 | CIO 3102 | CIO 3104 to <br> CIO 3105 | CIO 3106 to <br> CIO 3108 |  | CIO 3120 to <br> CIO 3129 |
| Polled Unit No. 2 | CIO 3103 | CIO 3106 to <br> CIO 3107 | CIO 3109 to <br> CIO 3111 |  | CIO 3130 to <br> CIO 3139 |
| Polled Unit No. 3 | CIO 3104 | CIO 3108 to <br> CIO 3109 | CIO 3112 to <br> CIO 3114 |  | CIO 3140 to <br> CIO 3149 |
| Polled Unit No. 4 | CIO 3105 | CIO 3110 to <br> CIO 3111 | CIO 3115 to <br> CIO 3117 |  | CIO 3150 to <br> CIO 3159 |
| Polled Unit No. 5 | CIO 3106 | CIO 3112 to <br> CIO 3113 | CIO 3118 to <br> CIO 3120 |  | CIO 3160 to <br> CIO 3169 |
| Polled Unit No. 6 | CIO 3107 | CIO 3114 to <br> CIO 3115 | CIO 3121 to <br> CIO 3123 |  | CIO 3170 to <br> CIO 3179 |
| Polled Unit No. 7 | CIO 3108 | CIO 3116 to <br> CIO 3117 | CIO 3124 to <br> CIO 3126 |  | CIO 3180 to <br> CIO 3189 |
| Not used. | CIO 3109 <br> to <br> CIO 3199 | CIO 3118 to <br> CIO 3199 | CIO 3127 to <br> CIO 3199 |  | CIO 3190 to <br> CIO 3199 |

Polling Unit Link Method

| Address |
| :---: | :---: |
| CIO 3100 |
|  |
| CIO 3199 |
| Sink Words PLC |
| Link |


| Link words | 1 word | 2 words | 3 words | to | 10 words |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Polling Unit | CIO 3100 | $\begin{array}{\|l} \hline \text { CIO } 3100 \text { to } \\ \text { CIO } 3101 \end{array}$ | $\begin{aligned} & \hline \text { CIO } 3100 \text { to } \\ & \text { CIO } 3102 \end{aligned}$ |  | $\begin{array}{\|l} \hline \text { CIO } 3100 \text { to } \\ \text { CIO } 3109 \end{array}$ |
| Polled Unit No. 0 | CIO 3101 | $\text { CIO } 3102 \text { to }$ $\text { CIO } 3103$ | $\begin{aligned} & \text { CIO } 3103 \text { to } \\ & \text { CIO } 3105 \end{aligned}$ |  | $\text { CIO } 3110 \text { to }$ $\text { CIO } 3119$ |
| Polled Unit No. 1 | CIO 3101 | $\text { CIO } 3102 \text { to }$ $\text { CIO } 3103$ | $\begin{aligned} & \text { CIO } 3103 \text { to } \\ & \text { CIO } 3105 \end{aligned}$ |  | $\text { CIO } 3110 \text { to }$ <br> CIO 3119 |
| Polled Unit No. 2 | CIO 3101 | $\begin{aligned} & \text { CIO } 3102 \text { to } \\ & \text { CIO } 3103 \end{aligned}$ | $\begin{aligned} & \text { CIO } 3103 \text { to } \\ & \text { CIO } 3105 \end{aligned}$ |  | $\begin{array}{\|l} \hline \text { CIO } 3110 \text { to } \\ \text { CIO } 3119 \end{array}$ |
| Polled Unit No. 3 | CIO 3101 | $\text { CIO } 3102 \text { to }$ $\text { CIO } 3103$ | $\text { CIO } 3103 \text { to }$ $\text { CIO } 3105$ |  | CIO 3110 to CIO 3119 |
| Polled Unit No. 4 | CIO 3101 | $\begin{array}{\|l} \text { CIO } 3102 \text { to } \\ \text { CIO } 3103 \end{array}$ | $\begin{aligned} & \text { CIO } 3103 \text { to } \\ & \text { CIO } 3105 \end{aligned}$ |  | $\text { CIO } 3110 \text { to }$ $\text { CIO } 3119$ |
| Polled Unit No. 5 | CIO 3101 | $\text { CIO } 3102 \text { to }$ $\text { CIO } 3103$ | $\begin{aligned} & \hline \text { CIO } 3103 \text { to } \\ & \text { CIO } 3105 \\ & \hline \end{aligned}$ |  | CIO 3110 to CIO 3119 |
| Polled Unit No. 6 | CIO 3101 | $\begin{aligned} & \text { CIO } 3102 \text { to } \\ & \text { CIO } 3103 \end{aligned}$ | $\begin{aligned} & \text { CIO } 3103 \text { to } \\ & \text { CIO } 3105 \end{aligned}$ |  | $\begin{array}{\|l} \hline \text { CIO } 3110 \text { to } \\ \text { CIO } 3119 \\ \hline \end{array}$ |
| Polled Unit No. 7 | CIO 3101 | $\begin{array}{\|l} \hline \text { CIO } 3102 \text { to } \\ \text { CIO } 3103 \end{array}$ | $\begin{aligned} & \text { CIO } 3103 \text { to } \\ & \text { CIO } 3105 \end{aligned}$ |  | $\begin{aligned} & \text { CIO } 3110 \text { to } \\ & \text { CIO } 3119 \end{aligned}$ |
| Not used. | CIO 3102 to CIO 3199 | $\begin{array}{\|l} \hline \text { CIO } 3104 \text { to } \\ \text { CIO } 3199 \end{array}$ | $\begin{aligned} & \text { CIO } 3106 \text { to } \\ & \text { CIO } 3199 \end{aligned}$ |  | $\begin{aligned} & \text { CIO } 3120 \text { to } \\ & \text { CIO } 3199 \end{aligned}$ |

## Procedure

The Serial PLC Links operate according to the following settings in the PLC Setup in the Polling Unit and Polled Units.

## Settings at the Polling Unit

1,2,3... 1. Set the serial communications mode of serial port 1 or 2 to Serial PLC Links (Polling Unit).
2. Set the link method to the Complete Link Method or Polling Unit Link Method.
3. Set the number of link words (up to 10 words for each Unit).
4. Set the maximum unit number in the Serial PLC Links (0 to 7).

Settings at the Polled Units
1,2,3... 1. Set the serial communications mode of serial port 1 or 2 to Serial PLC Links (Polled Unit).
2. Set the unit number of the Serial PLC Link Polled Unit.

## PLC Setup

Settings at the Polling Unit

| Item |  | Set value | Default | $\begin{array}{\|l\|} \hline \text { Refresh timing } \\ \hline \text { Every cycle } \\ \hline \end{array}$ |
| :---: | :---: | :---: | :---: | :---: |
| Serial port 1 or 2 | Mode: Communications mode | PC Link (Master): PLC Link Polling Unit | Host Link |  |
|  | Baud: Baud rate | 38,400 bps, 115,200 bps | 9,600 bps |  |
|  | PC link mode: PLC Link method | ALL: Complete link method Masters: Polling Unit method | ALL |  |
|  | Link words: No. of link words | 1 to 10 words | 10 words |  |
|  | PC Link Unit No.: Max. unit No. | 0 to 7 | 0 hex |  |

Settings at the Polled Unit

| Item |  | Set value | Default | Refresh timing <br> Every cycle |
| :---: | :---: | :---: | :---: | :---: |
| Serial port 1 or 2 | Mode: Communications mode | PC Link (Slave): PLC Link Polled Unit | Host Link |  |
|  | Baud: Baud rate | 38,400 bps, 115,200 bps | 9,600 bps |  |
|  | Unit number | 0 to 7 | 0 |  |

Note Both serial ports cannot be used for PLC Links at the same time. If both ports are set for PLC Links (either as polling node or polled node), a PLC Setup setting error (non-fatal error) will occur and the PLC Setup Setting Error Flag (A402.10) will turn ON. If PLC Links is set for one serial port, set the other serial port to a different mode.

Related Auxiliary Area Flags for Serial Port 1 of an M-type CPU Unit

| Name | Address | Details | Read/write | Refresh timing |
| :---: | :---: | :---: | :---: | :---: |
| Serial Port 1 Communications Error Flag | A392.12 | Turns ON when a communications error occurs at serial port 1. <br> ON: Error <br> OFF: Normal | Read | - Cleared when power is turned ON. <br> - Turns ON when a communications error occurs at serial port 1. <br> - Turns OFF when the port is restarted. <br> - Disabled in peripheral bus mode and NT link mode. |
| Serial Port 1 Communicating with PT Flags (See note.) | $\begin{array}{\|l} \hline \text { A394.00 to } \\ \text { A394.07 } \end{array}$ | When serial port 1 is being used in NT link mode, the bit corresponding to the Unit performing communications will be ON. Bits 00 to 07 correspond to unit numbers 0 to 7, respectively. <br> ON: Communicating OFF: Not communicating | Read | - Cleared when power is turned ON. <br> - Turns ON the bit corresponding to the unit number of the PT/Polled Unit that is communicating via serial port 1 in NT link mode or Serial PLC Link mode. <br> - Bits 00 to 07 correspond to unit numbers 0 to 7, respectively. |
| Serial Port 1 Restart Bit | A526.01 | Turn ON this bit to restart serial port 1. | Read/write | - Cleared when power is turned ON. <br> - Turn ON to restart serial port 1, (except when communicating in peripheral bus mode). <br> Note: The bit is automatically turned OFF by the system when restart processing has been completed. |
| Serial Port 1 Error Flags | $\begin{array}{\|l} \hline \text { A528.08 to } \\ \text { A528.15 } \end{array}$ | When an error occurs at serial port 1, the corresponding error bit is turned ON. <br> Bit 08: Not used. <br> Bit 09: Not used. <br> Bit 10: Parity error <br> Bit 11: Framing error <br> Bit 12: Overrun error <br> Bit 13: Timeout error <br> Bit 14: Not used. <br> Bit 15: Not used. | Read/write | - Cleared when power is turned ON. <br> - When an error occurs at serial port 1, the corresponding error bit is turned ON. <br> - The flag is automatically turned OFF by the system when serial port 1 is restarted. <br> - Disabled during peripheral bus mode. <br> - In NT link mode, only bit 05 (timeout error) is enabled. <br> In Serial PLC Link mode, only the following bits are enabled. <br> - Errors at the Polling Unit: <br> Bit 05: Timeout error <br> - Errors at Polled Units: <br> Bit 05: Timeout error <br> Bit 04: Overrun error <br> Bit 03: Framing error |
| Serial Port 1 Settings Changed Flag | A619.01 | Turns ON when the communications conditions of serial port 1 are being changed. <br> ON: Changed OFF: No change | Read/write | - Cleared when power is turned ON. <br> - Turns ON while communications conditions settings for serial port 1 are being changed. <br> - Turns ON when the CHANGE SERIAL PORT SETUP instruction (STUP(237)) is executed. <br> - Turns OFF when the changes to settings are completed. |

Note In the same way as for the existing 1:N NT Link, the status (communicating/not communicating) of PTs in Serial PLC Links can be checked from the Polling Unit (CPU Unit) by reading the Serial Port 1 Communicating with PT Flag (A394 bits 00 to 07 for unit numbers 0 to 7 ).

## Related Auxiliary Area Flags for Serial Port 2 of an M-type CPU Unit

| Name | Address | Details | Read/write | Refresh timing |
| :---: | :---: | :---: | :---: | :---: |
| Serial Port 2 Communications Error Flag | A392.04 | Turns ON when a communications error occurs at Serial Port 2. <br> ON: Error <br> OFF: Normal | Read | - Cleared when power is turned ON. <br> - Turns ON when a communications error occurs at Serial Port 2. <br> - Turns OFF when the port is restarted. <br> - Disabled in peripheral bus mode and NT link mode. |
| Serial Port 2 Communicating with PT Flags (See note.) | $\begin{aligned} & \text { A393.00 to } \\ & \text { A393.07 } \end{aligned}$ | When Serial Port 2 is being used in NT link mode, the bit corresponding to the Unit performing communications will be ON. Bits 00 to 07 correspond to unit numbers 0 to 7, respectively. <br> ON: Communicating OFF: Not communicating | Read | - Cleared when power is turned ON. <br> - Turns ON the bit corresponding to the unit number of the PT/Polled Unit that is communicating via Serial Port 2 in NT link mode or Serial PLC Link mode. <br> - Bits 00 to 07 correspond to unit numbers 0 to 7, respectively. |
| Serial Port 2 Restart Bit | A526.00 | Turn ON this bit to restart Serial Port 2. | Read/write | - Cleared when power is turned ON. <br> - Turn ON to restart Serial Port 2, (except when communicating in peripheral bus mode). <br> Note: The bit is automatically turned OFF by the system when restart processing has been completed. |
| Serial Port 2 Error Flags | $\begin{aligned} & \text { A528.00 to } \\ & \text { A528.07 } \end{aligned}$ | When an error occurs at Serial Port 2, the corresponding error bit is turned ON. <br> Bit 00: Not used. <br> Bit 01: Not used. <br> Bit 02: Parity error <br> Bit 03: Framing error <br> Bit 04: Overrun error <br> Bit 05: Timeout error <br> Bit 06: Not used. <br> Bit 07: Not used. | Read/write | - Cleared when power is turned ON. <br> - When an error occurs at Serial Port 2, the corresponding error bit is turned ON. <br> - The flag is automatically turned OFF by the system when Serial Port 2 is restarted. <br> - Disabled during peripheral bus mode. <br> - In NT link mode, only bit 05 (timeout error) is enabled. <br> In Serial PLC Link mode, only the following bits are enabled. <br> - Errors at the Polling Unit: <br> Bit 05: Timeout error <br> - Errors at Polled Units: <br> Bit 05: Timeout error <br> Bit 04: Overrun error <br> Bit 03: Framing error |
| Serial Port 2 Settings Changed Flag | A619.02 | Turns ON when the communications conditions of Serial Port 2 are being changed. <br> ON: Changed OFF: No change | Read/write | - Cleared when power is turned ON. <br> - Turns ON while communications conditions settings for Serial Port 2 are being changed. <br> - Turns ON when the CHANGE SERIAL PORT SETUP instruction (STUP(237)) is executed. <br> - Turns OFF when the changes to settings are completed. |

Note In the same way as for the existing 1:N NT Link, the status (communicating/not communicating) of PTs in Serial PLC Links can be checked from the Polling Unit (CPU Unit) by reading the Serial Port 2 Communicating with PT Flag (A393 bits 00 to 07 for unit numbers 0 to 7 ).

Related Auxiliary Area Flags for Serial Port 1 of an L-type CPU Unit

| Name | Address | Details | Read/write | Refresh timing |
| :---: | :---: | :---: | :---: | :---: |
| Serial Port 1 Communications Error Flag | A392.04 | Turns ON when a communications error occurs at Serial Port 1. <br> ON: Error <br> OFF: Normal | Read | - Cleared when power is turned ON. <br> - Turns ON when a communications error occurs at Serial Port 1. <br> - Turns OFF when the port is restarted. <br> - Disabled in peripheral bus mode and NT link mode. |
| Serial Port 1 Communicating with PT Flags (See note.) | $\begin{aligned} & \text { A393.00 to } \\ & \text { A393.07 } \end{aligned}$ | When Serial Port 1 is being used in NT link mode, the bit corresponding to the Unit performing communications will be ON. Bits 00 to 07 correspond to unit numbers 0 to 7, respectively. <br> ON: Communicating OFF: Not communicating | Read | - Cleared when power is turned ON. <br> - Turns ON the bit corresponding to the unit number of the PT/Polled Unit that is communicating via Serial Port 1 in NT link mode or Serial PLC Link mode. <br> - Bits 00 to 07 correspond to unit numbers 0 to 7, respectively. |
| Serial Port 1 Restart Bit | A526.00 | Turn ON this bit to restart Serial Port 1. | Read/write | - Cleared when power is turned ON. <br> - Turn ON to restart Serial Port 1, (except when communicating in peripheral bus mode). <br> Note: The bit is automatically turned OFF by the system when restart processing has been completed. |
| Serial Port 1 Error Flags | $\begin{aligned} & \text { A528.00 to } \\ & \text { A528.07 } \end{aligned}$ | When an error occurs at Serial Port 1, the corresponding error bit is turned ON. <br> Bit 00: Not used. <br> Bit 01: Not used. <br> Bit 02: Parity error <br> Bit 03: Framing error <br> Bit 04: Overrun error <br> Bit 05: Timeout error <br> Bit 06: Not used. <br> Bit 07: Not used. | Read/write | - Cleared when power is turned ON. <br> - When an error occurs at Serial Port 1, the corresponding error bit is turned ON. <br> - The flag is automatically turned OFF by the system when Serial Port 1 is restarted. <br> - Disabled during peripheral bus mode. <br> - In NT link mode, only bit 05 (timeout error) is enabled. <br> In Serial PLC Link mode, only the following bits are enabled. <br> - Errors at the Polling Unit: <br> Bit 05: Timeout error <br> - Errors at Polled Units: <br> Bit 05: Timeout error <br> Bit 04: Overrun error <br> Bit 03: Framing error |
| Serial Port 1 Settings Changed Flag | A619.02 | Turns ON when the communications conditions of Serial Port 1 are being changed. <br> ON: Changed OFF: No change | Read/write | - Cleared when power is turned ON. <br> - Turns ON while communications conditions settings for Serial Port 1 are being changed. <br> - Turns ON when the CHANGE SERIAL PORT SETUP instruction (STUP(237)) is executed. <br> - Turns OFF when the changes to settings are completed. |

In the same way as for the existing 1:N NT Link, the status (communicating/not communicating) of PTs in Serial PLC Links can be checked from the Polling Unit (CPU Unit) by reading the Serial Port 1 Communicating with PT Flag (A393 bits 00 to 07 for unit numbers 0 to 7 ).

## 6-3-6 1:1 Links

## Applicable PLCs

## Connections

## PLC Setup

## Link Area Size

Operation

Two PLCs can be connected through their RS-232C ports to create Link Areas.

A 1:1 Link can be create between any of the following SYSMAC PLCs:
CP1L, CQM1H, C200HX/HG/HE(-Z), CPM1A-V1, CPM2A, CPM2B, CPM2C, and SRM1(-V2)

To create a 1:1 Link, connect the RS-232C ports on the two PLCs.


Set the PLC to a 1:1 Link Master or a 1:1 Link Slave in the PLC Setup. Set the other PLC to the opposite setting.


The 1:1 Link Area in the CP1L is from CIO 3000 to CIO 3015 (16 words). Even if a $1: 1$ Link is created with a CQM1H or C200HX/HG/HE(-Z) PLC, the 1:1 Link Area will be only 16 words on both sides of the link, and only LR 00 to LR 15 will be used in the CQM1H or C200HX/HG/HE(-Z) PLC. LR 16 to LR 63 cannot be used for 1:1 Links

Here, operation is described assuming that the master is the CP1L and the slave is the CPM2A.


1:1 Link Area


CP1L is set as the link master, so CIO 3000 to CIO 3007 are its write area. Any data written to these words with the OUT or MOV instructions will be automatically transferred to LR 00 to LR 07 in the CPM2A. The CPM2A will use these words as its read area.
CIO 3008 to 3015 are the read area of the CP1L. The contents of LR 08 to LR 15 in the CPM2A will automatically be transferred to CIO 3008 to 3015 in the CP1L. The words in the PLC's read area cannot be written using the OUT, MOV, or any other write instructions.

## 6-3-7 1:N NT Links

In the CP Series, communications are possible with PTs (Programmable Terminals) using NT Links in 1:N mode.


Note Communications are not possible using the 1:1-mode NT Link protocol.
High-speed NT Links are possible in addition to the previous standard NT Links by using the PT system menu and the following PLC Setup. High-speed NT Links are possible, however, only with NS-series PTs or with the NT31(C)V2 or NT631(C)-V2 PTs.

## PLC Setup



| Port | Name | Settings contents | Default values | Other conditions |
| :---: | :---: | :---: | :---: | :---: |
| Serial port 1 or 2 | Mode: Communications mode | NT Link (1:N): 1:N NT Links | Host Link | Turn OFF pin 4 on the CPU Unit DIP switch hen using serial port 1 and turn OFF pin 5 when using serial port 2. |
|  | Baud: Baud rate | 38,400 (standard) <br> 115,200 (high speed) | $\begin{aligned} & 9,600 \\ & \text { (disabled) } \end{aligned}$ |  |
|  | NT/PC Link Max: Highest unit number | 0 to 7 | 0 | --- |

## PT System Menu

Set the PT as follows:
1,2,3... 1. Select NT Link (1:N) from Comm. A Method or Comm. B Method on the Memory Switch Menu under the System Menu on the PT Unit.
2. Press the SET Touch Switch to set the Comm. Speed to High Speed.

## 6-3-8 1:1 NT Links

The NT Link communications protocol was developed to enable high-speed communications between PLC and Programmable Terminals (PTs). There are two communications modes supported by the NT Link protocol: 1:1 NT Links, in which one PLC is connected to one PT, and 1:N NT Links, in which one PLC is connected to more than one PT.

## Connections

OMRON PT


With the NT Link protocol, the PLC automatically responds to commands sent from the PT, so no communications programming is required in the CP1L.

## PLC Setup

Select "NT Link (1:1) as the serial communications mode.


## 6-3-9 Host Link Communications

The following table shows the host link communication functions available in CP1L PLCs. Select the method that best suits your application.

| Command flow | Command type | Communications method | Configuration |
| :---: | :---: | :---: | :---: |
|  | Host link command (C Mode) <br> Host link command | Create frame in the host computer and send the command to the PLC. Receive the response from the PLC. <br> Application: <br> Use this method when communicating primarily from the host computer to the PLC. | Directly connect the host computer in a 1:1 or 1:N system. |


| Command flow | Command type | Communications method | Configuration |
| :---: | :---: | :---: | :---: |
|  | FINS command (with Host Link header and terminator) sent. | Create frame in the host computer and send the command to the PLC. Receive the response from the PLC. <br> Application: <br> Use these methods when communicating primarily from the host computer to PLCs in the network. <br> Remarks: <br> The FINS command must be placed between a Host Link header and terminator and then sent by the host computer. | Directly connect the host computer in a $1: 1$ or 1:N system. |
| Host computer | FINS command (with Host Link header and terminator) is sent. | Send the command frame with the CPU Unit's SEND, RECV, or CMND instruction. Receive response from the host computer. <br> Application: <br> Use this method when communicating primarily from the PLC to the host computer to transmit status information, such as error information. <br> Remarks: <br> The FINS command will be placed between a Host Link header and terminator when it is sent. The FINS command must be interpreted at the host computer and then the host computer must return a response. | Directly connect the host computer in a $1: 1$ system. |

## Procedure



Host Link Commands

The following table lists the host link commands. Refer to the SYSMAC CS/CJ-series Communications Commands Reference Manual (W342) for more details.

| Type | Header <br> code | Name | Function |
| :---: | :--- | :--- | :--- |
| I/O mem- <br> ory read <br> commands | RR | CIO AREA READ | Reads the contents of the specified number of CIO Area words starting from <br> the specified word. |
|  | RL | LINK AREA READ | Reads the contents of the specified number of Link Area words starting from <br> the specified word. |
|  | RH | HR AREA READ | Reads the contents of the specified number of Holding Area words starting <br> from the specified word. |
|  | RC | PV READ | Reads the contents of the specified number of timer/counter PVs (present <br> values) starting from the specified timer/counter. |
|  | RG | T/C STATUS READ | Reads the status of the Completion Flags of the specified number of tim- <br> ers/counters starting from the specified timer/counter. |
|  | RD | DM AREA READ | Reads the contents of the specified number of DM Area words starting from <br> the specified word. |
|  | RJ | AR AREA READ | Reads the contents of the specified number of Auxiliary Area words starting <br> from the specified word. |


| Type | Header code | Name | Function |
| :---: | :---: | :---: | :---: |
| I/O memory write commands | WR | CIO AREA WRITE | Writes the specified data (word units only) to the CIO Area, starting from the specified word. |
|  | WL | LINK AREA WRITE | Writes the specified data (word units only) to the Link Area, starting from the specified word. |
|  | WH | HR AREA WRITE | Writes the specified data (word units only) to the Holding Area, starting from the specified word. |
|  | WC | PV WRITE | Writes the PVs (present values) of the specified number of timers/counters, starting from the specified timer/counter. |
|  | WD | DM AREA WRITE | Writes the specified data (word units only) to the DM Area, starting from the specified word. |
|  | WJ | AR AREA WRITE | Writes the specified data (word units only) to the Auxiliary Area, starting from the specified word. |
| Timer/ counter SV read commands | R\# | SV READ 1 | Reads the 4-digit BCD constant or word address in the SV of the specified timer/counter instruction. |
|  | R\$ | SV READ 2 | Searches for the specified timer/counter instruction beginning at the specified program address and reads the 4-digit constant or word address in the SV. |
|  | R\% | SV READ 3 | Searches for the specified timer/counter instruction beginning at the specified program address and reads the 4-digit BCD constant or word address in the SV. |
| Timer/ counter SV write commands | W\# | SV CHANGE 1 | Changes the 4-digit BCD constant or word address in the SV of the specified timer/counter instruction. |
|  | W\$ | SV CHANGE 2 | Searches for the specified timer/counter instruction beginning at the specified program address and changes the 4-digit constant or word address in the SV. |
|  | W\% | SV CHANGE 3 | Searches for the specified timer/counter instruction beginning at the specified program address and changes the 4-digit constant or word address in the SV. |
| CPU Unit status commands | MS | STATUS READ | Reads the operating status of the CPU Unit (operating mode, force-set/reset status, fatal error status). |
|  | SC | STATUS CHANGE | Changes the CPU Unit's operating mode. |
|  | MF | ERROR READ | Reads and clears errors in the CPU Unit (non-fatal and fatal). |
| Force-set/forcereset commands | KS | FORCE SET | Force-sets the specified bit. |
|  | KR | FORCE RESET | Force-resets the specified bit. |
|  | FK | MULTIPLE FORCE SET/RESET | Force-sets, force-resets, or clears the forced status of the specified bits. |
|  | KC | FORCE <br> SET/RESET CANCEL | Cancels the forced status of all force-set and force-reset bits. |
| Model read command | MM | PLC MODEL READ | Reads the model type of the PLC. |
| Test command | TS | TEST | Returns, unaltered, one block of data transmitted from the host computer. |
| Program area access commands | RP | PROGRAM READ | Reads the contents of the CPU Unit's user program area in machine language (object code). |
|  | WP | PROGRAM WRITE | Writes the machine language (object code) program transmitted from the host computer into the CPU Unit's user program area. |
| I/O memory compound read commands | QQMR | COMPOUND COMMAND | Registers the desired bits and words in a table. |
|  | QQIR | COMPOUND READ | Reads the registered words and bits from I/O memory. |


| Type | Header <br> code | Name | Function |
| :--- | :--- | :--- | :--- |
| Host Link <br> communi- <br> cations <br> processing <br> commands | XZ | ABORT (command <br> only) | Aborts the host link command that is currently being processed. |
|  | IC | INITIALIZE (com- <br> mand only) | Inditializes the transmission control procedure of all PLCs connected to the <br> host computer. |
| Undend com- <br> (response only) | This response is returned if the header code of a command was not recog- <br> nized. |  |  |

FINS Commands

| Type | Command code |  | Name | Function |
| :---: | :---: | :---: | :---: | :---: |
| I/O Memory Area Access Commands | 01 | 01 | MEMORY AREA READ | Reads consecutive data from the I/O memory area. |
|  | 01 | 02 | MEMORY AREA WRITE | Writes consecutive data to the I/O memory area. |
|  | 01 | 03 | MEMORY AREA FILL | Fills the specified range of I/O memory with the same data. |
|  | 01 | 04 | MULTIPLE MEMORY AREA READ | Reads non-consecutive data from the I/O memory area. |
|  | 01 | 05 | MEMORY AREA TRANSFER | Copies and transfers consecutive data from one part of the I/O memory area to another. |
| Parameter Area Access Commands | 02 | 01 | PARAMETER AREA READ | Reads consecutive data from the parameter area. |
|  | 02 | 02 | PARAMETER AREA WRITE | Writes consecutive data to the parameter area. |
|  | 02 | 03 | PARAMETER AREA FILL | Fills the specified range of the parameter area with the same data. |
| Program Area Access Commands | 03 | 06 | PROGRAM AREA READ | Reads data from the user program area. |
|  | 03 | 07 | PROGRAM AREA WRITE | Writes data to the user program area. |
|  | 03 | 08 | PROGRAM AREA CLEAR | Clears the specified range of the user program area. |
| Execution Control Commands | 04 | 01 | RUN | Switches the CPU Unit to RUN or MONITOR mode. |
|  | 04 | 02 | STOP | Switches the CPU Unit to PROGRAM mode. |
| Configuration Read Commands | 05 | 01 | CONTROLLER DATA READ | Reads CPU Unit information. |
|  | 05 | 02 | CONNECTION DATA READ | Reads the model numbers of the specified Units. |
| Status Read Commands | 06 | 01 | CONTROLLER STATUS READ | Reads the CPU Unit's status information. |
|  | 06 | 20 | CYCLE TIME READ | Reads the average, maximum, and minimum cycle times. |
| Clock Access Commands | 07 | 01 | CLOCK READ | Reads the clock. |
|  | 07 | 02 | CLOCK WRITE | Sets the clock. |
| Message Access Commands | 09 | 20 | MESSAGE READ/CLEAR | Reads/clears messages and FAL (FALS) messages. |
| Access Right Commands | OC | 01 | ACCESS RIGHT ACQUIRE | Acquires the access right if no other device holds it. |
|  | OC | 02 | ACCESS RIGHT FORCED ACQUIRE | Acquires the access right even if another device currently holds it. |
|  | OC | 03 | ACCESS RIGHT RELEASE | Releases the access right regardless of what device holds it. |
| Error Access Commands | 21 | 01 | ERROR CLEAR | Clears errors and error messages. |
|  | 21 | 02 | ERROR LOG READ | Reads the error log. |
|  | 21 | 03 | ERROR LOG CLEAR | Clears the error log pointer to zero. |
| Forced Status Commands | 23 | 01 | FORCED SET/RESET | Force-sets, force-resets, or clears the forced status of the specified bits. |
|  | 23 | 02 | FORCED SET/RESET CANCEL | Cancels the forced status of all force-set and force-reset bits. |

## 6-4 Analog Adjuster and External Analog Setting Input <br> 6-4-1 Analog Adjuster

By turning the analog adjuster on the CP1L CPU Unit with a Phillips screwdriver, the PV in the Auxiliary Area (A642) can be changed to any value within a range of 0 to 255 .


## Application Example

Setting the value for timer T100 in A642 makes it possible to use T100 as a variable timer with a range of 0 to 25.5 s ( 0 to 255). A change in the set value is reflected with the next scan.


Note Set values from the analog adjuster may vary with changes in the ambient temperature and the power supply voltage. Do not use it for applications that require highly precise set values.

## 6-4-2 External Analog Setting Input

When a voltage of 0 to 10 V is applied to the CP1L CPU Unit's external analog setting input terminal, the voltage is converted from analog to digital and the PV in A643 can be changed to any value within a range of 0 to 256 ( 0000 to 0100 hex).


## External Analog Setting Input Wiring

Use the 1-m lead wire (included) for wiring to the external analog setting input connector on the CP1L CPU Unit.


Relationship between Input Voltage and PV in A643

## Application Example



The maximum input voltage is 11 VDC. Do not apply a voltage greater than that.

Setting the value for timer T101 in A643 makes it possible to use T101 as a variable timer with a range of 0 to 25.6 s ( 0 to 256). A change in the set value is reflected with the next scan.


Note External analog setting input values may vary with changes in the ambient temperature. Do not use the external analog setting input for applications that require highly precise set values.

## 6-5 Battery-free Operation

## 6-5-1 Overview

With the CP1L CPU Unit, saving backup data in the built-in flash memory (non-volatile memory) enables operation with no battery mounted (i.e., bat-tery-free operation).
I/O memory (such as CIO ), however, is constantly refreshed during operation, so backup data is not saved in the built-in flash memory. When battery-free operation is used, therefore, programs must be created assuming that I/O memory data will not be saved.
For example, if a battery is mounted, then HR, CNT, and DM data is saved during power interruptions if a battery is mounted but not when battery-free operation is used.

In that case it is necessary to set the required values in the ladder program. It is also possible to save to the built-in flash memory in advance the DM initial values that are to be set for the DM on RAM at startup.

## 6-5-2 Using Battery-free Operation

Precautions when Creating Programs for Battery-free Operation

## Saving DM Initial Values (Only when Required)

Be careful of the following points, and create programs for which it will not be a problem even if the correct I/O memory values are not held.

- For unstable parts of I/O memory, include programming at the start of operation to set required data.
- When battery-free operation is used, the Output OFF Flag (A500.15) in the Auxiliary Area becomes unstable. When the Output OFF Flag turns ON, all outputs turn OFF, so include the following program for clearing the Output OFF Flag at the start of operation.

- Do not reference the clock function, (the clock data in words A351 to A354 of the Auxiliary Area, or the various kinds of time data).

Use the following procedure to save to the built-in flash memory the DM initial values that are to be set at startup.

1,2,3... 1. First set in the DM Area the data that is to be set as initial values at startup.
2. Execute a backup to flash memory from the CX-Programmer's Memory Cassette Transfer/Data Memory Backup Dialog Box.
The procedure is as follows:
a. Select PLC - Edit - Memory Cassette/DM.

The following Memory Cassette Transfer/DM Backup Dialog Box will be displayed.

b. Select the Data Memory Option in the Backup to Flash Memory Area and click the Backup Button.
The DM data will be written to the built-in flash memory.
Note The DM data that is saved and written at startup is the entire DM Area (D0 to D32767).

## PLC Setup

1,2,3... 1. Set Do not detect Low Battery (run without battery) to Do not detect.
2. Set IOM Hold Bit Status at Startup and Forced Status Hold Bit Status at Startup to Clear (OFF).
3. Set Read DM from flash memory to Read. (Only when DM initial values have been saved as described above.)

Caution The CP1L CPU Units automatically back up the user program and parameter data to flash memory when these are written to the CPU Unit. Also, the CXProgrammer can be used to save all of the data in the DM Area to the flash memory for use as initial values when the power supply is turned ON. Neither of these functions saves the I/O memory data (including HR Area data, counter PVs and Completion Flags, and DM Area data other than initial values). The HR Area data, counter PVs and Completion Flags, and DM Area data other than initial values are held during power interruptions with a battery. If there is a battery error, the contents of these areas may not be accurate after a power interruption. If HR Area data, counter PVs and Completion Flags, and DM Area data other than initial values are used to control external outputs, prevent inappropriate outputs from being made whenever the Battery Error Flag (A402.04) is ON.

## 6-6 Memory Cassette Functions

## 6-6-1 Overview

CP1L CPU Units have Memory Cassette functions that enable data in the CPU Unit to be stored on and read from a special CP1W-ME05M Memory Cassette. These functions can be used for the following applications.

- Copying data to other CPU Units to produce duplicate devices.
- Backing up data in case the CPU Unit needs to be replaced due to any malfunction.
- Writing and updating data when existing device versions are upgraded.

Memory Cassette Specifications

Use the following Memory Cassette.

| Model |  | Specifications |
| :---: | :---: | :---: |
| CP1W-ME05M | - Memory size | 512 Kwords |
|  | - Storage capacity | The following CPU Unit data (for each Unit) |
|  |  | - User programs |
|  |  | - Parameters |
|  |  | - Comment memory |
|  |  | - Function Block (FB) sources |
|  |  | - DM initial values in the built-in flash memory <br> - DM in RAM |
|  | -Write method | Operations from the CX-Programmer |
|  | - Read method | Powering up with DIP switch pin SW2 set to ON, or operations from the CX-Programmer |

Data that Can be Stored on a Memory Cassette

The following data can be stored on a Memory Cassette.

| Data stored on Memory Cassette |  | Location in CPU Unit |
| :--- | :--- | :--- |
| User programs | Built-in RAM, built-in flash <br> memory (User Program Area) |  |
| Parameters | PLC Setup, CPU Bus Unit set- <br> tings, routing tables | Built-in RAM, built-in flash <br> memory (Parameter Area) |
| Comment data <br> for user pro- <br> grams | Variable tables | Built-in flash memory (Com- <br> ment Memory Area) |
|  | (I/O comments, rung com- <br> ments, program comments) | Built-in flash memory (Com- <br> ment Memory Area) |
|  | Program indexes (section <br> names, section comments, <br> program comments) | Built-in flash memory (Com- <br> ment Memory Area) |
| Function Block (FB) sources | Built-in flash memory (FB <br> Source Memory Area) |  |
| DM | Built-in RAM (D0 to D32767 in <br> DM Area) |  |
| DM initial values (See note.) | Built-in flash memory (DM Ini- <br> tial Values Area) |  |

The areas for storing various types of data have fixed allocations in the Memory Cassette, and a single Memory Cassette corresponds to a single CPU Unit.
Therefore it is not possible to simultaneously store multiple items of the same type of data (e.g., two user programs).
Also, the data can only be read to a CPU Unit. It cannot be directly managed from a personal computer like files.
The only data that can be stored on a Memory Cassette is the data from a CPU Unit.

Note The CX-Programmer's function for saving DM initial values is used for saving the values in the DM Area (D0 to D32767) to the built-in flash memory as initial values. By means of a setting in the PLC Setup, these initial values can then be automatically written to the DM Area (D0 to D32767) when the power is turned ON.

## 6-6-2 Mounting and Removing a Memory Cassette

## Mounting

1,2,3... 1. Turn OFF the power supply to the PLC and removed the cover to the Memory Cassette socket.

2. Holding the Memory Cassette with the side with the nameplate facing upwards, insert the Memory Cassette all the way into the slot.

CPU Units with 10, 14 or 20 I/O Points


CPU Units with 30,40 or 60 I/O Points


## Removal

1,2,3... 1. Turn OFF the power supply to the PLC.
2. Grasp the end of the Memory Cassette between the thumbnail and index finger, and slide it upwards to remove it.


Note (1) Turn OFF the power supply before mounting or removing the Memory Cassette.
(2) Absolutely do not remove the Memory Cassette while the BKUP indicator is flashing (i.e., during a data transfer or verification). Doing so could make the Memory Cassette unusable.
(3) The Memory Cassette is small, so be careful to not let it be dropped or lost when it is removed.

## 6-6-3 Operation Using the CX-Programmer

Use the following procedure for the Memory Cassette function.

## 1,2,3... 1. Select PLC - Edit - Memory Cassette/DM. <br> The following Memory Cassette Transfer/Data Memory Backup Dialog Box will be displayed.


2. Under Transfer Data Area, check whatever types of data are to be transferred.

Click the Valid Area Check Button to check the valid areas in the Memory Cassette mounted in the CPU Unit and the operating mode after automatic transfer at startup. If the user program is specified to be written, select the operating mode after automatic transfer at startup.

- PROGRAM mode (default): Used, e.g., to copy the system.
- Use PLC Setup: Used, e.g., for operation with the Memory Cassette.

3. Execute any of the following operations.

- To write data from the CPU Unit to the Memory Cassette: Click the PLC $\Rightarrow$ Memory Cassette Button.
- To read data from the Memory Cassette to the CPU Unit: Click the Memory Cassette $\Rightarrow$ PLC Button.
- To verify data transferred between the CPU Unit and the Memory Cassette:
Click the Compare Button. This will cause all areas to be verified regardless of the items checked under Transfer Area.
- To format the Memory Cassette:

Click the Format Button. This will cause all areas to be formatted regardless of the items checked under Transfer Area.

## 6-6-4 Memory Cassette Data Transfer Function

Writing from the CPU Unit to the Memory Cassette

CPU Unit and Memory Cassette Verification

The CX-Programmer's Memory Cassette function can be used to write data from the CPU Unit to the Memory Cassette. The data to be written can be individually specified.


- When creating a Memory Cassette for a device version upgrade, select and save only the required data (such as the user program and DM).
- When creating a Memory Cassette for backup or duplication, save all of the data to the Memory Cassette.

When using the CX-Programmer's Memory Cassette function to store data in the Memory Cassette, verify that data by comparing it to the data in the CPU Unit. The data to be verified can be specified individually


This function can be used for operations such as confirmation after data has been written to the Memory Cassette, or confirming that the data in the backup matches the data in the CPU Unit.

With just a simple DIP switch setting, data stored in advance in the Memory Cassette can be automatically read when the power is turned ON, and written to the corresponding areas in the CPU Unit.
Mount a Memory Card and set DIP switch pin SW2 to ON, and then turn the power OFF and back ON.

All valid data in the Memory Card will be automatically transferred to the CPU Unit.

Note When this function is executed, at least the user program must be stored on the Memory Cassette.


This function can be used to copy data to another CPU Unit without using the CX-Programmer.


User programs can be overwritten to upgrade equipment versions without using the CX-Programmer.
If writing data from the CPU Unit to the Memory Cassette and the CPU Unit is set to use the operating mode specified in the PLC Setup as the operating mode after automatic transfer at startup, operation can be started without cycling the power, enabling operation from the Memory Cassette.

Reading Data from the Memory Cassette to the CPU Unit

Precautions when
Using the Memory Cassette Data Transfer Function

The CX-Programmer's Memory Cassette function can be used to read data stored on the Memory Cassette, and transfer it to the corresponding areas in the CPU Unit. The data to be read can be individually specified.


This function can be used for operations such as writing the required backup data to the CPU Unit for maintenance.

- In order for Memory Cassette data to be transferred, the Memory Cassette must be mounted in the CPU Unit.
- The BKUP indicator will light while data is being transferred to or verified in a Memory Cassette. Never turn OFF the power to the PLC or remove the Memory Cassette while the BKUP indicator is lit. Doing either may make it impossible to use the Memory Cassette.
- Memory Cassette data transfers and verification are possible only when the CPU Unit operating mode is PROGRAM mode. The Memory Cassette transfer function cannot be used in either RUN or MONITOR mode.
- When using automatic transfer from a Memory Cassette at startup, be sure to transfer the data to the Memory Cassette if any changes are made using online editing.
- The operating mode cannot be switched from PROGRAM mode to RUN or MONITOR mode while a Memory Cassette data transfer or verification is in progress.
- The following table shows whether data transfers are enabled when the CPU Unit is protected in various ways.

| Type of protection | Transfer from CPU Unit <br> to Memory Cassette | Transfer from Memory <br> Cassette to CPU Unit |
| :--- | :--- | :--- |
| Not protected. | Yes | Yes |
| System protected by DIP switch <br> pin SW1 set to ON. | Yes | No |
| Protected by password. Over- <br> writing and duplication both per- <br> mitted. | Yes | Yes |
| Protected by password. Over- <br> writing prohibited and duplica- <br> tion permitted. | Yes | Transfer enabled only at <br> startup. |


| Type of protection | Transfer from CPU Unit <br> to Memory Cassette | Transfer from Memory <br> Cassette to CPU Unit |
| :--- | :--- | :--- |
| Protected by password. Over- <br> writing permitted and duplica- <br> tion prohibited. | No | Yes |
| Protected by password. Over- <br> writing and duplication both pro- <br> hibited. | No | Transfer enabled only at <br> startup. |

- If a Memory Cassette is not mounted, data will be read from the flash memory built into the CPU Unit to start operation regardless of the setting of DIP switch pin SW2.
- CP1L CPU Units with 10,14 or 20 I/O points. do not have D10000 to D31999. These words will be treated as follows when data from a CPU Unit with 10, 14 or 20 I/O points is transferred to a CPU Unit with 30,40 or 60 I/O points or visa versa.

| Transferring data from a CPU Unit with <br> 10,14 or 20 I/O points to one with 30, <br> 40 or 60 I/O points | "0000" will be written to D10000 to D31999 <br> in the CPU Unit with 30,40 or 60 I/O points. |
| :--- | :--- |
| Transferring data from a CPU Unit with <br> 30,40 or 60 I/O points to one with 10, <br> 14 or 20 I/O points | D10000 to D31999 in the CPU Unit with 30, <br> 40 |

## 6-6-5 Procedures for Automatic Transfer from the Memory Cassette at Startup

## Copying the System

1,2,3... 1. Prepare a Memory Cassette containing the required data.
When transferring the data to the Memory Cassette, set the operating mode after automatic transfer at startup to PROGRAM mode (default).
2. With the power supply turned OFF to the CPU Unit, remove the cover from the Memory Cassette slot and insert the Memory Cassette.
3. Open the cover for the CPU Unit's PERIPHERAL section and set DIP switch pin SW2 to ON.

4. Turn ON the power supply to the CPU Unit.
5. The automatic transfer from the Memory Cassette will begin. The rest of the procedure assumes that the operating mode after automatic transfer at startup to PROGRAM mode (default).
6. After the automatic transfer has been completed, turn OFF the power supply to the CPU Unit.
7. Remove the Memory Cassette, and replace the Memory Cassette slot cover.
8. Return the setting of DIP switch pin SW2 to OFF, and close the cover.
9. Turn the power supply to the CPU Unit back ON.

Note After the automatic transfer from the Memory Cassette at startup has been completed with the operating mode after automatic transfer at startup set to PROGRAM mode (default), the transfer will not start again automatically (regardless of the Startup Mode setting in the PLC Setup). As described in the procedure above, to start operation turn the power supply OFF, return the setting of DIP switch SW2 to OFF, and then turn the power supply back ON. If the the operating mode specified in the PLC Setup is set as the operating mode after automatic transfer at startup, operation will start without changing the DIP switch SW2 or Memory Cassette.

1,2,3... 1. Prepare a Memory Cassette containing the required data. When transferring the data to the Memory Cassette, set the operating mode after automatic transfer at startup to PROGRAM mode (default).
2. With the power supply turned OFF to the CPU Unit, remove the cover from the Memory Cassette slot and insert the Memory Cassette.
3. Open the cover for the CPU Unit's PERIPHERAL section and set DIP switch pin SW2 to ON.
4. Turn ON the power supply to the CPU Unit.

Note If, when the data is transferred to the Memory Cassette, the operating mode specified in the PLC Setup is set as the operating mode after automatic transfer at startup, operation will start automatically after data transfer, even if the power is not cycled. Be sure that starting operation will cause no problems before using automatic transfer at startup.

## 6-7 Program Protection

The following protection functions are supported by the CP1L CPU Units.

- Read protection from the CX-Programmer
- Write protection using a DIP switch setting
- Write protection setting from the CX-Programmer
- Write protection against FINS commands sent to the CPU Unit via networks
- Prohibiting creating a program file for file memory


## 6-7-1 Read Protection

## Overview

It is possible to read-protect individual program tasks (called task read protection) or the entire user program (called UM read protection).

Read protection prevents anyone from displaying or editing the read-protected set of tasks or entire user program from CX-Programmer without inputting the correct password. If the password is input incorrectly five times consecutively, password input will be disabled for two hours, providing even better security for PLC data.

## Operating Procedure

1,2,3... 1. Go online and select PLC - Protection - Release Password. The following Release Read Protection Dialog Box will be displayed.

2. Input the password. If the password is incorrect, one of the following messages will be displayed and protection will not be released.

## UM Read Protection



Task Read Protection

3. If an incorrect password is input five times consecutively, read protection will not be released even if the correct password is input on the sixth attempt and displaying and editing the entire user program or the specified tasks will be disabled for two hours.

## Read Protection for Individual Tasks Using Passwords

## Overview

It is possible to read-protect individual program tasks (referred to as "task read protection" below) or the entire PLC. The same password controls access to all of the read-protected tasks.
Task read protection prevents anyone from displaying or editing the read-protected set of tasks from CX-Programmer without inputting the correct password. In this case, the entire program can be uploaded, but the read-protected tasks cannot be displayed or edited without inputting the correct password. Tasks that are not read-protected can be displayed, edited, or modified with online editing.
Note Task read protection cannot be set if UM read protection is already set. However, it is possible to set UM read protection after task read protection has been set.

CX-Programmer


## Operating Procedure

1,2,3... 1. Right-click the tasks that will be password-protected, select Properties from the pop-up menu, and select the Task read protect Option on the Protection Tab Page.

2. Display the Protection Tab of the PLC Properties Dialog Box and register a password in the Task read protection Box.

3. Connect online and select PLC - Transfer - To PLC to transfer the program. The tasks registered in step 2 will be password-protected.
Note The program can be transferred after step 1, above, and then password protection be set by selecting PLC - Protection - Set Password. The tasks registered in step 1 will be password-protected.

## Usage

Apply read protection to tasks when you want to convert those task programs to "black box" programs.


Note 1. If the CX-Programmer is used to read a task with task read protection applied, an error will occur and the task will not be read. Likewise, if the PT Ladder Monitor function is used to read a password protected task, an error will occur and the task will not be read.
2. The entire program can be transferred to another CPU Unit even if individual tasks in the program are read-protected. The task read protection will remain in effective for the password-protected tasks.
3. When the CX-Programmer is used to compare a user program in the computer's memory with a user program in the CPU Unit, password-protected tasks will be compared too.

## Restrictions to Function Block Use

## Prohibiting Backing Up the Programs to a Memory Cassette

Function block definitions can be read even if the entire program or individual tasks in a program containing function blocks are read-protected. If required, set read protection individually for each function block.

## Overview

When a password is set for the entire user program or for a task from the CXProgrammer, prohibiting backing up the user program can be set as an option. Doing so will make it impossible to upload PLC data to the CX-Programmer and make it impossible to save PLC data offline to a storage device.


## Operating Procedure

1,2,3... 1. When registering a password in the UM read protection password Box or Task read protection Box, select the Prohibit from saving to a memory card, and transferring program from PLC Option.

Prohibiting Creating Program Files in File Memory

2. Go online and then either select PLC - Transfer - To PLC to transfer the program or select PLC - Protection - Set Password and click the OK button.

## Application

The above procedure enables using a password to protect against disclosure of the program to unauthorized persons.
(1) Copying the program is possible if read protection is not set.
(2) The setting to prohibit backing up the program is not effective until the program is transferred to the PLC. Always transfer the program after changing the setting.

When a password is set for the entire user program or for a task from the CXProgrammer, prohibiting creating a program file (.OBJ) as a backup can be set as an option. Doing so will make it impossible to create a program file in file memory using the file memory operations. (This setting will also prohibit uploading PLC data to the CX-Programmer and saving PLC data to a storage device.)


## Operating Procedure

1,2,3... 1. When registering a password in the UM read protection password Box or Task read protection Box, select the Prohibit from saving to a memory card, and transferring program from PLC Option.

2. Go online and then either select PLC - Transfer - To PLC to transfer the program or select PLC - Protection - Set Password and click the OK button.

## Application

The above procedure enables using a password to protect against disclosure of the program to unauthorized persons.

Note (1) Copying the program is possible if read protection is not set.
(2) The setting to prohibit backing up the program is not effective until the program is transferred to the PLC. Always transfer the program after changing the setting.

## Auxiliary Area Flags and Bits Related to Password Protection

| Name | Bit address | Description |
| :---: | :---: | :---: |
| UM Read Protection Flag | A99.00 | Indicates whether or not the PLC (the entire user program) is read-protected. <br> OFF: UM read protection is not set. <br> ON: UM read protection is set. |
| Task Read Protection Flag | A99.01 | Indicates whether or not selected program tasks are read-protected. <br> OFF: Task read protection is not set. <br> ON: Task read protection is set. |
| Program Write Protection for Read Protection | A99.02 | Indicates whether or not the write protection option has been selected to prevent overwriting of password-protected tasks or programs. <br> OFF: Overwriting allowed <br> ON: Overwriting prohibited (write-protected) |
| Enable/Disable Bit for Program Backup | A99.03 | Indicates whether or not a backup program file (.OBJ file) can be created when UM read protection or task read protection is set. <br> OFF: Creation of backup program file allowed ON: Creation of backup program file prohibited |
| UM Read Protection Release Enable Flag | A99.12 | Indicates when UM read protection cannot be released because an incorrect password was input five times consecutively. <br> OFF: Protection can be released <br> ON: Protection cannot be released |
| Task Read Protection Release Enable Flag | A99.13 | Indicates when task read protection cannot be released because an incorrect password was input five times consecutively. <br> OFF: Protection can be released <br> ON: Protection cannot be released |

## 6-7-2 Write Protection

## Write-protection <br> Using the DIP Switch

The user program can be write-protected by turning ON pin 1 of the CPU Unit's DIP switch. When this pin is ON, it won't be possible to change the user program or parameter area (e.g., PLC Setup and routing tables) from the CXProgrammer. This function can prevent the program from being overwritten inadvertently at the work site.
It is still possible to read and display the program from the CX-Programmer when it is write-protected.

CPU Unit DIP Switch

| Pin | Name | Settings |
| :---: | :--- | :--- |
| SW1 | User Program Memory Write Protection | ON: Protected <br> OFF: Not protected |

## Confirming the User Program Date

The dates the program and parameters were created can be confirmed by checking the contents of A90 to A97.

## Auxiliary Area Words

| Name | Address | Description |  |
| :---: | :---: | :---: | :---: |
| User Program Date | A90 to A93 | The time and date the user program was last overwritten in memory is given in BCD. |  |
|  |  | A90.00 to A90.07 | Seconds (00 to 59 BCD) |
|  |  | A90.08 to A90.15 | Minutes (00 to 59 BCD) |
|  |  | A91.00 to A91.07 | Hour (00 to 23 BCD) |
|  |  | A91.08 to A91.15 | Day of month (01 to 31 BCD) |
|  |  | A92.00 to A92.07 | Month (01 to 12 BCD ) |
|  |  | A92.08 to A92.15 | Year (00 to 99 BCD) |
|  |  | A93.00 to A93.07 | Day (00 to 06 BCD) |
|  |  |  | Day of the week: <br> 00: Sunday, 01: Monday, <br> 02: Tuesday, 03: Wednesday, <br> 04: Thursday, 05: Friday, <br> 06: Saturday |
| Parameter Date | A94 to A97 | The time and date the parameters were last overwritten in memory is given in BCD. The format is the same as that for the User Program Date given above. |  |

## Write-protection Using Passwords

The program (or selected tasks) can also be write-protected if the write protection option is selected from the CX-Programmer when a password is being registered for the entire program or those selected tasks. The write protection setting can prevent unauthorized or accidental overwriting of the program.

CX-Programmer


Note 1. If the selected tasks are write-protected by selecting this option when registering a password, only the tasks (program) that are password-protected will be protected from overwriting. It will still be possible to overwrite other tasks with operations such as online editing and task downloading.
2. All tasks (programs) can be overwritten when program read protection is not enabled.

## Operating Procedure

1,2,3... Table 1 When registering a password in the UM read protection password Box or Task read protection Box, select the Prohibit from overwriting to a protected program Option.

3. Either select PLC - Transfer - To PLC to transfer the program or select PLC - Protection - Set Password and click the OK button.

Note The setting to enable/disable creating file memory program files will not take effect unless the program is transferred to the CPU Unit. Always transfer the program after changing this setting.

## Write Protection against FINS Commands Sent to the CPU Unit via Networks

It is possible to prohibit write operations and other editing operations sent to the PLC's CPU Unit as FINS commands through a network (including write operations from CX-Programmer, CX-Protocol, CX-Process, and other applications using Fins Gateway). Read processes are not prohibited.
FINS write protection can disable write processes such as downloading the user program, PLC Setup, or I/O memory, changing the operating mode, and performing online editing.
It is possible to exclude selected nodes from write protection so that data can be written from those nodes.

An event log in the CPU Unit automatically records all write processes sent through the network and that log can be read with a FINS command.

## 6-7-3 Protecting Program Execution Using the Lot Number

The lot number is stored in A310 and A311 and can be used to prevent the program from being executed on a CPU Unit with the wrong lot number.
The following instructions can be added to the program to create a fatal error and thus prevent program execution if an attempt is made to execute the program on a CPU Unit with the incorrect lot number. A password can also be set to read-protect the program so that it cannot be copied, e.g., using a Memory Cassette.
The lot number stored in A310 and A311 cannot be changed by the user.
The upper digits of the lot number are stored in A311 and the lower digits are stored in A310, as shown below.

Manufacturing lot
number ( 5 digits)

$X, Y$, and $Z$ in the lot number are converted to 10 , 11, and 12, respectively, in A310 and A311. Some examples are given below.

| Lot number | A311 | A310 |
| :--- | :--- | :--- |
| 01805 | 0005 | 0801 |
| 30 Y 05 | 0005 | 1130 |

Ladder Programming Example

- The following instructions will create a fatal error to prevent the program from being executed when the lot number is not 23905.

- The following instructions will create a fatal error to prevent the program from being executed when the lot number does not end in 05.

- The following instructions will create a fatal error to prevent the program from being executed when the lot number does not begin with $23 Y$.



## 6-8 Failure Diagnosis Functions

This section introduces the following functions.

- Failure Alarm Instructions: FAL(006) and FALS(007)
- Failure Point Detection: FPD(269)
- Output OFF Bit


## 6-8-1 Failure Alarm Instructions: FAL(006) and FALS(007)

The FAL(006) and FALS(007) instructions generate user-defined errors. FAL(006) generates a non-fatal error that allows program execution to continue and FALS(007) generates a fatal error that stops program execution.
When the user-defined error conditions (i.e., the execution conditions for FAL(006) or FAL(007)) are met, the instruction will be executed and the following processing will be performed.

1,2,3... 1. The FAL Error Flag (A402.15) or FALS Error Flag (A401.06) is turned ON.
2. The corresponding error code is written to A400.
3. The error code and time of occurrence are stored in the Error Log.
4. The error indicator on the front of the CPU Unit will flash or light.
5. If FAL(006) has been executed, the CPU Unit will continue operating. If FALS(007) has been executed, the CPU Unit will stop operating. (Program execution will stop.)

Operation of FAL(006)


When execution condition A goes ON, an error with FAL number 002 is generated, A402.15 (FAL Error Flag) is turned ON, and A360.02 (FAL Number 002 Flag) is turned ON. Program execution continues.
Errors generated by FAL(006) can be cleared by executing FAL(006) with FAL number 00 or performing the error read/clear operation from the CX-Programmer.

Operation of FALS(007)


When execution condition B goes ON, an error with FALS number 003 is generated, and A401.06 (FALS Error Flag) is turned ON. Program execution is stopped.
Errors generated by $\operatorname{FAL}(006)$ can be cleared by eliminating the cause of the error and performing the error read/clear operation from the CX-Programmer.

## 6-8-2 Failure Point Detection: FPD(269)

FPD(269) performs time monitoring and logic diagnosis. The time monitoring function generates a non-fatal error if the diagnostic output isn't turned ON within the specified monitoring time. The logic diagnosis function indicates which input is preventing the diagnostic output from being turned ON.

Time Monitoring Function

## Logic Diagnosis

 FunctionFPD(269) starts timing when it is executed and turns ON the Carry Flag if the diagnostic output isn't turned ON within the specified monitoring time. The Carry Flag can be programmed as the execution condition for an error processing block. Also, $\operatorname{FPD}(269)$ can be programmed to generate a non-fatal FAL error with the desired FAL number.
When an FAL error is generated, a preset message will be registered and can be displayed on the CX-Programmer. $\operatorname{FPD}(269)$ can be set to output the results of logic diagnosis (the address of the bit preventing the diagnostic output from being turned ON) just before the message.
The teaching function can be used to automatically determine the actual time required for the diagnostic output to go ON and set the monitoring time.
$\operatorname{FPD}(269)$ determines which input bit is causing the diagnostic output to remain OFF and outputs the result. The output can be set to bit address output (PLC memory address) or message output (ASCII).
If bit address output is selected, the PLC memory address of the bit can be transferred to an Index Register and the Index Register can be indirectly addressed in later processing.

If the message output is selected, an error message can be displayed on the CX-Programmer at the same time as a FAL error is generated for time monitoring.


## Time Monitoring

Monitors whether output $C$ goes $O N$ with 10 seconds after input $A$. If $C$ doesn't go ON within 10 seconds, a failure is detected and the Carry Flag is turned ON. The Carry Flag executes the error-processing block. Also, an FAL error (non-fatal error) with FAL number 004 is generated.

## Logic Diagnosis

FPD(269) determines which input bit in block $B$ is preventing output $C$ from going ON. That bit address is output to D1000 and D1001.

## Auxiliary Area Flags and Words

| Name | Address | Operation |
| :--- | :--- | :--- |
| Error Code | A400 | When an error occurs, the error code is stored in <br> A400. |
| FAL Error Flag | A402.15 | Turns ON when FAL(006) is executed. |
| FALS Error Flag | A401.06 | Turns ON when FALS(007) is executed. |
| Executed FAL Num- <br> ber Flags | A360 to <br> A391 | The corresponding flag turns ON when an <br> FAL(006) error occurs. |
| Error Log Area | A100 to <br> A199 | The Error Log Area contains information on the <br> most recent 20 errors. |
| Error Log Pointer | A300 | When an error occurs, the Error Log Pointer is <br> incremented by 1 to indicate where the next error <br> record will be recorded as an offset from the <br> beginning of the Error Log Area (A100). |
| Error Log Pointer | A500.14 | Turn this bit ON to reset the Error Log Pointer <br> (A300) to 00. |
| FPD Teaching Bit | A598.00 | Turn this bit ON when you want the monitoring <br> time to be set automatically when FPD(269) is <br> executed. |

## 6-8-3 Simulating System Errors

FAL(006) and FALS(007) can be used to intentionally create fatal and nonfatal system errors. This can be used in system debugging to test display messages on Programmable Terminals (PTs) or other operator interfaces. Use the following procedure.

1,2,3... 1. Set the FAL or FALS number to use for simulation in A529. A529 is used when simulating errors for both FAL(006) and FALS(007).
2. Set the FAL or FALS number to use for simulation as the first operand of FAL(006) or FALS(007).
3. Set the error code and error to be simulated as the second operand (two words) of FAL(006) or FALS(007). Indicate a nonfatal error for FAL(006) and a fatal error for FALS(007).
To simulate more than one system error, use more than one FAL(006) or FALS(007) instruction with the same value in A529 and different values for the second operand.

## Auxiliary Area Flags and Words

| Name | Address | Operation |
| :--- | :--- | :--- |
| FAL/FALS Number <br> for System Error <br> Simulation | A529 | Set a dummy FAL/FALS number to use to simu- <br> late a system error. |
|  |  | 0001 to 01FF hex: FAL/FALS numbers 1 to 511 <br> 0000 or 0200 to FFFF hex: No FAL/FALS number <br> for system error simulation. |

## Example for a Battery Error



Note Use the same methods as for actual system errors to clear the simulated system errors. Refer to the 11-2 Troubleshooting for details. All system errors simulated with FAL(006) and FALS(007) can be cleared by cycling the power supply.

## 6-8-4 Output OFF Bit

As an emergency measure when an error occurs, all outputs from Output Units can be turned OFF by turning ON the Output OFF Bit (A500.15). The operating mode will remain in RUN or MONITOR mode, but all outputs will be turned OFF.

Note Normally (when IOM Hold Bit = OFF), all outputs from Output Units are turned OFF when the operating mode is changed from RUN/MONITOR mode to PROGRAM mode. The Output OFF Bit can be used to turn OFF all outputs without switching to PROGRAM mode.

## Application Precaution for DeviceNet

inputs to the master will be OFF.

## 6-9 Clock

A clock is built into the CP1L CPU Unit and is backed up by a battery. The current data is stored in the following words and refreshed each cycle.

| Name | Addresses | Function |
| :--- | :--- | :--- |
| Clock data: <br> A351 to A354 | A351.00 to A351.07 | Second: 00 to 59 (BCD) |
|  | A351.08 to A351.15 | Minute: 00 to 59 (BCD) |
|  | A352.00 to A352.07 | Hour: 00 to 23 (BCD) |
|  | A352.08 to A352.15 | Day of the month: 00 to 31 (BCD) |
|  | A353.00 to A353.07 | Month: 00 to 12 (BCD) |
|  | A353.08 to A353.15 | Year: 00 to 99 (BCD) |
|  | A354.00 to A354.07 | Day of the week: <br> 00: Sunday, 01: Monday, <br> 02: Tuesday, 03: Wednesday, <br> 04: Thursday, 05: Friday, 06: Saturday |

Note The clock cannot be used if a battery is not installed or the battery voltage is low.

## Auxiliary Area Flags and Words

| Name | Addresses | Contents |
| :---: | :---: | :---: |
| Start-up Time | A510 and A511 | The time at which the power was turned ON (year, month, day of month, hour, minutes, and seconds). |
| Power Interruption Time | A512 and A513 | The time at which the power was last interrupted (year, month, day of month, hour, minutes, and seconds). |
| Power ON Clock Data 1 | A720 to A722 | Consecutive times at which the power was turned ON (year, month, day of month, hour, minutes, and seconds). The times are progressively older from number 1 to number 10. |
| Power ON Clock Data 2 | A723 to A725 |  |
| Power ON Clock Data 3 | A726 to A728 |  |
| Power ON Clock Data 4 | A729 to A731 |  |
| Power ON Clock Data 5 | A732 to A734 |  |
| Power ON Clock Data 6 | A735 to A737 |  |
| Power ON Clock Data 7 | A738 to A740 |  |
| Power ON Clock Data 8 | A741 to A743 |  |
| Power ON Clock Data 9 | A744 to A746 |  |
| Power ON Clock Data 10 | A747 to A749 |  |
| Operation Start Time | A515 to A517 | The time that operation started (year, month, day of month, hour, minutes, and seconds). |
| Operation End Time | A518 to A520 | The time that operation stopped (year, month, day of month, hour, minutes, and seconds). |
| User Program Date | A90 to A93 | The time when the user program was last overwritten (year, month, day of month, hour, minutes, and seconds). |
| Parameter Date | A94 to A97 | The time when the parameters were last overwritten (year, month, day of month, hour, minutes, and seconds). |

## Time-related Instructions

| Name | Mnemonic | Function |
| :--- | :--- | :--- |
| HOURS TO SECONDS | SEC(065) | Converts time data in hours/minutes/sec- <br> onds format to an equivalent time in seconds <br> only. |
| SECONDS TO HOURS | HMS(066) | Converts seconds data to an equivalent time <br> in hours/minutes/seconds format. |
| CALENDAR ADD | CADD(730) | Adds time to the calendar data in the speci- <br> fied words. |
| CALENDAR SUBTRACT | CSUB(731) | Subtracts time from the calendar data in the <br> specified words. |
| CLOCK ADJUSTMENT | DATE(735) | Changes the internal clock setting to the set- <br> ting in the specified source words. |

This section describes how to use CP-series/CPM1A-series Expansion Units and Expansion I/O Units.
7-1 Connecting Expansion Units and Expansion I/O Units . . . . . . . . . . . . . . . . 422
7-2 Analog Input Units . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 423
7-3 Analog Output Units . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 432
7-4 Analog I/O Units . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 440
7-4-1 CPM1A-MAD01 Analog I/O Units . . . . . . . . . . . . . . . . . . . . . . . . . . 440
7-4-2 CP1W-MAD11/CPM1A-MAD11 Analog I/O Units . . . . . . . . . . . . 450
7-5 Temperature Sensor Units . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 463
7-6 CompoBus/S I/O Link Units . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 48
7-7 DeviceNet I/O Link Units . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 484

## 7-1 Connecting Expansion Units and Expansion I/O Units

CP-series and CPM1A-series Expansion Units and Expansion I/O Units can be connected to the CP1L. Up to three Expansion Units or Expansion I/O Units can be connected to a CPU Unit with 30,40 or 60 I/O points and one Expansion Unit or Expansion I/O Unit can be connected to a CPU Unit with 20 or 14 I/O points.
The functionality and performance of CP-series Expansion Units and Expansion I/O Units is the same as the functionality and performance of CP1MAseries Expansion Units and Expansion I/O Units. CP-series Units are black, and CPM1A-series Units are ivory.

## Number of I/O Words

| Unit name |  | Model | Currentconsumption (mA) |  | I/O words |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | 5 VDC | 24 VDC | Input | Output |
| Expansion Units | Analog Input Unit |  | CP1W-AD041 CPM1A-AD041 | 100 | 90 | 4 | 2 |
|  | Analog Output Unit | CP1W-DA021 | 40 | 95 | --- | 2 |
|  |  | $\begin{aligned} & \text { CP1W-DA041 } \\ & \text { CPM1A-DA041 } \end{aligned}$ | 80 | 124 | --- | 4 |
|  | Analog I/O Unit | CPM1A-MAD01 | 66 | 66 | 2 | 1 |
|  |  | CP1W-MAD11 CPM1A-MAD11 | 83 | 110 |  |  |
|  | Temperature Control Unit | $\begin{aligned} & \text { CP1W-TS001 } \\ & \text { CPM1A-TS001 } \end{aligned}$ | 40 | 59 | 2 | --- |
|  |  | $\begin{aligned} & \text { CP1W-TS101 } \\ & \text { CPM1A-TS101 } \end{aligned}$ | 54 | 73 |  |  |
|  |  | CP1W-TS002 CPM1A-TS002 | 40 | 59 | 4 | --- |
|  |  | $\begin{aligned} & \text { CP1W-TS102 } \\ & \text { CPM1A-TS102 } \end{aligned}$ | 54 | 73 |  |  |
|  | CompoBus/S I/O Link Unit | CP1W-SRT21 CPM1A-SRT21 | 29 | --- | 1 | 1 |
|  | DeviceNet I/O Link Unit | CPM1A-DRT21 | 48 | -- | 2 | 2 |
| Expansion I/O Units | 40-point I/O Unit | $\begin{aligned} & \text { CP1W-40EDR } \\ & \text { CPM1A-40EDR } \end{aligned}$ | 80 | 90 | 2 | 2 |
|  |  | CP1W-40EDT CPM1A-40EDT | 160 | --- |  |  |
|  |  | CP1W-40EDT1 CPM1A-40EDT1 | 160 | --- |  |  |
|  | 32-point Output Unit | CP1W-32ER | 49 | 131 | --- | 4 |
|  |  | CP1W-32ET | 113 | --- |  |  |
|  |  | CP1W-32ET1 | 113 | --- |  |  |
|  | 20-point I/O Unit | $\begin{aligned} & \text { CP1W-20EDR1 } \\ & \text { CPM1A-20EDR1 } \end{aligned}$ | 103 | 44 | 1 | 1 |
|  |  | $\begin{aligned} & \text { CP1W-20EDT } \\ & \text { CPM1A-20EDT } \end{aligned}$ | 130 | --- |  |  |
|  |  | CP1W-20EDT1 CPM1A-20EDT1 | 130 | --- |  |  |
|  | 16-point Output Unit | $\begin{aligned} & \text { CP1W-16ER } \\ & \text { CPM1A-16ER } \end{aligned}$ | 42 | 90 | --- | 2 |
|  |  | CP1W-16ET | 76 | --- |  |  |
|  |  | CP1W-16ET1 | 76 | --- |  |  |
|  | 8-point Input Unit | $\begin{aligned} & \text { CP1W-8ED } \\ & \text { CPM1A-8ED } \end{aligned}$ | 18 | --- | 1 | --- |
|  | 8-point Output Unit | CP1W-8ER CPM1A-8ER | 26 | 44 | --- | 1 |
|  |  | CP1W-8ET CPM1A-8ET | 75 | --- |  |  |
|  |  | CP1W-8ET1 CPM1A-8ET1 | 75 | --- |  |  |

Note CP1W-32ER/32ET/32ET1's maximum number of simultaneously ON points is 24 (75\%).

## Allocation of I/O Words

Expansion Units and Expansion I/O Units are allocated I/O bits in the order the Units are connected starting from the CPU Unit. When the power to the CPU Unit is turned ON, the CPU Unit checks for any Expansion Units and Expansion I/O Units connected to it and automatically allocates I/O bits


## 7-2 Analog Input Units

Each CP1W-AD041/CPM1A-AD041 Analog Input Unit provides four analog inputs.

- The analog input signal ranges are 0 to 5 V , 1 to $5 \mathrm{~V}, 0$ to $10 \mathrm{~V},-10$ to +10 V , 0 to 20 mA , and 4 to 20 mA . The resolution is $1 / 6,000$. The open-circuit detection function is activated in the ranges of 1 to 5 V and 4 to 20 mA .
- The Analog Input Unit uses four input words and two output words, so a maximum of three Units can be connected.

Part Names

## CP1W-AD041/CPM1A-AD041



1. Analog Input Terminals Connected to analog output devices.

■ Input Terminal Arrangement


| V IN1 | Voltage input 1 |
| :--- | :--- |
| I IN1 | Current input 1 |
| COM1 | Input common 1 |
| V IN2 | Voltage input 2 |
| I IN2 | Current input 2 |
| COM2 | Input common 2 |
| V IN3 | Voltage input 3 |
| I IN3 | Current input 3 |
| COM3 | Input common 3 |
| V IN4 | Voltage input 4 |
| I IN4 | Current input 4 |
| COM4 | Input common 4 |

Note When using current inputs, voltage input terminals must be short-circuited with current input terminals.
2. Expansion I/O Connecting Cable

Connected to the CPU Unit or Expansion Unit expansion connector. The cable is attached to the Analog Input Unit and cannot be removed.

Note Do not touch the cables during operation. Static electricity may cause operating errors.
3. Expansion Connector

Connected to the next Expansion Unit or Expansion I/O Unit to enable expansion.

## Main Analog Input Unit Specifications

Analog Input Units are connected to a CP1L CPU Unit. For CP1L M-type CPU Units, a maximum of three Units can be connected, including other Expansion Units and Expansion I/O Units. For CP1L L-type CPU Units, a maximum of one Unit can be connected.


| Item |  | Voltage Input | Current Input |
| :---: | :---: | :---: | :---: |
| Number of inputs |  | 4 inputs (4 words allocated) |  |
| Input signal range |  | 0 to 5 VDC, 1 to 5 VDC, 0 to 10 VDC, or -10 to 10 VDC | 0 to 20 mA or 4 to 20 mA |
| Max. rated input |  | $\pm 15 \mathrm{~V}$ | $\pm 30 \mathrm{~mA}$ |
| External input impedance |  | $1 \mathrm{M} \Omega \mathrm{min}$. | Approx. 250 ת |
| Resolution |  | 1/6000 (full scale) |  |
| Overall accuracy | $25^{\circ} \mathrm{C}$ | 0.3\% full scale | 0.4\% full scale |
|  | 0 to $55^{\circ} \mathrm{C}$ | 0.6\% full scale | 0.8\% full scale |
| A/D conversion data |  | 16-bit binary (4-digit hexadecimal) <br> Full scale for -10 to 10 V: F448 to 0BB8 Hex <br> Full scale for other ranges: 0000 to 1770 Hex |  |
| Averaging function |  | Supported (Set in output words $\mathrm{n}+1$ and $\mathrm{n}+2$. ) |  |
| Open-circuit detection function |  | Supported |  |
| Conversion time |  | $2 \mathrm{~ms} /$ point (8 ms/all points) |  |
| Isolation method |  | Photocoupler isolation between analog I/O terminals and internal circuits. No isolation between analog I/O signals. |  |
| Current consumption |  | 5 VDC: 100 mA max.; 24 VDC : 90 mA max. |  |

Analog Input Signal Ranges

Analog input data is digitally converted according to the input signal range as shown below.

Note When the input exceeds the specified range, the A/D conversion data will be fixed at either the lower limit or upper limit.

## Analog Input Signal

## Ranges

## -10 to 10 V Inputs



0 to 10 V Inputs


Voltage in the 0 to 10 V range corresponds to hexadecimal values 0000 to 1770 ( 0 to 6,000 ). The range of data that can be converted is FED4 to 189C hex ( -300 to 6,300 ). A negative voltage is expressed as two's complement.

## 0 to 5 V Inputs



Voltage in the 0 to 5 V range corresponds to hexadecimal values 0000 to 1770 ( 0 to $6,000)$. The range of data that can be converted is FED4 to 189C hex ( -300 to 6,300 ). A negative voltage is expressed as two's complement.

## 1 to 5 V Inputs



Voltage in the 1 to 5 V range corresponds to hexadecimal values 0000 to 1770 ( 0 to 6,000 ). The range of data that can be converted is FED4 to 189C hex (-300 to 6,300). Voltage in the range of 0.8 to 1 V is expressed as two's complement.
If an input is below the range (i.e., less than 0.8 V ), the open-circuit detection function is activated and the data becomes 8,000.

## 0 to 20 mA Inputs



Current in the 0 to 20 mA range corresponds to hexadecimal values 0000 to 1770 ( 0 to 6,000 ). The range of data that can be converted is FED4 to 189C hex (-300 to 6,300 ). A negative current is expressed as two's complement.

4 to 20 mA Inputs


Current in the 4 to 20 mA range corresponds to hexadecimal values 0000 to 1770 ( 0 to 6,000 ). The range of data that can be converted is FED4 to 189C hex (-300 to 6,300 ). Current in the range of 3.2 to 4 mA is expressed as two's complement.
If an input is below the range (i.e., less than 3.2 mA ), the open-circuit detection function is activated and the data becomes 8,000.

## Averaging Function

Open-circuit Detection Function

For analog inputs, the averaging function operates when the averaging bit is set to 1 . The averaging function outputs the average (a moving average) of the last eight input values as the converted value. If there is only a slight variation in inputs, it is handled by the averaging function as a smooth input.
The averaging function stores the average (a moving average) of the last eight input values as the converted value. Use this function to smooth inputs that vary at a short interval.

The open-circuit detection function is activated when the input range is set to 1 to 5 V and the voltage drops below 0.8 V , or when the input range is set to 4 to 20 mA and the current drops below 3.2 mA . When the open-circuit detection function is activated, the converted data will be set to 8,000 .
The time for enabling or clearing the open-circuit detection function is the same as the time for converting the data. If the input returns to the convertible range, the open-circuit detection is cleared automatically and the output returns to the normal range.

## Procedure



- Connect Analog Input Units.
- Wire to analog output devices.
- Write set data to output words ( $\mathrm{n}+1, \mathrm{n}+2$ ).
- Set use of inputs.
- Select input signals using range codes.
- Set use of averaging.
- Read A/D conversion values from input words ( $m+1$ to $m+4$ ).
- For current inputs, confirm that there is no open circuit.


## Writing Set Data and

 Reading A/D Conversion Values

Analog devices Temperature sensor Pressure sensor Speed sensor Flow sensor Voltage/current meter Other devices

## 1. Connecting the Analog Input Unit

Connect the Analog Input Unit to the CPU Unit.


## 2. Wiring Analog Inputs

## Internal Circuits



- Wiring for Analog Inputs


Note (1) Connect the shield to the FG terminal to prevent noise.
(2) When an input is not being used, short the + and - terminals.
(3) Separate wiring from power lines (AC power supply lines, high-voltage lines, etc.)
(4) When there is noise in the power supply line, install a noise filter on the input section and the power supply.
(5) Refer to the following information on open circuits when using voltage inputs.


For example, if analog input device 2 is outputting 5 V and the same power supply is being used as shown above, about $1 / 3$, or 1.6 V , will be applied at the input for input device 1.
Consider the following information on open input circuits when using voltage inputs. Either use separate power supplies, or install an isolator at each input. If the same power supply is used as shown in the following diagram and an open circuit occurs at point $A$ or $B$, an unwanted current flow will occur as shown by the dotted lines in the diagram, creating a voltage at the other input of about $1 / 3$ to $1 / 2$. If the 1 to $5-\mathrm{V}$ range is being used, the open-circuit detection function will not operate. Also, if there is an open circuit at $C$, the open-circuit detection function will not operate because the negative sides are the same.

## 3. Creating the Ladder Program

Allocating I/O Words
Four input words and two output words are allocated from the next words following the last I/O words allocated to the CPU Unit or an existing Expansion Unit or Expansion I/O Unit.


## Writing Set Data

Write the settings for input use, averaging use, and range codes for words $\mathrm{n}+1$ and $n+2$. When the set data is transferred from the CPU Unit to the Analog I/O Unit, the A/D conversion will be started.

## Averaging

Reading Analog Input Conversion Values

Handling Unit Errors

Wd ( $\mathrm{n}+1$ )


## - Set Data



- The Analog Input Unit will not start converting analog I/O values until the range code has been written.
- Once the range code has been set, it is not possible to change the setting while power is being supplied to the CPU Unit. To change the I/O range, turn the CPU Unit OFF then ON again.

Set whether averaging is to be used for set data. When the averaging bit is set to 1 , the average (moving average) for the past eight inputs is output as conversion data.

Read the conversion value storage area with the ladder program. With word $m$ as the last input word allocated to the CPU Unit or an already-connected Expansion Unit, the A/D conversion data will be output to the following words $\mathrm{m}+1$ to $\mathrm{m}+4$.

After the power is turned ON , it will require two cycle times plus approximately 50 ms before the first conversion data is stored in the input words. Therefore, create a program as shown below, so that when operation begins simultaneously with startup it will wait for valid conversion data.
The analog input data will be 0000 until the initial processing is completed.


TIM5 is started when the power is turned ON. After 0.1 to 0.2 s (100 to 200 ms ) elapses, the TIM5 contact turns ON and the analog input 1 conversion data stored in word 2 is transferred to DMO.

- When an error occurs in an Analog Input Unit, the analog input conversion data becomes 0000.


## Ladder Program Example

| Analog <br> input | Input range | Range code | Averaging | Set data | Destination <br> word |
| :--- | :--- | :--- | :--- | :--- | :--- |
| Input 1 | 0 to 10 V | 01 | Yes | $1101(\mathrm{D} \mathrm{hex})$ | $\mathrm{n}+1$ |
| Input 2 | 4 to 20 mA | 10 | Yes | $1110(\mathrm{E}$ hex $)$ | $\mathrm{n}+1$ |
| Input 3 | -10 to +10 V | 00 | No | $1000(8$ hex $)$ | $\mathrm{n}+2$ |
| Input 4 | Not used. | $-(00)$ | --- | $0000(0$ hex) | $\mathrm{n}+2$ |



## 7-3 Analog Output Units

Each CP1W-DA021 Analog Output provides two analog outputs.
Each CP1W-DA041/CPM1A-DA041 Analog Output Unit provides four analog outputs.

- The analog output signal ranges are 1 to $5 \mathrm{~V}, 0$ to $10 \mathrm{~V},-10$ to $+10 \mathrm{~V}, 0$ to 20 mA , and 4 to 20 mA . The resolution is $1 / 6,000$.
- The CP1W-DA021 uses two output words and the CP1W-DA041/CPM1ADA041 uses four output words, so a maximum of three Units can be connected.


1. Analog Output Terminals Connected to analog input devices.

■ Output Terminal Arrangement for CP1W-DA041/CPM1A-DA041


| V OUT1 | Voltage output 1 |
| :--- | :--- |
| I OUT1 | Current output 1 |
| COM1 | Output common 1 |
| V OUT2 | Voltage output 2 |
| I OUT2 | Current output 2 |
| COM2 | Output common 2 |
| V OUT3 | Voltage output 3 |
| I OUT3 | Current output 3 |
| COM3 | Output common 3 |
| V OUT4 | Voltage output 4 |
| I OUT4 | Current output 4 |
| COM4 | Output common 4 |

Output Terminal Arrangement for CP1W-DA021


| V OUT1 | Voltage output 1 |
| :--- | :--- |
| I OUT1 | Current output 1 |
| COM1 | Output common 1 |
| V OUT2 | Voltage output 2 |
| I OUT2 | Current output 2 |
| COM2 | Output common 2 |

Main Analog Output Unit Specifications
2. Expansion I/O Connecting Cable Connected to the CPU Unit or previous Expansion Unit. The cable is provided with the Unit and cannot be removed.
Note Do not touch the cables during operation. Static electricity may cause operating errors.
3. Expansion Connector Connected to the next Expansion Unit or Expansion I/O Unit.

Analog Output Units are connected to a CP1L CPU Unit. For CP1L M-type CPU Units, a maximum of three Units can be connected, including other Expansion Units and Expansion I/O Units. For CP1L L-type CPU Units, a maximum of one Unit can be connected.


| Item |  |  | Voltage Output | Current Output |
| :---: | :---: | :---: | :---: | :---: |
| Analog output section | Number of outputs |  | CP1W-DA021: 2 outputs (2 words allocated) CP1W/CPM1A-DA041: 4 outputs (4 words allocated) |  |
|  | Output signal range |  | 1 to $5 \mathrm{VDC}, 0$ to 10 VDC , or -10 to 10 VDC | 0 to 20 mA or 4 to 20 mA |
|  | External output allowable load resistance |  | $2 \mathrm{k} \Omega \mathrm{min}$. | $350 \Omega$ max. |
|  | External output impedance |  | $0.5 \Omega$ max. | --- |
|  | Resolution |  | 1/6000 (full scale) |  |
|  | Overall accuracy | $25^{\circ} \mathrm{C}$ | 0.4\% full scale |  |
|  |  | 0 to $55^{\circ} \mathrm{C}$ | 0.8\% full scale |  |
|  | D/A conversion data |  | 16-bit binary (4-digit hexadecimal) <br> Full scale for -10 to 10 V: F448 to 0BB8 Hex Full scale for other ranges: 0000 to 1770 Hex |  |
| Conversion time |  |  | CP1W-DA021: $2 \mathrm{~ms} /$ point ( $4 \mathrm{~ms} /$ all points) CP1W/CPM1A-DA041: $2 \mathrm{~ms} /$ point ( $8 \mathrm{~ms} /$ all points) |  |
| Isolation method |  |  | Photocoupler isolation between analog I/O terminals and internal circuits. No isolation between analog I/O signals. |  |
| Current consumption |  |  | CP1W-DA021: 5 VDC 40 mA max.; 24 VDC 95 mA max. CP1W/CPM1A-DA041: 5 VDC 80 mA max.; 24 VDC 124 mA max. |  |

## Analog Output Signal

 RangesThe analog values depend on the output signal ranges, as shown in the following diagrams.

Note When the output exceeds the specified range, the output signal will be fixed at either the lower limit or upper limit.

## Analog Output Signal

## Ranges

- $\mathbf{- 1 0}$ to 10 V

The hexadecimal values F448 to OBB8 ( -3000 to 3000) correspond to an analog voltage range of -10 to 10 V . The entire output range is -11 to 11 V . Specify a negative voltage as a two's complement.


## 0 to 10 V

The hexadecimal values 0000 to 1770 ( 0 to 6000) correspond to an analog voltage range of 0 to 10 V . The entire output range is -0.5 to 10.5 V . Specify a negative voltage as a two's complement.


## 1 to 5 V

The hexadecimal values 0000 to 1770 ( 0 to 6000) correspond to an analog voltage range of 1 to 5 V . The entire output range is 0.8 to 5.2 V .


## 0 to 20 mA

The hexadecimal values 0000 to 1770 ( 0 to 6000) correspond to an analog current range of 0 to 20 mA . The entire output range is 0 to 21 mA .


## 4 to 20 mA

The hexadecimal values 0000 to 1770 ( 0 to 6000) correspond to an analog current range of 4 to 20 mA . The entire output range is 3.2 to 20.8 mA .


## Procedure



- Connect Analog Output Units.
- Wire to analog input devices.
- Write range code to output words. CP1W-DA041/CPM1A-DA041: Words n+1, n+2 CP1W-DA021: Word n+1
- Set use of outputs.
- Select output signals using range codes.
- Write D/A conversion values to output words. CP1W-DA041/CPM1A-DA041: words $n+1$ to $n+4$ CP1W-DA021: Words n+1, n+2


## Writing D/A Conversion Data

## ■ CP1W-DA041/CPM1A-DA041



Analog devices Adjustment equipment Servo Controller Variable speed device Recorder Other

## CP1W-DA021



## 1. Connecting the Analog Output Unit

Connect the Analog Output Unit to the CPU Unit.


## 2. Wiring Analog Outputs

Internal Circuits
The following diagram shows the internal circuit using CP1W-DA041/CPM1ADA041 as an example, which wires analog outputs 1 to 4 . In the case of CP1W-DA021, analog outputs 1 and 2 can be used.


## Wiring for Analog Outputs



## Note

## Allocation of Output Words

(1) Connect the shield to the FG terminal to prevent noise.
(2) Separate wiring from power lines (AC power supply lines, high-voltage lines, etc.)
(3) When there is noise in the power supply line, install a noise filter on the input section and the power supply.
(4) When external power is supplied (when range codes are set), or when the power is interrupted, there may be a pulse status analog output of up to 1 ms . If this status is a problem, take the following measures.

- Turn ON the power to the CP1L CPU Unit, check the operation status, and then turn ON the power at the load.
- Turn OFF the power to the load and then turn OFF the power to the CP1L CPU Unit.


## 3. Ladder Program

Four output words ( $n+1$ to $n+4$ ) are allocated, beginning from the first word following the last I/O word allocated to the CPU Unit or already-connected Expansion I/O Unit or Expansion Unit. For CP1W-DA021, two output words $(\mathrm{n}+1, \mathrm{n}+2)$ are allocated.


## Writing the Range Code

Write the output use and the range code to words $\mathrm{n}+1$ and $\mathrm{n}+2$. For CP1WDA021, only word $n+1$ can be used. The D/A conversion will start when the set data is transferred from the CPU Unit to the Analog Output Unit.


## Range Code

## Writing Analog Output Set Values

## Startup Operation



- The Analog Output Unit will not start converting analog I/O values until the range code has been written. The output will be 0 V or 0 mA .
- From when the range code has been written until data in the analog output convertible range is written, 0 V or 0 mA will be output in the 0 to 10 V , -10 to +10 V , and 0 to 20 mA ranges, and 1 V or 4 mA will be output in the 1 to 5 V and 4 to 20 mA ranges.
- Once the range code has been set, it is not possible to change the setting while power is being supplied to the CPU Unit. To change the I/O range, turn the CPU Unit OFF then ON again.
The ladder program can be used to write data to the output word where the set value is stored. The output word will be " $n+1$ " when " $n$ " is the last output word allocated to the CPU Unit, or previous Expansion Unit or Expansion I/O Unit.

After power is turned ON , it will require two cycle times plus approximately 50 ms before the first data is converted.
The following table shows the output status after the initial processing is completed.

| Output type | Voltage output |  | Current output |  |
| :--- | :--- | :--- | :--- | :--- |
| Output range | 0 to 10 V, <br> $-10 ~ t o ~$ <br> +10 V | 1 to 5 V | 0 to 20 mA | 4 to 20 mA |
| Before range <br> code is written | 0 V | 0 mA |  |  |
| After range <br> code is written | 0 V | 1 V | 0 mA | 4 mA |

Therefore, create a program as shown below, so that when operation begins simultaneously with startup it will wait for valid set data.


TIM 005 will start as soon as power turns ON. After 0.1 to 0.2 s ( 100 to 200 ms ), the Completion Flag for TIM 005 will turn ON, and the data stored in DM 0100 will be moved to IR 102 as the conversion data for analog output 1.

- When an error occurs at the Analog Output Unit, the analog output will be 0 V or 0 mA . If a CPU Unit fatal error occurs when analog outputs are set in the 1 to 5 V or 4 to 20 mA range, 0 V or 0 mA will be output for a CPU error I/O bus error, and 1 V or 1 mA will be output for all other errors.
- Expansion Unit errors are output to bits 0 to 6 of word A436. The bits are allocated from A436.00 in order starting with the Unit nearest the CPU Unit. Use these flags in the program when it is necessary to detect errors.


## Program Example

CP1W-DA041/CPM1A-DA041

| Analog output | Output range | Range code | Set data | Destination <br> word |
| :--- | :--- | :--- | :--- | :--- |
| Output 1 | 0 to 10 V | 001 | $1001(9 \mathrm{hex})$ | $\mathrm{Wd} \mathrm{n}+1$ |
| Output 2 | 4 to 20 mA | 100 | $1100(\mathrm{C} \mathrm{hex})$ | $\mathrm{Wd} \mathrm{n}+1$ |
| Output 3 | -10 to 10 V | 000 | $1000(8 \mathrm{hex})$ | Wd n+2 |
| Output 4 | Not used. | $-(000)$ | $0000(0 \mathrm{hex})$ | Wd n+2 |



## 7-4 Analog I/O Units

## 7-4-1 CPM1A-MAD01 Analog I/O Units

Each CPM1A-MAD01 Analog I/O Unit provides 2 analog inputs and 1 analog output.

- The analog input range can be set to 0 to 10 VDC, 1 to 5 VDC , or 4 to 20 mA with a resolution of $1 / 256$.
An open-circuit detection function can be used with the 1 to 5 VDC and 4 to 20 mA settings.
- The analog output range can be set to 0 to 10 VDC, -10 to 10 VDC, or 4 to 20 mA . The output has a resolution of $1 / 256$ when the range is set to 0 to 10 VDC or 4 to 20 mA , or a resolution of $1 / 512$ when set to -10 to 10 VDC.


## Part Names


(1) Analog I/O Terminals Connected to analog I/O devices.

I/O Terminal Arrangement


Note When using current inputs, short terminal V IN1 with I IN1 and terminal V IN2 with I IN2.

| V OUT | Voltage output |
| :--- | :--- |
| I OUT | Current output |
| COM | Output common |
| V IN1 | Voltage input 1 |
| I IN1 | Current input 1 |
| COM1 | Input common 1 |
| V IN2 | Voltage input 2 |
| I IN2 | Current input 2 |
| COM2 | Input common 2 |

(2) Expansion I/O Connecting Cable

Connected to the expansion connector of a CP1L CPU Unit or an Expansion Unit or Expansion I/O Unit. The cable is provided with the Analog I/ O Unit and cannot be removed.

Caution Do not touch the cables during operation. Static electricity may cause operating errors.

## (3) Expansion Connector

Used for connecting Expansion Units or Expansion I/O Units.

## Main Analog I/O Unit Specifications

Analog I/O Units are connected to the CP1L CPU Unit. For CP1L M-type CPU Units, up to three Units can be connected, including any other Expansion Units and Expansion I/O Units. For CP1L L-type CPU Units, one Unit can be connected.


| Item |  | Voltage I/O | Current I/O |
| :---: | :---: | :---: | :---: |
| Analog Input Section | Number of inputs | 2 |  |
|  | Input signal range | 0 to $10 \mathrm{~V} / 1$ to 5 V | 4 to 20 mA |
|  | Max. rated input | $\pm 15 \mathrm{~V}$ | $\pm 30 \mathrm{~mA}$ |
|  | External input impedance | $1 \mathrm{M} \Omega$ min. | $250 \Omega$ rated current |
|  | Resolution | 1/256 |  |
|  | Accuracy | 1.0\% full scale |  |
|  | A/D conversion data | 8-bit binary |  |
| Analog Output Section (See note 2.) | Number of outputs | 1 |  |
|  | Output signal range | 0 to 10 V or -10 to 10 V | 4 to 20 mA |
|  | Max. external output current | 5 mA | --- |
|  | Allowable external output load resistance | --- | $350 \Omega$ |
|  | Resolution | $1 / 256(1 / 512$ when the output signal range is -10 to 10 V$)$ |  |
|  | Accuracy | 1.0\% of full scale |  |
|  | Set data | 8-bit signed binary |  |
| Conversion time |  | $10 \mathrm{~ms} \mathrm{max}$. per Unit (See note 1.) |  |
| Isolation method |  | Photocoupler isolation between I/O terminals and PC signals. No isolation between analog I/O signals. |  |
| Current consumption |  | 5 VDC: $66 \mathrm{~mA} \mathrm{max.}$,24 VDC: $66 \mathrm{~mA} \mathrm{max}$. |  |

Note (1) The conversion time is the total time for 2 analog inputs and 1 analog output.
(2) With analog outputs it is possible to use both voltage outputs and current outputs at the same time. In this case however, the total output current must not exceed 21 mA .

## Analog I/O Signal Ranges

## Analog Input Signal Ranges



Analog Output Signal Ranges


## Using Analog I/O



Connecting the Analog I/O Connect the Analog I/O Unit to the CPU Unit. Unit


## Wiring Analog I/O Devices

## Analog Input Wiring



## Analog I/O Wiring Example

Using analog input 1 as a voltage input


Using analog input 2 as a current input


## Analog Output Wiring

Voltage Outputs


Current Outputs


Analog I/O Wiring Example
Using analog output as a voltage output

(1) For analog outputs it is possible to use both voltage outputs and current outputs at the same time, but the total current output must not exceed 21 mA .
(2) Use 2-core shielded twisted-pair cables.
(3) Wire away from power lines (AC power supply wires, power lines, etc.)
(4) When an input is not being used, short V IN and I IN to the COM terminal.
(5) Use crimp terminals. (Tighten terminals to a torque of $0.5 \mathrm{~N} \cdot \mathrm{~m}$.)
(6) When using current inputs, short $\mathrm{V}_{\mathrm{IN}}$ to $\mathrm{I}_{\mathrm{IN}}$.
(7) When there is noise in the power supply line, install a noise filter on the input section and the power supply terminals.

## I/O Allocation

Two input words and one output word are allocated to the Analog I/O Unit, starting from the next word following the last allocated word on the CPU Unit or previous Expansion Unit or Expansion I/O Unit.


## Writing the Range Code

Write the range code to word $n+1$. A/D or D/A conversion begins when the range code is transferred from the CPU Unit to the Analog I/O Unit. There are eight range codes, FF00 to FF07, that combine both the analog input 1 and 2 and analog output signal ranges, as shown below.

| Range <br> code | Analog input 1 signal <br> range | Analog input 2 signal <br> range | Analog output signal <br> range |
| :--- | :--- | :--- | :--- |
| FF00 | 0 to 10 V | 0 to 10 V | 0 to $10 \mathrm{~V} / 4$ to 20 mA |
| FF01 | 0 to 10 V | 0 to 10 V | -10 to $10 \mathrm{~V} / 4$ to 20 mA |
| FF02 | 1 to $5 \mathrm{~V} / 4$ to 20 mA | 0 to 10 V | 0 to $10 \mathrm{~V} / 4$ to 20 mA |
| FF03 | 1 to $5 \mathrm{~V} / 4$ to 20 mA | 0 to 10 V | -10 to $10 \mathrm{~V} / 4$ to 20 mA |
| FF04 | 0 to 10 V | 1 to $5 \mathrm{~V} / 4$ to 20 mA | 0 to $10 \mathrm{~V} / 4$ to 20 mA |
| FF05 | 0 to 10 V | 1 to $5 \mathrm{~V} / 4$ to 20 mA | -10 to $10 \mathrm{~V} / 4$ to 20 mA |
| FF06 | 1 to $5 \mathrm{~V} / 4$ to 20 mA | 1 to $5 \mathrm{~V} / 4$ to 20 mA | 0 to $10 \mathrm{~V} / 4$ to 20 mA |
| FF07 | 1 to $5 \mathrm{~V} / 4$ to 20 mA | 1 to $5 \mathrm{~V} / 4$ to 20 mA | -10 to $10 \mathrm{~V} / 4$ to 20 mA |

- The voltage/current selection is made by switching the wiring.
- Write the range code to the Analog I/O Unit output word $(n+1)$ in the first cycle of program execution.

- The Analog I/O Unit will not start converting analog I/O values until the range code has been written.
- Once the range code has been set, it is not possible to change the setting while power is being supplied to the CPU Unit. To change the I/O range, turn the CPU Unit OFF then ON again.
- If a range code other than those specified in the above table is written to $\mathrm{n}+1$, the range code will not be received by the Analog I/O Unit and ana$\log \mathrm{I} / \mathrm{O}$ conversion will not start.


## Reading A/D Conversion Tables

Data converted from analog to digital is output to bits 00 to 07 in words $\mathrm{m}+1$ and $\mathrm{m}+2$.


Note The Open-circuit Detection Flag is turned ON if the input signal range is set to 1 to 5 V or 4 to 20 mA and the input signal falls below 1 V or 4 mA . (Open circuits are not detected when the input signal range is set to 0 to 10 V .)

## Setting D/A Conversion Data

Output data is written to the Analog I/O Unit's allocated output word, word $\mathrm{n}+1$.


Note Word $(n+1)$ can be used for either the range code or the analog output set value.

$1,2,3 \ldots$ 1. The set value range is 0000 to $00 F F$ hex when the output signal range is 0 to $10 \mathrm{~V} / 4$ to 20 mA .
2. The set value range is divided into two parts: 8000 to 80 FF hex ( -10 to 0 V ) and 0000 to 00FF hex ( 0 to 10 V ) when the output signal range is -10 to 10 V .
3. If $\mathrm{FF} \square \square$ is input, $0 \mathrm{~V} / 4 \mathrm{~mA}$ will be output.
4. If an output value is specified, the following bits will be ignored.

- Output range of -10 to 10 V : Bits 08 to 14
- Output range of 0 to $10 \mathrm{~V} / 4$ to 20 mA : Bits 08 to 15


## Startup Operation

After power is turned ON, it will require two cycle times plus approx. 100 ms before the first data is converted. The following instructions can be placed at the beginning of the program to delay reading converted data from analog inputs until conversion is actually possible.
Analog input data will be 0000 until initial processing has been completed. Analog output data will be 0 V or 0 mA until the range code has been written. After the range code has been written, the analog output data will be 0 V or 4 mA if the range is 0 to $10 \mathrm{~V},-10$ to 10 V , or 4 to 20 mA .


## Handling Unit Errors

- When an error occurs in the Analog I/O Unit, analog input data will be 0000 and 0 V or 4 mA will be output as the analog output.
- Expansion Unit/Expansion I/O Unit errors are output to bits 0 to 6 of word A436. The bits are allocated from A436.00 in order starting with the Unit nearest the CPU Unit. Use these flags in the program when it is necessary to detect errors.


## Programming Example

This programming example uses these ranges:
Analog input 0: 0 to 10 V
Analog input 1: 1 to 5 V or 4 to 20 mA
Analog output: 0 to 10 V or 4 to 20 mA


## 7-4-2 CP1W-MAD11/CPM1A-MAD11 Analog I/O Units

Each CP1W-MAD11/CPM1A-MAD11 Analog I/O Unit provides 2 analog inputs and 1 analog output.

- The analog input range can be set to 0 to 5 VDC, 1 to 5 VDC, 0 to $10 \mathrm{VDC},-10$ to $10 \mathrm{VDC}, 0$ to 20 mA , or 4 to 20 mA . The inputs have a resolution of $1 / 6000$.
An open-circuit detection function can be used with the 1 to 5 VDC and 4 to 20 mA settings.
- The analog output range can be set to 1 to 5 VDC, 0 to 10 VDC, -10 to $10 \mathrm{VDC}, 0$ to 20 mA , or 4 to 20 mA . The outputs have a resolution of 1/6000.


## Part Names

## CP1W-MAD11/CPM1A-MAD11


(1) Analog I/O Terminals Connected to analog I/O devices.


Note For current inputs, short V INO to I INO and V IN1 to I IN1.

| V OUT | Voltage output |
| :--- | :--- |
| I OUT | Current output |
| COM | Output common |
| V IN0 | Voltage input 0 |
| I IN0 | Current input 0 |
| COM0 | Input common 0 |
| V IN1 | Voltage input 1 |
| I IN1 | Current input 1 |
| COM1 | Input common 1 |

(2) Expansion I/O Connecting Cable

Connected to the expansion connector of a CP1L CPU Unit or a CMP1A Expansion Unit or Expansion I/O Unit. The cable is provided with the Analog I/O Unit and cannot be removed.

Caution Do not touch the cables during operation. Static electricity may cause operating errors.
(3) Expansion Connector Used for connecting Expansion Units or Expansion I/O Units.
(4) DIP Switch

Used to enable or disable averaging.


Pin1: Average processing for analog input 0
目 (OFF: Average processing not performed; ON: Average processing performed)
Pin2: Average processing for analog input 1
(OFF: Average processing not performed; ON: Average processing performed)

Main Analog I/O Unit Specifications

Analog I/O Units are connected to the CP1L CPU Unit. For CP1L M-type CPU Units, up to three Units can be connected, including any other Expansion Units and Expansion I/O Units. For CP1L L-type CPU Units, one unit can be connected.


| Item |  |  | Voltage I/O | Current I/O |
| :---: | :---: | :---: | :---: | :---: |
| Analog Input Section | Number of inputs |  | 2 inputs (2 words allocated) |  |
|  | Input signal range |  | 0 to 5 VDC, 1 to 5 VDC, 0 to 10 VDC, or -10 to 10 VDC | 0 to 20 mA or 4 to 20 mA |
|  | Max. rated input |  | $\pm 15 \mathrm{~V}$ | $\pm 30 \mathrm{~mA}$ |
|  | External input impedance |  | $1 \mathrm{M} \Omega \mathrm{min}$. | Approx. $250 \Omega$ |
|  | Resolution |  | 1/6000 (full scale) |  |
|  | Overall accuracy | $25^{\circ} \mathrm{C}$ | 0.3\% full scale | 0.4\% full scale |
|  |  | 0 to $55^{\circ} \mathrm{C}$ | 0.6\% full scale | 0.8\% full scale |
|  | A/D conversion data |  | 16-bit binary (4-digit hexadecimal) <br> Full scale for -10 to 10 V: F448 to 0BB8 hex Full scale for other ranges: 0000 to 1770 hex |  |
|  | Averaging function |  | Supported (Settable for individual inputs via DIP switch) |  |
|  | Open-circuit detection function |  | Supported |  |
| Analog Output Section | Number of outputs |  | 1 output (1 word allocated) |  |
|  | Output signal range |  | 1 to 5 VDC, 0 to 10 VDC, or -10 to 10 VDC, | 0 to 20 mA or 4 to 20 mA |
|  | Allowable external output load resistance |  | $1 \mathrm{k} \Omega \mathrm{min}$. | $600 \Omega$ max. |
|  | External output impedance |  | $0.5 \Omega$ max. |  |
|  | Resolution |  | 1/6000 (full scale) |  |
|  | Overall accuracy y | $25^{\circ} \mathrm{C}$ | 0.4\% full scale |  |
|  |  | 0 to $55^{\circ} \mathrm{C}$ | 0.8\% full scale |  |
|  | Set data (D/A conversion) |  | 16-bit binary (4-digit hexadecimal) <br> Full scale for -10 to 10 V: F448 to 0BB8 hex Full scale for other ranges: 0000 to 1770 hex |  |
| Conversion time |  |  | $2 \mathrm{~ms} /$ point (6 ms/all points) |  |
| Isolation method |  |  | Photocoupler isolation between analog I/O terminals and internal circuits. <br> No isolation between analog I/O signals. |  |
| Current consumption |  |  | 5 VDC: 83 mA max., 24 VDC : 110 mA max. |  |

## Analog I/O Signal Ranges

Analog I/O data is digitally converted according to the analog I/O signal range as shown below.

Note When the input exceeds the specified range, the AD converted data will be fixed at either the lower limit or upper limit.

## Analog Input Signal

 Ranges
## - $\mathbf{1 0}$ to 10 V

The $-10-$ to $10-\mathrm{V}$ range corresponds to the hexadecimal values F448 to 0BB8 ( -3000 to 3000). The entire data range is F31C to OCE4 ( -3300 to 3300 ).
A negative voltage is expressed as a two's complement.


## 0 to 10 V

The 0 - to $10-\mathrm{V}$ range corresponds to the hexadecimal values 0000 to 1770 ( 0 to 6000). The entire data range is FED4 to 189C ( -300 to 6300). A negative voltage is expressed as a two's complement.


## 0 to 5 V

The 0- to 5-V range corresponds to the hexadecimal values 0000 to 1770 ( 0 to 6000). The entire data range is FED4 to 189C ( -300 to 6300). A negative voltage is expressed as a two's complement.


## 1 to 5 V

The 1- to $5-\mathrm{V}$ range corresponds to the hexadecimal values 0000 to 1770 ( 0 to 6000). The entire data range is FED4 to 189C ( -300 to 6300). Inputs between 0.8 and 1 V are expressed as two's complements. If the input falls below 0.8 V , open-circuit detection will activate and converted data will be 8000.


## 0 to 20 mA

The 0 - to $20-\mathrm{mA}$ range corresponds to the hexadecimal values 0000 to 1770 ( 0 to 6000). The entire data range is FED4 to 189C ( -300 to 6300). A negative voltage is expressed as a two's complement.


## 4 to 20 mA

The 4- to 20-mA range corresponds to the hexadecimal values 0000 to 1770 ( 0 to 6000). The entire data range is FED4 to 189C ( -300 to 6300). Inputs between 3.2 and 4 mA are expressed as two's complements. If the input falls below 3.2 mA , open-circuit detection will activate and converted data will be 8000.


## Analog Output Signal

Ranges

## - 10 to 10 V

The hexadecimal values F448 to 0BB8 ( -3000 to 3000) correspond to an analog voltage range of -10 to 10 V . The entire output range is -11 to 11 V . Specify a negative voltage as a two's complement.


## 0 to 10 V

The hexadecimal values 0000 to 1770 ( 0 to 6000) correspond to an analog voltage range of 0 to 10 V . The entire output range is -0.5 to 10.5 V . Specify a negative voltage as a two's complement.


1 to 5 V
The hexadecimal values 0000 to 1770 ( 0 to 6000) correspond to an analog voltage range of 1 to 5 V . The entire output range is 0.8 to 5.2 V .


## 0 to 20 mA

The hexadecimal values 0000 to 1770 ( 0 to 6000) correspond to an analog current range of 0 to 20 mA . The entire output range is 0 to 21 mA .


## 4 to 20 mA

The hexadecimal values 0000 to 1770 ( 0 to 6000) correspond to an analog current range of 4 to 20 mA . The entire output range is 3.2 to 20.8 mA .


The averaging function can be enabled for inputs using the DIP switch. The averaging function stores the average (a moving average) of the last eight input values as the converted value. Use this function to smooth inputs that vary at a short interval.

The open-circuit detection function is activated when the input range is set to 1 to 5 V and the voltage drops below 0.8 V , or when the input range is set to 4 to 20 mA and the current drops below 3.2 mA . When the open-circuit detection function is activated, the converted data will be set to 8,000 .
The time for enabling or clearing the open-circuit detection function is the same as the time for converting the data. If the input returns to the convertible range, the open-circuit detection is cleared automatically and the output returns to the normal range.

## Using Analog I/O



Reading Range Code Settings and A/D Conversion Data

Writing D/A Conversion

## Data

CPU Unit


Note Word $(n+1)$ can be used for either the range code or the analog output set value.

Connecting the Analog I/O Unit and Setting the DIP Switch

This section describes how to connect an Analog I/O Unit to the CPU Unit.


## Setting the Averaging Function

DIP switch pins 1-1 and 1-2 are used to set the averaging function. When averaging is enabled, a moving average of the last eight input values is output as the converted value. The averaging function can be set separately for analog inputs 1 and 2.


| DIP switch <br> pin | Function | Setting | Default |
| :--- | :--- | :--- | :--- |
| $1-1$ | Averaging | Analog input 0 <br> OFF: Disabled; ON: Enabled | OFF |
|  |  | Analog input 1 <br> OFF: Disabled; ON: Enabled | OFF |
| 1 |  |  |  |

## Wiring Analog I/O Devices

Analog Inputs


Analog Outputs


## Terminal Arrangements



Note For current inputs, short V INO to I INO and V IN1 to I IN1.

| V OUT | Voltage output |
| :--- | :--- |
| I OUT | Current output |
| COM | Output common |
| V IN0 | Voltage input 0 |
| I IN0 | Current input 0 |
| COM0 | Input common 0 |
| V IN1 | Voltage input 1 |
| I IN1 | Current input 1 |
| COM1 | Input common 1 |

Wiring for Analog Inputs


Wiring for Analog Outputs


Note (1) Use shielded twisted-pair cables, but do not connect the shield.
(2) When an input is not being used, short the + and - terminals.
(3) Separate wiring from power lines (AC power supply lines, high-voltage lines, etc.)
(4) When there is noise in the power supply line, install a noise filter on the input section and the power supply terminals.
(5) Refer to the following diagram regarding wiring disconnections when voltage input is being used.


Example: If analog input device 2 is outputting 5 V and the same power supply is being used for both devices as shown above, approximately $1 / 3$, or 1.6 V , will be applied to the input for input device 1.
If a wiring disconnection occurs when voltage input is being used, the situation described below will result. Either separate the power supplies for the connected devices, or use an isolator for each input.
If the same power supply is being used by the connected devices and a disconnection occurs at points A or B in the above diagram, an unwanted circuit path will occur as shown along the dotted line in the diagram. If that occurs, a voltage of approximately $1 / 3$ to $1 / 2$ of the output voltage of the other connected device will be generated. If that voltage is generated while the setting is for 1 to 5 V , open-circuit detection may not be possible. Also, if a disconnection occurs at point C in the diagram, the negative $(-)$ side will be used in for both devices and open-circuit detection will not be possible.
This problem will not occur for current inputs even if the same power supply is used.

Note When external power is supplied (when setting the range code), or when there is a power interruption, pulse-form analog output of up to 1 ms may be generated. If this causes problems with operation, take countermeasures such as those suggested below.

- Turn ON the power supply for the CP1L CPU Unit first, and then turn ON the power supply for the load after confirming correct operation.
- Turn OFF the power supply for the load before turning OFF the power supply for the CP1L CPU Unit.

Creating a Ladder Program

## I/O Allocation

Two input words and one output word are allocated to the Analog I/O Unit starting from the next word following the last allocated word on the CPU Unit or previous Expansion Unit or Expansion I/O Unit.


## Writing the Range Code

Write the range code to word $n+1$. A/D or D/A conversion begins when the range code is transferred from the CPU Unit to the Analog I/O Unit. There are five range codes, 000 to 100, that combine the analog input 1 and 2 and analog output signal ranges, as shown below.

| Range <br> code | Analog input 0 signal <br> range | Analog input 1 signal <br> range | Analog output signal <br> range |
| :--- | :--- | :--- | :--- |
| 000 | -10 to 10 V | -10 to 10 V | -10 to 10 V |
| 001 | 0 to 10 V | 0 to 10 V | 0 to 10 V |
| 010 | 1 to $5 \mathrm{~V} / 4$ to 20 mA | 1 to $5 \mathrm{~V} / 4$ to 20 mA | 1 to 5 V |
| 011 | 0 to $5 \mathrm{~V} / 0$ to 20 mA | 0 to $5 \mathrm{~V} / 0$ to 20 mA | 0 to 20 mA |
| 100 | --- | --- | 4 to 20 mA |



Example
The following instructions set analog input 0 to 4 to 20 mA , analog input 1 to 0 to 10 V , and the analog output to -10 to 10 V .

First Cycle Flag


- The Analog I/O Unit will not start converting analog I/O values until the range code has been written. Until conversion starts, inputs will be 0000, and 0 V or 0 mA will be output.
- After the range code has been set, 0 V or 0 mA will be output for the 0 to $10-\mathrm{V},-10$ to $10-\mathrm{V}$, or 0 to $20-\mathrm{mA}$ ranges, and 1 V or 4 mA will be output for the 1 to $5-\mathrm{V}$ and 4 to $20-\mathrm{mA}$ ranges until a convertible value has been written to the output word.
- Once the range code has been set, it is not possible to change the setting while power is being supplied to the CPU Unit. To change the I/O range, turn the CPU Unit OFF then ON again.


## Reading Converted Analog Input Values

The ladder program can be used to read the memory area words where the converted values are stored. Values are output to the next two words ( $m+1$, $m+2$ ) following the last input word ( m ) allocated to the CPU Unit or previous Expansion Unit or Expansion I/O Unit.

## Writing Analog Output Set Values

The ladder program can be used to write data to the memory area where the set value is stored. The output word will be " $n+1$," where " $n$ " is the last output word allocated to the CPU Unit or previous Expansion Unit or Expansion I/O Unit.

## Startup Operation

After power is turned ON, it will require two cycle times plus approx. 50 ms before the first data is converted. The following instructions can be placed at the beginning of the program to delay reading converted data from analog inputs until conversion is actually possible.
Analog input data will be 0000 until initial processing has been completed. Analog output data will be 0 V or 0 mA until the range code has been written. After the range code has been written, the analog output data will be 0 V or 0 mA if the range is 0 to $10 \mathrm{~V},-10$ to 10 V , or 0 to 20 mA , or it will be 1 V or 4 mA if the range is 1 to 5 V or 4 to 20 mA .


## Handling Unit Errors

- When an error occurs in the Analog I/O Unit, analog input data will be 0000 and 0 V or 0 mA will be output as the analog output.
If a CPU error or an I/O bus error (fatal errors) occurs at the CPU Unit and the analog output is set to 1 to 5 V or 4 to $20 \mathrm{~mA}, 0 \mathrm{~V}$ or 0 mA will be output. For any other fatal errors at the CPU Unit, 1 V or 4 mA will be output.
- Expansion Unit and Expansion I/O Unit errors are output to bits 0 to 6 of word A436. The bits are allocated from A436.00 in order starting from the Unit nearest the CPU Unit. Use these flags in the program when it is necessary to detect errors.


## Programming Example

This programming example uses these ranges:
Analog input 0: 0 to 10 V
Analog input 1: 4 to 20 mA
Analog output: 0 to 10 V


## 7-5 Temperature Sensor Units

CP1W-TS002/TS102 and CPM1A-TS002/TS102 Temperature Sensor Units each provide up to four input points, and CP1W-TS001/TS001 and CPM1ATS001/TS101 Temperature Sensor Units each provide up to two input points. The inputs can be from thermocouples or platinum resistance thermometers. CP1W-TS002/TS102 and CPM1A-TS002/TS102 Temperature Sensor Units are each allocated four input words.

## Part Names

## Temperature Sensor Units:

 CP1W-TS $\square \square$and CPM1A-TS

(1) Temperature Sensor Input Terminals

Used to connect temperature sensors such as thermocouples or platinum resistance thermometers.
(2) DIP Switch

Used to set the temperature unit ( ${ }^{\circ} \mathrm{C}$ or ${ }^{\circ} \mathrm{F}$ ) and the number of decimal places used.
(3) Rotary Switch

Used to set the temperature input range. Make the setting according to the specifications of the temperature sensors that are connected.
(4) Expansion I/O Connecting Cable

Connected to the expansion connector of a CP1L CPU Unit or a Expansion Unit or Expansion I/O Unit. The cable is included with the Temperature Sensor Unit and cannot be removed.
Note Do not touch the cables during operation. Static electricity may cause operating errors.
(5) Expansion Connector

Used for connecting Expansion Units or Expansion I/O Units.

## Main Specifications



| Item | $\begin{aligned} & \text { CP1W-TS001 } \\ & \text { CPM1A-TS001 } \end{aligned}$ | $\begin{aligned} & \text { CP1W-TS002 } \\ & \text { CPM1A-TS002 } \end{aligned}$ | CP1W-TS101 CPM1A-TS101 | $\begin{aligned} & \text { CP1W-TS102 } \\ & \text { CPM1A-TS102 } \end{aligned}$ |
| :---: | :---: | :---: | :---: | :---: |
| Temperature sensors | Thermocouples |  | Platinum resistance thermometer |  |
|  | Switchable between K and J, but same type must be used for all inputs. |  | Switchable between Pt100 and JPt100, but same type must be used for all inputs. |  |
| Number of inputs | 2 | 4 | 2 | 4 |
| Allocated input words | 2 | 4 | 2 | 4 |
| Accuracy | (The larger of $\pm 0.5 \%$ of converted value or $\pm 2^{\circ} \mathrm{C}$ ) $\pm 1$ digit max. (See note.) |  | (The larger of $\pm 0.5 \%$ of converted value or $\left.\pm 1^{\circ} \mathrm{C}\right) \pm 1$ digit max. |  |
| Conversion time | 250 ms for 2 or 4 input points |  |  |  |
| Converted temperature data | 16-bit binary data (4-digit hexadecimal) |  |  |  |
| Isolation | Photocouplers between all temperature input signals |  |  |  |
| Current consumption | 5 VDC: 40 mA max., 24 VDC: 59 mA max. |  | 5 VDC: 54 mA max., 24 VDC: 73 mA max. |  |

Note Accuracy for a K-type sensor at $-100^{\circ} \mathrm{C}$ or less is $\pm 4^{\circ} \mathrm{C} \pm 1$ digit max.

## Using Temperature Sensor Units

## Connecting Temperature Sensor Units



For CP1L M-type CPU Units, a maximum of three CPM1A-TS002 and CPM1A-TS102 Temperature Sensor Units can be connected, because each is allocated four words. For CP1L L-type CPU Units, one Unit can be connected.


## Setting Temperature Ranges

Note (1) Always turn OFF the power supply before setting the temperature range.
(2) Never touch the DIP switch or rotary switch during Temperature Sensor Unit operation. Static electricity may cause operating errors.
The Temperature Sensor Unit's DIP switch and rotary switch are used to set the temperature unit, to select 2-decimal-place Mode is to be used, and to set the temperature input range.


## DIP Switch Settings

The DIP switch is used to set the temperature unit ( ${ }^{\circ} \mathrm{C}$ or ${ }^{\circ} \mathrm{F}$ ) and the number of decimal places used.


| SW1 | Setting |  |  |
| :--- | :--- | :--- | :--- |
| 1 | Temperature unit | OFF | ${ }^{\circ} \mathrm{C}$ |
|  |  | ON | ${ }^{\circ} \mathrm{F}$ |
| 2 | Number of decimal <br> places used (See note.) <br> (0.01 expression) | OFF | Normal (0 or 1 digit after the decimal <br> point, depending on the input range) |
|  | ON | 2-decimal-place Mode |  |

Note For details on 2-decimal-place Mode, refer to Two-decimal-place Mode on page 473.

## Rotary Switch Setting

Caution Set the temperature range according to the type of temperature sensor connected to the Unit. Temperature data will not be converted correctly if the temperature range does not match the sensor.

Caution Do not set the temperature range to any values other than those for which temperature ranges are given in the following table. An incorrect setting may cause operating errors.

The rotary switch is used to set the temperature range.

| Setting | $\begin{aligned} & \text { CP1W-TS001/TS002 } \\ & \text { CPM1A-TS001/002 } \end{aligned}$ |  |  | $\begin{gathered} \text { CP1W-TS101/TS102 } \\ \text { CPM1A-TS101/102 } \end{gathered}$ |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | Input type | Range ( ${ }^{\circ} \mathrm{C}$ ) | Range ( ${ }^{\circ} \mathrm{F}$ ) | Input type | Range ( ${ }^{\circ} \mathrm{C}$ ) | Range ( ${ }^{\circ} \mathrm{F}$ ) |
| 0 | K | -200 to 1,300 | -300 to 2,300 | Pt100 | -200.0 to 650.0 | $\begin{aligned} & \hline-300.0 \text { to } \\ & 1,200.0 \end{aligned}$ |
| 1 |  | 0.0 to 500.0 | 0.0 to 900.0 | JPt100 | -200.0 to 650.0 | $\begin{array}{\|l} \hline-300.0 \text { to } \\ 1,200.0 \end{array}$ |
| 2 | J | -100 to 850 | -100 to 1,500 | --- | Cannot be set. |  |
| 3 |  | 0.0 to 400.0 | 0.0 to 750.0 | --- |  |  |
| 4 to F | --- | Cannot be set. |  | --- |  |  |

## Connecting Temperature Sensors

## Thermocouples

## CP1W-TS001/CPM1A-TS001

Either K or J thermocouples can be connected, but both of the thermocouples must be of the same type and the same input range must be used for each.


## CP1W-TS002/CPM1A-TS002

Either K or J thermocouples can be connected, but all four of the thermocouples must be of the same type and the same input range must be used for each.


Note When using a Temperature Sensor Unit with a thermocouple input, observe the following precautions:

- Do not remove the cold junction compensator attached at the time of delivery. If the cold junction compensator is removed, the Unit will not be able to measure temperatures correctly.
- Each of the input circuits is calibrated with the cold junction compensator attached to the Unit. If the Unit is used with the cold junction compensator from other Units, the Unit will not be able to measure temperatures correctly.
- Do not touch the cold junction compensator. Doing so may result in incorrect temperature measurement.


## Platinum Resistance Thermometers

## CP1W-TS101/CPM1A-TS101

One or two Pt or JPt platinum resistance thermometers can be connected, but both of the thermometers must be of the same type and the same input range must be used for each.


Temperature input 0 Temperature input 1

## CP1W-TS102/CPM1A-TS102

Up to four Pt100 or JPt100 platinum resistance thermometers can be connected, but all four of the thermometers must be of the same type and the same input range must be used for each.


Note Do not connect anything to terminals not used for inputs.

## Word Allocations

Temperature Sensor Units are allocated words as Expansion Units, in order of connection. A Temperature Sensor Unit is allocated the next input words following the input words of the CPU Unit or previous Expansion Unit or Expansion I/O Unit. Four input words are allocated is to the 2-input CPM1A-TS001 or CPM1A-TS101 and four input words are allocated to the 4 -input CPM1ATS002 or CPM1A-TS102. No output words are allocated.

## Example 1



## Example 2



## Converted Temperature Data

The temperature data will be stored in the input words allocated to the Temperature Sensor Unit in 4-digit hexadecimal.

| TS002/TS102 |  |
| :---: | :---: |
| m+1 | Converted temperature data from input 0 |
| m+2 | Converted temperature data from input 1 |
| m+3 | Converted temperature data from input 2 |
| m+4 | Converted temperature data from input 3 |

TS001/TS101

| $m+1$ | Converted temperature data from input 0 |
| :--- | :--- |
| $m+2$ | Converted temperature data from input 1 |

"m" is the last input word allocated to the CPU Unit, Expansion I/O Unit, or Expansion Unit connected immediately before the Temperature Sensor Unit.

- Negative values are stored as 2's complements.
- Data for range codes that include one digit after the decimal point are stored without the decimal point, i.e., 10 times the actual value is stored.

| Input |  | Data conversion examples |  |
| :--- | :--- | :--- | :---: |
| Unit: $1^{\circ} \mathrm{C}$ | K or J | $850^{\circ} \mathrm{C} \rightarrow 0352$ hex |  |
| $-200^{\circ} \mathrm{C} \rightarrow \mathrm{FF} 38$ hex |  |  |  |

- If the input temperature exceeds the maximum or minimum value in the temperature input range that has been set by $\pm 20^{\circ} \mathrm{C}$ or $\pm 20^{\circ} \mathrm{F}$, the displayed value will be held.
- If the circuit disconnected, the open-circuit detection function will operate and the converted temperature data will be set to 7FFF.
- The open-circuit detection function will be automatically cleared and normal input temperature conversion will begin automatically when the input temperature returns to the convertible range.


## Startup Operation

After power is turned ON , approximately 1 s is required for the first conversion data to be stored in the input word. During that period, the data will be 7FFE. Therefore, create a program as shown below, so that when operation begins simultaneously with startup it will wait for valid conversion data.


## Handling Unit Errors

- Expansion Unit and Expansion I/O Unit errors are output to bits 0 to 6 of word A436. The bits are allocated from A436.00 in order starting from the Unit nearest the CPU Unit. CP1W-/CPM1A-TS002 and CP1W-/CPM1ATS102 Temperature Sensor Units are allocated two bits each. Use these flags in the program when it is necessary to detect Expansion Unit/Expansion I/O Unit errors.
- When an error occurs, the Temperature Sensor Unit data becomes 7FFF hex (the same as for an open-circuit detection). With an open-circuit detection, it is not reflected in word A436.


## Programming Example

1,2,3... 1. The following programming example shows how to convert the input data from 2 temperature sensor inputs to BCD and store the result in D0 and D1.


2. The following programming example shows how to convert the data for temperature input 0 to BCD and store the result in D0 and D1. "0001" is stored in D1 when the input data is a negative value. The following system configuration is used.


| Temperature unit setting | $0\left({ }^{\circ} \mathrm{C}\right)$ |
| :--- | :--- |
| Two-decimal-place Mode | 0 (normal) |
| Input range setting | $1\left(\mathrm{Pt} 100:-200.0\right.$ to $\left.650.0^{\circ} \mathrm{C}\right)$ |
| Input 0 | CIO 2 |

## Programming with $\mathrm{BCD}(24)$ Instruction



Programming with SCL2(-) Instruction

## Two-decimal-place Mode



Operation


If pin 2 on the DIP switch is turned ON, values are stored to two decimal places. In this case, temperature data is stored as 6-digit signed hexadecimal (binary) data with 4 digits in the integer portion and 2 digits after the decimal point. The actual data stored in memory is 100 times the actual value, i.e., the decimal point is not indicated. Methods for handling this data are described in this section.

Note When set to store values to two decimal places, temperature data as far as two digits after the decimal point is converted to 6-digit binary data, but the actual resolution is not $0.01^{\circ} \mathrm{C}\left({ }^{\circ} \mathrm{F}\right)$. For this reason, there may be skipping and inaccuracies in the first digit after the decimal point (0.1). Treat any resolution above that specified for the normal data format as reference data.

## Temperature Data Partitioning and Structure

## Temperature Data (Actual Temperature $\mathbf{x} 100$ Binary)



Leftmost/Rightmost Flag: Indicates whether the leftmost or rightmost 3 digits are provided. Temperature Unit Flag: Indicates whether the temperature is in ${ }^{\circ} \mathrm{C}$ or ${ }^{\circ} \mathrm{F}$.
Open-circuit Flag: Turns ON (1) when an open-circuit is detected. The temperature data will be 7FF FFF if this flag is ON.

Data Conversion Examples

## Example 1

Temperature: $\quad 1,130.25^{\circ} \mathrm{C}$ $\times 100$ : 113025
Temperature Data: 01B981 (hexadecimal for 113025)

## Leftmost 3 Digits and Flags

|  | Flags |  |  |  | $\times 16^{5}$ | $\times 16^{4}$ | $\times 16^{3}$ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Bits | 15 | 14 | 13 | 12 | 11 to 08 | 07 to 04 | 03 to 00 |
| Data | 0 | 0 | 0 | 0 | 0 | 1 | B |
|  |  |  |  |  |  |  |  |



Rightmost 3 Digits and Flags

|  | Flags |  |  |  | $\times 16^{2}$ | $\times 16{ }^{1}$ | $\times 16^{0}$ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Bits | 15 | 14 | 13 | 12 | 11 to 08 | 07 to 04 | 0 |
| Data | 1 | 0 | 0 | 0 | 9 | 8 | 1 |
|  |  |  |  |  |  |  |  |



## Example 2

Temperature: $\quad-100.12^{\circ} \mathrm{C}$
$\times 100$ : -10012
Temperature Data: FFD8E4 (hexadecimal for -10012)

## Leftmost 3 Digits and Flags



Rightmost 3 Digits and Flags

|  | Flags |  |  |  | $\times 16^{2}$ | $\times 16^{1}$ | $\times 16^{0}$ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Bits | 15 | 14 | 13 | 12 | 11 to 08 | 07 to 04 | 03 to 00 |
| Data | 1 | 0 | 0 | 0 | 8 | E | 4 |
|  |  |  |  |  |  |  |  |



## Example 3

Temperature: $\quad-200.12^{\circ} \mathrm{F}$
$\times 100$ : -20012
Temperature Data: FFB1D4 (hexadecimal for -20012)

## Leftmost 3 Digits and Flags

|  | Flags |  |  |  | $\times 16^{5}$ | $\times 16^{4}$ | $\times 16^{3}$ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Bits | 15 | 14 | 13 | 12 | 11 to 08 | 107 to 04 | 03 to 00 |
| Data | 0 | 1 | 0 | 0 | F | F | B |
|  |  |  |  |  |  |  |  |



Rightmost 3 Digits and Flags

|  | Flags |  |  |  | $\times 16^{2}$ | $\times 16^{1}$ | $\times 16^{0}$ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Bits | 15 | 14 | 13 | 12 | 11 to 08 | 07 to 04 | 03 to 00 |
| Data | 1 | 1 | 0 | 0 | 1 | D | 4 |
| $L^{\circ}{ }^{\circ} \mathrm{F} \text { Normal }$— Rightmost |  |  |  |  |  |  |  |



## Example 4

Temperature: Open circuit ( ${ }^{\circ} \mathrm{F}$ )
Temperature Data: 7FFF FFFF

## Leftmost 3 Digits and Flags



Rightmost 3 Digits and Flags


Note

Programming Example
(1) Leftmost digits are stored in the lower memory addresses. Treat the data in the lower memory address as the leftmost digits when programming.
(2) Be sure that the data is read at least once every 125 ms to allow for the CPU Unit's cycle time and communications time. Correct data may not be obtained if the read cycle is greater than 125 ms .
The following programming example shows how to use 2-decimal-place Mode for the following PC configuration.

| CPU Unit | CP1W/CPM1A-TS001 <br> Temperature Sensor Unit |  |
| :---: | :---: | :---: |
| Inputs <br> CIO 000 <br> CIO 001 | Inputs CIO 002 CIO 003 | Temperature unit setting: $0\left({ }^{\circ} \mathrm{C}\right)$ |
|  |  |  |
|  |  |  |
|  |  | Two-decimal-place Mode: |
|  |  | 1 (2 digits after decimal point stored) |
| Outputs CIO 100 | Outputs |  |
| CIO 100 CIO 101 | None |  |

In this example, 100 times the temperature data for temperature input 0 is stored in binary form in D100 to D102.



## Description of Operation



## 7-6 CompoBus/S I/O Link Units

The CP1L can function as a slave to a CompoBus/S Master Unit (or SRM1 CompoBus/S Master Control Unit) when a CP1W-SRT21/CPM1A-SRT21 CompoBus/S I/O Link Unit is connected. The CompoBus/S I/O Link Unit establishes an I/O link of 8 inputs and 8 outputs between the Master Unit and the PLC. Up to three CompoBus/S I/O Link Units, including other Expansion I/ O Units, can be connected to a CP1L CPU Unit.


From the standpoint of the CP1L CPU Unit, the 8 input bits and 8 output bits allocated to the CompoBus/S I/O Link Unit are identical to input and output bits allocated to Expansion I/O Units even though the CompoBus/S I/O Link Unit does not control actual inputs and outputs. The input and output bits allocated to the CompoBus/S I/O Link Unit are one side of an I/O link between the slave CPU Unit and the CPU Unit to which the Master Unit is connected.

Master PLC (CS Series)
CP1L


Specifications

| Model number | CP1W-SRT21/CPM1A-SRT21 |
| :--- | :--- |
| Master/slave | CompoBus/S Slave |
| Number of I/O points | 8 input points, 8 output points |
| Number of words allocated in <br> CPU Unit I/O memory | 1 input word, 1 output word <br> (Allocated in the same way as Expansion Units and <br> Expansion I/O Units.) |
| Node number setting | Set using the DIP switch <br> (Set before turning on the CPU Unit's power supply.) |

LED Indicators

| Indicator | Name | Color | Meaning |
| :--- | :--- | :--- | :--- |
| COMM | Communications <br> Indicator | Yellow | ON: Communications in progress. <br> OFF: Communications stopped or error <br> has occurred. |
| ERR | Error indicator | Red | ON: A communications error has |
| occurred. |  |  |  |
| OFF: Indicates normal communications |  |  |  |
| or stand-by. |  |  |  |

## CP1W-SRT21/CPM1A-SRT21 CompoBus/S I/O Link Unit


(1) CompoBus/S Terminals

The following CompoBus/S terminals are provided: CompoBus/S communications data high/low terminals, NC terminals for communications power supply plus (+) and minus (-), and an NC terminal. (Power is supplied internally for this Unit, so the NC terminals for communications power supply can be used as relay terminals.)

## (2) DIP Switch

Used to specify the node number for the CompoBus/S I/O Link Unit. (Refer to the following table.)


| $\begin{gathered} \text { Pin labels } \\ \hline 1 \end{gathered}$ | Contents |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Node Number Setting | SW1 |  |  |  |
| 248 |  |  | 8 | 4 | 2 | 1 |
|  |  | 0 | 0 | 0 | 0 | 0 |
|  |  | 1 | 0 | 0 | 0 | 1 |
|  |  | 2 | 0 | 0 | 1 | 0 |
|  |  | 3 | 0 | 0 | 1 | 1 |
|  |  | 4 | 0 | 1 | 0 | 0 |
|  |  | 5 | 0 | 1 | 0 | 1 |
|  |  | 6 | 0 | 1 | 1 | 0 |
|  |  | 7 | 0 | 1 | 1 | 1 |
|  |  | 8 | 1 | 0 | 0 | 0 |
|  |  | 9 | 1 | 0 | 0 | 1 |
|  |  | 10 | 1 | 0 | 1 | 0 |
|  |  | 11 | 1 | 0 | 1 | 1 |
|  |  | 12 | 1 | 1 | 0 | 0 |
|  |  | 13 | 1 | 1 | 0 | 1 |
|  |  | 14 | 1 | 1 | 1 | 0 |
|  |  | 15 | 1 | 1 | 1 | 1 |
|  | $1=\mathrm{ON}, 0=\mathrm{OFF}$ |  |  |  |  |  |
| DR | ON | Long-distance communications mode (See note.) |  |  |  |  |
|  | OFF | High-speed communications mode |  |  |  |  |
| HOLD | ON | Retain inputs after a communications error. |  |  |  |  |
|  | OFF | Clear inputs after a communications error. |  |  |  |  |

(3) LED Indicators

Used to show the CompoBus/S communications status.

| Indicator | Name | Color | Meaning |
| :--- | :--- | :--- | :--- |
| COMM | Communications <br> indicator | Yellow | ON: Communications in progress. <br> OFF: Communications stopped or error <br> has occurred. |
| ERR | Error indicator | Red | ON: A communications error has |
| occurred. |  |  |  |
| OFF: Indicates normal communications |  |  |  |
| or stand-by. |  |  |  |

(4) Expansion I/O Connecting Cable

Connected to the expansion connector of a CP1L CPU Unit or a Expansion Unit or Expansion I/O Unit. The cable is provided with the CompoBus/S I/O Link Unit and cannot be removed.

Note Do not touch the cables during operation. Static electricity may cause operating errors.
(5) Expansion Connector Used to connect Expansion Units or Expansion I/O Units.

## Operating Procedure

Connecting the CompoBus/S I/O Link Unit


CompoBus/S I/O Link Units are connected to the CP1L CPU Unit. For CP1L M-type CPU Units, up to three Units can be connected, including any other Expansion Units and Expansion I/O Units. The Units can be connected in any order from the CPU Unit. For CP1L L-type CPU Units, one Unit can be connected.

CP1L CPU Unit


CompoBus/S I/O Link Unit


## I/O Allocation

I/O words are allocated to the CompoBus/S I/O Link Unit in the same way as to other Expansion Units and Expansion I/O Units, i.e., the next available input and output words are allocated. As shown below, when " $m$ " is the last allocated input word and " $n$ " is the last allocated output word, the CompoBus/S I/ O Link Unit is allocated " $m+1$ " as its input word and " $n+1$ " as its output word.

CompoBus/S I/O Link Unit


In the following example, a CompoBus/S I/O Link Unit is connected as the first Unit after the CP1L CPU Unit.


The input word $(m+1)$ contains the 8 bits of data from the Master Unit and two CompoBus/S communications flags.


Write the data to be transmitted to the Master Unit in the output word ( $n+1$ ).


Data to be transferred to the Master Unit
Note (1) The 8 bits of I/O data are not always transmitted simultaneously. In other words, 8 bits of data transmitted from the Master CPU Unit at the same time will not always reach the Slave CPU Unit simultaneously, and 8 bits of data transmitted from the Slave CPU Unit at the same time will not always reach the Master CPU Unit simultaneously.
When the 8 bits of input data must be read together, modify the ladder program in the CPU Unit receiving the data. For example, read the input data twice in succession and accept the data only when the two values match.
(2) Unused bits in the CompoBus/S I/O Link Unit's output word can be used as work bits, but unused bits in the output slaves cannot be used as work bits.
(3) Unused bits in input word cannot be used as work bits.

Determining the Node Number and Making DIP Switch Settings

## Node Number

- The CompoBus/S I/O Link Unit is a Slave Unit with 8 input bits and 8 output bits. The node number setting is made using the DIP switch; the inputs and outputs share the same node number.
- The range of possible node number settings is determined by the type of PLC the Master Unit is mounted to and the settings on the Master Unit. For details refer to the CompoBus/S Operation Manual.


## DIP Switch Settings

Use the DIP switch to set the CompoBus/S I/O Link Unit's node number, communications mode, and the status of output data when a communications error occurs.



Note Always turn OFF the power supply before changing the DIP switch settings.

Wiring the CompoBus/S Communications Path

Wire the CompoBus/S communications path as shown in the following diagrams.


## 7-7 DeviceNet I/O Link Units

Connecting a CPM1A-DRT21 DeviceNet I/O Link Unit (with 32 inputs and 32 outputs as built-in I/O) to function as a slave allows the CP1L to be used as a DeviceNet slave. A maximum of three DeviceNet I/O Link Units can be connected to the CP1L to create I/O Links for up to 192 points ( 96 inputs and 96 outputs) between the CP1L and the DeviceNet master.


From the standpoint of the CP1L CPU Unit, the 32 input bits and 32 output bits allocated to the DeviceNet I/O Link Unit are identical to input and output bits allocated to Expansion I/O Units even though the DeviceNet I/O Link Unit does not control external inputs and outputs. The input and output bits allocated to the DeviceNet I/O Link Unit are one side of an I/O link between the slave CPU Unit and the CP1L CPU Unit to which the Master Unit is connected.

Master PLC (CS Series with fixed allocations)


Note Refer to the DeviceNet Slaves Operation Manual (W347) for details on DeviceNet networks.

Specifications

| Model number | CPM1A-DRT21 |
| :--- | :--- |
| Master/slave | DeviceNet Slave |


| Number of I/O points | 32 input points, 32 output points |
| :--- | :--- |
| Number of words allocated in <br> CPU Unit I/O memory | 2 input words, 2 output words <br> (Allocated in the same way as other Expansion Units <br> and Expansion I/O Units.) |
| Node number setting | Set using the rotary switches <br> (Set before turning ON the CPU Unit's power supply.) |
| Communications current con- <br> sumption | 48 mA |

## CPM1A-DRT21 DeviceNet I/O Link Unit


(1) DeviceNet Communications Connector

Used to connect DeviceNet communications. For the wiring, use the connector provided with the CPM1A-DRT21 or use a connector purchased separately.
(2) Rotary Switches (SW2, SW3)

Used to set DeviceNet node numbers.
SW3
SW2
 Setting range: 0 to 63 (Do not set 64 to 99.)
(3) DIP Switch (SW1)

Used to set the DeviceNet baud rate and the output hold function.
SW1


| Baud rate setting (See note.) |  |  |  |
| :--- | :--- | :--- | :--- |
| Pin 1 | Pin 2 | Baud rate | Max. transmission path <br> length |
| OFF | OFF | 125 kbps | 500 m |
| ON | OFF | 250 kbps | 250 m |
| OFF | ON | 500 kbps | 100 m |
| ON | ON | Not allowed. | --- |


| Output hold function setting |  |
| :--- | :--- |
| Pin 4 | DeviceNet baud rate |
| OFF | Clears remote outputs when communications error occurs. (Outputs <br> turned OFF for each logic value.) |
| ON | Holds remote outputs when communications error occurs. |

Note When using Expansion Unit/Expansion I/O Unit Error Flags (A436) in the program, set pin 4 on the DIP switch to ON. If communications are set to be cleared, the timing for clearing outputs and setting the Error Flags may not agree.
(4) LED Indicators

Used to indicate CPM1A-DRT21 status, as shown in the following table.

| Indicator | Color | Status | Condition | Meaning |
| :---: | :---: | :---: | :---: | :---: |
| MS | Green | Lit | Normal status | - Normal status |
|  |  | Flashing | Not set | - Switch settings being read |
|  | Red | Lit | Fatal error | - Fatal hardware error (watchdog timer) |
|  |  | Flashing | Nonfatal error | - Incorrect switch settings. |
|  | --- | OFF | Power not supplied. | - Power not supplied. <br> - Waiting for initialization to start. <br> - Reset in progress. |
| NS | Green | Lit | Online and communications established. | - Network normal and communications established. |
|  |  | Flashing | Online and communications not established. | - Network normal and communications not established. |
|  | Red | Lit | Fatal communications error | Unit has detected network status preventing normal communications. <br> - Node number duplications <br> - Bus OFF detected. |
|  |  | Flashing | Nonfatal communications error | - Communications timeout or communications error for one or more slaves. |
|  | --- | OFF | Online and power OFF. | Waiting for node number check by master. <br> - Switch setting error. <br> - Power not supplied. |

(5) Expansion I/O Connecting Cable

Connected to the expansion connector of a CP1L CPU Unit or an Expansion Unit or Expansion I/O Unit. The cable is included with the DeviceNet Unit and cannot be removed.
Note Do not touch the cables during operation. Static electricity may cause operating errors.
(6) Expansion Connector

Used for connecting Expansion Units or Expansion I/O Units.

## Handling Unit Errors

If a communications error occurs while the slave is on standby, the appropriate bit in word A436 will turn ON. The appropriate bit is determined by the order in which the Expansion Units and Expansion I/O Units are connected. The Unit nearest to the CPU Unit uses A436.00. Use these flags in the program when it is necessary to detect errors.

## Operating Procedure



Connecting the DeviceNet I/O Link Unit

DeviceNet I/O Link Units are connected to the CP1L CPU Unit. For CP1L M-type CPU Units, up to three Units can be connected, including any other Expansion Units and Expansion I/O Units. The Units can be connected in any order from the CPU Unit. For CP1L L-type CPU Units, one Unit can be connected.


## I/O Allocation

I/O words are allocated to the DeviceNet I/O Link Unit in the same way as to Expansion I/O Units or other Expansion Units, i.e., the next available input and output words are allocated. As shown below, when " $m$ " is the last allocated input word and " n " is the last allocated output word, the DeviceNet I/O Link Unit is allocated " $m+1$ " as its input word and " $n+1$ " as its output word.

DeviceNet I/O Link Unit


In the following example, a CompoBus/S I/O Link Unit is connected as the first Unit after the CP1L CPU Unit.


All of the words allocated to the DeviceNet I/O Link Unit are used to read and write data between the CPU Unit of the DeviceNet I/O Link Unit and the CPU Unit of the DeviceNet master, as shown in the following illustration.


## Note

## Determining the Node Number and Making DIP Switch Settings

(1) The 32 bits each of I/O data are not always transmitted simultaneously. In other words, 32 bits of data transmitted from the Master CPU Unit at the same time will not always reach the CP1L CPU Unit simultaneously, and 32 bits of data transmitted from the CP1L CPU Unit at the same time will not always reach the Master CPU Unit simultaneously.
When the 32 bits of input data must be read together, modify the ladder program in the CPU Unit receiving the data. For example, read the input data twice in succession and accept the data only when the two values match.
(2) Unused bits in the DeviceNet I/O Link Unit's output words can be used as work bits if they are not used for output from the slave.
(3) Unused bits in input words cannot be used as work bits.

## Setting Node Numbers

Use rotary switches SW2 and SW3 to set DeviceNet node number. The setting range is from 00 to 63 , and 64 to 99 cannot be set. Rotary switch settings go into effect when the power is turned ON.


Note The actual range of node numbers that can be set depends on the type of PLC to which the Master Unit is mounted, and on the Master Unit setting. For details, refer to the DeviceNet DRT1-series Slaves Operation Manual.

## Setting the DIP Switch (SW1)

Used to set the DeviceNet baud rate and the output hold function.


Baud Rate

| Pin 1 | Pin 2 | Baud rate | Max. transmission path length |
| :--- | :--- | :--- | :--- |
| OFF | OFF | 125 kbps | 500 m |
| ON | OFF | 250 kbps | 250 m |

## Wiring the DeviceNet Communications Path

| Pin 1 | Pin 2 | Baud rate | Max. transmission path length |
| :--- | :--- | :--- | :--- |
| OFF | ON | 500 kbps | 100 m |
| ON | ON | Not allowed. | --- |

Output Hold Function

| Pin 4 | DeviceNet baud rate |
| :--- | :--- |
| OFF | Clears remote outputs when communications error occurs. (Outputs turned <br> OFF for each logic value.) |
| ON | Holds remote outputs when communications error occurs. |

Note When using Expansion Unit/Expansion I/O Unit Error Flags (A436) in the program, turn ON pin 4 on the DIP switch. If communications are set to be cleared, the timing for clearing outputs and setting the Error Flags may not agree.
When using a CPM1A-DRT21 DeviceNet I/O Link Unit, wire the DeviceNet communications cable as shown in the following diagram.


## DeviceNet Connectors

Use the following connectors.

| Model | XW4B-05C1-H1-D | XW4B-05C4-TF-D |
| :--- | :--- | :--- |
| Form and <br> specifications | OMRON connector with screws <br> (provided with CPM1A-DRT21) | OMRON connector for multidrop <br> connections (See note.) |
|  |  |  |
|  |  |  |

Note Use the XW4B-05C4-TF-D when wiring multidrop connections using Thick Cables.
Use the following screwdriver for the above connector.
xW4Z-00C


## I/O Response Time

Refer to the DeviceNet Slaves Operation Manual (W347) for details on the response time. The data read/write time for one cycle for the CPM1A-DRT21 is approximately 0.5 ms . Add a maximum of 1 ms to the $\mathrm{I} / \mathrm{O}$ response time.

## SECTION 8 <br> LCD Option Board

This section gives an outline of the LCD Option Board, explains how to install and remove the LCD Option Board, and describes the functions including how to monitor and make settings for the PLC. It also lists the errors during operation and provides probable causes and countermeasures for troubleshooting.
$\qquad$8-1 Features492
8-2 Specifications ..... 493
8-3 Part Names ..... 494
8-4 Installation and Removing ..... 495
8-5 Basic Operation ..... 496
8-5-1 Startup ..... 496
8-5-2 Screen Transitions ..... 497
8-5-3 Operation Examples ..... 499
8-6 LCD Option Board Function ..... 501
8-6-1 Function Overview ..... 501
8-6-2 PLC Mode ..... 504
8-6-3 I/O Memory Setting ..... 505
8-6-4 PLC Setup ..... 510
8-6-5 Analog ..... 512
8-6-6 Error ..... 513
8-6-7 Memory Cassette ..... 516
8-6-8 User Monitor Screen ..... 520
8-6-9 Message Screen ..... 529
8-6-10 Timer Switch. ..... 534
8-6-11 Data Backup ..... 540
8-6-12 Language Selection ..... 543
8-6-13 PLC Cycle Time ..... 544
8-6-14 PLC Clock Setting ..... 545
8-6-15 PLC System Information ..... 546
8-6-16 LCD Backlight Setting ..... 547
8-6-17 LCD Contrast Setting ..... 548
8-6-18 LCD Factory Setting. ..... 549
8-7 Trouble Shooting ..... 550
8-7-1 Symptom at Power ON or during Operation ..... 550
8-7-2 Communication Error Message during Operation ..... 550
8-7-3 Deleting EEPROM Error ..... 551

## 8-1 Features

LCD Option Board is small but has a wide range of functions and is easy to use.

## Powerful Display and Setting Functions

Equipped for easy display and set up of user-specified messages, time or other data of the PLC.

## User Monitor Screen

Preset the screen, including I/O memory and text string, which user will monitor frequently. So it is easy for user to acquire his necessary data. It is possible to register up to 16 screens.

## Timer Switch

Preset the timer, including Day, Weekly and Calendar Timer. Each timer can execute a trans-day, trans-week or trans-year operation. So a 24-hours control will be effected by one-step setting. It is possible to register up to 16 timers for each kind.

## Easier to Identify with Backlight

When PLC error occurs, the red backlight of LCD display screen will begin to blink, quickly altering you of the error.
Normally the backlight is green. The automatic cutout time for the backlight can be set to occur from 2 to 30 minutes, or even set permanently to OFF or ON position.
The contrast level can also be adjusted.

## 8-2 Specifications

| Item | Specification |
| :---: | :---: |
| Model | CP1W-DAM01 |
| Type | Built-in |
| Serial port | Only port 1 |
| Communication protocol | Toolbus |
| DC consumption | $\begin{aligned} & 5 \mathrm{~V}: 40 \mathrm{~mA} \\ & 24 \mathrm{~V}: 0 \mathrm{~mA} \end{aligned}$ |
| Dimensions | $43 \times 36 \times 23 \mathrm{~mm}(\mathrm{~W} \times \mathrm{H} \times \mathrm{D})$ |
| Weight | 20 g max. |
| Screen size | $2.6 \mathrm{~cm} \times 1.45 \mathrm{~cm}$ |
| Total characters on screen | 4 lines $\times 12$ characters |
| Font size | $5 \times 7$ dot |
| Backlight color | Green / Red |
| Display language | English / Japanese (Katakana) |
| Ambient operating temperature | 0 to $55^{\circ} \mathrm{C}$ |
| Ambient operating humidity | 10\% to 90\% (with no condensation) |
| Atmosphere | No corrosive gas. |
| Ambient storage condition |  |

## 8-3 Part Names



Back


## Operation Button

| No. | Button | Function |  |
| :---: | :---: | :---: | :--- |
| a | Esc | ESC | Cancel the setting and return to the up-level menu. |
| b | $\boldsymbol{\eta}$ | Forward | Move the column cursor. <br> Press and hold the button, the column cursor will move for- <br> ward continuously. |
| c | $\boldsymbol{\lambda}$ | Up | Move the line cursor up. Change numerals and parameters. <br> Press and hold the button, the line cursor will move up contin- <br> uously and the parameters will increase continuously. |
| d | Down | Move the line cursor down. Change numerals and parame- <br> ters. <br> Press and hold the button, the line cursor will move down <br> continuously and the parameters will decrease continuously. |  |
| e | OK | OK | Confirm the setting. |

## Backlight

| Color | Meaning |
| :--- | :--- |
| Green | PLC is normal. |
| Red | PLC error has occurred. |

## 8-4 Installation and Removing

## Installation

The following processing explains how to install and remove a LCD Option Board.
! Caution Always turn OFF the power supply to the CPU Unit and wait until all the operation indicators go out before installing or removing the LCD Option Board.
1,2,3... 1. Press the up/down lock levers on both sides of the Option Board slot cover 1 at the same time to unlock the cover, and then pull the cover out.
2. Check the alignment to make the corner cut of the LCD Option Board fit in the Option Board slot 1, and firmly press the LCD Option Board in until it snaps into place.

3. Switch DipSW4 of the CPU Unit to ON.

Note DipSW4 is OFF at shipment


Removing
Press the up/down lock levers on both sides of the LCD Option Board at the same time to unlock the Option Board, and then pull it out.


## 8-5 Basic Operation

## 8-5-1 Startup

According to the operation status of the LCD Option Board, it will display different screens when the CPU Unit power is turned ON.

## Normal Startup

When the CPU Unit power is turned ON, the LCD Option Board will initialize hardware and check EEPROM, then check communication between the LCD Option Board and the CPU Unit. If startup is normal, LCD will display Clock Screen as shown below.

## Clock Screen



| No. | Description |
| :---: | :--- |
| a | Type of the CPU Unit |
| b | Date of the CPU Unit |
| c | Time of the CPU Unit |
| d | Week abbreviation of the CPU Unit |

## Startup Failure

- If EEPROM is faulty, LCD will display EEPROM Error Screen and the red backlight will blink. Refer to 8-7 Trouble Shooting.


## EEPROM Error Screen

## प्बा एe

E母\&i!t


- If the communication between the LCD Option Board and the CPU Unit has failed, LCD will display NG screen. Refer to 8-7 Trouble Shooting.
NG Screen


Note If the LCD Option Board receives no response from the CPU Unit within 3 seconds during operation, it will also display NG screen.

## 8-5-2 Screen Transitions

The screen transition of the LCD Option Board as shown in the following diagram.


Note 1. The screen will be displayed after making settings in the Setup Mode.
2. The Message Screen will disappeared automatically after control bit is OFF.
3. In the Setup Mode, if there is no operation for 10 minutes, LCD will automatically switch to the Monitor Mode.

## Screen Transition Example in the Monitor Mode

In this example, User Monitor Screen 1 and Message Screen 2, Message Screen 6 have been set.

## Control bit is OFF



Control bit 1 is ON


Message Screen 2 Clock Screen

## Control bit 5 is ON during control bit 1 ON



## Control bit 1 is OFF



## Control bit 1 and bit 5 are ON at the same time



Note 1. When one control bit is ON, the Clock Screen or the User Monitor Screen will switch to the Message Screen automatically.
2. If another control bit is ON when the Clock Screen or the User Monitor Screen is diplayed, the display will switch to another Message Screen.
3. If another control bit is ON when one Message Screen is diplayed, the display will not change until one of the control bit is OFF.
4. If another control bit is bigger, the display will swtich to another Message Screen after one of the control bit is OFF. If another control bit is smaller, the display will swtich to the Clock Screen after one of the control bit is OFF.
5. When no less than one control bit are ON at the same time, the Message Screen whose Screen No. is smaller will be displayed.
6. If one control bit is ON during the period that PLC error occurs, the display remains the Error Screen. Even if the error is eliminated, the display will not switch to the Message Screen, but return to the Clock Screen.

## 8-5-3 Operation Examples

With actual operation examples, the main operation flow of the LCD Option Board as shown below.

## Menu Selection

Display the Monitor Screen of I/O memory.
1,2,3...

1. Turn on the power to the CPU Unit. Clock Screen will be displayed.
```
M#ए।
Mm"$
    ण पिए
    FLE E+4
```

2. Press the $\mathbf{O K}+$ ESC button simultaneously to switch to the main menu. The line cursor ">" is always displayed on the first line of menu items.
*
*
Mes
Mes
\ा, \
\ा, \
PLE 5+G
PLE 5+G
3. Press the Down or Up button to select the menu item. Move the line cursor to IO Memory.

4. Press the $\mathbf{O K}$ button to enter the submenu.
TH
TH

- 
- 

Pा
Pा

```
णอएक # |
4:4+8
+1:#55
+2#+24
```

5. Press the Down or Up button to select the I/O memory type. Move the line cursor to $D M$.
6. Press the OK button to enter the Monitor Screen of I/O memory.

## Displaying I/O Memory

Display any data of I/O memory. In this example, two word data on D10001 to D10002, D10003 to D10004 with unsigned decimal number will be displayed.

## 1,2,3...



1. Line 1 will display the default address D00000 in I/O memory, Line 2 to 4 will display one word data on D00000, D00001, D00002 with hex number when entering the Monitor Screen of I/O memory.
The first digit of memory address " 0 " will flash. The column cursor is at the flashing position.
The digit under the column cursor can be changed, otherwise it is read only.
2. Use the Forward button to move the column cursor to the digit to be set. Use the Down or Up button to change the value of each digit.
The screen display will be updated immediately after the address is changed.
3. Use the Forward button to move the column cursor to another parameter to be set.
Use the Down or Up button to select the value of parameter.
The screen display will be updated immediately after the parameter is changed.

## Changing I/O Memory

Change any data of I/O memory. In this example, the data of I/O memory on D10001 will be changed.

1,2,3...

```
4!%
+4 %+5
+1:%"%
+2"+%
```


2. Press the OK button to enter the Change Screen of I/O memory. The column cursor is at the "\#" position. Use the Down or Up button to select the value of parameter.

3. Use the Forward button to move the column cursor to the data of I/O memory. Use the Down or Up button to change the value of each digit.
4. Press the OK button to save the setting.

Press the ESC button to return to the previous screen.
The data displayed in the Monitor Screen will be changed.

Press the ESC button to cancel the setting and return to the previous screen.

## 8-6 LCD Option Board Function

This section describes the functions of the LCD Option Board including how to monitor and make settings for the PLC.

## 8-6-1 Function Overview

## PLC Mode

Display the present PLC mode and change the PLC mode.
Refer to Page 504 for details.
म्यु एम

I/O Memory Setting
Monitor and change the data of I/O Memory.
Refer to Page 505 for details.


```
TH
    |ा
    M .
```

PLC Setup
Monitor and change the PLC Setup, especially fast access the CPU Unit Operating Mode.
Refer to Page 510 for details.

```
F!-:%#
आए
    \squareே%
```

Analog
Monitor the value from the analog adjuster and external analog setting input of the PLC.
Refer to Page 512 for details.


## Error History

## Memory Cassette

## User Monitor Screen

## Message Screen

Display the list of error history and the details of each error. It is possible to display up to 20 screens. User can also monitor the occurring errors.
Refer to Page 514 for details.


The LCD Option Board can execute any of the following operations.

- Load data from memory cassette to PLC.
- Save data from PLC to memory cassette.
- Compare data between PLC and memory cassette.
- Clear data in memory cassette.

Refer to Page 516 for details.


Set or delete User Monitor Screen, which includes some elements such as I/O word memory, bit memory or text string. It is possible to register up to 16 screens. User can monitor his necessory data in the User Monitor Screen.
Refer to Page 520 for details.


Set or delete Message Screen. It is possible to register up to 16 screens. User can monitor the text message in the Message Screen when control bit is ON.
Refer to Page 529 for details.

```
ME=5=%
Y5
    DE!=
```

Timer Switch

## Data Backup

## Language

## Other

Set day, weekly and calendar timers. It is possible to register up to 16 timers for each kind. Each timer can execute a trans-day, trans-week or trans-year operation.
Refer to Page 534 for details.

```
MM,
ME TME
    HE|TM
    \square1 TMe
```

The LCD Option Board can execute any of the following operations.

- Load user settings from DM area.
- Save user settings to DM area.

So user can save the user settings to the DM area of the PLC from one LCD Option Board and load to other LCD Option Boards from the DM area.
Refer to Page 540 for details.

```
प&-5\&F
ME=
    5%
```

Change the language of the LCD display between English and Japanese. Refer to page 543 for details.


- PLC Cycle Time
- PLC Clock Setting
- PLC System Information
- LCD Backlight Setting
- LCD Contrast Setting
- LCD Factory Setting

Refer to Page 545 to 548 for details.
प्र्प्र्धि

## 8-6-2 PLC Mode

This function can display the present PLC mode and change the PLC mode.

## Example Change the PLC Mode from RUN to PRG.

## 1,2,3...



1. Switch to the Setup Mode.
2. Press the OK button to enter the Mode Screen.

There is a choice of 3 PLC modes-RUN/MON/PRG.
The line cursor will point to the present PLC mode.
The present mode is RUN.
3. Press the Down button to select PROGRAM.
4. Press the OK button, then LCD will update the present mode to PRG.

## 8-6-3 I/O Memory Setting

## Displaying I/O Memory

Example Monitor two word data on D10001 to D10002, D10003 to D10004 with unsigned decimal number.

1,2,3...


1. Switch to the Setup Mode.
2. Press the Down button to select IO Memory.
3. Press the $\mathbf{O K}$ button to enter the I/O Memory menu.
4. Press the Down button to select $\boldsymbol{D M}$.

5. Press the OK button to enter the Monitor Screen of I/O memory DM. The following table shows the setting items.

| No. | Description |
| :---: | :--- |
| a | Leading word address |
| b | Display format |
| c | Data length |

The first digit of the leading word address will be flashing.
The present setting is the default address.
Line 2 to 4 will display one word data on D00000, D00001, D00002 with hex number.

```
DIणSL * U
+E%>+8
+1:+%=
+2#+\4
```

6. Use the Forward button to move the column cursor to the digit to be set. Use the Up button to change the leading word address to 10001.


The following table shows the default address and the setting range for each I/O memory type.

| I/O memory type | Default address | Range |
| :--- | :--- | :--- |
| TIM | 0000 | 0000 to 4095 |
| CNT | 0000 | 0000 to 4095 |
| DM | 00000 | 00000 to 32767 |
| AR | 000 | 000 to 959 |
| IO | 0000 | 0000 to 6143 |
| WR | 000 | 000 to 511 |
| HR | 000 | 000 to 511 |
| DR | 00 | 00 to 15 |
| IR | 00 | 00 to 15 |
| TK | 00 | 00 to 31 |

Note LongWord has only five display types, DM, IO, WR, HR and AR.
7. Use the Forward button to move the column cursor to the display format position.
Press the Down or Up button to select the display format $\boldsymbol{\&}$.
Select the display format in the following table.

| Display format | Meaning |
| :--- | :--- |
| $\#$ | Hex number |
| + | Signed decimal number |
| $\&$ | Unsigned decimal number |

8. Use the Forward button to move the column cursor to the data length position.
Press the Down or Up button to select the data length $\mathbf{L W}$.
Select the data length in the following table.

| Data length | Meaning |
| :--- | :--- |
| W | One word data |
| LW | Two word data |

Then it will display two word data on D10001 to D10002, D10003 to D10004 with unsigned decimal number.

Note The screen display will be updated immediately after the address, display format or data length is changed.

## Changing I/O Memory

Example

1,2,3...
ए+या
Mए=
प पिए PT: $=$


First change two word data on W000 to 12345678, then change one word data on W509 to 98F5 and set the control bit 509.05 to OFF.

1. Switch to the Setup Mode.
2. Press the Down button to select IO Memory.
3. Press the $\mathbf{O K}$ button to enter the I/O Memory menu.
4. Press the Down button to select WR.
5. Press the OK button to enter the Monitor Screen of I/O memory WR.
6. Use the Forward button to move the column cursor to the data length position.
Press the Down or Up button to select the data length $\boldsymbol{L W}$.
7. Press the OK button to enter the Change Screen of I/O memory W000. The following table shows the setting items.

| No. | Attributes |
| :--- | :--- |
| 1 | Head channel address (Read only) |
| 2 | Display format (Read only) |
| 3 | Data length (Read only) |
| 4 | Data of I/O memory before change (Read only) |
| 5 | Data of I/O memory after change |


11. Press the OK button to enter the Change Screen of I/O memory W509. The following table shows the setting items.

| No. | Description |
| :---: | :--- |
| a | Leading word address(Read only) |
| b | Display format(Read only) |
| c | Data length(Read only) |
| d | Data of I/O memory before change (Read only) |
| e | Data of I/O memory after change |
| f | Bit address |
| g | Bit flag |
| h | Bit state |

Note If the display format is a decimal number (\& or + ), or the data length is a LongWord, user cannot make a setting for bit.
12. Move the column cursor to the digit to be set. Use the Up button to change the data to 98F5.
13. Use the Forward button to move the column cursor to the position of bit address.
The present setting is the default address. The range is $00 \sim 15$.

15. Use the Forward button to move the column cursor to the bit flag position. The present setting is the default setting. Select the bit flag in the following table.

| Bit flag | Meaning |
| :--- | :--- |
| $N$ | Normal |
| $S$ | Force to SET |
| $R$ | Force to RESET |

16. Use the Forward button to move the column cursor to the bit state position. The present state is ON. The state ON or OFF is according to PLC.

17. Press the Down or Up button to select the bit state OFF.

Note If bit flag is $S$ or $R$, the setting of bit state is invalid.
14. Use the Up button to change the bit address to 05.

18. Press the OK button to save the setting.

Press the ESC button to return to the previous screen.
Then the data on W509 displayed in the Monitor Screen will be 98D5.

## 8-6-4 PLC Setup

## Example 1 Change the CPU Unit Operating Mode from PRG to RUN.

1,2,3...


1. Switch to the Setup Mode.
2. Press the Down button to select PLC Setup.
3. Press the OK button to enter the PLC Setup menu.
4. Press the OK button to enter the CPU Unit Operating Mode Screen. The following table shows the setting items.

| No. | Attributes |
| :--- | :--- |
| a | Address of CPU Unit Operating Mode (Read only) |
| b | Present PLC mode (Read only) |
| c | CPU Unit Operating Mode after change (Read only when PLC <br> mode is RUN or MON) |
| d | CPU Unit Operating Mode before change (Read only) |

The address of CPU Unit Operating Mode is always 081, so there is no need to change the address.
5. Use the Up button to select RUN.

Note Before changing the CPU Unit Operating Mode, make sure that the present PLC mode is PRG. If PLC is in RUN or MON mode, the CPU Unit Operating Mode is unchangeable.
6. Press the $\mathbf{O K}$ button to save the setting.
7. Press the ESC or OK button to return to the previous menu.

Example 2 Display the value of PLC Setup on 080. Then change the value to 0195.
1,2,3...
ME\#|
ME\#|
Mty
Mty
\square1+का=
\square1+का=
PT: 5-m
PT: 5-m

5. Press the OK button to enter the PLC Setup Screen The following table shows the setting items.

| No. | Description |
| :---: | :--- |
| a | Address of PLC Setup |
| b | PLC mode (Read only) |
| c | Value of PLC Setup after change <br> (Read only when PLC mode is RUN or MON) |
| d | Value of PLC Setup before change (Read only) |

The first digit of PLC Setup address will be flashing. The range of the address is 000 to 511 .
6. Use the Up button to change the address to $\mathbf{0 8 0}$.

After the address is changed, the value of PLC Setup will be updated immediately.
7. Use the Forward button to move the column cursor to the value of PLC Setup.
Use the Up button to change the value to 0195.
Note Before changing the value of PLC Setup, make sure that the PLC mode is PRG. If PLC is in RUN or MON mode, the value is unchangeable.
$\pm \mathrm{E}$
E:
\&\&:4%
\&\&:4%
Tिए एि
Tिए एि
फलण क्या
फलण क्या
8. Press the OK button to save the setting.
9. Press the ESC or OK button to return to the PLC Setup Screen.

## 8-6-5 Analog

## Displaying Analog Settings

Example Monitor the external analog setting input with unsigned decimal number.
1,2,3...


1. Switch to the Setup Mode.
2. Press the Down button to select Analog.
3. Press the OK button to enter the Analog menu.
4. Press the OK button to enter the Monitor Screen of analog setting.

Line 2 will display the value from the analog adjuster.
Line 4 will display the external analog setting input value.
The display format on line 2 will be flashing.
5. Use the Forward button to move the column cursor to the display format position on line 4.

6. Press the Down or Up button to change the display format to $\boldsymbol{\&}$.

7. Press the ESC button to return to the previous screen.

## 8-6-6 Error

This function can display the list of error history and the details of each error. It is possible to display up to 20 screens. User can also monitor the occurring errors in the Error Monitor Screen.

## Displaying and Clearing Error History

Example Display the list of error history and then clear it.
1,2,3...

5. Press the OK button to enter the Error Screen details.

The following table shows the display items.

| No. | Description |
| :---: | :--- |
| a | Error number(1 to 20) |
| b | Error type |
| c | The date error occurred |
| d | The time error occurred |
| e | Error code |


7. Press the ESC button to return to the Error History Screen.

Press the Down button to select CLR ErrLog which is always below the last error.

पए ETME प्एन区

पே $4 T \square \square$ צ-
ج
$\checkmark$
8. Press the OK button to enter the Error Clear Screen.
9. Press the Down button to select $\mathbf{O K}$.

Note Selecting Cancel will result in a return to the previous screen.
10. Press the OK button to clear the error history.

When the clearing is finished, it will display a complete screen.
11. Press the ESC button to return to the Error History Screen. All the errors have been cleared.

## Clearing Occurring Error List

Example Clear memory error in the list that occurs at the present time.
1,2,3...


1. Switch to the Setup Mode.
2. Press the Down button to select Error.
```
ETए"
Frraleg
    Eपणणा
```

```
#प+|
    FMoाएe
Fपण|ए
```

Frman लa
प्या
E\&
区्ए
$\cdot$
लपाए
पिए
Eu* Hht
शघ चा

```
TE ET]
प#-1
    %
```


3. Press the OK button to enter the Error menu.
4. Press the Down button to select ErrorMon.
5. Press the OK button to enter the Error Monitor Screen. Max. 2 errors that occur the earliest will be displayed.
6. Press the Down button to select CLR Err.
7. Press the OK button to enter the Error Clear Screen.
8. Press the Down button to select OK.
9. Press the OK button to clear the memory error in the list.

Note Only one error that occurs the earliest in the list will be cleared one time.
10. If the memory error itself has not been eliminated, when the Error Monitor Screen is updated, the error will be displayed again in the screen.

## 8-6-7 Memory Cassette

## Before Operation

- Memory Cassette should be equipped into the PLC. Otherwise LCD cannot operate Memory Cassette.
- Make sure that the PLC mode is PRG. If the PLC is in RUN or MON mode, the operation of Memory Cassette cannot be executed.


## Loading Data from Memory Cassette to PLC

Example Load data from Memory Cassette to the PLC.
1,2,3...

3. Press the OK button to enter the Memory Cassette menu.

The following table shows the setting items.

| No. | Description |
| :---: | :--- |
| a | PLC mode (Read only) |
| b | Operation mode |

Select the operation mode in the following table.

| Operation Mode | Meaning |
| :--- | :--- |
| MC->PLC | Load data from memory cassette to PLC |
| PLC->MC | Save data from PLC to memory cassette |
| Compare | Compare data between PLC and MC |
| Clear | Format memory cassette |

4. Press the $\mathbf{O K}$ button to enter the "MC->PLC" Operation Screen.
5. Press the Down button to select $\boldsymbol{O K}$.

Note Selecting Cancel will result in a return to the previous menu.

6. Press the OK button to start loading.

A rate of loading will be displayed in the screen.

```
M-F
```

7. When the rate comes up to $0 \%$, the loading is finished. Then it will display a complete screen.

एomie

## Saving Data from PLC to Memory Cassette

Example Save data from the PLC to Memory Cassette.

## 1,2,3...



1. Switch to the Setup Mode.

2. Press the Down button to select MC.
3. Press the OK button to enter the Memory Cassette menu.
4. Press the Down button to select PLC->MC.
5. Press the $[\mathrm{OK}]$ button to enter the "PLC->MC" menu. Select the saving mode in the following table.

| Saving Mode | Meaning |
| :--- | :--- |
| Autoboot PRG | If the power turns ON, the operation cannot be <br> executed. |
| Autoboot RUN | Even if the power turns ON, the operation can be <br> executed. |


6. Press the OK button to enter the "PLC->MC" Operation Screen.

7. Press the Down button to select OK.

Note Selecting Cancel will result in a return to the previous menu.

8. Press the OK button to start saving.

A rate of saving will be displayed in the screen.
9. When the rate comes up to $0 \%$, the saving is finished. Then it will display a complete screen.

## Comparing Data between PLC and MC

Example Compare the data between the PLC and Memory Cassette.
1,2,3...


1. Switch to the Setup Mode.
2. Press the Down button to select MC.

Mோ
Males
ETए
T

3. Press the OK button to enter the Memory Cassette Screen.

```
4]
4.
\(\mathrm{M}-\mathrm{F}\)
Pr-
लाए
```

जि
-

5. Press the OK button to enter the Compare Operation Screen.

6. Press the Down button to select $\mathbf{O K}$.

Note Selecting Cancel will result in a return to the previous menu.

7. Press the OK button to start comparing.

A rate of comparison will be displayed in the screen.
8. When the rate comes up to $0 \%$, the comparing is finished. Then it will display a result of comparison.

## Clearing Memory Cassette

Example Clear the data in Memory Cassette.
1,2,3...


1. Switch to the Setup Mode.
2. Press the Down button to select $\boldsymbol{M C}$.
3. Press the OK button to enter the Memory Cassette menu.
4. Press the Down button to select Clear.

5. Press the OK button to enter the Clear Operation Screen.

6. Press the Down button to select $\mathbf{O K}$.

Note Selecting Cancel will result in a return to the previous menu.
7. Press the $\mathbf{O K}$ button to start clearing.

A rate of clearance will be displayed in the screen.
8. When the rate comes up to $0 \%$, the clearing is finished. Then it will display a complete screen.

```
&"
एाएiete
```


## 8-6-8 User Monitor Screen

This function can set or delete User Monitor Screen. It is possible to register up to 16 screens. User can monitor his necessary data in the User Monitor Screen. Each User Monitor Screen includes 4 lines of content. Each line has three kinds of display type including word memory, bit memory and text string.

## Creating New User Monitor Screen

Example 1 Monitor one word data on the word address D09000 with unsigned decimal number through User Monitor Screen 2, displayed on Line 1.

1,2,3...


1. Switch to the Setup Mode.

2. Press the Down button to select UserMonitor.
3. Press the $\mathbf{O K}$ button to enter the User Monitor menu.
```
ME,m,
SEta
    Delete
```


4. Press the OK button to enter the User Monitor Setup Screen.

The final digit of the Screen No. will be flashing.
The following table shows the setting items for each display type.

| No. | Description | Display type |  |  |
| :---: | :--- | :---: | :---: | :---: |
|  |  | Word | Bit | Text <br> string |
| a | Monitor flag | Yes | Yes | Yes |
| b | User Monitor Screen No. (01 to 16) | Yes | Yes | Yes |
| c | Line No. (1 to 4) of the User Monitor <br> Screen | Yes | Yes | Yes |
| d | Display type | Yes | Yes | Yes |
| e | I/O memory address | Yes | Yes | No |
| f | Display format | Yes | No | No |
| g | Date length | Yes | No | No |
| h | I/O memory name | Yes | Yes | Yes |

5. Use the Up button to change the Screen No. to 2.
6. Use the Forward button to move the column cursor to the monitor flag position. Select the monitor flag in the following table.

| Monitor flag | Meaning |
| :--- | :--- |
| Y | User Monitor Screen in use |
| N | User Monitor Screen not in use |

7. Press the Up button to select the monitor flag $\boldsymbol{Y}$.

Then user can monitor this screen after the setting is complete.
8. Use the Forward button to move the column cursor to the Line No. position. The present setting is Line 1.
9. Use the Forward button to move the column cursor to the display type position.
The following table shows the display types which can be selected, including the default address and the setting range for each type.

| Display type |  | Default address | Range |
| :--- | :--- | :--- | :--- |
|  | IO | 0000 | 0000 to 6143 |
|  | WR | 000 | 000 to 511 |
|  | HR | 000 | 000 to 511 |
|  | AR | 000 | 000 to 959 |
|  | TIM | 0000 | 0000 to 4095 |
|  | CNT | 0000 | 0000 to 4095 |
|  | DM | 00000 | 00000 to 32767 |
|  | DR | 00 | 00 to 15 |
|  | IR | 00 | 00 to 15 |
|  | TK | 00 | 00 to 31 |
|  | TMF(Timer flag) | 0000 | 0000 to 4095 |
|  | CTF(Timer flag) | 0000 | 0000 to 4095 |
| Bit | IOB | 0000.00 | 0000.00 to 6143.15 |
|  | WRB | 000.00 | 000.00 to 511.15 |
|  | HRB | 000.00 | 000.00 to 959.15 |
|  | ARB | 000.00 | - |
| Text string | STR |  |  |


10. Press the Up button to select $D M$.
11. Use the Forward button to move the column cursor to the memory address position.
The present setting is the default address.
12. Move the column cursor to the digit to be set. Use the Up button to change the memory address to 09000.
13. Use the Forward button to move the column cursor to the display format position.
Select the display format in the following table.

| Display format | Meaning |
| :--- | :--- |
| $\#$ | Hex number |
| + | Signed decimal number |
| $\&$ | Unsigned decimal number |


20. Press the ESC button three times to return to the Monitor Mode. Switch to the User Monitor Screen 2 with the Down button.

Note 1. Setting of word or bit name is not necessary. The default name is NULL, and the memory address will be displayed at the name position in the User Monitor Screen.
2. One line setting will take 1 or 2 lines of space. If word or bit name length is more than 5 characters or data length is a LongWord, it will take 2 lines of space.
3. One screen only has 4 lines of space available. If one line setting has already taken 2 lines of space, the next line setting will be invalid. If the setting of line 4 takes 2 lines of space, its setting will be invalid.

Example 2 Display a text string "elevator" on the User Monitor Screen 2, Line 4, after the setting in example 1.

1,2,3...


1. Switch to the Setup Mode.

## Men।

प्राणnitm $M \leq 5$ TMए

5. Use the Up button to change the Screen No. to 2. The setting in example 1 will be displayed.
6. Use the Forward button to move the column cursor to the Line No. position. Use the Up button to change the Line No. to 4.
7. Use the Forward button to move the column cursor to the display type position.
Press the Up button to select $\boldsymbol{S} \boldsymbol{T} \boldsymbol{R}$.
8. Use the Forward button to move the column cursor to the position of String Name.

$\because \mathrm{T}$
Int
Еேோ．

ஈाए：
बिक
$\pm \boxed{\omega}$

9．Use the Down or Up button to select the character of each digit． Name the text string to elevator．

10．Press the OK button to save the setting．

11．Press the ESC or OK button to return to the User Monitor Setup Screen．

12．Press the ESC button three times to return to the Monitor Mode． Switch to the User Monitor Screen 2.

Note 1．The default text string is NULL．
2．The max length of text string is 12 characters．

## Changing User Monitor Screen

User can not only change the date displayed in the User Monitor Screen in the Setup Mode，but also in the Monitor Mode．

Example 1 Change the average to 0100 and the minimum to－00123．
1，2，3．．．

```
##प्ड
\ि +बएव
&
```



| サ＂ | W |
| :---: | :---: |
| ण1\％ | $+\square$ |
| サ® |  |
| 『－ | ¢\％ |

2．Press the Forward＋OK button simultaneously to enter the Data Change Screen．
The column cursor will be flashing on the digit before the value．

| जि $\quad$ ए－ <br> Mि + बा णே <br> बम्बएक |  |
| :---: | :---: |
|  |  |
|  |  |

3．Use the Forward button to move the column cursor to the digit to be set．


```
जि% Wए।ए
Mम +णएद
14
णயएक्र्
```

```
जु अणाए
1M +M, 子
ME:
कण+\्ड्
```


4. Use the Up button to change the value to $\mathbf{0 1 0 0}$.
5. Press the OK button to save the setting. The column cursor will return to the digit before the value.
6. Use the Down button to move the cursor to line 2.

Note Only when the cursor is on the digit before the value, press the Down or Up button to b move the cursor to other lines.
7. Use the Forward button to move the column cursor to the sign position. Press the Down or Up button to change the sign to -.
8. Press the OK button to save the setting. If the setting is invalid, the screen display will have no change.
9. Press the ESC button to return to the User Monitor Screen. The average has been changed to 0100, but the minimum is still +00123 .

## Example 2 Change bit0 from OFF to ON.

1,2,3...

2. Use the Down button to move the cursor to line 2.
3. Use the Forward button to move the column cursor to the bit state position.
4. Press the Down or Up button to change the bit state to $\mathbf{O N}$.
5. Press the OK button to save the setting.

Press the ESC button to return to the User Monitor Screen.

## Deleting User Monitor Screen

Example Delete the User Monitor Screen 2.
1,2,3...

```
\#ए
MEE
    T0 पिए=
    PC:5+
```


## Mேи৷

पड्पाए
$\mathrm{M} 5=5$
TME

```
4Em|
T+
    \square=1=
```

1. Switch to the Setup Mode.
2. Press the Down button to select UserMonitor.
3. Press the $\mathbf{O K}$ button to enter the User Monitor menu.


एघete मृ


```
4<ym
    5%
    \ere=
-1ே
```

4. Press the Down button to select Delete.
5. Press the $\mathbf{O K}$ button to enter the User Monitor Delete Screen. The final digit of the Screen No. will be flashing.
6. Use the Up button to change the Screen No. to 2.

Note Press and hold the UP button until the Screen No. changes to ALL, all the User Monitor Screen will be deleted if the setting is confirmed.
7. Press the $\mathbf{O K}$ button to delete the screen.
8. Press the ESC or OK button to return to the previous menu.

## 8-6-9 Message Screen

This function can set or delete Message Screen. It is possible to register up to 16 screens. User can monitor the text message in the Message Screen when control bit is ON.

## Creating New Message Screen



## Example

1,2,3...
4. Press the OK button to enter the Message Setup Screen. The final digit of the Screen No. will be flashing. The following table shows the setting items.

| No. | Description |
| :---: | :--- |
| a | Message Screen No. (01 to 16) |
| b | Leading word (Only DM) address |
| c | Message flag |
| d | Word (Only WR) address of control bit |

5. Use the Up button to change the Screen No. to 2.

The following table shows the relation between the Screen No. and the control bit when the word address is W000.

| Screen No. | Control bit |
| :--- | :--- |
| 01 | W000.00 |
| 02 | W000.01 |
| 03 | W000.02 |
| 04 | W000.03 |
| $\ldots$ | $\ldots$ |
| 16 | W000.15 |



## Et ए

## EE


6. Use the Forward button to move the column cursor to the position of leading word address.
The present setting is the default address.
The following table shows the default address and the setting range for each screen when the leading word address is D09000.

| Screen No. | Default address | Range |
| :--- | :--- | :--- |
| 01 | 09000 to 09035 |  |
| 02 | 09040 to 09075 | 0000 to 32732 |
| 03 | 09080 to 09115 |  |
| 04 | 09120 to 09155 |  |
| $\ldots$ | $\ldots$ |  |
| 16 | 09600 to 09635 |  |

7. Use the Forward button to move the column cursor to the message flag position. Select the message flag in the following table.

| Message flag | Meaning |
| :--- | :--- |
| Y | Message Screen in use |
| N | Message Screen not in use |

8. Press the Up button to select the message flag $\boldsymbol{Y}$.

The setting is available for all the screens.
9. Use the Forward button to move the column cursor to the position of word address.
The present setting is the default address. The range of the address is 000 to 511 .
10. Use the Up button to change the word address to 100.
11. Press the $\mathbf{O K}$ button to save the setting.
12. Press the ESC or OK button to return to the Message Setup Screen.
13. Press the ESC button three times to return to the Monitor Mode. Switch to the Message Screen 2 when control bit W100.01 is ON.

## DM Area Settings

The text message is stored in the DM area. One character is 1 byte and one DM word is 2 bytes, so 24 DM words need to be used to store one screen message. But not all of the area can be used.
The following table shows the setting area for each screen when the leading word address is D09000..

| Screen No. | Word | 0 |  | 1 |  | 2 |  | 3 |  | 4 |  | 5 |  | 6 | 7 | 8 | 9 |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 01 | D09000 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |  |  |  |  |  |
|  | D09010 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |  |  |  |  |  |
|  | D09020 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 |  |  |  |  |  |
|  | D09030 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 |  |  |  |  |  |
| 02 | D09040 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |  |  |  |  |  |
|  | D09050 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | Do not use. |  |  |  |  |
|  | D09060 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 |  |  |  |  |  |
|  | D09070 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 |  |  |  |  |  |
| ... |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 16 | D09600 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |  |  |  |  |  |
|  | D09610 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |  |  |  |  |  |
|  | D09620 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 |  |  |  |  |  |
|  | D09630 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 |  |  |  |  |  |

In this example, "Elevator Stop at 1F" is displayed on the Message Screen 2.
The data can be set in the DM area with the CX-Programmer.
The settings show as below.

| Line No. | Word | Setting | Character |  |
| :---: | :---: | :---: | :---: | :---: |
| 1 | D09040 | 2020 |  |  |
|  | D09041 | 2020 |  |  |
|  | D09042 | 2020 |  |  |
|  | D09043 | 2020 |  |  |
|  | D09044 | 2020 |  |  |
|  | D09045 | 2020 |  |  |
| 2 | D09050 | 456C | E | I |
|  | D09051 | 6576 | e | v |
|  | D09052 | 6174 | a | t |
|  | D09053 | 6F72 | - | r |
|  | D09054 | 2020 |  |  |
|  | D09055 | 2020 |  |  |
| 3 | D09060 | 2053 |  | S |
|  | D09061 | 746F | t | 0 |
|  | D09062 | 7020 | p |  |
|  | D09063 | 6174 | a | t |
|  | D09064 | 2031 |  | 1 |
|  | D09065 | 4620 | F |  |
| 4 | D09070 | 2020 |  |  |
|  | D09071 | 2020 |  |  |
|  | D09072 | 2020 |  |  |
|  | D09073 | 2020 |  |  |
|  | D09074 | 2020 |  |  |
|  | D09075 | 2020 |  |  |

Select the character codes in the following table.


## Deleting Message Screen

Example Delete the Message Screen 1.
1,2,3...


1. Switch to the Setup Mode.
2. Press the Down button to select Message.
3. Press the OK button to enter the Message menu.
4. Press the Down button to select Delete.
5. Press the OK button to enter the Message Delete Screen.

The present setting is Screen 01.
Note Press and hold the UP button until the Screen No. changes to ALL, all the User Monitor Screen will be deleted if the setting is confirmed.
6. Press the $\mathbf{O K}$ button to delete the screen.
7. Press the ESC or OK button to return to the previous menu.

## 8-6-10 Timer Switch

There are 3 kinds of timer, including Day, Weekly and Calendar Timer. It is possible to register up to 16 timers for each kind.

| Type | Description |
| :--- | :--- |
| Day timer | Sometime in a day, set the related control bit to ON. |
| Weekly timer | Sometime in a week, set the related control bit to ON. |
| Calendar timer | Sometime in a year, set the related control bit to ON. |

## Setting Day Timer

Example
1,2,3...

## Mसमा

 MesT1 Mam:
PL $=4$


घ म Mele ए :C\& पन पौ ए M T एाए

8:30 to 17:15 from Monday to Friday, control bit W509.15 is ON.

1. Switch to the Setup Mode.
2. Press the Down button to select Timer.
3. Press the $\mathbf{O K}$ button to enter the Timer Switch menu.
4. Press the OK button to enter the Day Timer Screen.

The final digit of the Timer No. will be flashing.
The following table shows the setting items.

| No. | Description |
| :---: | :--- |
| a | Timer flag |
| b | Timer No. (01 to 16) |
| c | ON time of PLC |
| d | OFF time of PLC |
| e | Word type |
| f | Word address |

5. Use the Up button to change the Timer No. to 16.

The following table shows the relation between the Timer No. and the control bit when the word address is W001.

| Timer No. | Control bit |
| :--- | :--- |
| 01 | W001.00 |
| 02 | W001.01 |
| 03 | W001.02 |
| 04 | W001.03 |
| $\ldots$ | $\ldots$ |
| 16 | W001.15 |



प :ோ पए पF:ோே प T एाए
 पF:17 1 m Tए एाए

6. Use the Forward button to move the column cursor to the timer flag position. Press the Up button to select the timer flag $\boldsymbol{Y}$. Select the timer flag in the following table.

| Timer flag | Meaning |
| :--- | :--- |
| Y | Timer in use |
| N | Timer not in use |

7. Use the Forward button to move the column cursor to the ON time position.
Use the Up button to change time to 08:30.
8. Use the Forward button to move the column cursor to the ON week position.
The present setting is Monday.
9. Use the Forward button to move the column cursor to the OFF time position.
Use the Up button to change time to 17:15.
10. Use the Forward button to move the column cursor to the OFF week position.
Press the Down or Up button to select Friday.
11. Use the Forward button to move the column cursor to the position of control bit.
12. Press the Up button to select WR.

The following table shows the word type which can be selected, including the default address and the setting range for each type.

| Timer | Word | Default address | Range |
| :--- | :--- | :--- | :--- |
| All | IO | 0100 | 0100 to 6143 |
| Day timer | WR | 001 | 001 to 511 |
| Weekly timer | WR | 002 | 002 to 511 |
| Calender timer | WR | 003 | 003 to 511 |
| All | HR | 000 | 000 to 511 |
| All | AR | 448 | 448 to 959 |


13. Use the Forward button to move the column cursor to the position of word address.
The present setting is the default address.
14. Move the column cursor to the digit to be set.

Use the Up button to change the word address to 509.
15. Press the OK button to save the setting.
16. Press the ESC or OK button to return to the Day Timer Screen.

Setting Calendar Timer
Example From $1^{\text {st }}$ June to $1^{\text {st }}$ October, control bit H209.05 is ON.
1,2,3...

```
M"#
Mए=
    W पूए=
    FI:=4=
```

पानय
$\dot{4}$ पाए
$\mathrm{M}=\mathrm{E}$
Tआय
TIT:
BE Tm
ण्बTm
- T TM
$\checkmark$

```
Mm
    DE TMए
    णE|TM
Y-1 Tाि
    *
```


5. Press the OK button to enter the Calendar Timer Screen.

The final digit of the Timer No. will be flashing.
The following table shows the setting items.

| No. |  |
| :---: | :--- |
| a | Timer flag |
| b | Timer No. (01 to 16) |
| c | ON date of PLC |
| d | OFF date of PLC |
| e | Word type |
| f | Word address |

6. Use the Up button to change the Timer No. to 6.
7. Use the Forward button to move the column cursor to the timer flag position.
Press the Up button to select the timer flag $\boldsymbol{Y}$.
8. Use the Forward button to move the column cursor to the ON date position. Use the Up button to change the date to 06/01.
9. Use the Forward button to move the column cursor to the OFF date position.
Use the Up button to change the date to 10/01.
10. Use the Forward button to move the column cursor to the position of control bit.
Press the Up button to select $\boldsymbol{H R}$.
11. Use the Forward button to move the column cursor to the position of word address.
The present setting is default address.
12. Move the column cursor to the digit to be set. Use the Up button to change the word address to 209.

13. Press the OK button to save the setting.
14. Press the ESC or OK button to return to the Calander Timer Screen.

Note 1. If a timer is in use, when the timer switch turns ON, the LCD Option Board will send command to PLC one time every 1 second to make control bit ON, when the timer switch turns OFF, the LCD Option Board will send command to PLC one time every 1 second to make control bit OFF.
2. Move the LCD Option Board from one PLC to another, the result of timer operation will be different if the time of two PLCs is not the same.

## Timing Curve

Each timer can execute a trans-day, trans-week or trans-year operation. The operation period will be shown in the following curve.

## Day Timer



Weekly Timer


## Calendar Timer



Note Set the OFF date to $1^{\text {st }}$ October, the Calendar Timer will turn OFF at 24:00 $31^{\text {st }}$ September.

## 8-6-11 Data Backup

User can save the user settings to DM memory area from one LCD Option Board and load to other LCD Option Boards from the DM memory area.

Note Please do not take the DM area (D8000 to D8999) for other use.
User settings which can be backed up as shown below.

| User setting | Quantity |  |
| :--- | :--- | :--- |
| User Monitor screen | 16 screens |  |
| Message screen | 16 screens |  |
| Timer Switch | $16 \times 3$ timers |  |
|  | Language | 1 |
|  | Backlight | 1 |
|  | Contrast | 1 |

## Loading User Setting



Load user settings from DM memory area.
1,2,3...

2. Press the Down button to select DataBackup.
3. Press the OK button to enter the Data Backup menu.

पேt-
Tred
$5=$


1. Switch to the Setup Mode.
2. Press the OK button to enter the Load Operation Screen. Select the operation mode in the following table.

| Operation Mode | Meaning |
| :--- | :--- |
| Load | Load user setting from DM area |
| Save | Save user setting into DM area |

5. Press the Down button to select $\mathbf{O K}$.

Note Selecting Cancel will result in a return to the previous menu.

6. Press the OK button to display a load confirming screen.
7. Press the $\mathbf{O K}$ button to start loading.

A rate of loading will be displayed in the screen.
8. When the rate comes up to $100 \%$, the loading is finished. Then it will display a complete screen.
9. Press the ESC or OK button to restart the LCD Option Board.

## Saving User Setting

Example Save user settings to DM memory area.
1,2,3...


1. Switch to the Setup Mode.
2. Press the Down button to select DataBackup.
3. Press the OK button to enter the Data Backup menu.
4. Press the Down button to select Save.

5. Press the OK button to enter the Save Operation Screen.

6. Press the Down button to select $\mathbf{O K}$.

Note Selecting Cancel will result in a return to the previous menu.
7. Press the OK button to display a save confirming screen.
8. Press the OK button to start saving.

A rate of saving will be displayed in the screen.
9. When the rate comes up to $100 \%$, the saving is finished. Then it will display a complete screen.

## 8－6－12 Language Selection

Display for the LCD Option Board is available in 2 languages－English and Japanese．

## Example Change the display language from English to Japanese．

1，2，3．．．

## Mयमा

Mएd
ए।
PT：$-4=$
，


2．Press the Down button to select ゲンゴ．

3．Press the OK button to enter the Language Setup Screen．
The present language is English．

4．Press the Down button to select ニホンゴ ．

5．Press the $\mathbf{O K}$ button to save the setting．
＋ul ए
EE

```
\"%"]
    णघणय"
\\+|=
    ण
```

6．Press the ESC or OK button return to the previous menu． The display language will change to Japanese．

## 8-6-13 PLC Cycle Time

This function can display the cycle time of the CPU Unit. The operation method will be shown in the following example.

1,2,3...


1. Switch to the Setup Mode.
2. Press the Down button to select Other.
3. Press the OK button to enter the Other menu.
4. Press the OK button to enter the Cycle Time Screen.

The average cycle time of the CPU Unit will be displayed.

5. Press the Down button to display the max. cycle time of the CPU Unit.
6. Press the Down button to display the min. cycle time of the CPU Unit.

```
पष्एक्य
M"%:
    #5%
```


## 8-6-14 PLC Clock Setting

## Example <br> Change PLC time to 12:00:00, PLC week to Saturday.

## 1,2,3...

```
My%|
Mme
    # पिएक
    FLC Sed=
```

        .
    1. Switch to the Setup Mode.
2. Press the Down button to select Other.
3. Press the OK button to enter the Other menu.
4. Press the Down button to select ClockSet.
5. Press the OK button to enter the Clock Setup Screen.

The present date, time and week of the CPU Unit will be displayed.

```
Glecse%
C#,
    12:G!C0
        लण%
```

    पाersel
    काद Cl
12ாோே
C

पौल
पौल
घघETM
घघETM
TIme=
TIme=
5 צ世लामए
5 צ世लामए


एथम
QuIETME
पाएव5 $5 \leq \operatorname{sim}$

6. Use the Forward button to move the column cursor to the position of PLC time.
Use the Down or Up button to change the time to 12:00:00.
7. Use the Forward button to move the column cursor to the position of PLC week.
Use the Down or Up button to select Sat.
8. Press the $\mathbf{O K}$ button to save the setting.
9. Press the ESC or OK button to return to the previous menu.
10. Press the ESC button to return to the Monitor Mode.

```
Q4
क्पेश्
    1%ण&
        G
```


## 8-6-15 PLC System Information

This function can display the system information of the CPU Unit. The operation method will be shown in the following example.

1,2,3...


Dt-EGM पन एीच.


```
एण%"
    \Psi\\!m
    &घேए
    \्+ब्पा,
            *
```

T1H
T1H
\&4णए-1
\&4णए-1
L"मए, श्ड
L"मए, श्ड
*
*

1. Switch to the Setup Mode.
2. Press the Down button to select Other.
3. Press the OK button to enter the Other menu.
4. Press the Down button to select SystemInfo.
5. Press the OK button to enter the System Information Screen. Line 1 to 3 will display the CPU Unit model, line 4 the lot No.
Stemmma पnitem 1.2Line 1 to 3 will display the CPU Unit model, line 4 the lot No.
6. Press the Down button to display the CPU Unit version.

## 8-6-16 LCD Backlight Setting

This function can make a setting for the LCD backlight.

## Example

The backlight turns off after LCD has not been used for 5 minutes.
1,2,3...


1. Switch to the Setup Mode.
2. Press the Down button to select Other.
3. Press the $\mathbf{O K}$ button to enter the Other menu.
4. Press the Down button to select BackLight.
5. Press the OK button to enter the Backlight Screen. The following table shows the setting items.

| No. | Description |  | Meaning |
| :---: | :--- | :--- | :--- |
| a | Timer interval |  | The range is 02 to 30 minutes. |
| b | Backlight mode | Timer | Backlight will turn OFF if LCD has not <br> been used for the timer interval. |
|  |  | ON | Backlight is always ON. |
|  |  | OFF | Backlight is always OFF. |

6. Use the Forward button to move the column cursor to the position of timer inerval.
Use the Up button to change the timer interval to $\mathbf{0 5}$.
7. Press the $\mathbf{O K}$ button to save the setting.

Et P

```
एe=
```


## Whym

PEAालt काtret. $\mathrm{FE+5}$
8. Press the ESC or OK button to return to the previous menu.

## 8-6-17 LCD Contrast Setting

This function can make a setting for the LCD contrast.

## Example Change the contrast of LCD display to 8.

1,2,3...


```
M"पा
    C-EEG#
\"
    ण|
```

```
\squareण",
Tr|=TM=
    \squareएबए
    &-4आ!ए
```



## पெका L-y 11 BE



```
\squareण"]
    B+K
F-t,
    FE+TEL
```

        -
    1. Switch to the Setup Mode.
2. Press the Down button to select Other.
3. Press the $\mathbf{O K}$ button to enter the Other menu.
4. Press the Down button to select Contrast.
5. Press the $\mathbf{O K}$ button to enter the Contrast Screen. The contrast level of LCD display is 1 to 16 .
6. Use the Up button to change the level to $\mathbf{0 8}$.
7. Press the $\mathbf{O K}$ button to save the setting.
8. Press the ESC or OK button to return to the previous menu.

## 8-6-18 LCD Factory Setting

This function can initialize the factory setting of the LCD Option Board. The operation method will be shown in the following example.
1,2,3...


1. Switch to the Setup Mode.
2. Press the Down button to select Other.
3. Press the OK button to enter the Other menu.
4. Press the Down button to select FactorySet.
5. Press the OK button to enter the Factory Setting Screen.
6. Press the Down button to select OK.

Note Selecting Cancel will result in a return to the previous menu.
7. Press the $\mathbf{O K}$ button to start initializing.
8. When the initializing is finished, it will display a complete screen.
9. Press the ESC or OK button to restart the LCD Option Board.

## 8-7 Trouble Shooting

## 8-7-1 Symptom at Power ON or during Operation

| Symptom | Probable cause | Possible solution |
| :---: | :---: | :---: |
| No LCD display | LCD connection error or no power supply from PLC. | Check if LCD is connected correctly and the PLC power supply is normal. |
|  | Still in startup waiting time. | It's not error. Just wait a moment. |
| Display EEPROM Error <br> Screen and blinking red backlight | EEPROM is damaged. | Replace the LCD Option Board. |
|  | User settings in EEPROM are corrupted. | Press the ESC button to exit the screen. User settings backed up in EEPROM will be replaced by default settings. Then proceed to reset the screens. (See 8-7-3 for details.) |
| Display NG Screen | LCD connection error. | Check if LCD is connected correctly. |
|  | Communication error between LCD and PLC. | Check the communication setting of PLC, switching DipSW4 to ON. |
| Display Error Screen and blinking red backlight | PLC error | Check PLC according to error code and eliminate the error. |
| Button unresponsing | Button is damaged. | Replace the LCD Option Board. |
|  | User setting error. | Check the settings and change it. |
|  | Noise disturbing. | Retry after the noise is reduced. |
| Display too faint | Backlight is damaged. | Replace the LCD Option Board. |
|  | Contrast level is too low or too high. | Reset the contrast level. |

## 8-7-2 Communication Error Message during Operation

When communication error occurs, the error message will be displayed at the LCD Option Board and the red backlight will blink.

| Error message | Probable cause | Possible solution |
| :---: | :---: | :---: |
| Parity Error or Framing Error or Overrun Error | Communication parameters or conditions of PLC are changed. | Check the communication setting of PLC. |
|  | LCD connection error. | Check if LCD is still connected correctly. |
|  | Noise disturbing. | - Return to normal automatically when the noise is reduced. <br> - If the display cannot return to normal, press the ESC button to restart LCD. |
| FCS Error(Sum check) | Noise disturbing. | - Return to normal automatically when the noise is reduced. <br> - If the display cannot return to normal, press the ESC button to restart LCD. |
| Buffer overflow | The length of receiving data is beyond the range of receiving memory. | Press the ESC button to restart LCD. |
|  | Noise disturbing. |  |
| Connecting Host... | LCD connection error. | Check if LCD is still connected correctly. |
|  | The communication between PLC and LCD is out of service. | Check if PLC is running normally. |
| Response code Error | Operation mistake. | Refer to CJ/CS Communication Manual for solutions according to an end code. |
|  | Noise disturbing. | - Return to normal automatically if the noise is reduced. <br> - Press the ESC button to exit the screen. |
|  | CX-Programmer and LCD execute some function at the same time. | Press the ESC button to exit the screen. Execute this function by either CX-Programmer or LCD. |

## 8-7-3 Deleting EEPROM Error

## 1,2,3...



1. A flashing error screen will be displayed when an error occurs.

The following table shows the display items.

| No. | Description |  |
| :---: | :--- | :--- |
|  | Error type | User Monitor setting error |
|  |  | Message setting error |
|  |  | Timer Switch setting error |
|  |  | Language setting error |
|  |  | Backlight setting error |
|  |  | Contrast setting error |
| b | Screen No. |  |
| c | Line No. |  |

According to the error message, the setting of User Monitor Screen 2, line 4 is corrupted.
2. Press the ESC button to exit the screen. Once the EEPROM Error Screen has disappeared, the display will return to normal.
3. Enter the User Monitor Setup Screen 2, line 4.

User settings backed up in EEPROM are replaced by default settings. Then reset the screen.

## SECTION 9 Ethernet Option Board

This section gives an outline of the Ethernet Option Board, explains how to install and remove the Ethernet Option Board, and how to monitor and make settings required for operation. It also lists the errors during operation and provides countermeasures for troubleshooting.
9-1 Ethernet Option Board Function Guide ..... 554
9-1-1 Overall system configuration example ..... 554
9-1-2 Connecting the CX-Programmer ..... 556
9-2 Features ..... 557
9-3 System Configuration ..... 558
9-3-1 System Configuration ..... 558
9-3-2 Devices Required for Constructing a Network ..... 558
9-4 Specifications ..... 559
9-5 FINS Communications ..... 560
9-5-1 FINS Communications Service Specifications ..... 560
9-5-2 Overview of FINS Communication Service ..... 561
9-6 Part Names ..... 562
9-7 Comparison with Previous Models ..... 563
9-8 Installation and Initial Setup ..... 564
9-8-1 Overview of Startup Procedure ..... 564
9-8-2 Installation and Removing ..... 565
9-8-3 Network Installation ..... 566
9-8-4 Web Browser Setting Function ..... 569
9-9 Memory Allocations ..... 571
9-9-1 CIO Area Allocation ..... 571
9-9-2 DM Area Allocation ..... 573
9-10 Web Browser Setup and Display ..... 577
9-10-1 Multi-language Function ..... 577
9-10-2 Overview of Web Browser Function ..... 577
9-10-3 System ..... 578
9-10-4 HTTP ..... 580
9-10-5 IP Address Table ..... 581
9-10-6 IP Router Table ..... 582
9-10-7 FINS/TCP ..... 583
9-10-8 Unit Information ..... 584
9-10-9 Unit Status ..... 585
9-10-10 FINS Status ..... 586
9-10-11 Error Log ..... 587
9-11 Trouble Shooting ..... 588
9-11-1 Error Log ..... 588
9-11-2 Trouble-shooting with Indicators and Error Code Display ..... 591
9-11-3 Error Status ..... 591
9-12 Sample Application ..... 592

## 9-1 Ethernet Option Board Function Guide

## 9-1-1 Overall system configuration example

Ethernet Option Board provides receiving commands by OMRON standard protocol FINS for CP1L and CP1H programmable controllers. The Ethernet Network Interface allows you to easily connect CP1L and CP1H Programmable Controllers onto new or existing Ethernet network and upload/download programs, communicate between controllers (do not support real-time scanning I/O on Ethernet Option Board).


Note 1. Please use CX-Programmer version 8.1 or higher (CX-ONE version 3.1 or higher).
2. Please use CX-Integrator version 2.33 or higher (CX-ONE version 3.1 or higher) to make the routing table. Except making the routing table for CP1W-CIF41, other functions, such as transferring the parameters and network structure, are not supported by CX-Integrator.
3. Use the Web browser to set the CP1W-CIF41.
4. NS-series HMI version 8.2 or higher can use CP1W-CIF41 through Ethernet.

## 9-1-2 Connecting the CX-Programmer to PLCs Online via Ethernet

## Connecting within the Same Segment

## Connecting through Multiple Segments

## Using Media with

 Unreliable Connections, Such as a Wireless LANConnecting from a Personal Computer with a Dynamic Private IP Address

Use the UDP/IP version of the FINS communications service (i.e., FINS/UDP). FINS/UDP is supported by many OMRON products and is compatible with earlier Ethernet Units (CS1W-ETN01/ETN11/ETN21 and CJ1WETN11/ETN21). The CX-Programmer can be connected and used with FINS/UDP.

Use the TCP/IP version of the FINS communications service (i.e., FINS/TCP). It provides automatic recovery at the TCP/IP layer from communications errors (such as packet loss) that occur during multilevel routing. For CX-Programmer, FINS/TCP can be used to directly connect to the PLC online.
Use the TCP/IP version of the FINS communications service (i.e., FINS/TCP). It provides automatic recovery at the TCP/IP layer from communications errors (such as packet loss) resulting from unreliable connections. For CXProgrammer, FINS/TCP can be used to directly connect to the PLC online.

Depending on whether or not the connection will be within the same segment, either use an IP address conversion method for dynamic IP addresses in the UDP/IP version of the FINS communications service or use the TCP/IP version of the FINS communications service.
It is possible to connect online to a PLC using the CX-Programmer from a computer serving as a temporarily connected node or a permanent DHCP client.
For CX-Programmer, FINS/TCP can be used to directly connect to the PLC online.


## 9-1-3 Receiving Data from OMRON PLCs using Ethernet

## Connecting within the Same Segment

Connecting through Multiple Segments

The CP1W-CIF41 Ethernet Option Board can only support receiving FINS commands from OMRON PLCs using Ethernet.

Use the UDP/IP version of the FINS communications service (i.e., FINS/UDP), and construct applications using the SEND(090), RECV(098), and CMND(490) instructions in the ladder program. FINS/UDP is supported by many OMRON products, and is compatible with earlier Ethernet Units (CS1W-ETN01/ETN11/ETN21 and CJ1W-ETN11/ETN21). The protocol processing for FINS/UDP is simpler than for FINS/TCP, giving FINS/UDP certain advantages in terms of performance. Another feature of FINS/UDP is that it can be used for broadcasting.
On the other hand, with FINS/UDP it is necessary to provide measures, such as retries, for handling communications errors.

Use the TCP/IP version of the FINS communications service (i.e., FINS/TCP), and construct applications using the SEND(090), RECV(098), and CMND(490) instructions in the ladder program. FINS/TCP is the initial function supported by this Ethernet Option Board (CP1W-CIF41). It provides automatic recovery at the TCP/IP layer from communications errors (such as packet loss) that occur during multilevel routing.


## 9-2 Features

## Compatibility and Speed

The transmission medium of Ethernet side has been upgraded to 100BaseTX, while compatibility with some functions and application interfaces of the existing Ethernet Unit models for CS/CJ series has been maintained.
Limited by the Toolbus protocol used on the serial side, the processing speed is only 115.2 kbps , slower than the existing Ethernet Unit. The FINS frame length is less than 1,004 bytes, so the system response performance for the same FINS message applications is longer than the existing Ethernet Unit.

## Various Protocols Available on Ethernet

A variety of protocols make a wide range of applications for use on an Ethernet network. The protocols that can be selected include receiving commands by OMRON's standard protocol FINS and reading Ethernet Option Board settings and status by HTTP.

A communications service can be selected according to need, allowing the PLC to be flexibly integrated with the Ethernet information network.

## Improved FINS Message Communications

The following functions have been maintained according to the existing Ethernet Unit models for CS/CJ series.

- The maximum number of nodes is 254.
- Communications are enabled even if the host computer's IP address is dynamic.
- An automatic client FINS node address allocation function makes it possible to connect online to the PLC even if no FINS node address has been set for the host computer.
- FINS message communications are enabled in both UDP/IP and TCP/IP, but it are only enabled in TCP/IP with up to 2 simultaneous connections . $\rightarrow$ Previously it are enabled in TCP/IP with up to 16 simultaneous connections and all can be set to client.
- Multiple FINS applications, such as the CX-Programmer, on the same computer can be connected online to the PLC via Ethernet.


## Use Web Function to Read Ethernet Option Board Settings and Status

A Web function is provided in Ethernet Option Board.
This enables use of a Web browser to read the Ethernet Option Board's system settings and statuses.

## Full Range of Functions for Handling Troubles

A full range of functions is provided for promptly handling any troubles.

- Self-diagnostic function when power is turned ON.
- Error log for recording error information when an error occurs.


## 9-3 System Configuration

## 9-3-1 System Configuration



## 9-3-2 Devices Required for Constructing a Network

The basic configuration for a 100Base-TX Ethernet System consists of one hub to which nodes are attached in star form using twisted-pair cable. The devices shown in the following table are required to configure a network with 100Base-TX-type CP1W-CIF41, so prepared them in advance.

| Network device | Contents |
| :--- | :--- |
| (1) Ethernet Option Board <br> (CP1W-CIF41) | The Ethernet Option Board is a Communication Unit <br> that connects a CP1H series or CP1L series PLC to <br> 100Base-TX Ethernet networks. <br> (They can also be used as 10Base-T.) |
| (2) Twisted-pair cable | This is twisted-pair cable for connecting 100Base-TX <br> type Ethernet Option Board to the hub, with an RJ45 <br> Modular Connector at each end. <br> Use a category 3, 4, 5, or 5e UTP (unshielded twisted <br> pair) or STP (shielded twisted-pair) cable. |
| (3) Hub | This is a relay device for connecting multiple nodes in a <br> star LAN. |

Recommended Hubs
For detail on recommended devices for constructing a network, refer to 9-8-3 Network Installation.

## 9-4 Specifications

| Item |  | Specifications |  |
| :---: | :---: | :---: | :---: |
| Model number |  | CP1W-CIF41 |  |
| Type |  | 100/10Base-TX (Auto-MDIX) |  |
| Applicable PLCs |  | CP1L and CP1H PLCs |  |
| Unit classification |  | CP1 option port unit |  |
| Mounting location |  | CP1L and CP1H micro PLC option port |  |
| Max. number of Units that can be mounted |  | 2 sets (See note.) |  |
| Size of Buffers |  | 8K bytes |  |
| Transfer | Media access method | CSMA/CD |  |
|  | Modulation method | Baseband |  |
|  | Transmission paths | Star form |  |
|  | Baud rate | $100 \mathrm{Mbit} / \mathrm{s}$ (100Base-TX) | $10 \mathrm{Mbit} / \mathrm{s}$ (10Base-T) |
|  |  | - Half/full auto-negotiation for each port <br> - Link speed auto-sensing for each port |  |
|  | Transmission media | - Unshielded twisted-pair (UDP) cable Categories: 5, 5e <br> - Shielded twisted-pair (STP) cable Categories: $100 \Omega$ at 5, 5e | - Unshielded twisted-pair (UDP) cable Categories: 3, 4, 5, 5e <br> - Shielded twisted-pair (STP) cable Categories: $100 \Omega$ at $3,4,5,5 e$ |
|  | Transmission distance | 100 m (distance between hub and node) |  |
| Current consumption (Unit) |  | 130 mA max. at 5 V DC |  |
| Vibration resistance |  | Conforms to JIS 0040. <br> 10 to 57 Hz : $0.075-\mathrm{mm}$ amplitude, 57 to 150 Hz : acceleration $9.8 \mathrm{~m} / \mathrm{s}^{2}$ in $\mathrm{X}, \mathrm{Y}$, and Z directions for 80 minutes each (sweep time: 8 minutes $\times 10$ sweeps $=80$ minutes) |  |
| Shock resistance |  | Conforms to JIS 0041. <br> $147 \mathrm{~m} / \mathrm{s}^{2}$, 3 times each in $\mathrm{X}, \mathrm{Y}$, and $Z$ directions |  |
| Ambient operating temperature |  | 0 to $55^{\circ} \mathrm{C}$ |  |
| Ambient humidity |  | 10\% to 90\% (with no condensation) |  |
| Atmosphere |  | Must be free of corrosive gas. |  |
| Ambient storage temperature |  | -20 to $75^{\circ} \mathrm{C}$ |  |
| Weight |  | 23 g max. |  |
| Dimensions |  | $36.4 \times 36.4 \times 28.2 \mathrm{~mm}(\mathrm{~W} \times \mathrm{H} \times \mathrm{D})$ |  |

Note 1. Two CP1W-CIF41 (unit version 2.0) can be mounted in the CP1L/CP1H system.
2. One CP1W-CIF41 (unit version 2.0) and one CP1W-CIF41 (unit version 1.0) can be mounted in the CP1L/CP1H system.
3. Only one CP1W-CIF41 (unit version 1.0) can be mounted in the CP1L/CP1H system. If two CP1W-CIF41 are mounted, the CP1W-CIF41 mounted on option board slot 1 will be abnormal and ERR indicator will be ON, the CP1W-CIF41 on option board slot 2 will work normally.

## 9-5 FINS Communications

## 9-5-1 FINS Communications Service Specifications

| Item | Specification |  |
| :---: | :---: | :---: |
| Number of nodes | 254 |  |
| Message Length | 1016 bytes max. |  |
| Date Length | 1004 bytes max. (See note) |  |
| Number of buffer | 14 (1016 bytes $\times 6+240$ bytes $\times 8$ ) |  |
| Protocol name | FINS/UDP method | FINS/TCP method |
| Protocol used | UDP/IP | TCP/IP |
|  | The selection of UDP/IP or TCP/IP is made from the FINS/TCP Tab by Web browser function. |  |
| Number of connections | --- | 2 |
| Port number | 9600 (default) Can be changed. | 9600 (default) Can be changed. |
| Protection | No | Yes (Specification of client IP addresses when unit is used as a server) |
| Other | Items set for each UDP port <br> - Broadcast <br> - Address conversion method | Items set for each connection <br> - Server specification <br> - Remote IP address spec. <br> Server: specify IP addresses of clients permitted to connect. <br> - Automatic FINS node address allocation Specify automatic allocation of client FINS node addresses |
| Internal table | This is a table of correspondences for remote FINS node addresses, remote IP addresses, TCP/UDP, and remote port numbers. It is created automatically when power is turned ON to the PLC or when the unit is restarted, and it is automatically changed when a connection is established by means of the FINS/TCP method or when a FINS command received. <br> The following functions are enabled by using this table. <br> - IP address conversion using the FINS/UDP method <br> - Automatic FINS node address conversion after a connection is established using the FINS/TCP method <br> - Automatic client FINS node address allocation using the FINS/TCP method <br> - Simultaneous connection of multiple FINS applications |  |

Note Refer to the following diagram for the relation between message length and date length.


## 9-5-2 Overview of FINS Communication Service

## Basic Functions

## Upgraded Functions

FINS commands can be received from other PLCs or computers on the same Ethernet network by executing SEND(090), RECV(098), or CMND (490) instructions in the ladder diagram program. This enables various control operations such as the reading and writing of I/O memory between PLCs, mode changes, and file memory operations.


Executing, from the host computer, FINS commands with UDP/IP or TCP/IP headers enables various control operations, such as the reading and writing of I/O memory between PLCs, mode changes, and file memory operations.
For example, it is possible to connect online via Ethernet from FINS communications applications such as the CX-Programmer, and to perform remote programming and monitoring.

With the CP1W-CIF41, the following functions have been upgraded.

- The FINS communications service can be executed not only with UDP/IP but also with TCP/IP, and it is even possible to use FINS communications with both UDP/IP and TCP/IP together on the same network. Using TCP/IP makes FINS communications highly reliable.
- Even if the IP address and UDP port number of the host computer (a DHCP client computer) are changed, it is still possible for the host computer to send FINS commands to PLCs on the Ethernet network and to receive responses. When UDP is used, either the automatic generation (dynamic) method or the IP address table method must be selected for IP address conversion. When TCP is used, changes in IP address and TCP port numbers are handled automatically.
- Multiple FINS applications (CX-Programmer and user-created application programs) at the same computer can be connected online to a PLC via Ethernet (using either TCP/IP or UDP/IP).

Note The message service does not guarantee that a message will reach the destination node. A message may be lost during transmission due to factors such as noise. To prevent this from occurring when using message services, it is common to set up retry processing at the node from which instructions are issued. With the SEND(090), RECV(098), and CMND(490) instructions, retry processing is executed automatically by specifying the number of retries, so specify a number other than 0 .

## 9-6 Part Names



## LED Indicators

| Indicator | Color | Status | Meaning |
| :--- | :--- | :--- | :--- |
| COMM | Yellow | Not lit | Not sending or receiving data. |
|  |  | Flashing | Sending or receiving data. |
| ERR | Red | Not lit | Unit normal. |
|  |  | Lit | An fatal error has occurred at the Unit. |
|  |  | Flashing | An no-fatal error has occurred at the unit. |

## 9-7 Comparison with Previous Models

|  | Item | Previous Models | New Models |
| :---: | :---: | :---: | :---: |
| Model number |  | CS1W-ETN21 CJ1W-ETN21 | CP1W-CIF41 |
| Physical layer |  | 100/10Base-TX | 100/10Base-TX (Auto-MDIX) |
| Number of nodes |  | 254 | 254 |
| PLC maintenance via the Internet |  | Can send commands, including FINS commands, by e-mail over the Internet from a computer to the PLC. | Not supported |
| Server specification |  | Specification by IP address or by host name | Not supported |
| FINS communications service | Automatic IP address acquisition | A computer automatically acquiring IP addresses can send commands to the PLC and receive responses. | Same as previous models |
|  | FINS communication with computer without fixed node address | Possible, with Automatic allocation by Ethernet Option Board (Client FINS automatic node address allocation function, TCP/IP only) | Same as previous models |
|  | Handling TCP/IP | With FINS communications, both UDP/IP and TCP/IP (with up to16 simultaneous connections and all can be set to client) possible. | With FINS communications, both UDP/IP and TCP/IP (with up to 2 simultaneous connections) possible. |
|  | Simultaneous connection of multiple applications in a computer | Possible (with both UDP/IP and TCP/IP) | Same as previous models |
| Mail functions |  | Supported | Not Supported |
| FTP server function |  | Supported | Not Supported |
| Socket services function |  | Supported | Not Supported |
| Automatic clock information adjustment |  | Supported | Not Supported |
| FINS frame length |  | 2012 | 1016 |
| Buffer numbers |  | 392 (2K bytes $\times 392$ ) | 14 (1016 bytes $\times 6+240$ bytes $\times 8$ ) |
| Inner bus |  | Parallel | Serial port |

Note Limited by the CP1W-CIF41 inner bus protocol (Toolbus, 115200kbps), the system response performance is longer than the existing Ethernet Unit. Please consider the FINS command processing time and buffer limitation when using the CP1W-CIF41 Ethernet Option Board.

Comparison with CP1W-CIF41 Version 1.0

| Item | CP1W-CIF41 Version 1.0 | CP1W-CIF41 Version 2.0 |
| :--- | :--- | :--- |
| Reset system settings function | Not supported | Turn on the Reset Flags A525.00/01 <br> in the Auxiliary Area. |
| Restart function | Click the Restart Button from the Web <br> browser. | Click the Restart Button from the <br> Web browser. <br> Turn on the Restart Flags <br> A525.08/09 in the Auxiliary Area. |
| Max. number of units that can be <br> mounted | 1 set | 2 sets (CP1H, CP1L M-type) <br> 1 set (CP1E, CP1L L-type) |
| PLC Modules that can be supported | CP1H, CP1L | CP1H, CP1L, CP1E |

## 9-8 Installation and Initial Setup

## 9-8-1 Overview of Startup Procedure

The following procedure is the same for the CS Series and CJ Series.


Note 1. When using this method, always leave the local IP address of system setup in the Ethernet Option Board set to the value of 0.0 .0 .0 . If this area contains any other value, any setting made in the allocated CIO words will be overwritten with it.
2. The local IP address and other parameters can be set from the Web browser.
3. It is not necessary step, and the CX-Integrator version 2.33 or higher (CX-ONE version 3.1 or higher) is required.
When the FINS communications service is used, routing tables must be created in advance. Routing tables are required in the following circumstances.

- When communicating with a PLC or computer on another network (e.g. remote programming or monitoring using FINS message or a CX-programmer)
- When multiple Communications Units are mounted to a single PLC (e.g. CPU unit)
- When routing tables are used for one or more other nodes on the same network

4. It is not necessary step, and the Web browser is required.

## 9-8-2 Installation and Removing

The following processing explains how to install and remove an Ethernet Option Board.

Caution Always turn OFF the power supply to the CPU unit and wait until all the operation indicators go out before installing or removing the Ethernet Option Board.

## Installation

1,2,3... 1. Press the up/down lock-levers on both sides of the Option Board slot cover at the same time to unlock the cover, and then pull the cover out.
2. Check the alignment to make the corner cut of the Ethernet Option Board fit in the Option Board slot, and firmly press the Ethernet Option Board in until it snaps into place.


Note If two CP1W-CIF41 (unit version 1.0) Ethernet Option Boards are mounted on the CP1L/CP1H PLC, the CP1W-CIF41 mounted on Option Board slot1 (left side) will run in abnormal status and ERR indicator will be ON. If the ladder program operates the with CP1W-CIF41 fatal error, the PLC will generate the non-fatal error.
3. For CPU Units with 30,40 or 60 I/O points, switch DipSW4 of the CPU unit to ON, if the Ethernet Option Board is mounted on the Option Board slot 1 (left side). Switch DipSW5 of the CPU unit to ON, if the Ethernet Option Board is mounted on the Option Board slot 2 (right side).
For CPU Units with 14 or 20 I/O points, switch DipSW4 of the CPU unit to ON.

Note DipSW4 and DipSW5 are OFF at shipment.


Removing
Press the up/down lock-levers on both sides of the Ethernet Option Board at the same time to unlock the Ethernet Option Board, and then pull it out.


## 9-8-3 Network Installation

## Basic Installation Precautions

- Take the greatest care when installing the Ethernet System, being sure to follow ISO 802-3 specifications. You must obtain a copy of these specifications and be sure you understand them before attempting to install an Ethernet System. Unless you are already experienced in installing communications systems, we strongly recommend that you employ a professional to install your system.
- Do not install Ethernet equipment near sources of noise. If noise-prone environments are unavoidable, be sure to take adequate measures against noise interference, such as installing network components in grounded metal cases, using optical links in the system, etc.

The following products are recommended for use with the Ethernet Option Board.

| Part | Maker | Model number | Specifications | Inquires |
| :---: | :---: | :---: | :---: | :---: |
| Hub | 100BASE-TX |  |  |  |
|  | OMRON | W4S1-03B | 10/100 Mbit/s 3-port hub |  |
|  | OMRON | W4S1-05B (C) | 10/100 Mbit/s 5-port hub |  |
|  | $\begin{aligned} & \text { PHOE- } \\ & \text { NIX } \\ & \text { CON- } \\ & \text { TACT } \end{aligned}$ | $\begin{aligned} & \text { SWITCH } \\ & \text { 5TX } \end{aligned}$ | 10/100 Mbit/s 5-port hub |  |
|  | Allied Telesis | RH509E | 9-port hub | Allied Telesis (0120) 86-0442 (in Japan only) |
|  |  | MR820TLX | 9-port hub with 10Base-5 backbone port |  |
|  | 10BASE-T |  |  |  |
|  | Allied Telesis | RH509E | 9-port hub | Allied Telesis (0120) 86-0442 (in Japan only) |
|  |  | MR820TLX | 9-port hub with 10Base-5 backbone port |  |
| Twisted-pair cable | 100BASE-TX |  |  |  |
|  | Fujikura | $\begin{array}{\|l\|} \hline \text { F-LINK-E } \\ 0.5 \mathrm{~mm} \times 4 \mathrm{P} \end{array}$ | STP (shielded twistedpair) cable: Category 5, 5e <br> Note: Impedance is limited to $100 \Omega$. | --- |
|  | Fujikura | $\begin{aligned} & \text { CTP-LAN5 } \\ & 0.5 \mathrm{~mm} \times 4 \mathrm{P} \end{aligned}$ | UTP (unshielded twistedpair) cable: Category 5, 5e |  |
|  | 10BASE-T |  |  |  |
|  | Fujikura | $\begin{aligned} & \text { F-LINK-E } \\ & 0.5 \mathrm{~mm} \times 4 \mathrm{P} \end{aligned}$ | STP (shielded twistedpair) cable: Category 3, 4, 5, 5e <br> Note: Impedance is limited to $100 \Omega$. |  |
|  | Fujikura | $\begin{aligned} & \text { CTP-LAN5 } \\ & 0.5 \mathrm{~mm} \times 4 \mathrm{P} \end{aligned}$ | UTP (unshielded twistedpair) cable: Category 3, $4,5,5 e$ |  |
| Connectors (Modular plug) | STP Plug |  |  |  |
|  | Panduit Corp | MPS588 | --- |  |
|  | UTP Plug |  |  |  |
|  | Panduit | MP588-C | --- |  |

## Precautions on Laying Twisted-pair Cable

Basic Precautions

## Environment Precautions

- Press the cable connector in firmly until it locks into place at both the hub and the Ethernet Option Board.
- After laying the twisted-pair cable, check the connection with a 10Base-T cable tester.
- The UTP cable is not shielded, and the hub is designed for use in OA environments. In environments subject to noise, construct a system with shielded twisted-pair (STP) cable and hubs suitable for an FA environment.


## Hub Installation Environment Precautions

## Hub Connection Methods

## Ethernet Connectors

The following standards and specifications apply to the connectors for the Ethernet twisted-pair cable.

- Electrical specifications: Conforming to IEEE802.3 standards
- Connector structure: RJ45 8-pin Modular Connector (conforming to ISO8877)


| Connector Pin | Signal Name | Abbr. | Signal Direction |
| :--- | :--- | :--- | :--- |
| 1 | Transmission data + | TD+ | Output |
| 2 | Transmission data - | TD- | Output |
| 3 | Reception data + | RD+ | Input |
| 4 | Not used | --- | --- |
| 5 | Not used | --- | --- |
| 6 | Reception data - | RD- | Input |
| 7 | Not used | --- | --- |
| 8 | Not used | --- | --- |
| Hood | Frame ground | FG | --- |

## Connecting the Cable

4 Caution Turn OFF the PLC's power supply before connection or disconnecting twistedpair cable.

Caution Allow enough space for the bending radius of the twisted-pair cable.
1,2,3... 1. Lay the twisted-pair cable.
2. Connect the cable to the hub. Be sure to press in the cable until it locks into place.
Request cable installation from a qualified professional.
3. Connect the cable to the connector on the Ethernet Option Board. Be sure to press in the cable until it locks into place.

## 9-8-4 Web Browser Setting Function

The Ethernet Option Board's system settings can be set using the Web browser of a personal computer or other device. The Ethernet Option Board's Web window is displayed by accessing the following URL from the Web browser.
English page: http://(Ethernet Option Board's IP address)/E00.htm
Japanese page: http://(Ethernet Option Board's IP address)/J00.htm
Chinese page: http://(Ethernet Option Board's IP address)/C00.htm

In this example, use the following procedure to set the IP address using Internet Explorer version 6.0 and the Ethernet Option Board's English Web pages.

1,2,3... 1. Connect to the Ethernet Option Board from the Web browser using the Ethernet Option Board's default IP address.
http://192.168.250.1/E00.htm

2. Input the default password "ETHERNET" and click the Login Button.

3. Select Settings from the menu on the left side of the window to display the Settings Menu.

4. Select 1. IP address and Protocols - System to display System menu.

5. Make the required settings (i.e., the IP address in this example).

| System Format |
| :--- |
| Parameter  Value  <br> IP Address 192 .168 .250 <br> Subnet Mask 255 .255 .255 <br> FINS Node Address 1 [0: default(1)]  <br> FINS/UDP Port 0 $\square$ Use Input Port No [Default( 9600$)$ ]  <br> FINSSTCP Port 0 $\square$ Use Input Port No [Default(9600)]  |

6. After entering the correct values, click the Transfer Button to transfer the settings to the Ethernet Option Board.

7. To enable the new settings, turn the power to the Ethernet Option Board OFF and ON again, or click the Restart Button.

## 9-9 Memory Allocations

## 9-9-1 CIO Area Allocation

The memory allocation about communication services status in the CIO area of PLC is shown as the following diagram. The beginning CIO channel m is calculated by the following equation:


The following table describes the unit address for each option port.

| Option Port No. | I/O Capacity | Unit Address | Range of Status Area |
| :--- | :--- | :--- | :--- |
| Option port 1 | $14 / 20$ | 0xFC | CIO2990 to CIO2992 |
|  | $30 / 40 / 60$ | 0xFD | CIO2980 to CIO2982 |
| Option port 2 | $30 / 40 / 60$ | 0xFC | CIO2990 to CIO2992 |

## Service Status

m


| Bit | Name | Unit operation |
| :--- | :--- | :--- |
| 0 to 13 | Reserved | Always 0. |
| 14 | Link Status | 0: The link between hubs is terminated. <br> 1: A link is established between hubs. |
| 15 | Reserved | Always 1. |

Caution Bit 15 is used for detect power condition of PLC, so do not change it at any time. Otherwise the CP1W-CIF41 Ethernet Option Board will generate error.

## Error Status

The status of errors that occur at the Ethernet Option Board is reflected as shown in the following diagram.


| Bit | Name | Correction |
| :--- | :--- | :--- |
| 0 to 1 | Reserved | Always 0. <br> 2 <br> error |
| 3 | IP address setting <br> error | The following cannot be used as IP address set- <br> tings. <br> - Host IDs that are all 0 or all 1. <br> - Network IDs that are all 0 or all 1. <br> - Subnetwork IDs that are all 1. <br> - Addresses beginning with 127 (7F hex). <br> Reset the IP address. |
| 4 | IP router table error | Correct the IP address table. If the problem cannot <br> be resolved, <br> replace the CPU Unit. |
| 5 | Correct the IP router table. If the problem cannot be <br> resolved, replace the CPU Unit. |  |
| 6 | Routing table error | Always 0. <br> Correct the routing tables. If the problem cannot be <br> resolved, replace the CPU Unit. |
| 7 to 13 | Reserved | Always 0. <br> 14 <br> Address disagree- <br> mentMake sure that the node number and the last byte <br> of the IP address are the same and then set other <br> host IDs to 0. Change the address conversion <br> method. |
| 15 | EEPROM error | Restart the PC. If the problem cannot be resolved, <br> replace the Ethernet Option Board. |

## FINS/TCP Connection Status



| Bit | Switch | Unit operation |
| :--- | :--- | :--- |
| 0 | FINS/TCP Connection No.1 | 0: The connection is terminated. <br> 1: A connection is established. |
| 1 | FINS/TCP Connection No.2 | 0: The connection is terminated. <br> 1: A connection is established. |
| 2 to 15 | Reserved | Always 0. |

## 9-9-2 DM Area Allocation

The memory allocation about system setup is shown as the following diagram. These data will be allocated to the DM area of PLC. The beginning DM channel $n$ is calculated by the following equation.

Note 1. DM area from n to $\mathrm{n}+154$ can only display all of the settings stared in the unit. Modification in this area is invalid to the CP1W-CIF41 Ethernet Option Board.
2. DM area $n+155$ and $n+156$ will display the IP address used by the CP1WCIF41 when the power is turned ON.
3. When the IP address is illegal, such as using CLASS D, CLASS E IP address, the values in words $\mathrm{n}+3$ and $\mathrm{n}+155$ will be different, and the CP1W-CIF41 will temporarily use the default IP address (192.168.250.1). Use this IP address to modify the IP address settings through Web browser.
4. When the system settings are wrong, A525 can be used to reset CP1WCIF41.
Refer to Appendix $C$ and Appendix $D$ for details.

$$
\mathrm{n}=\mathrm{DM} 32000+300 \times(0 x F D-\text { Unit Address) }
$$

| Offset | D15 | D0 |
| :---: | :---: | :---: |
| n | Mode setting (2 bytes) |  |
| $\mathrm{n}+1$ | FINS/TCP port number (2 bytes) |  |
| $\mathrm{n}+2$ | FINS/UDP port number (2 bytes) |  |
| $\mathrm{n}+3$ | IP address (4 bytes) |  |
| $\mathrm{n}+5$ | Subnet mask (4 bytes) |  |
| $\mathrm{n}+7$ | Reserved (2 bytes) |  |
| $\mathrm{n}+8$ | IP address table (194 bytes) |  |
| $\mathrm{n}+105$ | IP router table (66 bytes) |  |
| $\mathrm{n}+138$ | FINS/TCP connection setup (22 bytes) |  |
| $\mathrm{n}+149$ | HTTP server setup (10 bytes) |  |
| $\mathrm{n}+154$ | FINS node address (2 bytes) |  |
| $\mathrm{n}+155$ | Using IP Address Display/Setting Area (4 bytes) |  |

The following table describes the unit address for each option port.

| Option Port No. | I/O Capacity | Unit Address | Range of Status Area |
| :--- | :--- | :--- | :--- |
| Option port 1 | $14 / 20$ | OxFC | DM32300 to DM32456 |
|  | $30 / 40 / 60$ | 0xFD | DM32000 to DM32156 |
| Option port 2 | $30 / 40 / 60$ | 0xFC | DM32300 to DM32456 |

## Mode Setting



| Bit | Mode | Settings |
| :--- | :--- | :--- |
| 0 | Reserved | Always 0. |
| 1 | Broadcast address | 0: 4.3BSD specifications <br> $1: 4.2$ BSD specifications |
| 2 to 3 | IP address conversion method | 00, 01: Automatic generation method <br> 10: IP address table reference method <br> 11: Combined method |
| 4 | FINS/UDP port number | 0: Default (9600) <br> $1:$ Unit setup value |
| 5 | FINS/TCP port number | 0: Default (9600) <br> $1:$ Unit setup value |
| 6 | FINS/UDP destination IP mode | 0: Enable (Dynamically) <br> $1:$ Disable (Static) |
| 7 | FINS/TCP protect function | 0: Disable (Only FINS/TCP Server) <br> $1:$ Enable (Only FINS/TCP Server) |
| 8 to 15 | Reserved | Always 0. |

## FINS/TCP and FINS/UDP Port Number



When displaying 0000, the port number is 9600 .

## IP Address



The IP address is (1)(2).(3)(4).(5)(6).(7)(8) (hex)

## Subnet Mask



The Subnet mask is (1)(2).(3)(4).(5)(6).(7)(8) (hex)

## IP Address Table

## IP Router Table



Pointer of IP Address Table
Point to the last recorder in IP address table. For example, if the last recorder number in IP address table is 6 , the value of this channel is 6 .

## IP Address Table Records

Each IP address table record has 6 bytes. The max number of records is 32 . The configuration of the 6 bytes of data in each record is as shown in the following diagram.


## Pointer of IP Router Table

Point to the last recorder in IP router table. For example, if the last recorder number in IP router table is 6 , the value of this channel is 6 .

## IP Router Table Records

Each IP router table record has 8 bytes. The max number of records is 8 . The configuration of the 8 bytes of data in each record is as shown in the following diagram.


FINS/TCP Connection
Setup


FINS/TCP Connection No. 1 to 2


| Bit | Settings | Unit operation |
| :--- | :--- | :--- |
| 0 | Reserved | Always 0. |
| 1 | Protect setting | 0: The IP address of FINS/TCP connection No. 1 is not <br> under the protection. <br> $1:$The IP address of FINS/TCP connection No. 1 is under <br> the protection. <br> 2 to 4 <br> 5 <br> Reserved <br> Protect setting <br> Always 0. <br> 6 to The IP address of FINS/TCP connection No.2 is not <br> under the protection. <br> 1: The IP address of FINS/TCP connection No. 2 is under <br> the protection. |

Each FINS/TCP connection number has 5 bytes. The configuration of the 5 bytes of data in each number is as shown in the following diagram.


If the password for accessing the Ethernet Option Board's Web page is forgotten, find out it in this area. It is written in ASCII format.

The setting range is $0 \sim$ FE (hex).


The IP address is (1)(2).(3)(4).(5)(6).(7)(8) (hex)
If the local IP address in the system setup is set to a value other than 0.0.0.0, this area will act as an IP address display area and the local IP address in the system setup will be read and stored here when the power is turned ON or the Ethernet Option Board restarted.
If the local IP address in the system setup is set to 0.0.0.0, this area will act as an IP address setting area. The value will be read by the Ethernet Option Board when the power is turned ON or the Ethernet Option Board restarted and is used as the local IP address.

If the IP address for accessing the Ethernet Option Board through Web browser is forgotten, find out it in this area.

Note When IP address in system setup area and DM area are all set to 0.0.0.0, the IP address will be 192.168.250.1 (FINS node address).

## 9-10 Web Browser Setup and Display

## 9-10-1 Multi-language Function

The WEB server supports the multi-language function. The supported languages are English, Chinese and Japanese.
Before setting, users should select the appropriate language in the following ULC.
English page: http://(Ethernet Option Board's IP address)/E00.htm
Japanese page: http://(Ethernet Option Board's IP address)/J00.htm
Chinese page: http://(Ethernet Option Board's IP address)/C00.htm

## 9-10-2 Overview of Web Browser Function

System setup for the Ethernet Option Board is as follows.

| Setting Item |  | Reference Page |
| :---: | :---: | :---: |
| System | IP Address | 578 |
|  | Subnet Mask | 578 |
|  | FINS Node Address | 578 |
|  | FINS/UDP Port | 578 |
|  | FINS/TCP Port | 578 |
|  | Address Conversion Mode | 578 |
|  | FINS/UDP Option | 578 |
|  | Broadcast Option | 579 |
|  | FINS/TCP Protected | 579 |
| HTTP | WEB Password | 580 |
|  | Port No. | 580 |
| IP Address Table | FINS Node Address | 581 |
|  | IP Address | 581 |
| IP Router Table | IP Network Address | 582 |
|  | Router IP Address | 582 |
| FINS/TCP | IP Address | 583 |
|  | Auto-allocated FINS Node | 583 |

Monitor status for the Ethernet Option Board is as follows.

| Status | Reference Page |
| :--- | :--- |
| Unit information | 584 |
| Unit status | 585 |
| FINS status | 586 |
| Error log | 587 |

## 9-10-3 System



## System Format

| Item | Contents | Default |
| :---: | :---: | :---: |
| IP Address | Set the local IP address for the Ethernet Option Board. <br> - Setting range: 00.00.00.00 to 223.255.255.255 | 192.168.250.1 |
| Subnet Mask | Set the subnet mask for the Ethernet Option Board. <br> This is required if a method other than the IP address table method is used for address conversion. | 255.255.255.0 |
| FINS Node Address | Set the local FINS node address for the Ethernet Option Board. <br> - Setting range: 1 to 254 | 1 |
| FINS/UDP Port | Specify the local UDP port number to be used for the FINS communications service. The UDP port number is the number used for UDP identification of the application layer (i.e., FINS communications service). <br> - Setting range: 1 to 65,535 | 9,600 |
| FINS/TCP Port | Specify the local TCP port number to be used for the FINS communications service. The TCP port number is the number used for TCP identification of the application layer (i.e., the FINS communications service in this case). <br> - Setting range: 1 to 65,535 <br> Note Make the settings so that TCP port number 80 for HTTP does not overlap. <br> The port number setting only has an effect on the FINS/TCP server function, not on the FINS/TCP client function. FINS/TCP client port will switch from 3000 to 65535 automatically for connection. | 9,600 |
| Use Input Port No. | Check this checkbox, if you want the user defined port number for UDP or TCP to be in effect. | Unchecked |
| Address Conversion Mode | Select any of the following as the method for finding and converting IP addresses from FINS node addresses. (Enabled for FINS/UDP only.) <br> - Auto (dynamic): Automatic generation (dynamic) <br> - Auto (static): Automatic generation (static) <br> - Manual: IP address table method <br> - Auto \& Manual: Combined method | Auto (dynamic) |
| FINS/UDP Option | Select to dynamically change the remote (destination) IP address for FINS/UDP or not. To prohibit dynamic changes, check the second box. | Change dynamically |


| Item | Contents | Default |
| :--- | :--- | :--- |
| Broadcast Option | Set the method for specifying IP addresses for broadcasting in <br> FINS/UDP. | All '1' (4.3BSD) |
|  | - All '1' (4.3BSD): Broadcast with host number set to all ones. |  |
|  | - All '0' (4.2BSD): Broadcast with host number set to all zeros. |  |
| Normally the default setting should be used. |  |  |$\quad$| FINS/TCP Protected |
| :--- | | When this option is selected, if the FINS/TCP connection is set to a |
| :--- |
| server, and if an IP address other than 0.0.0.0 is set to destination IP |
| address, any connection request from other than the setting IP address |
| will be denied. |
| Select this option to prevent faulty operation (by FINS commands) from |
| specific nodes from affecting the PLC. |$\quad$|  |
| :--- |

The functions of the buttons are as follows.

| Button | Function |
| :--- | :--- |
| Transfer | Transfer the entered values from the personal computer to the Ethernet Option Board. (The <br> new settings are invalid until the Ethernet Option Board has been reset.) |
| Cancel | Cancel the entered values. |
| Restart | Restart the Ethernet Option Board to enable the new settings after transfer. <br> The Restart button is invalid to the PLC. |

## 9-10-4 HTTP



HTTP Server Setup

| Item | Contents | Default |
| :--- | :--- | :---: |
| WEB Password | Set the password for accessing the Ethernet <br> Option Board's settings and status monitor- <br> ing information. | ETHERNET |
| Port Number | Set the port No. used to connect to the Web <br> browser. | 80 |

The functions of the buttons are as follows.

| Button | Function |
| :--- | :--- |
| Transfer | Transfer the entered values from the personal computer to the <br> Ethernet Option Board. (The new settings are invalid until the <br> Ethernet Option Board has been reset.) |
| Cancel | Cancel the entered values. |
| Restart | Restart the Ethernet Option Board to enable the new settings <br> after transfer. <br> The Restart button is invalid to the PLC. |

## 9-10-5 IP Address Table

Set the IP address table that defines the relationship between FINS node addresses and IP addresses. With FINS/UDP, this is enabled only when the IP address table method is set to the IP address conversion method.


| Item | Contents | Default |
| :--- | :--- | :--- |
| FINS Node Address | Set the node address for the remote device. | None |
| IP Address | Set the related IP address for the remote device. | None |

The functions of the buttons are as follows.

| Button | Function |
| :--- | :--- |
| Transfer | Transfer the entered values from the personal computer to the <br> Ethernet Option Board. (The new settings are invalid until the <br> Ethernet Option Board has been reset.) |
| Cancel | Cancel the entered values. |
| Restart | Restart the Ethernet Option Board to enable the new settings after <br> transfer. <br> The Restart button is invalid to the PLC. |
| Show | Show the FINS node address and IP address of the selected No. |
| Delete | Delete the IP address table of the selected No. |

## 9-10-6 IP Router Table

Set the IP router table when the Ethernet Option Board is to communicate through the IP router with nodes on another IP network segment.

| OMRON Ethernet Option Board <br> [Settings ] <br> Menu <br> 1. IP Address and Protocols <br> - System <br> - HTTP <br> 2. IP Address/Router Table <br> - IP Address Table <br> - IP Router Table <br> 3. FINS/TCP <br> - Connection |
| :---: |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

IP Router Table
Setting Form


Transfer Cancel Restart
Setting List

| No IP Network Address | Router IP Address |
| :--- | :--- | :--- |

$01 \vee$ Show Delete

| Item | Contents | Default |
| :--- | :--- | :--- |
| IP Network <br> Address | Set the network ID from the IP address. | None |
| Router IP <br> Address | Set the related IP address of a router connected to a <br> network. | None |

The functions of the buttons are as follows.

| Button | Function |
| :--- | :--- |
| Transfer | Transfer the entered values from the personal computer to the <br> Ethernet Option Board. (The new settings are invalid until the <br> Ethernet Option Board has been reset.) |
| Cancel | Cancel the entered values. |
| Restart | Restart the Ethernet Option Board to enable the new settings after <br> transfer. <br> The Restart button is invalid to the PLC. |
| Show | Show the IP network address and Router IP address of the <br> selected No. |
| Delete | Delete the IP router table of the selected No. |

## 9-10-7 FINS/TCP



## FINS/TCP Connection Setup

| Item | Contents | Default |
| :---: | :---: | :---: |
| No. | Shows the connection number. This is a network API used when TCP is used for the FINS communications service. Up to 2 can be used at a time, and they are identified by connection numbers 1 to 2 . The Ethernet Option Board can thus simultaneously execute the FINS communications service by TCP with up to 2 remote nodes. | --- |
| IP Address | - When the Ethernet Option Board is used as a server: If the option is selected to use IP addresses to protect, set the IP addresses as required at clients from which connection is permitted. If not set for those connections, the default setting can be used. <br> -When the Ethernet Option Board is used as a client: Set the IP address for the remote Ethernet Unit (i.e., the server) that is to be connected by FINS/TCP. It is required that an IP address be set for the remote Ethernet Unit. | 0.0.0.0 |
| Auto-allocated FINS node | If the client (normally a personal computer) application supports FINS/TCP, and if FINS node addresses are not fixed, the client will take 0 as its node address. Then, when a FINS command arrives, the number set here (from 251 to 252) will automatically be allocated as the client's FINS node address. | From 251 to 252, for connection No. 1 to 2 |

The functions of the buttons are as follows.

| Button | Function |
| :--- | :--- |
| Transfer | Transfer the entered values from the personal computer to the <br> Ethernet Option Board. (The new settings are invalid until the <br> Ethernet Option Board has been reset.) |
| Cancel | Cancel the entered values. |
| Restart | Restart the Ethernet Option Board to enable the new settings <br> after transfer. <br> The Restart button is invalid to the PLC. |

## 9-10-8 Unit Information



| Parameter | Contents |
| :--- | :--- |
| Model | Show the model information of the Ethernet Option <br> Board. |
| Version | Show the version information of the Ethernet Option <br> Board. |
| IP Address | Show the IP address of the Ethernet Option Board. |
| Subnet Mask | Show the subnet mask of the Ethernet Option Board. |
| FINS/UDP Port Number | Show the FINS/UDP port number of the Ethernet <br> Option Board. |
| Use Input Port Number | Show the effective port number setting mode. |
| Broadcast Setting | Show the broadcast setting of the Ethernet Option <br> Board. |
| IP Address Conversion | Show the IP address conversion method of the Ether- <br> net Option Board. |
| Ethernet Address | Show the MAC ID of the Ethernet Option Board. |

## 9-10-9 Unit Status



| Parameter | Contents |
| :--- | :--- |
| Error Flags | Indicate the operating status and errors that occurred <br> when the Ethernet Option Board is turned ON. |
| Total Number of Packets <br> Received | Show the total number of packets received by the Ether- <br> net Option Board. |
| Total Number of Receive <br> Errors | Show the total number of errors detected while the Ether- <br> net Option Board was receiving. <br> The types of errors detected are short packet errors, <br> alignment errors, CRC errors, frame length errors and <br> communication controller overflow errors. |
| Total Number of Packets <br> Sent | Show the total number of packets sent by the Ethernet <br> Option Board. |
| Total Number of Errors <br> Sent | Show the total number of errors detected while the Ether- <br> net Option Board was sending. |

## 9-10-10 FINS Status



| Parameter | Contents |
| :--- | :--- |
| Node | Show the FINS node address. |
| Connection Type | Show the protocol used by connection with the related <br> node address. |
| Local Port No. | Show the port number of the Ethernet Option Board for <br> connection with the related node address. |
| Remote IP | Show the IP address of the related node address. |
| Remote Port No. | Show the remote port number of the related node address <br> for connection. |
| TCP Connection No. | If the connection is the FINS/TCP, show the connection <br> number (1 to 4). |
| TCP Status | If the connection is the FINS/TCP, show the current con- <br> nection status. |

The details of TCP status are listed as the following table.

| Status | Meaning |
| :--- | :--- |
| CLOSED | Connection closed |
| LISTEN | Waiting for connection |
| SYN SENT | SYN sent in active status |
| SYN RECEIVED | SYN received and sent |
| ESTABLISHED | Already established |
| CLOSE WAIT | FIN received and waiting for completion |
| FIN WAIT1 | Completed and FIN sent |
| CLOSING | Completed and exchanged FIN. Awaiting ACK. |
| LAST ACK | FIN sent and completed. Awaiting ACK. |
| FIN WAIT2 | Completed and ACK received. Awaiting FIN. |
| TIME WAIT | After closing, pauses twice the maximum segment life (2MSL). |

The function of the button is as follows.

| Button | Function |
| :--- | :--- |
| Send | Show the FINS status of the selected No. |

## 9-10-11 Error Log



| Parameter | Contents |
| :--- | :--- |
| No. | Show the error recorder number. |
| Error Code | Show the error code of the error recorder. |
| Detail Code | Show the detail error code of the error recorder. |
| Date | Show the date of the error recorder. |

The functions of the buttons are as follows.

| Button | Function |
| :--- | :--- |
| Send | Show the error log of the selected No. |
| Clear Error Log | Clear the error log table. |

## 9-11 Trouble Shooting

## 9-11-1 Error Log

Logged Errors

Error Log Table

Error Log Location

The Ethernet Option Board provides an error log that records errors occurred during Ethernet Option Board operation. The contents of the error log can be read or cleared from the Web Brower.

The following errors are recorded in the error log.

- Errors in network operation
- Errors in data transfers
- Error in the CPU unit

Each error is recorded as one record in an error log table. Up to 20 records can be saved. If more than 20 errors occur, the oldest errors will be deleted from the error log and the most recent error will be recorded.
The following information is recorded in the error log table.

- Main error code (see table later in this section)
- Detailed error code (see table later in this section)
- Time stamp (from the clock in the CPU unit)

Note During the initialization of the Ethernet Option Board, if an error occurs, the error log time stamp will record as 2000-00-00 00:00:00.

When an error is detected, the error codes and time stamp are recorded in the error log in RAM inside the Ethernet Option Board. Serious errors are also recorded in EEPROM. The maximum number of errors that can be saved to EEPROM is 20 for the CP1L and CP1H. The errors recorded in EEPROM will be saved even if the unit is restarted or power is turned OFF.

## Error Codes

| Error code | ERR <br> LED | Meaning | Detailed error code |  | Correction | EEPROM |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  |  | 1st byte | 2nd byte |  |  |
| 0002 | LIT | CPU Unit service monitoring error | Monitor tim |  | Check and correct the CPU Unit's operating environment. <br> Note <br> Recovery is possible for this error. When operation is restored, operations will return to normal. | Saved |
| 0015 | FLASH | CPU Unit fatal error |  |  | Eliminate the cause of the error in the CPU Unit. | --- |
| 010E | --- | No routing table entry (send failed) | Commands <br> Bit 15: OFF <br> Bits 08 to 14: SNA <br> Bits 00 to 07: SA1 <br> Responses <br> Bit 15: ON <br> Bits 08 to 14: DNA <br> Bits 00 to 07: DA1 |  | Set the local node, remote node, and relay nodes in the routing tables. | --- |
| 010F | --- | Routing table error (send failed) |  |  | Create the routing tables correctly. | --- |
| 0110 | --- | Too many relay points (send failed) |  |  | Reconstruct the network or correct the routing tables so that commands are sent to within a 3-level network range. | --- |
| 0111 | --- | Command too long (send failed) |  |  | Check the command format and set the correct command data. | --- |
| 0112 | --- | Header error (send failed) |  |  | Check the command format and set the correct command data. | --- |
| 0117 | --- | Internal buffers full; packet discarded |  |  | Change the network so that traffic is not concentrated. | --- |
| 0120 | --- | Unexpected routing error |  |  | Check the routing tables. | --- |
| 0123 | --- | Internal send buffers full; packet discarded |  |  | Change the network so that traffic is not concentrated. | --- |
| 0125 | --- | Time out error |  |  | Resend the command. | --- |
| 021A | FLASH | Logic error in setting table | 00 | 02: Network parameters 03: Routing tables 04: Unit Setup 05: CPU Bus Unit Words (CIO/DM) | Recreate the data specified by the 2nd byte of the detailed error code. | Saved |


| Error code | ERR <br> LED | Meaning | Detailed error code |  | Correction | EEPROM |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  |  | 1st byte | 2nd byte |  |  |
| 03C0 | FLASH | FINS/TCP setting error | 01 to 02: <br> Connection No. | 01: <br> Automatically allocated <br> FINS node address duplication 02: <br> Destination <br> IP address error <br> 03: <br> Destination <br> port number error | Set the FINS/TCP settings correctly. | --- |
| 03C2 | --- | FINS/TCP packet discarded | 01 to 02: Connection No. | 03: <br> Reception error 04: <br> Transmission error | Resend the command. | --- |
|  |  |  |  |  | There is too much load (traffic) on the Ethernet Option Board. Correct the system so that traffic is not concentrated. | --- |
| 03C3 | --- | FINS/UDP Packet discarded | 00 | 01 to FE Node address | The automatic generation (static) method was used as the IP address conversion method, so remote IP address information in internal memory could not be changed. | --- |
| 03C8 | --- | Socket Error | Arbitrary |  | Resend the packet or the destination node is not in the network. | --- |
| 03D0 | FLASH | System setup sum value error |  |  | Reset the value of system setup area, Restart CPU Unit. | Saved |
| 0601 | LIT | Option Board error | Arbitrary |  | Restart the CPU Unit. If the problem persists, replace the Ethernet Option Board. | Saved |
| 0602 | LIT | Option Board memory error | 01: Read error 02: Write error | 06: Error log | Restart the CPU Unit. If the problem persists, replace the Ethernet Option Board. | Saved (except error log) |

## 9-11-2 Trouble-shooting with Indicators and Error Code Display

| ERR <br> Indicator | Error | Probably Cause | Error code <br> (hex) | Correction |
| :--- | :--- | :--- | :--- | :--- |

## 9-11-3 Error Status

The status of errors that occur at the Ethernet Option Board is reflected in CIO relation area, Refer to 9-9-1 CIO Area Allocation, Error Status for details.

## 9-12 Sample Application

The following examples show how to connect online from a CX-Programmer on an Ethernet network to a PLC on the Ethernet network.

Note Please use CX-Programmer version 8.1 or higher (CX-ONE version 3.1 or higher).

- System Configuration Example 1: No Routing

In this example, an online connection is made by FINS/UDP to a PLC on an Ethernet network (PLC1 in the diagram below) from a CX-Programmer / CXIntegrator connected to the Ethernet network.


Ethernet (Network address: None)
FINS/UDP method
Web Browser Setting

| Item | Setting |
| :--- | :--- |
| Broadcast | All ones (4.3BSD) |
| FINS/UDP port | Default (9600) |
| IP address | 192.168 .250 .2 |
| Subnet mask | 255.255 .255 .0 |
| FINS Node Address | 2 |
| IP address conversion | Automatic generation method (dynamic) |
| Baud rate | Automatic Detaction |
| IP router tabel | None |




CX-Programmer's Change PLC Dialog Box

| Item |  | Setting |
| :--- | :--- | :--- |
| PLC name | PLC1 |  |
| Network classification | Ethernet |  |
| Network Tab | FINS transmission source address | 0 |
|  | FINS destination | Network number |
|  |  | 0 |
|  | Node address | 2 |
|  | Frame length | $1,004 b y t e s$ |
|  | Response monitor time | 5 seconds |
|  | Workstation node address | 1 |
|  | Automatic generation method | Not selected |
|  | Ethernet Option Board IP address | 192.168 .250 .2 |
|  | Port number | 9600 |

Note Limited by the CP1W-CIF41 inner bus protocol (Toolbus), the frame length and response monitor time are different from the existing Ethernet unit.

Inputs to the CX-Programmer's Setup Window

## Change PLC Settings



## Network Settings (Network Tab)



Network Settings (Driver Tab)


## ■ System Configuration Example 2: Using Routing Tables

In this example, an online connection is made via the Ethernet to a PLC on a Controller Link network (PLC3 in the diagram below) from a CX-Programmer / CX-Integrator connected to the Ethernet network.


## Web Browser Setting

Same as for System Configuration Example 1.

CX-Programmer's Change PLC Dialog Box

| Settings for target PLC (PLC1)'s Change PLC Dialog Box |  | Setting |  |  |
| :--- | :--- | :--- | :---: | :---: |
| PLC name |  |  |  | PLC2 |
| Network classification | PINS transmission source address | 2 |  |  |
| Network Tab | FIC1 |  |  |  |
|  | FINS destination | Network number |  |  |
|  |  | 1 |  |  |
|  | Node address | 1 |  |  |
|  | Frame length | 542 bytes (default) |  |  |
|  | Response monitor time | 5 seconds |  |  |

## Routing Table Settings and Transfer to Each PLC

Set the routing tables with CX-Integrator, and transfer them.
1,2,3... 1. Using CX-Integrator, connect online, and select Routing table - Settings. Then create FINS local routing tables (a local network table and a relay network table).
Example: PLC 1 Routing Table Settings

- Local Network Table

| Unit number | Local network number |
| :--- | :--- |
| 0 | 1 |
| 1 | 2 |

- Relay Network Table (None)

Example: PLC 2 and PLC 3 Routing Table Settings

- Local Network Table

| Option port No. | I/O capacity | Unit address | Local network <br> number |
| :--- | :--- | :--- | :--- |
| Option port 1 | $14 / 20$ | $252(0 x F C$ hex) | 1 |
|  | $30 / 40 / 60$ | $253(0 x F D$ hex) | 1 |
| Option port 2 | $30 / 40 / 60$ | $252(0 x F C$ hex) | 1 |

- Relay Network Table

In order to relay from PLC2/3 to the final network number 2, it is necessary to relay via node address 1 on relay network number 2.

| Final network number | Relay network number | Relay node address |
| :--- | :--- | :--- |
| 2 | 1 | 2 |

2. Save the routing table file (File - Save local routing table file).
3. Select New from the Project Menu, and save with a file name. Then select Add Device from the Project Menu. For each PLC, register a PLC with a direct serial connection (node address: 0), and select it.
4. With CX-Integrator, select Open from the PLC Menu.
5. Select Routing table - Setup, read the saved file, and select Options Transfer to PLC. Click Yes to transfer the routing tables to the connected PLCs.

## SECTION 10 <br> Program Transfer, Trial Operation, and Debugging

This section describes the processes used to transfer the program to the CPU Unit and the functions that can be used to test and debug the program.
10-1 Program Transfer. ..... 598
10-2 Trial Operation and Debugging ..... 598
10-2-1 Forced Set/Reset ..... 598
10-2-2 Differential Monitoring. ..... 599
10-2-3 Online Editing. ..... 600
10-2-4 Tracing Data ..... 602

## 10-1 Program Transfer

The CX-Programmer is used to transfer the programs, PLC Setup, I/O memory data, and I/O comments to the CPU Unit with the CPU Unit in PROGRAM mode. The following procedure is used.

1,2,3... 1. Select PLC - Transfer - To PLC. The Download Options Dialog Box will be displayed.
2. Specify the items to transfer.
3. Click the OK Button.

Note The program data on a Memory Cassette can be automatic transferred when the power is turned ON.

## 10-2 Trial Operation and Debugging

## 10-2-1 Forced Set/Reset

The CX-Programmer can force-set (ON) or reset (OFF) specified bits in the CIO Area, Auxiliary Area, and HR Area, as well as timer/counter Completion Flags. Forced status will take priority over status output from the program or I/O refreshing. This status cannot be overwritten by instructions, and will be stored regardless of the status of the program or external inputs until it is cleared from the CX-Programmer.
Force-set/reset operations are used to force input and output during a trial operation or to force certain conditions during debugging.
Force-set/reset operations can be executed in either MONITOR or PROGRAM modes, but not in RUN mode.

Note Turn ON the Forced Status Hold Bit (A500.13) and the IOM Hold Bit (A500.12) at the same time to retain the status of bits that have been force-set or reset when switching the operating mode.
Turn ON the Forced Status Hold Bit (A500.13) and the IOM Hold Bit (A500.12), and set the Forced Status Hold Bit at Startup parameter in the PLC Setup to retain the status of the Forced Status Hold Bit hold to retain the status of bits that have been force-set or reset when turning OFF the power.


The following areas can be force-set and reset: CIO Area, Work Area, Timer Completion Flags, HR Area, Counter Completion Flags.

## CX-Programmer Operation

- Selecting bits for forced setting/resetting
- Selecting forced set or forced reset status
- Clearing forced status (also clearing all forced status at the same time)


## 10-2-2 Differential Monitoring

When the CPU Unit detects that a bit set by the CX-Programmer has changed from OFF to ON or from ON to OFF, the results are indicated in the Differentiate Monitor Completed Flag (A508.09). The Flag will turn ON when conditions set for the differential monitor have been met. The CX-Programmer can monitor and display these results on screen.


## CX-Programmer Operation

1,2,3... 1. Right-click the bit for differential monitoring.
2. Click Differential Monitorfrom the PLC Menu. The Differential Monitor Dialog Box will be displayed.
3. Click Rising or Falling.
4. Click the Start Button. The buzzer will sound when the specified change is detected and the count will be incremented.
5. Click the Stop Button. Differential monitoring will stop.

Related Auxiliary Bits/Words

| Name | Address | Description |
| :--- | :--- | :--- |
| Differentiate Monitor <br> Completed Flag | A508.09 | Turns ON when the differential monitoring condition has been met dur- <br> ing differential monitoring. <br> Note: The flag will be cleared when differential monitoring is started. |

## 10-2-3 Online Editing

The Online Editing function is used to add to or change part of a program in a CPU Unit directly from the CX-Programmer when the CPU Unit is in MONITOR or PROGRAM mode. This function is designed for minor program changes without stopping the CPU Unit.
Online editing is possible simultaneously from more than one computer running the CX-Programmer as long as different tasks are edited.

Online Editing


The cycle time will be increased by from one to several cycle times if the program in the CPU Unit is edited online in MONITOR mode. The cycle time will also be increased to back up data in the flash memory after online editing. The BKUP indicator will be lit during this period and the progress of the backup will be displayed on the CX-Programmer. The increases per cycle are listed in the following table.

| CPU Unit | Increase in cycle time |  |
| :--- | :--- | :--- |
|  | Online editing | Backup to flash memory |
| CP1L CPU Units | 16 ms max. | $4 \%$ of cycle time |

There is a limit to the number of edits that can be made consecutively. The actual number depends on the type of editing that is performed, but 40 edits should be used as a guideline. A message will be displayed on the CX-Programmer if the limit is exceeded, and further editing will not be possible until the CPU Unit has completed backing up the data.
The length of time that the cycle time is extended due to online editing is almost unaffected by the size of the task program being edited.

## Precautions

The cycle time will be longer than normal when a program is overwritten using Online Editing in MONITOR mode, so make sure that the amount of time that it is extended will not exceed the cycle monitoring time set in the PLC Setup. If it does exceed the monitoring time, then a Cycle Time Over error will occur, and the CPU Unit will stop. Restart the CPU Unit by selecting PROGRAM mode first before changing to RUN or MONITOR mode.

Note If the task being edited online contains a block program, then previous execution information, such as Standby (WAIT) or Pause status, will be cleared by online editing, and the next execution will be from the beginning.
Online Editing from the CX-Programmer
1,2,3... 1. Display the program section that will be edited.
2. Select the instructions to be edited.
3. Select Program - Online Edit - Begin.
4. Edit the instructions.
5. Select Program - Online Edit - Send Changes The instructions will be check and, if there are no errors, they will be transferred to the CPU Unit. The instructions in the CPU Unit will be overwritten and cycle time will be increased at this time.

Caution Proceed with Online Editing only after verifying that the extended cycle time will not adversely affect operation. Input signals may not be read if the cycle time is too long.

## Temporarily Disabling Online Editing

It is possible to disable online editing for specific cycles to ensure response characteristics for machine control in those cycles. Online editing from the CX-Programmer will be disabled for those cycles and any requests for online editing received during those cycles will be held online editing is enables.
Online editing is disabled by setting the Online Editing Disable Bit Validator (A527.00 to A527.07) to 5A and then turning ON the Online Editing Disable Bit (A527.09). When these settings have been made and a request for online editing is received, online editing will be put on standby and the Online Editing Wait Flag (A201.10) will be turned ON.
When the Online Editing Disable Bit (A527.09) is turned OFF, online editing will be performed, the Online Editing Processing Flag (A201.11) will turn ON, and the Online Editing Wait Flag (A201.10) will turn OFF. When online editing has been completed, the Online Editing Processing Flag (A201.11) will turn OFF.
Online editing can also be temporarily disabled by turning ON the Online Editing Disable Bit (A527.09) while online editing is being performed. Here too, the Online Editing Wait Flag (A201.10) will turn ON.
If a second request for online editing is received while the first request is on standby, the second request will not be recorded and an error will occur.
Online editing can also be disabled to prevent accidental online editing. As described above, disable online editing by setting the Online Editing Disable Bit Validator (A527.00 to A527.07) to 5A and turning ON the Online Editing Disable Bit (A527.09).

## Enabling Online Editing from the CX-Programmer

When online editing cannot be enabled from the program, it can be enabled from the CX-Programmer. If operations continue with online editing in standby status, CX-Programmer may go offline. If this occurs, reconnect the computer to the CPU Unit and turn OFF the Online Edit Disable Bit (A527.09).

Note When using automatic transfer from a Memory Cassette at startup, be sure to transfer the data to the Memory Cassette if any changes are made using online editing. If power is turned OFF before the changes are transferred, the data from before online editing was performed will be read from the Memory Cassette.

Related Auxiliary Bits/Words

| Name | Address |  |
| :--- | :--- | :--- |
| Online Edit Disable Bit Validator | A527.00 to | Description |
|  | A527.07 |  |$\quad$| Enables using the Online Edit Disable Bit (A527.09). |
| :--- |
| Not 5A: Online Edit Disable Bit disabled. |
| 5A: Online Edit Disable Bit enabled. |


| Name | Address | Description |
| :---: | :---: | :--- |
| Online Editing Wait Flag | A201.10 | ON while an online editing process is on standby because online editing <br> is disabled. |
| Online Editing Processing Flag | A201.11 | ON while an online editing process is being executed. |

## 10-2-4 Tracing Data

The Data Trace function samples specified I/O memory data using any one of the following timing methods. It stores the sampled data in Trace Memory, where they can be read and checked later from the CX-Programmer.

- Specified sampling time (10 to $2,550 \mathrm{~ms}$ in $10-\mathrm{ms}$ units)
- One sample per cycle
- When the TRACE MEMORY SAMPLING instruction (TRSM(045)) is executed
Up to 31 bits and 6 words in I/O memory can be specified for sampling.


## Basic Procedure

1,2,3... 1. Sampling will start when the parameters have been set from the CX-Programmer and the command to start tracing has been executed.
2. Sampled data (after step 1 above) will be traced when the trace trigger condition is met, and the data just after the delay (see note 1) will be stored in Trace Memory.
3. Memory data will be sampled until the Trace Memory is full, and then the trace will be ended.

Note Delay value: Specifies how many sampling periods to offset the sampling in Trace Memory from when the trace condition is met. The setting ranges are shown in the following table.

| No. of words <br> sampled | Setting range |
| :--- | :--- |
| 0 | -1999 to 2000 |
| 1 | -1332 to 1333 |
| 2 | -999 to 1000 |
| 3 | -799 to 800 |
| 4 | -665 to 666 |
| 5 | -570 to 571 |
| 6 | -499 to 500 |

Positive delay: Store data delayed by the set delay.
Negative delay: Store previous data according go to the set delay.
Example: Sampling at 10 ms with a -30 ms delay time yields $-30 \times 10=$ 300 ms , so data 300 ms before the trigger will be stored.

Note Use the CX-Programmer to turn ON the Sampling Start Bit (A508.15). Never turn ON this bit from the user program.


The following traces can be executed.

Scheduled Data Trace

One-cycle Data Trace

Data Trace via TRSM(045) A sample will be taken once each time the TRACE MEMORY SAMPLING instruction (TRSM(045)) instruction is executed. When more than one $\operatorname{TRSM}(045)$ instruction is used in the program, a sample will be taken each time TRSM(045) is executed after the trace trigger condition has been met until trace memory is full.

## Data Trace Procedure

Use the following procedure to execute tracing.
1,2,3... 1. Use the CX-Programmer to set trace parameters (select PLC - Data Trace and then select Operation - Configure):
Addresses of the sampled words/bits, sampling period, delay time, and trigger conditions.
2. Use the CX-Programmer to start sampling or turn ON the Sampling Start Bit (A508.15).
3. Put the trace trigger condition into effect.
4. End tracing.
5. Use CX-Programmer to read the trace data.
a) Select Data Trace from the PLC Menu.
b) Select Select from the Operation Menu.
c) Select Execute from the Operation Menu.
d) Select Read from the Operation Menu.

Related Auxiliary Bits/Words

| Name | Address | Description |
| :--- | :--- | :--- |
| Sampling Start Bit | A508.15 | Use the CX-Programmer to turn ON this bit to start sampling. This bit must be <br> turned ON from the CX-Programmer. Do not turn this bit ON and OFF from the <br> user program. <br> Note: The bit will be turned OFF when the Data Trace has been completed. |
| Trace Start Bit | A508.14 | When this bit is turned ON, the trace trigger will be monitored and sampled data <br> will be stored in Trace Memory when the trigger condition is met. The following <br> traces are enabled with this bit. <br> 1) Scheduled trace (trace at fixed intervals of 10 to 2,550 ms) <br> 2) TRSM(045) instruction trace (trace when the TRSM(045) is executed) <br> 3) One-cycle trace (trace at the end of execution of all cyclic tasks) |
| Trace Trigger Monitor Flag | A508.11 | This flag turns ON when the trace trigger condition has been met after the Trace <br> Start Bit has turned ON. This flag will turn OFF when the sampling is started. |
| Trace Busy Flag | A508.13 | This flag turns ON when sampling is started and turns OFF when the trace has <br> been completed. |
| Trace Completed Flag | A508.12 | This flag turns ON when Trace Memory becomes full after the trace trigger con- <br> dition has been met during a trace operation and turns OFF when the next sam- <br> pling operation is started. |

## SECTION 11 Troubleshooting

This section provides information on hardware and software errors that occur during CP1L operation.
11-1 Error Classification and Confirmation ..... 606
11-2 Troubleshooting ..... 608
11-2-1 Error Processing Flowchart ..... 608
11-2-2 No Operation When Power Is Supplied ..... 608
11-2-3 Fatal Errors ..... 609
11-2-4 CPU Errors ..... 613
11-2-5 Non-fatal Errors ..... 614
11-2-6 Other Errors ..... 616
11-3 Error Log ..... 617
11-4 Troubleshooting Unit Errors ..... 618

## 11-1 Error Classification and Confirmation

## Error Categories

Errors in CP1L CPU Units can be broadly divided into the following four categories.

| Category | Comments |
| :--- | :--- |
| CPU Error | A WDT (watchdog timer) error is generated in the CPU Unit, the <br> CPU Unit will malfunction, and operation will stop. |
| CPU Standby | The CPU will go on standby because conditions for stating oper- <br> ation have not yet been met. |
| Fatal Error | Operation cannot continue. Operation will stop due to a serious <br> problem. |
| Non-fatal Error | A minor problem has occurred. Operation will continue |

## Confirming Errors

## CPU Unit Indicators



There are three sources of information on errors that have occurred.

- CPU Unit indicators
- Auxiliary Area

These indicators show the operating status of the CPU Unit.

| POWER (green) | Lit | Power is ON. |
| :---: | :---: | :---: |
|  | Not lit | Power is OFF. |
| RUN (green) | Lit | The CPU Unit is executing a program in either RUN or MONITOR mode. |
|  | Not lit | Operation is stopped in PROGRAM mode or due to a fatal error. |
| ERR/ALM (red) | Lit | A fatal error or CPU error (WDT error) has occurred. operation will stop and all outputs will be turned OFF. |
|  | Flashing | A non-fatal error has occurred. Operation will continue. |
|  | Not lit | Operation is normal. |
| INH (yellow) | Lit | The Output OFF Bit (A500.15) was turned ON. All outputs will be turned OFF. |
|  | Not lit | Operation is normal. |
| PRPHL (yellow) | Flashing | Communications (either sending or receiving) are in progress through the peripheral port. |
|  | Not lit | Other than the above. |
| BKUP (yellow) (See note.) | Lit | - The user program, parameters, or DM Area data is being written to or accessed in the built-in flash memory (backup memory). <br> - The user program, parameters, DM Area data, or DM initial values are being written to or accessed in a Memory Cassette. <br> - The BKUP indicator also lights while the user program is being restored when the power supply is turned ON. |
|  | Not lit | Other than the above. |

Note Do not turn OFF the CPU Unit power supply when this indicator is lit.

## CPU Unit Indicators and Error Meanings in RUN or MONITOR Mode

| Indicator | CPU <br> error | CPU <br> standby | Fatal <br> error | Non- <br> fatal <br> error | Peripheral port <br> communica- <br> tions error | Output <br> OFF Bit <br> turned ON |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| POWER | Lit | Lit | Lit | Lit | Lit | Lit |
| RUN | Not lit | Not lit | Not lit | Lit | Lit | Lit |
| ERR/ALM | Lit | Not lit | Lit | Flashing | --- | --- |
| INH | Not lit | --- | --- | --- | --- | Lit |
| PRPHL | --- | --- | --- | --- | Not lit | --- |
| BKUP | --- | --- | --- | --- | --- | --- |

## Auxiliary Area

## - Error Code Storage Word

The error code is stored in A400 when an error occurs. If two or more errors occur at the same time, the most serious error will be stored.

## - Error Flags

Flags that indicate the type of error are allocated in the Auxiliary Area.

## - Error Information

This area indicates specific information on the meaning of error flags and provides information on error location and error details.

## - Fatal Errors

| Error | Error code (A400) | Error flag | Error information |  |
| :---: | :---: | :---: | :---: | :---: |
|  |  |  | Meaning | Address |
| Memory error | 80F1 | A401.15 | Memory error location | A403 |
| I/O bus error | 80C0 to 80C7, 80CA, 80CE, 80CF | A401.14 | I/O bus error details | A404 |
| Too many I/O error | 80E1 | A401.11 | Details for too many l/O error | A407 |
| Program error | 80F0 | A401.09 | Program error details | A294 to A299 |
| Cycle time too long error | 809F | A401.08 | --- | --- |
| FALS instruction executed | C101 to C2FF | A401.06 | --- | --- |

## - Non-fatal Errors

| Error | Error code <br> (A400) | Error flag | Error information |  |
| :--- | :--- | :--- | :--- | :--- |
|  |  | Meaning | Address |  |
| FAL instruction <br> executed | 4101 to 42FF | A402.15 | Executed FAL <br> number | A360 to A391 |
| Flash memory <br> error | $00 F 1$ | A315.15 | --- | --- |
| Interrupt task error | 008B | A402.13 | Interrupt task <br> error unit num- <br> ber | A426 |
| PLC Setup error | 009B | A402.10 |  | A406 |
| Option Board error | 00D1, 00D2 | A315.13 | Error Option <br> Board Flags | A424 |
| Battery error | 00F7 | A402.04 | --- | --- |

## 11-2 Troubleshooting

Use the following procedure to check error details and remove the cause of the error if the CPU Unit does not operate when the power supply is ON, operation suddenly stops and the error indicator (ERR/ALM indicator) lights, or if the error indicator (ERR/ALM indicator) flashes during operating.

## 11-2-1 Error Processing Flowchart

Confirm the error category by referring to the status of the CPU Unit indicators, investigate the cause for the error in the error tables, and take corrective actions.


## 11-2-2 No Operation When Power Is Supplied

First confirm that the POWER indicator (green) is lit.

## POWER Indicator Not Lit

The power supply may not match the Unit rating, wiring may not be correct, or the Unit may be faulty.

1,2,3... 1. Confirm the Unit rating (i.e., is it 24 VDC or 100 to 240 VAC?) and see if the supply power matches the rating.
2. Check the wiring to see if it is correct and that nothing is disconnected.
3. Check the voltage at the power supply terminals. If the voltage is normal and the POWER indicator is lit, the Unit may be faulty. In that case, replace the Unit.

## POWER Indicator Turns OFF and ON

There may be fluctuations in the power supply voltage, disconnected wiring, or poor contacts. Check the power supply system and wiring.

## POWER Indicator Lit but No Operation

Check the RUN indicator if the POWER indicator is lit but the CPU Unit does not operate. The CPU Unit may be on standby if the RUN indicator is not lit.

## ■ CPU Standby

Detection of Special I/O Units and CPU Bus Units has not been completed.

- If a CPU Bus Unit has not started normally, check the Unit Setup.
- If a Special I/O Unit is not detected, replace the Special I/O Unit.


## 11-2-3 Fatal Errors

## ■ CPU Unit Indicators



| POWER | Lit |
| :--- | :--- |
| RUN | Not lit |
| ERR/ALM | Lit |
| INH | --- |
| PRPHL | --- |
| BKUP | --- |

There may be a CPU error or a fatal error if operation stops (i.e., the RUN indicator turns OFF) and the ERR/ALM indicator lights.
Data on fatal errors is displayed on the Error Tab Page of the CX-Programmer's PLC Error Window.
Take corrective actions after checking error details based on the CX-Programmer display message together with the Auxiliary Area Error Flags and error information.

Note 1. Errors are listed in order with the most serious errors first.
2. If two or more errors occur at the same time, the most serious error code will be stored in A400.
3. I/O memory will be cleared if a fatal error occurs (except those created with FALS instructions).
4. I/O memory will be held when the I/O Memory Hold Bit is ON, but outputs will be turned OFF.

## Memory Errors

| Probable cause | Possible remedy |
| :--- | :--- |
| Automatic transfer from the Memory Cassette <br> at startup failed because the required data is <br> not on the Memory Cassette. | Store the required data on the Memory Cas- <br> sette. |
| An error has occurred in memory. One or <br> more bits in A403 will turn ON to indicate <br> where the error has occurred. See below for <br> details. <br> - Memory Error Location | See below. |
| A403.00 is ON A checksum error <br> has occurred in the <br> user program. <br> The power supply <br> was turned OFF <br> when backing up the <br> user program to <br> flash memory. <br> A403.04 is ON A checksum error <br> has occurred in the <br> PLC Setup. | Transfer the user program again. |
| A403.07 is ON | A checksum error <br> has occurred in the <br> routing tables. | | Transfer the routing tables again. |
| :--- |
| A403.09 is ON |
| The contents of a <br> Memory Cassette <br> could not be nor- <br> mally read to the <br> CPU Unit when <br> power was turned <br> ON. |
| Check to see if the files required for automatic <br> transfer at startup are present on the Memory <br> Cassette. |
| A403.10 is ON |
| There is a problem <br> with flash memory. |
| A hardware error has occurred in the CPU Unit. <br> Replace the CPU Unit. |

## Reference Information

| Error flag | Memory Error Flag, A401.15 |
| :--- | :--- |
| Error code (A400) | $80 F 1$ |
| Error information | Memory Error Location, A403 |

An I/O bus error occurs in data transfer between the CPU Units and Units connected to the I/O bus. Cycle the power supply. If operation is not restored when the power supply has been cycled, turn OFF the power supply and check that connections are proper and that there is no damage.

| Probable cause | Possible remedy |
| :--- | :--- |
| An error occurred in data transfer between the CPU Unit | Try cycling the power supply. <br> and an Expansion Unit or Expansion I/O Unit. <br> Note OAOA hex will be stored in A404. |
| If the problem persists, turn OFF the power sup- <br> ply and check the Connecting Cables between <br> the Units to see if they are connected properly. <br> Check the Unit connections to be sure they are <br> ok (e.g., that there is no damage). <br> After correcting the problem, turn ON the power <br> to the Units again. |  |

Reference Information

| Error flag | I/O Bus Error Flag, A401.14 |
| :--- | :--- |
| Error code (A400) | 80C0, 80CA, 80CE, 80CF |
| Error information | I/O bus error details, A404 (OA0A) |

## Too Many I/O Points

The number of CP1W/CPM1A-series Expansion Units and Expansion I/O Units connected exceeds the restriction for the number of Units or words for the system configuration. Turn OFF the power supply and reconfigure the system within the restrictions.

| Probable cause | Possible remedy |
| :--- | :--- |
| The number of CP1W/CPM1A-series Expansion Units | • Connect a maximum of three Expansion Units and |
| and Expansion I/O Units exceeds the restriction. | Expansion I/O Units to the CP1L-M60D $\square-\square$, CP1L- |
|  | M40D $\square-\square$ or CP1L-M30D $\square-\square$. |

- Reference Information

| Error flag | Too Many I/O Points Flag, A401.11 |
| :--- | :--- |
| Error code (A400) | 80 E 1 |
| Error information | Too Many I/O Points Details, A407 |

## Program Error

A program error indicates a problem with the user program. Refer to the error information, check the program, and correct the mistakes. Clear the error once the problem has been corrected.

| Probable cause | Possible remedy |
| :--- | :--- |
| Instruction Processing Error <br> If the PLC Setup has been set to stop operation for an <br> instruction error, the Error Flag will be turned ON when an <br> instruction cannot be executed due to a problem in the <br> operand data. | Refer to A298 and A299 (instruction program address when <br> the program fails), check the specifications for the relevant <br> instruction, and set the correct operand data. <br> Alternatively, set the PLC Setup to not stop operation for an <br> instruction error. |
| Indirect DM Addressing BCD Error <br> If the PLC Setup has been set to stop operation for an indi- |  |
| rect DM BCD error, the Access Error Flag will turn ON when <br> the content of an indirectly addressed DM operand is not <br> BCD although BCD mode has been selected. | Refer to A298 and A299 (instruction program address when <br> the program fails), and correctly set the content for the indi- <br> rectly addressed DM operand (BCD mode) to BCD or <br> lhange the specified destination. <br> Alternatively, change the indirect addressing to binary <br> mode or set the PLC Setup to not stop operation for an indi- <br> rect DM addressing BCD error. |


| Probable cause | Possible remedy |
| :---: | :---: |
| Illegal Area Access Error <br> If the PLC Setup has been set to stop operation for an illegal access error, the Access Error Flag will turn ON when an illegal access error has occurred. <br> The following operations are considered illegal access: <br> - Reading/writing the parameter area <br> - Writing to an area without memory installed <br> - Writing to a write-protected area <br> - Indirect DM addressing BCD error | Refer to A298 and A299 (instruction program address when the program fails) and take corrective actions so that illegal area access errors will not occur. Alternatively, set the PLC Setup so that PLC operation will not stop when an instruction error occurs. |
| No END Error <br> This error occurs when there is not an $\operatorname{END}(001)$ instruction in the program within a task. | Insert an END(001) instruction at the end of the program allocated to the task stored in A294 (task number when the program fails). |
| Task Error <br> This error is generated by any of the following conditions. <br> 1. There is not an executable cyclic task (active). <br> 2. There is no specified interrupt task when an interrupt is generated (input interrupt, high-speed counter interrupt, scheduled interrupt, or external interrupt). | 1. Check the properties of the executable cyclic task and set at least one task to start when operation starts. <br> 2. Create a task for the number stored in A294 (task number when the program fails). |
| Differentiation Overflow Error <br> Differentiation instructions were repeatedly inserted or deleted using the online editor and the system restriction was exceeded. | Change the operating mode to PROGRAM mode and then return to MONITOR mode. |
| Illegal Instruction Error <br> Execution of an unexecutable instruction was attempted. <br> For a CPU Unit with 10, 14 or 20 I/O Points, the instruction operand will be given in D10000 to D31999. | Check the program, correct the problem, and transfer the program to the CPU Unit again. |
| UM Overflow Error <br> An attempt was made to execute a program that exceeds the user program capacity. | Transfer the program again using the CX-Programmer. |

- Reference Information

| Error flag | Program Error Flag, A401.09 |
| :--- | :--- |
| Error code (A400) | 80F0 |
| Error information | Program error details, A294 to A299 |

## Cycle Time Too Long

| Probable cause | Possible remedy |
| :---: | :---: |
| This error occurs when the cycle time PV exceeds the maximum cycle time set in the PLC Setup. | Review the program to decrease the cycle time or change the maximum cycle time set in the PLC Setup. <br> Refer to the Maximum Interrupt Task Processing Time (A440) and study the maximum cycle time. <br> The cycle time can be decreased using the following methods. <br> - Separate instructions not being executed into different tasks. <br> - Consider using jump instructions for areas in the task that are not executed. <br> - Prohibit cycle refreshing with Special I/O Units that do not require exchange of cycle data. |

■ Reference Information

| Error flag | Cycle Time Too Long Error, A401.08 |
| :--- | :--- |
| Error code (A400) | 809 F |
| Error information | --- |

Errors Created with FALS Instructions

| Probable cause | Possible remedy |
| :--- | :--- |
| FALS instruction executed (FALS number 001) | C100 hex will be added to the FALS number (001 to 1FF hex) <br> and the result will be stored in A400 as the error code (C100 to <br> C2FF hex). <br> Check the conditions for executing FALS instructions and <br> remove any causes for the user-defined error. |

- Reference Information

| Error flag | FALS Error Flag, A401.06 |
| :--- | :--- |
| Error code (A400) | C101 to C2FF |
| Error information | --- |

## 11-2-4 CPU Errors

## ■ CPU Unit Indicators



| POWER | Lit |
| :--- | :--- |
| RUN | Not lit |
| ERR/ALM | Lit |
| INH | --- |
| BKUP | --- |
| PRPHL | --- |

A CPU error or fatal error may have occurred if the ERR/ALM indicator lights during operation (RUN mode or MONITOR mode), the RUN indicator turns OFF, and operation stops.

## CPU Errors

| Probable cause | Possible remedy |
| :--- | :--- |
| A WDT (watchdog) error occurred in the <br> CPU Unit. (This does not occur in normal <br> use.) | Cycle the power supply. The Unit may be <br> faulty. Consult your OMRON representa- <br> tive. |

■ Reference Information

| Error flag | None |
| :--- | :--- |
| Error code (A400) | None |
| Error information | None |

Note Just as when a CPU error occurs, the RUN indicator will turn OFF and the ERR/ALM indicator will light when a fatal error occurs. Connecting the CXProgrammer, however, is possible for fatal errors but not for CPU errors. If the CX-Programmer cannot be connected (online), a CPU error has probably occurred.

## 11-2-5 Non-fatal Errors

A non-fatal error has occurred if both the RUN indicator and the ERR/ALM indicator are lit during operation (i.e., in RUN or MONITOR mode).

## ■ CPU Unit Indicators



| POWER | Lit |
| :--- | :--- |
| RUN | Lit |
| ERR/ALM | Flashing |
| INH | --- |
| BKUP | --- |
| PRPHL | --- |

Information on the non-fatal error can be obtained from the Error Tab Page of the CX-Programmer's PLC Error Window. Take corrective actions after checking error details using the display messages and the Auxiliary Area Error Flags and error information.

- Errors are listed in the following table in order, with the most serious ones first.
- If two or more errors occur at the same time, the most serious error code will be stored in A400.


## Errors Created with for FAL Instructions

A FAL instruction was executed in the program to create a non-fatal error.

| Probable cause | Possible remedy |
| :--- | :--- |
| The executed FAL number 001 to 511 will |  |
| be stored in A360 to A391. The number 4 | Check the conditions for executing FAL <br> will be added to the front of 101 to 2FF <br> (which correspond to executed FAL num- |
| the user-defined error. |  |
| bers 001 to 511) and the result will be |  |
| stored in A400 as error code 4101 to |  |
| 42FF. |  |

- Reference Information

| Error flag | FAL Error Flag, A402.15 |
| :--- | :--- |
| Error code (A400) | 4101 to 42FF |
| Error information | None |

## Flash Memory Errors

| Probable cause | Possible remedy |
| :--- | :--- |
| A315.15 will turn ON when writing to the <br> internal flash memory fails. | Replace the CPU Unit when the internal <br> flash memory has been written to more <br> than 100,000 times. |

- Reference Information

| Error flag | Flash Memory Error Flag, A315.15 <br> Other non-fatal flags, A402.00 |
| :--- | :--- |
| Error code (A400) | None |
| Error information | None |


| Probable cause | Possible remedy |
| :--- | :--- |
| An interrupt task error occurs when the | Review the program to see whether detect- |
| Detect Interrupt task errors setting in the | ing interrupt task errors can be disabled or |
| PLC Setup is set to Detect and an attempt | avoided. |
| is made to refresh a Special I/O Unit from |  |
| an interrupt task with IORF(097) while the |  |
| Unit's I/O is being refreshed by cyclic |  |
| refreshing (duplicate refreshing). |  |

## ■ Reference Information

| Error flag | Interrupt Task Error Flag, A402.13 |
| :--- | :--- |
| Error code (A400) | 008B |
| Error information | Interrupt Task Error, A426 |

## PLC Setup Errors

| Probable cause | Possible remedy |
| :--- | :--- |
| A set value error occurred in the PLC | Correct the PLC Setup with correct val- <br> Setup. <br> The address of the error is stored in A406 <br> in 16-bit binary. |

■ Reference Information

| Error flag | PLC Setup Error Flag, A402.10 |
| :--- | :--- |
| Error code (A400) | 009B |
| Error information | PLC Setup error location, A406 |

## Option Board Errors

| Probable cause | Possible remedy |
| :--- | :--- |
| A315.13 will turn ON if the Option Board <br> is removed while the power is being sup- <br> plied. | Turn OFF the power supply and then <br> install the Option Board again. |

■ Reference Information

| Error flags | Option Board Error Flag, A315.13 <br> Other Non-fatal Error Flag, A402.00 |
| :--- | :--- |
| Error code (A400) | --- |
| Error information | --- |

## Battery Error

| Probable cause | Possible remedy |
| :--- | :--- |
| If the PLC Setup is set to detect battery <br> errors, this error will occur when there is <br> an error in the battery in the CPU Unit <br> (i.e., the voltage is low or a battery is not <br> mounted). | Check the battery connections. <br> When using battery-free operation, dis- <br> able connecting battery errors in the PLC <br> Setup. |

- Reference Information

| Error flag | Battery Error Flag, A402.04 |
| :--- | :--- |
| Error code (A400) | 00F7 |
| Error information | --- |

## 11-2-6 Other Errors

## Communications Errors

■ CPU Unit Indicators


| POWER | Lit |
| :--- | :--- |
| RUN | Lit |
| ERR/ALM | --- |
| INH | --- |
| PRPHL | --- |
| BKUP | Not lit |


| Probable cause | Possible remedy |
| :--- | :--- |
| An error has occurred in the communica- <br> tions between the peripheral port and con- <br> nected device. | Confirm that the peripheral port settings in <br> the PLC Setup are correct. <br> Check the USB cable and replace it if neces- <br> sary. |
| An error has occurred in the communica- <br> tions between the serial port and connected <br> device. | Confirm that the serial port $1 / 2$ settings in <br> the PLC Setup are correct. Check the cable <br> wiring. If a host computer is connected, <br> check the serial port settings and program in <br> the host computer. |

## 11-3 Error Log

Each time an error occurs, the CPU Unit stores error information in the Error Log Area of the Auxiliary Area (A100 to A199). The error information includes the error code (stored in A400), error contents, and time that the error occurred. Up to 20 records can be stored in the Error Log.

In addition to system-generated errors, the CPU Unit records user-defined errors, making it easier to track the operating status of the system.
When more than 20 errors occur, the oldest error data (stored in A100 to A104) is deleted, the 19 errors stored in A105 to A199 shift one record, and the newest record is stored in A195 to A199.
The number of records stored in the error log is stored in the Error Log Pointer (A300). The Error Log Pointer is not incremented after 20 records have been stored.


## 11-4 Troubleshooting Unit Errors

## CPU Unit

| Symptom | Cause | Remedy |
| :---: | :---: | :---: |
| POWER indicator is not lit. | PCB short-circuited or damaged. | Replace Unit. |
| RUN indicator is not lit. | (1) Error in program (fatal error) | Correct program. |
|  | (2) Power line is faulty. | Replace Unit. |
| RUN indicator on the CPU Unit is lit. | Internal circuitry in the Unit is faulty. |  |
| Bits do not operate past a certain point. |  |  |
| Error occurs in units of 8 or 16 points. |  |  |
| I/O bit turns ON. |  |  |
| All bits in one Unit do not turn ON. |  |  |

## Inputs

| Symptom | Cause | Remedy |
| :---: | :---: | :---: |
| Not all inputs turn ON or indicators are not lit. | (1) External power is not supplied for the input. | Supply power |
|  | (2) Supply voltage is low. | Adjust supply voltage to within rated range. |
|  | (3) Terminal block mounting screws are loose. | Tighten screws. |
|  | (4) Faulty contact of terminal block connector. | Replace terminal block connector. |
| Not all inputs turn ON even though the indicator is lit. | Input circuit is faulty. (There is a short at the load or something else that caused an overcurrent to flow.) | Replace Unit. |
| Not all inputs turn OFF. | Input circuit is faulty. | Replace Unit. |
| Specific bit does not turn ON. | (1) Input device is faulty. | Replace input devices. |
|  | (2) Input wiring disconnected. | Check input wiring |
|  | (3) Terminal block screws are loose. | Tighten screws |
|  | (4) Faulty terminal block connector contact. | Replace terminal block connector. |
|  | (5) Too short ON time of external input. | Adjust input device |
|  | (6) Faulty input circuit | Replace Unit. |
|  | (7) Input bit number is used for output instruction. | Correct program. |
| Specific bit does not turn OFF. | (1) Input circuit is faulty. | Replace Unit. |
|  | (2) Input bit number is used for output instruction. | Correct program. |
| Input irregularly turns ON/ OFF. | (1) External input voltage is low or unstable. | Adjust external input voltage to within rated range. |
|  | (2) Malfunction due to noise. | Take protective measures against noise, such as: <br> - Install surge suppressor. <br> - Install insulation transformer. Install shielded cables between the Input Unit and the loads. |
|  | (3) Terminal block screws are loose. | Tighten screws |
|  | (4) Faulty terminal block connector contact. | Replace terminal block connector. |
| Error occurs in units of 8 points or 16 points, i.e., for the same common. | (1) Common terminal screws are loose. | Tighten screws |
|  | (2) Faulty terminal block connector contact. | Replace terminal block connector. |
|  | (3) Faulty data bus | Replace Unit. |
|  | (4) Faulty CPU | Replace CPU Unit. |
| Input indicator is not lit in normal operation. | Faulty indicator or indicator circuit. | Replace Unit. |

## Outputs

| Symptom | Cause | Remedy |
| :---: | :---: | :---: |
| Not all outputs turn ON | (1) Load is not supplied with power. | Supply power |
|  | (2) Load voltage is low. | Adjust voltage to within rated range. |
|  | (3) Terminal block screws are loose. | Tighten screws |
|  | (4) Faulty terminal block connector contact. | Replace terminal block connector. |
|  | (5) An overcurrent (possibly caused by a short at the load) resulted in a blown fuse for the output or the Unit is faulty. | Replace fuse or Unit. |
|  | (6) Faulty I/O bus connector contact. | Replace Unit. |
|  | (7) Output circuit is faulty. | Replace Unit. |
|  | (8) If the INH indicator is lit, the Output OFF Bit (A500.15) is ON. | Turn A500.15 OFF. |
| Not all outputs turn OFF | Output circuit is faulty. | Replace Unit. |
| Output of a specific bit number does not turn ON or indicator is not lit | (1) Output ON time too short because of a mistake in programming. | Correct program to increase the time that the output is ON . |
|  | (2) Bit status controlled by multiple instructions. | Correct program so that each output bit is controlled by only one instruction. |
|  | (3) Faulty output circuit. | Replace Unit. |
| Output of a specific bit number does not turn ON (indicator lit). | (1) Faulty output device. | Replace output device. |
|  | (2) Break in output wiring. | Check output wiring. |
|  | (3) Loose terminal block screws. | Tighten screws. |
|  | (4) Faulty terminal block connector faulty. | Replace terminal block connector. |
|  | (5) Faulty output bit (relay output only). | Replace Unit. |
|  | (6) Faulty output circuit (relay output only). | Replace Unit. |
| Output of a specific bit number does not turn OFF (indicator is not lit). | (1) Faulty output bit. | Replace Unit. |
|  | (2) Bit does not turn OFF due to leakage current or residual voltage. | Replace external load or add dummy resistor. |
| Output of a specific bit number does not turn OFF (indicator lit). | (1) Bit status controlled by multiple instructions. | Correct program. |
|  | (2) Faulty output circuit. | Replace Unit. |
| Output irregularly turns ON/ OFF. | (1) Low or unstable load voltage. | Adjust load voltage to within rated range |
|  | (2) Bit status controlled by multiple instructions. | Correct program so that each output bit is controlled by only one instruction. |
|  | (3) Malfunction due to noise. | Protective measures against noise: <br> - Install surge suppressor. <br> - Install insulation transformer. <br> - Use shielded cables between the output terminal and the load. |
|  | (4) Terminal block screws are loose. | Tighten screws. |
|  | (5) Faulty terminal block connector contact. | Replace terminal block connector. |
| Error occurs in units of 8 points or 16 points, i.e., for the same common. | (1) Loose common terminal screw. | Tighten screws. |
|  | (2) Faulty terminal block connector contact. | Replace terminal block connector. |
|  | (3) An overcurrent (possibly caused by a short at the load) resulted in a blown fuse for the output or the Unit is faulty. | Replace fuse or Unit. |
|  | (4) Faulty data bus. | Replace Unit. |
|  | (5) Faulty CPU. | Replace CPU Unit. |
| Output indicator is not lit (operation is normal). | Faulty indicator. | Replace Unit. |

## Inspection and Maintenance

This section provides inspection and maintenance information.
12-1 Inspections ..... 622
12-1-1 Inspection Points ..... 622
12-1-2 Unit Replacement Precautions ..... 623
12-2 Replacing User-serviceable Parts ..... 624

## 12-1 Inspections

Daily or periodic inspections are required in order to maintain the PLC's functions in peak operating condition.

## 12-1-1 Inspection Points

Although the major components in CP-series PLCs have an extremely long life time, they can deteriorate under improper environmental conditions. Periodic inspections are thus required to ensure that the required conditions are being kept.
Inspection is recommended at least once every six months to a year, but more frequent inspections will be necessary in adverse environments.
Take immediate steps to correct the situation if any of the conditions in the following table are not met.

| No. | Item | Inspection | Criteria | Action |
| :---: | :---: | :---: | :---: | :---: |
| 1 | Source Power Supply | Check for voltage fluctuations at the power supply terminals. | The voltage must be within the allowable voltage fluctuation range. (See note.) | Use a voltage tester to check the power supply at the terminals. Take necessary steps to bring voltage fluctuations within limits. |
| 2 | I/O Power Supply | Check for voltage fluctuations at the I/O terminals. | Voltages must be within specifications for each Unit. | Use a voltage tester to check the power supply at the terminals. Take necessary steps to bring voltage fluctuations within limits. |
| 3 | Ambient environment | Check the ambient temperature (inside the control panel if the PLC is in a control panel). | 0 to $55^{\circ} \mathrm{C}$ | Use a thermometer to check the temperature and ensure that the ambient temperature remains within the allowed range of 0 to $55^{\circ} \mathrm{C}$. |
|  |  | Check the ambient humidity (inside the control panel if the PLC is in a control panel). | Relative humidity must be $10 \%$ to $90 \%$ with no condensation. | Use a hygrometer to check the humidity and ensure that the ambient humidity remains within the allowed range. |
|  |  | Check that the PLC is not in direct sunlight. | Not in direct sunlight | Protect the PLC if necessary. |
|  |  | Check for accumulation of dirt, dust, salt, metal filings, etc. | No accumulation | Clean and protect the PLC if necessary. |
|  |  | Check for water, oil, or chemical sprays hitting the PLC. | No spray on the PLC | Clean and protect the PLC if necessary. |
|  |  | Check for corrosive or flammable gases in the area of the PLC. | No corrosive or flammable gases | Check by smell or use a sensor. |
|  |  | Check the level of vibration or shock. | Vibration and shock must be within specifications. | Install cushioning or shock absorbing equipment if necessary. |
|  |  | Check for noise sources near the PLC. | No significant noise sources | Either separate the PLC and noise source or protect the PLC. |


| No. | Item | Inspection | Criteria | Action |
| :---: | :---: | :---: | :---: | :---: |
| 4 | Installation and wiring | Check that each Unit is connected securely and locked in place. | No looseness | Press the connectors together completely and lock them with the sliders. |
|  |  | Check that the Option Boards and cable connectors are fully inserted and locked. | No looseness | Correct any improperly installed connectors. |
|  |  | Check for loose screws in external wiring. | No looseness | Tighten loose screws with a Phillips screwdriver. |
|  |  | Check crimp connectors in external wiring. | Adequate spacing between connectors | Check visually and adjust if necessary. |
|  |  | Check for damaged external wiring cables. | No damage | Check visually and replace cables if necessary. |
| 5 | User-serviceable parts | Check whether the battery has reached its service life. CJ1W-BAT01 Battery | Service life expectancy is 5 years at $25^{\circ} \mathrm{C}$, less at higher temperatures. (From 0.75 to 5 years depending on model, power supply rate, and ambient temperature.) | Replace the battery when its service life has passed even if a battery error has not occurred. |

Note The following table shows the allowable voltage fluctuation ranges for source power supplies.

| CPU Unit | Supply voltage | Allowable voltage range |
| :--- | :--- | :--- |
| CP1L-M $\square \square \mathrm{D} \square$-A | 100 to 240 V AC | 85 to 264 V AC <br> $(+10 \% /-15 \%)$ |
| CP1L-L $\square \square \mathrm{D} \square$-A |  | 20.4 to 26.4 V DC <br> $(+10 \% /-15 \%)$ |
| CP1L-M $\square \square \mathrm{D} \square$-D | 24 V DC |  |
| CP1L-L $\square \square \mathrm{D} \square$-D |  |  |

## Tools Required for Inspections

## Required Tools

- Phillips screwdrivers
- Voltage tester or digital voltmeter
- Industrial alcohol and clean cotton cloth


## Tools Required Occasionally

- Synchroscope
- Oscilloscope with pen plotter
- Thermometer and hygrometer


## 12-1-2 Unit Replacement Precautions

Check the following when replacing any faulty Unit.

- Do not replace a Unit until the power is turned OFF.
- Check the new Unit to make sure that there are no errors.
- If a faulty Unit is being returned for repair, describe the problem in as much detail as possible, enclose this description with the Unit, and return the Unit to your OMRON representative.
- For poor contact, take a clean cotton cloth, soak the cloth in industrial alcohol, and carefully wipe the contacts clean. Be sure to remove any lint prior to remounting the Unit.

Note When replacing a CPU Unit, be sure that not only the user program but also all other data required for operation is transferred to or set in the new CPU Unit before starting operation, including DM Area and HR Area settings. If
data area and other data are not correct for the user program, unexpected accidents may occur.

## 12-2 Replacing User-serviceable Parts

The following parts should be replaced periodically as preventative maintenance. The procedures for replacing these parts are described later in this section.

- Battery (backup for the CPU Unit's internal clock and RAM)

The battery maintains the internal clock and the following data of the CPU Unit's RAM while the main power supply is OFF.

- Retained regions of I/O memory (such as the Holding Area and DM Area) If the battery is not installed or battery voltage drops too low, the internal clock will stop and the data in RAM will be lost when the main power supply goes OFF.

At $25^{\circ} \mathrm{C}$, the maximum service life for batteries is five years whether or not power is supplied to the CPU Unit while the battery is installed. The battery's lifetime will be shorter when it is used at higher temperatures.
The following table shows the approximate minimum lifetimes and typical lifetimes for the backup battery (total time with power not supplied).

| Model | Approx. <br> maximum lifetime | Approx. minimum <br> lifetime (See note.) | Typical lifetime <br> (See note.) |
| :--- | :--- | :--- | :--- |
| CP1L-M $\square \square D \square-D$ | 5 years | 13,000 hours <br> CP1L-L $\square \square D \square-D$ | 43,000 hours <br> (approx. 1.5 years) |

Note The minimum lifetime is the memory backup time at an ambient temperature of $55^{\circ} \mathrm{C}$. The typical lifetime is the memory backup time at an ambient temperature of $25^{\circ} \mathrm{C}$.


[^1]Low Battery Indications

## Replacement Battery

## Replacement Procedure

The ERR/ALM indicator on the front of the CPU Unit will flash when the battery is nearly discharged.


When the ERR/ALM indicator flashes, connect the CX-Programmer to the peripheral port and read the error messages. If a low battery message appears on the CX-Programmer (see note 1) and the Battery Error Flag (A402.04) is ON (see note 1), first check whether the battery is properly connected to the CPU Unit. If the battery is properly connected, replace the battery as soon as possible.
Once a low-battery error has been detected, it will take 5 days before the battery fails assuming that power has been supplied at lease once a day (see note 2). Battery failure and the resulting loss of data in RAM can be delayed by ensuring that the CPU Unit power is not turned OFF until the battery has been replaced.

Note 1. The PLC Setup must be set to detect a low-battery error (Detect Low Battery). If this setting has not been made, the BATT LOW error message will not appear on the CX-Programmer and the Battery Error Flag (A402.04) will not go ON when the battery fails.
2. The battery will discharge faster at higher temperatures, e.g., 4 days at $40^{\circ} \mathrm{C}$ and 2 days at $55^{\circ} \mathrm{C}$.

Use the CJ1W-BAT01 Battery Set. Be sure to install a replacement battery within two years of the production date shown on the battery's label.

Production Date
OmROn
CJ1W-BAT01
Manufactured in July 2005. 05-07

Use the following procedure to replace the battery when the previous battery has become completely discharged. You must complete this procedure within five minutes after turning OFF the power to the CPU Unit to ensure memory backup.

Note 1. We recommend replacing the battery with the power OFF to prevent the CPU Unit's sensitive internal components from being damaged by static electricity. The battery can be replaced without turning OFF the power supply. To do so, always touch a grounded piece of metal to discharge static electricity from your body before starting the procedure.
2. After replacing the battery, connect the CX-Programmer and clear the battery error.

## Procedure

1,2,3... 1. Turn OFF the power to the CPU Unit.
or If the CPU Unit has not been ON, turn it ON for at least five minutes and then turn it OFF.
Note If power is not turned ON for at least five minutes before replacing the battery, the capacitor that backs up memory when the battery is removed will not be fully charged and memory may be lost before the new battery is inserted.
2. Open the compartment on the CPU Unit and carefully draw out the battery.
3. Remove the battery connector.
4. Connect the new battery, place it into the compartment, and close the cover.

CPU Units with 10, 14 or 20 I/O Points


CPU Units with 30,40 or $601 / O$ Points


WARNING Never short-circuit the battery terminals; never charge the battery; never disassemble the battery; and never heat or incinerate the battery. Doing any of these may cause the battery to leak, burn, or rupturing resulting in injury, fire, and possible loss of life or property. Also, never use a battery that has been dropped on the floor or otherwise subject to shock. It may leak.

Caution You must complete this procedure within five minutes after turning OFF the power to the CPU Unit to ensure memory backup. If the procedure is not completed within 5 minutes, data may be lost.

Caution UL standards require that batteries be replaced by experienced technicians. Always place an experienced technician in charge or battery replacement.
\. Caution Turn ON the power after replacing the battery for a CPU Unit that has been unused for a long time. Leaving the CPU Unit unused again without turning ON the power even once after the battery is replaced may result in a shorter battery life.

Note The battery error will automatically be cleared when a new battery is inserted.

## Appendix A

## Standard Models

## CPU Units

| Name and appearance | Model | Specifications |  |  | Remarks |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Power supply | Outputs | Inputs |  |
| CPU Units with 10 I/ O Points | CP1L-L10DR-A | $\begin{aligned} & 100 \text { to } \\ & 240 \text { VAC } \end{aligned}$ | 4 relay outputs | 24 VDC <br> 6 inputs | Memory capacity: 5 Ksteps <br> High-speed counters: <br> $100 \mathrm{kHz}, 4$ counters <br> Pulse outputs: <br> 2 axes at 100 kHz |
|  | CP1L-L10DR-D | 24 VDC | 4 relay outputs |  |  |
|  | CP1L-L10DT-A | $\begin{aligned} & 100 \text { to } \\ & 240 \text { VAC } \end{aligned}$ | 4 transistor outputs, sinking |  |  |
|  | CP1L-L10DT-D | 24 VDC | 4 transistor outputs, sinking |  |  |
|  | CP1L-L10DT1-D |  | 4 transistor outputs, sourcing |  |  |
| CPU Units with 14 O Points | CP1L-L14DR-A | $\begin{array}{\|l\|} \hline 100 \text { to } \\ 240 \text { VAC } \end{array}$ | 6 relay outputs | 24 VDC <br> 8 inputs |  |
|  | CP1L-L14DR-D | 24 VDC | 6 relay outputs |  |  |
|  | CP1L-L14DT-A | $\begin{array}{\|l\|} \hline 100 \text { to } \\ 240 \text { VAC } \\ \hline \end{array}$ | 6 transistor outputs, sinking |  |  |
|  | CP1L-L14DT-D | 24 VDC | 6 transistor outputs, sinking |  |  |
|  | CP1L-L14DT1-D |  | 6 transistor outputs, sourcing |  |  |
| CPU Units with 20 I/ O Points | CP1L-L20DR-A | $\begin{aligned} & \hline 100 \text { to } \\ & 240 \text { VAC } \end{aligned}$ | 8 relay outputs | 24 VDC <br> 12 inputs |  |
|  | CP1L-L20DR-D | 24 VDC | 8 relay outputs |  |  |
|  | CP1L-L20DT-A | $\begin{aligned} & 100 \text { to } \\ & 240 \text { VAC } \end{aligned}$ | 8 transistor outputs, sinking |  |  |
|  | CP1L-L20DT-D | 24 VDC | 8 transistor outputs, sinking |  |  |
|  | CP1L-L20DT1-D |  | 8 transistor outputs, sourcing |  |  |


| Name and appearance | Model | Specifications |  |  | Remarks |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Power supply | Outputs | Inputs |  |
| CPU Units with 30 I/ O Points | CP1L-M30DR-A | $\begin{aligned} & 100 \text { to } \\ & 240 \text { VAC } \end{aligned}$ | 12 relay outputs | $\begin{aligned} & 24 \mathrm{VDC} \\ & 18 \text { inputs } \end{aligned}$ | Memory capacity: 10 Ksteps <br> High-speed counters: <br> 100 kHz, 4 counters <br> Pulse outputs: <br> 2 axes at 100 kHz |
|  | CP1L-M30DR-D | 24 VDC | 12 relay outputs |  |  |
|  | CP1L-M30DT-A | $\begin{aligned} & 100 \text { to } \\ & 240 \text { VAC } \end{aligned}$ | 12 transistor outputs, sinking |  |  |
|  | CP1L-M30DT-D | 24 VDC | 12 transistor outputs, sinking |  |  |
|  | CP1L-M30DT1-D |  | 12 transistor outputs, sourcing |  |  |
| CPU Units with 40 I/ O Points | CP1L-M40DR-A | $\begin{aligned} & 100 \text { to } \\ & 240 \text { VAC } \end{aligned}$ | 16 relay outputs | 24 VDC <br> 24 inputs |  |
|  | CP1L-M40DR-D | 24 VDC | 16 relay outputs |  |  |
|  | CP1L-M40DT-A | $\begin{aligned} & 100 \text { to } \\ & 240 \text { VAC } \end{aligned}$ | 16 transistor outputs, sinking |  |  |
|  | CP1L-M40DT-D | 24 VDC | 16 transistor outputs, sinking |  |  |
|  | CP1L-M40DT1-D |  | 16 transistor outputs, sourcing |  |  |
| CPU Units with 60 I/ O Points | CP1L-M60DR-A | $\begin{array}{\|l\|} \hline 100 \text { to } \\ 240 \text { VAC } \end{array}$ | 24 relay outputs | 24 VDC <br> 36 inputs |  |
|  | CP1L-M60DR-D | 24 VDC | 24 relay outputs |  |  |
|  | CP1L-M60DT-A | $\begin{aligned} & 100 \text { to } \\ & 240 \text { VAC } \end{aligned}$ | 24 transistor outputs, sinking |  |  |
|  | CP1L-M60DT-D | 24 VDC | 24 transistor outputs, sinking |  |  |
|  | CP1L-M60DT1-D |  | 24 transistor outputs, sourcing |  |  |

## Programming Devices

| Name and <br> appearance | Model | Application | Remarks |
| :--- | :--- | :--- | :--- |
| CX-Programmer <br> Ver. 7.1 | WS02-CXPC1-E- <br> V71 | Programming and monitoring <br> from a Windows environment | • The CP1L with 14, 20, 30, 40 or 60 points <br> is supported by CX-Programmer version <br> 7.1 or higher. <br> $\bullet$ The CP1L with 10 points is supported by <br> CX-Programmer version 7.3 or higher. <br> - Use an off-the-shelf USB cable to connect <br> the computer running the CX-Program- <br> mer to the USB port on the CP1L CPU <br> Unit. |

## Optional Products

| Name and appearance | Model | Application | Remarks |
| :---: | :---: | :---: | :---: |
| RS-232C Option Board | CP1W-CIF01 | Mounted in option slot 1 or 2 on the CPU Unit to function as an RS-232C port. | --- |
| RS-422A/485 Option Board | $\begin{aligned} & \text { CP1W-CIF11/ } \\ & \text { CIF12 } \end{aligned}$ | Mounted in option slot 1 or 2 on the CPU Unit to function as an RS-422A/485 port. |  |
| LCD Option Board | CP1W-DAM01 | Used to monitor and change user-specified messages, time or other data of the CPU Unit. | --- |
| Ethernet Option Board | CP1W-CIF41 | Can be used to communicate with these units supported OMRON FINS/TCP, FINS/ UDP protocol. | --- |
| Memory Cassette MEMORY | CP1W-ME05M | Used to save CPU Unit user programming, parameters, and data or to copy these to another CPU Unit. | --- |

## Expansion I/O Units

| Name and appearance | Model | Specifications |  | Remarks |
| :---: | :---: | :---: | :---: | :---: |
|  |  | Inputs | Outputs |  |
|  | $\begin{aligned} & \text { CP1W-40EDR } \\ & \text { CPM1A-40EDR } \end{aligned}$ | 24 VDC 24 inputs | 16 relay outputs | --- |
|  | CP1W-40EDT CPM1A-40EDT |  | 16 transistor outputs, sinking |  |
|  | CP1W-40EDT1 CPM1A-40EDT1 |  | 16 transistor outputs, sourcing |  |
| 32-point Output Units | CP1W-32ER | None | 32 relay outputs | --- |
|  | CP1W-32ET |  | 32 transistor outputs, sinking |  |
|  | CP1W-32ET1 |  | 32 transistor outputs, sourcing |  |
| 20-point I/O Units | CP1W-20EDR1 CPM1A-20EDR1 | 24 VDC <br> 12 inputs | 8 relay outputs | --- |
|  | CP1W-20EDT CPM1A-20EDT |  | 8 transistor outputs, sinking |  |
|  | $\begin{aligned} & \text { CP1W-20EDT1 } \\ & \text { CPM1A-20EDT1 } \end{aligned}$ |  | 8 transistor outputs, sourcing |  |


| Name and appearance | Model | Specifications |  | Remarks |
| :---: | :---: | :---: | :---: | :---: |
|  |  | Inputs | Outputs |  |
| 16-point Output Units | CP1W-16ER <br> CPM1A-16ER | None | 16 relay outputs | --- |
|  | CP1W-16ET |  | 16 transistor outputs, sinking |  |
|  | CP1W-16ET1 |  | 16 transistor outputs, sourcing |  |
| 8-point Input Units | $\begin{aligned} & \text { CP1W-8ED } \\ & \text { CPM1A-8ED } \end{aligned}$ | $24 \text { VDC }$ <br> 8 inputs | None | --- |
| 8-point Output Units | CP1W-8ER CPM1A-8ER | None | 8 relay outputs | -- |
|  | $\begin{aligned} & \text { CP1W-8ET } \\ & \text { CPM1A-8ET } \end{aligned}$ |  | 8 transistor outputs, sinking |  |
|  | $\begin{aligned} & \text { CP1W-8ET1 } \\ & \text { CPM1A-8ET1 } \end{aligned}$ |  | 8 transistor outputs, sourcing |  |

## Expansion Units

| Name and appearance | Model | Specifications | Remarks |
| :---: | :---: | :---: | :---: |
| Analog I/O Unit | CPM1A-MAD01 | 2 analog inputs <br> 0 to $10 \mathrm{~V}, 1$ to $5 \mathrm{~V}, 4$ to 20 mA <br> 1 analog output <br> 0 to $10 \mathrm{~V},-10$ to $+10 \mathrm{~V}, 4$ to 20 mA <br> Resolution: 1/256 | --- |
| Analog I/O Unit | CP1W-MAD11 CPM1A-MAD11 | 2 analog inputs <br> 0 to $5 \mathrm{~V}, 1$ to $5 \mathrm{~V}, 0$ to $10 \mathrm{~V},-10$ to $+10 \mathrm{~V}, 0$ to $20 \mathrm{~mA}, 4$ to 20 mA <br> 1 analog output <br> 1 to $5 \mathrm{~V}, 0$ to $10 \mathrm{~V},-10$ to $+10 \mathrm{~V}, 0$ to $20 \mathrm{~mA}, 4$ to 20 mA <br> Resolution: 1/6,000 | --- |
| Analog Input Unit | CP1W-AD041 CPM1A-AD041 | 4 analog inputs <br> 0 to $5 \mathrm{~V}, 1$ to $5 \mathrm{~V}, 0$ to $10 \mathrm{~V},-10$ to $+10 \mathrm{~V}, 0$ to $20 \mathrm{~mA}, 4$ to 20 mA <br> Resolution: 1/6,000 | --- |
| Analog Output Unit $\square$ | CP1W-DA021 | $\begin{aligned} & 2 \text { analog outputs } \\ & 1 \text { to } 5 \mathrm{~V}, 0 \text { to } 10 \mathrm{~V},-10 \text { to }+10 \mathrm{~V}, 0 \text { to } 20 \mathrm{~mA}, 4 \text { to } \\ & 20 \mathrm{~mA} \\ & \text { Resolution: } 1 / 6,000 \end{aligned}$ | --- |
|  | CP1W-DA041 CPM1A-DA041 | 4 analog outputs <br> 1 to $5 \mathrm{~V}, 0$ to $10 \mathrm{~V},-10$ to $+10 \mathrm{~V}, 0$ to $20 \mathrm{~mA}, 4$ to 20 mA <br> Resolution: 1/6,000 | --- |


| Name and appearance | Model | Specifications | Remarks |
| :---: | :---: | :---: | :---: |
| Temperature Sensor Units | $\begin{aligned} & \hline \text { CP1W-TS001 } \\ & \text { CPM1A-TS001 } \end{aligned}$ | Thermocouple inputs K or J, 2 inputs | --- |
|  | CP1W-TS002 CPM1A-TS002 | Thermocouple inputs K or J, 4 inputs | --- |
|  | CP1W-TS101 CPM1A-TS101 | Platinum resistance thermometer inputs Pt100 or JPt100, 2 inputs | --- |
|  | CP1W-TS102 CPM1A-TS102 | Platinum resistance thermometer inputs Pt100 or JPt100, 4 inputs | --- |
| DeviceNet I/O Link Unit | CPM1A-DRT21 | As a DeviceNet Slave, 32 inputs and 32 outputs | e allocated. |
| CompoBus/S I/O Link Unit | CP1W-SRT21 CPM1A-SRT21 | As a CompoBus/S slave, 8 inputs and 8 output | are allocated. |

## Maintenance Products

| Name and appearance | Model | Specifications | Remarks |
| :--- | :--- | :--- | :--- |
| Battery | CJ1W-BAT01 | --- | Installed in the <br> CPU Unit. |
|  |  |  |  |

## Installation and Wiring Products

| Name and appearance | Model | Specifications | Remarks |
| :---: | :---: | :---: | :---: |
| DIN Track | PFP-50N | --- | --- |
|  | PFP-100N | --- |  |
|  | PFP-100N2 | --- |  |
| End Plate | PFP-M | --- |  |
| I/O Connecting Cable | CP1W-CN811 | Used to install CP-series/CPM1A-series Expansion Units and Expansion I/O Units in a second row. | --- |
| 䓢 |  | Only one I/O Connecting Cable can be used in each PLC. |  |

## Appendix B

## Dimensions Diagrams

## CP1L CPU Units

## CPU Units with 10 I/O Points



CPU Units with 14 or 20 I/O Points


Four, 4.5 dia.

## CPU Units with 30 I/O Points



Four, 4.5 dia.

## CPU Units with 40 I/O Points



## CPU Units with 60 I/O Points



## Optional Products

CP1W-CIF01/CIF11/CIF12 Option Boards


## CP1W-DAM01 LCD Option Board



## CP1W-CIF41 Ethernet Option Board



## CP1W-ME05M Memory Cassette



## Expansion I/O Units

## 40-point I/O Units (CP1W/CPM1A-40EDR/40EDT/40EDT1)



## 32-point Output Units (CP1W-32ER/32ET/32ET1)



## 20-point I/O Units (CP1W/CPM1A-20EDR1/20EDT/20EDT1)



## 16-point Output Unit (CP1W/CPM1A-16ER/CP1W-16ET/16ET1)



## 8-point I/O Units (CP1W/CPM1A-8ER/8ET/8ET1)



## Expansion Units

## CPM1A-MAD01/

CP1W/CPM1A/MAD11 Analog I/O Units


CP1W/CPM1A-TS Temperature Sensor Units


CPM1A-DRT21 DeviceNet I/O Link Unit


CP1W/CPM1A-SRT21 CompoBus/S I/O Link Unit


## Appendix C

## Auxiliary Area Allocations by Function

## Initial Settings

| Name | Address | Description | Access | Updated |
| :--- | :--- | :--- | :--- | :--- |
| IOM Hold Bit | A500.12 | Turn this bit ON to retain the status of the I/O Memory when shift- <br> ing from PROGRAM to RUN or MONITOR mode or vice versa or <br> when turning ON the power supply. <br> ON: I/O memory retained <br> OFF: I/O memory not retained | Read/write |  |
| Forced Status Hold <br> BIt | A500.13 | Turn this bit ON to preserve the status of bits that have been force- <br> set or force-reset when shifting from PROGRAM to MONITOR <br> mode or vice versa or when turning ON the power supply. | Read/write |  |

## CPU Unit Settings

| Name | Address | Description | Access | Updated |
| :--- | :--- | :--- | :--- | :--- |
| Status of DIP <br> Switch Pin 6 | A395.12 | The status of pin 6 on the DIP switch on the front of the CPU Unit <br> is written to this flag every cycle. | Read-only |  |
| Manufacturing Lot <br> Number | A310 and <br> A311 | The manufacturing lot number is stored in 5 digits hexadecimal. X, <br> Y, and Z in the lot number are converted to 10,11, and 12, <br> respectively. <br> Examples: <br> Lot number 23805 | Read-only |  |
| A310=0823, A311 = 0005 |  |  |  |  |
| Lot number 15X05 |  |  |  |  |
| A310 $=1015$, A311 = 0005 |  |  |  |  |

DM Initial Value Settings

| Name | Address | Description | Access | Updated |
| :--- | :--- | :--- | :--- | :--- |
| DM Initial Values <br> Flag | A345.04 | ON when DM initial values are stored in the flash memory. | Read-only |  |
| DM Initial Values <br> Read Error Flag | A751.11 | ON when an error occurred in transferring DM initial values from <br> the DM initial value area in flash memory to the DM Area. | Read-only |  |
| DM Initial Values <br> Save Execution <br> Error Flag | A751.12 | ON when the DM Initial Values Transfer Password (A752) is <br> incorrect or when the DM Initial values area was not specified <br> when starting to transfer DM initial values from the DM Area to <br> the DM initial value area in flash memory. | Read-only |  |
| DM Initial Values <br> Save Error Flag | A751.13 | ON when an error occurred in transferring DM initial values from <br> the DM Area to the DM initial value area in flash memory. | Read-only |  |
| DM Initial Values <br> Save Flag | A751.14 | ON while DM initial values are being transferred from the DM <br> Area to the DM initial value area in flash memory. <br> OFF when the transfer has been completed. | Read-only |  |
| DM Initial Values <br> Save Start Bit | A751.15 | Turn ON this bit to start transferring DM initial values. This bit is <br> valid only when a correct password is stored in A752 and the DM <br> Area Initial Value Area is specified (i.e., when A753.00 is ON). <br> The system will turn this bit OFF automatically when the transfer <br> has been completed. | Read/Write |  |
| DM Initial Values <br> Transfer Password | A752 | Set the passwords here to transfer DM initial values between the <br> DM area and the DM initial value area in flash memory. The <br> transfer will not be started unless the correct password is set. <br> The transfer is started when A751.15 is turned ON. <br> The password will be cleared by the system when the transfer <br> has been completed. <br> A5A5 hex: Save initial values from DM to flash | Read/Write |  |
| DM Initial Values <br> Save Area Specifi- <br> cations | A753.00 | Specifies the area to be transferred to flash memory. | Read/Write |  |

## Built-in Inputs

## Analog Adjustment and External Analog Setting Input

| Name | Address | Description | Access | Updated |
| :--- | :--- | :--- | :--- | :--- |
| Analog Adjustment <br> PV | A642 | Stores the value set on the analog adjuster as a hexadecimal <br> value (resolution: 1/256). <br> 0000 to 00FF hex | Read-only | When analog <br> adjustment is <br> turned |
| External Analog <br> Setting Input PV | A643 | Stores the value set from the external analog setting input as a <br> hexadecimal value (resolution: $1 / 256$ ). <br> 0000 to 00FF hex | Read-only |  |

## Input Interrupts, Interrupt Counters 0 to 5

| Interrupt counter | Counter SV | Counter PV |
| :--- | :--- | :--- |
| Interrupt counter 0 | A532 | A536 |
| Interrupt counter 1 | A533 | A537 |
| Interrupt counter 2 | A534 | A538 |
| Interrupt counter 3 | A535 | A539 |
| Interrupt counter 4 | A544 | A548 |
| Interrupt counter 5 | A545 | A549 |


| Name | Description | Access | Updated |
| :--- | :--- | :--- | :--- |
| Interrupt Counter <br> Counter SV | Used for an interrupt input in counter mode. <br> Sets the count value at which the interrupt task will start. The corre- <br> sponding interrupt task will start when the interrupt counter has <br> counted this number of pulses. | Read/Write <br> •Retained when power is <br> turned ON. <br> Retained when opera- <br> tion starts. |  |
| Interrupt Counter   <br> Counter PV These words contain the interrupt counter PVs for interrupt inputs <br> operating in counter mode. <br> In increment mode, the counter PV starts incrementing from 0. <br> When the counter PV reaches the counter SV, the PV is automati- <br> cally reset to 0. <br> In decrement mode, the counter PV starts decrementing from the <br> counter SV. When the counter PV reaches the 0, the PV is automat- <br> ically reset to the SV. Read/Write | - Retained when power is <br> turned ON. <br> $\bullet$ Cleared when operation <br> starts. <br> $\bullet$ Updated when interrupt <br> is generated. |  |  |

High-speed Counters 0 to 3

| Item |  | High-speed counter 0 | High-speed counter 1 | High-speed counter 2 | High-speed counter 3 |
| :---: | :---: | :---: | :---: | :---: | :---: |
| High-speed Counter PV | Leftmost 4 digits | A271 | A273 | A317 | A319 |
|  | Rightmost 4 digits | A270 | A272 | A316 | A318 |
| High-speed Counter Range Comparison Condition Met Flag | Range 1 | A274.00 | A275.00 | A320.00 | A321.00 |
|  | Range 2 | A274.01 | A275.01 | A320.01 | A321.01 |
|  | Range 3 | A274.02 | A275.02 | A320.02 | A321.02 |
|  | Range 4 | A274.03 | A275.03 | A320.03 | A321.03 |
|  | Range 5 | A274.04 | A275.04 | A320.04 | A321.04 |
|  | Range 6 | A274.05 | A275.05 | A320.05 | A321.05 |
|  | Range 7 | A274.06 | A275.06 | A320.06 | A321.06 |
|  | Range 8 | A274.07 | A275.07 | A320.07 | A321.07 |
| High-speed Counter Comparison In-progress Flag |  | A274.08 | A275.08 | A320.08 | A321.08 |
| High-speed Counter Overflow/Underflow Flag |  | A274.09 | A275.09 | A320.09 | A321.09 |
| High-speed Counter Count Direction |  | A274.10 | A275.10 | A320.10 | A321.10 |
| High-speed Counter Count Reset Bit |  | A531.00 | A531.01 | A531.02 | A531.03 |
| High-speed Counter Gate Flag |  | A531.08 | A531.09 | A531.10 | A531.11 |


| Name | Description | Read/Write | Updated |
| :--- | :--- | :--- | :--- |
| High-speed Counter PV | Contains the PV of the high-speed counter. | Read-only | $\bullet$ Cleared when power is turned ON. <br> $\bullet$ Cleared when operation starts. <br> $\bullet$ Updated each cycle during oversee- <br> ing process. <br> •Updated when PRV(881) instruction <br> is executed for the corresponding <br> counter. |
|  |  |  |  |

## Built-in Outputs

## Pulse Outputs 0, 1

| Item |  | Pulse output <br> $\mathbf{0}$ | Pulse output <br> $\mathbf{1}$ |
| :--- | :--- | :--- | :--- |
| Pulse Output PV | Leftmost 4 digits | A277 | A279 |
|  | Rightmost 4 digits | A276 | A278 |
| Pulse Output Accel/Decel Flag | A280.00 | A281.00 |  |
| Pulse Output Overflow/Underflow Flag | A280.01 | A281.01 |  |
| Pulse Output, Output Amount Set Flag | A280.02 | A281.02 |  |
| Pulse Output, Output Completed Flag | A280.03 | A281.03 |  |
| Pulse Output, Output In-progress Flag | A280.04 | A281.04 |  |
| Pulse Output No-origin Flag | A280.05 | A281.05 |  |
| Pulse Output At-origin Flag | A280.06 | A281.06 |  |
| Pulse Output, Output Stopped Error Flag | A280.07 | A281.07 |  |
| PWM Output, Output In-progress Flag | A283.00 | A283.08 |  |


| Item | Pulse output <br> $\mathbf{0}$ | Pulse output <br> $\mathbf{1}$ |
| :--- | :--- | :--- |
| Pulse Output Stop Error Code | A444 | A445 |
| Pulse Output Reset Bit | A540.00 | A541.00 |
| Pulse Output CW Limit Input Signal Flag | A540.08 | A541.08 |
| Pulse Output CCW Limit Input Signal Flag | A540.09 | A541.09 |
| Pulse Output Positioning Completed Signal | A540.10 | A541.10 |


| Name | Description | Read/Write | Updated |
| :---: | :---: | :---: | :---: |
| Pulse Output PV | Contain the number of pulses output from the corresponding pulse output port. PV range: 80000000 to 7FFF FFFF hex $(-2,147,483,648$ to $2,147,483,647)$ <br> When pulses are being output in the CW direction, the PV is incremented by 1 for each pulse. <br> When pulses are being output in the CCW direction, the PV is decremented by 1 for each pulse. <br> PV after overflow: 7FFF FFFF hex <br> PV after underflow: 80000000 hex <br> Note If the coordinate system uses relative coordinates (undefined origin), the PV will be cleared to 0 when a pulse output starts, i.e. when a pulse output instruction (SPED(885), ACC(888), or PLS2(887)) is executed. | Read-only | - Cleared when power is turned ON. <br> - Cleared when operation starts. <br> - Updated each cycle during oversee process. <br> - Updated when the PV is changed by the $\mathrm{INI}(880)$ instruction. |
| Pulse Output Accel/Decel Flag | This flag will be ON when pulses are being output according to an ACC(888) or PLS2(887) instruction and the output frequency is being changed in steps (accelerating or decelerating). <br> OFF: Constant speed <br> ON: Accelerating or decelerating | Read-only | - Cleared when power is turned ON. <br> - Cleared when operation starts or stops. <br> - Updated each cycle during oversee process. |
| Pulse Output Overflow/Underflow Flag | This flag indicates when an overflow or underflow has occurred in the pulse output PV. <br> OFF: Normal <br> ON: Overflow or underflow | Read-only | - Cleared when power is turned ON. <br> - Cleared when operation starts. <br> - Cleared when the PV is changed by the $\mathrm{INI}(880)$ instruction. <br> - Updated when an overflow or underflow occurs. |
| Pulse Output, Output Amount Set Flag | ON when the number of output pulses has been set with the PULS(886) instruction. <br> OFF: No setting <br> ON: Setting made | Read-only | - Cleared when power is turned ON. <br> - Cleared when operation starts or stops. <br> - Updated when the PULS(886) instruction is executed. <br> - Updated when pulse output stops. |
| Pulse Output, Output Completed Flag | ON when the number of output pulses set with the PULS(886) or PLS2(887) instruction has been output. OFF: Output not completed. ON: Output completed. | Read-only | - Cleared when power is turned ON. <br> - Cleared when operation starts or stops. <br> - Updated at the start or completion of pulse output in independent mode. |
| Pulse Output, Output In-progress Flag | ON when pulses are being output. OFF: Stopped ON: Outputting pulses. | Read-only | - Cleared when power is turned ON. <br> - Cleared when operation starts or stops. <br> - Updated when pulse output starts or stops. |
| Pulse Output No-origin Flag | ON when the origin has not been determined and goes OFF when the origin has been determined. <br> OFF: Origin established. <br> ON: Origin not established. | Read-only | - Cleared when power is turned ON. <br> - Cleared when operation starts. <br> - Updated when pulse output starts or stops. <br> - Updated each cycle during the overseeing processes. |
| Pulse Output At-origin Flag | ON when the pulse output PV matches the origin (0). OFF: Not stopped at origin. ON: Stopped at origin. | Read-only | - Cleared when power is turned ON. <br> - Updated each cycle during the overseeing processes. |


| Name | Description | Read/Write | Updated |
| :---: | :---: | :---: | :---: |
| Pulse Output, Output Stopped Error Flag | ON when an error occurred while outputting pulses in the pulse output 0 origin search function. <br> OFF: No error <br> ON: Stop error occurred. | Read-only | - Cleared when power is turned ON. <br> - Updated when origin search starts. <br> - Updated when a pulse output stop error occurs. |
| PWM Output, Output In-progress Flag | ON when pulses are being output from the PWM output. <br> OFF: Stopped <br> ON: Outputting pulses. | Read-only | - Cleared when power is turned ON. <br> - Cleared when operation starts or stops. <br> - Updated when pulse output starts or stops. |
| Pulse Output Stop Error Code | If a Pulse Output Stop Error occurs, the error code is written to this word. | Read-only | - Cleared when power is turned ON. <br> - Updated when origin search starts. <br> - Updated when a pulse output stop error occurs. |
| Pulse Output Reset Bit | The pulse output PV will be cleared when this bit is turned ON. | Read/Write | Cleared when power is turned ON. |
| Pulse Output CW Limit Input Signal Flag | This is the CW limit input signal for the pulse output, which is used in the origin search. To use this signal, write the input from the actual sensor as an input condition in the ladder program and output the result to this flag. | Read/Write | Cleared when power is turned ON. |
| Pulse Output CCW <br> Limit Input Signal Flag | This is the CCW limit input signal for the pulse output, which is used in the origin search. To use this signal, write the input from the actual sensor as an input condition in the ladder program and output the result to this flag. | Read/Write | Cleared when power is turned ON. |
| Pulse Output Positioning Completed Signal | This is the positioning completed input signal used in the origin search for the pulse output. The input signal from the servo driver is output to this bit from the ladder program to enable using the signal. | Read/Write | Cleared when power is turned ON. |

## Inverter Positioning

## Inverter Positioning 0 and 1

| Item |  | Inverter positioning 0 | Inverter positioning 1 |
| :---: | :---: | :---: | :---: |
| Inverter Frequency Command Value |  | A23 | A33 |
| Present Value of Unsigned Output Value | Leftmost 4 digits | A21 | A31 |
|  | Rightmost 4 digits | A20 | A30 |
| Present Value of Signed Output Value | Leftmost 4 digits | A25 | A35 |
|  | Rightmost 4 digits | A24 | A34 |
| Operation Command Flag |  | A26.00 | A36.00 |
| Forward Operation Command Flag |  | A26.01 | A36.01 |
| Reverse Operation Command Flag |  | A26.02 | A36.02 |
| In-position Flag |  | A26.03 | A36.03 |
| Error Counter Error Flag |  | A26.04 | A36.04 |
| Error Counter Pulse Output Flag |  | A26.05 | A36.05 |
| Error Counter Pulse Output Acceleration/Deceleration Flag |  | A26.06 | A36.06 |
| Error Counter Alarm Flag |  | A26.07 | A36.07 |
| Inverter Positioning Output Value Sign Flag |  | A26.15 | A36.15 |
| Error Counter Present Value, Signed |  | A22 | A32 |
| Present Value of Pulse Output to Inverter, Relative Value | Leftmost 4 digits | A29 | A39 |
|  | Rightmost 4 digits | A28 | A38 |
| Error Counter Reset Bit |  | A562.00 | A563.00 |
| Error Counter Disable Bit |  | A562.01 | A563.01 |
| Present Value of High-speed Counter | Leftmost 4 digits | A271 | A273 |
|  | Rightmost 4 digits | A270 | A272 |
| Present Value of Internal Pulse Output | Leftmost 4 digits | A271 | A279 |
|  | Rightmost 4 digits | A270 | A278 |


| Name | Description | Read/Write | Updated |
| :---: | :---: | :---: | :---: |
| Inverter Frequency Command Value | This word contains the automatically calculated frequency command value for the inverter. <br> Data range: 0000 to FFFF hex ( 0.00 to 655.35 Hz ) (0.01Hz increments, unsigned) <br> Set the Power Supply Frequency for One Motor Revolution per Second, Number of Encoder Pulses for One Motor Revolution, and Error Counter Cycle in the PLC Setup before using this value. | Read | - Cleared when power is turned ON. <br> - Cleared when operation starts. <br> - Cleared when an error occurs in the error counter. <br> - Updated each error counter cycle. |
| Present Value of Unsigned Output Value | These words contain the present value of the unsigned output value (output value $=$ present value of error counter $\times$ error counter cycle (s) $\times$ gain). <br> Data range: 00000000 to 80000000 hex (0 to 2,147,483,648) <br> The maximum and minimum output values are applied. | Read | - Cleared when power is turned ON. <br> - Cleared when operation starts. <br> - Cleared when an error occurs in the error counter. <br> - Updated each error counter cycle. |
| Present Value of Signed Output Value | These words contain the present value of the signed output value (output value $=$ present value of error counter $\times$ error counter cycle (s) $\times$ gain). <br> Data range: 80000000 to 7FFF FFFF hex $(-214,748,348 \text { to } 214,748,347)$ <br> The maximum and minimum output values are applied. | Read | - Cleared when power is turned ON. <br> - Cleared when operation starts. <br> - Cleared when an error occurs in the error counter. <br> - Updated each error counter cycle. |
| Operation Command Flag | This flag turns ON during an inverter positioning operation command. <br> ON: Operation command executed. <br> OFF: Stop command executed. | Read | - Turned OFF when power is turned ON. <br> - Turned OFF when operation starts. <br> - Turned OFF when CPU Unit operation stops. <br> - Turned OFF when inverter positioning is stopped (immediate stop) using INI instruction. |
| Forward Operation Command Flag | This flag turns ON during an inverter positioning forward operation command. <br> ON: Forward command in progress <br> OFF: Reverse command in progress or stopped | Read | - Turned OFF when power is turned ON. <br> - Turned OFF when CPU Unit operation starts. <br> - Turned OFF when CPU Unit operation stops. <br> - Turned OFF when error counter present value is 0 or less than 0 (negative). <br> - Turned ON when error counter present value is greater than 0 (positive). |
| Reverse Operation Command Flag | This flag turns ON during an inverter positioning reverse operation command. <br> ON: Reverse command in progress <br> OFF: Forward command in progress or stopped | Read | - Turned OFF when power is turned ON. <br> - Turned OFF when CPU Unit operation starts. <br> - Turned OFF when CPU Unit operation stops. <br> - Turned OFF when error counter present value is 0 or greater than 0 (positive). <br> - Turned ON when error counter present value is less than 0 (negative). |
| In-position Flag | This flag turns ON when inverter positioning is in position. ON: In position OFF: Not in position | Read | - Turned OFF when power is turned ON. <br> - Turned OFF when CPU Unit operation starts. <br> - Turned OFF when CPU Unit operation stops. <br> - Turned OFF when pulses are being output to error counter. <br> - Turned OFF absolute value of error counter present value is greater than in-position range. <br> - Turned OFF when pulse output to error counter is stopped and absolute value of error counter present value is less than in-position range |


| Name | Description | Read/Write | Updated |
| :---: | :---: | :---: | :---: |
| Error Counter Error Flag | This flag turns ON when an error occurs in the error counter for inverter positioning. <br> ON: Error counter error <br> OFF: No error | Read | - Turned OFF when power is turned ON. <br> - Turned OFF when CPU Unit operation starts. <br> - Turned OFF when CPU Unit operation stops. <br> - Turned OFF when error counter error is reset. <br> - Turned ON when pulse output to error counter is stopped and absolute value of error counter present value is greater than or equal to error counter error detection value. |
| Error Counter Pulse Output Flag | This flag is ON while pulses are being output to the output counter for inverter positioning. <br> ON: Pulses being output <br> OFF: Pulse output stopped | Read | - Turned OFF when power is turned ON. <br> - Turned OFF when CPU Unit operation starts. <br> - Turned OFF when CPU Unit operation stops. <br> - Turned OFF when error counter error is reset. <br> - Turned ON when pulse output to error counter is stopped and absolute value of error counter present value is greater than or equal to error counter error detection value. |
| Error Counter Pulse Output Acceleration/Deceleration Flag | This flag is ON while pulse output to the output counter for inverter positioning is accelerating or decelerating. <br> ON: Pulse output to the error counter is accelerating or decelerating (i.e., the frequency is changing) <br> OFF: Pulse output to the error counter is constant | Read | - Turned OFF when power is turned ON. <br> - Turned OFF when CPU Unit operation starts. <br> - Turned OFF when CPU Unit operation stops. <br> - Turned OFF during output of a constant pulse frequency to error counter. <br> - Turned OFF when pulse output to error counter is stopped (including immediate stops and deceleration stops). <br> - Turned ON when pulse output frequency to error counter is changed by ACC or PLS2 instruction. |
| Error Counter Alarm Flag | This flag turns ON when an alarm occurs in the error counter for inverter positioning. <br> ON: Error counter alarm <br> OFF: No alarm | Read | - Turned OFF when power is turned ON. <br> - Turned OFF when CPU Unit operation starts. <br> - Turned OFF when CPU Unit operation stops. <br> - Turned OFF when error counter alarm is reset. <br> - Turned ON when pulse output to error counter is stopped and absolute value of error counter present value is greater than or equal to error counter alarm detection value. |
| Inverter Positioning Output Value Sign Flag | This flag is ON when the inverter positioning output value is positive and is OFF when it is negative. <br> ON: Positive value <br> OFF: Negative value | Read | - Turned ON when signed output value is between 00000000 and 7FFF FFFF hex. <br> - Turned OFF when signed output value is between FFFF FFFF and 80000000 hex. |
| Error Counter Present Value, Signed | This word contains the present value of the error counter. Data range: 8000 to 7FFF hex $(-32,768$ to 32,767$)$ | Read | - Cleared when power is turned ON. <br> - Cleared when operation starts. <br> - Cleared when error counter is reset. <br> - Held when Error Counter Disable Bit (A562.01) is turned ON. <br> - Updated each error counter cycle. |
| Present Value of Pulse Output to Inverter, Relative Value | These words contain the relative value of the internal pulse output value when pulses are being output to the error counter. <br> Data range: 80000000 to 7FFF FFFF hex $(-2,147,483,648$ to $2,147,483,647)$ | Read | - Cleared when power is turned ON. <br> - Cleared when operation starts. <br> - Cleared when pulse output to error counter is started. <br> - Updated each error counter cycle. |


| Name | Description | Read/Write |  |
| :--- | :--- | :--- | :--- |
| Error Counter <br> Reset Bit | Turn ON this bit to reset the Error Counter Present Value <br> and turn OFF the Error Counter Error Flag. | Read/write | Updated |
| Error Counter Dis- <br> able Bit | Turn ON this bit to hold the error counter value. <br> ON: Error counter value held. <br> OFF: Error counter value not held. | Read/write | --- |
| Present Value of <br> High-speed <br> Counter | These words contain the present value of the high-speed <br> counter. | Read | $\bullet$ Cleared when power is turned ON. <br> $\bullet$ Cleared when operation starts. <br> $\bullet$ Updated each cycle during over- <br> see process. <br> Updated when present value is <br> read using PRV instruction. |
| Present Value of <br> Internal Pulse <br> Output, Absolute <br> Value for Abso- <br> lute Coordinates | These words contain the absolute value of the actual <br> movement in relation to the internal pulse origin when <br> pulses are being output to the error counter. <br> Data range: 8000 0000 to 7FFF FFFF hex <br> $(-2,147,483,648$ to 2,147,483,647) | Read | $\bullet$ Cleared when power is turned ON. |
| $\bullet$ Cleared when operation starts. |  |  |  |
| $\bullet$ Updated each error counter cycle. |  |  |  |

## System Flags

| Name | Address | Description | Access | Updated |
| :---: | :---: | :---: | :---: | :---: |
| First Cycle Flag | A200.11 | ON for one cycle after PLC operation begins (after the mode is switched from PROGRAM to RUN or MONITOR, for example). | Read-only |  |
| Initial Task Execution Flag | A200.15 | ON when a task is executed for the first time, i.e., when it changes from INI to RUN status. | Read-only |  |
| Task Started Flag | A200.14 | When a task switches from WAIT or INI to RUN status, this flag will be turned ON within the task for one cycle only. <br> Note The only difference between this flag and A200.15 is that this flag also turns ON when the task switches from WAIT to RUN status. | Read-only |  |
| Maximum Cycle Time | $\begin{aligned} & \text { A262 to } \\ & \text { A263 } \end{aligned}$ | These words contain the maximum cycle time since the start of PLC operation. The cycle time is recorded in 8digit hexadecimal with the leftmost 4 digits in A263 and the rightmost 4 digits in A262. <br> 0 to FFFFFFFF: 0 to $429,496,729.5 \mathrm{~ms}$ ( 0.1 -ms units) | Read-only |  |
| Present Cycle Time | $\begin{aligned} & \text { A264 to } \\ & \text { A265 } \end{aligned}$ | These words contain the present cycle time in 8-digit hexadecimal with the leftmost 4 digits in A265 and the rightmost 4 digits in A264. | Read-only |  |
| 10-ms Incrementing Free Running Timer | A0 | This word contains the system timer used after the power is turned ON. <br> A value of 0000 hex is set when the power is turned ON and this value is automatically incremented by 1 every 10 ms . The value returns to 0000 hex after reaching FFFF hex ( $655,350 \mathrm{~ms}$ ), and then continues to be automatically incremented by 1 every 10 ms . <br> Note: The timer will continue to be incremented when the operating mode is switched to RUN mode. <br> Example: The interval can be counted between processing A and processing B without requiring timer instructions. This is achieved by calculating the difference between the value in A0 for processing A and the value in A0 for processing B. The interval is counted in 10 ms units. | Read-only |  |
| 100-ms Incrementing Free Running Timer | A1 | This word contains the system timer used after the power is turned ON. <br> A value of 0000 hex is set when the power is turned ON and this value is automatically incremented by 1 every 100 ms . The value returns to 0000 hex after reaching FFFF hex ( $6,553,500 \mathrm{~ms}$ ), and then continues to be automatically incremented by 1 every 100 ms . <br> Note: The timer will continue to be incremented when the operating mode is switched to RUN mode. <br> Example: The interval can be counted between processing A and processing B without requiring timer instructions. This is achieved by calculating the difference between the value in A0 for processing A and the value in A0 for processing B. The interval is counted in 100 ms units. | Read-only |  |

Task Information

| Name | Address | Description | Access | Updated |
| :--- | :--- | :--- | :--- | :--- |
| Task Number when Program <br> Stopped | A294 | This word contains the task number of the task that was <br> being executed when program execution was stopped <br> because of a program error. | Read-only |  |
| Maximum Interrupt Task <br> Processing Time | A440 | Contains the Maximum Interrupt Task Processing Time <br> in units of 0.1 ms as hexadecimal data. | Read-only |  |
| Interrupt Task with Max. <br> Processing Time | A441 | Contains the task number of the interrupt task with the <br> maximum processing time. Hexadecimal values 8000 to <br> 80FF correspond to task numbers 00 to FF. Bit 15 is <br> turned ON when an interrupt has occurred. | Read-only |  |
| IR/DR Operation between <br> Tasks | A99.14 | ON when index and data registers are shared between <br> all tasks. <br> OFF: Independent <br> ON: Shared (default) | Read-only |  |

## Debugging Information

Online Editing

| Name | Address | Description | Access | Updated |
| :--- | :--- | :--- | :--- | :--- |
| Online Editing Wait Flag | A201.10 | ON when an online editing process is waiting. | Read-only |  |
| Online Editing Processing <br> Flag | A201.11 | ON when an online editing process is being executed. | Read-only |  |
| Online Editing Disable Bit <br> Validator | A527.00 to <br> A527.07 | The Online Editing Disable Bit (A527.09) is valid only <br> when this byte contains 5A. | Read/write |  |
| Online Editing Disable Bit | A527.09 | Turn this bit ON to disable online editing. The setting of <br> this bit is valid only when A527.00 to A527.07 have been <br> set to 5A. | Read/write |  |

Output Control

| Name | Address | Description | Access | Updated |
| :---: | :---: | :---: | :--- | :--- |
| Output OFF Bit | A500.15 | Turn this bit ON to turn OFF all outputs from the CPU <br> Unit, CPM1A Units, and Special I/O Units. | Read/write |  |

## Differentiate Monitor

| Name | Address | Description | Access | Updated |
| :--- | :--- | :--- | :--- | :--- |
| Differentiate Monitor Com- <br> pleted Flag | A508.09 | ON when the differentiate monitor condition has been <br> established during execution of differentiation monitoring. | Read/write |  |

## Data Tracing

| Name | Address | Description | Access | Updated |
| :--- | :--- | :--- | :--- | :--- |
| Sampling Start Bit | A508.15 | When a data trace is started by turning this bit ON from <br> the CX-Programmer, the PLC will begin storing data in <br> Trace Memory by one of the three following methods: <br> Data is sampled at regular intervals (10 to $2,550 \mathrm{~ms})$. <br> Data is sampled when TRSM(045) is executed in the <br> program. <br> Data is sampled at the end of every cycle. | Read/write |  |
| Trace Start Bit | A508.14 | Turn this bit ON to establish the trigger condition. The off- <br> set indicated by the delay value (positive or negative) <br> determines which data samples are valid. | Read/write |  |
| Trace Busy Flag | A508.13 | ON when the Sampling Start Bit (A508.15) is turned ON. <br> OFF when the trace is completed. | Read/write | Read/write |
| Trace Completed Flag | A508.12 | ON when sampling of a region of trace memory has <br> been completed during execution of a trace. | Read/write |  |
| Trace Trigger Monitor Flag | A508.11 | ON when a trigger condition is established by the Trace <br> Start Bit (A508.14). OFF when the next data trace is <br> started by the Sampling Start Bit (A508.15). | Ren |  |

Comment Memory

| Name | Address | Description | Access | Updated |
| :--- | :--- | :--- | :--- | :--- |
| Program Index File Flag | A345.01 | Turns ON when the comment memory contains a pro- <br> gram index file. <br> OFF: No file <br> ON: File present | Read-only |  |
| Comment File Flag | A345.02 | Turns ON when the comment memory contains a com- <br> ment file. <br> OFF: No file <br> ON: File present | Read-only |  |
| Symbol Table File Flag | A345.03 | Turns ON when the comment memory contains a symbol <br> table file. <br> OFF: No file <br> ON: File present | Read-only |  |

## Error Information

## Error Log, Error Code

| Name | Address | Description | Access | Updated |
| :--- | :--- | :--- | :--- | :--- |
| Error Log Area | A100 to <br> A199 | When an error has occurred, the error code, error con- <br> tents, and error's time and date are stored in the Error <br> Log Area. | Read-only |  |
| Error Log Pointer | A300 | When an error occurs, the Error Log Pointer is incre- <br> mented by 1 to indicate the location where the next error <br> record will be recorded as a hexadecimal offset from the <br> beginning of the Error Log Area (A100 to A199). | Read-only |  |
| Error Log Pointer Reset Bit | A500.14 | Turn this bit ON to reset the Error Log Pointer (A300) to <br> 00. | Read/write |  |
| Error Code | A400 | When a non-fatal error (user-defined FALS(006) or sys- <br> tem error) or a fatal error (user-defined FALS(007) or <br> system error) occurs, the 4-digit hexadecimal error code <br> is written to this word. | Read-only |  |

## Memory Error Information

| Name | Address | Description | Access | Updated |
| :--- | :--- | :--- | :--- | :--- |
| $\begin{array}{l}\text { Memory Error Flag } \\ \text { fatal error) }\end{array}$ | A401.15 | $\begin{array}{l}\text { ON when an error occurred in memory or there was an } \\ \text { error in automatic transfer from the Memory Cassette } \\ \text { when the power was turned ON. } \\ \text { CPU Unit operation will stop and the ERR/ALM indicator } \\ \text { on the front of the CPU Unit will light. } \\ \text { Note A403.09 will be turned ON if there was an error } \\ \text { during automatic transfer at startup. } \\ \text { The automatic transfer at startup error cannot be cleared } \\ \text { without turning OFF the PLC. }\end{array}$ | Read-only |  |$\}$

## Program Error Information

| Name | Address | Description | Access | Updated |
| :---: | :---: | :---: | :---: | :---: |
| Other Fatal Error Flag | A401.00 | ON when a fatal error that is not defined for A401.01 to A401.15 occurs. Detailed information is output to the bits of A314. <br> OFF: No other fatal error <br> ON: Other fatal error |  |  |
| Program Error Flag (fatal error) | A401.09 | ON when program contents are incorrect. CPU Unit operation will stop. | Read-only | When error occurs |
| Program Error Task | A294 | This word contains the task number of the task that was being executed when program execution was stopped because of a program error. | Read-only |  |
| Instruction Processing Error Flag | A295.08 | This flag and the Error Flag (ER) will be turned ON when an instruction processing error has occurred and the PLC Setup has been set to stop operation for an instruction error. | Read-only |  |
| Indirect DM/EM BCD Error Flag | A295.09 | This flag and the Access Error Flag (AER) will be turned ON when an indirect DM BCD error has occurred and the PLC Setup has been set to stop operation an indirect DM BCD error. (This error occurs when the content of an indirectly addressed DM word is not BCD although BCD mode has been selected.) | Read-only |  |
| Illegal Access Error Flag | A295.10 | This flag and the Access Error Flag (AER) will be turned ON when an illegal access error has occurred and the PLC Setup has been set to stop operation an illegal access error. (This error occurs when a region of memory is accessed illegally.) | Read-only |  |
| No END Error Flag | A295.11 | ON when there isn't an END(001) instruction in each program within a task | Read-only |  |
| Task Error Flag | A295.12 | ON when a task error has occurred. The following conditions generate a task error. <br> There isn't even one regular task that is executable (started). <br> There isn't a program allocated to the task. | Read-only |  |
| Differentiation Overflow Error Flag | A295.13 | ON when the allowed value for Differentiation Flags which correspond to differentiation instructions has been exceeded. | Read-only |  |
| Illegal Instruction Error Flag | A295.14 | ON when a program that cannot be executed has been stored. | Read-only |  |
| UM Overflow Error Flag | A295.15 | ON when the last address in UM (User Memory) has been exceeded | Read-only |  |
| Program Address Where Program Stopped | $\begin{array}{\|l} \hline \text { A298 and } \\ \text { A299 } \end{array}$ | These words contain the 8-digit binary program address of the instruction where program execution was stopped due to a program error. <br> A298: Rightmost 4 digits, A299: Leftmost 4 digits | Read-only |  |

FAL/FALS Error Information

| Name | Address | Description | Access | Updated |
| :--- | :--- | :--- | :--- | :--- |
| FAL Error Flag <br> (non-fatal error) | A402.15 | ON when a non-fatal error is generated by executing <br> FAL(006). The CPU Unit will continue operating. | Read-only |  |
| Executed FAL Number <br> Flags | A360 to <br> A391 | The flag corresponding to the specified FAL number will <br> be turned ON when FAL(006) is executed. Bits A360.01 <br> to A391.15 correspond to FAL numbers 001 to 511. | Read-only |  |
| FALS Error Flag <br> (fatal error) | A401.06 | ON when a fatal error is generated by the FALS(006) <br> instruction. The CPU Unit will stop operating. | Read-only |  |
| FAL/FALS Number for Sys- <br> tem Error Simulation | A529 | Set a dummy FAL/FALS number to use to simulate the <br> system error using FAL(006) or FALS(007). <br> Set the FAL/FALS number. <br> 0001 to 01FF hex: FAL/FALS numbers 1 to 511 <br> 0000 or 0200 to FFFF hex: No FAL/FALS number for sys- <br> tem error simulation. (No error will be generated.) | Read/write |  |

PLC Setup Error Information

| Name | Address | Description | Access | Updated |
| :--- | :--- | :--- | :--- | :--- |
| PLC Setup Error Flag <br> (non-fatal error) | A402.10 | ON when there is a setting error in the PLC Setup. | Read-only |  |
| PLC Setup Error Location | A406 | When there is a setting error in the PLC Setup, the loca- <br> tion of that error is written to A406 in 4-digit hexadecimal. | Read-only |  |

## I/O Information

| Name | Address | Description | Access | Updated |
| :---: | :---: | :---: | :---: | :---: |
| Too Many I/O Points Flag (fatal error) | A401.11 | ON when the number of Expansion Units and Expansion I/O Units exceeds the limit, when the number of words allocated to these Units exceeds the limit, are mounted. | Read-only |  |
| Too Many I/O Points, Details | $\begin{aligned} & \text { A407.00 to } \\ & \text { A407.12 } \end{aligned}$ | Always 0000 hex. | Read-only |  |
| Too Many I/O Points, Cause | $\begin{aligned} & \text { A407.13 to } \\ & \text { A407.15 } \end{aligned}$ | The 3-digit binary value of these bits indicates the cause of the Too Many I/O Points Error. <br> 010: Too many words <br> 011: Too many Units | Read-only |  |
| I/O Bus Error Flag (fatal error) | A401.14 | ON in the following cases: <br> - When an error occurs in a data transfer between the CPU Unit and a Expansion Unit or Expansion I/O Unit. If this happens, OA0A hex will be output to A404. <br> - When an error occurs in a data transfer between the CPU Unit. If this happens, 0000 hex will be output to A404 to indicate the first Unit, 0001 hex to indicate the second Unit, and 0F0F hex to indicate an undetermined Unit. <br> CPU Unit operation will stop and the ERR/ALM indicator on the front of the CPU Unit will light. <br> (This flag will be turned OFF when the error is cleared.) | Read-only |  |
| I/O Bus Error Slot Number | A404 | Contains information on I/O bus errors. <br> The CPU Unit will stop operating and the ERR/ALM indicator on the front of the CPU Unit will light. <br> (A401.04 (I/O Bus Error Flag) will turn ON.) <br> (This information will be cleared when the error is cleared.) <br> OAOA hex: Expansion Unit or Expansion I/O Unit error | Read-only |  |
| Duplication Error Flag (fatal error) | A401.13 | ON in the following cases: <br> - Two CPU Bus Units have been assigned the same unit number. | Read-only |  |
| Unit Error Flags | $\begin{aligned} & \text { A436.00 to } \\ & \text { A436.02 } \end{aligned}$ | ON when an error occurs in an Expansion Unit or Expansion I/O Unit. <br> A436.00: 1st Unit <br> A436.10: 2nd Unit <br> A436.02: 3rd Unit <br> CP1W/CPM1A-TS002 and CP1W/CPM1A-TS102 are each counted as two Units. | Read-only |  |
| Number of Connected Units | A437 | Stores the number of Expansion Units and Expansion I/O Units connected as a hexadecimal number. <br> Note This information is valid only when a Too Many I/O Points error has occurred. CP1W/CPM1ATS002 and CP1W/CPM1A-TS102 are each counted as two Units. | Read-only |  |

## Other PLC Operating Information

| Name | Address | Description | Access | Updated |
| :--- | :--- | :--- | :--- | :--- |
| Battery Error Flag <br> (non-fatal error) | A402.04 | ON if the CPU Unit's battery is disconnected or its volt- <br> age is low and the Detect Battery Error setting has been <br> set in the PLC Setup. | Read-only |  |
| Cycle Time Too Long Flag <br> (fatal error) | A401.08 | ON if the cycle time exceeds the maximum cycle time set <br> in the PLC Setup (the cycle time monitoring time). | Read-only |  |
| FPD Teaching Bit | A598.00 | Turn this bit ON to set the monitoring time automatically <br> with the teaching function. | Read/write |  |


| Name | Address | Description | Access | Updated |
| :--- | :--- | :--- | :--- | :--- |
| Option Board Error Flag | A315.13 | ON when the Option Board is removed while the power is <br> being supplied. <br> CPU Unit operation will continue and the ERR/ALM indi- <br> cator will flash. <br> OFF when the error has been cleared. | Read-only <br> OF | When an error <br> occurs |
| Flash Memory Error Flag | A315.15 | ON when writing to the internal flash memory fails. <br> CPU Unit operation will continue and the ERR/ALM indi- <br> cator will flash. <br> OFF when the error has been cleared. | Read-only | When an error <br> occurs |
| Other Fatal Error Flag | A402.00 | ON when a non-fatal error that is not defined for A402.01 <br> to A402.15 occurs. Detailed information is output to the <br> bits of A314. <br> OFF: No other fatal error <br> ON: Other fatal error | Read-only | When an error <br> occurs |

## Clock

## Clock Information

| Name | Address | Description | Access | Updated |
| :---: | :---: | :---: | :---: | :---: |
| Clock Data | The clock data from the clock built into the CPU Unit is stored here in BCD. |  | Read-only |  |
|  | A351.00 to A351.07 | Seconds: 00 to 59 (BCD) |  |  |
|  | A351.08 to A351.15 | Minutes: 00 to 59 (BCD) |  |  |
|  | A352.00 to A352.07 | Hour: 00 to 23 (BCD) |  |  |
|  | A352.08 to A352.15 | Day of the month: 01 to 31 (BCD) |  |  |
|  | A353.00 to A353.07 | Month: 01 to 12 (BCD) |  |  |
|  | A353.08 to A353.15 | Year: 00 to 99 (BCD) |  |  |
|  | A354.00 to A354.07 | Day of the week: 00: Sunday, 01: Monday, 02: Tuesday, 03: Wednesday, 04: Thursday, 05: Friday, 06: Saturday |  |  |

Note The clock data is stored in the CPU Unit as BCD.
Operation Start and End Times

| Name | Address | Description | Access | Updated |
| :---: | :---: | :---: | :---: | :---: |
| Operation Start Time | $\begin{array}{\|l} \hline \text { A515 to } \\ \text { A517 } \end{array}$ | The time that operation started as a result of changing the operating mode to RUN or MONITOR mode is stored here in BCD. <br> A515.00 to A515.07: Seconds (00 to 59) <br> A515.08 to A515.15: Minutes (00 to 59) <br> A516.00 to A516.07: Hour (00 to 23) <br> A516.08 to A516.15: Day of month (01 to 31) <br> A517.00 to A517.07: Month (01 to 12) <br> A517.08 to A517.15: Year (00 to 99) <br> Note The previous start time is stored after turning ON the power supply until operation is started. | Read/write |  |
| Operation End Time | $\begin{array}{\|l} \hline \text { A518 to } \\ \text { A520 } \end{array}$ | The time that operation stopped as a result of changing the operating mode to PROGRAM mode is stored here in BCD. <br> A518.00 to A518.07: Seconds (00 to 59) <br> A518.08 to A518.15: Minutes (01 to 59) <br> A519.00 to A519.07: Hour (00 to 23) <br> A519.08 to A519.15: Day of month (01 to 31) <br> A520.00 to A520.07: Month (01 to 12) <br> A520.08 to A520.15: Year (00 to 99) <br> Note If an error occurs in operation, the time of the error will be stored. If the operating mode is then changed to PROGRAM mode, the time that PROGRAM mode was entered will be stored. | Read/write |  |

Power Supply Information

| Name | Address | Description | Access | Updated |
| :--- | :--- | :--- | :--- | :--- |
| Startup Time | A510 and <br> A511 | These words contain the time at which the power was <br> turned ON. The contents are updated every time that the <br> power is turned ON. The data is stored in BCD. <br> A510.00 to A510.07: Second (00 to 59) <br> A510.08 to A510.15: Minute (00 to 59) <br> A511.00 to A511.07: Hour (00 to 23) <br> A511.08 to A511.15: Day of month (01 to 31) | Read/write |  |
| Power Interruption Time | A512 and <br> A513 | These words contain the time at which the power was <br> interrupted. The contents are updated every time that the <br> power is interrupted. The data is stored in BCD. <br> A512.00 to A512.07: Second (00 to 59) <br> A512.08 to A512.15: Minute (00 to 59) <br> A513.00 to A513.07: Hour (00 to 23) <br> A513.08 to A513.15: Day of month (01 to 31) <br> (These words are not cleared at startup.) | Read/write |  |
| Number of Power <br> Interruptions | A514 | Contains the number of times that power has been inter- <br> rupted since the power was first turned ON. The data is <br> stored in binary. To reset this value, overwrite the current <br> value with 0000. | Read/write |  |
| Total Power ON Time | A523 | Contains the total time that the PLC has been ON in 10- <br> hour units. The data is stored in binary and it is updated <br> every 10 hours. To reset this value, overwrite the current <br> value with 0000. | Read/write |  |

Flash Memory Backup Information

| Name | Address | Description | Access | Updated |
| :---: | :---: | :---: | :---: | :---: |
| User Program Date | A90 to A93 | These words contain in BCD the date and time that the user program was last overwritten. <br> A90.00 to A90.07: Seconds (00 to 59) <br> A90.08 to A90.15: Minutes (00 to 59) <br> A91.00 to A91.07: Hour (00 to 23) <br> A91.08 to A91.15: Day of month (01 to 31) <br> A92.00 to A92.07: Month (01 to 12) <br> A92.08 to A92.15: Year (00 to 99) <br> A93.00 to A93.07: Day of the week (00 to 06) <br> (00: Sunday, 01: Monday, 02: Tuesday, 03: Wednesday, <br> 04: Thursday, 05: Friday, 06: Saturday) | Read-only |  |
| Parameter Date | A94 to A97 | These words contain in BCD the date and time that the parameters were last overwritten. <br> A94.00 to A94.07: Seconds (00 to 59) <br> A94.08 to A94.15: Minutes (00 to 59) <br> A95.00 to A95.07: Hour (00 to 23) <br> A95.08 to A95.15: Day of month (01 to 31) <br> A96.00 to A96.07: Month (01 to 12) <br> A96.08 to A96.15: Year (00 to 99) <br> A97.00 to A97.07: Day of the week (00 to 06) <br> (00: Sunday, 01: Monday, 02: Tuesday, 03: Wednesday, <br> 04: Thursday, 05: Friday, 06: Saturday) | Read-only |  |

## Memory Cassette Information

| Name | Address | Description | Access | Updated |
| :---: | :---: | :---: | :---: | :---: |
| Memory Cassette Access Status | A342 | A342.03: ON when data is being written to the Memory Cassette or the Memory Cassette is being initialized. OFF when processing has been completed. <br> A342.04: ON when data is being read from the Memory Cassette. OFF when processing has been completed. <br> A342.05: ON when data is being compared with data on the Memory Cassette. OFF when processing has been completed. <br> A342.07: ON when an error occurs in initializing the Memory Cassette. <br> OFF the next time the Memory Cassette is accessed normally (initialized, written, read, or compared). <br> A342.08: ON when an error occurs in writing the Memory Cassette. <br> OFF the next time the Memory Cassette is accessed normally (initialized, written, read, or compared). <br> A342.10: ON when an error occurs in reading or comparing the Memory Cassette. <br> OFF the next time the Memory Cassette is accessed normally (initialized, written, read, or compared). <br> A342.12: ON when the data in the CPU Unit is not the same as the data in the Memory Cassette when a verification operation is performed. OFF the next time the Memory Cassette is accessed normally (initialized, written, read, or compared). <br> A342.13: ON when the Memory Cassette is being accessed. OFF when processing has been completed. <br> A342.15: ON when a Memory Cassette is mounted. OFF when a Memory Cassette is not mounted. | Read-only |  |
| Memory Casette Verification Results | A494 | Stores the results of comparing data in the Memory Cassette and CPU Unit. Each bit turns ON to indicate status. <br> A494.00: User program is different. <br> A494.01: Function block sources are different. <br> A494.02: Parameter area is different. <br> A494.03: Symbol table is different. <br> A494.04: Comments are different. <br> A494.05: Program indices are different. <br> A494.06: Data memory is different. <br> A494.07: DM initial values are different. | Read-only |  |

Information on Read Protection Using a Password

| Name | Address | Description | Access | Updated |
| :--- | :--- | :--- | :--- | :--- |
| UM Read Protection Flag | A99.00 | Indicates whether the entire user program in the PLC is <br> read-protected. <br> OFF: UM not read-protected. <br> ON: UM read-protected. | Read-only |  |
| Task Read Protection Flag | A99.01 | Indicates whether read protection is set for individual <br> tasks. <br> OFF: Tasks not read-protected. <br> ON: Tasks read-protected. | Read-only |  |
| Program Write Protection for <br> Read Protection | A99.02 | Indicates whether the program is write-protected. <br> OFF: Write-enabled. <br> ON: Write-protected. | Read-only |  |
| Enable/Disable Bit for Pro- <br> gram Backup | A99.03 | Indicates whether creating a backup program file (.OBJ) <br> is enabled or disabled. <br> OFF: Enabled. <br> ON: Disabled. | Read-only |  |
| UM Read Protection <br> Release Enable Flag | A99.12 | Indicates when UM read protection cannot be released <br> because an incorrect password was input five times con- <br> secutively. <br> OFF: Protection can be released <br> ON: Protection cannot be released | Read-only |  |
| Task Read Protection <br> Release Enable Flag | A99.13 | Indicates when task read protection cannot be released <br> because an incorrect password was input five times con- <br> secutively. <br> OFF: Protection can be released <br> ON: Protection cannot be released | Read-only |  |

## Communications

## Networks

## Network Communications Information

| Name | Address | Description | Access | Updated |
| :---: | :---: | :---: | :---: | :---: |
| Communications Port Enabled Flags | $\begin{array}{\|l} \hline \text { A202.00 to } \\ \text { A202.07 } \end{array}$ | ON when a network instruction or background execution can be executed with the corresponding port number. Bits 00 to 07 correspond to communications ports 0 to 7 . | Read-only |  |
| Communications Port Completion Codes | $\begin{aligned} & \text { A203 to } \\ & \text { A210 } \end{aligned}$ | These words contain the completion codes for the corresponding port numbers when network instructions have been executed. <br> Words A203 to A210 correspond to communications ports 0 to 7. | Read-only |  |
| Communications Port Error Flags | $\begin{aligned} & \text { A219.00 to } \\ & \text { A219.07 } \end{aligned}$ | ON when an error occurred during execution of a network instruction. <br> OFF when a normal response is returned. <br> Bits 00 to 07 correspond to communications ports 0 to 7 . | Read-only |  |

Information When Automatically Allocating Communications Ports.

| Name | Address | Description | Access | Updated |
| :---: | :---: | :---: | :---: | :---: |
| Network Communications Port Allocation Enabled Flag | A202.15 | ON when there is a communications port available for automatic allocation. <br> Note Use this flag to confirm whether a communications port is available for automatic allocation before executing communications instructions when using 9 or more communications instructions simultaneously. | Read-only |  |
| First Cycle Flags after Network Communications Finished | A214.00 to A214.07 | Each flag will turn ON for just one cycle after communications have been completed. Bits 00 to 07 correspond to ports 0 to 7 . Use the Used Communications Port Number stored in A218 to determine which flag to access. <br> Note These flags are not effective until the next cycle after the communications instruction is executed. Delay accessing them for at least one cycle. | Read-only |  |
| First Cycle Flags after Network Communications Error | $\begin{aligned} & \text { A215.00 to } \\ & \text { A215.07 } \end{aligned}$ | Each flag will turn ON for just one cycle after a communications error occurs. Bits 00 to 07 correspond to ports 0 to 7 . Use the Used Communications Port Number stored in A218 to determine which flag to access. Determine the cause of the error according to the Communications Port Completion Codes stored in A203 to A210. <br> Note These flags are not effective until the next cycle after the communications instruction is executed. Delay accessing them for at least one cycle. | Read-only |  |
| Network Communications Completion Code Storage Address | $\begin{aligned} & \text { A216 to } \\ & \text { A217 } \end{aligned}$ | The completion code for a communications instruction is automatically stored at the address with the I/O memory address given in these words. <br> Place this address into an index register and use indirect addressing through the index register to read the communications completion code. | Read-only |  |
| Used Communications Port Numbers | A218 | Stores the communications port numbers used when a communications instruction is executed using automatic communication port allocations. 0000 to 0007 hex: Communications port 0 to 7 | Read-only |  |

## Serial Port 1 Information (CP1L CPU Units with M CPU Type)

| Name | Address | Description | Access | Updated |
| :--- | :--- | :--- | :--- | :--- |
| Peripheral Port Communica- <br> tions Error Flag | A392.12 | ON when a communications error has occurred at the <br> serial port 1. | Read-only |  |
| Peripheral Port Restart Bit | A526.01 | Turn this bit ON to restart the serial port 1. | Read/write |  |
| Peripheral Port Settings <br> Change Bit | A619.01 | ON while the serial port 1's communications settings are <br> being changed. | Read/write |  |
| Peripheral Port Error Flags | A528.08 to <br> A528.15 | These flags indicate what kind of error has occurred at <br> the serial port 1. | Read/write |  |
| Serial Port 1 Send Ready <br> Flag <br> (No-protocol Mode) | A392.13 | ON when the serial port 1 is able to send data in no-pro- <br> tocol mode. | Read-only |  |


| Name | Address | Description | Access | Updated |
| :--- | :--- | :--- | :--- | :--- |
| Serial Port 1 Reception <br> Completed Flag <br> (No-protocol Mode) | A392.14 | ON when the serial port 1 has completed the reception in <br> no-protocol mode. | Read-only |  |
| Serial Port 1 Reception <br> Overflow Flag <br> (No-protocol Mode) | A392.15 | ON when a data overflow occurred during reception <br> through the serial port 1 in no-protocol mode. | Read-only |  |
| Peripheral Port PT Commu- <br> nications Flags | A394.00 to <br> A394.07 | The corresponding bit will be ON when the serial port 1 <br> is communicating with a PT in NT link mode. <br> Bits 0 to 7 correspond to units 0 to 7. | Read-only |  |
| Peripheral Port PT Priority <br> Registered Flags | A394.08 to <br> A394.15 | The corresponding bit will be ON for the PT that has pri- <br> ority when the serial port 1 is communicating in NT link <br> mode. | Read-only |  |
| Serial Port 1 Reception <br> Counter <br> (No-protocol Mode) | A394.00 to <br> A394.15 | Indicates (in binary) the number of bytes of data received <br> when serial port 1 is in no-protocol mode. | Read-only |  |

Serial Port 1 Information (CP1L CPU Units with L CPU Type)

| Name | Address | Description | Access | Updated |
| :--- | :--- | :--- | :--- | :--- |
| Serial Port 1 Communica- <br> tions Error Flag | A392.04 | ON when a communications error has occurred at the <br> serial port 1. <br> Note Not supported for 1:N NT Link Mode. | Read-only |  |
| Serial Port 1 Restart Bit | A526.00 | Turn this bit ON to restart the serial port 1. | Read/write |  |
| Serial Port 1 Settings <br> Change Bit | A619.02 | ON while the serial port 1's communications settings are <br> being changed. | Read/write |  |
| Serial Port 1 Error Flags | A528.00 to <br> A528.07 | These flags indicate what kind of error has occurred at <br> the serial port 1. | Read/write |  |
| Serial Port 1 Send Ready <br> Flag (No-protocol mode) | A392.05 | ON when the serial port 1 is able to send data in no-pro- <br> tocol mode. | Read-only |  |
| Serial Port 1 Reception <br> Completed Flag (No-proto- <br> col Mode) | A392.06 | ON when the serial port 1 has completed the reception in <br> no-protocol mode. | Read-only |  |
| Serial Port 1 Reception <br> Overflow Flag (No-protocol <br> mode) | A392.07 | ON when a data overflow occurred during reception <br> through the serial port 1 in no-protocol mode. | Read-only |  |
| Serial Port 1 PT Communi- <br> cations Flags | A393.00 to <br> A393.07 | The corresponding bit will be ON when the serial port 1 <br> is communicating with a PT in NT link mode. <br> Bits 0 to 7 correspond to units 0 to 7. | Read-only |  |
| Serial Port 1 PT Priority <br> Registered Flags | A393.08 to <br> A393.15 | The corresponding bit will be ON for the PT that has pri- <br> ority when the serial port 1 is communicating in NT link <br> mode. <br> Bits 0 to 7 correspond to units 0 to 7. | Read-only |  |
| Serial Port 1 Reception <br> Counter (No-protocol Mode) | A393.00 to <br> A393.15 | Indicates (in binary) the number of bytes of data received <br> when serial port 1 is in no-protocol mode. | Read-only |  |

## Serial Port 2 Information (CP1L CPU Units with M CPU Type)

| Name | Address | Description | Access | Updated |
| :--- | :--- | :--- | :--- | :--- |
| Serial Port 2 Communica- <br> tions Error Flag | A392.04 | ON when a communications error has occurred at the <br> serial port 2. | Read-only |  |
| Serial Port 2 Restart Bit | A526.00 | Turn this bit ON to restart the serial port 2. | Read/write |  |
| Serial Port 2 Settings <br> Change Bit | A619.02 | ON while the serial port 2's communications settings are <br> being changed. | Read/write |  |
| Serial Port 2 Error Flags | A528.00 to <br> A528.07 | These flags indicate what kind of error has occurred at <br> the serial port 2. | Read/write |  |
| Serial Port 2 Send Ready <br> Flag <br> (No-protocol mode) | A392.05 | ON when the serial port 2 is able to send data in no-pro- <br> tocol mode. | Read-only |  |
| Serial Port 2 Reception <br> Completed Flag <br> (No-protocol Mode) | A392.06 | ON when the serial port 2 has completed the reception in <br> no-protocol mode. | Read-only | Read-only |
| Serial Port 2 Reception <br> Overflow Flag <br> (No-protocol mode) | A392.07 | ON when a data overflow occurred during reception <br> through the serial port 2 in no-protocol mode. | Rel\| |  |


| Name | Address | Description | Access | Updated |
| :--- | :--- | :--- | :--- | :--- |
| Serial Port 2 PT Communi- <br> cations Flags | A393.00 to <br> A393.07 | The corresponding bit will be ON when the serial port 2 <br> is communicating with a PT in NT link mode. <br> Bits 0 to 7 correspond to units 0 to 7. | Read-only |  |
| RS-232C Port PT Priority <br> Registered Flags | A393.08 to <br> A393.15 | The corresponding bit will be ON for the PT that has pri- <br> ority when the serial port 2 is communicating in NT link <br> mode. | Read-only |  |
| RS-232C Port Reception <br> Counter <br> (No-protocol Mode) | A393.00 to <br> A393.15 | Indicates (in binary) the number of bytes of data received <br> when serial port 2 is in no-protocol mode. | Read-only |  |

Modbus-RTU Easy Master Information (CP1L CPU Units with M CPU Type)

| Name | Address | Description | Access | Updated |
| :---: | :---: | :---: | :---: | :---: |
| Serial Port 1 Modbus-RTU Master Execution Bit | A641.00 | Turn ON this bit to send a command and receive a response for serial port 1 using the Modbus-RTU easy master function. <br> This bit will be turned OFF automatically by the system when communications have been completed. <br> Turned ON: Execution started ON: Execution in progress. <br> OFF: Not executed or execution completed. | Read-only |  |
| Serial Port 1 Modbus-RTU <br> Master Execution Normal Flag | A641.01 | ON when one command has been sent and the response received for serial port 1 using the ModbusRTU easy master function. <br> ON: Execution normal. <br> OFF: Execution error or still in progress. | Read-only |  |
| Serial Port 1 Modbus-RTU Master Execution Error Flag | A641.02 | ON when an error has occurred in communications for serial port 1 using the Modbus-RTU easy master function. The error code is output to D32352 in the DM fixed allocation words for Modbus-RTU Easy Master. <br> ON: Execution error. <br> OFF: Execution normal or still in progress. | Read-only |  |
| Serial Port 2 Modbus-RTU Master Execution Bit | A640.00 | Turn ON this bit to send a command and receive a response for serial port 2 using the Modbus-RTU easy master function. <br> This bit will be turned OFF automatically by the system when communications have been completed. <br> Turned ON: Execution started ON: Execution in progress. <br> OFF: Not executed or execution completed. | Read-only |  |
| Serial Port 2 Modbus-RTU Master Execution Normal Flag | A640.01 | ON when one command has been sent and the response received for serial port 2 using the ModbusRTU easy master function. <br> ON: Execution normal. <br> OFF: Execution error or still in progress. | Read-only |  |
| Serial Port 2 Modbus-RTU Master Execution Error Flag | A640.02 | ON when an error has occurred in communications for serial port 2 using the Modbus-RTU easy master function. The error code is output to D32252 in the DM fixed allocation words for Modbus-RTU Easy Master. <br> ON: Execution error. <br> OFF: Execution normal or still in progress. | Read-only |  |

Note DM fixed allocation words for Modbus-RTU Easy Master for serial port 1: D32200 to D32299 DM fixed allocation words for Modbus-RTU Easy Master for serial port 2: D32300 to D32399

## Modbus-RTU Easy Master Information (CP1L CPU Units with L CPU Type)

| Name | Address | Description | Access | Updated |
| :--- | :--- | :--- | :--- | :--- |
| Serial Port 1 Modbus-RTU <br> Master Execution Bit | A640.00 | Turn ON this bit to send a command and receive a <br> response for serial port 2 using the Modbus-RTU easy <br> master function. <br> This bit will be turned OFF automatically by the system <br> when communications have been completed. <br> Turned ON: Execution started <br> ON: Execution in progress. <br> OFF: Not executed or execution completed. | Read-only |  |
| Serial Port 1 Modbus-RTU <br> Master Execution Normal <br> Flag | A640.01 | ON when one command has been sent and the <br> response received for serial port 2 using the Modbus- <br> RTU easy master function. <br> ON: Execution normal. <br> OFF: Execution error or still in progress. | Read-only |  |
| Serial Port 1 Modbus-RTU <br> Master Execution Error Flag | A640.02 | ON when an error has occurred in communications for <br> serial port 2 using the Modbus-RTU easy master func- <br> tion. The error code is output to D32252 in the DM fixed <br> allocation words for Modbus-RTU Easy Master. <br> ON: Execution error. <br> OFF: Execution normal or still in progress. | Read-only |  |

Note DM fixed allocation words for Modbus-RTU Easy Master for serial port 2: D32300 to D32399

## Instruction-related Information

| Name | Address | Description | Access | Updated |
| :--- | :--- | :--- | :--- | :--- |
| Step Flag | A200.12 | ON for one cycle when step execution is started with <br> STEP(008). | Read-only |  |
| Macro Area Input Words | A600 to <br> A603 | Before the subroutine specified in MCRO(099) is exe- <br> cuted, the source words for the subroutine are trans- <br> ferred to A600 through A603 (input parameter words). | Read/write |  |
| Macro Area Output Words | A604 to <br> A607 | After the subroutine specified in MCRO(099) has been <br> executed, the results of the subroutine are transferred <br> from A604 through A607 to the specified destination <br> words (output parameter words). | Read/write |  |

## Function Block Information

Function Block Memory Information

| Name | Address | Description | Access | Updated |
| :--- | :--- | :--- | :--- | :--- |
| FB Program Data Flag | A345.00 | Turns ON if the FB program memory contains FB pro- <br> gram data. <br> OFF: No data <br> ON: Data present | Read-only |  |

## OMRON FB Library Information

| Name | Address | Description | Access | Updated |
| :--- | :--- | :--- | :--- | :--- |
| FB Communications Instruc- <br> tion Response Required | A580.15 | 0: Not required <br> 1: Required | Read-only |  |
| FB Communications Instruc- <br> tion Port No. | A580.08 to <br> A580.11 | 0 to 7 hex: Communications port No. 0 to 7 <br> F hex: Automatic allocation | Read-only |  |
| FB Communications Instruc- <br> tion Retries | A580.00 to <br> A580.03 | Automatically stores the number of retries in the FB com- <br> munications instruction settings specified in the PLC <br> Setup. | Read-only |  |
| FB Communications Instruc- <br> tion Response Monitoring <br> Time | A581 | Automatically stores the FB communications instruction <br> response monitoring time set in the PLC Setup. <br> 0001 to FFFF hex (Unit: 0.1 s; Range: 0.1 to 6553.5) <br> 0000 hex: 2 s | Read-only |  |
| FB DeviceNet Communica- <br> tions Instruction Response <br> Monitoring Time | A582 | Automatically stores the FB DeviceNet communications <br> instruction response monitoring time set in the PLC <br> Setup. <br> 0001 to FFFF hex (Unit: 0.1 s; Range: 0.1 to 6553.5) <br> 0000 hex: 2 s | Read-only |  |

Note These Auxiliary Area bits/words are not to be written by the user. The number of resends and response monitoring time must be set by the user in the FB communications instructions settings in the PLC Setup, particularly when using function blocks from the OMRON FB Library to execute FINS messages or DeviceNet explicit messages communications. The values set in the Settings for

OMRON FB Library in the PLC Setup will be automatically stored in the related Auxiliary Area words A580 to A582 and used by the function blocks from the OMRON FB Library.

## Ethernet Option Board Flag

System Settings Reset

| Name | Address | Description | Access | Updated |
| :--- | :---: | :--- | :--- | :--- |
| Ethernet Option Board <br> Reset Flag | A525.00 to <br> A525.01 | If the flag turns ON, the Ethernet Option Board mounted <br> on the corresponding option board slot will reset the sys- <br> tem settings to default values. | Read/write |  |

## Unit Restart

| Name | Address | Description | Access | Updated |
| :--- | :--- | :--- | :--- | :--- |
| Ethernet Option Board | A525.08 to | If the flag turns ON, the Ethernet Option Board mounted | Read/write |  |
| Restart Flag | A525.09 | on the corresponding option board slot will be restarted. |  |  |

## Appendix D

## Auxiliary Area Allocations by Address

## Read-only Area (Set by System)

| Address |  | Name | Function | Settings | Status after mode change | Status <br> at startup | Write timing | Related flags, settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Words | Bits |  |  |  |  |  |  |  |
| A0 | --- | 10-ms Incrementing Free Running Timer | This word contains the system timer used after the power is turned ON. <br> A value of 0000 hex is set when the power is turned ON and this value is automatically incremented by 1 every 10 ms . The value returns to 0000 hex after reaching FFFF hex <br> ( $655,350 \mathrm{~ms}$ ), and then continues to be automatically incremented by 1 every 10 ms . <br> Note: The timer will continue to be incremented when the operating mode is switched to RUN mode. <br> Example: The interval can be counted between processing $A$ and processing $B$ without requiring timer instructions. This is achieved by calculating the difference between the value in A0 for processing A and the value in $A 0$ for processing $B$. The interval is counted in 10 ms units. | --- | Retained | Cleared | Every 10 ms after power is turned ON | --- |
| A1 | --- | 100-ms Incrementing Free Running Timer | This word contains the system timer used after the power is turned ON. <br> A value of 0000 hex is set when the power is turned ON and this value is automatically incremented by 1 every 100 ms . The value returns to 0000 hex after reaching FFFF hex ( $6,553,500 \mathrm{~ms}$ ), and then continues to be automatically incremented by 1 every 100 ms . <br> Note: The timer will continue to be incremented when the operating mode is switched to RUN mode. <br> Example: The interval can be counted between processing $A$ and processing $B$ without requiring timer instructions. This is achieved by calculating the difference between the value in A0 for processing $A$ and the value in $A 0$ for processing $B$. The interval is counted in 100 ms units. | --- | Retained | Cleared | Every 100 ms after power is turned ON | --- |
| $\begin{aligned} & \text { A20 } \\ & \text { and } \\ & \text { A21 } \end{aligned}$ | --- | Present Value of Unsigned Output Value 0 | These words contain the present value of the unsigned output value (output value $=$ present value of error counter $\times$ error counter cycle (s) $\times$ gain) for inverter positioning 0. <br> Data range: 00000000 to 80000000 hex (0 to 2,147,483,648) The maximum and minimum output values set for inverter positioning 0 in the PLC Setup are applied. <br> A21 contains the leftmost 4 digits and A20 contains the rightmost 4 digits. | --- | --- | Cleared | Every error counter 0 cycle | --- |


| Address |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | ```Related flags, settings``` |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Words | Bits |  |  |  |  |  |  |  |
| A22 | --- | Error Counter 0 Present Value, Signed | This word contains the present value of the error counter for inverter positioning 0. <br> Data range: 8000 to 7FFF hex (-32,768 to 32,767) (signed) | --- | --- | Cleared | Every error counter 0 cycle | --- |
| A23 | --- | Inverter Frequency 0 Command Value | This word contains the automatically calculated frequency command value for the inverter for inverter positioning 0. <br> Data range: 0000 to FFFF hex (0.00 to 655.35 Hz$)(0.01-\mathrm{Hz}$ increments, unsigned) <br> Set the Power Supply Frequency for One Motor Revolution per Second, Number of Encoder Pulses for One Motor Revolution, and Error Counter Cycle for inverter positioning 0 in the PLC Setup before using this value. | --- | --- | Cleared | Every error counter 0 cycle | --- |
| $\begin{aligned} & \text { A24 } \\ & \text { and } \\ & \text { A25 } \end{aligned}$ | --- | Present Value of Signed Output Value 0 | These words contain the present value of the signed output value (output value $=$ present value of error counter $\times$ error counter cycle (s) $\times$ gain) for inverter positioning 0. <br> Data range: 80000000 to 7FFF <br> FFFF hex (-214,748,348 to $214,748,347$ ) (signed) <br> The maximum and minimum output values set for inverter positioning 0 in the PLC Setup are applied. <br> A25 contains the leftmost 4 digits and A24 contains the rightmost 4 digits. | --- | --- | Cleared | Every error counter 0 cycle | --- |
| A26 | 00 | Operation Command Flag 0 | This flag turns ON during an operation command for inverter positioning 0. | ON: Operation command executed. <br> OFF: Stop command executed. | --- | Cleared | When inverter positioning 0 is started | --- |
|  | 01 | Forward Operation Command Flag 0 | This flag turns ON during a forward operation command for inverter positioning 0 . | ON: Forward command in progress <br> OFF: Reverse command in progress or stopped | --- | Cleared | When present value of error counter 0 is positive | --- |
|  | 02 | Reverse Operation Command Flag 0 | This flag turns ON during a reverse operation command for inverter positioning 0 . | ON: Reverse command in progress OFF: Forward command in progress or stopped | --- | Cleared | When present value of error counter 0 is negative | --- |
|  | 03 | In-position Flag 0 | This flag turns ON when inverter positioning 0 is in position. | ON: In position OFF: Not in position | --- | Cleared | When pulse output to error counter 0 is stopped and the present value of error counter 0 is within the in-position range | --- |


| Address |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related flags, settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Words | Bits |  |  |  |  |  |  |  |
| A26 | 04 | Error Counter Error Flag 0 | This flag turns ON when an error occurs in the error counter for inverter positioning 0 . | ON: Error counter error OFF: No error | --- | Cleared | When pulse output to error counter is stopped and absolute value of error counter present value is greater than or equal to error counter error detection value | --- |
|  | 05 | Error Counter Pulse Output Flag 0 | This flag is ON while pulses are being output to the output counter for inverter positioning 0. | ON: Pulses being output OFF: Pulse output stopped | --- | Cleared | When pulse output to error counter 0 is started | --- |
|  | 06 | Error Counter Pulse Output Acceleration/ Deceleration Flag 0 | This flag is ON while pulse output to the output counter for inverter positioning 0 is accelerating or decelerating. | ON: Pulse output to the error counter is accelerating or decelerating (i.e., the frequency is changing) OFF: Pulse output to the error counter is constant | --- | Cleared | When pulse output frequency to error counter is changed by ACC or PLS2 instruction | --- |
|  | 07 | Error Counter Alarm Flag 0 | This flag turns ON when an alarm occurs in the error counter for inverter positioning 0. | ON: Error counter alarm OFF: No error counter alarm | --- | Cleared | When pulse output to error counter is stopped and absolute value of error counter present value is greater than or equal to error counter alarm detection value | - |
|  | 15 | Inverter Positioning Output Value Sign Flag 0 | This flag is ON when the inverter positioning 0 output value is positive and is OFF when it is negative. | ON: Positive value OFF: Negative value | --- | Cleared | When signed output value is between 00000000 and 7FFF FFFF hex | --- |
| $\begin{aligned} & \text { A28 } \\ & \text { and } \\ & \text { A29 } \end{aligned}$ | --- | Present Value of Pulse Output to Inverter 0, Relative Value | These words contain the relative value of the internal pulse output when pulses are being output to the error counter for inverter positioning 0. <br> Data range: 80000000 to 7FFF <br> FFFF hex $(-2,147,483,648$ to $2,147,483,647)$ | --- | --- | Cleared | Every error counter 0 cycle | --- |


| Address |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related flags, settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Words | Bits |  |  |  |  |  |  |  |
| $\begin{aligned} & \text { A30 } \\ & \text { and } \\ & \text { A31 } \end{aligned}$ | --- | Present Value of Unsigned Output Value 1 | These words contain the present value of the unsigned output value (output value $=$ present value of error counter $\times$ error counter cycle (s) $\times$ gain) for inverter positioning 1. <br> Data range: 00000000 to 80000000 hex ( 0 to $2,147,483,648$ ) The maximum and minimum output values set for inverter positioning 1 in the PLC Setup are applied. <br> A30 contains the leftmost 4 digits and A31 contains the rightmost 4 digits. | --- | --- | Cleared | Every error counter 1 cycle | --- |
| A32 | --- | Error Counter 1 Present Value, Signed | This word contains the present value of the error counter for inverter positioning 1. <br> Data range: 8000 to 7FFF hex (-32,768 to 32,767) (signed) | --- | --- | Cleared | Every error counter 1 cycle | --- |
| A33 | --- | Inverter Frequency Command Value 1 | This word contains the automatically calculated frequency command value for the inverter for inverter positioning 1. <br> Data range: 0000 to FFFF hex (0.00 to 655.35 Hz$)(0.01-\mathrm{Hz}$ increments, unsigned) <br> Set the Power Supply Frequency for One Motor Revolution per Second, Number of Encoder Pulses for One Motor Revolution, and Error Counter Cycle in the PLC Setup before using this value. | --- | --- | Cleared | Every error counter 1 cycle | --- |
| $\begin{aligned} & \text { A34 } \\ & \text { and } \\ & \text { A35 } \end{aligned}$ | --- | Present Value of Signed Output Value 1 | These words contain the present value of the signed output value (output value $=$ present value of error counter $\times$ error counter cycle (s) $\times$ gain) for inverter positioning 1. <br> Data range: 80000000 to 7FFF <br> FFFF hex (-214,748,348 to $214,748,347$ ) (signed) <br> The maximum and minimum output values set for inverter positioning 1 in the PLC Setup are applied. <br> A34 contains the leftmost 4 digits and A35 contains the rightmost 4 digits. | --- | --- | Cleared | Every error counter 1 cycle | --- |


| Address |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related flags, settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Words | Bits |  |  |  |  |  |  |  |
| A36 | 00 | Operation Command Flag 1 | This flag turns ON during an operation command for inverter positioning 1. | ON: Operation command executed. <br> OFF: Stop command executed. | --- | Cleared | When inverter positioning 1 is started | --- |
|  | 01 | Forward Operation Command Flag 1 | This flag turns ON during a forward operation command for inverter positioning 1. | ON: Forward command in progress <br> OFF: Reverse command in progress or stopped | --- | Cleared | When present value of error counter 1 is positive | --- |
|  | 02 | Reverse Operation Command Flag 1 | This flag turns ON during a reverse operation command for inverter positioning 1 . | ON: Reverse command in progress <br> OFF: Forward command in progress or stopped | --- | Cleared | When present value of error counter 1 is negative | --- |
|  | 03 | In-position Flag 1 | This flag turns ON when inverter positioning 1 is in position. | ON: In position OFF: Not in position | --- | Cleared | When pulse output to error counter 1 is stopped and the present value of error counter 1 is within the in-position range | --- |
|  | 04 | Error Counter Error Flag 1 | This flag turns ON when an error occurs in the error counter for inverter positioning 1. | ON: Error counter error OFF: No error | --- | Cleared | When pulse output to error counter is stopped and absolute value of error counter present value is greater than or equal to error counter error detection value | --- |
|  | 05 | Error Counter Pulse Output Flag 1 | This flag is ON while pulses are being output to the output counter for inverter positioning 1. | ON: Pulses being output OFF: Pulse output stopped | --- | Cleared | When pulse output to error counter 1 is started | --- |
|  | 06 | Error Counter Pulse Output Acceleration/ Deceleration Flag 1 | This flag is ON while pulse output to the output counter for inverter positioning 1 is accelerating or decelerating. | ON: Pulse output to the error counter is accelerating or decelerating (i.e., the frequency is changing) OFF: Pulse output to the error counter is constant | --- | Cleared | When pulse output frequency to error counter is changed by ACC or PLS2 instruction | --- |


| Address |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related flags, settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Words | Bits |  |  |  |  |  |  |  |
| A36 | 07 | Error Counter Alarm Flag 1 | This flag turns ON when an alarm occurs in the error counter for inverter positioning 1. | ON: Error counter alarm OFF: No error counter alarm | --- | Cleared | When pulse output to error counter is stopped and absolute value of error counter present value is greater than or equal to error counter alarm detection value | --- |
|  | 15 | Inverter Positioning Output Value Sign Flag 1 | This flag is ON when the inverter positioning 1 output value is positive and is OFF when it is negative. | ON: Positive value OFF: Negative value | --- | Cleared | When signed output value is between FFFF FFFF and 80000000 | --- |
| A38 and A39 |  | Present Value of Pulse Output to Inverter 1, Relative Value | These words contain the relative value of the internal pulse output when pulses are being output to the error counter for inverter positioning 1. <br> Data range: 80000000 to 7FFF FFFF hex $(-2,147,483,648$ to 2,147,483,647) <br> A38 contains the leftmost 4 digits and A39 contains the rightmost 4 digits. | --- | --- | Cleared | Every error counter 1 cycle | --- |
| A90 to A93 | All | User Program Date | These words contain in BCD the date and time that the user program was last overwritten. <br> A90.00 to A90.07: <br> Seconds (00 to 59) <br> A90.08 to A90.15: <br> Minutes (00 to 59) <br> A91.00 to A91.07: Hour (00 to 23) A91.08 to A91.15: <br> Day of month (01 to 31) <br> A92.00 to A92.07: Month (01 to 12) <br> A92.08 to A92.15: Year (00 to 99) <br> A93.00 to A93.07: Day of the week <br> (00: Sunday, 01: Monday, 02: Tuesday, 03: Wednesday, 04: Thursday, 05: Friday, 06: Saturday) | --- | Retained | Retained | --- | --- |
| A94 to A97 | All | Parameter Date | These words contain in BCD the date and time that the parameters were last overwritten. <br> The format is the same as above. | --- | Retained | Retained | --- | --- |


| Address |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related flags, settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Words | Bits |  |  |  |  |  |  |  |
| A99 | A99.00 | UM Read Protection Status | Indicates whether the entire user program in the PLC is read-protected. | OFF: UM not read-protected. <br> ON: UM readprotected. | Retained | Retained | When protection is set or cleared | --- |
|  | A99.01 | Task Read Protection Status | Indicates whether read protection is set for individual tasks. | OFF: Tasks not read-protected. <br> ON: Tasks read-protected. | Retained | Retained | When protection is set or cleared | --- |
|  | A99.02 | Program Write Protection Status when Read Protection Is Set | Indicates whether the program is write-protected. | OFF: Writeenabled. ON: Writeprotected. | Retained | Retained | When protection is set or cleared | --- |
|  | A99.03 | Enable/Disable Status for Backing Up the Program to a Memory Cassette | Indicates whether creating a backup program file (.OBJ) is enabled or disabled. | OFF: <br> Enabled. <br> ON: Disabled. | Retained | Retained | When protection is set or cleared | --- |
|  | A99.12 | UM Read Protection Release Enable Flag | Indicates when UM read protection cannot be released because an incorrect password was input five times consecutively. | OFF: Protection can be released <br> ON: Protection cannot be released | Retained | Retained | When wrong pass word is input for the fifth time, when | --- |
|  | A99.13 | Task Read Protection Release Enable Flag | Indicates when task read protection cannot be released because an incorrect password was input five times consecutively. | OFF: Protection can be released ON: Protection cannot be released | Retained | Retained | cleared, and two hours after releasing protection is disabled | --- |
|  | A99.14 | IR/DR Operation between Tasks Retained | ON when index and data registers are shared between all tasks. <br> OFF when separate index and data registers are being used in each task. | OFF: Independent ON: Shared (default) | Retained | Retained |  | --- |
|  | A99.15 | Timer/Counter PV Refresh Mode Flag | Indicates whether the CPU Unit is operating in BCD mode or binary mode. | OFF: BCD mode ON: Binary mode | Retained | Retained |  | --- |


| Address |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related flags, settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Words | Bits |  |  |  |  |  |  |  |
| $\begin{aligned} & \text { A100 } \\ & \text { to } \\ & \text { A199 } \end{aligned}$ | All | Error Log Area | When an error has occurred, the error code, error contents, and error's time and date are stored in the Error Log Area. Information on the 20 most recent errors can be stored. <br> Each error record occupies 5 words; the function of these 5 words is as follows: <br> 1) Error code (bits 0 to 15) <br> 2) Error contents (bits 0 to 15) <br> 3) Minutes (bits 8 to 15), Seconds (bits 0 to 7) <br> 4) Day of month (bits 8 to 15), Hours (bits 0 to 7) <br> 5) Year (bits 8 to 15), Month (bits 0 to 7 ) <br> Errors generated by FAL(006) and FALS(007) will also be stored in this Error Log. <br> The Error Log Area can be reset from the CX-Programmer. <br> If the Error Log Area is full (20 records) and another error occurs, the oldest record in A100 to A104 will be cleared, the other 19 records are shifted down, and the new record is stored in A195 to A199. | Error code <br> Error contents: <br> Address of Aux. Area word with details or 0000. <br> Seconds: 00 to 59, BCD Minutes: 00 to 59, BCD Hours: 00 to 23, BCD Day of month: 01 to 31, BCD Month: 01 to 12, BCD Year: 00 to $99, B C D$ | Retained | Retained | Refreshed when error occurs. | $\begin{array}{\|l} \hline \text { A500.14 } \\ \text { A300 } \\ \text { A400 } \end{array}$ |
| A200 | A200.11 | First Cycle Flag | ON for one cycle after PLC operation begins (after the mode is switched from PROGRAM to RUN or MONITOR, for example). | ON for the first cycle | --- | --- | --- | --- |
|  | A200.12 | Step Flag | ON for one cycle when step execution is started with STEP(008). This flag can be used for initialization processing at the beginning of a step. | ON for the first cycle after execution of STEP(008). | Cleared | --- | --- | - |
|  | A200.14 | Task Started Flag | When a task switches from WAIT or INI to RUN status, this flag will be turned ON within the task for one cycle only. <br> The only difference between this flag and A200.15 is that this flag also turns ON when the task switches from WAIT to RUN status. | ON: ON for first cycle (including transitions from WAIT and IN) OFF: Other | Cleared | Cleared | --- | --- |
|  | A20015 | First Task Startup Flag | ON when a task is executed for the first time. This flag can be used to check whether the current task is being executed for the first time so that initialization processing can be performed if necessary. | ON: First execution OFF: Not executable for the first time or not being executed. | Cleared | --- | --- | --- |
| A201 | A201.10 | Online Editing Wait Flag | ON when an online editing process is waiting. <br> (If another online editing command is received while waiting, the other command won't be recorded and an error will occur.) | ON: Waiting for online editing <br> OFF: Not waiting for online editing | Cleared | Cleared | --- | A527 |
|  | A201.11 | Online Editing Flag | ON when an online editing process is being executed. | ON: Online editing in progress OFF: Online editing not in progress | Cleared | Cleared | --- | A527 |


| Address |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Relatedflags, set-tings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Words | Bits |  |  |  |  |  |  |  |
| A202 | $\begin{aligned} & \hline \text { A202.00 } \\ & \text { to } \\ & \text { A202.07 } \end{aligned}$ | $\begin{aligned} & \text { Communica- } \\ & \text { tions Port } \\ & \text { Enabled Flags } \end{aligned}$ | ON when a network instruction (SEND, RECV, CMND, or PMCR) can be executed with the corresponding port number. Bits 00 to 07 correspond to communications ports 0 to 7. <br> When two or more network instructions are programmed with the same port number, use the corresponding flag as an execution condition to prevent the instructions from being executed simultaneously. <br> (The flag for a given port is turned OFF while a network instruction with that port number is being executed.) | ON: Network instruction is not being executed <br> OFF: Network instruction is being executed (port busy) | Cleared | --- | --- | --- |
|  | A202.15 | Network Communications Port Allocation Enabled Flag | ON when there is a communications port available for automatic allocation. <br> Note Use this flag to confirm whether a communications port is available for automatic allocation before executing communications instructions when using 9 or more communications instructions simultaneously. | ON: Port available OFF: Port not available | Cleared | --- | --- | --- |
| $\begin{aligned} & \text { A203 } \\ & \text { to } \\ & \text { A210 } \end{aligned}$ | All | Communications Port Completion Codes | These words contain the completion codes for the corresponding port numbers when network instructions (SEND, RECV, CMND, or PMCR) have been executed. <br> (The corresponding word will be cleared when background execution has been completed.) <br> Words A203 to A210 correspond to communications ports 0 to 7 . | Non-zero: <br> Error code <br> 0000: <br> Normal condition | Retained | --- | --- | --- |
| A214 | $\begin{aligned} & \text { A214.00 } \\ & \text { to } \\ & \text { A214.07 } \end{aligned}$ | First Cycle Flags after Network Communications Finished | Each flag will turn ON for just one cycle after communications have been completed. Bits 00 to 07 correspond to ports 0 to 7 . Use the Used Communications Port Number stored in A218 to determine which flag to access. <br> Note These flags are not effective until the next cycle after the communications instruction is executed. Delay accessing them for at least one cycle. | ON: First cycle after communications finish only <br> OFF: Other status |  |  | --- | --- |
| A215 | $\begin{aligned} & \text { A215.00 } \\ & \text { to } \\ & \text { A215.07 } \end{aligned}$ | First Cycle Flags after Network Communications Error | Each flag will turn ON for just one cycle after a communications error occurs. Bits 00 to 07 correspond to ports 0 to 7 . Use the Used Communications Port Number stored in A218 to determine which flag to access. Determine the cause of the error according to the Communications Port Completion Codes stored in A203 to A210. <br> Note These flags are not effective until the next cycle after the communications instruction is executed. Delay accessing them for at least one cycle. | ON: First cycle after communications error only <br> OFF: Other status |  |  | --- | --- |
| A216 to A217 | All | Network Communications Completion Code Storage Address | The completion code for a communications instruction is automatically stored at the address with the I/O memory address given in these words. <br> Place this address into an index register and use indirect addressing through the index register to read the communications completion code. | I/O memory address for the network communications completion code storage |  |  | --- | --- |


| Address |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related flags, settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Words | Bits |  |  |  |  |  |  |  |
| A218 | All | Used Communications Port Numbers | Stores the communications port numbers used when a communications instruction is executed using automatic communication port allocations. | 0000 to 0007 hex: Communications port 0 to 7 |  |  | --- | --- |
| A219 | $\begin{aligned} & \text { A219.00 } \\ & \text { to } \\ & \text { A219.07 } \end{aligned}$ | Communications Port Error Flags | ON when an error occurred during execution of a network instruction (SEND, RECV, CMND, or PMCR). <br> Bits 00 to 07 correspond to communications ports 0 to 7. | ON: Error occurred OFF: Normal condition | Retained | --- | --- | --- |
| $\begin{aligned} & \text { A262 } \\ & \text { and } \\ & \text { A263 } \end{aligned}$ | All | Maximum Cycle Time | These words contain the maximum cycle time since the start of PLC operation. The cycle time is recorded in 8-digit hexadecimal with the leftmost 4 digits in A263 and the rightmost 4 digits in A262. | 0 to <br> FFFFFFFF: <br> 0 to <br> 429,496,729. <br> 5 ms <br> (0.1-ms units) | --- | --- | --- | --- |
| $\begin{aligned} & \text { A264 } \\ & \text { and } \\ & \text { A265 } \end{aligned}$ | All | Present Cycle Time | These words contain the present cycle time in 8-digit hexadecimal with the leftmost 4 digits in A265 and the rightmost 4 digits in A264. | 0 to <br> FFFFFFFF: <br> 0 to 429,496,729. <br> 5 ms | --- | --- | --- | --- |
| $\begin{aligned} & \text { A270 } \\ & \text { to } \\ & \text { A271 } \end{aligned}$ | All | High-speed Counter 0 PV | Contains the PV of high-speed counter 0 . A271 contains the leftmost 4 digits and A270 contains the rightmost 4 digits. <br> The PV is cleared when operation starts. | --- | --- | Cleared | Refreshed each cycle during oversee process. Refreshed when PRV(881) instruction is executed. | --- |
| $\begin{aligned} & \text { A272 } \\ & \text { to } \\ & \text { A273 } \end{aligned}$ | All | High-speed Counter 1 PV | Contains the PV of high-speed counter 1. A273 contains the leftmost 4 digits and A272 contains the rightmost 4 digits. <br> The PV is cleared when operation starts. | --- | --- | Cleared | Refreshed each cycle during oversee process. Refreshed when PRV(881) instruction is executed. | --- |


| Address |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related flags, settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Words | Bits |  |  |  |  |  |  |  |
| A274 | A274.00 | High-speed Counter 0 Range 1 Comparison Condition Met Flag | These flags indicate whether the PV is within the specified ranges when high-speed counter 0 is being operated in range-comparison mode. Cleared at beginning of operation. Cleared when range comparison table is registered. <br> OFF: PV not in range ON: PV in range | --- | --- | Cleared | Refreshed each cycle during oversee process. <br> Refreshed when PRV(881) instruction is executed. | --- |
|  | A274.01 | High-speed Counter 0 Range 2 Comparison Condition Met Flag |  |  |  |  |  |  |
|  | A274.02 | High-speed Counter 0 Range 3 Comparison Condition Met Flag |  |  |  |  |  |  |
|  | A274.03 | High-speed Counter 0 Range 4 Comparison Condition Met Flag |  |  |  |  |  |  |
|  | A274.04 | High-speed Counter 0 Range 5 Comparison Condition Met Flag |  |  |  |  |  |  |
|  | A274.05 | High-speed Counter 0 Range 6 Comparison Condition Met Flag |  |  |  |  |  |  |
|  | A274.06 | High-speed Counter 0 Range 7 Comparison Condition Met Flag |  |  |  |  |  |  |
|  | A274.07 | High-speed Counter 0 Range 8 Comparison Condition Met Flag |  |  |  |  |  |  |
| A274 | A274.08 | High-speed Counter 0 Comparison In-progress Flag | This flag indicates whether a comparison operation is being executed for high-speed counter 0. <br> Cleared at beginning of operation. <br> OFF: Stopped. <br> ON: Being executed. | --- | --- | Cleared | Refreshed when comparison operation starts or stops. | --- |
|  | A274.09 | High-speed Counter 0 Overflow/ Underflow Flag | This flag indicates when an overflow or underflow has occurred in the high-speed counter 0 PV. (Used with the linear mode counting range only.) Cleared when operation starts. Cleared when PV is changed. <br> OFF: Normal <br> ON: Overflow or underflow | --- | --- | Cleared | Refreshed when an overflow or underflow occurs. | --- |
|  | A274.10 | High-speed Counter 0 Count Direction | This flag indicates whether the highspeed counter is currently being incremented or decremented. The counter PV for the current cycle is compared with the PLC in last cycle to determine the direction. <br> OFF: Decrementing <br> ON: Incrementing | --- | --- | Cleared | Setting used for high-speed counter, valid during counter operation. | Read only |


| Address |  | Name | Function | Settings | Status after mode change | Status <br> at startup | Write timing | Related flags, settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Words | Bits |  |  |  |  |  |  |  |
| A275 |  |  |  |  |  |  | Refreshed each cycle during overseeing pro- | --- |
|  |  |  |  |  |  |  | Refreshed when PRV(881) instruction is exe- |  |
|  |  |  |  |  |  |  | cuted for the corresponding counter. |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | Refreshed when comparison operation starts or stops. | --- |
| A275 |  |  |  |  |  |  | Refreshed when an overflow or underflow occurs. | --- |
|  |  |  |  |  |  |  | Setting used for high-speed counter, valid during counter operation. | --- |


| Address |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related flags, settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Words | Bits |  |  |  |  |  |  |  |
| $\begin{aligned} & \text { A276 } \\ & \text { and } \\ & \text { A277 } \end{aligned}$ | All | Pulse Output 0 PV | Contain the number of pulses output from the corresponding pulse output port. <br> PV range: 80000000 to 7FFF FFFF hex <br> (-2,147,483,648 to $2,147,483,647$ ) <br> When pulses are being output in the CW direction, the PV is incremented by 1 for each pulse. <br> When pulses are being output in the CCW direction, the PV is decremented by 1 for each pulse. <br> PV after overflow: 7FFF FFFF hex PV after underflow: 80000000 hex A277 contains the leftmost 4 digits and A276 contains the rightmost 4 digits of the pulse output 0 PV. <br> A279 contains the leftmost 4 digits and A278 contains the rightmost 4 digits of the pulse output 1 PV . <br> Cleared when operation starts. <br> Note If the coordinate system is relative coordinates (undefined origin), the PV will be cleared to 0 when a pulse output starts, i.e. when a pulse output instruction (SPED(885), ACC(888), or PLS2(887)) is executed. | --- | --- | Cleared | Refreshed each cycle during oversee process. <br> Refreshed when the INI(880) instruction is executed (PV change). | --- |
| $\begin{aligned} & \text { A278 } \\ & \text { and } \\ & \text { A279 } \end{aligned}$ | All | Pulse Output 1 PV |  |  |  | Cleared |  | --- |
| A280 | A280.00 | Pulse Output 0 Accel/Decel Flag | This flag will be ON when pulses are being output from pulse output 0 according to an ACC(888) or PLS2(887) instruction and the output frequency is being changed in steps (accelerating or decelerating). <br> Cleared when operation starts or stops. <br> OFF: Constant speed <br> ON: Accelerating or decelerating | - | --- | Cleared | Refreshed each cycle during oversee process. | --- |
|  | A280.01 | Pulse Output 0 <br> Overflow/ <br> Underflow Flag | This flag indicates when an overflow or underflow has occurred in the pulse output 0 PV. <br> Cleared when operation starts. <br> OFF: Normal <br> ON: Overflow or underflow | --- | --- | Cleared | Cleared when the PV is changed by the $\mathrm{NI}(880)$ instruction. Refreshed when an overflow or underflow occurs. | --- |


| Address |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related flags, settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Words | Bits |  |  |  |  |  |  |  |
| A280 | A280.02 | Pulse Output 0 Output Amount Set Flag | ON when the number of output pulses for pulse output 0 has been set with the PULS(886) instruction. Cleared when operation starts or stops. <br> OFF: No setting <br> ON: Setting made | --- | --- | Cleared | Refreshed when the PULS(886) instruction is executed. <br> Refreshed when pulse output stops. | --- |
|  | A280.03 | Pulse Output 0 Output Completed Flag | ON when the number of output pulses set with the PULS(886) or PLS2(887) instruction has been output through pulse output 0. <br> Cleared when operation starts or stops. <br> OFF: Output not completed. <br> ON: Output completed. | --- | --- | Cleared | Refreshed at the start or completion of pulse output in independent mode. | --- |
|  | A280.04 | Pulse Output 0 Output Inprogress Flag | ON when pulses are being output from pulse output 0. <br> Cleared when operation starts or stops. <br> OFF: Stopped <br> ON: Outputting pulses. | --- | --- | Cleared | Refreshed when pulse output starts or stops. | --- |
|  | A280.05 | Pulse Output 0 No-origin Flag | ON when the origin has not been determined for pulse output 0 and goes OFF when the origin has been determined. <br> Turned ON when power is turned ON. <br> Turned ON when operation starts. <br> OFF: Origin established. <br> ON: Origin not established. | --- | --- | Cleared | Refreshed each cycle during the overseeing processes. | --- |
|  | A280.06 | Pulse Output 0 At-origin Flag | ON when the pulse output PV matches the origin (0). <br> OFF: Not stopped at origin. ON: Stopped at origin. | --- | --- | Cleared | Refreshed each cycle during the overseeing processes. | --- |
|  | A280.07 | Pulse Output O Output Stopped Error Flag | ON when an error occurred while outputting pulses in the pulse output 0 origin search function. <br> The Pulse Output 0 Output Stop Error code will be written to A444. <br> OFF: No error <br> ON: Stop error occurred. | - | --- | Cleared | Refreshed when origin search starts. <br> Refreshed when a pulse output stop error occurs. | --- |
| A281 | A281.00 | Pulse Output <br> 1 Accel/Decel Flag | This flag will be ON when pulses are being output from pulse output 1 according to an ACC(888) or PLS2(887) instruction and the output frequency is being changed in steps (accelerating or decelerating). <br> Cleared when operation starts or stops. <br> OFF: Constant speed <br> ON: Accelerating or decelerating | --- | --- | Cleared | Refreshed each cycle during oversee process. | --- |


| Address |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related flags, settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Words | Bits |  |  |  |  |  |  |  |
| A281 | A281.01 | Pulse Output 1 <br> Overflow/ <br> Underflow <br> Flag | This flag indicates when an overflow or underflow has occurred in the pulse output 1 PV. <br> Cleared when operation starts. <br> OFF: Normal <br> ON: Overflow or underflow | --- | --- | Cleared | Refreshed when the PV is changed by the $\mathrm{INI}(880)$ instruction. <br> Refreshed when an overflow or underflow occurs. | --- |
|  | A281.02 | Pulse Output 1 Output Amount Set Flag | ON when the number of output pulses for pulse output 1 has been set with the PULS(886) instruction. Cleared when operation starts or stops. <br> OFF: No setting <br> ON: Setting made | --- | --- | Cleared | Refreshed when the PULS(886) instruction is executed. | --- |
|  | A281.03 | Pulse Output 1 Output Completed Flag | ON when the number of output pulses set with the PULS(886) or PLS2(887) instruction has been output through pulse output 1. <br> Cleared when operation starts or stops. <br> OFF: Output not completed. <br> ON: Output completed. | --- | --- | Cleared | Refreshed when <br> PULS(886) (886) instruction is executed. <br> Refreshed at the start or completion of pulse output. | --- |
|  | A281.04 | Pulse Output 1 Output Inprogress Flag | ON when pulses are being output from pulse output 1. <br> Cleared when operation starts or stops. <br> OFF: Stopped <br> ON: Outputting pulses. | --- | --- | Cleared | Refreshed when pulse output starts or stops. | --- |
|  | A281.05 | Pulse Output 1 No-origin Flag | ON when the origin has not been determined for pulse output 1 and goes OFF when the origin has been determined. <br> Turned ON when power is turned ON. <br> Turned ON when operation starts. <br> OFF: Origin established. <br> ON: Origin not established. | -- | --- | Cleared | Refreshed each cycle during overseeing processes. | --- |
|  | A281.06 | Pulse Output 1 At-origin Flag | ON when the pulse output PV matches the origin (0). <br> OFF: Not stopped at origin. ON: Stopped at origin. | --- | --- | Cleared | Refreshed each cycle during overseeing processes. | --- |
|  | A281.07 | Pulse Output 1 Output Stopped Error Flag | ON when an error occurred while outputting pulses in the pulse output 1 origin search function. <br> The Pulse Output 1 Output Stop Error code will be written to A445. <br> OFF: No error <br> ON: Stop error occurred. | --- | --- | Cleared | Refreshed when origin search starts. <br> Refreshed when pulse output stop error occurs. | --- |


| Address |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related flags, settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Words | Bits |  |  |  |  |  |  |  |
| A283 | A283.00 | PWM Output 0 Output Inprogress Flag | ON when pulses are being output from PWM output 0. <br> Cleared when operation starts or stops. <br> OFF: Stopped <br> ON: Outputting pulses. | --- | --- | Cleared | Refreshed when pulse output starts or stops. | --- |
|  | A283.08 | PWM Output 1 Output Inprogress Flag | ON when pulses are being output from PWM output 1. <br> OFF: Stopped <br> ON: Outputting pulses. | -- | --- | Cleared |  | --- |
| A294 | All | Task Number when Program Stopped | This word contains the task number of the task that was being executed when program execution was stopped because of a program error. <br> (A298 and A299 contain the program address where program execution was stopped.) | Normal tasks: 0000 to 001F (task 0 to 31) Interrupt tasks: 8000 to 80FF (task 0 to 255) | Cleared | Cleared | When program error occurs. | $\begin{aligned} & \hline \text { A298/ } \\ & \text { A299 } \end{aligned}$ |
| A295 | A295.08 | Instruction Processing Error Flag | This flag and the Error Flag (ER) will be turned ON when an instruction processing error has occurred and the PLC Setup has been set to stop operation for an instruction error. CPU Unit operation will stop and the ERR/ALM indicator will light when this flag goes ON. <br> (The task number where the error occurred will be stored in A294 and the program address will be stored in A298 and A299.) | ON: Error Flag ON OFF: Error Flag OFF | Cleared | Cleared | When program error occurs. | A294, <br> A298/ <br> A299 <br> PLC <br> Setup <br> (Operation when instruction error has occurred) |
|  | A295.09 | Indirect DM BCD Error Flag | This flag and the Access Error Flag (AER) will be turned ON when an indirect DM BCD error has occurred and the PLC Setup has been set to stop operation an indirect DM BCD error. (This error occurs when the content of an indirectly addressed DM word is not BCD although BCD mode has been selected.) CPU Unit operation will stop and the ERR/ALM indicator will light when this flag goes ON. <br> (The task number where the error occurred will be stored in A294 and the program address will be stored in A298 and A299.) | ON: Not BCD OFF: Normal | Cleared | Cleared | When program error occurs. | A294, <br> A298/ <br> A299 <br> PLC Setup (Operation when instruction error has occurred) |
|  | A295.10 | Illegal Access Error Flag | This flag and the Access Error Flag (AER) will be turned ON when an illegal access error has occurred and the PLC Setup has been set to stop operation an illegal access error. (This error occurs when a region of memory is accessed illegally.) CPU Unit operation will stop and the ERR/ ALM indicator will light when this flag goes ON. <br> The following operations are considered illegal access: <br> 1) Reading/writing the system area <br> 2) Indirect DM BCD error (in BCD mode) <br> (The task number where the error occurred will be stored in A294 and the program address will be stored in A298 and A299.) | ON: Illegal access occurred OFF: Normal condition | Cleared | Cleared | When program error occurs. | A294, <br> A298/ <br> A299 <br> PLC <br> Setup (Operation when instruction error has occurred) |


| Address |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related flags, settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Words | Bits |  |  |  |  |  |  |  |
| A295 | A295.11 | No END Error Flag | ON when there isn't an END(001) instruction in each program within a task. <br> CPU Unit operation will stop and the ERR/ALM indicator will light when this flag goes ON. <br> (The task number where the error occurred will be stored in A294 and the program address will be stored in A298 and A299.) | ON: No END OFF: Normal condition | Cleared | Cleared | --- | $\begin{array}{\|l} \hline \text { A294, } \\ \text { A298/ } \\ \text { A299 } \end{array}$ |
|  | A295.12 | Task Error Flag | ON when a task error has occurred. The following conditions generate a task error. <br> There isn't even one regular task that is executable (started). <br> There isn't a program allocated to the task. <br> (The task number where the error occurred will be stored in A294 and the program address will be stored in A298 and A299.) | ON: Error OFF: Normal | Cleared | Cleared | --- | $\begin{aligned} & \text { A294, } \\ & \text { A298/ } \\ & \text { A299 } \end{aligned}$ |
|  | A295.13 | Differentiation Overflow Error Flag | The allowed value for Differentiation Flags which correspond to differentiation instructions has been exceeded. CPU Unit operation will stop and the ERR/ALM indicator will light when this flag goes ON. <br> (The task number where the error occurred will be stored in A294 and the program address will be stored in A298 and A299.) | ON: Error OFF: Normal | Cleared | Cleared | --- | $\begin{array}{\|l\|} \hline \text { A294, } \\ \text { A298/ } \\ \text { A299 } \end{array}$ |
|  | A295.14 | Illegal Instruction Error Flag | ON when a program that cannot be executed has been stored. CPU Unit operation will stop and the ERR/ALM indicator will light when this flag goes ON. | ON: Error OFF: Normal | Cleared | Cleared | --- | $\begin{array}{\|l\|} \hline \text { A294, } \\ \text { A298/ } \\ \text { A299 } \end{array}$ |
|  | A295.15 | UM Overflow Error Flag | ON when the last address in UM (User Memory) has been exceeded. CPU Unit operation will stop and the ERR/ALM indicator will light when this flag goes ON. | ON: Error OFF: Normal | Cleared | Cleared | --- | $\begin{aligned} & \text { A294, } \\ & \text { A298/ } \\ & \text { A299 } \end{aligned}$ |
| A298 | All | Program <br> Address <br> Where Program Stopped <br> (Rightmost 4 digits) | These words contain the 8-digit binary program address of the instruction where program execution was stopped due to a program error. <br> (A294 contains the task number of the task where program execution | Right 4 digits of the program address | Cleared | Cleared | --- | A294 |
| A299 | All | Program <br> Address <br> Where Program Stopped (Leftmost 4 digits) | was stopp | Left 4 digits of the program address | Cleared | Cleared | --- |  |
| A300 | All | Error Log Pointer | When an error occurs, the Error Log Pointer is incremented by 1 to indicate the location where the next error record will be recorded as an offset from the beginning of the Error Log Area (A100 to A199). <br> The Error Log Pointer can be cleared to 00 by turning A500.14 (the Error Log Reset Bit) ON. <br> When the Error Log Pointer has reached 14 hex ( 20 decimal), the next record is stored in A195 to A199 when the next error occurs. | 00 to 14 hexadecimal | Retained | Retained | Refreshed when error occurs. | A500.14 |


| Address |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related flags, settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Words | Bits |  |  |  |  |  |  |  |
| A310 | All | Manufacturing Lot Number, Lower Digits | The manufacturing lot number is stored in 6 digits hexadecimal. X, Y, and $Z$ in the lot number are converted to 10, 11, and 12, respectively. <br> Examples: <br> Lot number 01805 <br> $\mathrm{A} 310=0801, \mathrm{~A} 311=0005$ <br> Lot number 30Y05 <br> A310 $=1130, \mathrm{~A} 311=0005$ | --- | Retained | Retained | --- | --- |
| A311 | All | Manufacturing Lot Number, Upper Digits |  |  |  |  |  |  |
| A315 | A315.13 | Option Board Error Flag | ON when the Option Board is removed while the power is being supplied. CPU Unit operation will continue and the ERR/ALM indicator will flash. <br> OFF when the error has been cleared. | --- | Cleared | Cleared | Refreshed when error occurs. | $\begin{aligned} & \text { A402.00, } \\ & \text { A424 } \end{aligned}$ |
|  | A315.15 | Flash Memory Error Flag | ON when writing to the internal flash memory fails. CPU Unit operation will continue and the ERR/ALM indicator will flash. <br> OFF when the error has been cleared. | --- | Cleared | Cleared | Refreshed when error occurs. | A402.00 |
| $\begin{array}{\|l} \hline \text { A316 } \\ \text { to } \\ \text { A317 } \end{array}$ | All | High-speed Counter 2 PV | Contains the PV of high-speed counter 2. A317 contains the leftmost 4 digits and A316 contains the rightmost 4 digits. <br> The PV is cleared when operation starts. | --- | --- | Cleared | Refreshed each cycle during oversee process. Refreshed | --- |
| A318 to A319 | All | High-speed Counter 3 PV | Contains the PV of high-speed counter 3. A319 contains the leftmost 4 digits and A318 contains the rightmost 4 digits. <br> The PV is cleared when operation starts. | --- | --- | Cleared | when PRV(881) instruction is executed. | --- |
| A320 | A320.00 | High-speed Counter 2 <br> Range 1 Comparison Condition Met Flag | These flags indicate whether the PV is within the specified ranges when high-speed counter 2 is being operated in range-comparison mode. <br> Cleared at beginning of operation. <br> Cleared when range comparison table is registered. <br> OFF: PV not in range <br> ON: PV in range | --- | --- | Cleared | Refreshed each cycle during oversee process. Refreshed when PRV(881) instruction is executed. | --- |


| Address |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related flags, settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Words | Bits |  |  |  |  |  |  |  |
| A320 | A320.01 | High-speed Counter 2 <br> Range 2 Comparison Condition Met Flag | These flags indicate whether the PV is within the specified ranges when high-speed counter 2 is being operated in range-comparison mode. <br> Cleared at beginning of operation. <br> Cleared when range comparison table is registered. <br> OFF: PV not in range <br> ON: PV in range | --- | --- | Cleared | Refreshed each cycle during oversee process. <br> Refreshed when PRV(881) instruction is executed. | --- |
|  | A320.02 | High-speed Counter 2 Range 3 Comparison Condition Met Flag |  |  |  |  |  |  |
|  | A320.03 | High-speed Counter 2 <br> Range 4 Comparison Condition Met Flag |  |  |  |  |  |  |
|  | A320.04 | High-speed Counter 2 <br> Range 5 Comparison Condition Met Flag |  |  |  |  |  |  |
|  | A320.05 | High-speed Counter 2 <br> Range 6 Comparison Condition Met Flag |  |  |  |  |  |  |
|  | A320.06 | High-speed Counter 2 <br> Range 7 Comparison Condition Met Flag |  |  |  |  |  |  |
|  | A320.07 | High-speed Counter 2 Range 8 Comparison Condition Met Flag |  |  |  |  |  |  |
|  | A320.08 | High-speed Counter 2 Comparison In-progress Flag | This flag indicates whether a comparison operation is being executed for high-speed counter 2. <br> Cleared at beginning of operation. OFF: Stopped. <br> ON: Being executed. | -- | --- | Cleared | Refreshed when comparison operation starts or stops. | --- |
|  | A320.09 | High-speed Counter 2 Overflow/ Underflow Flag | This flag indicates when an overflow or underflow has occurred in the high-speed counter 2 PV. (Used with the linear mode counting range only.) Cleared when operation starts. Cleared when PV is changed. <br> OFF: Normal <br> ON: Overflow or underflow | --- | --- | Cleared | Refreshed when an overflow or underflow occurs. | --- |
|  | A320.10 | High-speed Counter 2 Count Direction | This flag indicates whether the highspeed counter is currently being incremented or decremented. The counter PV for the current cycle is compared with the PLC in last cycle to determine the direction. <br> OFF: Decrementing <br> ON: Incrementing | --- | --- | Cleared | Setting used for high-speed counter, valid during counter operation. | --- |


| Address |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related flags, settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Words | Bits |  |  |  |  |  |  |  |
| A321 | A321.00 | High-speed Counter 3 <br> Range 1 Comparison Condition Met Flag | These flags indicate whether the PV is within the specified ranges when high-speed counter 3 is being operated in range-comparison mode. <br> Cleared when operation starts. <br> Cleared when range comparison table is registered. <br> OFF: PV not in range <br> ON: PV in range | --- | --- | Cleared | Refreshed each cycle during overseeing process. <br> Refreshed when PRV(881) instruction is executed for the corresponding counter. | --- |
|  | A321.01 | High-speed Counter 3 Range 2 Comparison Condition Met Flag |  |  |  |  |  |  |
|  | A321.02 | High-speed Counter 3 <br> Range 3 Comparison Condition Met Flag |  |  |  |  |  |  |
|  | A321.03 | High-speed Counter 3 Range 4 Comparison Condition Met Flag |  |  |  |  |  |  |
|  | A321.04 | High-speed Counter 3 <br> Range 5 Comparison Condition Met Flag |  |  |  |  |  |  |
|  | A321.05 | High-speed Counter 3 Range 6 Comparison Condition Met Flag |  |  |  |  |  |  |
|  | A321.06 | High-speed Counter 3 Range 7 Comparison Condition Met Flag |  |  |  |  |  |  |
|  | A321.07 | High-speed Counter 3 Range 8 Comparison Condition Met Flag |  |  |  |  |  |  |
|  | A321.08 | High-speed Counter 3 Comparison In-progress Flag | This flag indicates whether a comparison operation is being executed for high-speed counter 3. <br> Cleared when operation starts. <br> OFF: Stopped. <br> ON: Being executed | --- | --- | Cleared | Refreshed when comparison operation starts or stops. | --- |
|  | A321.09 | High-speed Counter 3 Overflow/ Underflow Flag | This flag indicates when an overflow or underflow has occurred in the high-speed counter 3 PV. (Used with the linear mode counting range only.) Cleared when operation starts. Cleared when the PV is changed. <br> OFF: Normal <br> ON: Overflow or underflow | --- | --- | Cleared | Refreshed when an overflow or underflow occurs. | --- |
|  | A321.10 | High-speed Counter 3 Count Direction | This flag indicates whether the highspeed counter is currently being incremented or decremented. The counter PV for the current cycle is compared with the PC in last cycle to determine the direction. <br> OFF: Decrementing <br> ON: Incrementing | --- | --- | Cleared | Setting used for high-speed counter, valid during counter operation. | --- |
| A339 and A340 | All | Maximum Differentiation Flag Number | These words contain the maximum value of the differentiation flag numbers being used by differentiation instructions. | --- | See Function column. | Cleared | Written at the start of operation | A295.13 |


| Address |  | Name | Function | Settings | Status after mode change | Status <br> at startup | Write timing | Related flags, settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Words | Bits |  |  |  |  |  |  |  |
| A342 | A342.03 | Memory Cassette Write Flag | ON when data is being written to the Memory Cassette. | OFF: Not writing <br> ON: Writing | Retained | Cleared | --- | --- |
|  | A342.04 | Memory Cassette Read Flag | ON when data is being read from the Memory Cassette. | OFF: Not reading ON: Reading | Retained | Cleared | --- | --- |
|  | A342.05 | Memory Cassette Verify Flag | ON when data is being compared with data on the Memory Cassette. | OFF: Not verifying ON: Verifying | Retained | Cleared | --- | --- |
|  | A342.07 | Memory Cassette Initialization Error Flag | ON when an error occurs in initializing the Memory Cassette. <br> OFF the next time the Memory Cassette is accessed normally (initialized, written, read, or compared). | OFF: No error ON: Error | Retained | Cleared | -- | --- |
|  | A342.08 | Memory Cassette Write Error Flag | ON when an error occurs in writing the Memory Cassette. <br> OFF the next time the Memory Cassette is accessed normally (initialized, written, read, or compared). | OFF: No error ON: Error | Retained | Cleared | --- | --- |
|  | A342.10 | $\begin{aligned} & \text { Memory Cas- } \\ & \text { sette Read } \\ & \text { Error Flag } \end{aligned}$ | ON when an error occurs in reading the Memory Cassette. <br> OFF the next time the Memory Cassette is accessed normally (initialized, written, read, or compared). | OFF: No error ON: Error | Retained | Cleared | --- | --- |
|  | A342.12 | Memory Cassette Mismatch Flag | ON the data in the CPU Unit is not the same as the data in the Memory Cassette when a verification operation is performed. <br> OFF the next time the Memory Cassette is accessed normally (initialized, written, read, or compared). | OFF: Match ON: Mismatch | Retained | Cleared | --- | --- |
|  | A342.13 | Memory Cassette Access Flag | ON when the Memory Cassette is being accessed. <br> OFF when access is completed. | OFF: Not being accessed ON: Being accessed |  | Cleared | --- | --- |
|  | A342.15 | Memory Cassette Flag | ON when a Memory Cassette is mounted. <br> OFF when a Memory Cassette is not mounted. | OFF: No Memory Cassette <br> ON: Memory Cassette mounted | Retained | Cleared | --- | --- |
| A345 | A345.00 | FB Program Data Flag | Turns ON if the FB program memory contains FB program data. | OFF: No data ON: Data present | Retained | Cleared | Downloading programs from CX-Programmer or Memory Cassette or clearing VM | --- |
|  | A345.01 | Program Index File Flag | Turns ON when the comment memory contains a program index file. | OFF: No file ON: File present |  |  | Downloading programs from CX-Programmer or Memory Cassette |  |
|  | A345.02 | Comment File Flag | Turns ON when the comment memory contains a comment file. | OFF: No file ON: File present |  |  |  |  |
|  | A345.03 | Symbol Table File Flag | Turns ON when the comment memory contains a symbol table file. | OFF: No file ON: File present |  |  |  |  |
|  | A345.04 | DM Initial Values Flag | ON when DM initial values are stored in the flash memory. | OFF: No values stored ON: Values stored | --- | --- | --- | --- |


| Address |  | Name | Function | Settings | Status after mode change | Status <br> at startup | Write timing | Related flags, settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Words | Bits |  |  |  |  |  |  |  |
| $\begin{array}{\|l\|} \hline \text { A351 } \\ \text { to } \\ \text { A354 } \end{array}$ | All | Calendar/ Clock Area | These words contain the CPU Unit's internal clock data in BCD. The clock can be set from the CX-Programmer such as a Programming Console, with the DATE(735) instruction, or with a FINS command (CLOCK WRITE, 0702). | --- | Retained | Retained | Written every cycle | --- |
|  | $\begin{aligned} & \text { A351.00 } \\ & \text { to } \\ & \text { A351.07 } \end{aligned}$ |  | Seconds (00 to 59) (BCD) |  |  |  |  |  |
|  | $\begin{aligned} & \text { A351.08 } \\ & \text { to } \\ & \text { A351.15 } \end{aligned}$ |  | Minutes (00 to 59) (BCD) |  |  |  |  |  |
|  | $\begin{aligned} & \text { A352.00 } \\ & \text { to } \\ & \text { A352.07 } \end{aligned}$ |  | Hours (00 to 23) (BCD) |  |  |  |  |  |
|  | $\begin{aligned} & \text { A352.08 } \\ & \text { to } \\ & \text { A352.15 } \end{aligned}$ |  | Day of the month (01 to 31) (BCD) |  |  |  |  |  |
|  | $\begin{aligned} & \text { A353.00 } \\ & \text { to } \\ & \text { A353.07 } \end{aligned}$ |  | Month (01 to 12) (BCD) |  |  |  |  |  |
|  | $\begin{aligned} & \text { A353.08 } \\ & \text { to } \\ & \text { A353.15 } \end{aligned}$ |  | Year (00 to 99) (BCD) |  |  |  |  |  |
|  | $\begin{aligned} & \text { A354.00 } \\ & \text { to } \\ & \text { A354.07 } \end{aligned}$ |  | Day of the week (00 to 06) (BCD) 00: Sunday, 01: Monday, 02: Tuesday, <br> 03: Wednesday, 04: Thursday, <br> 05: Friday, 06: Saturday |  |  |  |  |  |
| $\begin{aligned} & \text { A360 } \\ & \text { to } \\ & \text { A391 } \end{aligned}$ | $\begin{aligned} & \text { A360.01 } \\ & \text { to } \\ & \text { A391.15 } \end{aligned}$ | Executed FAL Number Flags | The flag corresponding to the specified FAL number will be turned ON when $\operatorname{FAL}(006)$ is executed. Bits A360.01 to A391.15 correspond to FAL numbers 001 to 511 . <br> The flag will be turned OFF when the error is cleared. | ON: That FAL was executed OFF: That FAL wasn't executed | Retained | Cleared | Refreshed when error occurs. | A402.15 |


| Address |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related flags, settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Words | Bits |  |  |  |  |  |  |  |
| A392 | A392.04 | Serial Port 2 Error Flag (CP1L M-type CPU Units) <br> Serial Port 1 Error Flag (CP1L L-type CPU Units) | ON when an error has occurred at serial port 2 of a CP1L M-type CPU Unit. (Not valid in Peripheral Bus Mode or NT Link mode.) <br> ON when an error has occurred at serial port 1 of a CP1L L-type CPU Unit. (Not valid in Peripheral Bus Mode or NT Link mode.) | ON: Error OFF: No error | Retained | Cleared | Refreshed when error occurs. | --- |
|  | A392.05 | Serial Port 2 Send Ready Flag (No-protocol mode) (CP1L M-type CPU Units) <br> Serial Port 1 Send Ready Flag (No-protocol mode) (CP1L L-type CPU Units) | ON when the serial port 2 of a CP1L M-type CPU Unit is able to send data in no-protocol mode. <br> ON when the serial port 1 of a CP1L L-type CPU Unit is able to send data in no-protocol mode. | ON: Able-tosend OFF: Unable-to-send | Retained | Cleared | Written after transmission | --- |
|  | A392.06 | Serial Port 2 Reception Completed Flag (No-protocol mode) (CP1L M-type CPU Units) <br> Serial Port 1 Reception Completed Flag (No-protocol mode) (CP1L L-type CPU Units) | ON when the serial port 2 of a CP1L M-type CPU Unit has completed the reception in no-protocol mode. <br> - When the number of bytes was specified: ON when the specified number of bytes is received. <br> - When the end code was specified: ON when the end code is received or 256 bytes are received. <br> ON when the serial port 1 of a CP1L L-type CPU Unit has completed the reception in no-protocol mode. <br> - When the number of bytes was specified: ON when the specified number of bytes is received. <br> - When the end code was specified: ON when the end code is received or 256 bytes are received. | ON: Reception completed OFF: Reception not completed | Retained | Cleared | Written after reception | --- |
|  | A392.07 | Serial Port 2 Reception Overflow Flag (No-protocol mode) (CP1L M-type CPU Units) | ON when a data overflow occurred during reception through the serial port 2 of a CP1L M-type CPU Unit in no-protocol mode. <br> - When the number of bytes was specified: ON when more data is received after the reception was completed but before RXD (235) was executed. <br> - When the end code was specified: ON when more data is received after the end code was received but before RXD (235) was executed. <br> ON when 257 bytes are received before the end code. | ON: Overflow OFF: No overflow | Retained | Cleared | --- | --- |
|  |  | Serial Port 1 Reception Overflow Flag (No-protocol mode) (CP1L L-type CPU Units) | ON when a data overflow occurred during reception through the serial port 1 of a CP1L L-type CPU Unit in no-protocol mode. <br> - When the number of bytes was specified: ON when more data is received after the reception was completed but before RXD(235) was executed. <br> - When the end code was specified: ON when more data is received after the end code was received but before RXD(235) was executed. <br> ON when 257 bytes are received before the end code. |  |  |  |  |  |


| Address |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related flags, settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Words | Bits |  |  |  |  |  |  |  |
| A392 | A392.12 | Serial Port 1 Communications Error Flag (CP1L Mtype CPU Units) | - ON when a communications error has occurred at the serial port 1 of a CP1L M-type CPU Unit. (Not valid in Peripheral Bus Mode or NT Link mode.) <br> - ON when a timeout error, overrun error, framing error, parity error, or BCC error occurs in Serial Gateway mode. | ON: Error OFF: No error | Retained | Cleared | --- | --- |
|  | A392.13 | Serial Port 1 Send Ready Flag (No-protocol Mode) (CP1L M-type CPU Units) | ON when the serial port 1 of a CP1L M-type CPU Unit is able to send data in no-protocol mode. | ON: Able-tosend OFF: Unable-to-send | Retained | Cleared | Written after transmission | --- |
|  | A392.14 | Serial Port 1 Reception Completed Flag (No-protocol Mode) (CP1L M-type CPU Units) | ON when the serial port 1 of a CP1L M-type CPU Unit has completed the reception in no-protocol mode. <br> - When the number of bytes was specified: ON when the specified number of bytes is received. <br> - When the end code was specified: ON when the end code is received or 256 bytes are received. | ON: Reception completed OFF: Reception not completed | Retained | Cleared | Written after reception | --- |
|  | A392.15 | Serial Port 1 Reception Overflow Flag (No-protocol Mode) (CP1L M-type CPU Units) | ON when a data overflow occurred during reception through the serial port 1 of a CP1L M-type CPU Unit in no-protocol mode. <br> - When the number of bytes was specified: ON when more data is received after the reception was completed but before RXD(235) was executed. <br> - When the end code was specified: ON when more data is received after the end code was received but before RXD (235) was executed. <br> ON when 257 bytes are received before the end code. | ON: Overflow OFF: No overflow | Retained | Cleared | --- | --- |


| Address |  | Name | Function | Settings | Status after mode change | Status <br> at startup | Write timing | Related flags, settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Words | Bits |  |  |  |  |  |  |  |
| A393 | $\begin{aligned} & \text { A393.00 } \\ & \text { to } \\ & \text { A393.07 } \end{aligned}$ | Serial Port 2 <br> PT Communications Flags (CP1L M-type CPU Units) <br> Serial Port 1 PT Communications Flags (CP1L L-type CPU Units) | The corresponding bit will be ON when the serial port 2 of a CP1L Mtype CPU Unit is communicating with a PT in NT Link or Serial PLC Link mode. <br> Bits 0 to 7 correspond to units 0 to 7 . <br> The corresponding bit will be ON when the serial port 1 of a CP1L Ltype CPU Unit is communicating with a PT in NT Link or Serial PLC Link mode. <br> Bits 0 to 7 correspond to units 0 to 7 . | ON: Communicating OFF: Not communicating | Retained | Cleared | Refreshed when there is a normal response to the token. | --- |
|  | $\begin{aligned} & \text { A393.08 } \\ & \text { to } \\ & \text { A393.15 } \end{aligned}$ | Serial Port 2 <br> PT Priority <br> Registered <br> Flags (CP1L <br> M-type CPU <br> Units) <br> Serial Port 1 <br> PT Priority <br> Registered <br> Flags (CP1L <br> L-type CPU <br> Units) | The corresponding bit will be ON for the PT that has priority when the serial port 2 of a CP1L M-type CPU Unit is communicating in NT link mode. <br> Bits 0 to 7 correspond to units 0 to 7 . <br> These flags are written when the priority registration command is received. <br> The corresponding bit will be ON for the PT that has priority when the serial port 1 of a CP1L L-type CPU Unit is communicating in NT link mode. <br> Bits 0 to 7 correspond to units 0 to 7 . <br> These flags are written when the priority registration command is received. | ON: Priority registered OFF: Priority not registered | Retained | Cleared | See Function column. | --- |
|  | $\begin{aligned} & \text { A393.00 } \\ & \text { to } \\ & \text { A393.15 } \end{aligned}$ | Serial Port 2 <br> Reception <br> Counter (Noprotocol <br> Mode) (CP1L <br> M-type CPU <br> Units) <br> Serial Port 1 <br> Reception <br> Counter (Noprotocol <br> Mode) (CP1L <br> L-type CPU <br> Units) | Indicates (in binary) the number of bytes of data received when the serial port 2 of a CP1L M-type CPU Unit is in no-protocol mode. <br> Indicates (in binary) the number of bytes of data received when the serial port 1 of a CP1L L-type CPU Unit is in no-protocol mode. | --- | Retained | Cleared | Refreshed when data is received. | --- |
| A394 | $\begin{aligned} & \text { A394.00 } \\ & \text { to } \\ & \text { A394.07 } \end{aligned}$ | Serial Port 1 PT Communications Flags (CP1L M-type CPU Units) | The corresponding bit will be ON when the serial port 1 of a CP1L Mtype CPU Unit is communicating with a PT in NT link mode. <br> Bits 0 to 7 correspond to units 0 to 7 . | ON: Communicating OFF: Not communicating | Retained | Cleared | Refreshed when there is a normal response to the token. | --- |
|  | $\begin{aligned} & \text { A394.08 } \\ & \text { to } \\ & \text { A394.15 } \end{aligned}$ | Serial Port 1 <br> PT Priority <br> Registered <br> Flags (CP1L <br> M-type CPU <br> Units) | The corresponding bit will be ON for the PT that has priority when the serial port 1 of a CP1L M-type CPU Unit is communicating in NT link mode. <br> Bits 0 to 7 correspond to units 0 to 7 . | ON: Priority registered OFF: Priority not registered | Retained | Cleared | See Function column. | --- |
|  | $\begin{aligned} & \hline \begin{array}{l} \text { A394.00 } \\ \text { to } \\ \text { A394.15 } \end{array} \end{aligned}$ | Serial Port 1 <br> Reception Counter (Noprotocol Mode) (CP1L L-type CPU Units) | Indicates (in binary) the number of bytes of data received when serial port 1 of a CP1L L-type CPU Unit is in no-protocol mode. | --- | Retained | Cleared | Refreshed when data is received. | --- |
| A395 | A395.12 | DIP Switch Pin 6 Status Flag | The status of pin 6 on the DIP switch on the front of the CPU Unit is written to this flag every cycle. | ON: Pin 6 ON OFF: Pin 6 OFF | Retained | See Function column. | Written every cycle. | --- |


| Address |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related flags, settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Words | Bits |  |  |  |  |  |  |  |
| A400 | All | Error code | When a non-fatal error (user-defined FALS(006) or system error) or a fatal error (user-defined FALS(007) or system error) occurs, the 4-digit hexadecimal error code is written to this word. When two or more errors occur simultaneously, the highest error code will be recorded. | --- | Cleared | Cleared | Refreshed when error occurs. | --- |
| A401 | A401.00 | Other Fatal Error Flag | ON when a fatal error that is not defined for A401.01 to A401.15 occurs. Detailed information is output to the bits of A314. <br> There are no errors that affect this flag at this time. This flag is reserved by the system. | OFF: No other fatal error ON: Other fatal error | Cleared | Cleared | Refreshed when error occurs. | A314 |
|  | A401.06 | FALS Error Flag (fatal error) | ON when a fatal error is generated by the FALS(006) instruction. The CPU Unit will stop operating and the ERR/ALM indicator will light. <br> The corresponding error code will be written to A400. Error codes C101 to C2FF correspond to FALS numbers 001 to 511. <br> This flag will be turned OFF when the FALS errors are cleared. | ON: <br> FALS(006) <br> executed <br> OFF: <br> FALS(006) not executed | Cleared | Cleared | Refreshed when error occurs. | A400 |
|  | A401.08 | Cycle Time Too Long Flag (fatal error) | ON if the cycle time exceeds the maximum cycle time set in the PLC Setup (the cycle time monitoring time). CPU Unit operation will stop and the ERR/ALM indicator on the front of the CPU Unit will light. <br> This flag will be turned OFF when the error is cleared. | OFF: Cycle time under max. <br> ON: Cycle time over max. | Cleared | Cleared | Refreshed when the cycle time exceeds maximum. | PLC <br> Setup (Cycle time monitoring time) |
|  | A401.09 | Program Error Flag (fatal error) | ON when program contents are incorrect. <br> CPU Unit operation will stop and the ERR/ALM indicator on the front of the CPU Unit will light. The task number where the error occurred will be stored in A294 and the program address will be stored in A298 and A299. <br> The type of program error that occurred will be stored in A295.08 to A295.15. Refer to the description of A295 for more details on program errors. <br> This flag will be turned OFF when the error is cleared. | ON: Error OFF: No error | Cleared | Cleared | Refreshed when error occurs. | $\begin{aligned} & \text { A294, } \\ & \text { A295, } \\ & \text { A298 and } \\ & \text { A299 } \end{aligned}$ |


| Address |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related flags, settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Words | Bits |  |  |  |  |  |  |  |
| A401 | A401.11 | Too Many I/O Points Flag (fatal error) | ON when the number of Expansion Units and Expansion I/O Units exceeds the limit, when the number of words allocated to these Units exceeds the limit, are mounted. <br> CPU Unit operation will stop and the ERR/ALM indicator on the front of the CPU Unit will light. <br> This flag will be turned OFF when the error is cleared. | ON: Error OFF: No error | Cleared | Cleared | Refreshed when error occurs. | A407 |
|  | A401.14 | I/O Bus Error Flag (fatal error) | ON in the following cases: <br> - When an error occurs in a data transfer between the CPU Unit and a Expansion Unit or Expansion I/O Unit. If this happens, OAOA hex will be output to A404. <br> CPU Unit operation will stop and the ERR/ALM indicator on the front of the CPU Unit will light. <br> (This flag will be turned OFF when the error is cleared.) | ON: Error OFF: No error | Cleared | Cleared | Refreshed when error occurs. | A404 |
|  | A401.15 | Memory Error Flag (fatal error) | ON when an error occurred in memory or there was an error in automatic transfer from the Memory Cassette when the power was turned ON. <br> CPU Unit operation will stop and the ERR/ALM indicator on the front of the CPU Unit will light. <br> The location where the error occurred is indicated in A403.00 to A403.08, and A403.09 will be turned ON if there was an error during automatic transfer at startup. <br> This flag will be turned OFF when the error is cleared. (The automatic transfer at startup error cannot be cleared without turning OFF the PLC.) | ON: Error OFF: No error | Cleared | Cleared | Refreshed when error occurs. | $\begin{aligned} & \text { A403.00 } \\ & \text { to } \\ & \text { A403.08, } \\ & \text { A403.09 } \end{aligned}$ |
| A402 | A402.00 | Other Fatal Error Flag | ON when a non-fatal error that is not defined for A402.01 to A402.15 occurs. Detailed information is output to the bits of A314. <br> There are no errors that affect this flag at this time. This flag is reserved by the system. | OFF: No other fatal error ON: Other fatal error | Cleared | Cleared | Refreshed when error occurs. | A315 |
|  | A402.04 | Battery Error Flag (non-fatal error) | ON if the CPU Unit's battery is disconnected or its voltage is low and the Detect Battery Error setting has been set in the PLC Setup. <br> The CPU Unit will continue operating and the ERR/ALM indicator on the front of the CPU Unit will flash. <br> This flag can be used to control an external warning light or other indicator to indicate that the battery needs to be replaced. <br> (This flag will be turned OFF when the error is cleared.) | ON: Error OFF: No error | Cleared | Cleared | Refreshed when error occurs. | PLC <br> Setup (Detect Battery Error) |
|  | A402.10 | PLC Setup Error Flag (non-fatal error) | ON when there is a setting error in the PLC Setup. The CPU Unit will continue operating and the ERR/ ALM indicator on the front of the CPU Unit will flash. The location of the error will be written to A406. <br> (This flag will be turned OFF when the error is cleared.) | ON: Error OFF: No error | Cleared | Cleared | Refreshed when error occurs. | A406 |


| Address |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related flags, settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Words | Bits |  |  |  |  |  |  |  |
| A402 | A402.15 | FAL Error Flag (non-fatal error) | ON when a non-fatal error is generated by executing $\operatorname{FAL}(006)$. The CPU Únit will continue operating and the ERR/ALM indicator on the front of the CPU Unit will flash. <br> The bit in A360 to A391 that corresponds to the FAL number specified in FALS(006) will be turned ON and the corresponding error code will be written to A400. Error codes 4101 to 42FF correspond to FAL numbers 001 to 2FF (0 to 511). <br> (This flag will be turned OFF when the error is cleared.) | ON: <br> FALS(006) error occurred OFF: FALS(006) not executed | Cleared | Cleared | Refreshed when error occurs. | A360 to A391, <br> A400 |
| A403 | $\begin{aligned} & \text { A403.00 } \\ & \text { to } \\ & \text { A403.08 } \end{aligned}$ | Memory Error Location | When a memory error occurs, the Memory Error Flag (A40115) is turned ON and one of the following flags is turned ON to indicate the memory area where the error occurred <br> A403.00: User program <br> A403.04: PLC Setup <br> A403.07: Routing Table <br> When a memory error occurs, the CPU Unit will continue operating and the ERR/ALM indicator on the front of the CPU Unit will flash. <br> (The corresponding flag will be turned OFF when the error is cleared.) | ON: Error OFF: No error | Cleared | Cleared | Refreshed when error occurs. | A401.15 |
|  | A403.09 | Memory Cassette startup Transfer Error Flag | ON when automatic transfer at startup has been selected and an error occurs during automatic transfer. An error will occur if there is a transfer error, the specified file does not exist, or the Memory Cassette is not installed. <br> (This flag will be turned OFF when the error is cleared by turning the power OFF. The error cannot be cleared without turning the power OFF.) | ON: Error OFF: No error | Cleared | Cleared | Refreshed when power is turned ON. | --- |
|  | A403.10 | Flash Memory Error Flag | ON when the flash memory is physically destroyed. | ON: Error OFF: No error | Cleared | Cleared | Refreshed when error is detected. | --- |
| A404 | All | I/O Bus Error Details | Contains information on I/O bus errors. <br> The CPU Unit will stop operating and the ERR/ALM indicator on the front of the CPU Unit will light. <br> (A401.04 (I/O Bus Error Flag) will turn ON.) <br> (This information will be cleared when the error is cleared.) | OAOA hex: CPM1A Unit error <br> 0000 hex: CJseries Unit error, 1st Unit 0001 hex: CJseries Unit error, 2nd Unit OFOF hex: CJseries Unit error, unknown Unit OEOE hex: CJ-series Unit error, no End cover | Cleared | Cleared | Refreshed when error is detected. | A401.14 |


| Address |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related flags, settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Words | Bits |  |  |  |  |  |  |  |
| A406 | All | PLC Setup Error Location | When there is a setting error in the PLC Setup, the location of that error is written to A406 in 4-digit hexadecimal. <br> The CPU Unit will continue operating and the ERR/ALM indicator on the front of the CPU Unit will flash. <br> (A406 will be cleared when the cause of the error is eliminated.) | 0000 to 01FF hexadecimal | Cleared | Cleared | Refreshed when error occurs. | A402.10 |
| A407 | $\begin{aligned} & \text { A407.00 } \\ & \text { to } \\ & \text { A407.12 } \end{aligned}$ | Too Many I/O Points, Details | Always 0000 hex. | 0000 hex | Cleared | Cleared | --- | $\begin{aligned} & \text { A401.11, } \\ & \text { A407.13 } \\ & \text { to } \\ & \text { A407.15 } \end{aligned}$ |
|  | $\begin{aligned} & \text { A407.13 } \\ & \text { to } \\ & \text { A407.15 } \end{aligned}$ | Too Many I/O Points, Cause | The 3-digit binary value of these bits indicates the cause of the Too Many I/O Points Error. <br> Note These bits will be cleared when the error is cleared. | 010: Too many Expansion Unit and Expansion I/O Unit words 011: Too many Expansion Units and Expansion I/O Units | Cleared | Cleared | Refreshed when error occurs. | --- |
| A424 | $\begin{aligned} & \text { A424.00 } \\ & \text { to } \\ & \text { A424.15 } \end{aligned}$ | Error Option Board Flags | The bit corresponding to the option slot turns ON when an error occurs in an Option Board (A315.13 will be ON). <br> Bit 00: Option slot 1 <br> Bit 01: Option slot 2 | ON: Error OFF: No error | Cleared | Cleared | --- | A353.13 |
| A436 | $\begin{aligned} & \text { A436.00 } \\ & \text { to } \\ & \text { A436.02 } \end{aligned}$ | CPM1A Unit Error Flags | ON when an error occurs in a CPseries Expansion Unit or Expansion I/O Unit. <br> A436.00: 1st Unit <br> A436.10: 2nd Unit <br> A436.02: 3rd Unit | OFF: No error ON: Error | Retained | Cleared | --- | --- |
| A437 | All | Number of Connected CPM1A Units | Stores the number of Expansion Units and Expansion I/O Units connected as a hexadecimal number. <br> Note This information is valid only when a Too Many I/O Points error has occurred. CP1W/ CPM1A-TS002 and CP1W/ CPM1A-TS102 are each counted as two Units. | $\begin{aligned} & 0000 \text { to } 0007 \\ & \text { hex } \end{aligned}$ | Retained | Cleared | --- | --- |
| A438 | All | Pulse Output 2 Stop Error Code | If a Pulse Output Stop Error occurs for pulse output 2, the error code is stored. | --- | Retained | Cleared | Refreshed when origin search starts. <br> Refreshed when a pulse output stop error occurs. | --- |
| A439 | All | Pulse Output 3 Stop Error Code | If a Pulse Output Stop Error occurs for pulse output 3, the error code is stored. | --- | Retained | Cleared | Refreshed when origin search starts. <br> Refreshed when a pulse output stop error occurs. | --- |
| A440 | All | Max. Interrupt Task Processing Time | Contains the Maximum Interrupt Task Processing Time in units of 0.1 ms . <br> (This value is written after the interrupt task with the max. processing time is executed and cleared when PLC operation begins.) | 0000 to FFFF hexadecimal | Cleared | Cleared | See Function column. | --- |


| Address |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related flags, settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Words | Bits |  |  |  |  |  |  |  |
| A441 | All | Interrupt Task With Max. <br> Processing Time | Contains the task number of the interrupt task with the maximum processing time. Hexadecimal values 8000 to 80FF correspond to task numbers 00 to FF. Bit 15 is turned ON when an interrupt has occurred. <br> (This value is written after the interrupt task with the max. processing time is executed and cleared when PLC operation begins.) | 8000 to $80 F F$ hexadecimal | Cleared | Cleared | See Function column. | --- |
| A444 | All | Pulse Output 0 Stop Error Code | If a Pulse Output Stop Error occurs for pulse output 0 , the error code is written to this word. | --- | --- | Cleared | Refreshed when origin search | --- |
| A445 |  | Pulse Output 1 Stop Error Code | If a Pulse Output Stop Error occurs for pulse output 1, the error code is written to this word. | --- | --- |  | starts. <br> Refreshed when a pulse output stop error occurs. | --- |
| A494 | $\begin{aligned} & \text { A494.00 } \\ & \text { to } \\ & \text { A494.07 } \end{aligned}$ | Memory Casette Verification Results | Stores the results of comparing data in the Memory Cassette and CPU Unit. This information is cleared the next time the Memory Cassette is accessed normally (initialized, written, read, or compared). <br> A494.00: User program is different. A494.01: Function block sources are different. <br> A494.02: Parameter area is different. <br> A494.03: Symbol table is different. <br> A494.04: Comments are different. <br> A494.05: Program indices are different. <br> A494.06: Data memory is different. <br> A494.07: DM initial values are different. | OFF: Match ON: Mismatch |  |  | When Memory Cassette is compared. |  |

Read/Write Area (Set by User)

| Addresses |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related Flags, Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bits |  |  |  |  |  |  |  |
| A500 | A500.12 | IOM Hold Bit | Turn this bit ON to preserve the status of the I/O Memory when shifting from PROGRAM to RUN or MONITOR mode or vice versa. The I/O Memory includes the CIO Area, Transition Flags, Timer Flags and PVs, Index Registers, and Data Registers. <br> (If the status of the IOM Hold Bit itself is preserved in the PLC Setup (IOM Hold Bit Status), the status of the I/O Memory Area will be retained when the PLC is turned ON or power is interrupted.) | ON: Retained OFF: Not retained | Retained | See Function column. | See Function column. | PLC Setup (IOM Hold Bit Status setting) |
|  | A500.13 | Forced Status Hold Bit | Turn this bit ON to preserve the status of bits that have been force-set or force-reset when shifting from PROGRAM to MONITOR mode or vice versa. Bits that have been force-set or force-reset will always return to their default status when shifting to RUN mode. <br> (If the status of the Forced Status Hold Bit itself is preserved in the PLC Setup (Forced Status Hold Bit Status), the status of force-set and force-reset bits will be retained when the PLC is turned ON or power is interrupted.) | ON: Retained OFF: Not retained | Retained | See Function column. | See Function column. | PLC Setup (Forced Status Hold Bit Status setting) |
|  | A500.14 | Error Log Reset Bit | Turn this bit ON to reset the Error Log Pointer (A300) to 00. <br> The contents of the Error Log Area itself (A100 to A199) are not cleared. <br> (This bit is automatically reset to 0 after the Error Log Pointer is reset.) | OFF to ON: Clear | Retained | Cleared | --- | A100 to A199, A300 |
|  | A500.15 | Output OFF Bit | Turn this bit ON to turn OFF all outputs from the CPU Unit, CPM1A Units, and Special I/O Units. The INH indicator on the front of the CPU Unit will light while this bit is ON. <br> (The status of the Output OFF Bit is retained through power interruptions.) | --- | Retained | Retained | --- | --- |


| Addresses |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related Flags, Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bits |  |  |  |  |  |  |  |
| A508 | A508.09 | Differentiate Monitor Completed Flag | ON when the differentiate monitor condition has been established during execution of differentiation monitoring. <br> (This flag will be cleared to 0 when differentiation monitoring starts.) | ON: Monitor condition established OFF: Not yet established | Retained | Cleared | --- | --- |
|  | A508.11 | Trace Trigger Monitor Flag | ON when a trigger condition is established by the Trace Start Bit (A508.14). OFF when the next Data Trace is started by the Sampling Start bit (A508.15). | ON: Trigger condition established OFF: Not yet established or not tracing | Retained | Cleared | --- | --- |
|  | A508.12 | Trace Completed Flag | ON when sampling of a region of trace memory has been completed during execution of a Trace. <br> OFF when the next time the Sampling Start Bit (A508.15) is turned ON. | ON: Trace completed OFF: Not tracing or trace in progress | Retained | Cleared | --- | --- |
|  | A508.13 | Trace Busy Flag | ON when the Sampling Start Bit (A508.15) is turned ON. OFF when the trace is completed. | ON: Trace in progress OFF: Not tracing (not sampling) | Retained | Cleared | --- | --- |
|  | A508.14 | Trace Start Bit | Turn this bit ON to establish the trigger condition. The offset indicated by the delay value (positive or negative) determines which data samples are valid. | ON: Trace trigger condition established OFF: Not established | Retained | Cleared | --- | --- |
|  | A508.15 | Sampling Start Bit | When a data trace is started by turning this bit ON from the CX-Programmer, the PLC will begin storing data in Trace Memory by one of the three following methods: <br> 1) Data is sampled at regular intervals ( 10 to $2,550 \mathrm{~ms}$ ). <br> 2) Data is sampled when $\operatorname{TRSM}(045)$ is executed in the program. <br> 3) Data is sampled at the end of every cycle. <br> The operation of A508.15 can be controlled only from the CX-Programmer. | OFF to ON: Starts data trace (sampling) <br> Turned ON from Programming Device. | Retained | Cleared | --- | --- |
| A510 to A511 | All | startup Time | These words contain the time at which the power was turned ON. The contents are updated every time that the power is turned ON. The data is stored in BCD. <br> A510.00 to A510.07: Second (00 to 59) <br> A510.08 to A510.15: Minute (00 to 59) <br> A511.00 to A511.07: Hour (00 to 23) <br> A511.08 to A511.15: Day of month (01 to 31) | See Function column. | Retained | See Function column. | Refreshed when power is turned ON. | --- |
| $\begin{aligned} & \text { A512 to } \\ & \text { A513 } \end{aligned}$ | All | Power Interruption Time | These words contain the time at which the power was interrupted. The contents are updated every time that the power is interrupted. The data is stored in BCD. <br> A512.00 to A512.07: Second (00 to 59) <br> A512.08 to A512.15: Minute (00 to 59) <br> A513.00 to A513.07: Hour (00 to 23) <br> A513.08 to A513.15: Day of month (01 to 31) <br> (These words are not cleared at startup.) | See Function column. | Retained | Retained | Written at power interruption | --- |


| Addresses |  | Name | Function | Settings | Status after mode change | Statusat startup | Write timing | Related Flags, Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bits |  |  |  |  |  |  |  |
| A514 | All | Number of Power Interruptions | Contains the number of times that power has been interrupted since the power was first turned ON. The data is stored in binary. To reset this value, overwrite the current value with 0000. <br> (This word is not cleared at startup, but it is cleared when the Memory Corruption Detected Flag (A395.11) goes ON.) | 0000 to FFFF hexadecimal | Retained | Retained | Refreshed when power is turned ON. | A395.11 |
| $\begin{array}{\|l} \text { A515 to } \\ \text { A517 } \end{array}$ | All | Operation Start Time | The time that operation started as a result of changing the operating mode to RUN or MONITOR mode is stored here in BCD. <br> A515.00 to A515.07: Seconds (00 to 59) <br> A515.08 to A515.15: Minutes (00 to 59) <br> A516.00 to A516.07: Hour (00 to 23) A516.08 to A516.15: Day of month (01 to 31) <br> A517.00 to A517.07: Month (01 to 12) <br> A517.08 to A517.15: Year (00 to 99) <br> Note The previous start time is stored after turning ON the power supply until operation is started. | See at left. | Retained | Retained | See at left. | --- |
| $\begin{aligned} & \text { A518 to } \\ & \text { A520 } \end{aligned}$ | All | Operation End Time | The time that operation stopped as a result of changing the operating mode to PROGRAM mode is stored here in BCD. <br> A518.00 to A518.07: Seconds (00 to 59) <br> A518.08 to A518.15: Minutes (01 to 59) <br> A519.00 to A519.07: Hour (00 to 23) A519.08 to A519.15: Day of month (01 to 31) <br> A520.00 to A520.07: Month (01 to 12) <br> A520.08 to A520.15: Year (00 to 99) <br> Note If an error occurs in operation, the time of the error will be stored. If the operating mode is then changed to PROGRAM mode, the time that PROGRAM mode was entered will be stored. | See at left. | Retained | Retained | See at left. | --- |
| A523 | All | Total Power ON Time | Contains the total time that the PLC has been ON in 10-hour units. The data is stored in binary and it is updated every 10 hours. To reset this value, overwrite the current value with 0000 . <br> (This word is not cleared at startup, but it is cleared to 0000 when the Memory Corruption Detected Flag (A395.11) goes ON.) | 0000 to FFFF hexadecimal | Retained | Retained | --- | --- |


| Addresses |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related Flags, Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bits |  |  |  |  |  |  |  |
| A525 | A525.00 | Ethernet Option Board Reset Flag (Serial Port 2) | Turn this bit ON to reset the Ethernet Option Board mounted on the serial port 2. | OFF to ON: Reset | Retained | Cleared | --- | --- |
|  | A525.01 | Ethernet Option Board Reset Flag (Serial Port 1) | Turn this bit ON to reset the Ethernet Option Board mounted on the serial port 1. | OFF to ON: Reset | Retained | Cleared | --- | --- |
|  | A525.08 | Ethernet <br> Option <br> Board Restart Flag (Serial Port 2) | Turn this bit ON to restart the Ethernet Option Board mounted on the serial port 2. | OFF to ON: Restart | Retained | Cleared | --- | --- |
|  | A525.09 | Ethernet <br> Option <br> Board <br> Restart Flag (Serial Port 1) | Turn this bit ON to restart Ethernet Option Board mounted on the serial port 1. | OFF to ON: Restart | Retained | Cleared | --- | --- |
| A526 | A526.00 | Serial Port 2 Restart Bit (CP1L Mtype CPU Units) | Turn this bit ON to restart the serial port 2 of a CP1L M-type CPU Unit. (Do not use this bit when the port is operating in Peripheral Bus Mode.) <br> Note This bit is turned OFF automatically when the restart processing is completed. | OFF to ON: Restart | Retained | Cleared | --- | --- |
|  |  | Serial Port 1 Restart Bit (CP1L Ltype CPU Units) | Turn this bit ON to restart the serial port 1 of a CP1L L-type CPU Unit. (Do not use this bit when the port is operating in Peripheral Bus Mode.) <br> Note This bit is turned OFF automatically when the restart processing is completed. |  |  |  |  |  |
|  | A526.01 | Serial Port 1 Restart Bit (CP1L Mtype CPU Units) | Turn this bit ON to restart the serial port 1 of a CP1L M-type CPU Unit. <br> Note This bit is turned OFF automatically when the restart processing is completed. | 0 to ON: Restart | Retained | Cleared | --- | --- |
| A527 | $\begin{aligned} & \hline \text { A527.00 } \\ & \text { to } \\ & \text { A527.07 } \end{aligned}$ | Online Editing Disable Bit Validator | The Online Editing Disable Bit (A527.09) is valid only when this byte contains 5A. <br> To disable online editing from the CX-Programmer, set this byte to 5A and turn ON A527.09. <br> (Online editing refers to changing or adding to the program while the PLC is operating in MONITOR mode.) | 5A: <br> A527.09 enabled <br> Other value: A527.09 disabled | Retained | Cleared | --- | A527.09 |
|  | A527.09 | Online Editing Disable Bit | Turn this bit ON to disable online editing. The setting of this bit is valid only when A527.00 to A527.07 have been set to 5A. | ON: Disabled OFF: Not disabled | Retained | Cleared | --- | $\begin{aligned} & \text { A527.00 } \\ & \text { to } \\ & \text { A527.07 } \end{aligned}$ |


| Addresses |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related Flags, Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bits |  |  |  |  |  |  |  |
| A528 | $\begin{aligned} & \hline \text { A528.00 } \\ & \text { to } \\ & \text { A528.07 } \end{aligned}$ | Serial Port 2 Error Flags (CP1L Mtype CPU Units) | These flags indicate what kind of error has occurred at the serial port 2 of a CP1L M-type CPU Unit; they are automatically turned OFF when the serial port 2 is restarted. <br> (These flags are not valid in peripheral bus mode and only bit 5 is valid in NT Link mode.) <br> PLC Link Polling Unit: <br> Bit 05: ON for timeout error. <br> PLC Link Polled Unit: <br> Bit 03: ON for framing error. <br> Bit 04: ON for overrun error. <br> Bit 05: ON for timeout error. <br> These bits can be cleared by the CXProgrammer. <br> These flags indicate what kind of error has occurred at the serial port 1 of a CP1L L-type CPU Unit; they are automatically turned OFF when the serial port 1 is restarted. <br> (These flags are not valid in peripheral bus mode and only bit 5 is valid in NT Link mode.) <br> PLC Link Polling Unit: <br> Bit 05: ON for timeout error. <br> PLC Link Polled Unit: <br> Bit 03: ON for framing error. <br> Bit 04: ON for overrun error. <br> Bit 05: ON for timeout error. <br> These bits can be cleared by the CXProgrammer. | Bits 00 and 01: Not used. <br> Bit 02: ON for parity error. <br> Bit 03: ON for framing error. <br> Bit 04: ON for overrun error. <br> Bit 05: ON for timeout error. <br> Bits 06 and 07: Not used. | Retained | Cleared | --- | --- |
|  | $\begin{array}{\|l} \text { A528.08 } \\ \text { to } \\ \text { A528.15 } \end{array}$ | Serial Port 1 Error Code (CP1L Mtype CPU Units) | These flags indicate what kind of error has occurred at the serial port 1 of a CP1L M-type CPU Unit; they are automatically turned OFF when the serial port 1 is restarted. <br> (These flags are not valid in peripheral bus mode and only bit 5 is valid in NT Link mode.) <br> PLC Link Polling Unit: <br> Bit 13: ON for timeout error. <br> PLC Link Polled Unit: <br> Bit 11: ON for framing error. <br> Bit 12: ON for overrun error. <br> Bit 13: ON for timeout error. <br> These bits can be cleared by the CXProgrammer. | Bits 08 and 09: Not used. Bit 10: ON for parity error. Bit 11: ON for framing error. Bit 12: ON for overrun error. Bit 13: ON for timeout error. Bits 14 and 15: Not used. | Retained | Cleared | --- | --- |
| A529 | All | FAL/FALS Number for System Error Simulation | Set a dummy FAL/FALS number to use to simulate the system error using FAL(006) or FALS(007). <br> When FAL(006) or FALS(007) is executed and the number in A529 is the same as the one specified in the operand of the instruction, the system error given in the operand of the instruction will be generated instead of a user-defined error. | $\begin{aligned} & \hline 0001 \text { to 01FF } \\ & \text { hex: FAL/FALS } \\ & \text { numbers } 1 \text { to } \\ & 511 \\ & 0000 \text { or } 0200 \\ & \text { to FFFF hex: } \\ & \text { No FAL/FALS } \\ & \text { number for sys- } \\ & \text { tem error simu- } \\ & \text { lation. (No error } \\ & \text { will be gener- } \\ & \text { ated.) } \end{aligned}$ | Retained | Cleared | --- | --- |
| A531 | A531.00 | High-speed Counter 0 Reset Bit | When the reset method is set to Phase-Z signal + Software reset, the corresponding high-speed counter's | --- | Retained | Cleared | --- | --- |
|  | A531.01 | High-speed Counter 1 Reset Bit | PV will be reset if the phase- $Z$ signal is received while this bit is ON . <br> When the reset method is set to Software reset, the corresponding highspeed counter's PV will be reset in the cycle when this bit turns ON. | --- | Retained | Cleared | --- | --- |


| Addresses |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related Flags, Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bits |  |  |  |  |  |  |  |
| A531 | A531.08 | High-speed Counter 0 Gate Bit | When a counter's Gate Bit is ON, the counter's PV will not be changed even if pulse inputs are received for the counter. <br> When the bit is turned OFF again, counting will restart and the highspeed counter's PV will be refreshed. When the reset method is set to Phase-Z signal + Software reset, the Gate Bit is disabled while the corresponding Reset Bit (A531.00 or A531.01) is ON. | --- | Retained | Cleared | --- | --- |
|  | A531.09 | High-speed Counter 1 Gate Bit |  | --- | Retained | Cleared | --- | --- |
| A532 | All | Interrupt Counter 0 Counter SV | Used for interrupt input 0 in counter mode. <br> Sets the count value at which the interrupt task will start. Interrupt task 140 will start when interrupt counter 0 has counted this number of pulses. Retained when operation starts. | --- | Retained | Retained | --- | --- |
| A533 | All | Interrupt Counter 1 Counter SV | Used for interrupt input 1 in counter mode. <br> Sets the count value at which the interrupt task will start. Interrupt task 141 will start when interrupt counter 1 has counted this number of pulses. | --- | Retained | Retained | --- | --- |
| A534 | All | Interrupt Counter 2 Counter SV | Used for interrupt input 2 in counter mode. <br> Sets the count value at which the interrupt task will start. Interrupt task 142 will start when interrupt counter 2 has counted this number of pulses. | --- | Retained | Retained | --- | --- |
| A535 | All | Interrupt Counter 3 Counter SV | Used for interrupt input 3 in counter mode. <br> Sets the count value at which the interrupt task will start. Interrupt task 143 will start when interrupt counter 3 has counted this number of pulses. | --- | Retained | Retained | --- | --- |
| A536 | All | Interrupt Counter 0 Counter PV | These words contain the interrupt counter PVs for interrupt inputs operating in counter mode. | --- | --- | Retained | Refreshed when interrupt | --- |
| A537 | All | Interrupt Counter 1 Counter PV | In increment mode, the counter PV starts incrementing from 0 . When the counter PV reaches the counter SV, | --- | --- |  | is generated. <br> Refresh- | --- |
| A538 | All | Interrupt Counter 2 Counter PV | In decrement mode, the counter PV starts decrementing from the counter SV. When the counter PV reaches | --- | --- |  | $\mathrm{INI}(880)$ instruction is | --- |
| A539 | All | Interrupt Counter 3 Counter PV | the 0 , the PV is automatically reset to the SV. <br> Cleared when operation starts. | --- | --- |  | executed. | --- |
| A540 | A540.00 | Pulse Output 0 Reset Bit | The pulse output 0 PV (contained in A276 and A277) will be cleared when this bit is turned ON. | --- | Retained | Cleared | --- | $\begin{aligned} & \hline \text { A276 and } \\ & \text { A277 } \end{aligned}$ |
|  | A540.08 | Pulse Output 0 CW Limit Input Signal Flag | This is the CW limit input signal for pulse output 0, which is used in the origin search. To use this signal, write the input from the actual sensor as an input condition in the ladder program and output the result to this flag. | --- | Retained | Cleared | --- | --- |
|  | A540.09 | Pulse Output 0 CCW Limit Input Signal Flag | This is the CCW limit input signal for pulse output 0 , which is used in the origin search. To use this signal, write the input from the actual sensor as an input condition in the ladder program and output the result to this flag. | --- | Retained | Cleared | --- | --- |


| Addresses |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related Flags, Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bits |  |  |  |  |  |  |  |
| A540 | A540.10 | Pulse Output 0 Positioning Completed Signal | This is the positioning completed input signal used in the origin search for pulse output 0 . The input signal from the servo driver is output to this bit from the ladder program to enable using the signal. | --- | Retained | Cleared |  | -- |
| A541 | A541.00 | Pulse Output 1 Reset Bit | The pulse output 1 PV (contained in A278 and A279) will be cleared when this bit is turned ON. | --- | Retained | Cleared | --- | $\begin{aligned} & \text { A278 and } \\ & \text { A279 } \end{aligned}$ |
|  | A541.08 | Pulse Output 1 CW Limit Input Signal Flag | This is the CW limit input signal for pulse output 1 , which is used in the origin search. To use this signal, write the input from the actual sensor as an input condition in the ladder program and output the result to this flag. | --- | Retained | Cleared | --- | --- |
|  | A541.09 | Pulse Output 1 CCW Limit Input Signal Flag | This is the CCW limit input signal for pulse output 1, which is used in the origin search. To use this signal, write the input from the actual sensor as an input condition in the ladder program and output the result to this flag. | --- | Retained | Cleared | --- | --- |
|  | A541.10 | Pulse Output 1 Positioning Completed Signal | This is the positioning completed input signal used in the origin search for pulse output 1. The input signal from the servo driver is output to this bit from the ladder program to enable using the signal. | --- | Retained | Cleared | --- | --- |
| A562 | A562.00 | Error Counter 0 Reset Bit | Turn ON this bit to reset the Error Counter 0 Present Value and turn OFF the Error Counter 0 Error Flag. | --- | --- | Cleared | --- | --- |
|  | A562.01 | Error Counter 0 Disable Bit | Turn ON this bit to hold the present value of error counter 0 . | ON: Error counter value held. <br> OFF: Error counter value not held. | --- | Cleared | --- | --- |
| A563 | A563.00 | Error Counter 1 Reset Bit | Turn ON this bit to reset the Error Counter 1 Present Value and turn OFF the Error Counter 0 Error Flag. | --- | --- | Cleared | --- | --- |
|  | A563.01 | Error Counter 1 Disable Bit | Turn ON this bit to hold the present value of error counter 1. | ON: Error counter value held. <br> OFF: Error counter value not held. | --- | Cleared | --- | --- |
| A580 | $\begin{array}{\|l} \hline \text { A580.00 } \\ \text { to } \\ \text { A580.03 } \end{array}$ | FB Communications Instruction Retries | Automatically stores the number of retries in the FB communications instruction settings specified in the PLC Setup. | 0 to F hex | --- | Cleared | Written at start of operation |  |
| A581 | All | FB Communications Instruction Response Monitoring Time | Automatically stores the FB communications instruction response monitoring time set in the PLC Setup. | 0001 to FFFF hex (Unit: 0.1 s ; Range: 0.1 to 6553.5) 0000 hex: 2 s | --- | Cleared | Written at start of operation | --- |


| Addresses |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related Flags, Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bits |  |  |  |  |  |  |  |
| A598 | A598.00 | FPD Teaching Bit | Turn this bit ON to set the monitoring time automatically with the teaching function. <br> While A598.00 is ON, FPD(269) measures how long it takes for the diagnostic output to go ON after the execution condition goes ON. If the measured time exceeds the monitoring time, the measured time is multiplied by 1.5 and that value is stored as the new monitoring time. <br> (The teaching function can be used only when a word address has been specified for the monitoring time operand.) | ON: Teach monitoring time OFF: Teaching function OFF | Cleared | Cleared | --- | --- |
| $\begin{aligned} & \text { A600 to } \\ & \text { A603 } \end{aligned}$ | All | Macro Area Input Words | Before the subroutine specified in MCRO(099) is executed, the source words for the subroutine are transferred to A600 through A603 (input parameter words). | Input data: 4 words | Cleared | Cleared | --- | --- |
| $\begin{aligned} & \text { A604 to } \\ & \text { A607 } \end{aligned}$ | All | Macro Area Output Words | After the subroutine specified in MCRO(099) has been executed, the results of the subroutine are transferred from A604 through A607 to the specified destination words (output parameter words). | Output data: 4 words | Cleared | Cleared | --- | --- |
| A619 | A619.01 | Serial Port 1 Settings Changing Flag (CP1L M-type CPU Units) | ON while the serial port 1's communications settings are being changed for a CP1L M-type CPU Unit. This flag will be turned ON when STUP(237) is executed and it will be turned OFF after the settings have been changed. | ON: Changing OFF: Not changing | Retained | Cleared | --- | --- |
|  | A619.02 | Serial Port 2 <br> Settings <br> Changing <br> Flag (CP1L <br> M-type CPU <br> Units) | ON while the serial port 2's communications settings are being changed for a CP1L M-type CPU Unit. This flag will be turned ON when STUP(237) is executed and it will be turned OFF after the settings have been changed. | ON: Changing OFF: Not changing | Retained | Cleared | --- | --- |
|  |  | Serial Port 1 Settings Changing Flag (CP1L L-type CPU Units) | ON while the serial port 1's communications settings are being changed for a CP1L L-type CPU Unit. This flag will be turned ON when STUP(237) is executed and it will be turned OFF after the settings have been changed. |  |  |  |  |  |


| Addresses |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related Flags, Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bits |  |  |  |  |  |  |  |
| A640 | A640.00 | Serial Port 2 ModbusRTU Easy Master Execution Bit (CP1L Mtype CPU Units) | Turn ON this bit to send a command and receive a response for serial port 2 of a CP1L M-type CPU Unit using the Modbus-RTU easy master function. <br> Note This bit will be turned OFF automatically by the system when communications have been completed. | Turned ON: <br> Execution started <br> ON: Execution in progress. <br> OFF: Not executed or execution completed. | Retained | Cleared | --- | DM fixed allocation words for ModbusRTU Easy Master: <br> D32300 to D32399 |
|  |  | Serial Port 1 ModbusRTU Easy Master Execution Bit (CP1L Ltype CPU Units) | Turn ON this bit to send a command and receive a response for serial port 1 of a CP1L L-type CPU Unit using the Modbus-RTU easy master function. <br> Note This bit will be turned OFF automatically by the system when communications have been completed. |  |  |  |  |  |
|  | A640.01 | Serial Port 2 Modbus- <br> RTU Easy Master Normal End Flag (CP1L M-type CPU Units) | ON when one command has been sent and the response received for serial port 2 of a CP1L M-type CPU Unit using the Modbus-RTU easy master function. | ON: Execution normal. <br> OFF: Execution error or still in progress. | Retained | Cleared | --- |  |
|  |  | Serial Port 1 ModbusRTU Easy Master Normal End Flag (CP1L L-type CPU Units) | ON when one command has been sent and the response received for serial port 1 of a CP1L L-type CPU Unit using the Modbus-RTU easy master function. |  |  |  |  |  |
|  | A640.02 | Serial Port 2 ModbusRTU Easy Master Error End Flag (CP1L Mtype CPU Units) | ON when an error has occurred in communications for serial port 2 of a CP1L M-type CPU Unit using the Modbus-RTU easy master function. <br> The error code is output to D32252 in the DM fixed allocation words for Modbus-RTU Easy Master. | ON: Execution error. <br> OFF: Execution normal or still in progress. | Retained | Cleared | --- |  |
|  |  | Serial Port 1 ModbusRTU Easy Master Error End Flag (CP1L Ltype CPU Units) | ON when an error has occurred in communications for serial port 1 of a CP1L L-type CPU Unit using the Modbus-RTU easy master function. <br> The error code is output to D32252 in the DM fixed allocation words for Modbus-RTU Easy Master. |  |  |  |  |  |


| Addresses |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related Flags, Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bits |  |  |  |  |  |  |  |
| A641 | A641.00 | Serial Port 1 Modbus- <br> RTU Master Execution Bit (CP1L M-type CPU Units) | Turn ON this bit to send a command and receive a response for serial port 1 of a CP1L M-type CPU Unit using the Modbus-RTU easy master function. <br> This bit will be turned OFF automatically by the system when communications have been completed. | Turned ON: Execution started ON: Execution in progress. <br> OFF: Not executed or execution completed. | Retained | Cleared | --- | DM fixed allocation words for ModbusRTU Easy Master:D32200 toD32299 |
|  | A641.01 | Serial Port 1 ModbusRTU Master Execution Normal Flag (CP1L Mtype CPU Units) | ON when one command has been sent and the response received for serial port 1 of a CP1L M-type CPU Unit using the Modbus-RTU easy master function. | ON: Execution normal. <br> OFF: Execution error or still in progress. | Retained | Cleared | --- |  |
|  | A641.02 | Serial Port 1 ModbusRTU Master Execution Error Flag (CP1L Mtype CPU Units) | ON when an error has occurred in communications for serial port 1 of a CP1L M-type CPU Unit using the Modbus-RTU easy master function. The error code is output to D32352 in the DM fixed allocation words for Modbus-RTU Easy Master. | ON: Execution error. <br> OFF: Execution normal or still in progress. | Retained | Cleared | --- |  |
| A642 | All | Analog Adjustment PV | Stores the value set on the analog adjuster as a hexadecimal value (resolution: 1/256). | $\begin{aligned} & 0000 \text { to 00FF } \\ & \text { hex } \end{aligned}$ | Retained | Cleared | --- | --- |
| A643 | All | External Analog Setting Input PV | Stores the value set from the external analog setting input as a hexadecimal value (resolution: 1/256). | $\begin{aligned} & 0000 \text { to 00FF } \\ & \text { hex } \end{aligned}$ | Retained | Cleared | --- | --- |
| $\begin{aligned} & \text { A720 to } \\ & \text { A722 } \end{aligned}$ | All | Power ON Clock Data 1 | These words contain the time at which the power was turned ON one time before the startup time stored in words A510 to A511. <br> A720.00 to A720.07: Seconds (00 to 59) <br> A720.08 to A720.15: Minutes (00 to 59) <br> A721.00 to A721.07: Hour (00 to 23) A721.08 to A721.15: Day of month (00 to 31) <br> A722.00 to A722.07: Month (01 to 12) <br> A722.08 to A722.15: Year (00 to 99) | See at left. | Retained | Retained | Written when power is turned ON. |  |
| $\begin{aligned} & \text { A723 to } \\ & \text { A725 } \end{aligned}$ | All | Power ON Clock Data 2 | These words contain the time at which the power was turned ON two times before the startup time stored in words A510 to A511. <br> A723.00 to A723.07: Seconds (00 to 59) <br> A723.08 to A723.15: Minutes (00 to 59) <br> A724.00 to A724.07: Hour (00 to 23) A724.08 to A724.15: Day of month (00 to 31) <br> A725.00 to A725.07: Month (01 to 12) <br> A725.08 to A725.15: Year (00 to 99) | See at left. | Retained | Retained | Written when power is turned ON. |  |
| $\begin{array}{\|l} \hline \text { A726 to } \\ \text { A728 } \end{array}$ | All | Power ON Clock Data 3 | These words contain the time at which the power was turned ON three times before the startup time stored in words A510 to A511. <br> A726.00 to A726.07: Seconds (00 to 59) <br> A726.08 to A726.15: Minutes (00 to 59) <br> A727.00 to A727.07: Hour (00 to 23) A727.08 to A727.15: Day of month (00 to 31) <br> A728.00 to A728.07: Month (01 to 12) <br> A728.08 to A728.15: Year (00 to 99) | See at left. | Retained | Retained | Written when power is turned ON. |  |


| Addresses |  | Name | Function | Settings | Status after mode change | Status at startup | Write timing | Related Flags, Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bits |  |  |  |  |  |  |  |
| $\begin{aligned} & \text { A729 to } \\ & \text { A731 } \end{aligned}$ | All | Power ON Clock Data 4 | These words contain the time at which the power was turned ON four times before the startup time stored in words A510 to A511. <br> A729.00 to A729.07: Seconds (00 to 59) <br> A729.08 to A729.15: Minutes (00 to 59) <br> A730.00 to A730.07: Hour (00 to 23) A730.08 to A730.15: Day of month (00 to 31) <br> A731.00 to A731.07: Month (01 to 12) <br> A731.08 to A731.15: Year (00 to 99) | See at left. | Retained | Retained | Written when power is turned ON. |  |
| $\begin{aligned} & \text { A732 to } \\ & \text { A734 } \end{aligned}$ | All | Power ON Clock Data 5 | These words contain the time at which the power was turned ON five times before the startup time stored in words A510 to A511. <br> A732.00 to A732.07: Seconds (00 to 59) <br> A732.08 to A732.15: Minutes (00 to 59) <br> A733.00 to A733.07: Hour (00 to 23) A733.08 to A733.15: Day of month (00 to 31) <br> A734.00 to A734.07: Month (01 to 12) <br> A734.08 to A734.15: Year (00 to 99) | See at left. | Retained | Retained | Written when power is turned ON. |  |
| $\begin{aligned} & \text { A735 to } \\ & \text { A737 } \end{aligned}$ | All | Power ON Clock Data 6 | These words contain the time at which the power was turned ON six times before the startup time stored in words A510 to A511. <br> A735.00 to A735.07: Seconds (00 to 59) <br> A735.08 to A735.15: Minutes (00 to 59) <br> A736.00 to A736.07: Hour (00 to 23) A736.08 to A736.15: Day of month (00 to 31) <br> A737.00 to A737.07: Month (01 to 12) <br> A737.08 to A737.15: Year (00 to 99) | See at left. | Retained | Retained | Written when power is turned ON. |  |
| $\begin{aligned} & \text { A738 to } \\ & \text { A740 } \end{aligned}$ | All | Power ON Clock Data 7 | These words contain the time at which the power was turned ON seven times before the startup time stored in words A510 to A511. <br> A738.00 to A738.07: Seconds (00 to 59) <br> A738.08 to A738.15: Minutes (00 to 59) <br> A739.00 to A739.07: Hour (00 to 23) A739.08 to A739.15: Day of month (00 to 31) <br> A740.00 to A740.07: Month (01 to 12) <br> A740.08 to A740.15: Year (00 to 99) | See at left. | Retained | Retained | Written when power is turned ON. |  |
| $\begin{aligned} & \text { A741 to } \\ & \text { A743 } \end{aligned}$ | All | Power ON Clock Data 8 | These words contain the time at which the power was turned ON eight times before the startup time stored in words A510 to A511. <br> A741.00 to A741.07: Seconds (00 to 59) <br> A741.08 to A741.15: Minutes (00 to 59) <br> A742.00 to A742.07: Hour (00 to 23) A742.08 to A742.15: Day of month (00 to 31) <br> A743.00 to A743.07: Month (01 to 12) <br> A743.08 to A743.15: Year (00 to 99) | See at left. | Retained | Retained | Written when power is turned ON. |  |


| Addresses |  | Name | Function | Settings | Status after mode change | Statusat startup | Write timing | Related Flags, Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Word | Bits |  |  |  |  |  |  |  |
| $\begin{aligned} & \text { A744 to } \\ & \text { A746 } \end{aligned}$ | All | Power ON Clock Data 9 | These words contain the time at which the power was turned ON nine times before the startup time stored in words A510 to A511. <br> A744.00 to A744.07: Seconds (00 to 59) <br> A744.08 to A744.15: Minutes (00 to 59) <br> A745.00 to A745.07: Hour (00 to 23) A745.08 to A745.15: Day of month (00 to 31) <br> A746.00 to A746.07: Month (01 to 12) <br> A746.08 to A746.15: Year (00 to 99) | See at left. | Retained | Retained | Written when power is turned ON. |  |
| $\begin{array}{\|l\|l} \text { A747 to } \\ \text { A749 } \end{array}$ | All | Power ON Clock Data 10 | These words contain the time at which the power was turned ON ten times before the startup time stored in words A510 to A511. <br> A747.00 to A747.07: Seconds (00 to 59) <br> A747.08 to A747.15: Minutes (00 to 59) <br> A748.00 to A748.07: Hour (00 to 23) A748.08 to A748.15: Day of month (00 to 31) <br> A749.00 to A749.07: Month (01 to 12) <br> A749.08 to A749.15: Year (00 to 99) | See at left. | Retained | Retained | Written when power is turned ON. |  |
| A751 | A751.11 | DM Initial Values Read Error Flag | ON when an error occurred in transferring DM initial values from the DM initial value area in flash memory to the DM Area. | OFF: Normal <br> ON: Error (failed to load) | Retained | Cleared | --- | --- |
|  | A751.12 | DM Initial Values Save Execution Error Flag | ON when the DM Initial Values Transfer Password (A752) is incorrect or when the DM Initial values area was not specified when starting to transfer DM initial values from the DM Area to the DM initial value area in flash memory. | OFF: Normal ON: Error (failed to save) | Retained | Cleared | --- | --- |
|  | A751.13 | DM Initial Values Save Error Flag | ON when an error occurred in transferring DM initial values from the DM Area to the DM initial value area in flash memory. | OFF: Normal <br> ON: Error (failed to save) | Retained | Cleared | --- | --- |
|  | A751.14 | DM Initial Values Save Flag | ON while DM initial values are being transferred from the DM Area to the DM initial value area in flash memory. <br> OFF when the transfer has been completed. | OFF: Not being saved ON: Being saved | Retained | Cleared | --- | --- |
|  | A751.15 | DM Initial Values Save Start Bit | Turn ON this bit to start transferring DM initial values. This bit is valid only when a correct password is stored in A752 and the DM Area Initial Value Area is specified (i.e., when A753.00 is ON ). <br> The system will turn this bit OFF automatically when the transfer has been completed. | Turned ON: <br> Transfer started <br> OFF: Not transferring ON: Transferring | Retained | Cleared | --- | --- |
| A752 | All | DM Initial Values Save Password | Set the passwords here to transfer DM initial values between the DM area and the DM initial value area in flash memory. The transfer will not be started unless the correct password is set. <br> The transfer is started when A751.15 is turned ON. <br> The password will be cleared by the system when the transfer has been completed. | A5A5 hex: Save initial values from DM to flash | Retained | Cleared | --- | --- |
| A753 | All | DM Initial Values Save Area Specification | Specifies the area to be transferred to flash memory. | 0001 hex: DM Area specified | Retained | Cleared | --- | --- |

Note The following flags are provided in a special read-only area and can be specified with the labels given in the table. These flags are not contained in the Auxiliary Area. Refer to 4-14 Condition Flags and 4-15 Clock Pulses for details.

| Flag area | Name | Label | Meaning |
| :---: | :---: | :---: | :---: |
| Condition Code Area | Error Flag | ER | Turns ON when an error occurs in processing an instructions, indicating an error end to the instruction. |
|  | Access Error Flag | AER | Turns ON when an attempt is made to access an illegal area. The status of this flag is maintain only during the current cycle and only in the task in which it occurred. |
|  | Carry Flag | CY | Turns ON when there is a carry or borrow in a math operation, when a bit is shifted into the Carry Flag, etc. |
|  | Greater Than Flag | > | Turns ON when the result of comparing two values is "greater than," when a value exceeds a specified range, etc. |
|  | Equals Flag | $=$ | Turns ON when the result of comparing two values is "equals," when the result of a math operation is 0 , etc. |
|  | Less Than Flag | < | Turns ON when the result of comparing two values is "less than," when a value is below a specified range, etc. |
|  | Negative Flag | N | Turns ON when the MSB in the result of a math operation is 1. |
|  | Overflow Flag | OF | Turns ON when the result of a math operation overflows. |
|  | Underflow Flag | UF | Turns ON when the result of a math operation underflows. |
|  | Greater Than or Equals Flag | >= | Turns ON when the result of comparing two values is "greater than or equals." |
|  | Not Equal Flag | <> | Turns ON when the result of comparing two values is "not equal." |
|  | Less than or Equals Flag | <= | Turns ON when the result of comparing two values is "less than or equals." |
|  | Always ON Flag | A1 | This flag is always ON. |
|  | Always OFF Flag | A0 | This flag is always OFF. |
| Clock Pulse Area | 0.02-s clock pulse | 0.02s | Repeatedly turns ON for 0.02 s and OFF for 0.02 s . |
|  | 0.1-s clock pulse | 0.1 s | Repeatedly turns ON for 0.1 s and OFF for 0.1 s . |
|  | 0.2-s clock pulse | 0.2s | Repeatedly turns ON for 0.2 s and OFF for 0.2 s . |
|  | 1-s clock pulse | 1s | Repeatedly turns ON for 1 s and OFF for 1 s . |
|  | 1-min clock pulse | 1 min | Repeatedly turns ON for 1 min and OFF for 1 min . |

## Details on Auxiliary Area Operation

## A100 to A199: Error Log Area



The following data would be generated in an error record if a memory error (error code 80F1) occurred on 1 April 1998 at 17:10:30 with the error located in the PLC Setup (04 hex).

| 80 | $F 1$ |
| :---: | :---: |
| 00 | 04 |
| 10 | 30 |
| 01 | 17 |
| 98 | 04 |

The following data would be generated in an error record if an FALS error with FALS number 001 occurred on 2 May 1997 at 8:30:15.

| $C 1$ | 01 |
| :--- | :--- |
| 00 | 00 |
| 30 | 15 |
| 02 | 08 |
| 97 | 05 |

## Error Codes and Error Flags

| Classification | Error code | Meaning | Error flags |
| :---: | :---: | :---: | :---: |
| System-defined fatal errors | 80F1 | Memory error | A403 |
|  | 80CA | I/O bus error | A404 |
|  | 80E9 | Duplicate number error | A410, A411 to 416 (See note 3.) |
|  | 80E1 | Too many l/O error | A407 |
|  | 80E0 | I/O setting error | --- |
|  | 80F0 | Program error | A295 to A299 (See note 4.) |
|  | 809F | Cycle time too long error | --- |
| User-defined fatal errors | C101 to C2FF | FALS instruction executed (See note 1.) | --- |
| User-defined non-fatal errors | 4101 to 42FF | FAL instruction executed (See note 2.) | --- |
| System-defined non-fatal errors | 008B | Interrupt task error | A426 |
|  | 009A | Basic I/O error | A408 |
|  | 009B | PLC Setup setting error | A406 |
|  | 0200 to 020F | CPU Bus Unit error | A417 |
|  | 0300 to 035F | Special I/O Unit error | A418 to A423 (See note 5.) |
|  | 00F7 | Battery error | --- |
|  | 0400 to 040F | CPU Bus Unit setup error | A427 |
|  | 0500 to 055F | Special I/O Unit setup error | A428 to A433 (See note 5.) |

Note 1. C101 to C2FF will be stored for FALS numbers 001 to 511.
2. 4101 to $42 F F$ will be stored for FAL numbers 001 to 511 .
3. Only the contents of A295 is stored as the error flag contents for program errors.

## A200.11: First Cycle Flag



## A200.15: Initial Task Flag

A200.15 will turn ON during the first time a task is executed after it has reached executable status. It will be ON only while the task is being executed and will not turn ON if following cycles.


## A201.10: Online Editing Wait Flag



## A202.00 to A202.07: Communications Port Enabled Flags



Network communications in struction executed for port 0 .


The program is designed so that CMND(490) will be executed only when A202.00 is ON.

## A300: Error Record Pointer



## A401.09: Program Error Flag

|  | Error |  |
| :--- | :--- | :--- |
| Program Error Flag <br> (A401.09): ON | UM Overflow Error Flag | Address |
|  | Illegal Instruction Flag | A295.15 |
|  | Distribution Overflow Error Flag | A295.13 |
|  | Task Error Flag | A259.12 |
|  | No END(001) Error Flag | A295.11 |
|  | Illegal Area Access Error Flag | A295.10 |
|  | Indirect DM Addressing Error Flag | A295.09 |
|  | Instruction Processing Error Flag (ER <br> Flag goes ON) | A295.08 |

## Appendix E <br> Memory Map

## PLC Memory Addresses

PLC memory addresses are set in Index Registers (IR00 to IR15) to indirectly address I/O memory. Normally, use the MOVE TO REGISTER (MOVR(560)) and MOVE TIMER/COUNTER PV TO REGISTER (MOVRW(561)) instructions to set PLC memory addresses into the Index Registers.
Some instructions, such as DATA SEARCH (SRCH(181)), FIND MAXIMUM (MAX(182)), and FIND MINIMUM ( $\operatorname{MIN}(183)$ ), output the results of processing to an Index Register to indicate an PLC memory address.
There are also instructions for which Index Registers can be directly designated to use the PLC memory addresses stored in them by other instructions. These instructions include DOUBLE MOVE (MOVL(498)), some symbol comparison instructions (=L, <>L, <L, >L, <=L, and >=L), DOUBLE COMPARE (CMPL(060)), DOUBLE DATA EXCHANGE (XCGL(562)), DOUBLE INCREMENT BINARY (++L(591)), DOUBLE DECREMENT BINARY (-L(593)), DOUBLE SIGNED BINARY ADD WITHOUT CARRY (+L(401)), DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY (-L(411)), SET RECORD LOCATION (SETR(635)), and GET RECORD LOCATION (GETR(636)).
The PLC memory addresses all are continuous and the user must be aware of the order and boundaries of the memory areas. As reference, the PLC memory addresses are provided in a table at the end of this appendix.
Note Directly setting PLC memory addresses in the program should be avoided whenever possible. If PLC memory addresses are set in the program, the program will be less compatible with new CPU Unit models or CPU Units for which changes have been made to the layout of the memory.

## Memory Configuration

There are two classifications of the RAM memory (with battery backup) in a CP-series CPU Unit.
Parameter Areas: These areas contain CPU Unit system setting data, such as the PLC Setup, CPU Bus Unit Setups, etc. An illegal access error will occur if an attempt is made to access any of the parameter areas from an instruction in the user program.
I/O Memory Areas: These are the areas that can be specified as operands in the instructions in user programs.

## Memory Map

Note Do not access the areas indicated Reserved for system.

| Classification | PLC memory addresses (hex) | User addresses | Area |
| :---: | :---: | :---: | :---: |
| I/O memory areas | 0B100 to 0B7FF | --- | Reserved for system. |
|  | 0B800 to 0B801 | TK00 to TK31 | Task Flag Area |
|  | 0B802 to 0B83F | --- | Reserved for system. |
|  | 0B840 to 0B9FF | A0 to A447 | Read-only Auxiliary Area |
|  | OBA00 to OBBFF | A448 to A959 | Read/Write Auxiliary Area |
|  | OBC00 to OBDFF | --- | Reserved for system. |
|  | OBE00 to OBEFF | T0000 to T4095 | Timer Completion Flags |
|  | OBF00 to OBFFF | C0000 to C4095 | Counter Completion Flags |
|  | 0C000 to 0D7FF | CIO 0 to CIO 6143 | CIO Area |
|  | 0D800 to 0D9FF | H0 to H511 | Holding Area |
|  | 0DA00 to 0DDFF | --- | Reserved for system. |
|  | ODE00 to ODFFF | W0 to W511 | Work Area |
|  | 0E000 to OEFFF | T0000 to T4095 | Timer PVs |
|  | 0F000 to OFFFF | C0000 to C4095 | Counter PVs |
|  | 10000 to 17FFF | D0 to D32767 | DM Area (See note 2.) |
|  | 18000 to FFFFF | --- | Reserved for system. |

Note (1) Do not access areas reserved for the system.
(2) D10000 to D31999 (PLC memory addresses 12710 to 17CFF hex) cannot be used with CPU Units with 10, 14 or 20 I/O Points.

## Appendix F

## Connections to Serial Communications Option Boards

## Connection Methods

## Communications Modes and Ports

The following table shows the relationship between the communications ports and the communications modes for the Serial Communications Option Boards.

| Communications mode | $\begin{gathered} \text { RS-232C } \\ \text { CP1W-CIF01 } \end{gathered}$ |  | RS-422A/485CP1W-CIF11/CIF12 |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | 1:1 |  | 1:1 4-wire | 1:N 4-wire | 1:1 2-wire | 1:N 2-wire |
| Host Link | YES | YES (See note 2.) | YES | YES | No | No |
| Serial PLC Links | YES | YES | YES | YES | YES | YES |
| Serial Gateway | YES | YES | YES | YES | YES | YES |
| No-protocol | YES | YES | YES | YES | YES | YES |
| 1:N NT Link | YES | YES | YES | YES | YES | YES |
| 1:1 NT Link | YES | No | YES | No | YES | No |
| 1:1 Link Master | YES | No | YES | No | YES | No |
| 1:1 Link Slave | YES | No | YES | No | YES | No |

Note (1) The NT-AL001 Link Adapter can be used to convert between RS-232C and RS-422A/485 to enable $1: \mathrm{N}$ communications.
(2) Use 4-wire connections between Link Adapters.

Models of Serial Communications Option Board

| Model | Port | Maximum transmission <br> distance | Connection method |
| :--- | :--- | :--- | :--- |
| CP1W-CIF01 | One RS-232C port | 15 m | Connector (D-sub, 9-pin female) |
| CP1W-CIF11 | One RS-422A/485 port | 50 m (See note.) | Terminal block (using ferrules) |
| CP1W-CIF12 | One RS-422A/485 port | 500 m | Terminal block (using ferrules) |

Note The CP1W-CIF11 is a non-isolated board, so the maximum transmission distance is 50 m . For distances over 50 m , use the RS-232C port on the CP1W-CIF01 and then connect through the NT-AL001 Link Adapter, which is isolated. Doing so will enable a maximum transmission distance of 500 m .

## Reducing Electrical Noise for External Wiring

Observe the following precautions when wiring communications cables, PLC power lines, and high-power lines. When multi-conductor signal cable is being used, avoid using I/O wires and other control wires in the same cable.

- If wiring racks are running in parallel, allow at least 300 mm between them.

- If the I/O wiring and power cables must be placed in the same duct, they must be shielded from each other using grounded steel sheet metal.


Ground to $100 \Omega$ or less.

## 2-Wire and 4-Wire Connections

The transmission circuits for 2 -wire and 4 -wire connections are different, as shown in the following diagram.



Note (1) Use the same transmission circuit (2-wire or 4 -wire) for all nodes.
(2) Do not use 4 -wire connections when the $2 / 4$-wire switch on the Board is set to 2 -wire.

## NT-AL001 Link Adapter Settings

The NT-AL001 Link Adapter has a DIP switch for setting RS-422A/485 communications conditions. When connecting the Serial Communications Option Board, refer to the DIP switch settings shown in the following table.

| Pin | Function | Factory setting |
| :--- | :--- | :--- |
| 1 | Not used. Always set this pin to ON. | ON |
| 2 | Built-in terminating resistance setting <br> ON: Connects terminating resistance. <br> OFF: Disconnects terminating resistance. | ON |
| 3 | 2/4-wire setting |  |
| 4 | 2-wire: Set both pins to ON. <br> 4-wire: Set both pins to OFF. | OFF <br> 5 |
| Transmission mode (See note.) <br> Constant transmission: Set both pins to OFF. <br> Transmission performed when CTS signal in RS-232C interface is at high level: <br> Set pin 5 to OFF and pin 6 to ON. <br> Transmission performed when CTS signal in RS-232C interface is at low level: <br> Set pin 5 to ON and pin 6 to OFF. | ON |  |
|  |  |  |

Note When connecting to a CP-series CPU Unit, turn OFF pin 5 and turn ON pin 6.

## Connections for Host Link Communications

Port connections for Host Link communications are shown in the following table. Up to 32 nodes can be connected for 1:N connections.

| Port | Configuration | Schematic diagram, RS-232C ports | Schematic diagram, RS-422A/485 ports |
| :---: | :---: | :---: | :---: |
| Computer to PLC: C-mode or FINS commands PLC to computer: FINS commands | 1:1 |  |  |
| Computer to PLC: C-mode or FINS commands | $1: N$ |  |  |

Note (1) Four-wire connections must be used for RS-422A/485 connections with Host Link communications.
(2) "Resistance ON" indicates the terminating resistance must be turned ON.
(3) " $5-\mathrm{V}$ power" indicates that a $5-\mathrm{V}$ power supply is required for the Link Adapter. Refer to the Link Adapter manual for details. A $5-\mathrm{V}$ power supply is not required for a Link Adapter connected to an RS-232C Option Board mounted on the CPU Unit because power is supplied from pin 6 of the connector.
(4) The maximum cable length for RS-232C is 15 m . The RS-232C standard, however, does not cover baud rates above 19.2 Kbps . Refer to the manual for the device being connected to confirm support.

## Connection Examples

The connection examples in the remainder of this section show only the basic connection diagrams. We recommend that appropriate noise countermeasures be taken in actual applications, including the use of shielded twisted-pair cables. Refer to Recommended RS-422A/485 Wiring Examples on page 735 for actual wiring methods.

## Host Computer Connections

## 1:1 Connections Using RS-232C Ports

- IBM PC/AT or Compatible Computers

- Using NT-AL001 Converting Link Adapters


Note We recommend using the following NT-AL001 Link Adapter Connecting Cables to connect to NT-ALO01 Link Adapters.
XW2Z-070T-1: 0.7 m
XW2Z-200T-1: 2 m
Caution Do not use the 5-V power from pin 6 of the RS-232C Option Board for anything but the NT-AL001 Link Adapter. Using this power supply for any other external device may damage the RS-232C Option Board or the external device.

## 1:N Connections Using RS-232C Ports



Note We recommend using the following NT-AL001 Link Adapter Connecting Cables to connect to NT-AL001 Link Adapters.
XW2Z-070T-1: 0.7 m
XW2Z-200T-1: 2 m

## 1:1 Connections Using RS-422A/485 Port



DIP Switch Settings
Pin 1: ON
Pin 2: ON
(terminating resistance)
Pin 3: OFF
Pin 4: OFF
Pin 5: OFF
Pin 6: OFF

## 1:N Connections Using RS-422A/485 Ports




## Programmable Terminal (PT) Connections

## Direct Connections from RS-232C to RS-232C Ports



- Communications Mode: Host Link (unit number 0 only for Host Link) NT Link (1:N, N = 1 Unit only)
- OMRON Cables with Connectors:

XW2Z-200T-1: 2 m
XW2Z-500T-1: 5 m
1:1 Connections from RS-422A/485 to RS-422A/485 Ports


- Communications Mode: Host Link (unit number 0 only for Host Link) NT Link (1:N, N = 1 Unit only)

Note (1) RS-422A/485 Option Board settings:
Terminating resistance ON, 4-wire.
(2) The terminating resistant setting shown above is an example for the NT631/NT631C. The setting method varies with the PT. Refer to the manual for you PT for details.

## 1:N, 4-wire Connections from RS-422A/485 to RS-422A/485 Ports



Terminal block or D-sub connector

- Communications Mode: 1:N NT Link

Note (1) RS-422A/485 Option Board settings:
Terminating resistance ON, 4-wire.
(2) The terminating resistant setting shown above is an example for the NT631/NT631C. The setting method varies with the PT. Refer to the manual for you PT for details.

## 1:N, 2-wire Connections from RS-422A/485 to RS-422A/485 Ports



- Communications Mode: 1:N NT Link

Note (1) RS-422A/485 Option Board settings:
Terminating resistance ON, 2-wire.
(2) The terminating resistant setting shown above is an example for the NT631/NT631C. The setting method varies with the PT. Refer to the manual for you PT for details.

## Connections for Serial Gateway and No-protocol Communications

This section describes the connections for Serial Gateway, and no-protocol communications. Up to 32 nodes can be used for 1:N connections.


Note (1) The maximum cable length for RS-232C is 15 m . The RS-232C standard, however, does not cover baud rates above 19.2 Kbps. Refer to the manual for the device being connected to confirm support.
(2) The combined cable length for RS-422A/485 is 500 m including branch lines.
(3) The maximum cable length is limited to 2 m when an NT-AL001 Link Adapter is connected.
(4) Branch lines must be a maximum of 10 m long.

| Port | Configuration | Schematic diagram |
| :---: | :---: | :---: |
| $\begin{aligned} & \text { RS-422A/ } \\ & 485 \end{aligned}$ | 1:1 |  |
| $\begin{aligned} & \text { RS-422A/ } \\ & 485 \end{aligned}$ | 1:N |  |

Note (1) The maximum cable length for RS-232C is 15 m . The RS-232C standard, however, does not cover baud rates above 19.2 Kbps. Refer to the manual for the device being connected to confirm support.
(2) The CP1W-CIF11 is a non-isolated board, so the maximum transmission distance is 50 m . The CP1W-CIF12 is an isolated board, so the maximum transmission distance is 500 m . For distances over 50 m , use the RS-422A/485 port on the CP1W-CIF12 directly, or the RS-232C port on the CP1W-CIF01 and then connect through the NT-AL001 Link Adapter, which is isolated. Doing so will enable a maximum transmission distance of 500 m .
(3) The maximum cable length is limited to 2 m when an NT-AL001 Link Adapter is connected.
(4) Branch lines must be a maximum of 10 m long.

## Connection Examples

The connection examples in the remainder of this section show only the basic connection diagrams. We recommend that appropriate noise countermeasures be taken in actual applications, including the use of shielded twisted-pair cables. Refer to RS-232C and RS-422A/485 Wiring for actual wiring methods.

## Connecting RS-232C Ports 1:1

## Connections to E5CK Controller



## Connections to a Host Computer



## Connections to a Personal Computer with RTS-CTS Flow Control

CPU Unit

| Computer |
| :--- |
| 1  <br> 2 RD <br> 3 SD <br> 4 ER <br> 5 SG <br> 6 DR <br> 7 RS <br> 8 CS <br> 9  |

## Connecting a Host Computer with NT-AL001 Converting Link Adapters



Note We recommend using the following NT-ALO01 Link Adapter Connecting Cables to connect to NT-AL001 Link Adapters.
XW2Z-200T-1: 2 m
XW2Z-500T-1: 5 m

## Connections to a Modem

| Modem |  |  | CPU Unit <br> RS-232C <br> Option Board |  |
| :---: | :---: | :---: | :---: | :---: |
| 1 | FG |  | 1 | FG |
| 2 | SD | , | 2 | SD |
| 3 | RD |  | 3 | RD |
| 4 | RS |  | 4 | RS |
| 5 | CS |  | 5 | CS |
| 6 | DR |  | 6 | 5 V |
| 7 | SG |  | 7 | DR |
| 8 | CD |  | 8 | ER |
| 9 |  |  | 9 | SG |
| 10 |  |  |  |  |
| 11 |  |  |  |  |
| 12 |  |  |  |  |
| 13 |  |  |  |  |
| 14 |  |  |  |  |
| 15 | ST2 |  |  |  |
| 16 |  |  |  |  |
| 17 | RT |  |  |  |
| 18 |  |  |  |  |
| 19 |  |  |  |  |
| 20 | ER |  |  |  |
| 21 |  |  |  |  |
| 22 | Cl |  |  |  |
| 23 |  |  |  |  |
| 24 | ST1 |  |  |  |
| 25 |  |  |  |  |

## 1:N Connections Using RS-232C Ports



DIP SW
Pin 1: ON
Pin 2: ON Terminating
resistance
Pin 3: ON 2-wire
Pin 4: ON
Pin 5: OFF
Pin 6: ON
Note We recommend using the following NT-AL001 Link Adapter Connecting Cables to connect to NT-AL001 Link Adapters.
XW2Z-070T-1: 0.7 m
XW2Z-200T-1: 2 m

## 1:1 Connections Using RS-422A/485 Ports



Pin 1: ON
Pin 2: ON
(terminating resistance)
Pin 3: OFF
Pin 4. OFF
Pin 4: OFF
Pin 6: OFF

## 1:N Connections Using RS-422A/485 Ports




## 1:N NT Link Connections with Programmable Terminals

Direct Connections from RS-232C to RS-232C Ports


- Communications Mode: Host Link (unit number 0 only for Host Link) NT Link ( $1: \mathrm{N}, \mathrm{N}=1$ Unit only)
- OMRON Cables with Connectors:

XW2Z-070T-1: 0.7 m
XW2Z-200T-1: 2 m

## 1:N, 4-wire Connections from RS-422A/485 to RS-422A/485 Ports



> Terminal block or D-sub connector

- Communications Mode: 1:N NT Link

Note (1) RS-422A/485 Option Board settings:
Terminating resistance ON, 4-wire.
(2) The terminating resistant setting shown above is an example for the NT631/NT631C. The setting method varies with the PT. Refer to the manual for you PT for details.

## 1:N, 2-wire Connections from RS-422A/485 to RS-422A/485 Ports



- Communications Mode: 1:N NT Link

Note (1) RS-422A/485 Option Board settings:
Terminating resistance ON, 2-wire.
(2) The terminating resistant setting shown above is an example for the NT631/NT631C. The setting method varies with the PT. Refer to the manual for you PT for details.

## Serial PLC Link Connection Examples

This section provides connection examples for using Serial PLC Link. The communications mode used here is Serial PLC Link.

## Connecting an RS-422A Converter

CP1L CPU Unit
(Polling Unit)


CP1L CPU Unit
(Polled Unit \#0)


Serial PLC Link
(Total transmission length: 50 m max.)

Note The CP1W-CIF11 is not insulated, so the total transmission distance for the whole transmission path is 50 m max. If the total transmission distance is greater than 50 m , use the RS-422A/485 port on the CP1W-CIF12 directly, or the insulated NT-AL001, and do not use the CP1W-CIF11. If the CP1W-CIF12 or NT-AL001 is used, the total transmission distance for the whole transmission path is 500 m max.

## Connection with an RS-232C Port

RS-232C connection is also possible when using a Serial PLC Link to connect two CP1L CPU Units.


## Connection Examples



## Connections in Loopback Test

Connect the communications ports as shown below.
RS-232C port

| Pin | Signal |
| :---: | :---: |
| 2 | SD |
| 3 | RD |
| 4 | RS |
| $\mathbf{5}$ |  |
| 5 | CS |
| 1 | FG |
| 8 | ER |
| 7 | DR |

RS-422A/485 port

| Pin | Signal |
| :---: | :---: |
| 3 | SDA- |
|  |  |
| 4 | SDB + |
| 1 | RDA- |
| 2 | RDB + |
|  |  |
| 5 | FG |

## RS-232C and RS-422A/485 Wiring

## Recommended RS-232C Wiring Examples

It is recommended that RS-232C cables be connected as described below especially when the Option Board is used in an environment where it is likely to be subject to electrical noise.

1. Always use shielded twisted-pair cables as communications cables.

| Model | Manufacturer |
| :--- | :--- |
| UL2464 AWG28x5P IFS-RVV-SB (UL product) | Fujikura Ltd. |
| AWG28x5P IFVV-SB (non-UL product) |  |
| UL2464-SB (MA) 5Px28AWG (7/0.127) (UL product) CO-MA-VV-SB | Hitachi Cable, Ltd. |
| 5Px28AWG (7/0.127) (non-UL product) |  |

2. Combine signal wires and SG (signal ground) wires in a twisted-pair cable. At the same time, bundle the SG wires to the connectors on Option Board and the remote device.
3. Connect the shield of the communications cable to the Hood (FG) terminal of the RS-232C connector on the Option Board. At the same time, ground the ground (GR) terminal of the CPU Unit to $100 \Omega$ or less.
4. A connection example is shown below.

Example: Twisted-pair Cable Connecting SD-SG, RD-SG, RTS-SG, and CTS-SG Terminals in Toolbus Mode
Actual Wiring Example


Note The Hood (FG) is internally connected to the ground terminal (GR) on the CPU Unit. Therefore, FG is grounded by grounding the ground terminal (GR) on the power supply terminal block. Although there is conductivity between the Hood (FG) and pin 1 (FG), connect the Hood (FG) to the shield because the Hood (FG) has smaller contact resistance with the shield than pin 1 (FG), and thus provides better noise resistance.


## Recommended RS-422A/485 Wiring Examples

Use the following wiring methods for RS-422A/485 to maintain transmission quality.

1. Always use shielded twisted-pair cables as communications cables.

| Model | Manufacturer |
| :--- | :--- |
| CO-HC-ESV-3Px7/0.2 | Hirakawa Hewtech Corp. |

2. Connect the shield of the communications cable to the FG terminal on the RS-422A/485 Option Board. At the same time, ground the ground (GR) terminal of the CPU Unit to $100 \Omega$ or less.

Note Always ground the shield only at the RS-422A/485 Option Board end. Grounding both ends of the shield may damage the device due to the potential difference between the ground terminals.

Connection examples are shown below.

- 2-Wire Connections

-4-Wire Connections
CP1H CPU Unit Option Board Remote device

- Using a B500-AL001-E Link Adapter

- With NT-AL001 RS-232C/RS-422 Link Adapter


Note (1) The following cables are available for this connection.

| Length | Model |
| :--- | :--- |
| 70 cm | XW2Z-070T-1 |
| 2 m | XW2Z-200T-1 |

It is recommended that one of these cables be used to connect the RS-232C port on the Option Board to the NT-AL001 RS-232C/RS-422 Link Adapter. The recommended wiring for these cables is shown below.

- Wiring for the Recommended Cables (XW2Z-070T-1 and XW2Z-200T-1, 10-conductor Cables)

(2) The XW2Z-070T-1 and XW2Z-200T-1 Connecting Cables for the NT-AL001 Link Adapter uses special wiring for the DTS and RTS signals. Do not use these signals with other devices; they may be damaged.
(3) The Hood (FG) is internally connected to the ground terminal (GR) on the CPU Unit. Therefore, FG is grounded by grounding the ground terminal (GR) on the power supply terminal block.


## Wiring Connectors

Use the following steps to wire connectors.
See the following diagrams for the length of the cable portion to be cut in each step.

## Shield Connected to Hood (FG)

1. Cut the cable to the required length.
2. Remove the specified length of the sheath from the cable using a knife. Be careful not to scratch the braided shield.

3. Trim off the braided shield using scissors so that the remaining shield length is 10 mm .

4. Remove the insulation from each conductor using a stripper so that the exposed conductor length is 5 mm .

5. Fold back the braided shield.

6. Wrap aluminum foil tape around the folded shield.


## Shield Not Connected to Hood (FG)

1. Cut the cable to the required length.
2. Remove the specified length of the sheath from the cable using a knife. Be careful not to scratch the braided shield.

3. Trim off all the braided shield using scissors.

4. Remove the insulation from each conductor using a stripper so that the exposed conductor length is 5 mm .

5. Wrap adhesive tape around the conductor from which the braided shield was removed.


## Soldering

1. Thread a heat-shrinking tube through each conductor.
2. Temporarily solder each conductor to the corresponding connector terminals.
3. Completely solder each conductor.

4. Return the heat-shrinking tube to the soldered portion, then heat the tube to shrink it in place.


## Assembling Connector Hood

Assemble the connector hood as shown below.


## Connecting to Unit



## Appendix G

## PLC Setup

## Startup Settings

## Startup Hold Settings

|  | Name | Default | Settings | When setting is read by CPU Unit | Internal address | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | Force Status Hold Bit | Not held. | Not held. | When power is turned ON | 80 | 14 | 0 |
|  |  |  | Held. |  |  |  | 1 |
| 2 | IOM Hold Bit | Not held. | Not held. | When power is turned ON | 80 | 15 | 0 |
|  |  |  | Held. |  |  |  | 1 |

## Startup Data Read Setting

|  | Name | Default | Settings | When setting is read <br> by CPU Unit | Internal <br> address | Bits | Settings |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| 1 | Read DM from flash <br> memory | Do not read. | Do not read. | When power is turned | 82 | 15 | 0 |
|  |  | Read. | ON |  |  |  |  |

## Mode: CPU Unit Operating Mode

|  | Name | Default | Settings | When setting is read by CPU Unit | Internal address | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | Use programming console (RUN mode) | Use programming console (RUN mode) (See note.) | Use programming console: RUN mode | When power is turned ON | 81 | $\begin{aligned} & 00 \text { to } \\ & 15 \end{aligned}$ | 0000 hex |
|  |  |  | Program: PROGRAM mode |  |  |  | 8000 hex |
|  |  |  | Monitor: MONITOR mode |  |  |  | 8001 hex |
|  |  |  | Run: RUN mode |  |  |  | 8002 hex |

Note A Programming Console cannot be connected to the CP1L. If the default setting, "Use programming console," is set, the CPU Unit will start in RUN mode.

## Settings: CPU Unit Settings

## Execute Process Settings

|  | Name | Default | Settings | When setting is read by CPU Unit | Internal address | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | Do not detect Low Battery (run without battery) | Detect. | Detect | Every cycle | 128 | 15 | 0 |
|  |  |  | Do not detect. |  |  |  | 1 |
| 2 | Detect Interrupt Task Error | Detect. | Detect | Every cycle | 128 | 14 | 0 |
|  |  |  | Do not detect. |  |  |  | 1 |
| 3 | Stop CPU on Instruction Error | Do not stop. | Do not stop. | At start of operation | 197 | 15 | 0 |
|  |  |  | Stop |  |  |  | 1 |
| 4 | Don't resister FAL to error log | Register. | Register. | Every cycle | 129 | 15 | 0 |
|  |  |  | Do not register. |  |  |  | 1 |

## Comms Instructions Settings in FB:

## Settings for Communications Instructions in Function Blocks

|  | Name | Default | Settings | When setting is read by CPU Unit | Internal address | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | Retry Counts: Number of retries | 0 | 0 to 15 | At start of operation | 200 | $\begin{array}{\|l\|} \hline 00 \text { to } \\ 03 \end{array}$ | $\begin{aligned} & \hline 0 \text { hex } \\ & \vdots \\ & \text { F hex } \end{aligned}$ |
| 2 | Response Timeout (default 2s), Comms Instructions in FB | 2 s | 2 s | At start of operation | 201 | $\begin{aligned} & 00 \text { to } \\ & 15 \end{aligned}$ | 0000 hex |
|  |  |  | $1: 1 \times 0.1 \mathrm{~s}$ |  |  |  | 0001 hex |
|  |  |  |  |  |  |  | : |
|  |  |  | 65535: 65,535 $\times 0.1 \mathrm{~s}$ |  |  |  | FFFF hex |
| 3 | Response Timeout (default 2s), DeviceNet Comms Instruction in FB | 2 s | 2 s | At start of operation | 202 | $\begin{array}{\|l\|} \hline 00 \text { to } \\ 15 \end{array}$ | 0000 hex |
|  |  |  | 1: $1 \times 0.1 \mathrm{~s}$ |  |  |  | 0001 hex |
|  |  |  | : |  |  |  | : |
|  |  |  | 65535: $65,535 \times 0.1 \mathrm{~s}$ |  |  |  | FFFF hex |

## Timings: Time and Interrupt Settings

## Cycle Time Settings

|  |  | Name | Default | Settings | When setting is read by CPU Unit | Internal address | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | Watch Cycle Time (default 1000 ms ) |  | Use default. | Use default. (Default: 1 s) | At start of operation | 209 | 15 | 0 |
|  |  |  | Use user setting. | 1 |  |  |  |
|  | 1-1 | Watch Cycle Time (default 1000 ms ) |  | 1,000 ms | 1: $1 \times 10 \mathrm{~ms}$ | At start of operation | 209 | $\begin{array}{\|l\|} \hline 00 \text { to } \\ 14 \end{array}$ | 001 hex |
|  |  |  | : |  | : |  |  |  |
|  |  |  | $\begin{aligned} & 40,000: \\ & 40,000 \times 10 \mathrm{~ms} \end{aligned}$ |  | FAO hex |  |  |  |
| 2 | Cycle Time (No Setting) |  | No minimum cycle time | No minimum cycle time | At start of operation | 208 | $\begin{aligned} & \hline 00 \text { to } \\ & 15 \end{aligned}$ | 0000 hex |
|  |  |  | 1 ms | 0001 hex |  |  |  |  |
|  |  |  | : | : |  |  |  |  |
|  |  |  | $32,000 \mathrm{~ms}$ | 7D00 hex |  |  |  |  |

## Interrupt Setting

|  | Name | Default | Settings | When setting is read by CPU Unit | Internal address | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | Scheduled Interrupt Interval | 10 ms | 10 ms | At start of operation | 195 | $\begin{array}{\|l\|} \hline 00 \text { to } \\ 03 \end{array}$ | 0 hex |
|  |  |  | 1 ms |  |  |  | 1 hex |
|  |  |  | 0.1 ms |  |  |  | 2 hex |

## Input Constant Settings

|  | Name | Default | Settings | When setting is read by CPU Unit | Internal address | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | 0CH: ClO 0 | 8 ms Default (8ms) | No filter (0 ms) | When power is turned ON | 10 | 00 to 07 | 10 hex |
|  |  |  | 0.5 ms |  |  |  | 11 hex |
|  |  |  | 1 ms |  |  |  | 12 hex |
|  |  |  | 2 ms |  |  |  | 13 hex |
|  |  |  | 4 ms |  |  |  | 14 hex |
|  |  |  | 8 ms |  |  |  | 15 hex |
|  |  |  | 16 ms |  |  |  | 16 hex |
|  |  |  | 32 ms |  |  |  | 17 hex |
| 2 | $1 \mathrm{CH}: \mathrm{ClO} 1$ | Same as above. | Same as above. | Same as above. | 10 | 08 to 15 | Same as above. |
| 3 | $2 \mathrm{CH}: \mathrm{CIO} 2$ |  |  |  | 11 | 00 to 07 |  |
| 4 | $3 \mathrm{CH}: \mathrm{ClO} 3$ |  |  |  | 11 | 08 to 15 |  |
| 5 | $4 \mathrm{CH}: \mathrm{CIO} 4$ |  |  |  | 12 | 00 to 07 |  |
| 6 | $5 \mathrm{CH}: \mathrm{ClO} 5$ |  |  |  | 12 | 08 to 15 |  |
| 7 | $6 \mathrm{CH}: \mathrm{CIO} 6$ |  |  |  | 13 | 00 to 07 |  |
| 8 | 7 CH : ClO 7 |  |  |  | 13 | 08 to 15 |  |
| 9 | $8 \mathrm{CH}: \mathrm{CIO} 8$ |  |  |  | 14 | 00 to 07 |  |
| 10 | $9 \mathrm{CH}: \mathrm{CIO} 9$ |  |  |  | 14 | 08 to 15 |  |
| 11 | $10 \mathrm{CH}: \mathrm{CIO} 10$ |  |  |  | 15 | 00 to 07 |  |
| 12 | 11 CH : CIO 11 |  |  |  | 15 | 08 to 15 |  |
| 13 | $12 \mathrm{CH}: \mathrm{CIO} 12$ |  |  |  | 16 | 00 to 07 |  |
| 14 | $13 \mathrm{CH}: \mathrm{CIO} 13$ |  |  |  | 16 | 08 to 15 |  |
| 15 | $14 \mathrm{CH}: \mathrm{CIO} 14$ |  |  |  | 17 | 00 to 07 |  |
| 16 | 15 CH : CIO 15 |  |  |  | 17 | 08 to 15 |  |
| 17 | $16 \mathrm{CH}: \mathrm{CIO} 16$ |  |  |  | 18 | 00 to 07 |  |
| 18 | 17 CH : CIO 17 |  |  |  | 18 | 08 to 15 |  |

## Serial Port 1 Settings

## Serial Communications Settings



|  | Name |  |  | Default | Settings | When setting is read | Internal | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 2 | 2-1 | 2-1-3 | Unit Number | 0 | 0 | Every cycle | $147$ <br> (CP1L M- <br> type CPU <br> Unit) | $\begin{aligned} & 00 \text { to } \\ & 07 \end{aligned}$ | 00 hex |
|  |  |  |  |  | : |  |  |  | : |
|  |  |  |  |  | 31 |  | 161 <br> (CP1L L- <br> type CPU <br> Unit) |  | 1F hex |
|  | 2-2 | NT Link (1:N): 1:N NT Links |  |  |  |  |  |  |  |
|  |  | 2-2-1 | Baud | $\begin{array}{\|l\|} \hline \text { 9,600 } \\ \text { (disabled) } \end{array}$ | 38,400 (standard) | Every cycle | $\begin{aligned} & 145 \\ & \text { (CP1L M- } \\ & \text { type CPU } \\ & \text { Unit) } \end{aligned}$ | $\begin{aligned} & 00 \text { to } \\ & 07 \end{aligned}$ | 00 hex |
|  |  |  |  |  | 115,200 (high speed) |  |  |  | OA hex |
|  |  |  |  |  |  |  | $161$ <br> (CP1L L- <br> type CPU <br> Unit) |  |  |
|  |  | 2-2-2 | NT/PC <br> Link Max: <br> Highest unit number | 0 | 0 | Every cycle | $\begin{array}{\|l} \hline 150 \\ \text { (CP1L M- } \\ \text { type CPU } \\ \text { Unit) } \end{array}$ | $\begin{aligned} & 00 \text { to } \\ & 03 \end{aligned}$ | 0 hex |
|  |  |  |  |  | : |  |  |  | : |
|  |  |  |  |  | 7 |  | $\begin{array}{\|l\|} \hline 166 \\ \text { (CP1L L- } \\ \text { type CPU } \\ \text { Unit) } \\ \hline \end{array}$ |  | 7 hex |
|  | 2-3 | RS-232C |  |  |  |  |  |  |  |
|  |  | 2-3-1 | Baud | 9600 bps | 300 bps | Every cycle | $\begin{aligned} & 145 \\ & \text { (CP1L M- } \\ & \text { type CPU } \\ & \text { Unit) } \end{aligned}$ | $\begin{aligned} & 00 \text { to } \\ & 07 \end{aligned}$ | 01 hex |
|  |  |  |  |  | 600 bps |  |  |  | 02 hex |
|  |  |  |  |  | 1,200 bps |  |  |  | 03 hex |
|  |  |  |  |  | 2,400 bps |  | 161 <br> (CP1L L- <br> type CPU <br> Unit) |  | 04 hex |
|  |  |  |  |  | 4,800 bps |  |  |  | 05 hex |
|  |  |  |  |  | 9,600 bps |  |  |  | 00 or 06 hex |
|  |  |  |  |  | 19,200 bps |  |  |  | 07 hex |
|  |  |  |  |  | 38,400 bps |  |  |  | 08 hex |
|  |  |  |  |  | 57,600 bps |  |  |  | 09 hex |
|  |  |  |  |  | 115,200 bps |  |  |  | 0A hex |
|  |  | 2-3-2 | Format (data length, stop bits, parity) | 7,2,E: 7-bit data, 2 stop bits, even parity | 7,2,E: 7-bit data, 2 stop bits, even parity | Every cycle | 144 <br> (CP1L M- <br> type CPU <br> Unit) | $\begin{aligned} & 00 \text { to } \\ & 03 \end{aligned}$ | 0 hex |
|  |  |  |  |  | 7,2,0: 7-bit data, 2 stop bits, odd parity |  |  |  | 1 hex |
|  |  |  |  |  | 7,2,N: 7-bit data, 2 stop bits, no parity |  |  |  | 2 hex |
|  |  |  |  |  | 7,1,E: 7-bit data, 2 stop bits, even parity |  | $160$ <br> (CP1L Ltype CPU Unit) |  | 4 hex |
|  |  |  |  |  | 7,1,O: 7-bit data, 1 stop bit, odd parity |  |  |  | 5 hex |
|  |  |  |  |  | 7,1,N: 7-bit data, 1 stop bit, no parity |  |  |  | 6 hex |
|  |  |  |  |  | 8,2,E: 8-bit data, 2 stop bits, even parity |  |  |  | 8 hex |
|  |  |  |  |  | 8,2,0: 8-bit data, 2 stop bits, odd parity |  |  |  | 9 hex |
|  |  |  |  |  | 8,2,N: 8-bit data, 2 stop bits, no parity |  |  |  | A hex |
|  |  |  |  |  | 8,1,E: 8-bit data, 1 stop bit, even parity |  |  |  | C hex |
|  |  |  |  |  | 8,1,O: 8-bit data, 1 stop bit, odd parity |  |  |  | D hex |
|  |  |  |  |  | 8,1,N: 8-bit data, 1 stop bit, no parity |  |  |  | E hex |


|  | Name |  |  | Default | Settings | When setting is read | Internal | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 2 | 2-3 | 2-3-3 | Start Code | Disable. | Disable. | Every cycle | $\begin{aligned} & 149 \\ & \text { (CP1L M- } \\ & \text { type CPU } \\ & \text { Unit) } \end{aligned}$ | 12 | 0 |
|  |  |  |  |  | Set. |  | $165$ <br> (CP1L Ltype CPU Unit) |  | 1 |
|  |  | 2-3-4 | Start Code | 00 hex0x0000 | 0x0000 | Every cycle | 148 | 08 to | 00 hex |
|  |  |  |  |  | : |  | type CPU <br> Unit) |  | : |
|  |  |  |  |  | 0x00FF |  | $164$ <br> (CP1L L- <br> type CPU <br> Unit) |  | FF hex |
|  |  | 2-3-5 | End Code | Received Bytes: Receive specified num- | Received Bytes: Receive specified number of bytes. | Every cycle | $\begin{aligned} & 149 \\ & \text { (CP1L M- } \\ & \text { type CPU } \end{aligned}$ | $\begin{aligned} & 08 \text { and } \\ & 09 \end{aligned}$ | 00 |
|  |  |  |  | ber of bytes. | CR,LF |  | Unit) |  | 10 |
|  |  |  |  |  | Set End Code |  | $\begin{aligned} & 165 \\ & \text { (CP1L L- } \\ & \text { type CPU } \\ & \text { Unit) } \end{aligned}$ |  | 01 |
|  |  | 2-3-6 | Received | 256 bytes | 256 bytes | Every cycle | 149 | 00 to | 00 hex |
|  |  |  | Bytes |  | 1 byte |  | P1L M- | 07 | 01 hex |
|  |  |  |  |  | : |  | Uni) |  | : |
|  |  |  |  |  | 255 bytes |  | $165$ <br> (CP1L L- <br> type CPU <br> Unit) |  | FF hex |
|  |  | 2-3-7 | Set End | 0x0000 | 1 byte | Every cycle | 148 | 00 to | 00 hex |
|  |  |  | Co |  | : |  | (CP1L Mtype CPU Unit) |  | : |
|  |  |  |  |  | 255 bytes |  | 164 <br> (CP1L L- <br> type CPU <br> Unit) |  | FF hex |
|  |  | 2-3-8 | Delay | 0 ms | $0: 0 \times 10 \mathrm{~ms}$ | Every cycle | 146 | 00 to | 0000 hex |
|  |  |  |  |  | : |  | (CP1L M- <br> type CPU <br> Unit) | 15 | : |
|  |  |  |  |  | 9999: $9999 \times 10 \mathrm{~ms}$ |  | $162$ <br> (CP1L Ltype CPU Unit) |  | 270F hex |
|  | 2-4 | ToolBu | (peripheral | bus) |  |  |  |  |  |
|  |  | 2-4-1 | Baud | 9,600 bps | 9,600 bps | Every cycle | $\begin{aligned} & 145 \\ & \text { (CP1L M- } \end{aligned}$ | $\begin{aligned} & 00 \text { to } \\ & 07 \end{aligned}$ | 00 or 06 hex |
|  |  |  |  |  | 19,200 bps |  | type CPU Unit) |  | 07 hex |
|  |  |  |  |  | 38,400 bps |  | 161 |  | 08 hex |
|  |  |  |  |  | 57,600 bps |  | (CP1L L- |  | 09 hex |
|  |  |  |  |  | 115,200 bps |  | Unit) |  | 0A hex |


|  | Name |  |  | Default | Settings | When setting is read | Internal | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 2 | 2-5 ${ }^{3}$ Serial |  | Serial Gateway |  |  |  |  |  |  |
|  |  |  | Baud | 9,600 bps | 300 bps | Every cycle | $\begin{aligned} & 145 \\ & \text { (CP1L M- } \\ & \text { type CPU } \\ & \text { Unit) } \end{aligned}$ | $\begin{aligned} & 00 \text { to } \\ & 07 \end{aligned}$ | 01 hex |
|  |  |  | 600 bps |  | 02 hex |  |  |  |
|  |  |  | 1,200 bps |  | 03 hex |  |  |  |
|  |  |  | 2,400 bps |  | $161$ <br> (CP1L Ltype CPU Unit) |  | 04 hex |  |
|  |  |  | 4,800 bps |  |  |  | 05 hex |  |
|  |  |  | 9,600 bps |  |  |  | 00 or 06 hex |  |
|  |  |  | 19,200 bps |  |  |  | 07 hex |  |
|  |  |  | 38,400 bps |  |  |  | 08 hex |  |
|  |  |  | 57,600 bps |  |  |  | 09 hex |  |
|  |  |  | 115,200 bps |  |  |  | 0A hex |  |
|  |  | 2-5-2 |  | Format (data | 7,2,E: 7-bit data, 2 stop | 7,2,E: 7-bit data, 2 stop bits, even parity | Every cycle | $\begin{aligned} & 144 \\ & (\mathrm{CP} 1 \mathrm{~L} \text { M- } \end{aligned}$ | $\begin{aligned} & 00 \text { to } \\ & 03 \end{aligned}$ | 0 hex |
|  |  |  |  | length, stop bits, parity) | bits, even parity | 7,2,0: 7-bit data, 2 stop bits, odd parity |  | type CPU <br> Unit) |  | 1 hex |
|  |  |  |  |  |  | 7,2,N: 7-bit data, 2 stop bits, no parity |  |  |  | 2 hex |
|  |  |  |  |  |  | 7,1,E: 7-bit data, 2 stop bits, even parity |  | $\begin{aligned} & \hline 160 \\ & \text { (CP1L L- } \end{aligned}$ |  | 4 hex |
|  |  |  |  |  |  | 7,1,O: 7-bit data, 1 stop bit, odd parity |  | type CPU <br> Unit) |  | 5 hex |
|  |  |  |  |  |  | 7,1,N: 7-bit data, 1 stop bit, no parity |  |  |  | 6 hex |
|  |  |  |  |  |  | 8,2,E: 8-bit data, 2 stop bits, even parity |  |  |  | 8 hex |
|  |  |  |  |  |  | 8,2,O: 8-bit data, 2 stop bits, odd parity |  |  |  | 9 hex |
|  |  |  |  |  |  | 8,2,N: 8-bit data, 2 stop bits, no parity |  |  |  | A hex |
|  |  |  |  |  | 8,1,E: 8-bit data, 1 stop bit, even parity |  |  |  | C hex |
|  |  |  |  |  | 8,1,O: 8-bit data, 1 stop bit, odd parity |  |  |  | D hex |
|  |  |  |  |  | 8,1,N: 8-bit data, 1 stop bit, no parity |  |  |  | E hex |
|  |  | 2-5-3 | Response |  | $50: 50 \times 100 \mathrm{~ms}=5 \mathrm{~s}$ | Every cycle | 151 | 08 to | 00 hex |
|  |  |  | Timeout | $50 \times 100 \mathrm{~ms}=$ | 1: $1 \times 100 \mathrm{~ms}$ |  |  |  | 01 hex |
|  |  |  |  |  |  |  | Unit) |  | : |
|  |  |  |  |  | 255: $255 \times 100 \mathrm{~ms}$ |  | $167$ <br> (CP1L Ltype CPU Unit) |  | FF hex |
|  | 2-6 | PC Lin | (Slave) |  |  |  |  |  |  |
|  |  | 2-6-1 | Baud | 9,600 bps (disabled) | 38,400 (standard) | Every cycle | $\begin{aligned} & 145 \\ & \text { (CP1L M- } \\ & \text { type CPU } \\ & \text { Unit) } \end{aligned}$ | $\begin{aligned} & 00 \text { to } \\ & 07 \end{aligned}$ | 00 hex |
|  |  |  |  |  | 115,200 (high speed) |  | $161$ <br> (CP1L Ltype CPU Unit) |  | 0A hex |
|  |  | 2-6-2 | PC Link | 0 | 0 | Every cycle | 151 | 00 to | 0 hex |
|  |  |  | Unit No. |  | : |  | (CP1L Mtype CPU Unit) | 03 | : |
|  |  |  |  |  | 7 |  | $167$ <br> (CP1L Ltype CPU Unit) |  | 7 hex |


|  | Name |  |  | Default | Settings | When setting is read | Internal | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 2 | 2-7 | PC Link (Master) |  |  |  |  |  |  |  |
|  |  | 2-7-1 | Baud | 9,600 bps (disabled) | 38,400 (standard) | Every cycle | $\begin{aligned} & 145 \\ & \text { (CP1L M- } \\ & \text { type CPU } \\ & \text { Unit) } \end{aligned}$ | $\begin{aligned} & 00 \text { to } \\ & 07 \end{aligned}$ | 00 hex |
|  |  |  |  |  | 115,200 (high speed) |  | $161$ <br> (CP1L Ltype CPU Unit) |  | 0A hex |
|  |  | 2-7-2 | Link Words | 10 (default) | $\begin{aligned} & 1 \\ & \vdots \\ & 10 \text { (default) } \end{aligned}$ | Every cycle | $\begin{aligned} & 150 \\ & \text { (CP1L M- } \\ & \text { type CPU } \\ & \text { Unit) } \end{aligned}$ | $\begin{aligned} & 04 \text { to } \\ & 07 \end{aligned}$ | $\begin{aligned} & 1 \text { hex } \\ & \vdots \\ & 0 \text { or A hex } \end{aligned}$ |
|  |  |  |  |  |  |  | $166$ <br> (CP1L Ltype CPU Unit) |  |  |
|  |  | 2-7-3 | PC Link Mode | ALL | ALL | Every cycle | $\begin{aligned} & 150 \\ & \text { (CP1L M- } \\ & \text { type CPU } \\ & \text { Unit) } \end{aligned}$ | 15 | 0 |
|  |  |  |  |  | Masters |  | $166$ <br> (CP1L Ltype CPU Unit) |  | 1 |

## Serial Port 2 Settings

## Serial Communications Settings

|  | Name |  |  | Default | Settings | When setting is read by CPU Unit | Internal address | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | Communications Settings |  |  | $\begin{aligned} & \hline \text { Standard } \\ & (9600 ; 1,7,2, \mathrm{E}) \end{aligned}$ | Standard (9600; 1,7,2,E) (The standard settings are as follows: 9,600 baud, 1 start bit, 7 -bit data, even parity, and 2 stop bits.) | Every cycle | $\begin{aligned} & 160 \\ & \text { (CP1L M- } \\ & \text { type CPU } \\ & \text { Unit) } \end{aligned}$ | 15 | 0 |
|  |  |  |  | Custom | 1 |  |  |  |
| 2 | Mode |  |  |  | Host Link | Host Link | Every cycle | $\begin{aligned} & 160 \\ & \text { (CP1L M- } \\ & \text { type CPU } \\ & \text { Unit) } \end{aligned}$ | $\begin{aligned} & 08 \text { to } \\ & 11 \end{aligned}$ | 0 hex 5 hex |
|  |  |  |  | NT Link (1:N): 1:N NT Links |  | 2 hex |  |  |  |
|  |  |  |  | RS-232C |  | 3 hex |  |  |  |
|  |  |  |  | ToolBus (peripheral bus) |  | 4 hex |  |  |  |
|  |  |  |  | Serial Gateway |  | 9 hex |  |  |  |
|  |  |  |  | PC Link (Slave) |  | 7 hex |  |  |  |
|  |  |  |  | PC Link (Master) |  | 8 hex |  |  |  |
|  | $2-1$ Host <br>  $2-1-1$ |  |  |  |  |  |  |  |  |
|  |  |  | Baud | 9,600 bps | 300 bps | Every cycle | $\begin{aligned} & 161 \\ & \text { (CP1L M- } \\ & \text { type CPU } \\ & \text { Unit) } \end{aligned}$ | $00 \text { to }$$07$ | 01 hex |
|  |  |  | 600 bps |  | 02 hex |  |  |  |  |
|  |  |  | 1,200 bps |  | 03 hex |  |  |  |  |
|  |  |  | 2,400 bps |  | 04 hex |  |  |  |  |
|  |  |  | 4,800 bps |  | 05 hex |  |  |  |  |
|  |  |  | 9,600 bps |  | 00 or 06 hex |  |  |  |  |
|  |  |  | 19,200 bps |  | 07 hex |  |  |  |  |
|  |  |  | 38,400 bps |  | 08 hex |  |  |  |  |
|  |  |  | 57,600 bps |  | 09 hex |  |  |  |  |
|  |  |  | 115,200 bps |  | 0A hex |  |  |  |  |
|  |  | 2-1-2 |  | Format (data length, stop bits, parity) | 7,2,E: 7-bit data, 2 stop bits, even parity | 7,2,E: 7-bit data, 2 stop bits, even parity | Every cycle | $\begin{aligned} & 160 \\ & \text { (CP1L M- } \\ & \text { type CPU } \\ & \text { Unit) } \end{aligned}$ | $\begin{aligned} & 00 \text { to } \\ & 03 \end{aligned}$ | 0 hex |
|  |  |  |  |  |  | 7,2,0: 7-bit data, 2 stop bits, odd parity |  |  |  | 1 hex |
|  |  |  |  |  |  | 7,2,N: 7-bit data, 2 stop bits, no parity |  |  |  | 2 hex |
|  |  |  |  |  |  | 7,1,E: 7-bit data, 2 stop bits, even parity |  |  |  | 4 hex |
|  |  |  |  |  |  | 7,1,0: 7-bit data, 1 stop bit, odd parity |  |  |  | 5 hex |
|  |  |  |  |  |  | 7,1,N: 7-bit data, 1 stop bit, no parity |  |  |  | 6 hex |
|  |  |  |  |  |  | 8,2,E: 8-bit data, 2 stop bits, even parity |  |  |  | 8 hex |
|  |  |  |  |  |  | 8,2,0: 8-bit data, 2 stop bits, odd parity |  |  |  | 9 hex |
|  |  |  |  |  |  | 8,2,N: 8-bit data, 2 stop bits, no parity |  |  |  | A hex |
|  |  |  | 8,1,E: 8-bit data, 1 stop bit, even parity |  |  | $C$ hex |  |  |  |
|  |  |  | 8,1,O: 8-bit data, 1 stop bit, odd parity |  |  | D hex |  |  |  |
|  |  |  | 8,1,N: 8-bit data, 1 stop bit, no parity |  |  | E hex |  |  |  |
|  |  | 2-1-3 | Unit Number | 0 | 0 | Every cycle | $161$ <br> (CP1L Mtype CPU Unit) | $\begin{aligned} & 00 \text { to } \\ & 07 \end{aligned}$ | 00 hex |
|  |  |  |  |  | : |  |  |  | : |
|  |  |  |  |  | 31 |  |  |  | 1F hex |


|  | Name |  |  | Default | Settings | When setting is read | Internal | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 2 | 2-2 | NT Link (1:N) |  |  |  |  |  |  |  |
|  |  | 2-2-1 | Baud | $\begin{array}{\|l} \hline 9,600 \\ \text { (disabled) } \end{array}$ | 38,400 (standard) | Every cycle | $\begin{aligned} & 161 \\ & \text { (CP1L M- } \\ & \text { type CPU } \\ & \text { Unit) } \end{aligned}$ | $\begin{aligned} & 00 \text { to } \\ & 07 \end{aligned}$ | 00 hex |
|  |  |  |  |  | 115,200 (high speed) |  |  |  | 0A hex |
|  |  | 2-2-2 | NT/PC Link Max: Highest unit number | 0 | 0 | Every cycle | $\begin{aligned} & 166 \\ & \text { (CP1L M- } \\ & \text { type CPU } \\ & \text { Unit) } \end{aligned}$ | $\begin{aligned} & 00 \text { to } \\ & 03 \end{aligned}$ | 0 hex |
|  |  |  |  |  | : |  |  |  | : |
|  |  |  |  |  | 7 |  |  |  | 7 hex |
|  | $2-3$ $R S-23$ <br>  $2-3-1$ |  | RS-232C |  |  |  |  |  |  |
|  |  |  | Baud | 9600 bps | 300 bps | Every cycle | $\begin{aligned} & \text { 161 } \\ & \text { (CP1L M- } \\ & \text { type CPU } \\ & \text { Unit) } \end{aligned}$ | $\begin{aligned} & 00 \text { to } \\ & 07 \end{aligned}$ | 01 hex |
|  |  |  | 600 bps |  | 02 hex |  |  |  |
|  |  |  | 1,200 bps |  | 03 hex |  |  |  |
|  |  |  | 2,400 bps |  | 04 hex |  |  |  |
|  |  |  | 4,800 bps |  | 05 hex |  |  |  |
|  |  |  | 9,600 bps |  | 00 or 06 hex |  |  |  |
|  |  |  | 19,200 bps |  | 07 hex |  |  |  |
|  |  |  | 38,400 bps |  | 08 hex |  |  |  |
|  |  |  | 57,600 bps |  | 09 hex |  |  |  |
|  |  |  | 115,200 bps |  | 0A hex |  |  |  |
|  |  | 2-3-2 |  | Format (data length, stop bits, parity) | 7,2,E: 7-bit data, 2 stop bits, even parity | 7,2,E: 7-bit data, 2 stop bits, even parity | Every cycle | $\begin{aligned} & 160 \\ & \text { (CP1L M- } \\ & \text { type CPU } \\ & \text { Unit) } \end{aligned}$ | $\begin{aligned} & 00 \text { to } \\ & 03 \end{aligned}$ | 0 hex |
|  |  |  |  |  |  | 7,2,0: 7-bit data, 2 stop bits, odd parity |  |  |  | 1 hex |
|  |  |  |  |  |  | 7,2,N: 7-bit data, 2 stop bits, no parity |  |  |  | 2 hex |
|  |  |  |  |  |  | 7,1,E: 7-bit data, 2 stop bits, even parity |  |  |  | 4 hex |
|  |  |  |  |  |  | 7,1,O: 7-bit data, 1 stop bit, odd parity |  |  |  | 5 hex |
|  |  |  |  |  |  | 7,1,N: 7-bit data, 1 stop bit, no parity |  |  |  | 6 hex |
|  |  |  |  |  |  | 8,2,E: 8-bit data, 2 stop bits, even parity |  |  |  | 8 hex |
|  |  |  |  |  |  | 8,2,O: 8-bit data, 2 stop bits, odd parity |  |  |  | 9 hex |
|  |  |  |  |  |  | 8,2,N: 8-bit data, 2 stop bits, no parity |  |  |  | A hex |
|  |  |  | 8,1,E: 8-bit data, 1 stop bit, even parity |  |  | C hex |  |  |  |
|  |  |  | 8,1,0: 8-bit data, 1 stop bit, odd parity |  |  | D hex |  |  |  |
|  |  |  | 8,1,N: 8-bit data, 1 stop bit, no parity |  |  | E hex |  |  |  |
|  |  | 2-3-3 | Start Code | Disable. | Disable. | Every cycle | $\begin{aligned} & 165 \\ & \text { (CP1L M- } \\ & \text { type CPU } \\ & \text { Unit) } \end{aligned}$ | 12 | 0 |
|  |  |  |  |  | Set. |  |  |  | 1 |
|  |  | 2-3-4 | Start Code | 00 hex0x0000 | 0x0000 | Every cycle | $164$ <br> (CP1L Mtype CPU Unit) | $\begin{aligned} & 08 \text { to } \\ & 15 \end{aligned}$ | 00 hex |
|  |  |  |  |  | : |  |  |  | : |
|  |  |  |  |  | 0x00FF |  |  |  | FF hex |
|  |  | 2-3-5 | End Code | Received Bytes: Receive specified number of bytes. | Received Bytes: Receive specified number of bytes. | Every cycle | $\begin{aligned} & 165 \\ & \text { (CP1L M- } \\ & \text { type CPU } \\ & \text { Unit) } \end{aligned}$ | $08 \text { and }$$09$ | 00 |
|  |  |  |  |  | CR,LF |  |  |  | 10 |
|  |  |  |  |  | Set End Code |  |  |  | 01 |


|  | Name |  |  | Default | Settings | When setting is read | Internal | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 2 | 2-3 | 2-3-6 | Received Bytes | 256 bytes | 256 bytes | Every cycle | $\begin{aligned} & 165 \\ & \text { (CP1L M- } \\ & \text { type CPU } \\ & \text { Unit) } \end{aligned}$ | $\begin{aligned} & 00 \text { to } \\ & 07 \end{aligned}$ | 00 hex |
|  |  |  |  |  | 1 byte |  |  |  | 01 hex |
|  |  |  |  |  | : |  |  |  | : |
|  |  |  |  |  | 255 bytes |  |  |  | FF hex |
|  |  | 2-3-7 | Set End Code | $0 \times 0000$ | 0x0000 | Every cycle | $164$ <br> (CP1L M- <br> type CPU <br> Unit) | $\begin{aligned} & 00 \text { to } \\ & 07 \end{aligned}$ | 00 hex |
|  |  |  |  |  | : |  |  |  | : |
|  |  |  |  |  | 0x00FF |  |  |  | FF hex |
|  |  | 2-3-8 | Delay | $0: 0 \times 10 \mathrm{~ms}$ | 0: $0 \times 10 \mathrm{~ms}$ | Every cycle | $\begin{aligned} & 162 \\ & \text { (CP1L M- } \\ & \text { type CPU } \\ & \text { Unit) } \end{aligned}$ | $\begin{aligned} & 00 \text { to } \\ & 15 \end{aligned}$ | 0000 hex |
|  |  |  |  |  | : |  |  |  | : |
|  |  |  |  |  | 9999: $9999 \times 10 \mathrm{~ms}$ |  |  |  | 270F hex |
|  | 2-4 | ToolBus (peripheral bus) |  |  |  |  |  |  |  |
|  |  | 2-4-1 | Baud | 9,600 bps | 9,600 bps | Every cycle | $161$ <br> (CP1L Mtype CPU Unit) | $\begin{aligned} & 00 \text { to } \\ & 07 \end{aligned}$ | 00 or 06 hex |
|  |  |  |  |  | 19,200 bps |  |  |  | 07 hex |
|  |  |  |  |  | 38,400 bps |  |  |  | 08 hex |
|  |  |  |  |  | 57,600 bps |  |  |  | 09 hex |
|  |  |  |  |  | 115,200 bps |  |  |  | 0A hex |
|  | 2-5 | Serial Gateway |  |  |  |  |  |  |  |
|  |  | 2-5-1 | Baud | 9,600 bps | 300 bps | Every cycle | $161$ <br> (CP1L Mtype CPU Unit) | $\begin{aligned} & 00 \text { to } \\ & 07 \end{aligned}$ | 01 hex |
|  |  |  |  |  | 600 bps |  |  |  | 02 hex |
|  |  |  |  |  | 1,200 bps |  |  |  | 03 hex |
|  |  |  |  |  | 2,400 bps |  |  |  | 04 hex |
|  |  |  |  |  | 4,800 bps |  |  |  | 05 hex |
|  |  |  |  |  | 9,600 bps |  |  |  | 00 or 06 hex |
|  |  |  |  |  | 19,200 bps |  |  |  | 07 hex |
|  |  |  |  |  | 38,400 bps |  |  |  | 08 hex |
|  |  |  |  |  | 57,600 bps |  |  |  | 09 hex |
|  |  |  |  |  | 115,200 bps |  |  |  | 0A hex |
|  |  | 2-5-2 | Format (data length, stop bits, parity) | 7,2,E: 7-bit data, 2 stop bits, even parity | 7,2,E: 7-bit data, 2 stop bits, even parity | Every cycle | $\begin{aligned} & 160 \\ & \text { (CP1L M- } \\ & \text { type CPU } \\ & \text { Unit) } \end{aligned}$ | $\begin{aligned} & 00 \text { to } \\ & 03 \end{aligned}$ | 0 hex |
|  |  |  |  |  | 7,2,0: 7-bit data, 2 stop bits, odd parity |  |  |  | 1 hex |
|  |  |  |  |  | 7,2,N: 7-bit data, 2 stop bits, no parity |  |  |  | 2 hex |
|  |  |  |  |  | 7,1,E: 7-bit data, 2 stop bits, even parity |  |  |  | 4 hex |
|  |  |  |  |  | 7,1,O: 7-bit data, 1 stop bit, odd parity |  |  |  | 5 hex |
|  |  |  |  |  | 7,1,N: 7-bit data, 1 stop bit, no parity |  |  |  | 6 hex |
|  |  |  |  |  | 8,2,E: 8-bit data, 2 stop bits, even parity |  |  |  | 8 hex |
|  |  |  |  |  | 8,2,O: 8-bit data, 2 stop bits, odd parity |  |  |  | 9 hex |
|  |  |  |  |  | 8,2,N: 8-bit data, 2 stop bits, no parity |  |  |  | A hex |
|  |  |  |  |  | 8,1,E: 8-bit data, 1 stop bit, even parity |  |  |  | C hex |
|  |  |  |  |  | 8,1,0: 8-bit data, 1 stop bit, odd parity |  |  |  | D hex |
|  |  |  |  |  | 8,1,N: 8-bit data, 1 stop bit, no parity |  |  |  | E hex |
|  |  | 2-5-3 | Response Timeout | $\begin{aligned} & \text { 50: } \\ & 50 \times 100 \mathrm{~ms}= \\ & 5 \mathrm{~s} \end{aligned}$ | 50: $50 \times 100 \mathrm{~ms}=5 \mathrm{~s}$ | Every cycle | $167$ <br> (CP1L M- <br> type CPU <br> Unit) | $\begin{aligned} & 08 \text { to } \\ & 15 \end{aligned}$ | 00 hex |
|  |  |  |  |  | 1: $1 \times 100 \mathrm{~ms}$ |  |  |  | 01 hex |
|  |  |  |  |  | : |  |  |  | : |
|  |  |  |  |  | 255: $255 \times 100 \mathrm{~ms}$ |  |  |  | FF hex |


|  | Name |  |  | Default | Settings | When setting is read by CPU Unit | Internal | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 2 | 2-6 | PC Link (Slave) |  |  |  |  |  |  |  |
|  |  | 2-6-1 | Baud | 9,600 bps (disabled) | 38,400 (standard) | Every cycle | $161$ <br> (CP1L M- <br> type CPU <br> Unit) | $\begin{aligned} & 00 \text { to } \\ & 07 \end{aligned}$ | 00 hex |
|  |  |  |  |  | 115,200 (high speed) |  |  |  | 0A hex |
|  |  | 2-6-2 | PC Link Unit No. | 0 | 0 | Every cycle | $167$ <br> (CP1L M- <br> type CPU <br> Unit) | $\begin{aligned} & 00 \text { to } \\ & 03 \end{aligned}$ | 0 hex |
|  |  |  |  |  | : |  |  |  | : |
|  |  |  |  |  | 7 |  |  |  | 7 hex |
|  | 2-7 | PC Link (Master) |  |  |  |  |  |  |  |
|  |  | 2-7-1 | Baud | 9,600 bps (disabled) | 38,400 (standard) | Every cycle | $161$ <br> (CP1L M- <br> type CPU <br> Unit) | $\begin{aligned} & 00 \text { to } \\ & 07 \end{aligned}$ | 00 hex |
|  |  |  |  |  | 115,200 (high speed) |  |  |  | 0A hex |
|  |  | 2-7-2 | Link Words | 10 (default) | $\begin{aligned} & 1 \\ & \vdots \\ & 10 \text { (default) } \end{aligned}$ | Every cycle | $\begin{aligned} & \text { 166 } \\ & \text { (CP1L M- } \\ & \text { type CPU } \\ & \text { Unit) } \end{aligned}$ | $\begin{aligned} & 04 \text { to } \\ & 07 \end{aligned}$ | $\begin{aligned} & 1 \text { hex } \\ & \vdots \\ & 0 \text { or A hex } \end{aligned}$ |
|  |  | 2-7-3 | PC Link Mode | ALL | ALL | Every cycle | $166$ <br> (CP1L M- <br> type CPU <br> Unit) | 15 | 0 |
|  |  |  |  |  | Masters |  |  |  | 1 |

## Peripheral Service Settings

## Set Time to All Events: Time Setting for Services

|  |  | Name | Default | Settings | When setting is read by CPU Unit | Internal address | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | Set time to all events |  | Default | Default (4\% of cycle time) | At start of operation | 218 | 15 | 0 |
|  |  |  | Use user setting. | 1 |  |  |  |
|  | 1-1 | Time allocated to services |  | $\begin{aligned} & 0: 0 \times 0.1 \mathrm{~ms}= \\ & 0 \mathrm{~ms} \end{aligned}$ | $0: 0 \times 0.1 \mathrm{~ms}=0 \mathrm{~ms}$ | At start of operation | 218 | $00 \text { to }$$07$ | 00 hex |
|  |  |  | : |  | : |  |  |  |
|  |  |  | 255: $255 \times 0.1 \mathrm{~ms}$ |  | FF hex |  |  |  |

## Built-in Input Settings

High Speed Counter Settings

|  | Name |  |  | Default | Settings | When setting is read by CPU Unit | Internal address | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | Use high speed counter 0 |  |  | Do not use. | Do not use. | When power is turned ON | 50 | $\begin{aligned} & 12 \text { to } \\ & 15 \end{aligned}$ | 0 hex |
|  |  |  |  | Use. | 1 hex |  |  |  |
|  | 1-1 | Counting mode |  |  | Linear mode | Linear mode | At start of operation | 50 | $\begin{aligned} & 08 \text { to } \\ & 11 \end{aligned}$ | 0 hex |
|  |  |  |  | Circular mode |  | 1 hex |  |  |  |
|  |  | 1-1-1 | Circular Max. | 0 | 0 | At start of operation | 52 and 51 | $\begin{aligned} & 00 \text { to } \\ & 15 \end{aligned}$ | $\begin{aligned} & 0000 \\ & 0000 \text { hex } \end{aligned}$ |
|  |  |  | C |  | : |  |  |  | : |
|  |  |  |  |  | 4,294,967,295 |  |  |  | FFFF FFFF hex |
|  | 1-2 | Reset |  | Z phase, software reset | Z phase, software reset | When power is turned ON | 50 | 04 to 07 | 0 hex |
|  |  |  |  | Software reset | 1 hex |  |  |  |  |
|  |  |  |  | Z phase, software reset (comparing) | 2 hex |  |  |  |  |
|  |  |  |  | Software reset (comparing) | 3 hex |  |  |  |  |
|  | 1-3 | Input Setting |  |  | Differential phase input | Differential phase input | When power is turned ON | 50 | $\begin{aligned} & 00 \text { to } \\ & 03 \end{aligned}$ | 0 hex |
|  |  |  |  | Pulse + direction input |  | 1 hex |  |  |  |
|  |  |  |  | Up/Down input |  | 2 hex |  |  |  |
|  |  |  |  | Increment pulse input |  | 3 hex |  |  |  |
| 2 | Use high speed counter 1 |  |  |  | Do not use. | Do not use. | When power is turned ON | 53 | $\begin{aligned} & 12 \text { to } \\ & 15 \end{aligned}$ | 0 hex |
|  |  |  |  | Use. |  | 1 hex |  |  |  |
|  | 2-1 | Counting mode |  |  | Linear mode | Linear mode | At start of operation | 53 | $\begin{aligned} & 08 \text { to } \\ & 11 \end{aligned}$ | 0 hex |
|  |  |  |  | Circular mode |  | 1 hex |  |  |  |
|  |  | 2-1-1 | Circular Max. | 0 | 0 | At start of operation | 55 and 54 | $\begin{aligned} & 00 \text { to } \\ & 15 \end{aligned}$ | $\begin{aligned} & 0000 \\ & 0000 \text { hex } \end{aligned}$ |
|  |  |  |  |  | : |  |  |  | : |
|  |  |  |  |  | 4,294,967,295 |  |  |  | FFFF <br> FFFF hex |
|  | 2-2 | Reset |  | Z phase, software reset | Z phase, software reset | When power is turned ON | 53 | 04 to 07 | 0 hex |
|  |  |  |  | Software reset | 1 hex |  |  |  |  |
|  |  |  |  | Z phase, software reset (comparing) | 2 hex |  |  |  |  |
|  |  |  |  | Software reset (comparing) | 3 hex |  |  |  |  |
|  | 2-3 | Input Setting |  |  | Differential phase input | Differential phase input | When power is turned ON | 53 | $\begin{aligned} & 00 \text { to } \\ & 03 \end{aligned}$ | 0 hex |
|  |  |  |  | Pulse + direction input |  | 1 hex |  |  |  |
|  |  |  |  | Up/Down input |  | 2 hex |  |  |  |
|  |  |  |  | Increment pulse input |  | 3 hex |  |  |  |


|  | Name |  |  | Default | Settings | When setting is read by CPU Unit | Internal address | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 3 | Use high speed counter 2 |  |  | Do not use. | Do not use. | When power is turned ON | 95 | $\begin{aligned} & 12 \text { to } \\ & 15 \end{aligned}$ | $\begin{array}{\|l\|} \hline 0 \text { hex } \\ \hline 1 \text { hex } \\ \hline \end{array}$ |
|  |  |  |  | Use. |  |  |  |  |
|  | 3-1 | Counting mode |  |  | Linear mode | Linear mode | At start of operation | 95 | $\begin{aligned} & 08 \text { to } \\ & 11 \end{aligned}$ | $\begin{array}{\|l\|} \hline 0 \text { hex } \\ \hline 1 \text { hex } \\ \hline \end{array}$ |
|  |  |  |  | Circular mode |  |  |  |  |  |  |
|  |  | 3-1-1 | Circular Max. Count | 0 | 0 | At start of operation | 97 and 96 | $\begin{aligned} & 00 \text { to } \\ & 15 \end{aligned}$ | $\begin{array}{\|l\|} \hline 0000 \\ 0000 \text { hex } \\ \hline \end{array}$ |  |
|  |  |  |  |  | : |  |  |  | : |  |
|  |  |  |  |  | 4,294,967,295 |  |  |  | FFFF <br> FFFF hex |  |
|  | 3-2 | Reset |  | Z phase, software reset | Z phase, software reset | When power is turned ON | 95 | $\begin{aligned} & 04 \text { to } \\ & 07 \end{aligned}$ | 0 hex |  |
|  |  |  |  | Software reset | 1 hex |  |  |  |  |  |
|  |  |  |  | Z phase, software reset (comparing) | 2 hex |  |  |  |  |  |
|  |  |  |  | Software reset (comparing) | 3 hex |  |  |  |  |  |
|  | 3-3 | Input Setting |  |  | Differential phase input | Differential phase input | When power is turned ON | 95 | $\begin{aligned} & 00 \text { to } \\ & 03 \end{aligned}$ | 0 hex |
|  |  |  |  | Pulse + direction input |  | 1 hex |  |  |  |  |
|  |  |  |  | Up/Down input |  | 2 hex |  |  |  |  |
|  |  |  |  | Increment pulse input |  | 3 hex |  |  |  |  |
| 4 | Use high speed counter 3 |  |  |  | Do not use. | Do not use. | When power is turned ON | 98 | $\begin{aligned} & 12 \text { to } \\ & 15 \end{aligned}$ | 0 hex |
|  |  |  |  | Use. |  | 1 hex |  |  |  |  |
|  | 4-1 | Counting mode |  |  | Linear mode | Linear mode | At start of operation | 98 | $\begin{aligned} & 08 \text { to } \\ & 11 \end{aligned}$ | 0 hex |
|  |  |  |  | Circular mode |  | 1 hex |  |  |  |  |
|  |  | 4-1-1 | Circular Max. | 0 | 0 | At start of operation | $\begin{aligned} & 100 \text { and } \\ & 99 \end{aligned}$ | $\begin{aligned} & 00 \text { to } \\ & 15 \end{aligned}$ | $\begin{aligned} & 0000 \\ & 0000 \text { hex } \end{aligned}$ |  |
|  |  |  | Count |  | . |  |  |  | : |  |
|  |  |  |  |  | 4,294,967,295 |  |  |  | FFFF FFFF hex |  |
|  | 4-2 | Reset |  | Z phase, software reset | Z phase, software reset | When power is turned ON | 98 | $\begin{aligned} & 04 \text { to } \\ & 07 \end{aligned}$ | 0 hex |  |
|  |  |  |  | Software reset | 1 hex |  |  |  |  |  |
|  |  |  |  | Z phase, software reset (comparing) | 2 hex |  |  |  |  |  |
|  |  |  |  | Software reset (comparing) | 3 hex |  |  |  |  |  |
|  | 4-3 | Input Setting |  |  | Differential phase input | Differential phase input | When power is turned ON | 98 | $\begin{aligned} & 00 \text { to } \\ & 03 \end{aligned}$ | 0 hex |
|  |  |  |  | Pulse + direction input |  | 1 hex |  |  |  |  |
|  |  |  |  | Up/Down input |  | 2 hex |  |  |  |  |
|  |  |  |  | Increment pulse input |  | 3 hex |  |  |  |  |

## Interrupt Input Settings

|  | Name | Default | Settings | When setting is read by CPU Unit | Internal address | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | $\begin{array}{\|l\|} \hline \text { INO } \\ \text { (ClO 0.04) } \end{array}$ | Normal | Normal | When power is turned ON | 60 | $\begin{aligned} & 00 \text { to } \\ & 03 \end{aligned}$ | 0 hex |
|  |  |  | Interrupt |  |  |  | 1 hex |
|  |  |  | Quick |  |  |  | 2 hex |
| 2 | $\begin{array}{\|l\|} \hline \text { IN1 } \\ \text { (CIO 0.05) } \end{array}$ | Normal | Normal | When power is turned ON | 60 | $04 \text { to }$$07$ | 0 hex |
|  |  |  | Interrupt |  |  |  | 1 hex |
|  |  |  | Quick |  |  |  | 2 hex |
| 3 | $\begin{aligned} & \text { IN2 } \\ & \text { (CIO 0.06) } \end{aligned}$ | Normal | Normal | When power is turned ON | 60 | $\begin{aligned} & 08 \text { to } \\ & 11 \end{aligned}$ | 0 hex |
|  |  |  | Interrupt |  |  |  | 1 hex |
|  |  |  | Quick |  |  |  | 2 hex |
| 4 | $\begin{aligned} & \text { IN3 } \\ & \text { (CIO 0.07) } \end{aligned}$ | Normal | Normal | When power is turned ON | 60 | $\begin{aligned} & 12 \text { to } \\ & 15 \end{aligned}$ | 0 hex |
|  |  |  | Interrupt |  |  |  | 1 hex |
|  |  |  | Quick |  |  |  | 2 hex |
| 5 | $\begin{array}{\|l} \hline \text { IN4 } \\ \text { (CIO 0.08) } \end{array}$ | Normal | Normal | When power is turned ON | 59 | $\begin{array}{\|l} \hline 00 \text { to } \\ 03 \end{array}$ | 0 hex |
|  |  |  | Interrupt |  |  |  | 1 hex |
|  |  |  | Quick |  |  |  | 2 hex |
| 6 | $\begin{array}{\|l} \hline \text { IN5 } \\ \text { (CIO 0.09) } \end{array}$ | Normal | Normal | When power is turned ON | 59 | $04 \text { to }$$07$ | 0 hex |
|  |  |  | Interrupt |  |  |  | 1 hex |
|  |  |  | Quick |  |  |  | 2 hex |

## Pulse Output 0 Settings

## Base Settings

|  | Name | Default | Settings | When setting is read by CPU Unit | Internal address | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | Undefined Origin (operation for limit signal turning ON) | Hold | Hold | At start of operation | 268 | $\begin{aligned} & 12 \text { to } \\ & 15 \end{aligned}$ | 0 hex |
|  |  |  | Undefined |  |  |  | 1 hex |
| 2 | Limited Input Signal Operation | Search Only | Search Only | When power is turned ON | 256 | 04 to 07 | 0 hex |
|  |  |  | Always |  |  |  | 1 hex |
| 3 | Limit Input Signal | NC | NC | At start of operation | 268 | $\begin{aligned} & 00 \text { to } \\ & 03 \end{aligned}$ | 0 hex |
|  |  |  | NO |  |  |  | 1 hex |
| 4 | Search/Return Initial Speed | 0 pps | 0 pps | At start of operation | $\begin{aligned} & 259 \text { and } \\ & 258 \end{aligned}$ | $\begin{aligned} & 00 \text { to } \\ & 15 \end{aligned}$ | $\begin{aligned} & 00000000 \\ & \text { hex } \end{aligned}$ |
|  |  |  | : |  |  |  | : |
|  |  |  | 100,000 pps |  |  |  | $\begin{aligned} & 0001 \text { 86A0 } \\ & \text { hex } \\ & \text { 000F } 4240 \\ & \text { hex } \end{aligned}$ |
| 5 | Speed Curve | Trapezium | Trapezium | When power is turned ON | 256 | $\begin{aligned} & 12 \text { to } \\ & 15 \end{aligned}$ | 0 hex |
|  |  |  | S-shaped |  |  |  | 1 hex |

## Define Origin Operation Settings: Origin Search Settings

|  |  | Name | Default | Settings | When setting is read by CPU Unit | Internal address | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | Use define origin operation |  | Do not use. | Do not use. | When power is turned ON | 256 | $\begin{aligned} & 00 \text { to } \\ & 03 \end{aligned}$ | 0 hex |
| 1 |  |  | Use. | 1 hex |  |  |  |
|  | 1-1 | Search Direction |  | CW | CW | At start of operation | 257 | $\begin{aligned} & 12 \text { to } \\ & 15 \end{aligned}$ | 0 hex |
|  |  |  | CCW |  | 1 hex |  |  |  |
|  | 1-2 | Detection Method | Method 0 | Method 0 | At start of operation | 257 | $\begin{aligned} & 08 \text { to } \\ & 11 \end{aligned}$ | 0 hex |
|  |  |  |  | Method 1 |  |  |  | 1 hex |
|  |  |  |  | Method 2 |  |  |  | 2 hex |
|  | 1-3 | Search Operation | Inverse 1 | Inverse 1 | At start of operation | 257 | $04 \text { to }$$07$ | 0 hex |
|  |  |  |  | Inverse 2 |  |  |  | 1 hex |
|  | 1-4 | Operation Mode | Mode 0 | Mode 0 | At start of operation | 257 | $\begin{aligned} & 00 \text { to } \\ & 03 \end{aligned}$ | 0 hex |
|  |  |  |  | Mode 1 |  |  |  | 1 hex |
|  |  |  |  | Mode 2 |  |  |  | 2 hex |
|  | 1-5 | Origin Input Signal (X/XA CPU Units) | NC | NC | Unit version 1.0 and earlier: At start of operation <br> Unit version 1.1 and later: When power is turned ON | 268 | $\begin{aligned} & 08 \text { to } \\ & 11 \end{aligned}$ | 0 hex |
|  |  |  |  | NO |  |  |  | 1 hex |
|  |  | Origin Input Signal (Y CPU Units) |  | NC (line driver) | At start of operation |  |  | 2 hex |
|  |  |  |  | NO (line driver) |  |  |  | 3 hex |
|  | 1-6 | Proximity Input Signal | NC | NC | At start of operation | 268 | $\begin{aligned} & 04 \text { to } \\ & 07 \end{aligned}$ | 0 hex |
|  |  |  |  | NO |  |  |  | 1 hex |
|  | 1-7 | Search High Speed | 0 pps (disabled) | 1 pps | At start of operation | $\begin{aligned} & 261 \text { and } \\ & 260 \end{aligned}$ | $\begin{aligned} & 00 \text { to } \\ & 15 \end{aligned}$ | $\begin{aligned} & 00000001 \\ & \text { hex } \end{aligned}$ |
|  |  |  |  | : |  |  |  | : |
|  |  |  |  | 100,000 pps |  |  |  | $\begin{aligned} & 0001 \text { 86A0 } \\ & \text { hex } \\ & 000 \mathrm{~F} 4240 \\ & \text { hex } \end{aligned}$ |
|  | 1-8 | Search Proximity Speed | 0 pps (disabled) | 1 pps | At start of operation | $\begin{aligned} & 263 \text { and } \\ & 262 \end{aligned}$ | $\begin{aligned} & 00 \text { to } \\ & 15 \end{aligned}$ | $\begin{array}{\|l} 00000001 \\ \text { hex } \end{array}$ |
|  |  |  |  | : |  |  |  | : |
|  |  |  |  | 100,000 pps |  |  |  | $\begin{aligned} & 0001 \text { 86A0 } \\ & \text { hex } \\ & 000 \mathrm{~F} 4240 \\ & \text { hex } \end{aligned}$ |
|  | 1-9 | Search Compensation Value | 0 pps | -2,147,483,648 | At start of operation | $\begin{aligned} & 265 \text { and } \\ & 264 \end{aligned}$ | $\begin{aligned} & 00 \text { to } \\ & 15 \end{aligned}$ | $\begin{aligned} & 80000000 \\ & \text { hex } \end{aligned}$ |
|  |  |  |  | : |  |  |  | : |
|  |  |  |  | 0 |  |  |  | $\begin{aligned} & 00000000 \\ & \text { hex } \end{aligned}$ |
|  |  |  |  | : |  |  |  | : |
|  |  |  |  | +2,147,483,647 |  |  |  | 7FFF FFFF hex |
|  | 1-10 | Search Acceleration Ratio | 0 (disabled) | 1 (pulses/4 ms) | At start of operation | 266 | $\begin{aligned} & 00 \text { to } \\ & 15 \end{aligned}$ | 0001 hex |
|  |  |  |  | : |  |  |  | : |
|  |  |  |  | 65,535 (pulses/4 ms) |  |  |  | FFFF hex |
|  | 1-11 | Search Deceleration Ratio | 0 (disabled) | 1 (pulses/4 ms) | At start of operation | 267 | $\begin{aligned} & 00 \text { to } \\ & 15 \end{aligned}$ | 0001 hex |
|  |  |  |  | : |  |  |  | : |
|  |  |  |  | 65,535 (pulses/4 ms) |  |  |  | FFFF hex |
|  | 1-12 | Positioning Monitor Time | 0 ms | 0 ms | At start of operation | 269 | $\begin{aligned} & 00 \text { to } \\ & 15 \end{aligned}$ | 0000 hex |
|  |  |  |  | : |  |  |  | : |
|  |  |  |  | 9,999 ms |  |  |  | 270F hex |

Origin Return Settings

|  | Name | Default | Settings | When setting is read by CPU Unit | Internal address | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | Speed | 0 pps (disabled) | 1 pps | At start of operation | $\begin{aligned} & 271 \text { and } \\ & 270 \end{aligned}$ | 00 to 15 | $\begin{array}{\|l} \hline 00000001 \\ \text { hex } \\ \hline \end{array}$ |
|  |  |  | : |  |  |  | : |
|  |  |  | 100,000 pps |  |  |  | $\begin{aligned} & 0001 \text { 86A0 } \\ & \text { hex } \\ & 000 \mathrm{~F} 4240 \\ & \text { hex } \end{aligned}$ |
| 2 | Acceleration Ratio | 0 (disabled) | 1 (pulses/4 ms) | At start of operation | 272 | 00 to 15 | 0001Hex |
|  |  |  | : |  |  |  | : |
|  |  |  | $\begin{aligned} & 65535 \\ & \text { (pulses/4 ms) } \end{aligned}$ |  |  |  | FFFF hex |
| 3 | Deceleration Ratio | 0 (disabled) | 1 (pulses/4 ms) | At start of operation | 273 | 00 to 15 | 0001 hex |
|  |  |  | : |  |  |  | : |
|  |  |  | 65535 <br> (pulses/4 ms) |  |  |  | FFFF hex |

## Pulse Output 1 Settings

## Base Settings

|  | Name | Default | Settings | When setting is read by CPU Unit | Internal address | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | Undefined Origin (operation for limit signal turning ON) | Hold | Hold | At start of operation | 286 | $\begin{array}{\|l\|} \hline 12 \text { to } \\ 15 \end{array}$ | 0 hex |
|  |  |  | Undefined |  |  |  | 1 hex |
| 2 | Limited Input Signal Operation | Search Only | Search Only | When power is turned ON | 274 | $\begin{array}{\|l} \hline 04 \text { to } \\ 07 \end{array}$ | 0 hex |
|  |  |  | Always |  |  |  | 1 hex |
| 3 | Limit Input Signal | NC | NC | At start of operation | 286 | $\begin{aligned} & 00 \text { to } \\ & 03 \end{aligned}$ | 0 hex |
|  |  |  | NO |  |  |  | 1 hex |
| 4 | Search/Return Initial Speed | 0 pps | 0 pps | At start of operation | $\begin{aligned} & 277 \text { and } \\ & 276 \end{aligned}$ | $\begin{aligned} & \hline 00 \text { to } \\ & 15 \end{aligned}$ | $\begin{aligned} & \hline \begin{array}{l} 0000 \\ \text { hex } \end{array} \\ & \hline \end{aligned}$ |
|  |  |  |  |  |  |  |  |
|  |  |  | 100,000 pps |  |  |  | $\begin{aligned} & \hline 0001 \text { 86A0 } \\ & \text { hex } \\ & \text { 000F } 4240 \\ & \text { hex } \end{aligned}$ |
| 5 | Speed Curve | Trapezium | Trapezium | When power is turned ON | 274 | $\begin{aligned} & \hline 12 \text { to } \\ & 15 \end{aligned}$ | 0 hex |
|  |  |  | S-shaped |  |  |  | 1 hex |

## Define Origin Operation Settings: Origin Search Settings

|  |  | Name | Default | Settings | When setting is read by CPU Unit | Internal address | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | Use define origin operation |  | Do not use. | Do not use. | When power is turned ON | 274 | $\begin{aligned} & 00 \text { to } \\ & 03 \end{aligned}$ | 0 hex |
|  |  |  | Use. | 1 hex |  |  |  |
|  | 1-1 | Search Direction |  | CW | CW | At start of operation | 275 | $\begin{aligned} & 12 \text { to } \\ & 15 \end{aligned}$ | 0 hex |
|  |  |  | CCW |  | 1 hex |  |  |  |
|  | 1-2 | Detection Method | Method 0 | Method 0 | At start of operation | 275 | $\begin{aligned} & 08 \text { to } \\ & 11 \end{aligned}$ | 0 hex |
|  |  |  |  | Method 1 |  |  |  | 1 hex |
|  |  |  |  | Method 2 |  |  |  | 2 hex |
|  | 1-3 | Search Operation | Inverse 1 | Inverse 1 | At start of operation | 275 | $\begin{aligned} & 04 \text { to } \\ & 07 \end{aligned}$ | 0 hex |
|  |  |  |  | Inverse 2 |  |  |  | 1 hex |
|  | 1-4 | Operation Mode | Mode 0 | Mode 0 | At start of operation | 275 | $\begin{aligned} & 00 \text { to } \\ & 03 \end{aligned}$ | 0 hex |
|  |  |  |  | Mode 1 |  |  |  | 1 hex |
|  |  |  |  | Mode 2 |  |  |  | 2 hex |
|  | 1-5 | Origin Input Signal (X/XA CPU Units) | NC | NC | Unit version 1.0 and earlier: At start of operation <br> Unit version 1.1 and later: When power is turned ON | 286 | $\begin{aligned} & 08 \text { to } \\ & 11 \end{aligned}$ | 0 hex |
|  |  |  |  | NO |  |  |  | 1 hex |
|  |  | Origin Input Signal <br> (Y CPU Units) |  | NC (line driver) | At start of operation |  |  | 2 hex |
|  |  |  |  | NO (line driver) |  |  |  | 3 hex |
|  | 1-6 | Proximity Input Signal | NC | NC | At start of operation | 286 | $\begin{aligned} & 04 \text { to } \\ & 07 \end{aligned}$ | 0 hex |
|  |  |  |  | NO |  |  |  | 1 hex |
|  | 1-7 | Search High Speed | 0 pps (disabled) | 1 pps | At start of operation | $\begin{aligned} & 279 \text { and } \\ & 278 \end{aligned}$ | $\begin{aligned} & 00 \text { to } \\ & 15 \end{aligned}$ | $\begin{aligned} & 00000001 \\ & \text { hex } \end{aligned}$ |
|  |  |  |  | : |  |  |  | : |
|  |  |  |  | 100,000 pps |  |  |  | $\begin{aligned} & 0001 \text { 86A0 } \\ & \text { hex } \\ & \text { 000F } 4240 \\ & \text { hex } \end{aligned}$ |
|  | 1-8 | Search Proximity Speed | 0 pps (disabled) | 1 pps | At start of operation | $\begin{aligned} & 281 \text { and } \\ & 280 \end{aligned}$ | $\begin{aligned} & 00 \text { to } \\ & 15 \end{aligned}$ | $\begin{aligned} & 00000001 \\ & \text { hex } \end{aligned}$ |
|  |  |  |  | : |  |  |  | : |
|  |  |  |  | 100,000 pps |  |  |  | $\begin{aligned} & 0001 \text { 86A0 } \\ & \text { hex } \\ & \text { 000F } 4240 \\ & \text { hex } \end{aligned}$ |
|  | 1-9 | Search Compensation Value | 0 pps | -2,147,483,648 | At start of operation | $\begin{aligned} & 283 \text { and } \\ & 282 \end{aligned}$ | $\begin{aligned} & 00 \text { to } \\ & 15 \end{aligned}$ | $\begin{aligned} & 80000000 \\ & \text { hex } \end{aligned}$ |
|  |  |  |  | : |  |  |  | : |
|  |  |  |  | 0 |  |  |  | $\begin{array}{\|l\|} \hline 00000000 \\ \text { hex } \end{array}$ |
|  |  |  |  | : |  |  |  | : |
|  |  |  |  | +2,147,483,647 |  |  |  | 7FFF FFFF hex |
|  | 1-10 | Search Acceleration Ratio | 0 (disabled) | 1 (pulses/4 ms) | At start of operation | 284 | $\begin{array}{\|l} 00 \text { to } \\ 15 \end{array}$ | 0001 hex |
|  |  |  |  | : |  |  |  | : |
|  |  |  |  | 65,535 (pulses/4 ms) |  |  |  | FFFF hex |
|  | 1-11 | Search Deceleration Ratio | 0 (disabled) | 1 (pulses/4 ms) | At start of operation | 285 | $\begin{aligned} & 00 \text { to } \\ & 15 \end{aligned}$ | 0001 hex |
|  |  |  |  | : |  |  |  | : |
|  |  |  |  | 65,535 (pulses/4 ms) |  |  |  | FFFF hex |
|  | 1-12 | Positioning Monitor Time | 0 ms | 0 ms | At start of operation | 287 | $\begin{aligned} & 00 \text { to } \\ & 15 \end{aligned}$ | 0000 hex |
|  |  |  |  | : |  |  |  | : |
|  |  |  |  | 9,999 ms |  |  |  | 270F hex |

## Origin Return Settings

|  | Name | Default | Settings | When setting is read by CPU Unit | Internal address | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | Speed | 0 pps (disabled) | 1 pps | At start of operation | $\begin{aligned} & 289 \text { and } \\ & 288 \end{aligned}$ | 00 to 15 | $\begin{aligned} & 00000001 \\ & \text { hex } \end{aligned}$ |
|  |  |  | : |  |  |  | : |
|  |  |  | 100,000 pps |  |  |  | $\begin{aligned} & 0001 \text { 86A0 } \\ & \text { hex } \\ & 000 \mathrm{~F} 4240 \\ & \text { hex } \end{aligned}$ |
| 2 | Acceleration Ratio | 0 (disabled) | 1 (pulses/4 ms) | At start of operation | 290 | 00 to 15 | 0001Hex |
|  |  |  | : |  |  |  | : |
|  |  |  | $\begin{aligned} & 65535 \\ & \text { (pulses/4 ms) } \end{aligned}$ |  |  |  | FFFF hex |
| 3 | Deceleration Ratio | 0 (disabled) | 1 (pulses/4 ms) | At start of operation | 291 | 00 to 15 | 0001 hex |
|  |  |  |  |  |  |  | : |
|  |  |  | $\begin{aligned} & 65535 \\ & \text { (pulses/4 ms) } \end{aligned}$ |  |  |  | FFFF hex |

## Inverter Positioning 0

## Basic Settings

|  | Name | Default | Settings | When setting is read by CPU Unit | Internal address | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | Use inverter positioning | Do not use | Use | When power is turned ON | 416 | 00 to 03 | 0 hex |
|  |  |  | Do not use |  |  |  | 1 hex |
| 2 | Gain | 0: 10 (0.1 increments) | 0: 10 (0.1 increments) | When power is turned ON | 418 | 00 to 15 | 0000 hex |
|  |  |  | 1 (0.1 increments) |  |  |  | 0001 hex |
|  |  |  | : |  |  |  | : |
|  |  |  | $\begin{aligned} & 65,535 \\ & \text { (0.1 increments) } \end{aligned}$ |  |  |  | FFFF hex |
| 3 | In-position range | 0: 1 | 0: 1 | When power is turned ON | 419 | 00 to 15 | 0000 hex |
|  |  |  | 1 |  |  |  | 0001 hex |
|  |  |  | : |  |  |  | : |
|  |  |  | 65,535 |  |  |  | FFFF hex |
| 4 | Min. output value | 0: 1 | 0: 1 | When power is turned ON | 420 | 00 to 15 | 0000 hex |
|  |  |  | 1 |  |  |  | 0001 hex |
|  |  |  | : |  |  |  | : |
|  |  |  | 65,535 |  |  |  | FFFF hex |
| 5 | Max. output value | 0: 2,000,000 | 0: 2,000,000 | When power is turned ON | 421, 422 | 00 to 15 | $\begin{array}{\|l} 00000000 \\ \text { hex } \end{array}$ |
|  |  |  | 1 |  |  |  | $\begin{array}{\|l} \hline 00000001 \\ \text { hex } \\ \hline \end{array}$ |
|  |  |  | : |  |  |  | : |
|  |  |  | 4,294,967,295 |  |  |  | FFFFFFFF hex |
| 6 | Error counter overflow detection value | 0: 10,000 | 0: 10,000 | When power is turned ON | 423 | 00 to 15 | 0000 hex |
|  |  |  | 1 |  |  |  | 0001 hex |
|  |  |  | : |  |  |  | : |
|  |  |  | 32,767 |  |  |  | 7FFF hex |
| 7 | Error counter alarm detection value | 0: 10,000 | 0: 10,000 | When power is turned ON | 424 | 00 to 15 | 0000 hex |
|  |  |  | 1 |  |  |  | 0001 hex |
|  |  |  | : |  |  |  | : |
|  |  |  | 32,767 |  |  |  | 7FFF hex |


|  | Name | Default | Settings | When setting is read by CPU Unit | Internal address | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 8 | Error counter cycle | 0: 3 (4-ms increments) | 0: 3 (4-ms increments) | When power is turned ON | 417 | 00 to 07 | 00 hex |
|  |  |  | 1 (4-ms increments) |  |  |  | 01 hex |
|  |  |  | : |  |  |  | : |
|  |  |  | 255 (4-ms increments) |  |  |  | FF hex |
| 9 | Power Supply Freq. for One Motor Revolution per Sec. | 0 (0.1-Hz increments) | 0 (0.1-Hz increments) | When power is turned ON | 436 | 00 to 15 | 0000 hex |
|  |  |  |  |  |  |  | : |
|  |  |  | $\begin{array}{\|l} \hline 65,535 \mathrm{~Hz}(0.1-\mathrm{Hz} \\ \text { increments) } \end{array}$ |  |  |  | FFFF hex |
| 10 | Number of Encoder Pulses for One Motor Revolution | 0 | 0 | When power is turned ON | 437 | 00 to 15 | 0000 hex |
|  |  |  | : |  |  |  | : |
|  |  |  | 65,535 |  |  |  | FFFF hex |

## Operation Adjustment Settings

|  | Name | Default | Settings | When setting is read by CPU Unit | Internal address | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | Limit output during acceleration and constant speed | Do not use | Use | When power is turned ON | 432 | 00 to 03 | 0 hex |
|  |  |  | Do not use |  |  |  | 1 hex |
| 2 | Limit output during deceleration and when stopped | Do not use | Use | When power is turned ON | 432 | 04 to 07 | 0 hex |
|  |  |  | Do not use |  |  |  | 1 hex |
| 3 | Output coefficient during acceleration and constant speed | 0: 6 (0.01 increments) | 0: 6 (0.01 increments) | When power is turned ON | 433 | 00 to 07 | 0 hex |
|  |  |  | 1 (0.01 increments) |  |  |  | 1 hex |
|  |  |  |  |  |  |  | : |
|  |  |  | 255 (0.01 increments) |  |  |  | FF hex |
| 4 | Output coefficient during deceleration | $\begin{aligned} & \text { 0: } 96 \text { (0.01 } \\ & \text { increments) } \end{aligned}$ | 0: 96 (0.01 increments) | When power is turned ON | 434 | 00 to 07 | 0 hex |
|  |  |  | 1 (0.01 increments) |  |  |  | 1 hex |
|  |  |  |  |  |  |  | : |
|  |  |  | 255 (0.01 increments) |  |  |  | FF hex |
| 5 | Output coefficient after pulse output | $\begin{aligned} & \text { 0: } 50 \text { (0.01 } \\ & \text { increments) } \end{aligned}$ | 0: 50 (0.01 increments) | When power is turned ON | 435 | 00 to 07 | 0 hex |
|  |  |  | 1 (0.01 increments) |  |  |  | 1 hex |
|  |  |  | : |  |  |  | : |
|  |  |  | 255 (0.01 increments) |  |  |  | FF hex |

## Inverter Positioning 1

## Basic Settings

|  | Name | Default | Settings | When setting is read by CPU Unit | Internal address | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | Use inverter positioning | Do not use | Use | When power is turned ON | 416 | 08 to 11 | 0 hex |
|  |  |  | Do not use |  |  |  | 1 hex |
| 2 | Gain | 0: 10 (0.1 increments) | 0: 10 (0.1 increments) | When power is turned ON | 425 | 00 to 15 | 0000 hex |
|  |  |  | 1 (0.1 increments) |  |  |  | 0001 hex |
|  |  |  | : |  |  |  | : |
|  |  |  | $\begin{aligned} & 65,535 \\ & \text { (0.1 increments) } \end{aligned}$ |  |  |  | FFFF hex |


|  | Name | Default | Settings | When setting is read by CPU Unit | Internal address | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 3 | In-position range | 0: 1 | 0: 1 | When power is turned ON | 426 | 00 to 15 | 0000 hex |
|  |  |  | 1 |  |  |  | 0001 hex |
|  |  |  | : |  |  |  | : |
|  |  |  | 65,535 |  |  |  | FFFF hex |
| 4 | Min. output value | 0: 1 | 0: 1 | When power is turned ON | 427 | 00 to 15 | 0000 hex |
|  |  |  | 1 |  |  |  | 0001 hex |
|  |  |  | : |  |  |  | : |
|  |  |  | 65,535 |  |  |  | FFFF hex |
| 5 | Max. output value | 0: 2,000,000 | 0: 2,000,000 | When power is turned ON | 429, 428 | 00 to 15 | $\begin{array}{\|l\|} \hline 00000000 \\ \text { hex } \end{array}$ |
|  |  |  | 1 |  |  |  | $\begin{aligned} & 00000001 \\ & \text { hex } \end{aligned}$ |
|  |  |  | : |  |  |  | : |
|  |  |  | 4,294,967,295 |  |  |  | FFFFFFFF hex |
| 6 | Error counter overflow detection value | 0: 10,000 | 0: 10,000 | When power is turned ON | 430 | 00 to 15 | 0000 hex |
|  |  |  | 1 |  |  |  | 0001 hex |
|  |  |  | : |  |  |  | : |
|  |  |  | 32,767 |  |  |  | 7FFF hex |
| 7 | Error counter alarm detection value | 0: 10,000 | 0: 10,000 | When power is turned ON | 431 | 00 to 15 | 0000 hex |
|  |  |  | 1 |  |  |  | 0001 hex |
|  |  |  | , |  |  |  | - |
|  |  |  | 32,767 |  |  |  | 7FFF hex |
| 8 | Error counter cycle | 0: 3 (4-ms increments) | 0: 3 (4-ms increments) | When power is turned ON | 417 | 00 to 07 | 00 hex |
|  |  |  | 1 (4-ms increments) |  |  |  | 01 hex |
|  |  |  | : |  |  |  | : |
|  |  |  | 255 (4-ms increments) |  |  |  | FF hex |
| 9 | Power Supply Freq. for One Motor Revolution per Sec. | 0 (0.1-Hz increments) | 0 (0.1-Hz increments) | When power is turned ON | 438 | 00 to 15 | 0000 hex |
|  |  |  | : |  |  |  | : |
|  |  |  | $65,535 \mathrm{~Hz}(0.1-\mathrm{Hz}$ increments) |  |  |  | FFFF hex |
| 10 | Number of Encoder Pulses for One Motor Revolution | 0 | 0 | When power is turned ON | 439 | 00 to 15 | 0000 hex |
|  |  |  | : |  |  |  | : |
|  |  |  | 65,535 |  |  |  | FFFF hex |

## Operation Adjustment Settings

|  | Name | Default | Settings | When setting is read by CPU Unit | Internal address | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | Limit output during acceleration and constant speed | Do not use | Use | When power is turned ON | 432 | 08 to 11 | 0 hex |
|  |  |  | Do not use |  |  |  | 1 hex |
| 2 | Limit output during deceleration and when stopped | Do not use | Use | When power is turned ON | 432 | 12 to 15 | 0 hex |
|  |  |  | Do not use |  |  |  | 1 hex |
| 3 | Output coefficient during acceleration and constant speed | 0: 6 (0.01 increments) | 0: 6 (0.01 increments) | When power is turned ON | 433 | 08 to 15 | 0 hex |
|  |  |  | 1 (0.01 increments) |  |  |  | 1 hex |
|  |  |  | : |  |  |  | : |
|  |  |  | 255 (0.01 increments) |  |  |  | FF hex |


|  | Name | Default | Settings | When setting is read by CPU Unit | Internal address | Bits | Settings |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 4 | Output coefficient during deceleration | $\begin{aligned} & \hline \text { 0: } 96 \text { (0.01 } \\ & \text { increments) } \end{aligned}$ | 0: 96 (0.01 increments) | When power is turned ON | 434 | 08 to 15 | 0 hex |
|  |  |  | 1 (0.01 increments) |  |  |  | 1 hex |
|  |  |  | : |  |  |  | : |
|  |  |  | 255 (0.01 increments) |  |  |  | FF hex |
| 5 | Output coefficient after pulse output | $\begin{aligned} & \text { 0: } 50 \text { (0.01 } \\ & \text { increments) } \end{aligned}$ | 0: 50 (0.01 increments) | When power is turned ON | 435 | 08 to 15 | 0 hex |
|  |  |  | 1 (0.01 increments) |  |  |  | 1 hex |
|  |  |  | : |  |  |  | : |
|  |  |  | 255 (0.01 increments) |  |  |  | FF hex |

## Appendix H

## Specifications for External Power Supply Expansion

When using the external power supply for AC－power－supply CPU Units，the following limitations apply：

## －30－point，40－point and 60－point AC－power－supply CPU Units（Model CP1L－MロロDR－A）

The external power supply for the 30－point，40－point and 60－point AC－power－supply CPU Units（Model CP1L－MロロDR－A）can be used beyond 300 mA within the capacity range of the power supply built in the CPU Units．

Please calculate the usable capacity of external power supply according to the following example．

## Calculation example of the capacity limit of external power supply

Example 1

|  | CPU Units | Expansion（I／O）Units |  |  | Total | Limit |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | $1^{\text {st }}$ unit | $2^{\text {nd }}$ unit | $3^{\text {rd }}$ unit |  |  |
|  | Model CP1L－M40DR－A | Model CP1W－DA041 | Model CP1W－DA041 | $\begin{aligned} & \text { Model } \\ & \text { CP1W-DA041 } \end{aligned}$ |  |  |
| 5 V | 0．22A | 0．08A | 0．08A | 0．08A | 0．46A | $\leq 1.2 \mathrm{~A}$ |
| 24V | 0．08A | 0.124 A | 0.124 A | 0.124 A | 0．452A | $\leq 0.7 \mathrm{~A}$ |
| Power consumption | $\begin{aligned} & 5 \mathrm{~V} \times 0.46 \mathrm{~A}=2.3 \mathrm{~W} \\ & 24 \mathrm{~V} \times 0.452 \mathrm{~A}=10.848 \mathrm{~W} \end{aligned}$ |  |  |  | $\begin{aligned} & \hline \text { Total } \\ & 13.148 \mathrm{~W} \end{aligned}$ | $\leq 18.5 \mathrm{~W}$ |
| Usable capac－ ity of external power supply | 18.5 W （Total of usable power consumption）－13．148W＝5．352W $5.352 \mathrm{~W} / 24 \mathrm{~V}=0.223 \mathrm{~A}$ |  |  |  | 0．223A | $\leq 0.7 \mathrm{~A}$ |

Example 2

|  | CPU Units | Expansion（I／O）Units |  |  | Total | Limit |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | $1^{\text {st }}$ unit | $2^{\text {nd }}$ unit | $3^{\text {rd }}$ unit |  |  |
|  | Model CP1L－M40DR－A | Model CP1W－20EDT | None | None |  |  |
| 5 V | 0．22A | 0．130A | 0．0A | 0．0A | 0．35A | $\leq 1.2 \mathrm{~A}$ |
| 24V | 0．08A | 0．0A | 0．0A | 0．0A | 0．08A | $\leq 0.7 \mathrm{~A}$ |
| Power consumption | $\begin{aligned} & 5 \mathrm{~V} \times 0.35 \mathrm{~A}=1.75 \mathrm{~W} \\ & 24 \mathrm{~V} \times 0.08 \mathrm{~A}=1.92 \mathrm{~W} \end{aligned}$ |  |  |  | $\begin{array}{l\|} \hline \text { Total } \\ 3.67 \mathrm{~W} \end{array}$ | $\leq 18.5 \mathrm{~W}$ |
| Usable capac－ ity of external power supply | 18．5W（Total of usable power consumption）－3．67W＝14．83W $14.83 \mathrm{~W} / 24 \mathrm{~V}=0.618 \mathrm{~A}$ |  |  |  | 0．618A | $\leq 0.7 \mathrm{~A}$ |

－10－point，14－point and 20－point AC－power－supply CPU Units（Model CP1L－LロロDR－A）
The external power supply for 10－point AC－power－supply CPU Units（Model CP1L－LDIDR－A）as well as 14－point and 20－point power－supply CPU Units manufactured since May 2008 （Model CP1L－LロロDR－A） can be used beyond 200 mA within the capacity range of the power supply built in the CPU Units．

Please calculate the usable capacity of external power supply according to the following example．
Calculation example of the capacity limit of external power supply
Example 1

|  | CPU Units | Expansion（I／O）Units |  |  | Total | Limit |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | $1^{\text {st }}$ unit | $2^{\text {nd }}$ unit | $3^{\text {rd }}$ unit |  |  |
|  | Model CP1L－M20DR－A | Model CP1W－DA041 | Disabled | Disabled |  |  |
| 5V | 0．20A | 0．08A | 0．0A | 0．0A | 0．28A | $\leq 0.85 \mathrm{~A}$ |
| 24 V | 0．05A | 0．124A | 0．0A | 0．0A | 0.174 A | $\leq 0.35 \mathrm{~A}$ |
| Power consumption | $\begin{aligned} & 5 \mathrm{~V} \times 0.28 \mathrm{~A}=1.4 \mathrm{~W} \\ & 24 \mathrm{~V} \times 0.174 \mathrm{~A}=4.176 \mathrm{~W} \end{aligned}$ |  |  |  | $\begin{array}{\|l\|} \text { Total } \\ 5.576 \mathrm{~W} \end{array}$ | $\leq 9.5 \mathrm{~W}$ |
| Usable capac－ ity of external power supply | 18.5 W （Total of usable power consumption）－$-5.576 \mathrm{~W}=3.924 \mathrm{~W}$ $3.924 \mathrm{~W} / 24 \mathrm{~V}=0.163 \mathrm{~A}$ |  |  |  | 0．163A | $\leq 0.35 \mathrm{~A}$ |

Example 2

|  | CPU Units | Expansion（I／O）Units |  |  | Total | Limit |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |
|  |  | $1^{\text {st }}$ unit | $2^{\text {nd }}$ unit | $3^{\text {rd }}$ unit |  |  |
|  | Model <br> CP1L－M14DR－A | None | Disabled | Disabled |  |  |
|  | 0.18 A | 0.0 A | 0.0 A | 0.0 A | 0.18 A | $\leq 0.85 \mathrm{~A}$ |
| 24 V | 0.04 A | 0.0 A | 0.0 A | 0.0 A | 0.04 A | $\leq 0.35 \mathrm{~A}$ |
| Power <br> consumption | $5 \mathrm{~V} \times 0.18 \mathrm{~A}=0.9 \mathrm{~W}$ <br> $24 \mathrm{~V} \times 0.04 \mathrm{~A}=0.96 \mathrm{~W}$ | Total <br> 1.860 W | $\leq 9.5 \mathrm{~W}$ |  |  |  |
| Usable capac－ <br> ity of external <br> power supply | $18.5 \mathrm{~W}(T o t a l$ <br> $7.64 \mathrm{~W} / 24 \mathrm{~V}=0.318 \mathrm{~A}$ |  | 0.318 A | $\leq 0.35 \mathrm{~A}$ |  |  |

For 14－point and 20－point AC－power－supply CPU Units manufactured before May 2008，the specifications are as written in the section 1－2－3 Restrictions on System Configuration．

## A

absolute coordinates
selecting, 214
absolute pulse outputs, 214
Access Error Flag, 171
addresses
memory map, 713
Always OFF Flag, 172
Always ON Flag, 172
applications
precautions, xxvii
Auxiliary Area, 155
read/write section, 695, 707
read-only section, 665

## B

background execution
special flags, 662
backup
precautions, xxvi
Basic I/O Units
error information, 654
battery
error flag, 654, 691
precautions, xxix
service life, 624
Battery Error Flag, xxvi, 691
bit allocations
PWM(891) outputs, 250

## C

Carry Flag, 172
CIO Area, 139
clock
clock data, 655, 686
clock pulses
flags, 173
communications
Communications Port Enabled Flags, 710
flags, 659
no-protocol, 369
Communications Port Enabled Flags, 710
complete link method, 378

## Condition Flag

saving and loading status, 173
Condition Flags, 171
connectors
recommended models, 567
coordinate systems (absolute or relative), 215
Counter Area, 157
countermeasures
noise, xxxi
CPU Unit
initialization, 84
cycle time
flags, 650
maximum cycle time, 92, 650, 674
present cycle time, 92, 650, 674
setting, 92
Cycle Time Too Long Flag, 654, 690

## D

data areas
overview, 139
data links, 153
Data Registers, 169
data registers
sharing, 142
data tracing, 602
related flags/bits, 651
dates
program and parameters, 411
debugging, 598
flags, 651
DeviceNet
precaution, 417
differential phase mode
details, 178
DIP switch
pin 6 status, 643
direction
automatic direction selection, 217

## E

EC Directives, xxx
electromagnetic fields, xxvi
EMC Directives, xxx
Equals Flag, 172
error codes, 709
pulse output stop error codes, 233
Error Flag, 171
error log, 708
Error Log Area, 617, 652, 672, 708
errors
basic I/O errors, 662
communications error flags, 659
error codes, 709
error log, 652, 708
failure point detection, 415
FAL/FALS flags, 653
flags, 171
flash memory, 652
Input Units, 618
memory error flags, 652
Output Units, 619
PLC Setup errors, 662
programming error flags, 662
Special I/O Units, 662
troubleshooting, 605

## F

failure point detection, 415
FAL Error Flag, 653
FAL errors
flag, 692
FAL/FALS Number for System Error Simulation, 653
FALS Error Flag, 653
FALS errors
flag, 653, 655, 690, 691
FALS instruction, xxv, xxix
FINS commands
list, 393
First Cycle Flag, 650, 672, 709
flash memory
data dates, 656
errors, 652
Flash Memory Error, 652
flowchart
overall CPU operation, 81
Forced Status Hold Bit, 643, 695
force-resetting bits
debugging, 598
force-setting bits
debugging, 598
frequency
frequency measurement, 196

## G

Gate Bit
high-speed counters, 195
Greater Than Flag, 172
Greater Than or Equals Flag, 172

## H

High-speed Counter Gate Bit, 195
high-speed counter inputs
reset methods, 181
restrictions, 193
Holding Area, 152, 154
Host Link

$$
\text { ports, } 715
$$

Host Link commands, 391
Host Link communications, 389
hot starting, 143
hot stopping, 143

I/O Hold Bit, xxix
I/O interrupts response time, 96
I/O memory
addresses, 713
areas, 714
effects of operating mode changes, 86
I/O response time
calculating, 95
immediate refreshing
input bits and words, 83
increment mode
details, 179
index register
sharing, 168
index registers, 161
sharing, 142
indirect addressing
DM Area, 160
index registers, 161

Initial Task Execution Flag, 650
Initial Task Flag, 709
Initial Task Startup Flag, 672
initialization
CPU Unit, 84
Input Units
troubleshooting, 618
inspection
procedures, 622
installation
location, xxvi
interrupt feeding
ladder program, 279
Interrupt Input Units
response time, 96
interrupts
processing times, 651
IOM Hold Bit, 144, 643, 695
IORF(097) refreshing input bits and words, 83

IR/DR Operation between Tasks, 651

## L

Less Than Flag, 172
Less Than or Equals Flag, 172
Limit Input Signal Type, 232
linear mode counting
details, 180
Low Voltage Directive, xxx

## M

maintenance
procedures, 624

## memory

memory map, 714
See also data areas
Memory Error Flag, 652, 691
momentary power interruption, 87
MONITOR mode, 85
monitoring
differential monitoring, 599
multiple progressive positioning
ladder program, 264
wiring example, 261

## N

Negative Flag, 172
networks
related flags/bits, 659
noise, xxvi
reducing, xxx
no-protocol communications, 369
Not Equal Flag, 172
NT Link
ports, 715

## 0

online editing, 600
effect on cycle time, 94
Online Editing Flags, 672
Online Editing Wait Flag, 710
related flags/bits, 651
operating environment, xxvi
precautions, xxvi
operating modes
description, 85
effects of mode changes on counters, 158
operation
debugging, 598
trial operation, 598
Origin Compensation, 231
Origin Detection Method, 227
Origin Input Signal Type, 232
Origin Proximity Input Signal Type, 232
origin return function
details, 220
examples, 237
Origin Return Parameters, 238
origin search
executing, 232
Origin Search Acceleration Rate, 231
Origin Search Deceleration Rate, 231
Origin Search Direction specifying, 231
origin search function
details, 220
Origin Search High Speed, 231
Origin Search Operation Setting, 227
Origin Search Parameters, 224
Origin Search Proximity Speed, 231
Origin Search/Return Initial Speed, 231
origin status
operations affecting, 216
Output, 218
Output OFF Bit, 651, 695
Output Units
troubleshooting, 619
outputs
precautions, xxv, xxix
Overflow Flag, 172

## P

Parameter Date, 411, 419, 656
parts
replacing parts, 624
peripheral port
related flags/bits, 689
PLC Setup
error information, 654
Polled Units
settings, 382
Polling Unit
setting, 382
Polling Unit link method, 378
positioning
vertically conveying PCBs, 260
Positioning Monitor Time, 232
power interruptions
information, 656, 696
momentary interruptions, 87
power OFF processing, 87
power supply, xxvi
precautions, xxix
precautions, xxiii
applications, xxvii
general, xxiv
handling precautions, 623
operating environment, xxvi
periodic inspections, 622
safety, xxiv
Program Error Flag, 690
program errors, 711
PROGRAM mode, 85
program transfer, 598
programming
error flag, 690
program error information, 653
program errors, 711
program protection, 411
protecting the program, 411
transferring the program, 598
pulse + direction mode
details, 178
pulse frequency conversion, 197
pulse input modes
details, 178
Pulse Output 0 settings speed curve, 219
pulse output modes, 200
pulse output stop error codes, 233
pulse outputs, 198
PWM(891) outputs
bit allocations, 250
details, 250

## Q

quick-response inputs
details, 362

## R

radioactivity, xxvi
read/write-protection, 412
refreshing
immediate refreshing, 83
IORF(097), 83
refreshing data, 378
relative coordinates
selecting, 214
reset methods, 181
Restart Bits
peripheral port, 659
RS-232C port, 660
ring mode counting
details, 180
RS-232C port
related flags/bits, $659,660,687,688$
RUN mode, 85

## S

safety precautions, xxiv
scheduled interrupts
response time, 97

S-curve acceleration/deceleration
output pattern, 218
restrictions, 219
self-maintaining bits, 154
serial communications
functions, 366
Serial PLC Links, 378
allocated words, 381
PLC Setup, 382
related flags, 383, 385
simulating system errors, 653
software reset, 181
Special I/O Units
error information, 662
specifications
PWM(891) output, 250
startup
hot starting and stopping, 143
static electricity, xxvi
Step Flag, 672
stocker, 260

## T

target value comparison
for interrupt tasks, 194
Task Error Flag, 681
Task Flags, 171
Task Started Flag, 650
tasks
related flags/bits, 651
See also interrupt tasks
Task Flags, 171
Timer Area, 156
TR Area, 155
tracing
See also data tracing
trial operation, 598
troubleshooting, 605
twisted-pair cables
recommended models, 567
two-axis multipoint positioning
ladder program, 272
wiring example, 267

## U

Underflow Flag, 172
up/down mode
details, 179
User Program Date, 411, 419, 656

## V

variable duty ratio pulse outputs
details, 250
vertical conveyor, 260

## W

Work Area, 153
work bits, 153
work words, 153
write-protection, 411

## Revision History

A manual revision code appears as a suffix to the catalog number on the front cover of the manual.

Cat. No. W462-E1-07
Revision code

The following table outlines the changes made to the manual during each revision. Page numbers refer to the previous version.

| Revision code | Date | Revised content |
| :---: | :---: | :--- |
| 01 | May 2007 | Original production |
| 02 | June 2007 | Errors were corrected. |
| 03 | April 2008 | Information on CP1L CPU Units with 10 or 60 I/O points, CP1W Expansion I/O Units <br> with 16 or 32 I/O points and LCD Option Board were added. |
| 04 | June 2008 | Errors were corrected. |
| 05 | August 2008 | CP1W-CIF12 added for Serial Communications Option Board. <br> Errors were corrected. |
| 06 | March 2009 | Information added on Ethernet Option Board. <br> 07 May 2010 | | CP1W-DA021 added for CP-series Expansion Units. |
| :--- |
| Unit version 2.0 added for CP1W-CIF41 Ethernet Option Board. |
| Errors were corrected. |

## OMRON Corporation Industrial Automation Company

Contact: www.ia.omron.com

Regional Headquarters
OMRON EUROPE B.V.
Wegalaan 67-69-2132 JD Hoofddorp
The Netherlands
Tel: (31)2356-81-300/Fax: (31)2356-81-388
OMRON ASIA PACIFIC PTE. LTD.
No. 438A Alexandra Road \# 05-05/08 (Lobby 2),
Alexandra Technopark,
Singapore 119967
Tel: (65) 6835-3011/Fax: (65) 6835-2711

## OMRON ELECTRONICS LLC

One Commerce Drive Schaumburg,
IL 60173-5302 U.S.A.
Tel: (1) 847-843-7900/Fax: (1) 847-843-7787
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

## Authorized Distributor:

OMR Condin
© OMRON Corporation 2009 All Rights Reserved. In the interest of product improvement, specifications are subject to change without notice.


[^0]:    Instructions Used

[^1]:    This graphic is for reference only.

